
QGIS User Guide

Release 2.6

QGIS Project

May 22, 2015

1	Preâmbulo	3
2	Convenções	5
2.1	Convenções GUI	5
2.2	Convenções de Texto ou Teclado	5
2.3	Instruções específicas da plataforma	6
3	Prefácio	7
4	Características	9
4.1	Visualização de dados	9
4.2	Exploração de dados e compositores de mapas	9
4.3	Criar, editar, gerir e exportar dados	10
4.4	Análise de dados	10
4.5	Publicação de mapas na internet	10
4.6	Funcionalidades do QGIS expandida através de módulos	10
4.7	Consola Python	11
4.8	Known Issues	11
5	What's new in QGIS 2.6	13
5.1	Application and Project Options	13
5.2	Data Providers	13
5.3	Map Composer	13
5.4	QGIS Server	14
5.5	Symbology	14
5.6	User Interface	14
6	Iniciando	15
6.1	Instalação	15
6.2	Amostra de Dados	15
6.3	Sessão Amostra	16
6.4	Starting and Stopping QGIS	17
6.5	Opções da Linha de Comandos	17
6.6	Projectos	19
6.7	Ficheiro de Saída	20
7	QGIS GUI	21
7.1	Barra de Menus	22
7.2	Barra de Ferramentas	28
7.3	Legenda do Mapa	29
7.4	Vista do Mapa	31
7.5	Barra de Estado	32

8	Ferramentas gerais	33
8.1	Atalhos do teclado	33
8.2	Ajuda de contexto	33
8.3	Renderização	33
8.4	Medindo	35
8.5	Identificar elementos	37
8.6	Decorações	38
8.7	Ferramentas de Anotação	40
8.8	Marcadores espaciais	41
8.9	Nesting Projects	43
9	Configuração QGIS	45
9.1	Painéis e Barras de Ferramentas	45
9.2	Propriedades do Projecto	46
9.3	Opções	46
9.4	Personalização	55
10	Trabalhando com Projecções	57
10.1	Visão geral do Suporte a Projecções	57
10.2	Especificação Geral da Projecção	57
10.3	Definir Reprojecção Dinâmica (RD)	59
10.4	Sistema de Coordenadas personalizado	60
10.5	Transformações de datum pré-definidas	61
11	QGIS Browser	63
12	Trabalhando com Informação Vectorial	65
12.1	Supported Data Formats	65
12.2	The Symbol Library	77
12.3	Janela das Propriedades da Camada Vectorial	81
12.4	Expressions	107
12.5	Editando	114
12.6	Ferramenta de Consulta	131
12.7	Calculadora de Campos	132
13	Trabalhando com Informação Matricial	135
13.1	A trabalhar com Dados Matriciais	135
13.2	Janela de Propriedades do Projecto	136
13.3	Calculadora Matricial	144
14	Trabalhando com dados OGC	147
14.1	QGIS como Cliente de Dados OGC	147
14.2	QGIS com Servidor de Dados OGC	156
15	Trabalhando com dados GPS	161
15.1	Módulo GPS	161
15.2	Live GPS tracking	165
16	Integração GRASS SIG	171
16.1	Iniciando o módulo GRASS	171
16.2	Carregando as camadas raster e vectoriais GRASS	172
16.3	LOCALIZAÇÃO GRASS e CONJUNTO DE MAPAS	172
16.4	Importando dados para uma LOCALIZAÇÃO GRASS	175
16.5	O modelo de dados vectoriais do GRASS	175
16.6	Criando uma nova camada vectorial GRASS	176
16.7	Digitalizando e editando as camadas vectoriais GRASS	176
16.8	A ferramenta da região GRASS	179
16.9	The GRASS Toolbox	179

17	Infraestrutura do Processamento QGIS	189
17.1	Introdução	189
17.2	A caixa de ferramentas	190
17.3	O modelador gráfico	199
17.4	A interface do processamento em lote	205
17.5	Usando os algoritmos do processamento a partir da consola	207
17.6	Gestão do histórico	213
17.7	Writing new Processing algorithms as python scripts	214
17.8	Handing data produced by the algorithm	215
17.9	Communicating with the user	216
17.10	Documenting your scripts	216
17.11	Example scripts	216
17.12	Best practices for writing script algorithms	216
17.13	Pre- and post-execution script hooks	217
17.14	Configurando as aplicações externas	217
17.15	The QGIS Commander	223
18	Processing providers and algorithms	227
18.1	GDAL algorithm provider	227
18.2	LAStools	260
18.3	Modeler Tools	285
18.4	OrfeoToolbox algorithm provider	287
18.5	QGIS algorithm provider	362
18.6	R algorithm provider	416
18.7	SAGA algorithm provider	426
18.8	TauDEM algorithm provider	597
19	Compositor de Impressão	629
19.1	Primeiros passos	630
19.2	Modo de Renderização	634
19.3	Itens do Compositor	635
19.4	Manage items	658
19.5	Ferramentas de Reverter e Restaurar	660
19.6	Geração de Atlas	660
19.7	Criando um ficheiro de Saída	662
19.8	Gerir o Compositor	663
20	Módulos	665
20.1	Módulos QGIS	665
20.2	Usando os Módulos QGIS Core	670
20.3	Módulo de Captura de Coordenadas	671
20.4	Módulo Gestor BD	671
20.5	Módulo de Conversão Dxf2Shp	672
20.6	Módulo eVis	673
20.7	Módulo fTools	682
20.8	Módulo de Ferramentas GDAL	686
20.9	Módulo Georeferenciador	689
20.10	Módulo de Interpolação	694
20.11	Módulo Edição Offiline	695
20.12	Módulo Oracle Spatial GeoRaster	695
20.13	Módulo de Análise do Terreno Matricial	698
20.14	Módulo de Mapa de Densidade	699
20.15	MetaSearch Catalogue Client	703
20.16	Módulo de Cálculo de Rotas	706
20.17	Módulo de Consulta Espacial	707
20.18	Módulo SPIT	709
20.19	Módulo SQL Anywhere	709
20.20	Módulo Verificador de Topologia	711
20.21	Módulo de Estatística Zonal	712

21 Ajuda e Suporte	715
21.1 Listas de Discussão	715
21.2 IRC	716
21.3 BugTracker	716
21.4 Blogue	717
21.5 Módulos	717
21.6 Wiki	717
22 Apêndice	719
22.1 Licença Geral Pública GNU	719
22.2 Licença de Documentação Livre GNU	722
23 Literatura e Referências Web	729
Índice	731

·
·

Preâmbulo

This document is the original user guide of the described software QGIS. The software and hardware described in this document are in most cases registered trademarks and are therefore subject to legal requirements. QGIS is subject to the GNU General Public License. Find more information on the QGIS homepage, <http://www.qgis.org>.

The details, data, and results in this document have been written and verified to the best of the knowledge and responsibility of the authors and editors. Nevertheless, mistakes concerning the content are possible.

Therefore, data are not liable to any duties or guarantees. The authors, editors and publishers do not take any responsibility or liability for failures and their consequences. You are always welcome to report possible mistakes.

This document has been typeset with reStructuredText. It is available as reST source code via [github](#) and online as HTML and PDF via <http://www.qgis.org/en/docs/>. Translated versions of this document can be downloaded in several formats via the documentation area of the QGIS project as well. For more information about contributing to this document and about translating it, please visit <http://www.qgis.org/wiki/>.

Ligações neste Documento

Este documento contém ligações internas e externas. Ao clicar numa ligação interna este move dentro do documento, enquanto se clicar numa ligação externa abre um endereço de internet. No formato de PDF, as ligações internas e externas são exibidas a azul e são gerida pelo pesquisador do sistema. No formato HTML, o pesquisador exibe e gere ambas igualmente.

Utilização, Instalação, e Autores e Editores do Guia de Código:

Tara Athan	Radim Blazek	Godofredo Contreras	Otto Dassau	Martin Dobias
Peter Ersts	Anne Ghisla	Stephan Holl	N. Horning	Magnus Homann
Werner Macho	Carson J.Q. Farmer	Tyler Mitchell	K. Koy	Lars Luthman
Claudia A. Engel	Brendan Morely	David Willis	Jürgen E. Fischer	Marco Hugentobler
Larissa Junek	Diethard Jansen	Paolo Corti	Gavin Macaulay	Gary E. Sherman
Tim Sutton	Alex Bruy	Raymond Nijssen	Richard Duivenvoorde	Andreas Neumann
Astrid Emde	Yves Jacolin	Alexandre Neto	Andy Schmid	Hien Tran-Quang

Copyright (c) 2004 - 2014 QGIS Development Team

Internet: <http://www.qgis.org>

Licença deste documento


É concedida a permissão para copiar, distribuir e / ou modificar este documento sob os termos da Licença de Documentação Livre GNU, Versão 1.3 ou qualquer versão posterior publicada pela Fundação de Software Livre; sem Secções Invariantes, sem textos de Capa e sem contra-capas. A cópia da licença é incluída no Apêndice *Licença de Documentação Livre GNU*.

Convenções

Esta secção descreve uma colecção de estilos uniformizados ao longo do manual.

2.1 Convenções GUI

Os estilos da convenção GUI destinam-se a copiar a aparência do GUI. Geralmente, o objectivo é usar a aparência, para que o utilizador possa visualizar e procurar o GUI e encontrar alguma coisa parecida no manual.

- Menu Opções: *Camada* → *Adicionar Camada Raster* ou *Configurações* → *Barra de Ferramentas* → *Digitalização*
- Ferramenta:  Adicionar Camada Raster
- Botão : **[Guardar como Padrão]**
- Título da Caixa de Diálogo: *Propriedades da Camada*
- Separador: *Geral*
- Caixa de Verificação: *Renderização*
- Botão de Opção: *Postgis SRID* *EPSG ID*
- Seleccionar um número:
- Seleccionar uma cadeia de texto:
- Pesquisar por um ficheiro:
- Seleccionar uma cor:
- Slider:
- Texto de Entrada:

Uma sombra indica um componente clicável no GUI.

2.2 Convenções de Texto ou Teclado

O manual também inclui estilos relacionados com o texto, comandos do teclado e codificação que indica diferentes entidades, tais como, classes, ou métodos, Estes estilos não correspondem a qualquer aparência actual de qualquer texto ou código dentro do QGIS.

- Hiperligação: <http://qgis.org>
- Combinações de Atalho: pressione `Ctrl+B`, significa pressionar e segura a tecla `Ctrl` e de seguida pressionar a tecla `B`.

Prefácio

Bem-vindo ao mundo maravilhoso dos Sistemas de Informação Geográfica (SIG)!

O QGIS é um Sistema de Informação Geográfica de Código Aberto. O projecto nasceu em Maio de 2002 e foi estabelecido como um projecto do SourceForge em Junho do mesmo ano. Trabalhámos arduamente para fazer um software SIG (que tradicionalmente são softwares proprietários dispendiosos) com uma perspectiva viável para qualquer um com o acesso básico a um computador pessoal. O QGIS actualmente corre na maioria das plataformas Unix, Windows, e OS X. O QGIS é desenvolvido usando o toolkit do Qt (<http://qt.digia.com>) e C++. Isto significa que o QGIS mostra-se rápido no seu uso e tem uma agradável interface (GUI) e de fácil uso para o utilizador.

O QGIS tem como objectivo ser um SIG fácil de usar, fornecendo funções e características comuns. O objectivo inicial foi fornecer um SIG de visualização de dados. O QGIS chegou ao ponto na sua evolução onde está a ser usado por muitos na sua necessidade diária de visualização de dados SIG. O QGIS suporta um número de formatos de dados vectoriais e raster, e suporta facilmente a adição de novos formatos usando a arquitectura de módulos.

O QGIS foi lançado segundo a GNU Licença Pública Geral (GPL). Desenvolver o QGIS segundo esta licença significa que pode inspeccionar e modificar o código fonte, e garantir que você, o nosso utilizador contente, terá sempre acesso a um programa SIG que é gratuito e pode ser gratuitamente modificado. Deve receber uma cópia completa da licença com a sua cópia do QGIS, e pode também encontrá-la no Apêndice *Licença Geral Pública GNU*.

Tip: Documentação Actualizada

A última versão deste documento pode ser sempre encontrado na área de documentação no sítio da internet do QGIS em <http://www.qgis.org/en/docs/>.

Características

QGIS offers many common GIS functionalities provided by core features and plugins. A short summary of six general categories of features and plugins is presented below, followed by first insights into the integrated Python console.

4.1 Visualização de dados

Pode ver ou sobrepor dados vectoriais e matriciais em diferentes formatos e projecções sem conversão para um formato interno ou comum. Os formatos suportados incluídos são:

- Spatially-enabled tables and views using PostGIS, SpatiaLite and MS SQL Spatial, Oracle Spatial, vector formats supported by the installed OGR library, including ESRI shapefiles, MapInfo, SDTS, GML and many more. See section *Trabalhando com Informação Vectorial*.
- Raster and imagery formats supported by the installed GDAL (Geospatial Data Abstraction Library) library, such as GeoTIFF, ERDAS IMG, ArcInfo ASCII GRID, JPEG, PNG and many more. See section *Trabalhando com Informação Matricial*.
- GRASS raster and vector data from GRASS databases (location/mapset). See section *Integração GRASS SIG*.
- Online spatial data served as OGC Web Services, including WMS, WMTS, WCS, WFS, and WFS-T. See section *Trabalhando com dados OGC*.

4.2 Exploração de dados e compositores de mapas

You can compose maps and interactively explore spatial data with a friendly GUI. The many helpful tools available in the GUI include:

- Pesquisador QGIS
- Reprojecção On-the-fly
- Gestor BD
- Compositor de Mapas
- Painel de Vista Global
- Marcadores espaciais
- Ferramentas de anotação
- Identificar/seleccionar elementos
- Editar/ver/procurar atributos
- Data-defined feature labeling

- Data-defined vector and raster symbology tools
- Atlas map composition with graticule layers
- North arrow scale bar and copyright label for maps
- Support for saving and restoring projects

4.3 Criar, editar, gerir e exportar dados

You can create, edit, manage and export vector and raster layers in several formats. QGIS offers the following:

- Digitizing tools for OGR-supported formats and GRASS vector layers
- Ability to create and edit shapefiles and GRASS vector layers
- Georeferencer plugin to geocode images
- GPS tools to import and export GPX format, and convert other GPS formats to GPX or down/upload directly to a GPS unit (On Linux, usb: has been added to list of GPS devices.)
- Support for visualizing and editing OpenStreetMap data
- Ability to create spatial database tables from shapefiles with DB Manager plugin
- Tratamento melhorado de tabelas de bases de dados espaciais
- Tools for managing vector attribute tables
- Option to save screenshots as georeferenced images

4.4 Análise de dados

You can perform spatial data analysis on spatial databases and other OGR- supported formats. QGIS currently offers vector analysis, sampling, geoprocessing, geometry and database management tools. You can also use the integrated GRASS tools, which include the complete GRASS functionality of more than 400 modules. (See section *Integração GRASS SIG.*) Or, you can work with the Processing Plugin, which provides a powerful geospatial analysis framework to call native and third-party algorithms from QGIS, such as GDAL, SAGA, GRASS, fTools and more. (See section *Introdução.*)

4.5 Publicação de mapas na internet

QGIS can be used as a WMS, WMTS, WMS-C or WFS and WFS-T client, and as a WMS, WCS or WFS server. (See section *Trabalhando com dados OGC.*) Additionally, you can publish your data on the Internet using a webserver with UMN MapServer or GeoServer installed.

4.6 Funcionalidades do QGIS expandida através de módulos

QGIS can be adapted to your special needs with the extensible plugin architecture and libraries that can be used to create plugins. You can even create new applications with C++ or Python!

4.6.1 Módulos Core

Core plugins include:

1. Coordinate Capture (Capture mouse coordinates in different CRSs)
2. Gestor BD (Troca, edição, e visualização de camadas e tabelas; execução de consultas SQL)

3. Diagram Overlay (Place diagrams on vector layers)
4. Dxf2Shp Converter (Convert DXF files to shapefiles)
5. eVIS (Visualize events)
6. fTools (Analyze and manage vector data)
7. GDALTools (Integração das Ferramentas GDAL no QGIS)
8. Georeferencer GDAL (Add projection information to rasters using GDAL)
9. GPS Tools (Load and import GPS data)
10. GRASS (Integrate GRASS GIS)
11. Heatmap (Generate raster heatmaps from point data)
12. Interpolation Plugin (Interpolate based on vertices of a vector layer)
13. Offline Editing (Allow offline editing and synchronizing with databases)
14. Oracle GeoRaster Espacial
15. Processamento (anteriormente designado de SEXTANTE)
16. Raster Terrain Analysis (Analyze raster-based terrain)
17. Road Graph Plugin (Analyze a shortest-path network)
18. Módulo de Consulta Espacial
19. SPIT (Import shapefiles to PostgreSQL/PostGIS)
20. Módulo SQL Anywhere (Armazena camadas vectoriais dentro de base de dados SQL Anywhere)
21. Topology Checker (Find topological errors in vector layers)
22. Zonal Statistics Plugin (Calculate count, sum, and mean of a raster for each polygon of a vector layer)

4.6.2 Módulos Externos Python

QGIS offers a growing number of external Python plugins that are provided by the community. These plugins reside in the official Plugins Repository and can be easily installed using the Python Plugin Installer. See Section *The Plugins Dialog*.

4.7 Consola Python

For scripting, it is possible to take advantage of an integrated Python console, which can be opened from menu: *Plugins* → *Python Console*. The console opens as a non-modal utility window. For interaction with the QGIS environment, there is the `qgis.utils iface` variable, which is an instance of `QgsInterface`. This interface allows access to the map canvas, menus, toolbars and other parts of the QGIS application.

For further information about working with the Python console and programming QGIS plugins and applications, please refer to http://www.qgis.org/html/en/docs/pyqgis_developer_cookbook/index.html.

4.8 Known Issues

4.8.1 Number of open files limitation

If you are opening a large QGIS project and you are sure that all layers are valid, but some layers are flagged as bad, you are probably faced with this issue. Linux (and other OSs, likewise) has a limit of opened files by process. Resource limits are per-process and inherited. The `ulimit` command, which is a shell built-in, changes the limits only for the current shell process; the new limit will be inherited by any child processes.

You can see all current ulimit info by typing

```
user@host:~$ ulimit -aS
```

You can see the current allowed number of opened files per process with the following command on a console

```
user@host:~$ ulimit -Sn
```

To change the limits for an **existing session**, you may be able to use something like

```
user@host:~$ ulimit -Sn #number_of_allowed_open_files
user@host:~$ ulimit -Sn
user@host:~$ qgis
```

To fix it forever

On most Linux systems, resource limits are set on login by the `pam_limits` module according to the settings contained in `/etc/security/limits.conf` or `/etc/security/limits.d/*.conf`. You should be able to edit those files if you have root privilege (also via `sudo`), but you will need to log in again before any changes take effect.

More info:

<http://www.cyberciti.biz/faq/linux-increase-the-maximum-number-of-open-files/> <http://linuxaria.com/article/open-files-in-linux?lang=en>

What's new in QGIS 2.6

This release contains new features and extends the programmatic interface over previous versions. We recommend that you use this version over previous releases.

This release includes hundreds of bug fixes and many new features and enhancements that will be described in this manual. You may also review the visual changelog at <http://changelog.linfiniti.com/qgis/version/2.6.0/>.

5.1 Application and Project Options

- **Project filename in properties:** You can now see the full path for the QGIS project file in the project properties dialog.

5.2 Data Providers

- **DXF Export tool improvements:**
 - Tree view and attribute selection for layer assignment in dialog
 - support fill polygons/HATCH
 - represent texts as MTEXT instead of TEXT (including font, slant and weight)
 - support for RGB colors when there's no exact color match
 - use AutoCAD 2000 DXF (R15) instead of R12

5.3 Map Composer

- **Update map canvas extent from map composer extent:** **On the Item** properties of a Map element there are now two extra buttons which allow you to (1) set the Map canvas extent according with the extent of your Map element and (2) view in Map canvas the extent currently set on your Map element.
- **Multiple grid support:** It is now possible to have more than one grid in your Map element. Each grid is fully customizable and can be assigned to a different CRS. This means, for example, you can now have a map layout with both geographic and projected grids.
- **Selective export:** To every item of your map composer layout, under Rendering options, you may exclude that object from map exports.

5.4 QGIS Server

5.5 Symbology

5.6 User Interface

Iniciando

This chapter gives a quick overview of installing QGIS, some sample data from the QGIS web page, and running a first and simple session visualizing raster and vector layers.

6.1 Instalação

Installation of QGIS is very simple. Standard installer packages are available for MS Windows and Mac OS X. For many flavors of GNU/Linux, binary packages (rpm and deb) or software repositories are provided to add to your installation manager. Get the latest information on binary packages at the QGIS website at <http://download.qgis.org>.

6.1.1 Instalação à partir da fonte


If you need to build QGIS from source, please refer to the installation instructions. They are distributed with the QGIS source code in a file called `INSTALL`. You can also find them online at <http://htmlpreview.github.io/?https://raw.githubusercontent.com/qgis/QGIS/master/doc/INSTALL.html>

6.1.2 Instalação no dispositivo de armazenamento externo


QGIS allows you to define a `--configpath` option that overrides the default path for user configuration (e.g., `~/.qgis2` under Linux) and forces **QSettings** to use this directory, too. This allows you to, for instance, carry a QGIS installation on a flash drive together with all plugins and settings. See section *Menu Sistema* for additional information.

6.2 Amostra de Dados

O guia de utilizador contém exemplos baseados no conjunto de amostra de dados do QGIS.

 The Windows installer has an option to download the QGIS sample dataset. If checked, the data will be downloaded to your `My Documents` folder and placed in a folder called `GIS Database`. You may use Windows Explorer to move this folder to any convenient location. If you did not select the checkbox to install the sample dataset during the initial QGIS installation, you may do one of the following:

- Use GIS data that you already have
- Download sample data from http://download.osgeo.org/qgis/data/qgis_sample_data.zip
- Uninstall QGIS and reinstall with the data download option checked (only recommended if the above solutions are unsuccessful)

 **X** For GNU/Linux and Mac OS X, there are not yet dataset installation packages available as rpm, deb or dmg. To use the sample dataset, download the file `qgis_sample_data` as a ZIP archive from http://download.osgeo.org/qgis/data/qgis_sample_data.zip and unzip the archive on your system.

The Alaska dataset includes all GIS data that are used for examples and screenshots in the user guide; it also includes a small GRASS database. The projection for the QGIS sample dataset is Alaska Albers Equal Area with units feet. The EPSG code is 2964.




```
PROJCS["Albers Equal Area",
GEOGCS["NAD27",
DATUM["North_American_Datum_1927",
SPHEROID["Clarke 1866",6378206.4,294.978698213898,
AUTHORITY["EPSG","7008"]],
TOWGS84[-3,142,183,0,0,0,0],
AUTHORITY["EPSG","6267"]],
PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],
UNIT["degree",0.0174532925199433,
AUTHORITY["EPSG","9108"]],
AUTHORITY["EPSG","4267"]],
PROJECTION["Albers_Conic_Equal_Area"],
PARAMETER["standard_parallel_1",55],
PARAMETER["standard_parallel_2",65],
PARAMETER["latitude_of_center",50],
PARAMETER["longitude_of_center",-154],
PARAMETER["false_easting",0],
PARAMETER["false_northing",0],
UNIT["us_survey_feet",0.3048006096012192]]
```

If you intend to use QGIS as a graphical front end for GRASS, you can find a selection of sample locations (e.g., Spearfish or South Dakota) at the official GRASS GIS website, <http://grass.osgeo.org/download/sample-data/>.




6.3 Sessão Amostra





Now that you have QGIS installed and a sample dataset available, we would like to demonstrate a short and simple QGIS sample session. We will visualize a raster and a vector layer. We will use the landcover raster layer, `qgis_sample_data/raster/landcover.img`, and the lakes vector layer, `qgis_sample_data/gml/lakes.gml`.

6.3.1 Iniciar o QGIS

-  Start QGIS by typing “QGIS” at a command prompt, or if using a precompiled binary, by using the Applications menu.
-  Inicie o QGIS usando o menu Iniciar ou o atalho do ambiente de trabalho, ou faça duplo clique no ficheiro de projecto QGIS.
-  Faça duplo clique no ícone na sua pasta de Aplicações.

6.3.2 Carregue camadas vectoriais ou raster a partir do conjunto de amostras de dados




1. Clique no ícone  Carregar Raster
2. Browse to the folder `qgis_sample_data/raster/`, select the ERDAS IMG file `landcover.img` and click **[Open]**.
3. If the file is not listed, check if the *Files of type*  combo box at the bottom of the dialog is set on the right type, in this case “Erdas Imagine Images (*.img, *.IMG)”.
4. Agora clique no ícone  Carregar Vector

5.  *File* should be selected as *Source Type* in the new *Add vector layer* dialog. Now click [**Browse**] to select the vector layer.
6. Browse to the folder `qgis_sample_data/gml/`, select 'Geography Markup Language [GML] [OGR] (.gml,.GML)' from the *Files of type*  combo box, then select the GML file `lakes.gml` and click [**Open**]. In the *Add vector layer* dialog, click [**OK**]. The *Coordinate Reference System Selector* dialog opens with *NAD27 / Alaska Albers* selected, click [**OK**].
7. Amplie um pouco até à sua área favorita com alguns lagos.
8. Faça duplo clique na camada `lakes` da legenda do mapa para abrir o diálogo *Propriedades*
9. Click on the *Style* tab and select a blue as fill color.
10. Click on the *Labels* tab and check the  *Label this layer with* checkbox to enable labeling. Choose the "NAMES" field as the field containing labels.
11. To improve readability of labels, you can add a white buffer around them by clicking "Buffer" in the list on the left, checking  *Draw text buffer* and choosing 3 as buffer size.
12. Click [**Apply**]. Check if the result looks good, and finally click [**OK**].

You can see how easy it is to visualize raster and vector layers in QGIS. Let's move on to the sections that follow to learn more about the available functionality, features and settings, and how to use them.


6.4 Starting and Stopping QGIS

In section *Sessão Amostra* you already learned how to start QGIS. We will repeat this here, and you will see that QGIS also provides further command line options.

-  Assuming that QGIS is installed in the PATH, you can start QGIS by typing `qgis` at a command prompt or by double clicking on the QGIS application link (or shortcut) on the desktop or in the Applications menu.
-  Inicie o QGIS usando o menu Iniciar ou o atalho do ambiente de trabalho, ou faça duplo clique no ficheiro de projecto QGIS.
-  Double click the icon in your Applications folder. If you need to start QGIS in a shell, run `/path-to-installation-executable/Contents/MacOS/Qgis`.

To stop QGIS, click the menu option   **File X QGIS** → *Quit*, or use the shortcut `Ctrl+Q`.

6.5 Opções da Linha de Comandos

 QGIS supports a number of options when started from the command line. To get a list of the options, enter `qgis --help` on the command line. The usage statement for QGIS is:

```
qgis --help
QGIS - 2.6.0-Brighton 'Brighton' (exported)
QGIS is a user friendly Open Source Geographic Information System.
Usage: /usr/bin/qgis.bin [OPTION] [FILE]
OPTION:
  [--snapshot filename]  emit snapshot of loaded datasets to given file
  [--width width]        width of snapshot to emit
  [--height height]      height of snapshot to emit
  [--lang language]      use language for interface text
  [--project projectfile] load the given QGIS project
  [--extent xmin,ymin,xmax,ymax] set initial map extent
  [--nologo]             hide splash screen
  [--noplugins]          don't restore plugins on startup
  [--nocustomization]    don't apply GUI customization
```

```
[--customizationfile] use the given ini file as GUI customization
[--optionspath path] use the given QSettings path
[--configpath path] use the given path for all user configuration
[--code path] run the given python file on load
[--defaultui] start by resetting user ui settings to default
[--help] this text
```

FILE:

Files specified on the command line can include rasters, vectors, and QGIS project files (.qgs):

1. Rasters - supported formats include GeoTiff, DEM and others supported by GDAL
2. Vectors - supported formats include ESRI Shapefiles and others supported by OGR and PostgreSQL layers using the PostGIS extension

Tip: Exemplo do Uso dos argumentos da linha de comandos

You can start QGIS by specifying one or more data files on the command line. For example, assuming you are in the `qgis_sample_data` directory, you could start QGIS with a vector layer and a raster file set to load on startup using the following command: `qgis ./raster/landcover.img ./gml/lakes.gml`

Opção da linha de comandos `--snapshot`

Esta opção permite que possa criar uma captura de ecrã no formato PNG da vista actual. Isto vem a calhar quando tem vários projectos e quer gerar capturas de ecrã dos seus dados.

Currently, it generates a PNG file with 800x600 pixels. This can be adjusted using the `--width` and `--height` command line arguments. A filename can be added after `--snapshot`.

Opção da linha de comandos `--lang`

Based on your locale, QGIS selects the correct localization. If you would like to change your language, you can specify a language code. For example, `--lang=it` starts QGIS in italian localization. A list of currently supported languages with language code and status is provided at http://hub.qgis.org/wiki/quantum-gis/GUI_Translation_Progress.

Opção da linha de comandos `--project`

Starting QGIS with an existing project file is also possible. Just add the command line option `--project` followed by your project name and QGIS will open with all layers in the given file loaded.

Opção da linha de comandos `--extent`

To start with a specific map extent use this option. You need to add the bounding box of your extent in the following order separated by a comma:

```
--extent xmin,ymin,xmax,ymax
```

Opção da linha de comandos `--nologo`

This command line argument hides the splash screen when you start QGIS.

Opção da linha de comandos `--noplugins`

If you have trouble at start-up with plugins, you can avoid loading them at start-up with this option. They will still be available from the Plugins Manager afterwards.

Command line option `--customizationfile`

Using this command line argument, you can define a GUI customization file, that will be used at startup.

Opção da linha de comandos `--nocustomization`

Using this command line argument, existing GUI customization will not be applied at startup.

Opção da linha de comandos `--optionspath`

You can have multiple configurations and decide which one to use when starting QGIS with this option. See *Opções* to confirm where the operating system saves the settings files. Presently, there is no way to specify a file to write settings to; therefore, you can create a copy of the original settings file and rename it. The option specifies path to directory with settings. For example, to use `/path/to/config/QGIS/QGIS2.ini` settings file, use option:

```
--optionspath /path/to/config/
```

Opção da linha de comandos `--configpath`

This option is similar to the one above, but furthermore overrides the default path for user configuration (`~/.qgis2`) and forces **QSettings** to use this directory, too. This allows users to, for instance, carry a QGIS installation on a flash drive together with all plugins and settings.

Command line option `--code`



This option can be used to run a given python file directly after QGIS has started.


For example, when you have a python file named `load_alaska.py` with following content:


```
from qgis.utils import iface
raster_file = "/home/gisadmin/Documents/qgis_sample_data/raster/landcover.img"
layer_name = "Alaska"
iface.addRasterLayer(raster_file, layer_name)
```

Assuming you are in the directory where the file `load_alaska.py` is located, you can start QGIS, load the raster file `landcover.img` and give the layer the name 'Alaska' using the following command: `qgis --code load_alaska.py`

6.6 Projectos

The state of your QGIS session is considered a project. QGIS works on one project at a time. Settings are considered as being either per-project or as a default for new projects (see section *Opções*). QGIS can save the state of your workspace into a project file using the menu options *Project* →  *Save* or *Project* →  *Save As...*

Load saved projects into a QGIS session using *Project* →  *Open...*, *Project* → *New from template* or *Project* → *Open Recent* →.

If you wish to clear your session and start fresh, choose *Project* →  *New*. Either of these menu options will prompt you to save the existing project if changes have been made since it was opened or last saved.

O tipo de informação guardado num ficheiro de projecto inclui:

- Camadas adicionadas
- Propriedades da Camada, incluindo a simbolização
- Projecção para a vista do mapa
- Última extensão visualizada



The project file is saved in XML format, so it is possible to edit the file outside QGIS if you know what you are doing. The file format has been updated several times compared with earlier QGIS versions. Project files from older QGIS versions may not work properly anymore. To be made aware of this, in the *General* tab under *Settings* → *Options* you can select:

- *Avisar para guardar o projecto e alterações de fontes de dados quando necessário*
- *Avisar quando abrir um projecto guardado com uma versão antiga do |qg|*

Whenever you save a project in QGIS 2.2 now a backup of the project file is made.

6.7 Ficheiro de Saída

There are several ways to generate output from your QGIS session. We have discussed one already in section *Projectos*, saving as a project file. Here is a sampling of other ways to produce output files:

- A opção do menu *Projecto* →  Guardar como Imagem abre um diálogo de ficheiro onde pode seleccionar nome, caminho e tipo de imagem (formato PNG ou JPG). Um ficheiro world file com a extensão PNGW ou JPGW é guardado na mesma pasta georeferenciando a imagem.
- Menu option *Project* → *DXF Export ...* opens a dialog where you can define the ‘Symbology mode’, the ‘Symbology scale’ and vector layers you want to export to DXF.
- Menu option *Project* →  *New Print Composer* opens a dialog where you can lay out and print the current map canvas (see section *Compositor de Impressão*).

QGIS GUI

Quando o QGIS inicia, é apresentado um GUI como é demonstrado na figura: (os números de 1 a 5 em círculos amarelos referem-se a cinco grandes áreas da interface estão abordados em baixo):

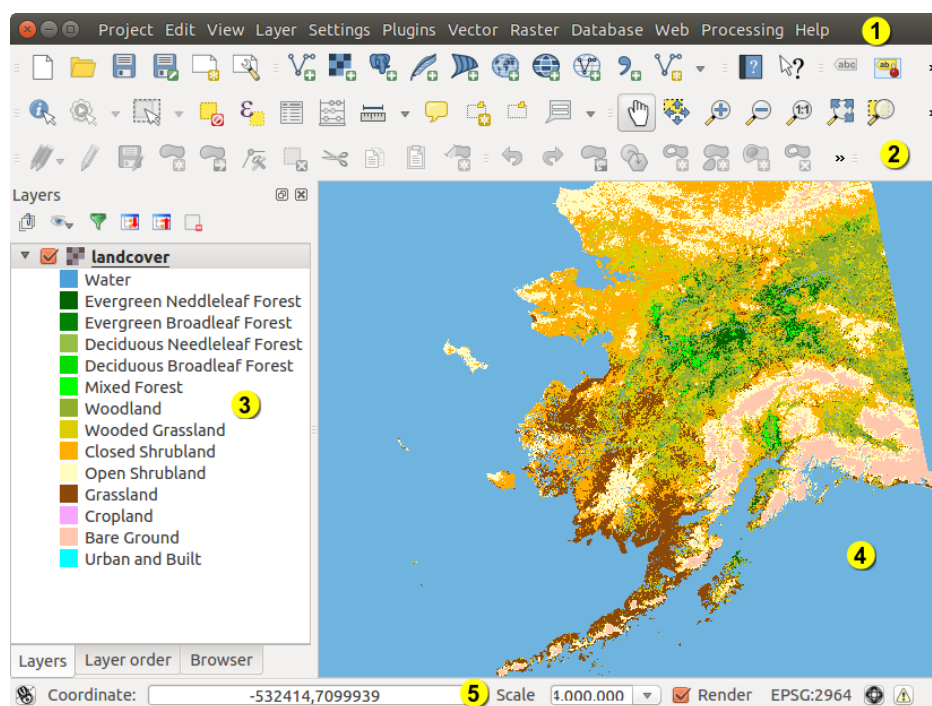


Figura 7.1: GUI do QGIS com dados amostra do Alaska 🐧

Note: As decorações na sua janela (Barra de Títulos, etc.) podem aparecer diferente maneira dependendo do seu sistema operativo e gestor de janelas.

O GUI do QGIS é dividido em cinco áreas:

1. Barra de Menus
2. Barra de Ferramentas
3. Legenda do Mapa
4. Vista do Mapa
5. Barra de Estado









Estes cinco componentes da interface do QGIS são descritos em mais detalhe nas seguintes secções. Mais duas secções apresentam atalhos de teclado e ajuda do contexto.

7.1 Barra de Menus























A barra de menus fornece o acesso a várias características QGIS usando um menu hierárquico padrão. Os menus de topo e um sumário de algumas opções dos menus estão listados em baixo, juntamente com os ícones correspondentes às ferramentas como aparecem na barra de ferramentas, assim como os atalhos de teclado. Os atalhos de teclado podem ser configurados manualmente usando o diálogo *Configurar Atalhos* aberto a partir do *Configurações* → *Configurar Atalhos...*


Embora a maioria das opções do menu ter ferramentas correspondentes e vice-versa, os menus não estão organizados como as barras de ferramentas. A barra de ferramenta que contém a ferramenta é listada depois de cada opção de menu como uma entrada de caixa de verificação. Algumas opções de menu apenas aparecem se o módulo correspondente estiver carregado. Para mais informação sobre as ferramentas e barra de ferramentas, veja secção *Barra de Ferramentas*.

7.1.1 Projecto




Menu de Opções	Atalho	Referência	Barra de Ferramentas
 <i>Novo</i>	Ctrl+N	veja <i>Projectos</i>	<i>Projecto</i>
 <i>Abrir</i>	Ctrl+O	veja <i>Projectos</i>	<i>Projecto</i>
<i>Novo a partir do modelo</i> →		veja <i>Projectos</i>	<i>Projecto</i>
<i>Abrir Recente</i> →		veja <i>Projectos</i>	
 <i>Guardar</i>	Ctrl+S	veja <i>Projectos</i>	<i>Projecto</i>
 <i>Guardar Como...</i>	Ctrl+Shift+S	veja <i>Projectos</i>	<i>Projecto</i>
 <i>Guardar como Imagem...</i>		see <i>Ficheiro de Saída</i>	
<i>Exportar como DXF ...</i>		see <i>Ficheiro de Saída</i>	
 <i>Novo Compositor de Impressão</i>	Ctrl+P	veja <i>Compositor de Impressão</i>	<i>Projecto</i>
 <i>Gestor de compositores ...</i>		veja <i>Compositor de Impressão</i>	<i>Projecto</i>
<i>Imprimir Compositores</i> →		veja <i>Compositor de Impressão</i>	
 <i>Sair do lqgl</i>	Ctrl+Q		

7.1.2 Editar
















Menu de Opções	Atalho	Referência	Barra de Ferramentas
 <i>Retroceder</i>	Ctrl+Z	veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Refazer</i>	Ctrl+Shift+Z	veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Cortar Elementos</i>	Ctrl+X	veja <i>Digitalizar uma camada existente</i>	<i>Digitalização</i>
 <i>Copiar Elementos</i>	Ctrl+C	veja <i>Digitalizar uma camada existente</i>	<i>Digitalização</i>
 <i>Colar Elementos</i>	Ctrl+V	veja <i>Digitalizar uma camada existente</i>	<i>Digitalização</i>
<i>Colar elementos como →</i>		Veja <i>Trabalhando com a Tabela de Atributos</i>	
 <i>Adicionar Elemento</i>	Ctrl+.	veja <i>Digitalizar uma camada existente</i>	<i>Digitalização</i>
 <i>Mover Elemento(s)</i>		veja <i>Digitalizar uma camada existente</i>	<i>Digitalização</i>
 <i>Apagar Seleccionados</i>		veja <i>Digitalizar uma camada existente</i>	<i>Digitalização</i>
 <i>Rodar Elemento(s)</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Simplificar Elemento</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Adicionar Anel</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Adicionar Parte</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Preencher Anel</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Apagar Anel</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Apagar Parte</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Redesenhar Elementos</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Curva de Afastamento</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Dividir Elementos</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Dividir Partes</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Juntar Elementos Seleccionados</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Juntar Atributos dos Elementos Seleccionados</i>		veja <i>Digitalização Avançada</i>	<i>Digitalização Avançada</i>
 <i>Ferramenta de Nós</i>		veja <i>Digitalizar uma camada existente</i>	<i>Digitalização</i>

Após activar o  Modo de edição para uma camada, irá encontrar o ícone Adicionar Elemento no menu *Editar* dependendo do tipo de camada (ponto, linha ou polígono).






















7.1.3 Editar (extra)

Menu de Opções	Atalho	Referência	Barra de Ferramentas
 Adicionar Elemento		veja <i>Digitalizar uma camada existente</i>	Digitalização
 Adicionar Elemento		veja <i>Digitalizar uma camada existente</i>	Digitalização
 Adicionar Elemento		veja <i>Digitalizar uma camada existente</i>	Digitalização






7.1.4 Ver

Menu de Opções	Atalho	Referência	Barra de Ferramentas
 Mover Mapa			Navegação no Mapa
 Ajustar Mapa à Selecção			Navegação no Mapa
 Aproximar	Ctrl++		Navegação no Mapa
 Afastar	Ctrl+-		Navegação no Mapa
Seleccionar →		veja <i>Seleccionar e desseleccionar elementos</i>	Atributos
 Identificar Elementos	Ctrl+Shift+I		Atributos
Medição →		veja <i>Medindo</i>	Atributos
 Ver Tudo	Ctrl+Shift+F		Navegação no Mapa
 Aproximar à Camada			Navegação no Mapa
 Aproximar à Selecção	Ctrl+J		Navegação no Mapa
 Última Vista			Navegação no Mapa
 Próxima Vista			Navegação no Mapa
 Aproximar à Resolução Natural			Navegação no Mapa
Decorações →		veja <i>Decorações</i>	
 Dicas de Mapa			Atributos
 Novo Marcador	Ctrl+B	veja <i>Marcadores espaciais</i>	Atributos
 Mostrar Marcadores	Ctrl+Shift+B	veja <i>Marcadores espaciais</i>	Atributos
 Actualizar	Ctrl+R		Navegação no Mapa


7.1.5 Camada

Menu de Opções	Atalho	Referência	Barra de Ferramentas
Novo →		veja <i>Criando novas camadas Vectoriais</i>	Gerir Camada
Incorporar Camadas e Grupos ...		veja <i>Nesting Projects</i>	
 Adicionar Camada Vectorial	Ctrl+Shift+V	veja <i>Trabalhando com Informação Vectorial</i>	Gerir Camada
 Adicionar Camada Matricial	Ctrl+Shift+R	veja <i>Carregar dados raster no QGIS</i>	Gerir Camada
 Adicionar Camada PostGIS	Ctrl+Shift+D	veja <i>PostGIS Layers</i>	Gerir Camada
 Adicionar Camada SpatialLite	Ctrl+Shift+L	veja <i>SpatialLite Layers</i>	Gerir Camada
 Adicionar Camada Espacial MSSQL	Ctrl+Shift+M	veja <i>MSSQL Spatial Layers</i>	Gerir Camada
 Adicionar camada Oracle GeoRaster		veja <i>Módulo Oracle Spatial GeoRaster</i>	Gerir Camada
 Adicionar camada SQL Anywhere		veja <i>Módulo SQL Anywhere</i>	Gerir Camada
 Adicionar Camada WMS/WMTS	Ctrl+Shift+W	veja <i>Cliente WMS/WMTS</i>	Gerir Camada
 Adicionar Camada WCS		veja <i>WCS Cliente</i>	Gerir Camada
 Adicionar Camada WFS		veja <i>WFS e WFS-T Cliente</i>	Gerir Camada
 Adicionar Camada de Texto Delimitado		veja <i>Delimited Text Files</i>	Gerir Camada
 Copiar estilo		veja <i>Estilos</i>	
 Colar Estilo		veja <i>Estilos</i>	
 Abrir Tabela de Atributos		Veja <i>Trabalhando com a Tabela de Atributos</i>	Atributos
 Opções de Ressalto		veja <i>Digitalizar uma camada existente</i>	Digitalização
 Guardar edições da camada		veja <i>Digitalizar uma camada existente</i>	Digitalização
 Edições Actuais →		veja <i>Digitalizar uma camada existente</i>	Digitalização
Guardar como...			
Guardar selecção como ficheiro vectorial...		Veja <i>Trabalhando com a Tabela de Atributos</i>	
 Remover Camada(s)	Ctrl+D		
 Duplicar Camada (s)			
Definir SRC da Camada(s)	Ctrl+Shift+C		
Definir SRC do projecto a partir da Camada			
Propriedades			
Consultar...			
 Rotular			
 Adicionar ao Enquadramento	Ctrl+Shift+O		Gerir Camada
 Adicionar Tudo ao Enquadramento			
 Remover Tudo do Enquadramento			
 Mostrar Todas as Camadas	Ctrl+Shift+U		Gerir Camada
 Ocultar Todas as Camadas	Ctrl+Shift+H		Gerir Camada

7.1.6 Configurações






Menu de Opções	Atalho	Referência	Barra de Ferramentas
<i>Painéis →</i> <i>Barra de Ferramentas →</i> <i>Activar Modo de Ecrã Completo</i>  <i>Propriedades do Projecto ...</i>  <i>SRC Personalizado ...</i> <i>Gestor de Estilo...</i>  <i>Configurar atalhos ...</i>  <i>Personalização ...</i>  <i>Opções ...</i> <i>Opções de Ressalto ...</i>	 F 11 Ctrl+Shift+P	<i>veja Painéis e Barras de Ferramentas</i> <i>veja Painéis e Barras de Ferramentas</i> <i>veja Projectos</i> <i>veja Sistema de Coordenadas personalizado</i> <i>veja Presentation</i> <i>veja Personalização</i> <i>veja Opções</i>	

7.1.7 Módulos

Menu de Opções	Atalho	Referência	Barra de Ferramentas
 <i>Gerir e Instalar Módulos</i> <i>Consola Python</i>		<i>veja The Plugins Dialog</i>	

O começo do QGIS pela primeira vez não carrega todos os módulos core.

7.1.8 Vector

Menu de Opções	Atalho	Referência	Barra de Ferramentas
<i>Open Street Map →</i>  <i>Ferramentas de Análise →</i>  <i>Ferramentas de Investigação →</i>  <i>Ferramentas de Geoprocessamento →</i>  <i>Ferramentas de Geometria →</i>  <i>Ferramenta de Gestão de Dados →</i>		<i>veja Loading OpenStreetMap Vectors</i> <i>veja Módulo fTools</i> <i>veja Módulo fTools</i> <i>veja Módulo fTools</i> <i>veja Módulo fTools</i> <i>veja Módulo fTools</i>	







O começo do QGIS pela primeira vez não carrega todos os módulos core.

7.1.9 Matricial

Menu de Opções	Atalho	Referência	Barra de Ferramentas
<i>Calculadora Raster ...</i>		<i>veja Calculadora Matricial</i>	







O começo do QGIS pela primeira vez não carrega todos os módulos core.


7.1.10 Processamento





Menu de Opções	Atalho	Referência	Barra de Ferramentas
 <i>Caixa de Ferramentas</i>		veja <i>A caixa de ferramentas</i>	
 <i>Modelador Gráfico</i>		veja <i>O modelador gráfico</i>	
 <i>Histórico e registo</i>		veja <i>Gestão do histórico</i>	
 <i>Opções e configurações</i>		veja <i>Configurando a infraestrutura do processamento</i>	
 <i>Visualizador de resultados</i>		veja <i>Configurando as aplicações externas</i>	
 <i>Comandos</i>	Ctrl+Alt+M	veja <i>The QGIS Commander</i>	

O começo do QGIS pela primeira vez não carrega todos os módulos core.

7.1.11 Ajuda

Menu de Opções	Atalho	Referência	Barra de Ferramentas
 <i>Conteúdos da Ajuda</i>	F1		<i>Ajuda</i>
 <i>O que é isto?</i> <i>Documentação API</i> <i>Necessita suporte comercial?</i>	Shift+F1		<i>Ajuda</i>
 <i>Sítio na Internet do QGIS</i>	Ctrl+H		
 <i>Verificar a Versão qg </i>			
 <i>Sobre</i>			
 <i>Patrocinadores qg </i>			

Repare que para o Linux , os itens da Barra de Menus estão listados acima são os padrões para o gestor de janelas no KDE. No GNOME, o menu *Configurações* tem conteúdos diferentes e os itens são encontrados aqui:

 <i>Propriedades do Projecto</i>	<i>Projecto</i>
 <i>Opções</i>	<i>Editar</i>
 <i>Configurar atalhos</i>	<i>Editar</i>
<i>Gestor de Estilo</i>	<i>Editar</i>
 <i>SRC Personalizado</i>	<i>Editar</i>
<i>Painéis →</i>	<i>Ver</i>
<i>Barra de Ferramentas →</i>	<i>Ver</i>
<i>Activar Modo de Ecrã Completo</i>	<i>Ver</i>
<i>Escala da quadrícula</i>	<i>Ver</i>
<i>Informação do GPS</i>	<i>Ver</i>

7.2 Barra de Ferramentas




A barra de ferramentas fornece o acesso à maioria das mesmas funções que dos menus, mais as ferramentas adicionais para interagir com o mapa. Cada item da barra de ferramentas tem uma janela de ajuda disponível. Mantenha o seu rato em cima do item e uma descrição curta da finalidade da ferramenta irá ser exibida.

Todas as barras de menu podem ser movidas de acordo com as suas necessidades. Adicionalmente todas as barras de menus podem ser desligadas com o botão direito do rato no menu de contexto segurando o rato sobre as barras de ferramentas (leia também *Painéis e Barras de Ferramentas*).

Tip: Restaurar as Barras de Ferramentas

Se acidentalmente escondeu todas as suas barras de ferramentas, pode ir buscá-las escolhendo a opção do menu *Configurações* → *Barra de Ferramentas* →. Se a barra de ferramentas desaparecer no Windows, que costuma ser um problema de tempo a tempo no QGIS, necessita de remover `\HKEY_CURRENT_USER\Software\QGIS\qgis\UI\state` no registo. Quando restaurar o QGIS, a chave é escrita novamente com o estado padrão, e todas as barras de ferramentas estarão visíveis outra vez.

7.3 Legenda do Mapa


The map legend area lists all the layers in the project. The checkbox in each legend entry can be used to show or hide the layer. The Legend toolbar in the map legend area list allow you to **Add group**, **Manage Layer Visibility** of all layers or manage preset layers combination, **Filter Legend by Map Content**, **Expand All** or **Collapse All** and **Remove Layer or Group**. The button  allows you to add **Presets** views in the legend. It means that you can choose to display some layer with specific categorization and add this view to the **Presets** list. To add a preset view just click on , choose *Add Preset...* from the drop down menu and give a name to the preset. After that you will see a list with all the presets that you can recall pressing on the  button.

All the added presets are also present in the map composer in order to allow you to create a map layout based on your specific views (see *Propriedades principais*).

A camada pode ser seleccionada e arrastada para cima e para abaixo na legenda para alterar a ordenação-z. A Ordenação-Z significa que as camadas listadas perto do topo da legenda são desenhadas sobre as camadas listadas mais abaixo na legenda.


Note: Este comportamentos pode ser reescrito pelo painel 'Ordem da Camada'.

As camadas da legenda da janela podem ser organizadas por grupos. Existem duas maneiras de o fazer:

1. Press the  icon to add a new group. Type in a name for the group and press `Enter`. Now click on an existing layer and drag it onto the group.
2. Selecciona algumas camadas, e clique com o botão direito do rato na legenda da janela e escolha *Agrupar Seleccionados*. As camadas seleccionadas irão automaticamente incorporar o novo grupo.

Para trazer a camada para fora do grupo pode arrastar para fora, ou clicar no botão direito do rato em cima e escolha *Faça item de topo*. Os grupos podem ser também agrupados dentro de outros grupos.

A caixa de verificação para o grupo irá mostrar ou esconder todas as camadas do grupo com um clique.

O conteúdo do menu contexto do botão do direito do rato varia no tipo de item de legenda seleccionada seja uma camada matricial ou vector. Para camadas vectoriais GRASS  Alternar Edição não está disponível. Veja a secção *Digitalizando e editando as camadas vectoriais GRASS* para informação sobre como editar camadas vectoriais GRASS.

Opções do botão direito do rato para camadas matriciais

- *Aproximar à extensão da camada*
- *Adicionar ao enquadramento*
- *Ampliar à Melhor Escala (100%)*
- *Esticar Usando o Enquadramento Actual*
- *Remover*

- *Duplicar*
- *Set Layer Scale Visibility*
- *Definir SRC da Camada*
- *Definir SRC do projecto a partir da Camada*
- *Guardar como ...*
- *Save As Layer Definition Style*
- *Propriedades*
- *Renomear*
- *Copiar Estilo*

Adicionalmente, de acordo com a posição da camada e selecção

- *Faça Item de Topo*
- *Agrupar Seleccionados*

Right mouse button menu for vector layers

- *Aproximar à Extensão da Camada*
- *Adicionar ao Enquadramento*
- *Remover*
- *Duplicar*
- *Set Layer Scale Visibility*
- *Definir SRC da Camada*
- *Definir SRC do projecto a partir da Camada*
- *Abrir Tabela de Atributos*
- *Alternar Edição (não está disponível para camadas GRASS)*
- *Guardar Como ...*
- *Save As Layer Definition Style*
- *Filtrar*
- *Exibir Contagem de Elementos*
- *Propriedades*
- *Renomear*
- *Copiar Estilo*

Adicionalmente, de acordo com a posição da camada e selecção

- *Faça Item de Topo*
- *Agrupar Seleccionados*

Opções do botão direito do rato para grupos de camadas

- *Ampliação ao Grupo*
- *Remover*
- *Definir SRC do Grupo*
- *Renomear*
- *Add Group*

É possível seleccionar mais de uma camada ou grupo ao mesmo tempo segurando a tecla `Ctrl` enquanto selecciona as camadas com o botão esquerdo do rato. Pode mover todas as camadas seleccionadas para um novo grupo ao mesmo tempo.

Poderá também apagar mais que uma camada ou grupo de uma só vez através da selecção de várias camadas com a tecla `Ctrl` e pressionando `Ctrl+D` depois. Desta forma todos as camadas ou grupos serão removidos da lista de camadas.

7.3.1 Trabalhando com a Ordem da legenda de camada independente

Existe um painel que permite definir a ordem independente de desenho da legenda do mapa. Pode activar neste menu *Settings* → *Painéis* → *Ordem da camada*. Esta característica permite por exemplo, ordenar as camadas por ordem de importância, para exibi-los na ordem correcta (veja [figure_layer_order](#)). Marcando a caixa *Controlar ordem de renderização* num fundo da lista de camadas irá retroceder para o opção padrão.

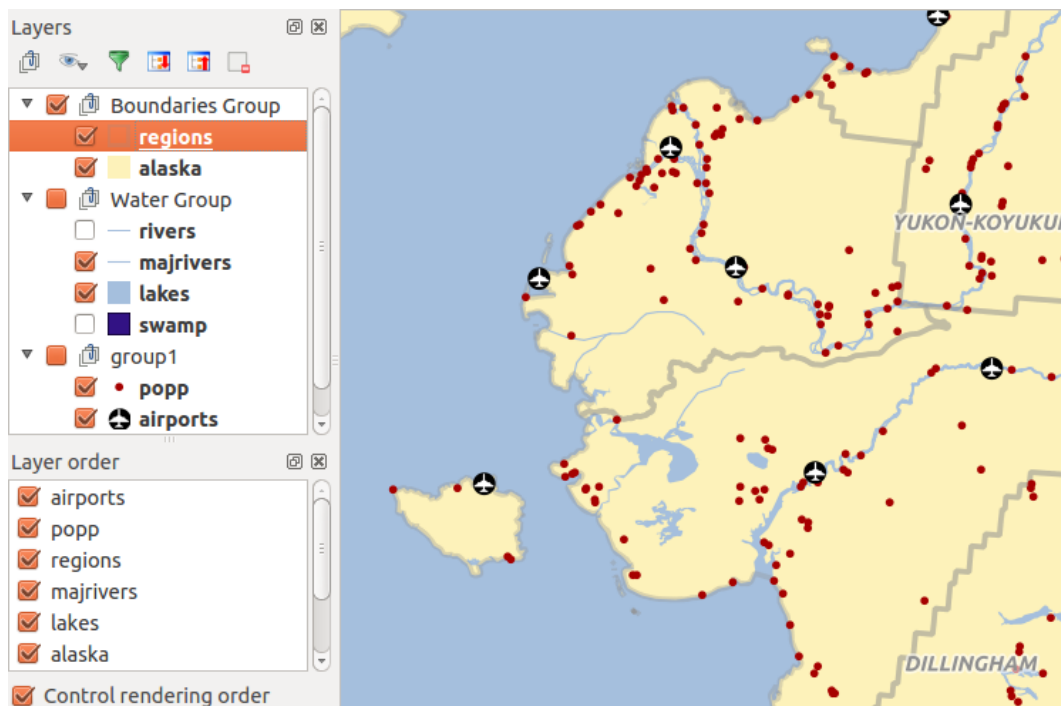


Figura 7.2: Definir a ordem da legenda de camada independente 

7.4 Vista do Mapa

Este é o “rendimento” do QGIS – os mapas são exibidos nesta área! O mapa exibido nesta janela irá depender na camada vectorial e raster que escolheu no carregamento (veja as secções a seguir para mais informação de como carregar camadas). A vista do mapa pode ser movida (mudando o foco da tela do mapa para outra região) e ampliada ou afastada. Outras variadas operações podem ser efectuadas no mapa com é descrito na descrição da barra de ferramenta acima. A vista de mapa e a legenda estão estritamente ligadas uma a outra — os mapas na vista reflectem alterações que faça na área da legenda.

Tip: Aumentado o Mapa com a roda do Rato

Pode usar a roda do rato para ampliar ou afastar o mapa. Posicione o cursores do rato dentro da área do mapa e rode a roda para a frente (longe de si) para ampliar e para trás (perto de si) para afastar. A posição do cursor do mapa é centrada onde a ampliação ocorre. Pode personalizar o comportamento de ampliação da roda do rato usando o separador *Ferramentas de mapa* no menu *Configurações* → *Opções*.

Tip: Movendo o Mapa com as Setas de Direcção e Barra de Espaço


Pode usar as teclas de direcção para mover o mapa. Coloque o cursor do mapa dentro da área do mapa e clique na tecla direita de direcção para mover para este, tecla esquerda de direcção para mover para oeste, tecla para cima para mover para norte e a tecla de direcção para baixo para mover para sul. Pode também mover o mapa usando a barra de espaço ou clicando na roda do rato: apenas mova o rato enquanto segura a barra de espaço ou clica a roda do rato.

7.5 Barra de Estado

A barra de estado mostra a posição actual das coordenadas do mapa (ex.: metros ou graus decimais) à medida que o apontador do rato é movido ao longo da vista do mapa. No lado esquerdo da exibição de coordenadas na barra de estado está um botão pequeno que irá alternar entre a exibição de coordenadas ou o enquadramento da vista de mapa à medida que move, amplia e afasta.


Perto da exibição das coordenadas encontra a exibição da escala. Mostra a escala da vista de mapa. Se ampliar ou afastar, o QGIS mostra a escala actual. Existe um seleccionador de escala, que permite que possa escolher entre escalas pré-definidas de 1:500 até 1:1000000.

Uma barra de progresso na barra de estado mostra o progresso da renderização para cada camada que é desenhada na vista de mapa. Em alguns casos, tais como a recolha de dados estatísticos em camadas matriciais, a barra de progresso será usada para mostrar o estado das operações demoradas.

Se um novo módulo ou actualização de módulo estiver disponível, poderá ver uma mensagem no lado direito mais afastado da barra de estado. No lado direito da barra de estado está uma pequena caixa de verificação que pode ser usada para evitar temporariamente a renderização da vista de mapa. (veja secção *Renderização* em baixo). O ícone  pára imediatamente o processo de renderização do mapa actual.

No lado direito das funções de renderização encontra o código EPSG do SRC actual do projecto e um ícone da projecção. Clicando este, abrirá as propriedades da projecção do projecto actual.

Tip: Calculando a Escala Correcta para o Seu Enquadramento do Mapa

Quando inicia o QGIS, os graus são a unidade padrão, o que significa que QGIS irá interpretar as coordenadas na sua camada em graus. Para obter os valores correctos da escala, pode mudar para metros manualmente no separador *Geral* no menu *Configurações* → *Propriedades do Projecto*, ou pode seleccionar o SRC clicando no ícone  Estado do SRC no canto inferior direito da barra de estado. No último caso, as unidades são definidas para o que a projecção do projecto especifica (ex.: '+units=m').

Ferramentas gerais

8.1 Atalhos do teclado

QGIS provides default keyboard shortcuts for many features. You can find them in section *Barra de Menus*. Additionally, the menu option *Settings* → *Configure Shortcuts*.. allows you to change the default keyboard shortcuts and to add new keyboard shortcuts to QGIS features.

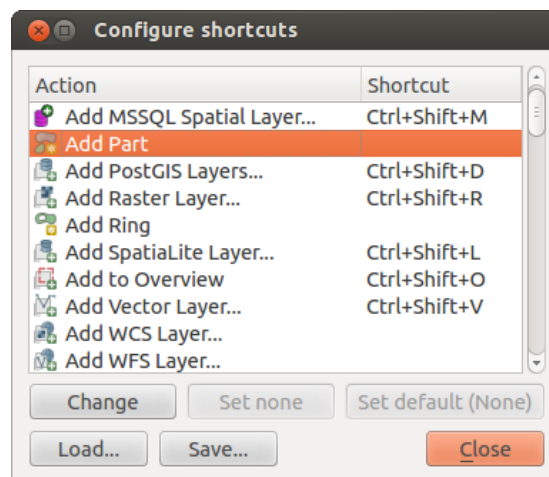


Figura 8.1: Define shortcut options 🐧 (Gnome)

Configuration is very simple. Just select a feature from the list and click on **[Change]**, **[Set none]** or **[Set default]**. Once you have finished your configuration, you can save it as an XML file and load it to another QGIS installation.

8.2 Ajuda de contexto

When you need help on a specific topic, you can access context help via the **[Help]** button available in most dialogs — please note that third-party plugins can point to dedicated web pages.

8.3 Renderização

By default, QGIS renders all visible layers whenever the map canvas is refreshed. The events that trigger a refresh of the map canvas include:

- Adicionando uma camada
- Movendo e ampliando

- Resizing the QGIS window
- Mudar a visibilidade da camada ou camadas

O QGIS permite que possa controlar o processo de renderização de várias formas.

8.3.1 Renderização Dependente da Escala

Scale-dependent rendering allows you to specify the minimum and maximum scales at which a layer will be visible. To set scale-dependent rendering, open the *Properties* dialog by double-clicking on the layer in the legend. On the *General* tab, click on the *Scale dependent visibility* checkbox to activate the feature, then set the minimum and maximum scale values.

Pode determinar os valores da escala em primeiro lugar aumentando a um nível que quer usar e não ter valores de escala na barra de estado do QGIS.

8.3.2 Controlando a Renderização do Mapa

A renderização do mapa pode ser controlada de várias maneiras, como é descrito em baixo.

Suspendendo a Renderização

To suspend rendering, click the *Render* checkbox in the lower right corner of the status bar. When the *Render* checkbox is not checked, QGIS does not redraw the canvas in response to any of the events described in section *Renderização*. Examples of when you might want to suspend rendering include:

- Adding many layers and symbolizing them prior to drawing
- Adding one or more large layers and setting scale dependency before drawing
- Adding one or more large layers and zooming to a specific view before drawing
- Qualquer combinação acima

A activação da caixa de verificação *Renderização* permite a renderização e actualização imediata do enquadramento do mapa.

Setting Layer Add Option

You can set an option to always load new layers without drawing them. This means the layer will be added to the map, but its visibility checkbox in the legend will be unchecked by default. To set this option, choose menu option *Settings* → *Options* and click on the *Rendering* tab. Uncheck the *By default new layers added to the map should be displayed* checkbox. Any layer subsequently added to the map will be off (invisible) by default.

Terminando Renderização

To stop the map drawing, press the ESC key. This will halt the refresh of the map canvas and leave the map partially drawn. It may take a bit of time between pressing ESC and the time the map drawing is halted.

Note: It is currently not possible to stop rendering — this was disabled in the Qt4 port because of User Interface (UI) problems and crashes.

Atualizando a Visualização do Mapa Durante a Renderização

You can set an option to update the map display as features are drawn. By default, QGIS does not display any features for a layer until the entire layer has been rendered. To update the display as features are read from the datastore, choose menu option *Settings* → *Options* and click on the *Rendering* tab. Set the feature count to an appropriate value to update the display during rendering. Setting a value of 0 disables update during drawing (this is the default). Setting a value too low will result in poor performance, as the map canvas is continually updated during the reading of the features. A suggested value to start with is 500.

Influência da Qualidade de Renderização

To influence the rendering quality of the map, you have two options. Choose menu option *Settings* → *Options*, click on the *Rendering* tab and select or deselect following checkboxes:

- *Make lines appear less jagged at the expense of some drawing performance*
- *Fix problems with incorrectly filled polygons*


Acelerar a renderização

There are two settings that allow you to improve rendering speed. Open the QGIS options dialog using *Settings* → *Options*, go to the *Rendering* tab and select or deselect the following checkboxes:


- *Enable back buffer*. This provides better graphics performance at the cost of losing the possibility to cancel rendering and incrementally draw features. If it is unchecked, you can set the *Number of features to draw before updating the display*, otherwise this option is inactive.
- *Use render caching where possible to speed up redraws*


8.4 Medindo

Measuring works within projected coordinate systems (e.g., UTM) and unprojected data. If the loaded map is defined with a geographic coordinate system (latitude/longitude), the results from line or area measurements will be incorrect. To fix this, you need to set an appropriate map coordinate system (see section *Trabalhando com Projeções*). All measuring modules also use the snapping settings from the digitizing module. This is useful, if you want to measure along lines or areas in vector layers.

Para seleccionar a ferramenta de medição, clique em  e seleccione a ferramenta que quer usar.

8.4.1 Medir comprimentos, áreas e ângulos

 **Measure Line**: QGIS is able to measure real distances between given points according to a defined ellipsoid. To configure this, choose menu option *Settings* → *Options*, click on the *Map tools* tab and select the appropriate ellipsoid. There, you can also define a rubberband color and your preferred measurement units (meters or feet) and angle units (degrees, radians and gon). The tool then allows you to click points on the map. Each segment length, as well as the total, shows up in the measure window. To stop measuring, click your right mouse button.

 **Measure Area**: Areas can also be measured. In the measure window, the accumulated area size appears. In addition, the measuring tool will snap to the currently selected layer, provided that layer has its snapping tolerance set (see section *Configurando a Tolerância de Atracção e Raio de Pesquisa*). So, if you want to measure exactly along a line feature, or around a polygon feature, first set its snapping tolerance, then select the layer. Now, when using the measuring tools, each mouse click (within the tolerance setting) will snap to that layer.

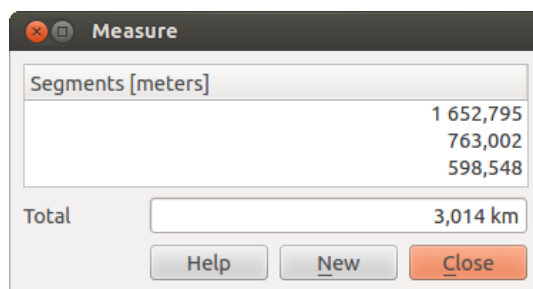


Figura 8.2: Measure Distance 🐧 (Gnome)

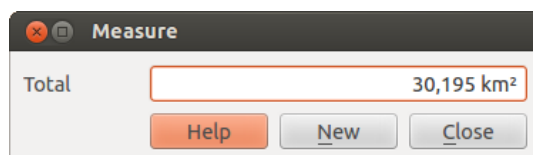



Figura 8.3: Measure Area 🐧 (Gnome)

 **Measure Angle:** You can also measure angles. The cursor becomes cross-shaped. Click to draw the first segment of the angle you wish to measure, then move the cursor to draw the desired angle. The measure is displayed in a pop-up dialog.

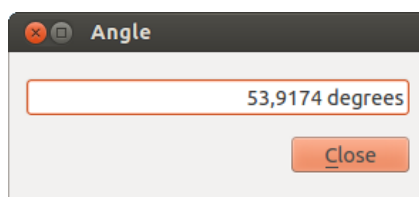









Figura 8.4: Measure Angle 🐧 (Gnome)

8.4.2 Selecionar e desselecionar elementos

A caixa de ferramentas QGIS fornece várias ferramentas para seleccionar elementos no enquadramento do mapa. Para seleccionar um ou mais elementos, simplesmente clique em  e seleccione a sua ferramenta:


-  Seleccionar Elemento Único
-  Seleccionar Elementos por Rectângulo
-  Seleccionar Elementos por Polígono
-  Seleccionar Elementos por Delimitação Livre
-  Seleccionar Elementos pelo Raio

To deselect all selected features click on  Deselect features from all layers.

 **Select feature using an expression** allow user to select feature using expression dialog. See *Expressions* chapter for some example.

Users can save features selection into a **New Memory Vector Layer** or a **New Vector Layer** using *Edit* → *Paste Feature as ...* and choose the mode you want.

8.5 Identificar elementos

The Identify tool allows you to interact with the map canvas and get information on features in a pop-up window. To identify features, use *View* → *Identify features* or press **Ctrl + Shift + I**, or click on the  Identify features icon in the toolbar.

If you click on several features, the *Identify results* dialog will list information about all the selected features. The first item is the number of the feature in the list of results, followed by the layer name. Then, its first child will be the name of a field with its value. Finally, all information about the feature is displayed.

Esta janela pode ser personalizada para exibir campos personalizados, mas por defeito irá exibir três tipos de informação:

- **Acções:** Acções podem ser adicionadas à janela de indentificação de elementos. Quando clica no rótulo de acção, uma acção correrá. Por defeito, apenas uma acção é adicionada, que é ver o formulário do elemento em modo edição.
- **Derived:** This information is calculated or derived from other information. You can find clicked coordinate, X and Y coordinates, area in map units and perimeter in map units for polygons, length in map units for lines and feature ids.
- **Data attributes:** This is the list of attribute fields from the data.

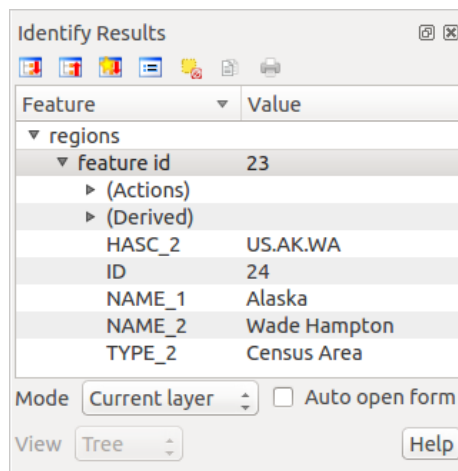








Figura 8.5: Diálogo de Identificar elementos  (Gnome)

No fundo da janela, tem cinco ícones:

-  Expand tree
-  Collapse tree
-  Default behaviour
-  Copiar atributos
-  Print selected HTML response

Outras funções podem ser encontradas no menu de contexto do item identificado. Por exemplo, a partir do menu de contexto poderá:


- Ver o formulário do elemento
- Aproximar ao elemento
- Copiar elemento: Copiar todas as geometrias do elemento e atributos
- Toggle feature selection: adds identified feature to selection

- Copiar o valor do atributo: Copia apenas o valor do atributo que clicou
- Copy feature attributes: Copy only attributes
- Limpar resultado: remove os resultados na janela
- Clear highlights: Remove features highlighted on the map
- Destacar todos
- Destacar camada
- Activate layer: Choose a layer to be activated
- Propriedades da camada: Abre a janela das propriedades da camada
- Expandir todos
- Fechar todos

8.6 Decorações

The Decorations of QGIS include the Grid, the Copyright Label, the North Arrow and the Scale Bar. They are used to 'decorate' the map by adding cartographic elements.

8.6.1 Grelha

 Grid allows you to add a coordinate grid and coordinate annotations to the map canvas.

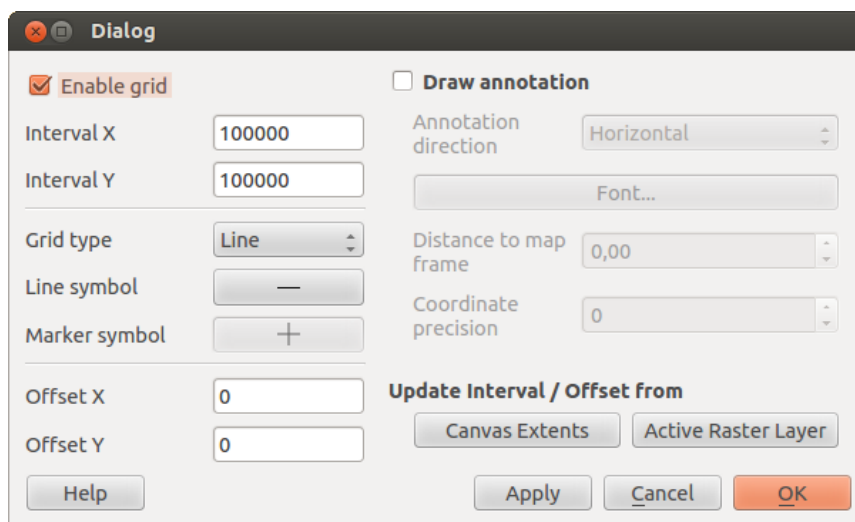



Figura 8.6: Janela da Grelha 

1. Seleccione a partir do menu *Ver* → *Decorações* → *Grelha*. A janela de diálogo inicia. (veja [figure_decorations_1](#)).
2. Active the *Enable grid* checkbox and set grid definitions according to the layers loaded in the map canvas.
3. Active the *Draw annotations* checkbox and set annotation definitions according to the layers loaded in the map canvas.
4. Clique [**A**pl~~i~~c~~a~~r] para verificar se tudo está como o esperado.
5. Clique [**O**K] para fechar a janela de diálogo.

8.6.2 Etiqueta de Direitos de autor

 Copyright label adds a copyright label using the text you prefer to the map.

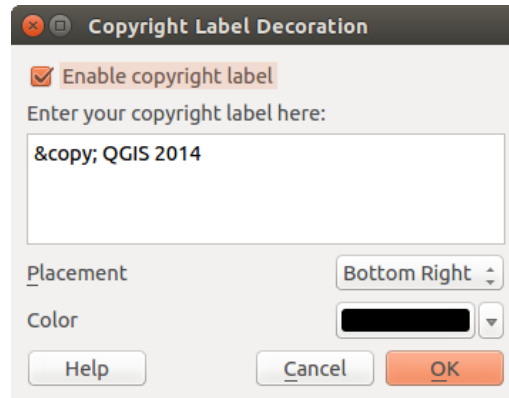




Figura 8.7: The Copyright Dialog 

1. Selecione a partir do menu *Ver* → *Decorações* → *Etiqueta Copyright*. O diálogo iniciará (veja [figure_decorations_2](#)).
2. Enter the text you want to place on the map. You can use HTML as shown in the example.
3. Choose the placement of the label from the *Placement*  combo box.
4. Make sure the *Enable Copyright Label* checkbox is checked.
5. Clique [OK].

In the example above, which is the default, QGIS places a copyright symbol followed by the date in the lower right-hand corner of the map canvas.

8.6.3 Seta do Norte

 North Arrow places a simple north arrow on the map canvas. At present, there is only one style available. You can adjust the angle of the arrow or let QGIS set the direction automatically. If you choose to let QGIS determine the direction, it makes its best guess as to how the arrow should be oriented. For placement of the arrow, you have four options, corresponding to the four corners of the map canvas.

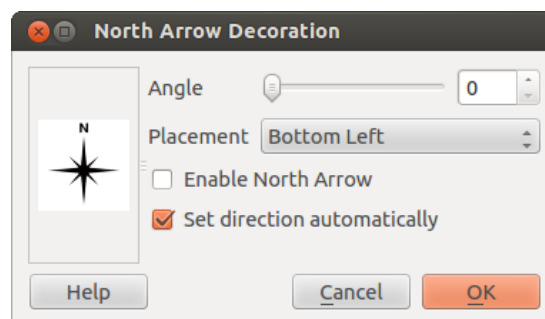




Figura 8.8: Diálogo da Seta do Norte 

8.6.4 Barra de Escala

 Scale Bar adds a simple scale bar to the map canvas. You can control the style and placement, as well as the labeling of the bar.

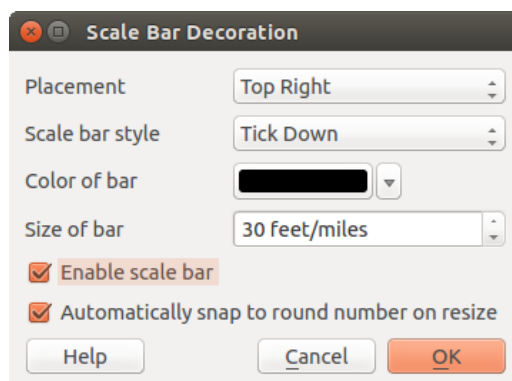







Figura 8.9: Diálogo da Barra de Escala 

QGIS only supports displaying the scale in the same units as your map frame. So if the units of your layers are in meters, you can't create a scale bar in feet. Likewise, if you are using decimal degrees, you can't create a scale bar to display distance in meters.


Para adicionar a barra de escala:

1. Select from menu *View* → *Decorations* → *Scale Bar*. The dialog starts (see [figure_decorations_4](#)).
2. Choose the placement from the *Placement*  combo box.
3. Choose the style from the *Scale bar style*  combo box.
4. Select the color for the bar *Color of bar*  or use the default black color.
5. Set the size of the bar and its label *Size of bar* .
6. Make sure the *Enable scale bar* checkbox is checked.
7. Optionally, check *Automatically snap to round number on resize*.
8. Clique [OK].

Tip: Separador das Decorações

When you save a .qgs project, any changes you have made to Grid, North Arrow, Scale Bar and Copyright will be saved in the project and restored the next time you load the project.

8.7 Ferramentas de Anotação

The  Text Annotation tool in the attribute toolbar provides the possibility to place formatted text in a balloon on the QGIS map canvas. Use the *Text Annotation* tool and click into the map canvas.

Double clicking on the item opens a dialog with various options. There is the text editor to enter the formatted text and other item settings. For instance, there is the choice of having the item placed on a map position (displayed by a marker symbol) or to have the item on a screen position (not related to the map). The item can be moved by map position (by dragging the map marker) or by moving only the balloon. The icons are part of the GIS theme, and they are used by default in the other themes, too.

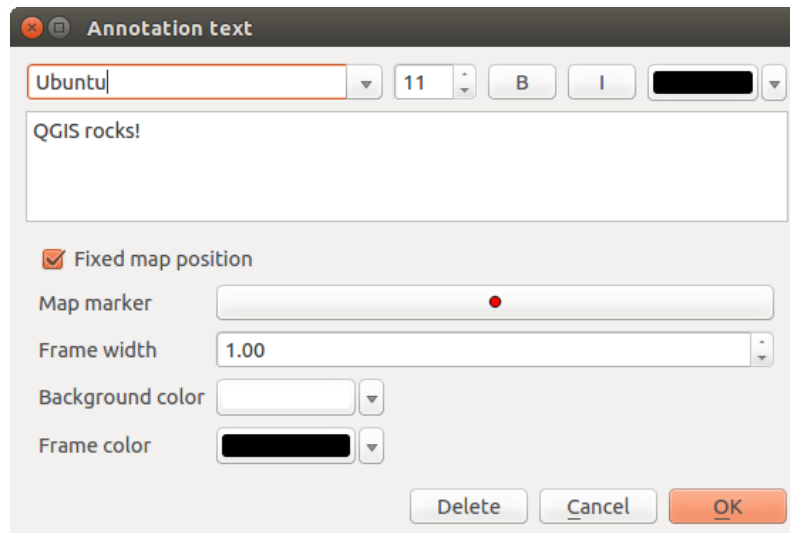






Figura 8.10: Diálogo do texto de anotação 

The  **Move Annotation** tool allows you to move the annotation on the map canvas.


8.7.1 Anotações Html

The  **Html Annotation** tools in the attribute toolbar provides the possibility to place the content of an html file in a balloon on the QGIS map canvas. Using the *Html Annotation* tool, click into the map canvas and add the path to the html file into the dialog.

8.7.2 Anotações SVG

The  **SVG Annotation** tool in the attribute toolbar provides the possibility to place an SVG symbol in a balloon on the QGIS map canvas. Using the *SVG Annotation* tool, click into the map canvas and add the path to the SVG file into the dialog.

8.7.3 Formulários de anotações

Additionally, you can also create your own annotation forms. The  **Form Annotation** tool is useful to display attributes of a vector layer in a customized Qt Designer form (see [figure_custom_annotation](#)). This is similar to the designer forms for the *Identify features* tool, but displayed in an annotation item. Also see this video <https://www.youtube.com/watch?v=0pDBuSbQ02o> from Tim Sutton for more information.

Note: Se clicar `Ctrl+T` enquanto a ferramenta *Anotação* é activada (move a anotação, anotação de texto, formulário da anotação), o estado de visibilidade desses itens são invertidos.

8.8 Marcadores espaciais

Marcadores espaciais permite-lhe “marcar” uma localização geográfica e voltar a ele mais tarde.

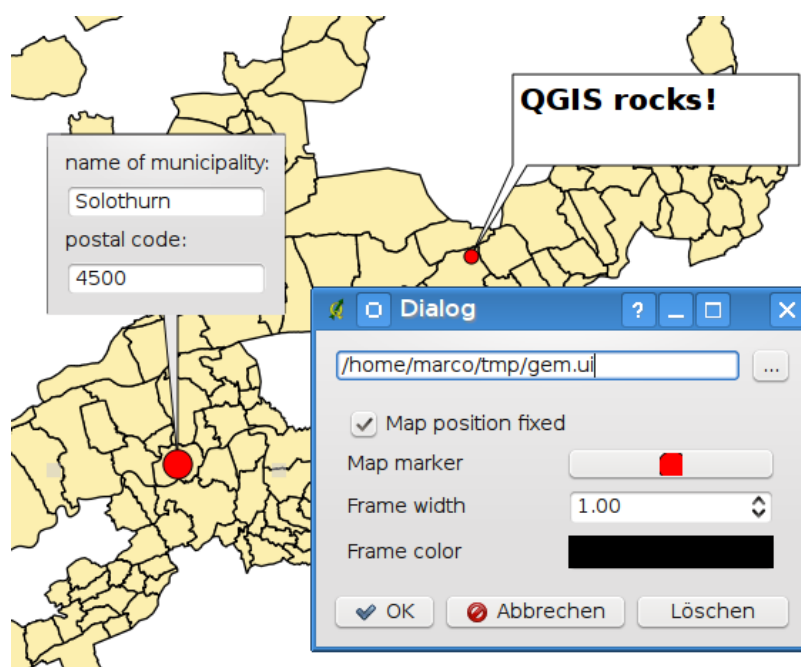


Figura 8.11: Formulário de anotação personalizado no Qt Designer 🐧

8.8.1 Criando um Marcador

Para criar um marcador:

1. Aproximar ou mover para uma área de interesse.
2. Seleccione a opção do menu *Ver* → *Novo Marcador* ou carregue **Ctrl-B**.
3. Introduza um nome descritivo para o marcador (até 255 caracteres).
4. Pressione **Enter** para adicionar o marcador ou **[Apagar]** para remover o marcador.

Repare que tem múltiplos marcadores com o mesmo nome.

8.8.2 Trabalhando com Marcadores

To use or manage bookmarks, select the menu option *View* → *Show Bookmarks*. The *Geospatial Bookmarks* dialog allows you to zoom to or delete a bookmark. You cannot edit the bookmark name or coordinates.

8.8.3 Aproximando a um Marcador

From the *Geospatial Bookmarks* dialog, select the desired bookmark by clicking on it, then click **[Zoom To]**. You can also zoom to a bookmark by double-clicking on it.

8.8.4 Apagando um Marcador


To delete a bookmark from the *Geospatial Bookmarks* dialog, click on it, then click **[Delete]**. Confirm your choice by clicking **[Yes]**, or cancel the delete by clicking **[No]**.

8.9 Nesting Projects

If you want to embed content from other project files into your project, you can choose *Layer → Embed Layers and Groups*.

8.9.1 Embebendo camadas

A seguinte janela permite que possa embeber camadas de outros projectos. Aqui está um pequeno exemplo:

1. Pressione  para ver outro projeto a partir do conjunto de dados Alaska.
2. Select the project file `grassland`. You can see the content of the project (see [figure_embed_dialog](#)).
3. Press `Ctrl` and click on the layers `grassland` and `regions`. Press **[OK]**. The selected layers are embedded in the map legend and the map view now.

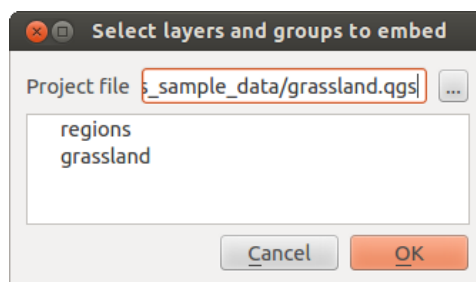



Figura 8.12: Selecione as camadas e grupos para incorporar 

While the embedded layers are editable, you can't change their properties like style and labeling.

8.9.2 Removendo camadas embebidas

Right-click on the embedded layer and choose  Remove.

Configuração QGIS

O QGIS é altamente personalizável através do menu *Configurações*. Escolha entre Painéis, Caixa de Ferramentas, Propriedades do Projecto, Opções e Personalização.

Note: QGIS follows desktop guidelines for the location of options and project properties item. Consequently related to the OS you are using, location of some of items described above could be located in the :menuselection'view' menu (Panels and Toolbars) or in *Project* for Options.

9.1 Painéis e Barras de Ferramentas

No menu *Painéis*→ pode desligar os widgets do QGIS. O menu *Caixa de Ferramentas*→ fornece a possibilidade de trocar activar ou desactivar grupos de ícones na barra de ferramentas do QGIS (veja [figure_panels_toolbars](#)).

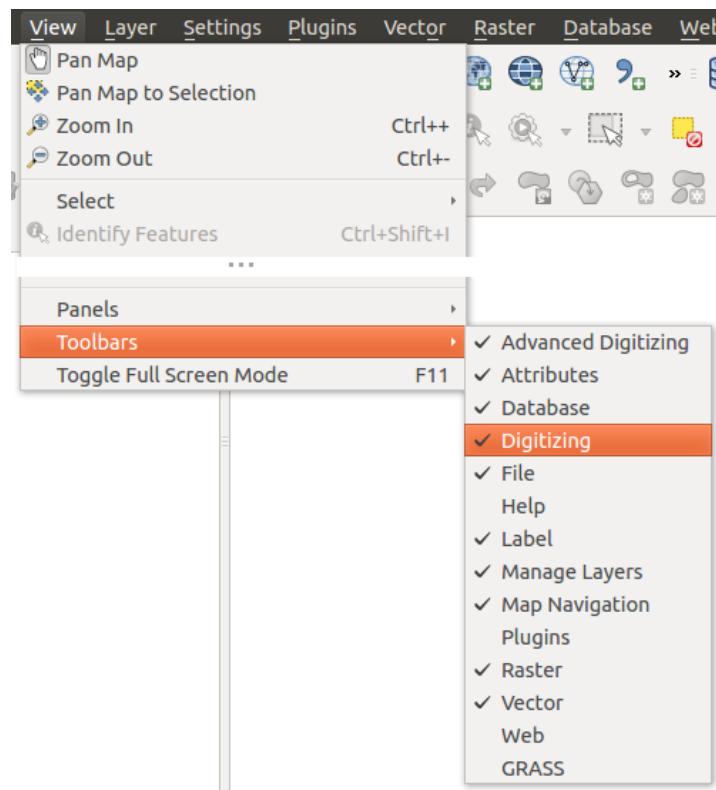







Figura 9.1: Os Painéis e o Menu de Barra de Ferramentas 




Tip: Activando o Enquadramento QGIS

No QGIS pode usar o painel do enquadramento que fornece a extensão total das camadas adicionadas. Pode ser seleccionada no menu  *Configurações* → *Painéis* ou  *Ver* → *Painéis*. Dentro da vista existe um rectângulo a mostrar a extensão actual do mapa. Isto permite rapidamente determinar que área do mapa está a ver actualmente. É de notar que os rótulos não serão renderizados no enquadramento do mapa mesmo que seja activada a rotulagem. Se clicar e arrastar o rectângulo vermelho no enquadramento que mostra a actual extensão, este irá actualizar de acordo como o mapa principal.

Tip: Mostrar Mensagens de Registo



É possível seguir as mensagens do QGIS. Pode activar o  *Registo de Mensagens* no menu  *Configurações* → *Painéis* ou  *Vier* → *Painéis* e seguir as mensagens que aparecem nos diferentes separadores durante o carregamento e operação.

9.2 Propriedades do Projecto

In the properties window for the project under  *Settings* → *Project Properties* (kde) or   *Project* → *Project Properties* (Gnome), you can set project-specific options. These include:

- No menu *Geral* podem ser definidos o título do projecto, a cor de selecção e fundo, unidades da camadas, precisão, e os caminhos relativos onde serão guardadas as camadas. Se a transformação SRC estiver ligada pode escolher o cálculo de distâncias recorrendo ao elipsóide. Pode definir as unidades do enquadramento (apenas usado quando a transformação SRC está desactivada) e a precisão das casas decimais a usar. Pode definir também uma escala de projecto, que rescreverá sobre as escalas globais pré-definidas.
- O menu *SRC* permite que escolha o Sistema de Coordenadas Referência para o projecto, e para activar a projecção on-the-fly das camadas matricias e vectoriais na exibição de camadas de diferentes SRC.
- Com o terceiro menu *Identificar camadas* pode definir (ou desactivar) que camadas irão responder à ferramenta identificar. (Veja o parágrafo das “Ferramentas de Mapa” da secção *Opções* Secção para activar a identificação de múltiplas camadas).
- The *Default Styles* menu lets you control how new layers will be drawn when they do not have an existing `.qml` style defined. You can also set the default transparency level for new layers and whether symbols should have random colours assigned to them. There is also an additional section where you can define specific colors for the running project. You can find the added colors in the drop down menu of the color dialog window present in each renderer.
- O separador *Servidor OWS* permite definir a informação sobre as Capacidades do WMS e WFS, a extensão e as restrições SRC do Servidor QGIS.
- O menu *Macros* é para editar macros Python para os projectos. Actualmente, apenas estão disponíveis três macros: `openProject()`, `saveProject()` e `closeProject()`.
- O menu *Relations* é usado para definir relações 1:n. As relações são definidas no diálogo das propriedades do projecto. Quando existirem relações para uma camada, um novo elemento de interface do utilizador na vista de formulário (p. ex. quando identificar um elemento e abrir o seu formulário) irá listar os elementos relacionados. Isto fornece uma maneira poderosa para expressar p. ex. o histórico de inspecção ao longo de um segmento de tubagem ou estrada. Poderá encontrar mais informação sobre suporte de relações 1:n na Secção *Creating one to many relations*.

9.3 Opções

 Algumas opções básicas para QGIS podem ser seleccionadas usando o diálogo *Options*. Selecione a opção de menu *Settings* →  *Options*. Os separadores onde poderá personalizar as suas opções são descritos abaixo.

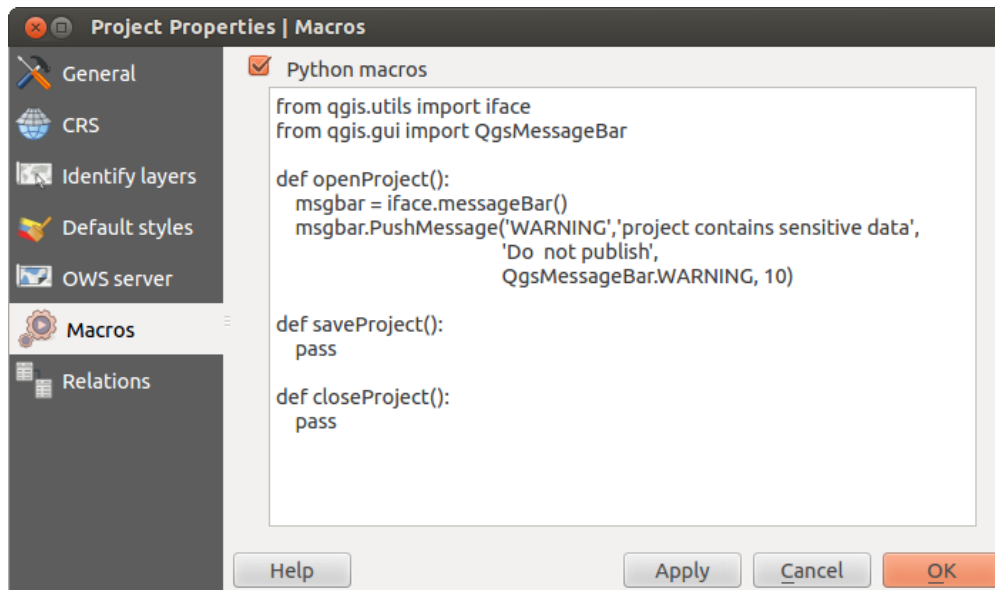









Figura 9.2: Configurações de Macro no QGIS

9.3.1 Menu Geral

Aplicação

- Seleccione a *Estilo* (Necessário reiniciar o QGIS)  e escolha entre 'Oxygen', 'Windows', 'Motif', 'CDE', 'Plastique' e 'Cleanlux' .
- Definir o *Tema de Ícone* . Actualmente só a opção 'default' é possível.
- Definir o *Tamanho do Ícone* .
- Definir a *Fonte*. Escolha entre *QT padrão* e uma fonte definida pelo utilizador.
- Altera o *Tempo limite para mensagens temporizadas ou diálogos* .
- *Esconder o ecrã inicial no arranque*
- *Mostrar dicas ao iniciar*
- *Títulos da caixa de grupos a negrito*
- :guilabel:' Caixas de grupo em estilo QGIS'
- *Use diálogos selectores de cor com actualizações ao vivo*

Ficheiros de projecto

- *Abrir o projecto no arranque*  (escolha entre 'Novo', 'Mais recente' e 'Específico'). Quando escolher 'Específico' use  para definir um projecto.
- *Create new project from default project*. Tem a possibilidade de carregar em *Set current project as default* ou em *Reset default*. Pode navegar através dos seus ficheiros e definir uma directoria onde encontrar os modelos de projecto definidos pelo utilizador. Isto será adicionado a *Project → New From Template*. Primeiro active *Create new project from default project* e a seguir salve o projecto dentro da pasta de modelos de projecto.
- *Avisar para salvar projecto e alterações de fontes de dados quando necessário*

- Avisar quando abrir um projecto guardado com uma versão antiga do |qg|
- *Enable macros* . Esta opção foi criada para lidar com as macros que são escritas para executar uma acção nos eventos do projecto. Pode escolher entre ‘Never’, ‘Ask’, ‘For this session only’ e ‘Always (not recommended)’.

9.3.2 Menu Sistema

Ambiente

Variáveis de ambiente do sistema podem ser vistas agora, e muitas configuradas, no grupo **Environment** (ver [figure_environment_variables](#)). Isto é útil para plataformas, tais como Mac, onde um aplicativo GUI não herda necessariamente o ambiente do utilizador da linha de comandos. Também é útil para configuração e visualização de variáveis de ambiente para os conjuntos de ferramentas externas controlados pela caixa de ferramentas de Processamento (p. ex., SAGA, GRASS), e para activar a saída de depuração (“debugging”) para secções específicas do código-fonte.

- *Use custom variables (restart required - include separators)*. You can **[Add]** and **[Remove]** variables. Already-defined environment variables are displayed in *Current environment variables*, and it’s possible to filter them by activating *Show only QGIS-specific variables*.

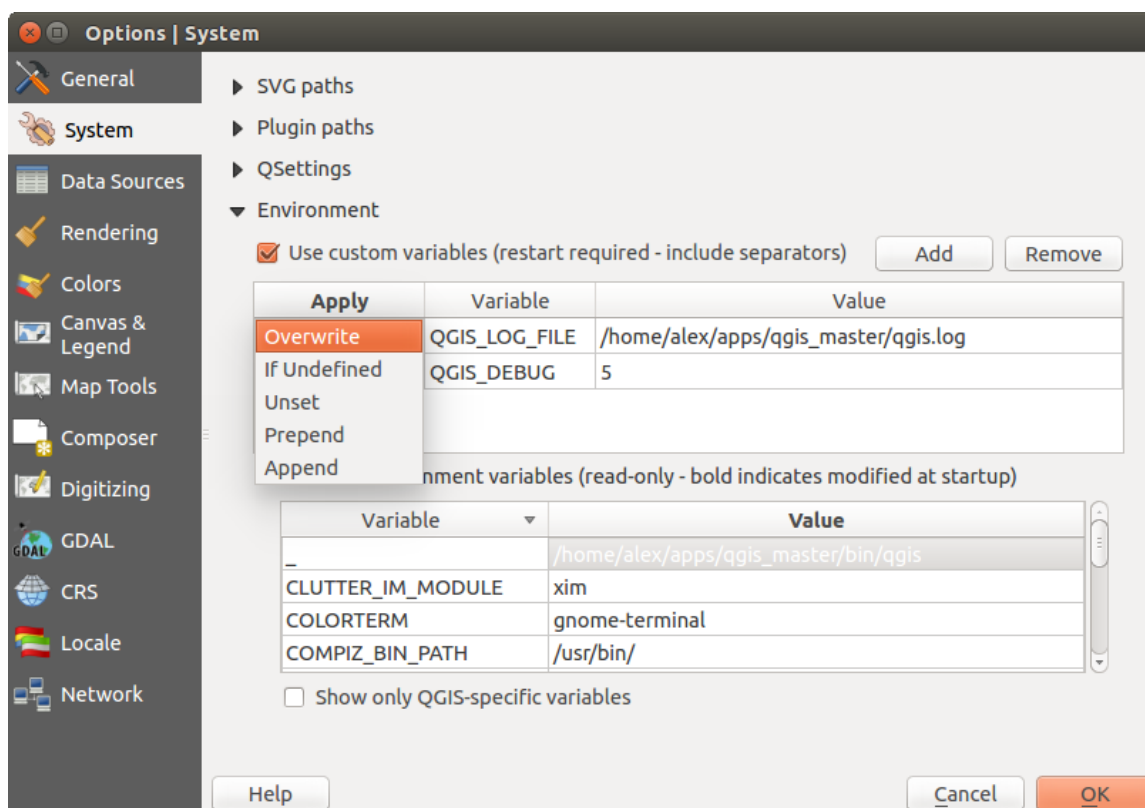





Figura 9.3: Variáveis de ambiente do Sistema no QGIS

Directórios dos módulos



[Adicionar] ou [Remover] Caminho(s) para pesquisar bibliotecas de módulos C++ adicionais

9.3.3 Menu Fonte de Dados

Atributos dos elementos e tabela

- *Abre a tabela de atributos na janela principal (Necessário reiniciar o QGIS)*
- *Copy geometry in WKT representation from attribute table.* When using  Copy selected rows to clipboard from the *Attribute table* dialog, this has the result that the coordinates of points or vertices are also copied to the clipboard.
- *Comportamento da tabela de atributos* . Existem três possibilidades: ‘Mostrar todos os elementos’, ‘Mostrar elementos seleccionados’ e ‘Mostrar elementos visíveis no mapa’.
- *Attribute table row cache* . This row cache makes it possible to save the last loaded N attribute rows so that working with the attribute table will be quicker. The cache will be deleted when closing the attribute table.
- *Representação para valores NULL.* Aqui, pode definir um valor para os campos de dados que contêm valores NULL.

Manipulação das fontes de dados

- *Pesquisar por atributos válidos na janela do navegador* . Pode escolher entre ‘Verificar extensão’ e ‘Verificar conteúdo do ficheiro’.
- *Pesquisar por conteúdo de ficheiros compactados (.zip) na janela do navegador* . ‘Não’, ‘Verificação básica’ e ‘Verificação completa’ são possíveis.
- *Prompt for raster sublayers when opening.* Some rasters support sublayers — they are called subdatasets in GDAL. An example is netCDF files — if there are many netCDF variables, GDAL sees every variable as a subdataset. The option allows you to control how to deal with sublayers when a file with sublayers is opened. You have the following choices:
 - ‘Sempre’: Perguntar sempre (se existem subcamadas)
 - ‘Se necessário’: Perguntar se a camada não tem bandas, mas tem subcamadas
 - ‘Nunca’: Nunca pede, não irá carregar nada
 - ‘Carregar tudo’: Nunca pede, mas carrega todas as subcamadas
- *Ignora a declaração de codificação da shapefile.* Se a shapefile tiver informação de codificação, este será ignorado pelo QGIS.
- *Adicionar uma camada PostGIS com duplo clique e seleccione em modo alargado*
- *Adicione camadas Oracle com duplo clique e seleccione em modo alargado*

9.3.4 Menu de Renderização

Comportamento da renderização

- *By default new layers added to the map should be displayed*
- *Use render caching where possible to speed up redraws*
- *Render layers in parallel using many CPU cores*
- *Max cores to use*
- *Map update interval (default to 250 ms)*
- *Enable feature simplification by default for newly added layers*
- *Simplification threshold*
- *Simplify on provider side if possible*
- *Maximum scale at which the layer should be simplified*





Qualidade de renderização

- *Fazer com que as linhas apareçam menos irregulares, em detrimento de algum desempenho do desenho*

Matriciais

- Com a *Seleção de banda RGB* pode definir o número para a banda Vermelha, Verde e Azul.

Contrast enhancement

- *Banda cinzenta única* . Uma banda cinzenta única pode ter 'Sem esticar', 'Esticar para MinMax', 'Esticar e Cortar para MinMax' e também 'Cortar para MinMax'.
- *Multi band color (byte/band)* . Options are 'No stretch', 'Stretch to MinMax', 'Stretch and Clip to MinMax' and 'Clip to MinMax'.
- *Multi band color (>byte/band)* . Options are 'No stretch', 'Stretch to MinMax', 'Stretch and Clip to MinMax' and 'Clip to MinMax'.
- *Limits (minimum/maximum)* . Options are 'Cumulative pixel count cut', 'Minimum/Maximum', 'Mean +/- standard deviation'.
- *Limites de contagem cumulativa de pixeis de corte*
- *Multiplicador do desvio-padrão*

Corrigindo Erros

- *Map canvas refresh*

9.3.5 Colors Menu


This menu allows you to add some custom color that you can find in each color dialog window of the renderes. You will see a set of predefined colors in the tab: you can delete or edit all of them. Moreover you can add the color you want and perform some copy and paste operation. Finally you can export the color set as a `gpl` file or import them.

9.3.6 Menu de Janela e Legenda

Aparência padrão do mapa (reescrita pelas propriedades do projecto)

- Define a *Cor da selecção* e a *Cor de fundo*.

Legenda da camada



- *Duplo clique na legenda* . Pode 'Abrir propriedades da camada' ou 'Abrir tabela de atributos' com duplo clique.
- The following *Legend item styles* are possible:
 - *Maiusculizar os nomes das camadas*
 - *Tornar negrito os nomes das camadas*
 - *Tornar negrito os nomes dos grupos*
 - *Mostrar nomes de atributos de classificação*
 - *Criar ícones matriciais (pode ser lento)*
 - *Adicionar novas camadas ao grupo actual ou seleccionado*

9.3.7 Menu Ferramentas de Mapa


This menu offers some options regarding the behaviour of the *Identify tool*.

- *Search radius for identifying and displaying map tips* is a tolerance factor expressed as a percentage of the map width. This means the identify tool will depict results as long as you click within this tolerance.
- *Highlight color* allows you to choose with which color should features being identified are to be highlighted.
- *Buffer* expressed as a percentage of the map width, determines a buffer distance to be rendered from the outline of the identify highlight.
- *Minimum width* expressed as a percentage of the map width, determines how thick should the outline of a highlighted object be.

Ferramenta de medida

- Define *Cor do elástico* para as ferramentas de medida
- Define *Casas decimais*
- *Manter unidade base*
- *Preferred measurements units*  ('Meters', 'Feet', 'Nautical Miles' or 'Degrees')
- *Preferred angle units*  ('Degrees', 'Radians' or 'Gon')

Movendo e ampliando

- Define a *Acção da roda do rato*  ('Visualizar', 'Visualizar e Centrar', 'Aproximar ao cursor do rato', 'Nada')
- Define o *factor de aproximação/afastamento* para a roda do rato

Escalas pré-definidas

Here, you find a list of predefined scales. With the [+] and [-] buttons you can add or remove your individual scales.

9.3.8 Composer Menu

Composition defaults

You can define the *Default* font here.

Aparência da Grelha

- Define the *Grid style*  ('Solid', 'Dots', 'Crosses')
- Define the *Color...*

Grid defaults

- Define the *Spacing*
- Define the *Grid offset* for x and y
- Define the *Snap tolerance*

Guide defaults

- Define the *Snap tolerance*

9.3.9 Menu Digitalizar


Criação de elementos

- *Suprimir atributos de janelas pop-up depois de cada elemento criado*
- *Reutilizar últimos valores de atributos inseridos*
- *Validate geometries.* Editing complex lines and polygons with many nodes can result in very slow rendering. This is because the default validation procedures in QGIS can take a lot of time. To speed up rendering, it is possible to select GEOS geometry validation (starting from GEOS 3.3) or to switch it off. GEOS geometry validation is much faster, but the disadvantage is that only the first geometry problem will be reported.


Elástico

- Define a Borracha *Espessura da linha* e *Cor da linha*


Ajuste

- *Abrir opções de atracção na janela principal (necessário reiniciar o QGIS)*
- Define o *Modo de atracção padrão*  ('Ao vértice', 'Ao segmento', 'Ao vértice e segmento', 'Desligado')
- Define *Tolerância de atracção pré-definida* em unidades de mapa ou pixels
- Define o *Raio de pesquisa para editar vértices* em unidades de mapa ou pixels

Marcadores de Vértices

- *Mostrar marcadores apenas para elementos seleccionados*
- Define o vértice do *Estilo do Marcador*  ('Cruz' (padrão), 'Circulo semi-transparente' ou 'Nenhum')
- Definir o vértice *Tamanho do Marcador*

Ferramenta de curva de afastamento

The next 3 options refer to the  Offset Curve tool in *Digitalização Avançada*. Through the various settings, it is possible to influence the shape of the line offset. These options are possible starting from GEOS 3.3.

- *Join style*
- *Quadrant segments*
- *Miter limit*

9.3.10 Menu GDAL

GDAL is a data exchange library for raster files. In this tab, you can *Edit create options* and *Edit Pyramids Options* of the raster formats. Define which GDAL driver is to be used for a raster format, as in some cases more than one GDAL driver is available.

9.3.11 Menu SRC

SRC padrão para novos projectos

- *Don't enable 'on the fly' reprojection*
- *Automatically enable 'on the fly' reprojection if layers have different CRS*
- *Enable 'on the fly' reprojection by default*
- Seleccione um SRC em *Iniciar sempre um novo projecto com este SRC*

SRC para novas camadas

This area allows you to define the action to take when a new layer is created, or when a layer without a CRS is loaded.

- *Pergunta por SRC*
- *Usar SRC do projecto*
- *Utilizar SRC padrão mostrado em baixo*

Default datum transformations

- *Ask for datum transformation when no default is defined*
- If you have worked with the ‘on-the-fly’ CRS transformation you can see the result of the transformation in the window below. You can find information about ‘Source CRS’ and ‘Destination CRS’ as well as ‘Source datum transform’ and ‘Destination datum transform’.

9.3.12 Locale Menu

- *Sobrepôr idioma do sistema e Idioma a utilizar em alternativa*
- Informação sobre a região do sistema activo local


9.3.13 Network Menu

Geral

- Define *Pesquisa de endereço WMS*, padrão como `http://geopole.org/wms/search?search=%1\&type=rss`
- Define *Tempo esgotado para pedidos de rede (ms)* - o padrão é 60000
- Define *Período de validade por defeito para tiles WMSC/WMTS (horas)* - por defeito é 24
- Define *Max retry in case of tile request errors*
- Define *User-Agent*

Configurações de cache

Define a *Pasta* e *Tamanho* para o cache.

- *Usar proxy para aceder à web* e define ‘Máquina’, ‘Porta’, ‘Utilizador’, e ‘Palavra-chave’.
- Configura o *Tipo de proxy*  de acordo com as necessidades.
 - *Default Proxy*: Proxy é determinado baseando-se na definição do proxy da aplicação em uso
 - *Socks5Proxy*: Proxy genérico para qualquer tipo de ligação. Suporta TCP, UDP, unindo a uma porta (ligações de entrada) e autenticação.
 - *HttpProxy*: Implementado usando o comando “LIGAR”, apenas suporta ligações TCP de saída; suporta autenticação.
 - *HttpCachingProxy*: Implementado usando comandos HTTP normais, é útil apenas no contexto de pedidos HTTP.
 - *FtpCachingProxy*: Implementado usando um proxy FTP, é útil no contexto de pedidos FTP.

Alguns URLs excluídos podem ser adicionados na caixa de texto debaixo das configurações de proxy (veja [Figure_Network_Tab](#)).

If you need more detailed information about the different proxy settings, please refer to the manual of the underlying QT library documentation at <http://doc.trolltech.com/4.5/qnetworkproxy.html#ProxyType-enum>.

Tip: Usando Proxies

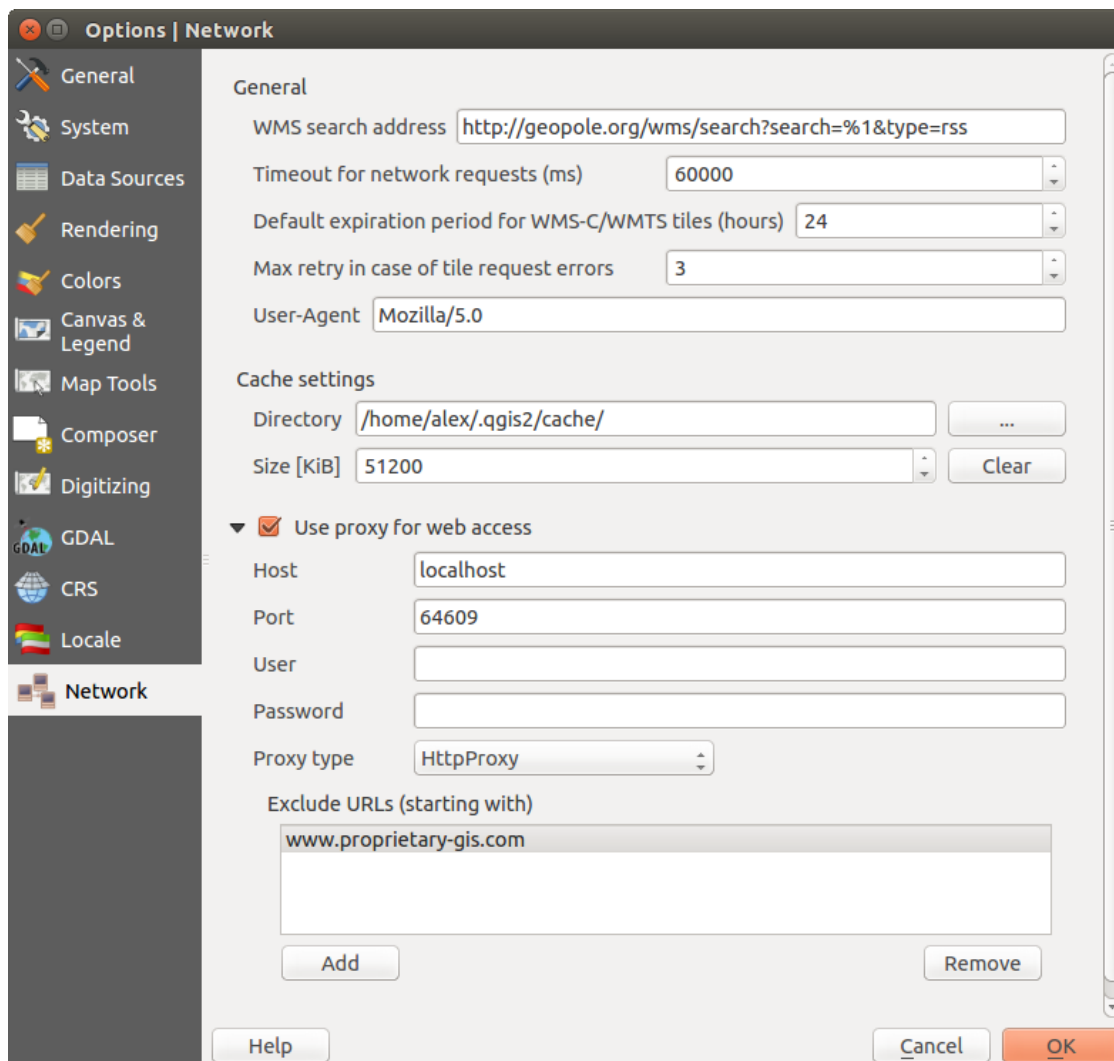




Figura 9.4: Configurações-proxy no QGIS

Using proxies can sometimes be tricky. It is useful to proceed by ‘trial and error’ with the above proxy types, to check to see if they succeed in your case.

Pode modificar as opções de acordo com as suas necessidades. Algumas das alterações pode requerer o reinício do QGIS antes de ser efectiva.

-  Settings are saved in a text file: `$HOME/.config/QGIS/QGIS2.conf`
- **X** pode encontrar as configurações em: `$HOME/Library/Preferences/org.qgis.qgis.plist`
- As configurações  são armazenadas no registo em: `HKEY\CURRENT_USER\Software\QGIS\qgis`

9.4 Personalização

A ferramenta de personalização permite que (des)active a maioria dos elementos na interface de utilizador QGIS. Isto pode ser muito útil se tiver um número elevado de módulos instalados que nunca usa e que estão a encher o ecrã.

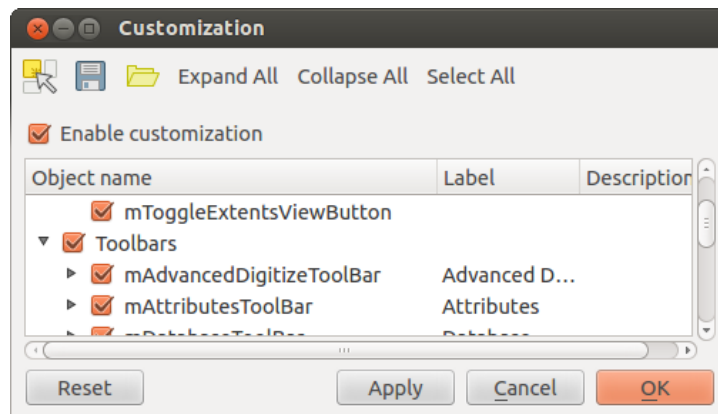









Figura 9.5: A janela de Personalização 

QGIS A Personalização está dividida em cinco grupos. No  *Menus* pode esconder as entradas na Barra Menu. No  *Painel* pode encontrar o painel de janelas. As janelas do Painel são aplicações que pode ser iniciadas e usadas como flutuantes, janelas de topo de nível ou embebidas na janela principal do QGIS como um widget ancorado (veja also *Painéis e Barras de Ferramentas*). Na  *Barra de Estado* características como a informação das coordenadas podem ser desactivadas. Na  *Caixa de Ferramentas* pode (des)activar os ícones da barraa de ferramentas do QGIS e em  *Widgets* pode (des)activatar janelas assim como os seus botões.

With  *Switch to catching widgets in main application*, you can click on elements in QGIS that you want to be hidden and find the corresponding entry in Customization (see [figure_customization](#)). You can also save your various setups for different use cases as well. Before your changes are applied, you need to restart QGIS.

Trabalhando com Projecções

O QGIS permite que os utilizadores definam de forma global e para todo o projecto um SC (Sistema de Coordenadas) para temas que não têm um SC pré-definido. Também permite que o utilizador defina sistemas de coordenadas personalizados e suporta a projecção dinâmica no ecrã de temas com diferentes SCs conseguindo sobrepondo-os correctamente.

10.1 Visão geral do Suporte a Projecções

O QGIS suporta aproximadamente 2.700 SCs conhecidos. Definições para cada SC são guardados numa base de dados SQLite que é instalada com o QGIS. Normalmente, não terá necessidade de manipular a base de dados directamente. De facto, ao fazê-lo poderá causar problemas. SCs definidos manualmente são guardados numa base de dados do utilizador. Veja a secção *Sistema de Coordenadas personalizado* para informação sobre a gestão dos seus sistemas de coordenadas.


Os SCs disponíveis no QGIS são baseados nos definidos pelo Grupo Europeu de Pesquisa Petrolífera (EPSG) e pelo Instituto Geográfico Nacional Francês (IGNF) e são em grande parte derivados das tabelas de referência espacial usadas pelo GDAL. Os identificadores EPSG presentes na base de dados podem ser usados para especificar um SC no QGIS.

Para poder usar a projecção dinâmica, os seus dados devem conter informação sobre o seu sistema de coordenadas ou vocês deverá definir um sistema global, para temas ou para o projecto. Para temas PostGIS, o QGIS usa o identificador de referência espacial que foi especificado aquando da criação do tema. Para dados suportados pelo OGR, o QGIS recorre à presença de um meio reconhecível para especificar o SC. No caso de shapefiles, isto significa um ficheiro contendo um texto bem-conhecido (WKT) especificando o SC. Este ficheiro de projecção tem o mesmo nome base que o shapefile e uma extensão `.prj`. Por exemplo, um shapefile chamado `alaska.shp` teria um ficheiro de projecção correspondente chamado `alaska.prj`.

Sempre que seleccionar um novo SC, as unidades do temas serão mudadas automaticamente no separador *Geral* das **ImActionsOptions** na janela de *Propriedades do Projecto*, no menu *Projecto* (Gnome, OSX) ou *Definições* (KDE, Windows).

10.2 Especificação Geral da Projecção

O QGIS inicia cada novo projecto usando a projecção global pré-definida. O SC global por omissão é o EPSG:4326 - WGS 84 (`proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`), e vem pré-definido no QGIS. Este valor pode ser alterado via o botão **[Selecionar...]** na primeira secção, que é usada para definir o sistema de coordenadas por omissão para novos projectos, como mostrado na [figure_projection_1](#). Este opção pode ser gravada para uso em sessões subsequentes do QGIS.

Quando usa temas que não têm um SC, tem de definir como o QGIS reage a estes temas. Isto pode ser feito globalmente ou por projecto no separador *SC* no *Definições* →  *Opções*.

As opções mostradas na [figure_projections_1](#) são:

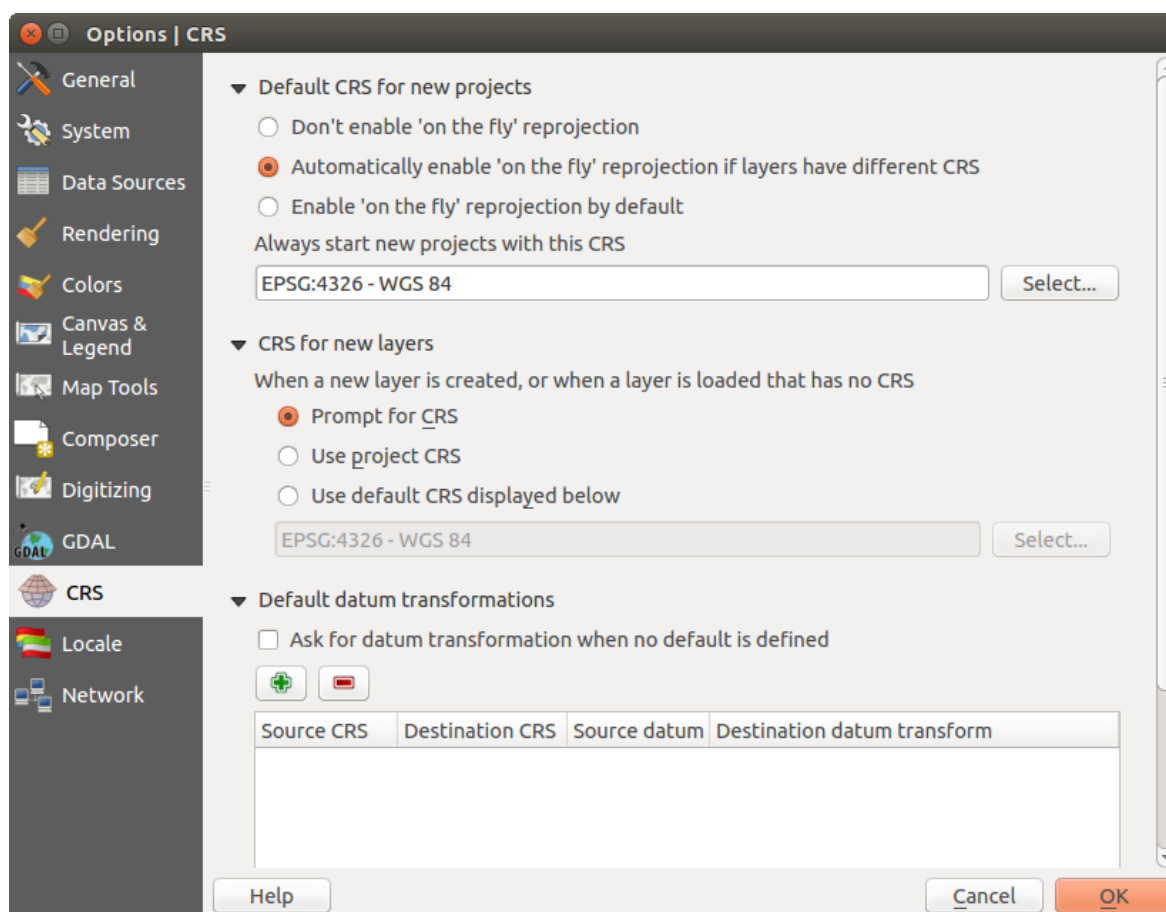





Figura 10.1: Separador SC na Janela de Opções QGIS 🐧



-  *Entrada de SC*
-  *Usar SC do projecto*
-  *Usar o SC pré-definido mostrado abaixo*

Se quer definir o sistema de coordenadas para um dado tema sem informação de SC, pode também fazê-lo no separador *Geral* da janela de propriedade raster e vector (ver *Menu Geral* para rasters e *Menu Geral* para vectores). Se o seu tema já tem um SC definido, este será mostrado como na *Janela das Propriedades da Camada Vectorial*.




Tip: SC na Legenda do Mapa



Clicando com o botão direito num tema na Legenda do Mapa (secção *Legenda do Mapa*) mostra dois atalhos SC. *Definir SC do tema* abre a janela de Selecção de Sistema de Coordenadas (ver *figure_projection_2*). *Definir SC do projecto a partir do Tema* redefine o SC do projecto usando o SC do tema.

10.3 Definir Reprojecção Dinâmica (RD)

O QGIS suporta reprojecção dinâmica para dados raster e vector. Contudo, RD não está activa por omissão. Para usar projecção dinâmica, deve activar a caixa de opção  *Activar transformação dinâmica de SC* no separador *SC* na janela  *Propriedades do Projecto*.

Há três formas de fazer isto:

1. Selecciono  **menuselecion: 'Propriedades do Projecto'** no menu *Projecto* (Gnome, OSX) ou *Definições* (KDE, Windows).
2. Clique no ícone  *estado SC* no canto inferior direito da barra de estado.
3. Active a projecção dinâmica por omissão no separador *SC* da janela *Opções* seleccionando a  *Activar reprojecção dinâmica por omissão* ou *Automaticamente activar a reprojecção dinâmica se temas têm diferentes SC*.

Se tem já carregado um tema e quer activar a projecção dinâmica, a melhor prática é abrir o separador *SC* na janela *Propriedades do Projecto*, seleccionar um SC, e activar a caixa de selecção  *Activar transformação dinâmica de SC*. O ícone  *Estado SC* deixará de estar desactivado (cinzento), e todos os temas serão dinamicamente reprojectados para o SC mostrado junto ao ícone.

O separador *SC* na janela *Propriedades do Projecto* contem cinco componentes importantes, como mostrado na *Figure_projections_2* e descritos abaixo:

1. **Activar transformação 'dinâmica' de SC** — Esta caixa de selecção é usada para activar ou desactivar a projecção dinâmica. Quando desligada, cada tema é desenhado usando as coordenadas lidas dos seus dados, e os componentes descritos abaixo estão inactivos. Quando ligada, as coordenadas de cada tema são projectadas para o sistema de coordenadas definido para o mapa.
2. **Filtro** — Se conhece o código EPSG, o identificador, ou o nome para um sistema de coordenadas, pode usar a função de pesquisa para o encontrar. Introduza o código EPSG, o identificador ou o nome.
3. **Sistemas de coordenadas usados recentemente** — Se tem certos SCs que usa frequentemente no seu trabalho SIG diário, estes serão mostrados nesta lista. Clique num destes itens para seleccionar o SC correspondente.
4. **Sistemas de coordenadas do mundo** — Esta é a lista de todos os SCs suportados pelo QGIS, incluindo sistemas de coordenadas Geográficas, Projectadas, e Personalizados. Para definir um SC, seleccione-o na lista expandindo o nó apropriado e seleccionando o SC. O SC activo está pré-seleccionado.
5. **Texto PROJ.4** — Este é um texto de SC usado pelo motor de projecções PROJ.4. Este texto é de apenas leitura e fornecido para fins de informação.

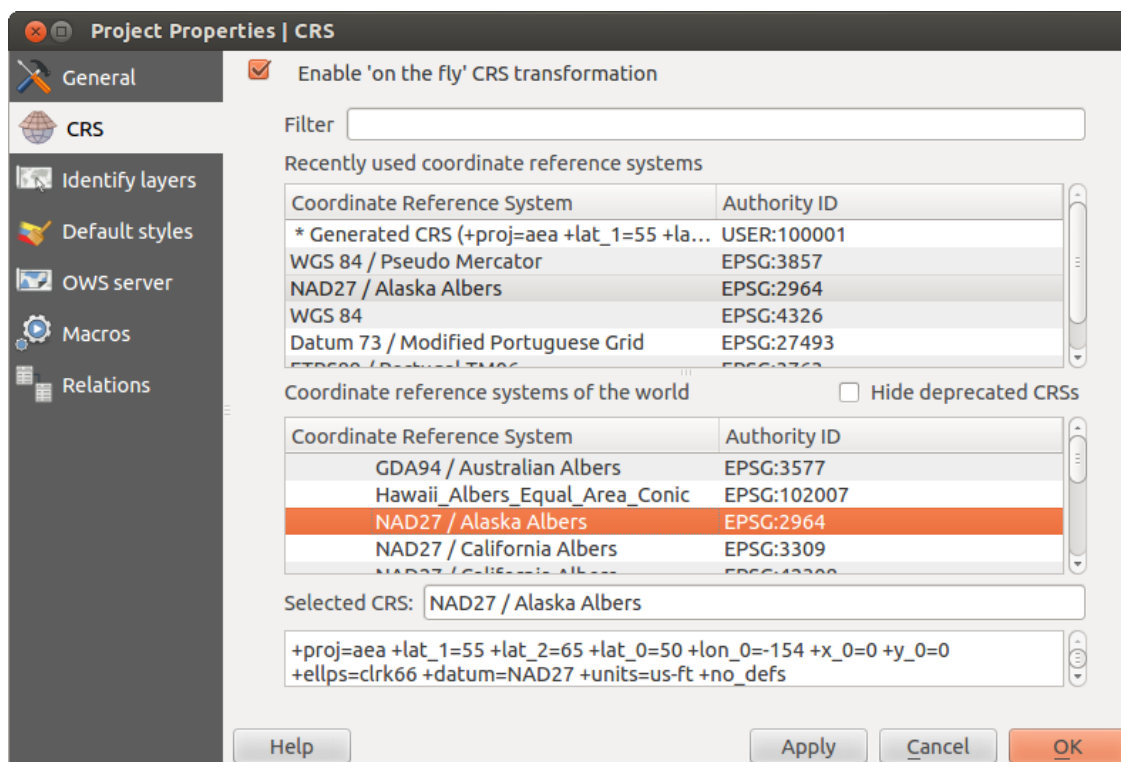



Figura 10.2: Janela de Propriedades do Projecto 

Tip: Janela de Propriedades do Projecto

Se abrir a janela *Propriedades do Projecto* a partir do menu **menuselecion:‘Projecto‘**, deve clicar no separador *SC* para ver as definições de SC.

Abriendo a janela a partir do ícone  Estado SC abrirá automaticamente o separador *SC*.

10.4 Sistema de Coordenadas personalizado


Se o QGIS não fornece o sistema de coordenadas que necessita, pode definir um SC personalizado. Para definir um SC, seleccione  *SC personalizado...* a partir do menu *Definições*. SCs personalizados são guardados na sua base de dados de utilizador QGIS. Além dos seus SCs, esta base de dados também contem os seus marcadores espaciais e outros dados personalizados.

Definir um SC personalizado no QGIS exige uma boa compreensão da biblioteca de projecção PROJ.4. Para começar, veja o “Cartographic Projection Procedures for the UNIX Environment - A User’s Manual” de Gerald I. Evenden, U.S. Geological Survey Open-File Report 90.284, 1990 (disponível em <ftp://ftp.remotesensing.org/proj/OF90-284.pdf>).

Este manual descreve o uso do `proj.4` e utilidades de linha de comando relacionados. Os parâmetros cartográficos usados com o `proj.4` são descritos no manual do utilizador e são os mesmo que os usados pelo QGIS.

A janela *Definição de Sistema de Coordenadas personalizado* exige apenas dois parâmetros para definir um SC do utilizador:

1. Um nome descritivo
2. Os parâmetros cartográficos no formato PROJ.4

Para criar um novo SC, clique no botão  Adicionar novo SC e introduza um nome descritivo e os parâmetros do SC.

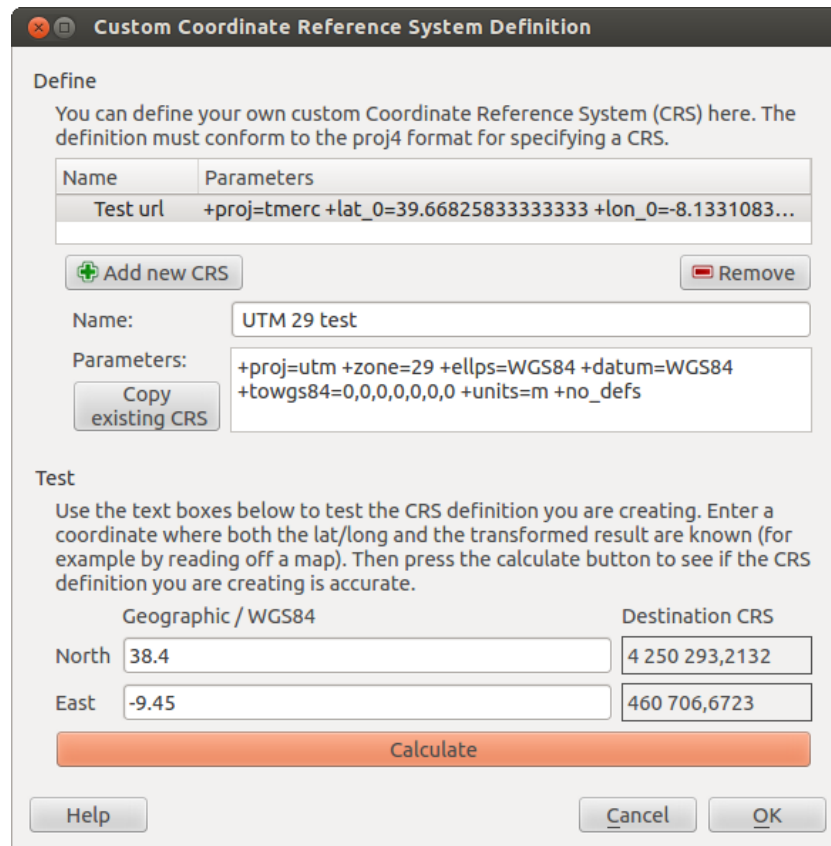



Figura 10.3: Janela de SC personalizado 

Note que os *Parâmetros* devem começar com um bloco `+proj=`, para representar o novo sistema de coordenadas.

Pode testar os parâmetros do seu SC para ver se fornecem resultados aceitáveis. Para isto, introduza valores conhecidos de latitude e longitude WGS 84 nos campos *Norte* e *Este*, respectivamente. Clique em **[Calcular]**, e compare os resultados com os seus valores conhecidos no seu sistema de coordenadas.

10.5 Transformações de datum pré-definidas

A projecção dinâmica depende da capacidade de transformar dados num 'SC pré-definido', e o QGIS usa o WGS 84. Para alguns SC existe um número de transformações disponíveis. O QGIS permite que defina a transformação a usar, de outro modo o QGIS usa a transformação pré-definida.

No separador *SC* dentro de *Seleção* →  *Opções* pode:

- configurar o QGIS para perguntar-lhe quando necessitar de definir uma transformação usando o **radiobutton** *Solicitar a transformação de datum quando nenhuma pré-definida existir*
- editar uma lista do utilizador de transformações pré-definidas.

O QGIS pergunta qual a transformação a usar abrindo uma janela mostrado o texto PROJ.4 que descreve as transformações de origem e destino. Mais informação pode ser encontrada ao passar com o rato sobre uma transformação. Pré-definições do utilizador podem ser gravadas seleccionando o **radiobutton** *Lembrar selecção*.

QGIS Browser

The QGIS Browser is a panel in QGIS that lets you easily navigate in your filesystem and manage geodata. You can have access to common vector files (e.g., ESRI shapefiles or MapInfo files), databases (e.g., PostGIS, Oracle, SpatiaLite or MS SQL Spatial) and WMS/WFS connections. You can also view your GRASS data (to get the data into QGIS, see *Integração GRASS SIG*).

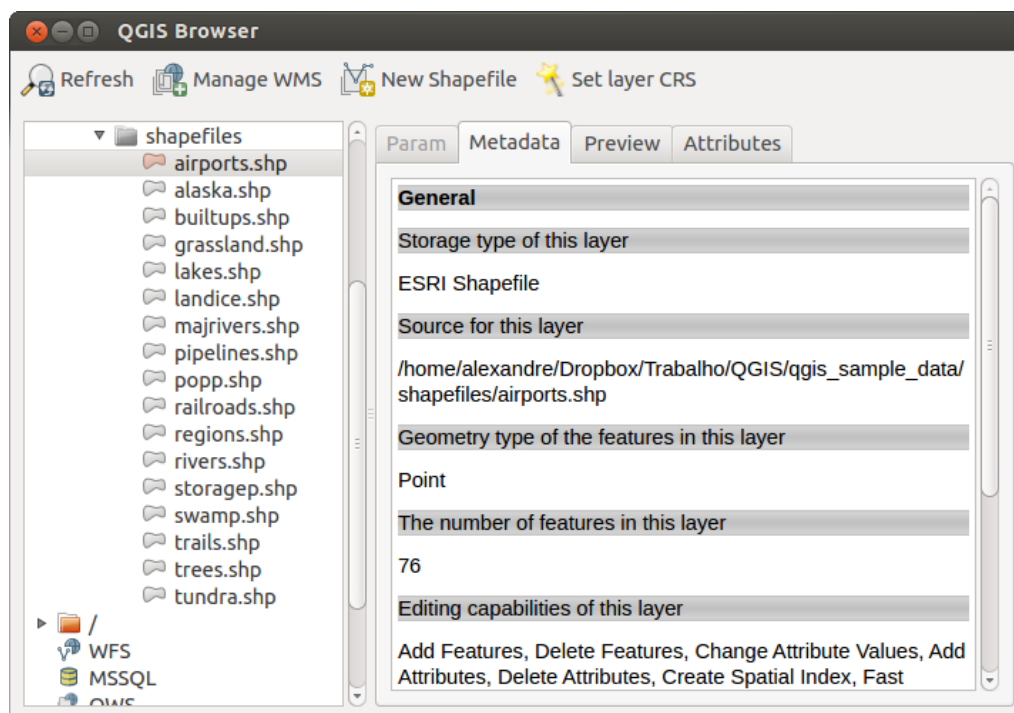





Figura 11.1: QGIS browser as a stand alone application 🐧

Use the QGIS Browser to preview your data. The drag-and-drop function makes it easy to get your data into the map view and the map legend.


1. Activate the QGIS Browser: Right-click on the toolbar and check *Browser* or select it from *Settings* → *Panels*.
2. Drag the panel into the legend window and release it.
3. Clique no separador *Pesquisar*.
4. Browse in your filesystem and choose the *shapefile* folder from *qgis_sample_data* directory.
5. Press the *Shift* key and select the *airports.shp* and *alaska.shp* files.
6. Press the left mouse button, then drag and drop the files into the map canvas.

7. Clique com o direito do rato na camada e escolha *Definir o SRC do projecto a partir da camada* . Para mais informações veja *Trabalhando com Projecções*.
8. Clique em  Ampliação Total para tornar as camadas visíveis.

There is a second browser available under *Settings* → *Panels*. This is handy when you need to move files or layers between locations.




1. Activate a second QGIS Browser: Right-click on the toolbar and check  *Browser (2)*, or select it from *Settings* → *Panels*.
2. Arraste o painel para a janela de legenda.
3. Navegue para o separador *Pesquisador (2)* e pesquise pela shapefile no seu sistema de ficheiros.
4. Select a file with the left mouse button. Now you can use the  *Add Selected Layers* icon to add it into the current project.

QGIS automatically looks for the coordinate reference system (CRS) and zooms to the layer extent if you work in a blank QGIS project. If there are already files in your project, the file will just be added, and in the case that it has the same extent and CRS, it will be visualized. If the file has another CRS and layer extent, you must first right-click on the layer and choose *Set Project CRS from Layer*. Then choose *Zoom to Layer Extent*.

The  *Filter files* function works on a directory level. Browse to the folder where you want to filter files and enter a search word or wildcard. The Browser will show only matching filenames – other data won't be displayed.

It's also possible to run the QGIS Browser as a stand-alone application.

Start the QGIS browser

-  Escreva na linha de comandos “qbrowser” .
-  Start the QGIS Browser using the Start menu or desktop shortcut.
-  The QGIS Browser is available from your Applications folder.

In [figure_browser_standalone_metadata](#), you can see the enhanced functionality of the stand-alone QGIS Browser. The *Param* tab provides the details of your connection-based datasets, like PostGIS or MSSQL Spatial. The *Metadata* tab contains general information about the file (see *Menu Metadados*). With the *Preview* tab, you can have a look at your files without importing them into your QGIS project. It's also possible to preview the attributes of your files in the *Attributes* tab.

Trabalhando com Informação Vectorial

12.1 Supported Data Formats

QGIS uses the OGR library to read and write vector data formats, including ESRI shapefiles, MapInfo and MicroStation file formats, AutoCAD DXF, PostGIS, SpatiaLite, Oracle Spatial and MSSQL Spatial databases, and many more. GRASS vector and PostgreSQL support is supplied by native QGIS data provider plugins. Vector data can also be loaded in read mode from zip and gzip archives into QGIS. As of the date of this document, 69 vector formats are supported by the OGR library (see OGR-SOFTWARE-SUITE in *Literatura e Referências Web*). The complete list is available at http://www.gdal.org/ogr/ogr_formats.html.

Note: Not all of the listed formats may work in QGIS for various reasons. For example, some require external commercial libraries, or the GDAL/OGR installation of your OS may not have been built to support the format you want to use. Only those formats that have been well tested will appear in the list of file types when loading a vector into QGIS. Other untested formats can be loaded by selecting *.*.

Working with GRASS vector data is described in Section *Integração GRASS SIG*.

This section describes how to work with several common formats: ESRI shapefiles, PostGIS layers, SpatiaLite layers, OpenStreetMap vectors, and Comma Separated data (CSV). Many of the features available in QGIS work the same, regardless of the vector data source. This is by design, and it includes the identify, select, labeling and attributes functions.

12.1.1 ESRI Shapefiles


The standard vector file format used in QGIS is the ESRI shapefile. Support is provided by the OGR Simple Feature Library (<http://www.gdal.org/ogr/>).

A shapefile actually consists of several files. The following three are required:

1. `.shp` file containing the feature geometries
2. `.dbf` file containing the attributes in dBase format
3. `.shx` index file

Shapefiles also can include a file with a `.prj` suffix, which contains the projection information. While it is very useful to have a projection file, it is not mandatory. A shapefile dataset can contain additional files. For further details, see the ESRI technical specification at <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.

Loading a Shapefile

To load a shapefile, start QGIS and click on the  Add Vector Layer toolbar button, or simply press `Ctrl+Shift+V`. This will bring up a new window (see [figure_vector_1](#)).

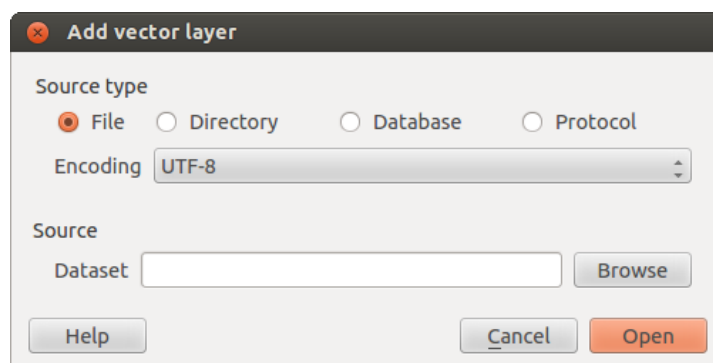



Figura 12.1: Add Vector Layer Dialog 

From the available options check *File*. Click on **[Browse]**. That will bring up a standard open file dialog (see [figure_vector_2](#)), which allows you to navigate the file system and load a shapefile or other supported data source. The selection box *Filter*  allows you to preselect some OGR-supported file formats.

You can also select the encoding for the shapefile if desired.

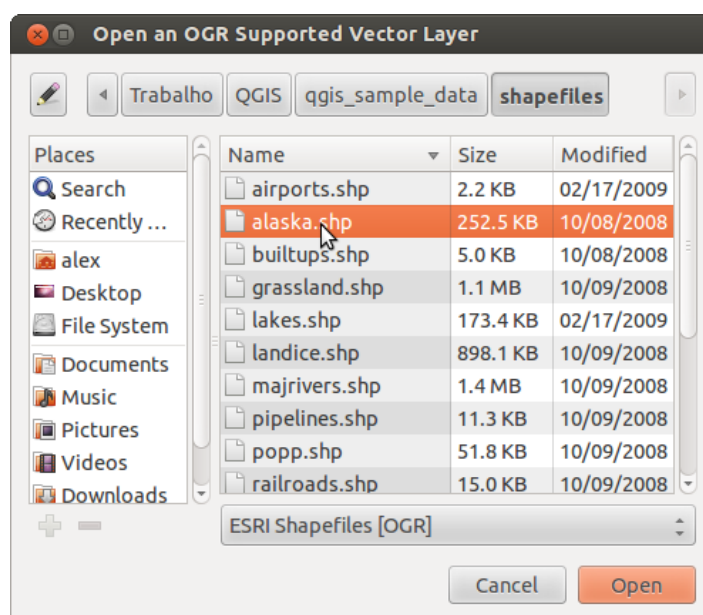


Figura 12.2: Open an OGR Supported Vector Layer Dialog 

Selecting a shapefile from the list and clicking **[Open]** loads it into QGIS. [Figure_vector_3](#) shows QGIS after loading the `alaska.shp` file.

Tip: Layer Colors

When you add a layer to the map, it is assigned a random color. When adding more than one layer at a time, different colors are assigned to each layer.

Once a shapefile is loaded, you can zoom around it using the map navigation tools. To change the style of a layer, open the *Layer Properties* dialog by double clicking on the layer name or by right-clicking on the name in

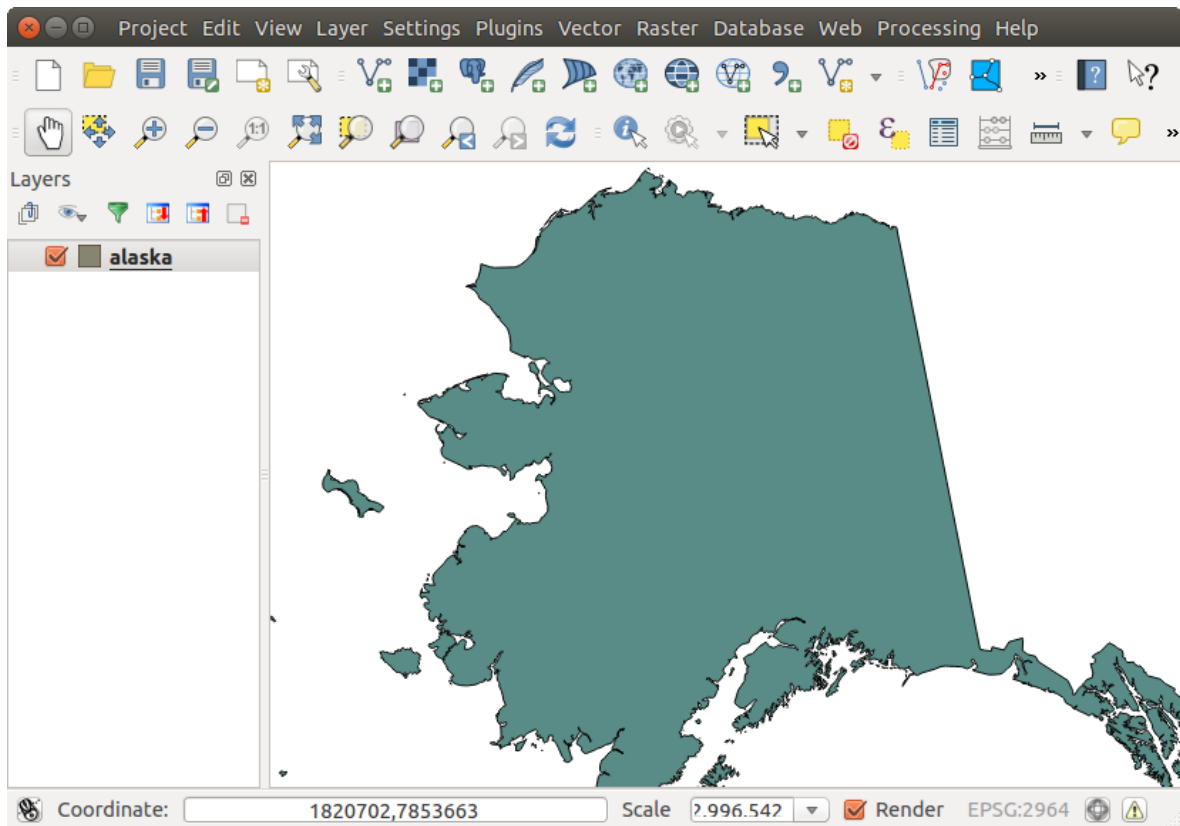


Figura 12.3: QGIS with Shapefile of Alaska loaded 🐧

the legend and choosing *Properties* from the context menu. See section *Estilos* for more information on setting symbology of vector layers.


Tip: Load layer and project from mounted external drives on OS X

On OS X, portable drives that are mounted beside the primary hard drive do not show up as expected under *File* → *Open Project*. We are working on a more OSX-native open/save dialog to fix this. As a workaround, you can type `/Volumes` in the *File name* box and press `Enter`. Then you can navigate to external drives and network mounts.

Improving Performance for Shapefiles

To improve the performance of drawing a shapefile, you can create a spatial index. A spatial index will improve the speed of both zooming and panning. Spatial indexes used by QGIS have a `.qix` extension.

Use these steps to create the index:

- Load a shapefile by clicking on the  `Add Vector Layer` toolbar button or pressing `Ctrl+Shift+V`.
- Open the *Layer Properties* dialog by double-clicking on the shapefile name in the legend or by right-clicking and choosing *Properties* from the context menu.
- In the *General* tab, click the **[Create Spatial Index]** button.




Problem loading a shape .prj file

If you load a shapefile with a `.prj` file and QGIS is not able to read the coordinate reference system from that file, you will need to define the proper projection manually within the *General* tab of the *Layer Properties* dialog





of the layer by clicking the **[Specify...]** button. This is due to the fact that `.prj` files often do not provide the complete projection parameters as used in QGIS and listed in the *CRS* dialog.

For the same reason, if you create a new shapefile with QGIS, two different projection files are created: a `.prj` file with limited projection parameters, compatible with ESRI software, and a `.qpj` file, providing the complete parameters of the used CRS. Whenever QGIS finds a `.qpj` file, it will be used instead of the `.prj`.

12.1.2 Loading a MapInfo Layer

 To load a MapInfo layer, click on the  **Add Vector Layer** toolbar button; or type `Ctrl+Shift+V`, change the file type filter *Files of type*  to 'Mapinfo File [OGR] (*.mif *.tab *.MIF *.TAB)' and select the MapInfo layer you want to load.

12.1.3 Loading an ArcInfo Binary Coverage

 To load an ArcInfo Binary Coverage, click on the  **Add Vector Layer** toolbar button or press `Ctrl+Shift+V` to open the *Add Vector Layer* dialog. Select  *Directory* as *Source type*. Change the file type filter *Files of type*  to 'Arc/Info Binary Coverage'. Navigate to the directory that contains the coverage file, and select it.

Similarly, you can load directory-based vector files in the UK National Transfer Format, as well as the raw TIGER Format of the US Census Bureau.

12.1.4 Delimited Text Files

Tabular data is a very common and widely used format because of its simplicity and readability – data can be viewed and edited even in a plain text editor. A delimited text file is an attribute table with each column separated by a defined character and each row separated by a line break. The first row usually contains the column names. A common type of delimited text file is a CSV (Comma Separated Values), with each column separated by a comma.

Such data files can also contain positional information in two main forms:

- As point coordinates in separate columns
- As well-known text (WKT) representation of geometry

QGIS allows you to load a delimited text file as a layer or ordinal table. But first check that the file meets the following requirements:

1. The file must have a delimited header row of field names. This must be the first line in the text file.
2. The header row must contain field(s) with geometry definition. These field(s) can have any name.
3. The X and Y coordinates (if geometry is defined by coordinates) must be specified as numbers. The coordinate system is not important.

As an example of a valid text file, we import the elevation point data file `elevp.csv` that comes with the QGIS sample dataset (see section *Amostra de Dados*):

```
X;Y;ELEV
-300120;7689960;13
-654360;7562040;52
1640;7512840;3
[...]
```

Some items to note about the text file:

1. The example text file uses `;` (semicolon) as delimiter. Any character can be used to delimit the fields.
2. The first row is the header row. It contains the fields X, Y and ELEV.
3. No quotes (") are used to delimit text fields.

4. The X coordinates are contained in the X field.
5. The Y coordinates are contained in the Y field.

Loading a delimited text file

Click the toolbar icon  Add Delimited Text Layer in the *Manage layers* toolbar to open the *Create a Layer from a Delimited Text File* dialog, as shown in [figure_delimited_text_1](#).

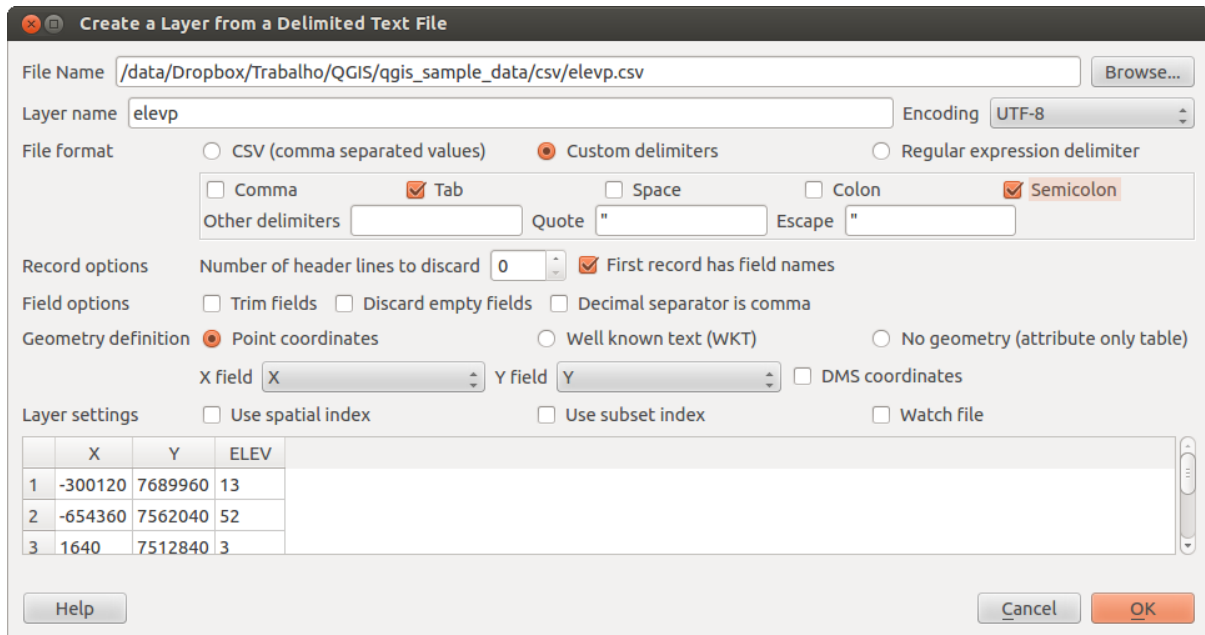



Figura 12.4: Delimited Text Dialog 

First, select the file to import (e.g., `qgis_sample_data/csv/elevp.csv`) by clicking on the **[Browse]** button. Once the file is selected, QGIS attempts to parse the file with the most recently used delimiter. To enable QGIS to properly parse the file, it is important to select the correct delimiter. You can specify a delimiter by activating *Custom delimiters*, or by activating *Regular expression delimiter* and entering text into the *Expression* field. For example, to change the delimiter to tab, use `\t` (this is a regular expression for the tab character).

Once the file is parsed, set *Geometry definition* to *Point coordinates* and choose the X and Y fields from the dropdown lists. If the coordinates are defined as degrees/minutes/seconds, activate the *DMS coordinates* checkbox.



Finally, enter a layer name (e.g., `elevp`), as shown in [figure_delimited_text_1](#). To add the layer to the map, click **[OK]**. The delimited text file now behaves as any other map layer in QGIS.

There is also a helper option that allows you to trim leading and trailing spaces from fields — *Trim fields*. Also, it is possible to *Discard empty fields*. If necessary, you can force a comma to be the decimal separator by activating *Decimal separator is comma*.

If spatial information is represented by WKT, activate the *Well Known Text* option and select the field with the WKT definition for point, line or polygon objects. If the file contains non-spatial data, activate *No geometry (attribute only table)* and it will be loaded as an ordinal table.

Additionally, you can enable:

- *Use spatial index* to improve the performance of displaying and spatially selecting features.



-  *Use subset index.*
-  *Watch file* to watch for changes to the file by other applications while QGIS is running.

12.1.5 OpenStreetMap data

In recent years, the OpenStreetMap project has gained popularity because in many countries no free geodata such as digital road maps are available. The objective of the OSM project is to create a free editable map of the world from GPS data, aerial photography or local knowledge. To support this objective, QGIS provides support for OSM data.

Loading OpenStreetMap Vectors





QGIS integrates OpenStreetMap import as a core functionality.

- To connect to the OSM server and download data, open the menu *Vector* → *Openstreetmap* → *Load data*. You can skip this step if you already obtained an `.osm` XML file using JOSM, Overpass API or any other source.
- The menu *Vector* → *Openstreetmap* → *Import topology from an XML file* will convert your `.osm` file into a SpatialLite database and create a corresponding database connection.
- The menu *Vector* → *Openstreetmap* → *Export topology to SpatialLite* then allows you to open the database connection, select the type of data you want (points, lines, or polygons) and choose tags to import. This creates a SpatialLite geometry layer that you can add to your project by clicking on the  *Add SpatialLite Layer* toolbar button or by selecting the  *Add SpatialLite Layer...* option from the *Layer* menu (see section *SpatialLite Layers*).

12.1.6 PostGIS Layers

PostGIS layers are stored in a PostgreSQL database. The advantages of PostGIS are the spatial indexing, filtering and query capabilities it provides. Using PostGIS, vector functions such as select and identify work more accurately than they do with OGR layers in QGIS.

Creating a stored Connection

 The first time you use a PostGIS data source, you must create a connection to the PostgreSQL database that contains the data. Begin by clicking on the  *Add PostGIS Layer* toolbar button, selecting the  *Add PostGIS Layer...* option from the *Layer* menu, or typing `Ctrl+Shift+D`. You can also open the *Add Vector Layer* dialog and select  *Database*. The *Add PostGIS Table(s)* dialog will be displayed. To access the connection manager, click on the **[New]** button to display the *Create a New PostGIS Connection* dialog. The parameters required for a connection are:

- **Name:** A name for this connection. It can be the same as *Database*.
- **Service:** Service parameter to be used alternatively to hostname/port (and potentially database). This can be defined in `pg_service.conf`.
- **Host:** Name of the database host. This must be a resolvable host name such as would be used to open a telnet connection or ping the host. If the database is on the same computer as QGIS, simply enter `'localhost'` here.
- **Port:** Port number the PostgreSQL database server listens on. The default port is 5432.
- **Database:** Name of the database.

- **SSL mode:** How the SSL connection will be negotiated with the server. Note that massive speedups in PostGIS layer rendering can be achieved by disabling SSL in the connection editor. The following options are available:
 - Disable: Only try an unencrypted SSL connection.
 - Allow: Try a non-SSL connection. If that fails, try an SSL connection.
 - Prefer (the default): Try an SSL connection. If that fails, try a non-SSL connection.
 - Require: Only try an SSL connection.
- **Username:** User name used to log in to the database.
- **Password:** Password used with *Username* to connect to the database.

Optionally, you can activate the following checkboxes:

- *Save Username*
- *Save Password*
- *Only look in the geometry_columns table*
- *Don't resolve type of unrestricted columns (GEOMETRY)*
- *Only look in the 'public' schema*
- *Also list tables with no geometry*
- *Use estimated table metadata*

Once all parameters and options are set, you can test the connection by clicking on the **[Test Connect]** button.

Loading a PostGIS Layer



Once you have one or more connections defined, you can load layers from the PostgreSQL database. Of course, this requires having data in PostgreSQL. See section [Importing Data into PostgreSQL](#) for a discussion on importing data into the database.

To load a layer from PostGIS, perform the following steps:

- If the *Add PostGIS layers* dialog is not already open, selecting the **Add PostGIS Layer...** option from the *Layer* menu or typing `Ctrl+Shift+D` opens the dialog.
- Choose the connection from the drop-down list and click **[Connect]**.
- Select or unselect *Also list tables with no geometry*.
- Optionally, use some *Search Options* to define which features to load from the layer, or use the **[Build query]** button to start the *Query builder* dialog.
- Find the layer(s) you wish to add in the list of available layers.
- Select it by clicking on it. You can select multiple layers by holding down the `Shift` key while clicking. See section [Ferramenta de Consulta](#) for information on using the PostgreSQL Query Builder to further define the layer.
- Click on the **[Add]** button to add the layer to the map.

Tip: PostGIS Layers

Normally, a PostGIS layer is defined by an entry in the `geometry_columns` table. From version 0.9.0 on, QGIS can load layers that do not have an entry in the `geometry_columns` table. This includes both tables and views. Defining a spatial view provides a powerful means to visualize your data. Refer to your PostgreSQL manual for information on creating views.

Some details about PostgreSQL layers

This section contains some details on how QGIS accesses PostgreSQL layers. Most of the time, QGIS should simply provide you with a list of database tables that can be loaded, and it will load them on request. However, if you have trouble loading a PostgreSQL table into QGIS, the information below may help you understand any QGIS messages and give you direction on changing the PostgreSQL table or view definition to allow QGIS to load it.

QGIS requires that PostgreSQL layers contain a column that can be used as a unique key for the layer. For tables, this usually means that the table needs a primary key, or a column with a unique constraint on it. In QGIS, this column needs to be of type int4 (an integer of size 4 bytes). Alternatively, the ctid column can be used as primary key. If a table lacks these items, the oid column will be used instead. Performance will be improved if the column is indexed (note that primary keys are automatically indexed in PostgreSQL).


If the PostgreSQL layer is a view, the same requirement exists, but views do not have primary keys or columns with unique constraints on them. You have to define a primary key field (has to be integer) in the QGIS dialog before you can load the view. If a suitable column does not exist in the view, QGIS will not load the layer. If this occurs, the solution is to alter the view so that it does include a suitable column (a type of integer and either a primary key or with a unique constraint, preferably indexed).

QGIS offers a checkbox **Select at id** that is activated by default. This option gets the ids without the attributes which is faster in most cases. It can make sense to disable this option when you use expensive views.

12.1.7 Importing Data into PostgreSQL

Data can be imported into PostgreSQL/PostGIS using several tools, including the SPIT plugin and the command line tools shp2pgsql and ogr2ogr.

DB Manager

QGIS comes with a core plugin named  DB Manager. It can be used to load shapefiles and other data formats, and it includes support for schemas. See section *Módulo Gestor BD* for more information.

shp2pgsql

PostGIS includes an utility called **shp2pgsql** that can be used to import shapefiles into a PostGIS-enabled database. For example, to import a shapefile named `lakes.shp` into a PostgreSQL database named `gis_data`, use the following command:

```
shp2pgsql -s 2964 lakes.shp lakes_new | psql gis_data
```

This creates a new layer named `lakes_new` in the `gis_data` database. The new layer will have a spatial reference identifier (SRID) of 2964. See section *Trabalhando com Projeções* for more information on spatial reference systems and projections.

Tip: Exporting datasets from PostGIS

Like the import tool **shp2pgsql**, there is also a tool to export PostGIS datasets as shapefiles: **pgsql2shp**. This is shipped within your PostGIS distribution.


ogr2ogr

Besides **shp2pgsql** and **DB Manager**, there is another tool for feeding geodata in PostGIS: **ogr2ogr**. This is part of your GDAL installation.


To import a shapefile into PostGIS, do the following:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres
password=topsecret" alaska.shp
```

This will import the shapefile `alaska.shp` into the PostGIS database `postgis` using the user `postgres` with the password `topsecret` on host server `myhost.de`.

Note that OGR must be built with PostgreSQL to support PostGIS. You can verify this by typing (in )

```
ogrinfo --formats | grep -i post
```

If you prefer to use PostgreSQL's **COPY** command instead of the default **INSERT INTO** method, you can export the following environment variable (at least available on  and **X**):

```
export PG_USE_COPY=YES
```

ogr2ogr does not create spatial indexes like **shp2pgsql** does. You need to create them manually, using the normal SQL command **CREATE INDEX** afterwards as an extra step (as described in the next section *Improving Performance*).

Improving Performance

Retrieving features from a PostgreSQL database can be time-consuming, especially over a network. You can improve the drawing performance of PostgreSQL layers by ensuring that a PostGIS spatial index exists on each layer in the database. PostGIS supports creation of a GiST (Generalized Search Tree) index to speed up spatial searches of the data (GiST index information is taken from the PostGIS documentation available at <http://postgis.refrains.net>).

The syntax for creating a GiST index is:

```
CREATE INDEX [indexname] ON [tablename]
  USING GIST ( [geometryfield] GIST_GEOMETRY_OPS );
```

Note that for large tables, creating the index can take a long time. Once the index is created, you should perform a `VACUUM ANALYZE`. See the PostGIS documentation (POSTGIS-PROJECT *Literatura e Referências Web*) for more information.

The following is an example of creating a GiST index:

```
gsherman@madison:~/current$ psql gis_data
Welcome to psql 8.3.0, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit

gis_data=# CREATE INDEX sidx_alaska_lakes ON alaska_lakes
gis_data=# USING GIST (the_geom GIST_GEOMETRY_OPS);
CREATE INDEX
gis_data=# VACUUM ANALYZE alaska_lakes;
VACUUM
gis_data=# \q
gsherman@madison:~/current$
```

12.1.8 Vector layers crossing 180° longitude

Many GIS packages don't wrap vector maps with a geographic reference system (lat/lon) crossing the 180 degrees longitude line (http://postgis.refrains.net/documentation/manual-2.0/ST_Shift_Longitude.html). As result, if

we open such a map in QGIS, we will see two far, distinct locations, that should appear near each other. In [Figure_vector_4](#), the tiny point on the far left of the map canvas (Chatham Islands) should be within the grid, to the right of the New Zealand main islands.



Figura 12.5: Map in lat/lon crossing the 180° longitude line 🐧

A work-around is to transform the longitude values using PostGIS and the **ST_Shift_Longitude** function. This function reads every point/vertex in every component of every feature in a geometry, and if the longitude coordinate is $< 0^\circ$, it adds 360° to it. The result is a $0^\circ - 360^\circ$ version of the data to be plotted in a 180° -centric map.

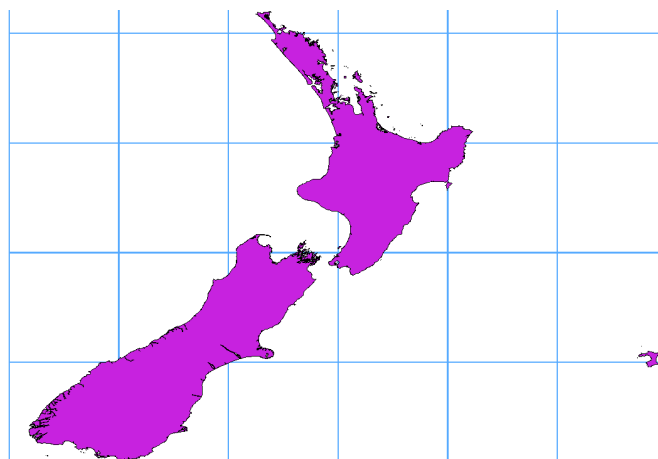


Figura 12.6: Crossing 180° longitude applying the **ST_Shift_Longitude** function

Usage

- Import data into PostGIS (*Importing Data into PostgreSQL*) using, for example, the DB Manager plugin.
- Use the PostGIS command line interface to issue the following command (in this example, “TABLE” is the actual name of your PostGIS table): `gis_data=# update TABLE set the_geom=ST_Shift_Longitude(the_geom);`
- If everything went well, you should receive a confirmation about the number of features that were updated. Then you’ll be able to load the map and see the difference ([Figure_vector_5](#)).

12.1.9 Spatialite Layers

🐧 The first time you load data from a Spatialite database, begin by clicking on the 🐧 Add Spatialite Layer toolbar button, or by selecting the 🐧 Add Spatialite Layer... option from the *Layer* menu, or by typing `Ctrl+Shift+L`. This will bring up a window that will allow you either to connect to a Spatialite database already known to QGIS, which you can choose from the drop-down menu, or to define a new connection to a new database. To define a new connection, click on **[New]** and use the file browser to point to your Spatialite database, which is a file with a `.sqlite` extension.

If you want to save a vector layer to SpatiaLite format, you can do this by right clicking the layer in the legend. Then, click on *Save as...*, define the name of the output file, and select 'SpatiaLite' as format and the CRS. Also, you can select 'SQLite' as format and then add `SPATIALITE=YES` in the OGR data source creation option field. This tells OGR to create a SpatiaLite database. See also http://www.gdal.org/ogr/drv_sqlite.html.

QGIS also supports editable views in SpatiaLite.




Creating a new SpatiaLite layer

If you want to create a new SpatiaLite layer, please refer to section *Criando uma nova camada SpatiaLite*.

Tip: SpatiaLite data management Plugins

For SpatiaLite data management, you can also use several Python plugins: QSpatiaLite, SpatiaLite Manager or DB Manager (core plugin, recommended). If necessary, they can be downloaded and installed with the Plugin Installer.




12.1.10 MSSQL Spatial Layers

 QGIS also provides native MS SQL 2008 support. The first time you load MSSQL Spatial data, begin by clicking on the  `Add MSSQL Spatial Layer` toolbar button or by selecting the  `Add MSSQL Spatial Layer...` option from the *Layer* menu, or by typing `Ctrl+Shift+M`.

12.1.11 Oracle Spatial Layers

The spatial features in Oracle Spatial aid users in managing geographic and location data in a native type within an Oracle database. QGIS now has support for such layers.






Creating a stored Connection

 The first time you use an Oracle Spatial data source, you must create a connection to the database that contains the data. Begin by clicking on the  `Add Oracle Spatial Layer` toolbar button, selecting the  `Add Oracle Spatial Layer...` option from the *Layer* menu, or typing `Ctrl+Shift+O`. To access the connection manager, click on the **[New]** button to display the *Create a New Oracle Spatial Connection* dialog. The parameters required for a connection are:

- **Name:** A name for this connection. It can be the same as *Database*
- **Database:** SID or SERVICE_NAME of the Oracle instance.
- **Host:** Name of the database host. This must be a resolvable host name such as would be used to open a telnet connection or ping the host. If the database is on the same computer as QGIS, simply enter '*localhost*' here.
- **Port:** Port number the PostgreSQL database server listens on. The default port is 1521.
- **Username:** Username used to login to the database.
- **Password:** Password used with *Username* to connect to the database.

Optionally, you can activate following checkboxes:



- *Save Username* Indicates whether to save the database username in the connection configuration.
- *Save Password* Indicates whether to save the database password in the connection settings.

-  *Only look in meta data table* Restricts the displayed tables to those that are in the `all_sdo_geom_metadata` view. This can speed up the initial display of spatial tables.
-  *Only look for user's tables* When searching for spatial tables, restrict the search to tables that are owned by the user.
-  *Also list tables with no geometry* Indicates that tables without geometry should also be listed by default.
-  *Use estimated table statistics for the layer metadata* When the layer is set up, various metadata are required for the Oracle table. This includes information such as the table row count, geometry type and spatial extents of the data in the geometry column. If the table contains a large number of rows, determining this metadata can be time-consuming. By activating this option, the following fast table metadata operations are done: Row count is determined from `all_tables.num_rows`. Table extents are always determined with the `SDO_TUNE.EXTENTS_OF` function, even if a layer filter is applied. Table geometry is determined from the first 100 non-null geometry rows in the table.
-  *Only existing geometry types* Only list the existing geometry types and don't offer to add others.


Once all parameters and options are set, you can test the connection by clicking on the **[Test Connect]** button.

Tip: QGIS User Settings and Security




Depending on your computing environment, storing passwords in your QGIS settings may be a security risk. Passwords are saved in clear text in the system configuration and in the project files! Your customized settings for QGIS are stored based on the operating system:

-  The settings are stored in your home directory in `~/ .qgis2`.
-  The settings are stored in the registry.

Loading an Oracle Spatial Layer

 Once you have one or more connections defined, you can load layers from the Oracle database. Of course, this requires having data in Oracle.

To load a layer from Oracle Spatial, perform the following steps:

- If the *Add Oracle Spatial layers* dialog is not already open, click on the  **Add Oracle Spatial Layer** toolbar button.
- Choose the connection from the drop-down list and click **[Connect]**.
- Select or unselect  *Also list tables with no geometry*.
- Optionally, use some  *Search Options* to define which features to load from the layer or use the **[Build query]** button to start the *Query builder* dialog.
- Find the layer(s) you wish to add in the list of available layers.
- Select it by clicking on it. You can select multiple layers by holding down the `Shift` key while clicking. See section *Ferramenta de Consulta* for information on using the Oracle Query Builder to further define the layer.
- Click on the **[Add]** button to add the layer to the map.

Tip: Oracle Spatial Layers

Normally, an Oracle Spatial layer is defined by an entry in the `USER_SDO_METADATA` table.


12.2 The Symbol Library

12.2.1 Presentation

The Symbol Library is the place where users can create generic symbols to be used in several QGIS projects. It allows users to export and import symbols, groups symbols and add, edit and remove symbols. You can open it with the *Settings* → *Style Library* or from the **Style** tab in the vector layer's *Properties*.


Share and import symbols


Users can export and import symbols in two main formats: qml (QGIS format) and SLD (OGC standard). Note that SLD format is not fully supported by QGIS.

 share item displays a drop down list to let the user import or export symbols.

Groups and smart groups






Groups are categories of Symbols and smart groups are dynamic groups.

To create a group, right-click on an existing group or on the main **Groups** directory in the left of the library. You can also select a group and click on the  add item button.

To add a symbol into a group, you can either right click on a symbol then choose *Apply group* and then the group name added before. There is a second way to add several symbols into group: just select a group and click  and choose **Group Symbols**. All symbols display a checkbox that allow you to add the symbol into the selected groups. When finished, you can click on the same button, and choose **Finish Grouping**.

Create **Smart Symbols** is similar to creating group, but instead select **Smart Groups**. The dialog box allow user to choose the expression to select symbols in order to appear in the smart group (contains some tags, member of a group, have a string in its name, etc.)

Add, edit, remove symbol

With the *Style manager* from the [**Symbol**]  menu you can manage your symbols. You can  add item,  edit item,  remove item and  share item. 'Marker' symbols, 'Line' symbols, 'Fill' patterns and 'colour ramps' can be used to create the symbols. The symbols are then assigned to 'All Symbols', 'Groups' or 'Smart groups'.

For each kind of symbols, you will find always the same dialog structure:

- at the top left side a symbol representation
- under the symbol representation the symbol tree show the symbol layers
- at the right you can setup some parameter (unit,transparency, color, size and rotation)
- under these parameteres you find some symbol from the symbol library

The symbol tree allow adding, removing or protect new simple symbol. You can move up or down the symbol layer.

More detailed settings can be made when clicking on the second level in the *Symbol layers* dialog. You can define *Symbol layers* that are combined afterwards. A symbol can consist of several *Symbol layers*. Settings will be shown later in this chapter.

Tip: Note that once you have set the size in the lower levels of the *Symbol layers* dialog, the size of the whole symbol can be changed with the *Size* menu in the first level again. The size of the lower levels changes accordingly, while the size ratio is maintained.

12.2.2 Marker Symbols

Marker symbols have several symbol layer types:

- Ellipse marker
- Font marker
- Simple marker (default)
- SVG marker
- Vector Field marker

The following settings are possible:

- *Symbol layer type*: You have the option to use Ellipse markers, Font markers, Simple markers, SVG markers and Vector Field markers.
- *colors*
- *Size*
- *Outline style*
- *Outline width*
- *Angle*
- *Offset X,Y*: You can shift the symbol in the x- or y-direction.
- *Anchor point*
- *Data defined properties ...*

12.2.3 Line Symbols

Line marker symbols have only two symbol layer types:

- Marker line
- Simple line (default)

The default symbol layer type draws a simple line whereas the other display a marker point regularly on the line. You can choose different location vertex, interval or central point. Marker line can have offset along the line or offset line. Finally, *rotation* allows you to change the orientation of the symbol.

The following settings are possible:

- *colour*
- *Pen width*
- *Offset*
- *Pen style*
- *Join style*
- *Cap style*
- *Use custom dash pattern*
- *Dash pattern unit*
- *Data defined properties ...*

12.2.4 Polygon Symbols

Polygon marker symbols have also several symbol layer types:

- Centroid fill
- Gradient fill
- Line pattern fill
- Point pattern fill
- SVG fill
- Shapeburst fill
- Simple fill (default)
- Outline: Marker line (same as line marker)
- Outline: simple line (same as line marker)

The following settings are possible:

- *Colors* for the border and the fill.
- *Fill style*
- *Border style*
- *Border width*
- *Offset X,Y*
- *Data defined properties ...*

Using the color combo box, you can drag and drop color for one color button to another button, copy-paste color, pick color from somewhere, choose a color from the palette or from recent or standard color. The combo box allow you to fill in the feature with transparency. You can also just clic on the button to open the palette dialog. Note that you can import color from some external software like GIMP.

‘Gradient Fill’ *Symbol layer type* allows you to select between a *Two color* and *Color ramp* setting. You can use the *Feature centroid as Referencepoint*. All fills ‘Gradient Fill’ *Symbol layer type* is also available through the *Symbol* menu of the Categorized and Graduated Renderer and through the *Rule properties* menu of the Rule-based renderer. Other possibility is to choose a ‘shapeburst fill’ which is a buffered gradient fill, where a gradient is drawn from the boundary of a polygon towards the polygon’s centre. Configurable parameters include distance from the boundary to shade, use of color ramps or simple two color gradients, optional blurring of the fill and offsets.

It is possible to only draw polygon borders inside the polygon. Using ‘Outline: Simple line’ select *Draw line only inside polygon*.

12.2.5 Color ramp

You can create a custom color ramp choosing *New color ramp...* from the *color ramp* drop-down menu. A dialog will prompt for the ramp type: Gradient, Random, colorBrewer, or cpt-city. The first three have options for number of steps and/or multiple stops in the color ramp. You can use the *Invert* option while classifying the data with a color ramp. See [figure_symbology_3](#) for an example of custom color ramp and [figure_symbology_3a](#) for the cpt-city dialog.

The cpt-city option opens a new dialog with hundreds of themes included ‘out of the box’.

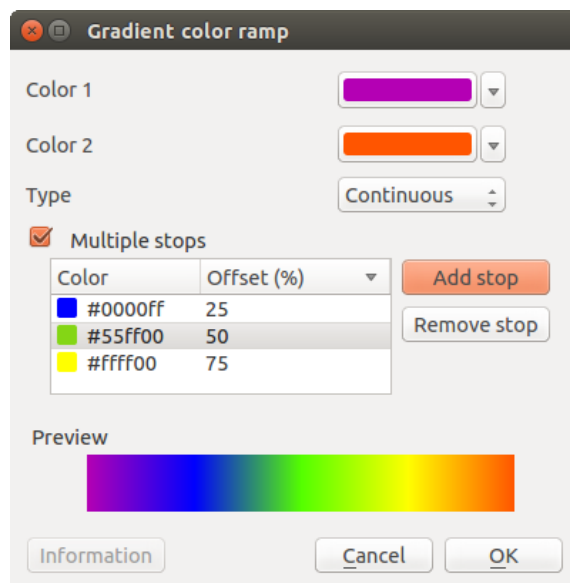


Figura 12.7: Example of custom gradient color ramp with multiple stops 

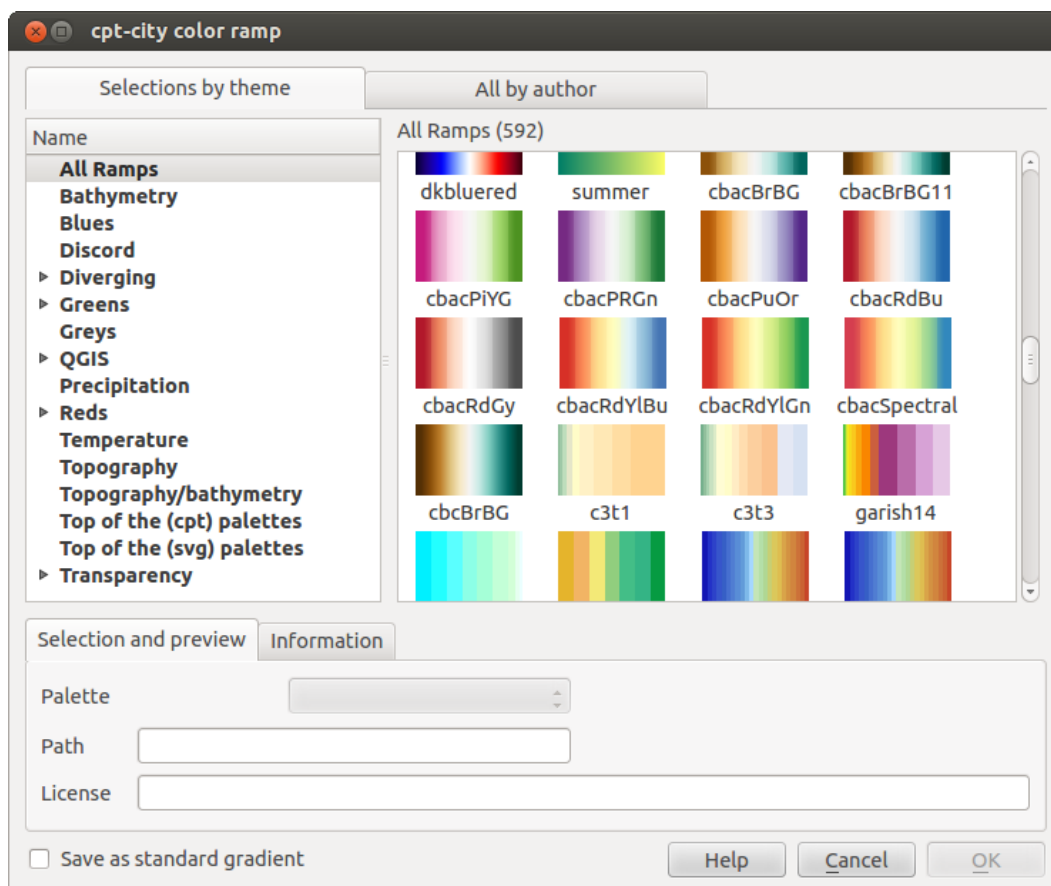



Figura 12.8: cpt-city dialog with hundreds of color ramps 

12.3 Janela das Propriedades da Camada Vectorial

The *Layer Properties* dialog for a vector layer provides information about the layer, symbology settings and labeling options. If your vector layer has been loaded from a PostgreSQL/PostGIS datastore, you can also alter the underlying SQL for the layer by invoking the *Query Builder* dialog on the *General* tab. To access the *Layer Properties* dialog, double-click on a layer in the legend or right-click on the layer and select *Properties* from the pop-up menu.

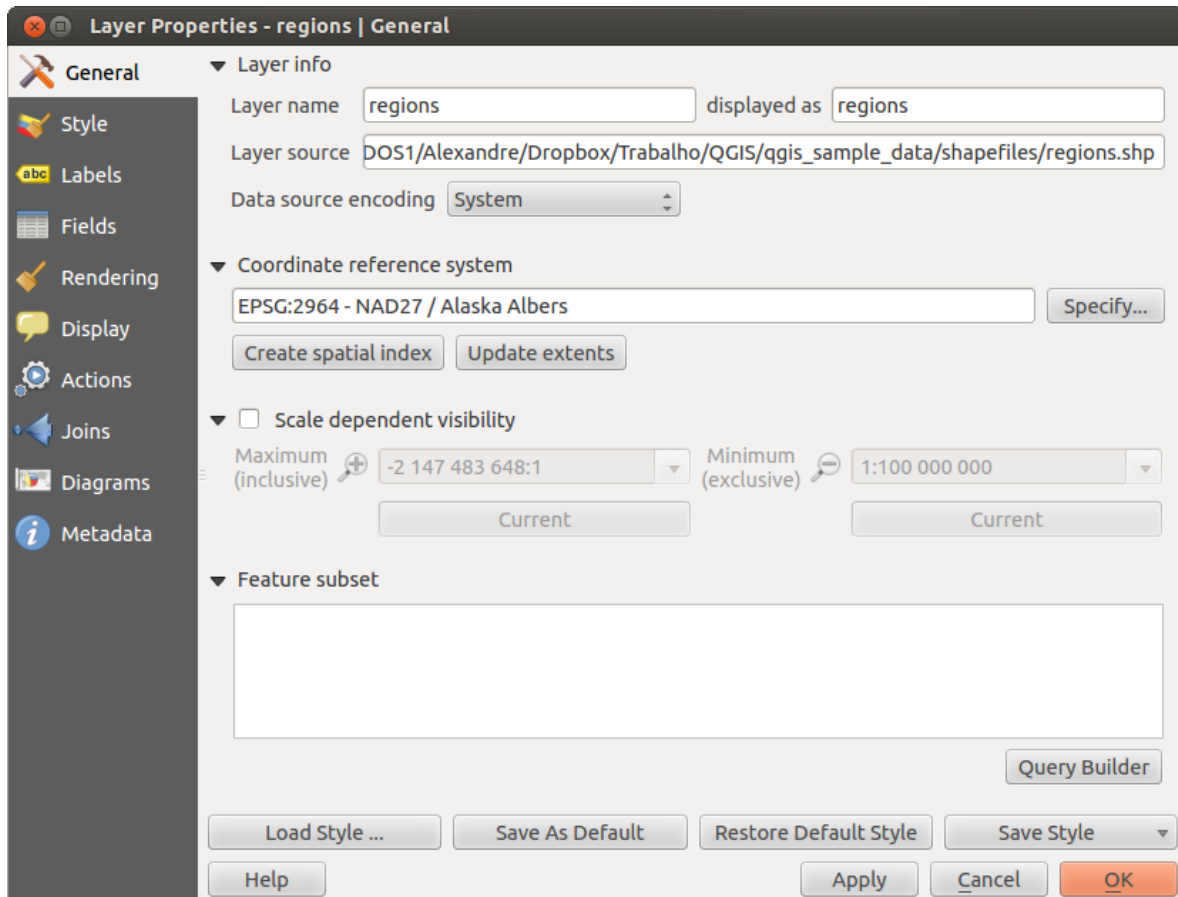




Figura 12.9: Janela das Propriedades da Camada Vectorial 🐧

12.3.1 Estilos

The *Style* menu provides you with a comprehensive tool for rendering and symbolizing your vector data. You can use *Layer rendering* → tools that are common to all vector data, as well as special symbolizing tools that were designed for the different kinds of vector data.

Renderizadores

The renderer is responsible for drawing a feature together with the correct symbol. There are four types of renderers: single symbol, categorized, graduated and rule-based. There is no continuous color renderer, because it is in fact only a special case of the graduated renderer. The categorized and graduated renderers can be created by specifying a symbol and a color ramp - they will set the colors for symbols appropriately. For point layers, there is a point displacement renderer available. For each data type (points, lines and polygons), vector symbol layer types are available. Depending on the chosen renderer, the *Style* menu provides different additional sections. On the bottom right of the symbology dialog, there is a **[Symbol]** button, which gives access to the Style Manager (see *Presentation*). The Style Manager allows you to edit and remove existing symbols and add new ones.

After having made any needed changes, the symbol can be added to the list of current style symbols (using [Symbol]  *Save in symbol library*), and then it can easily be used in the future. Furthermore, you can use the [Save Style]  button to save the symbol as a QGIS layer style file (.qml) or SLD file (.sld). SLDs can be exported from any type of renderer – single symbol, categorized, graduated or rule-based – but when importing an SLD, either a single symbol or rule-based renderer is created. That means that categorized or graduated styles are converted to rule-based. If you want to preserve those renderers, you have to stick to the QML format. On the other hand, it can be very handy sometimes to have this easy way of converting styles to rule-based.

If you change the renderer type when setting the style of a vector layer the settings you made for the symbol will be maintained. Be aware that this procedure only works for one change. If you repeat changing the renderer type the settings for the symbol will get lost.

If the datasource of the layer is a database (PostGIS or Spatialite for example), you can save your layer style inside a table of the database. Just clic on :guilabel:‘ Save Style‘ comboxbox and choose **Save in database** item then fill in the dialog to define a style name, add a description, an ui file and if the style is a default style. When loading a layer from the database, if a style already exists for this layer, QGIS will load the layer and its style. You can add several style in the database. Only one will be the default style anyway.

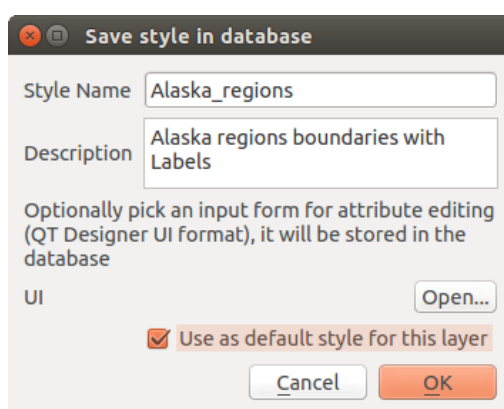


Figura 12.10: Save Style in database Dialog 

Tip: Selecionar e alterar múltiplos símbolos


A simbologia permite selecionar múltiplos símbolos e com o clique do direito do rato alterar a cor, trnsparência, tamanho, ou largura das entradas seleccionadas.

Renderização por Símbolo Único

The Single Symbol Renderer is used to render all features of the layer using a single user-defined symbol. The properties, which can be adjusted in the *Style* menu, depend partially on the type of layer, but all types share the following dialog structure. In the top-left part of the menu, there is a preview of the current symbol to be rendered. On the right part of the menu, there is a list of symbols already defined for the current style, prepared to be used by selecting them from the list. The current symbol can be modified using the menu on the right side. If you click on the first level in the *Symbol layers* dialog on the left side, it's possible to define basic parameters like *Size*, *Transparency*, *color* and *Rotation*. Here, the layers are joined together.

Renderizar por Categorias

The Categorized Renderer is used to render all features from a layer, using a single user-defined symbol whose color reflects the value of a selected feature's attribute. The *Style* menu allows you to select:

- The attribute (using the Column listbox or the  *Set column expression* function, see [Expressions](#))
- O símbolo (usando a janela de diálogo da Simbologia)
- The colors (using the color Ramp listbox)

Then click on **Classify** button to create classes from the distinct value of the attribute column. Each classes can be disabled unchecking the checkbox at the left of the class name.

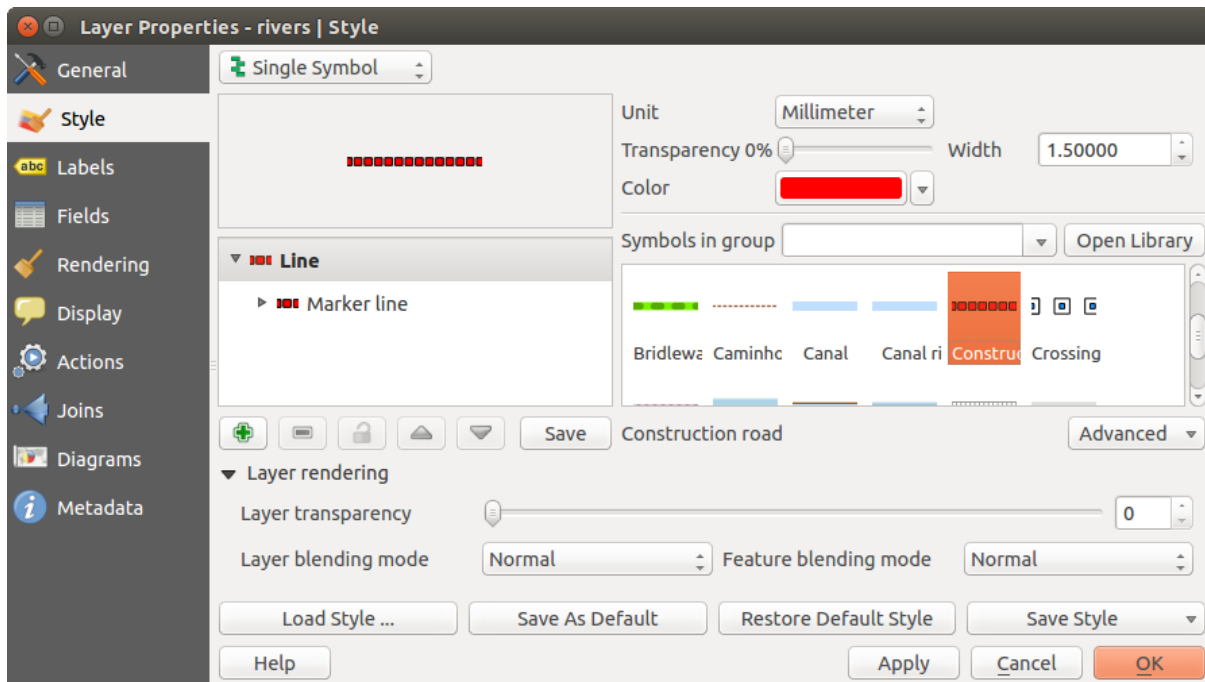


Figura 12.11: Propriedade da linha do símbolo único 🐧

You can change symbol, value and/or label of the clic, just double clicking on the item you want to change.

Right-clic shows a contextual menu to **Copy/Paste**, **Change color**, **Change transparency**, **Change output unit**, **Change symbol width**.

The [Advanced] button in the lower-right corner of the dialog allows you to set the fields containing rotation and size scale information. For convenience, the center of the menu lists the values of all currently selected attributes together, including the symbols that will be rendered.

The example in [figure_symbology_2](#) shows the category rendering dialog used for the rivers layer of the QGIS sample dataset.

Renderizador Graduado

The Graduated Renderer is used to render all the features from a layer, using a single user-defined symbol whose color reflects the assignment of a selected feature's attribute to a class.

Like the Categorized Renderer, the Graduated Renderer allows you to define rotation and size scale from specified columns.

Also, analogous to the Categorized Renderer, the *Style* tab allows you to select:

- The attribute (using the Column listbox or the ϵ ... *Set column expression* function, see [Expressions](#) chapter)
- O símbolo (usando o botão das Propriedades do Símbolo)
- The colors (using the color Ramp list)

Additionally, you can specify the number of classes and also the mode for classifying features within the classes (using the Mode list). The available modes are:

- Equal Interval: each class has the same size (e.g. values from 0 to 16 and 4 classes, each class has a size of 4);
- Quantile: each class will have the same number of element inside (the idea of a boxplot);
- Natural Breaks (Jenks): the variance within each class is minimal while the variance between classes is maximal;
- Standard Deviation: classes are built depending on the standard deviation of the values;

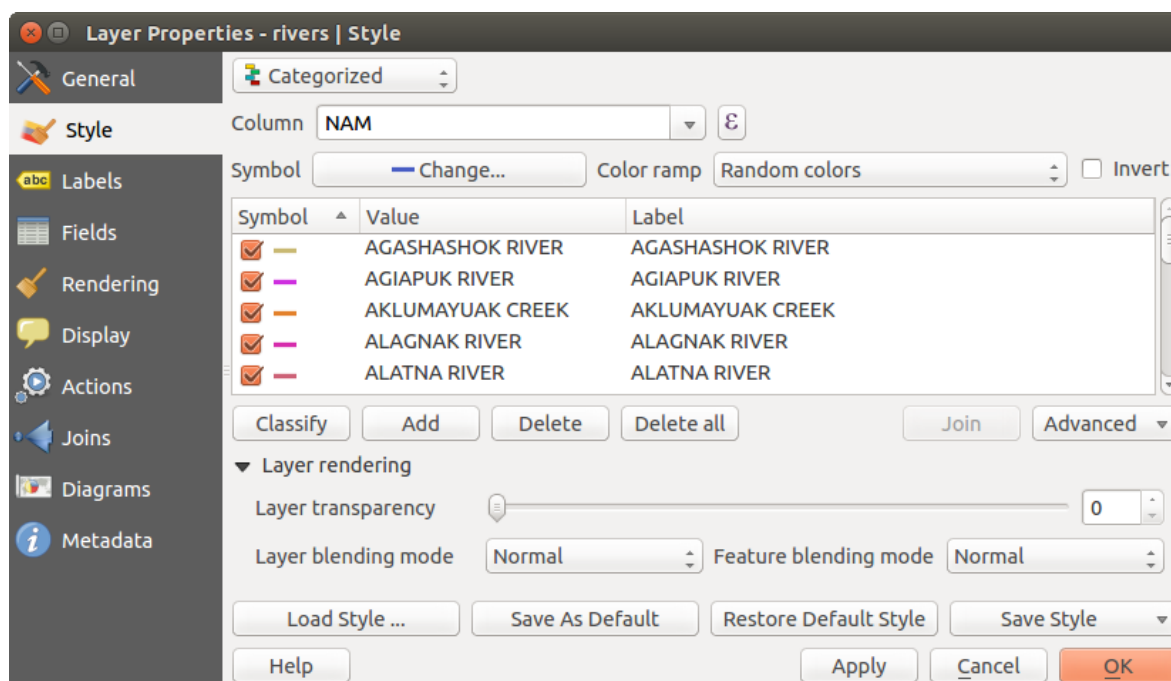


Figura 12.12: Opções de Simbolização de categorias 🐧

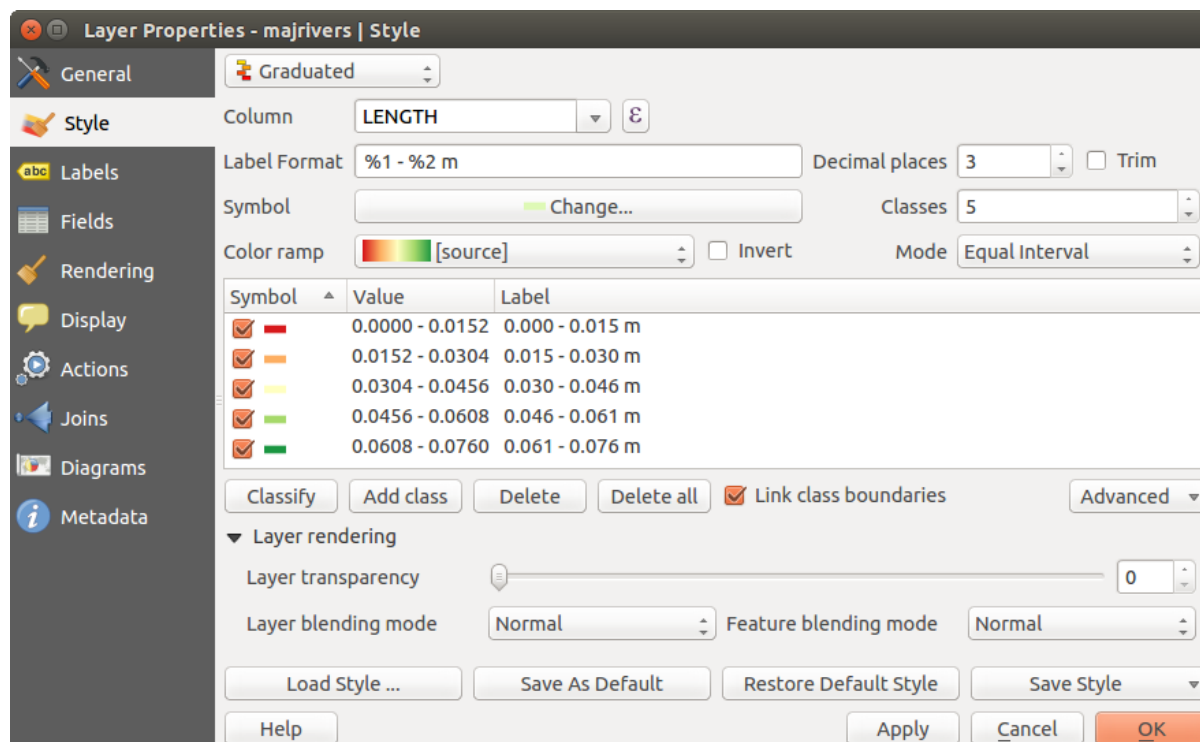


Figura 12.13: Opções da Simbolização Graduada 🐧

- Pretty Breaks: the same of natural breaks but the extremes number of each class are integers.

A caixa da lista no centro do menu *Estilo* lista as classes juntas com os seus intervalos, etiquetas e símbolos que serão renderizados.

Click on **Classify** button to create classes using the choosen mode. Each classes can be disabled unchecking the checkbox at the left of the class name.

You can change symbol, value and/or label of the clic, just double clicking on the item you want to change.

Right-clic shows a contextual menu to **Copy/Paste**, **Change color**, **Change transparency**, **Change output unit**, **Change symbol width**.

The example in [figure_symbology_4](#) shows the graduated rendering dialog for the rivers layer of the QGIS sample dataset.


Tip: Mapas temáticos usando uma expressão

Categorized and graduated thematic maps can now be created using the result of an expression. In the properties dialog for vector layers, the attribute chooser has been augmented with a \mathcal{E} ... *Set column expression* function. So now you no longer need to write the classification attribute to a new column in your attribute table if you want the classification attribute to be a composite of multiple fields, or a formula of some sort.

Renderização baseada em regras

The Rule-based Renderer is used to render all the features from a layer, using rule based symbols whose color reflects the assignment of a selected feature’s attribute to a class. The rules are based on SQL statements. The dialog allows rule grouping by filter or scale, and you can decide if you want to enable symbol levels or use only the first-matched rule.

The example in [figure_symbology_5](#) shows the rule-based rendering dialog for the rivers layer of the QGIS sample dataset.

To create a rule, activate an existing row by double-clicking on it, or click on ‘+’ and click on the new rule. In the *Rule properties* dialog, you can define a label for the rule. Press the  button to open the expression string builder. In the **Function List**, click on *Fields and Values* to view all attributes of the attribute table to be searched. To add an attribute to the field calculator **Expression** field, double click its name in the *Fields and Values* list. Generally, you can use the various fields, values and functions to construct the calculation expression, or you can just type it into the box (see *Expressions*). You can create a new rule by copying and pasting an existing rule with the right mouse button. You can also use the ‘ELSE’ rule that will be run if none of the other rules on that level match. Since QGIS 2.6 the label for the rules appears in a pseudotree in the map legend. Just double-klick the rules in the map legend and the Style menu of the layer properties appears showing the rule that is the background for the symbol in the pseudotree.

Deslocador de pontos

The Point Displacement Renderer works to visualize all features of a point layer, even if they have the same location. To do this, the symbols of the points are placed on a displacement circle around a center symbol.

Tip: Exportar simbologia do vector

You have the option to export vector symbology from QGIS into Google *.kml, *.dxf and MapInfo *.tab files. Just open the right mouse menu of the layer and click on *Save selection as* → to specify the name of the output file and its format. In the dialog, use the *Symbology export* menu to save the symbology either as *Feature symbology* → or as *Symbol layer symbology* →. If you have used symbol layers, it is recommended to use the second setting.

Inverted Polygon

Inverted polygon renderer allows user to define a symbol to fill in outside of the layer’s polygons. As before you can select a subrenderers. These subrenderers are the same as for the main renderers.

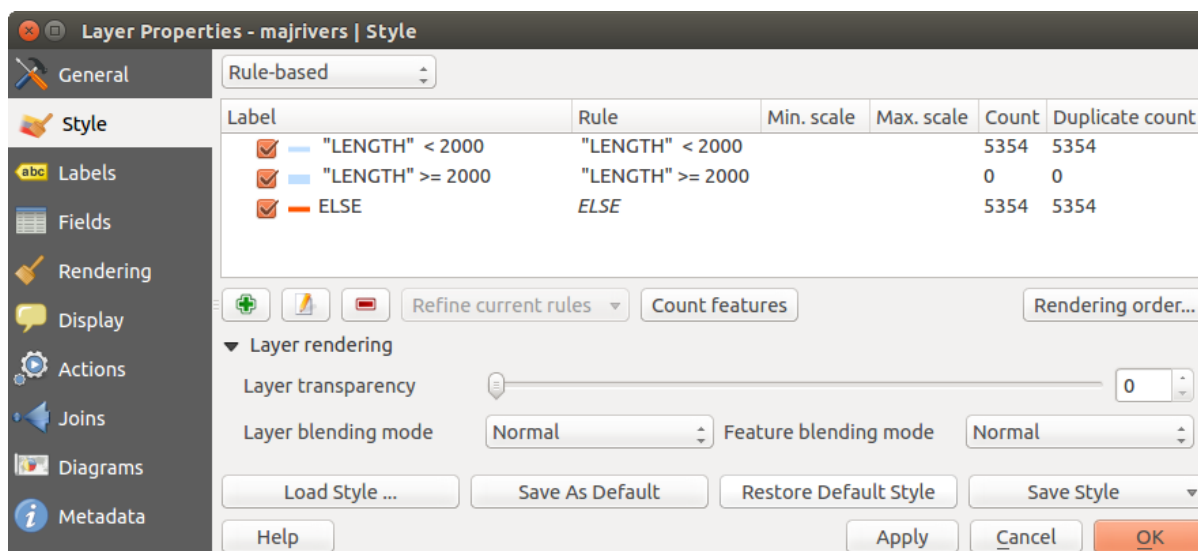


Figura 12.14: Opções de Simbolização Baseadas em Regras

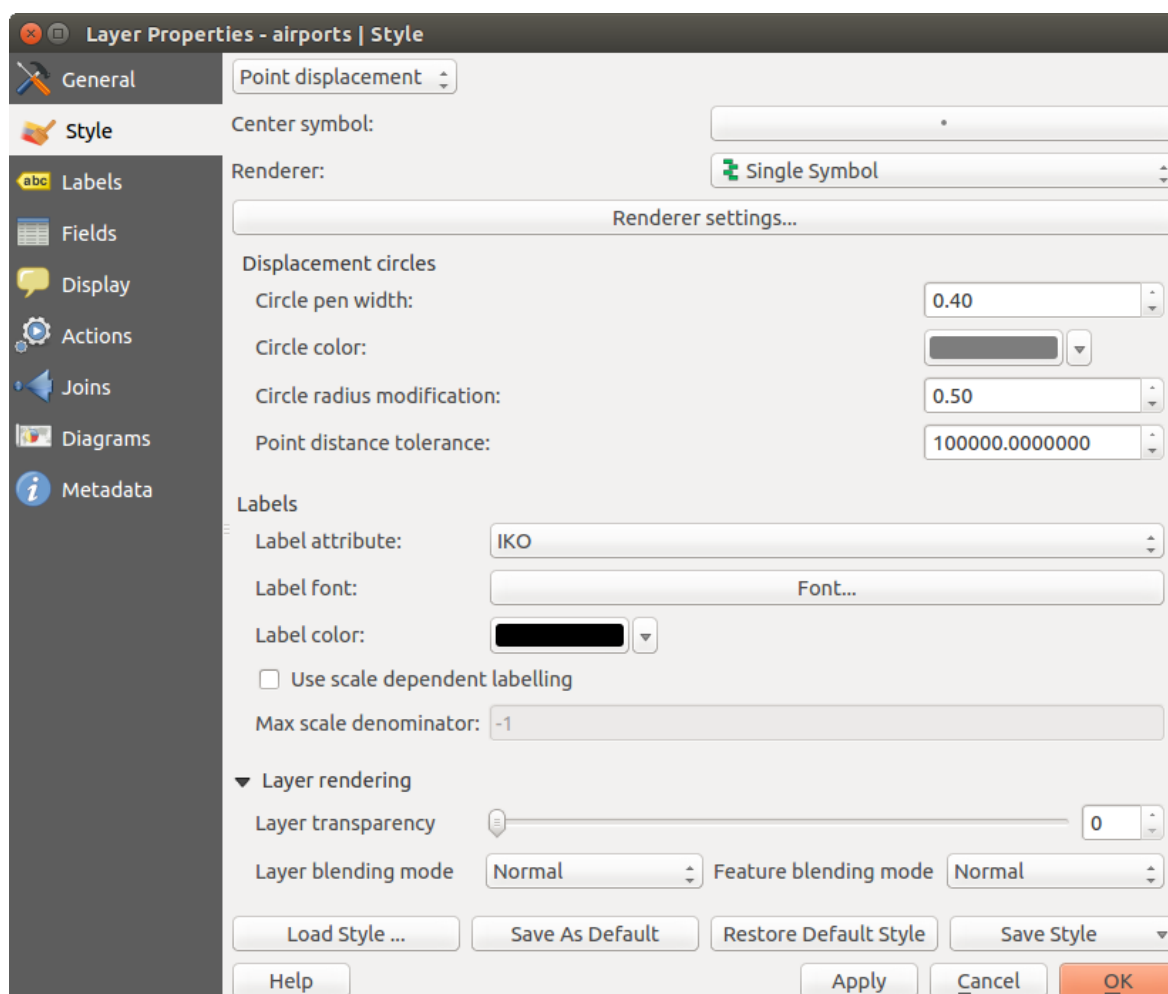


Figura 12.15: Janela do Deslocador de pontos

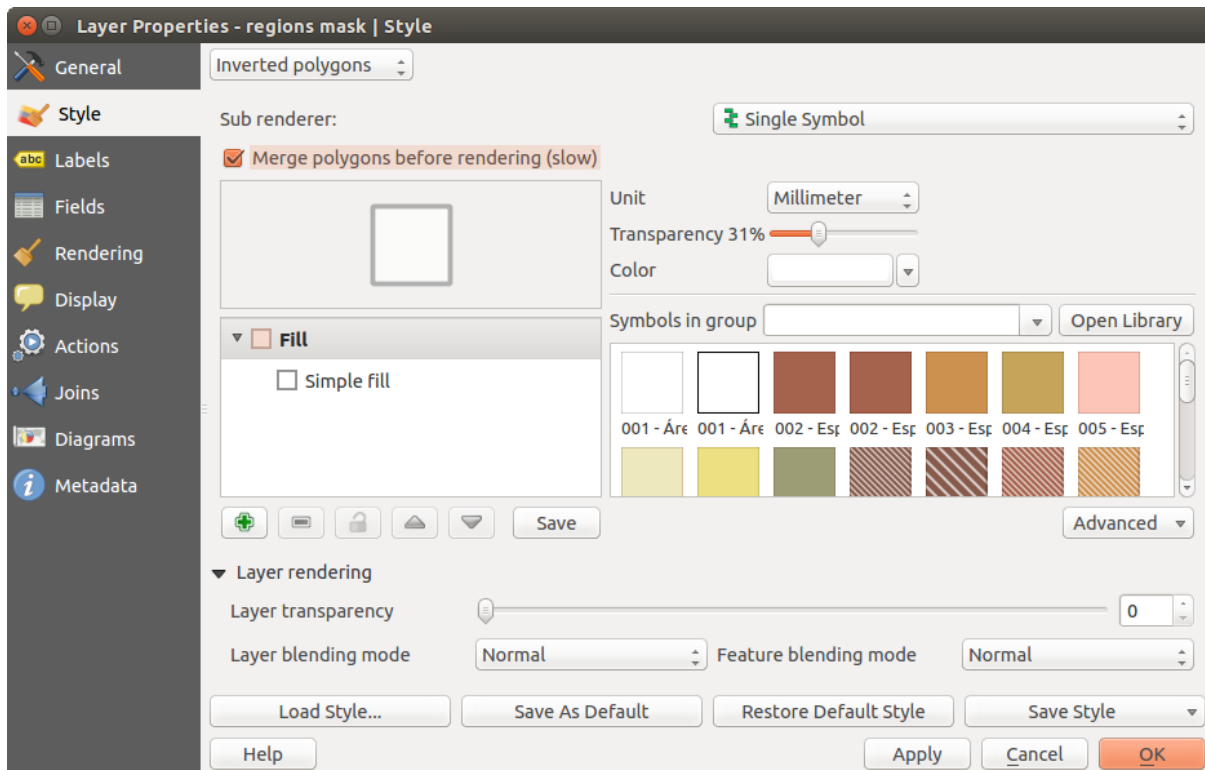










Figura 12.16: Inverted Polygon dialog 

Color Picker

Regardless the type of style to be used, the *select color* dialog will show when you click to choose a color - either border or fill color. This dialog has four different tabs which allow you to select colors by  color ramp,  color wheel,  color swatches or  color picker.

Whatever method you use, the selected color is always described through color sliders for HSV (Hue, Saturation, Value) and RGB (Red, Green, Blue) values. There is also an *opacity* slider to set transparency level. On the lower left part of the dialog you can see a comparison between the *current* and the *new* color you are presently selecting and on the lower right part you have the option to add the color you just tweaked into a color slot button.


With  color ramp or with  color wheel, you can browse to all possible color combinations. There are other possibilities though. By using *color swatches*  you can choose from a preselected list. This selected list is populated with one of three methods: *Recent colors*, *Standard colors* or *Project colors*

Another option is to use the  color picker which allows you to sample a color from under your mouse pointer at any part of QGIS or even from another application by pressing the space bar. Please note that the color picker is OS dependent and is currently not supported by OSX.

Tip: quick color picker + copy/paste colors

You can quickly choose from *Recent colors*, from *Standard colors* or simply *copy* or *paste* a color by clicking the drop-down arrow that follows a current color box.

Renderização da camada

- *Layer transparency* : You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your vector layer to your needs. You can also make a

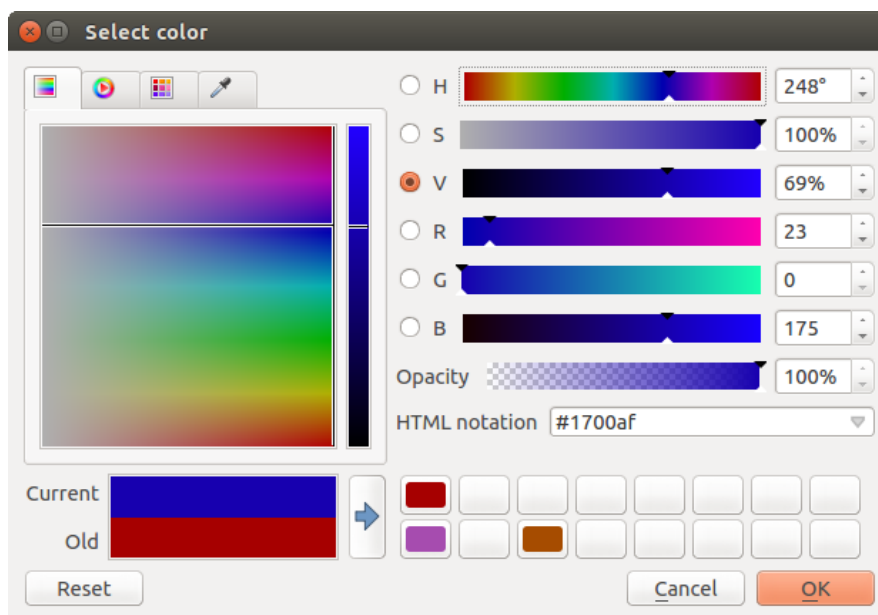


Figura 12.17: Color picker ramp tab

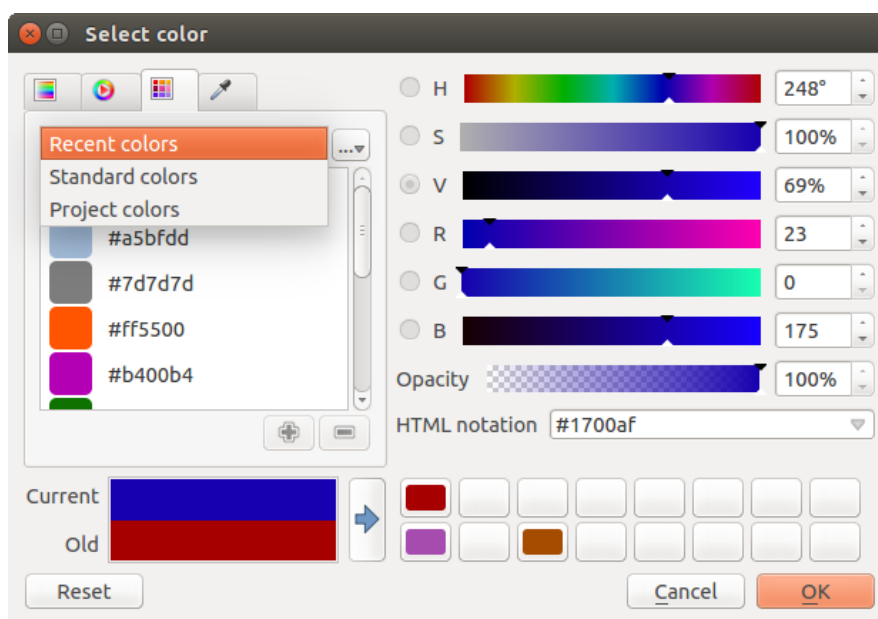


Figura 12.18: Color picker swatcher tab

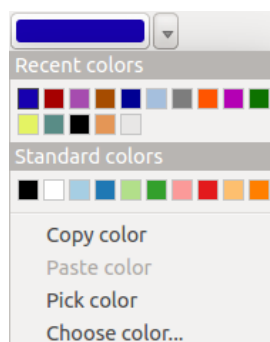



Figura 12.19: Quick color picker menu

precise definition of the percentage of visibility in the the menu beside the slider.

- *Modo de mistura e Modo de mistura do Elemento*: Pode alcançar efeitos especiais de renderização com estas ferramentas que anteriormente só era conhecido de programas gráficos. Os pixels dos itens sobrejacente e subjacente são misturados através das configurações descritas em baixo.
 - Normal: Este é o modo padrão de renderização que usa o canal alfa do pixel superior para renderizar com o pixel abaixo dele; as cores não estão misturadas.
 - Mais claro: Selecciona o máximo de cada componente dos pixels de primeiro plano e de fundo. Tenha atenção que os resultados tendem a ser irregulares e “áspero”.
 - Ecrã: Os pixels claros da fonte são pintados sobre o destino e os pixels escuros não são. Este modo é muito útil para misturar texturas de uma camada com outra camada. (Ex.: pode usar o mapa de sombras como textura noutra camada).
 - Subexposição: A subexposição irá clarear e saturar os pixels subjacentes baseados na luminosidade do pixel superior. A maior claridade do pixel superior causa o aumento da saturação e brilho dos pixels subjacentes. Isto funciona melhor nos pixels superiores que não brilham muito, caso contrário o efeito é muito extremo.
 - Adição: Este modo de renderização simplesmente adiciona os valores dos pixels de uma camada noutra. Nos casos que os valores são acima de 1 (no caso do RGB), o branco é exibido. Este modo é adequado para destacar elementos.
 - Escurecido: Cria um pixel resultante que retêm os componentes mais pequenos dos pixels do primeiro plano e do fundo. Como o mais claro, o resultado tende a ser irregular e “áspero”.
 - Multiplicar: Multiplica o número para cada pixel superior da camada com o pixel correspondente da camada abaixo. Os resultados são imagens mais escuras.
 - Queimar: As cores escuras da camada superior torna mais escuro as camadas subjacentes. Pode ser usado para ajustar e colorizar camadas subjacentes.
 - Sobreposição: É uma combinação entre os modos de renderização de multiplicar e ecrã. Como resultado as partes claras da imagem tornam-se mais claras e as partes escuras ficam mais escuras.
 - Pouca luz: Muito semelhante à sobreposição, mas em vez de usar multiplicar/ecrã usa o queimar/subexposição. Neste modo é suposto imitar brilhar uma luz suave em uma imagem.
 - Muita luz: Este modo é muito semelhante ao modo de sobreposição. É suposto simular a projecção de uma luz muito intensa numa imagem.
 - Diferença: A diferença subtrair o pixel superior com pixel de baixo e vice-versa, para obter sempre o valor positivo. A mistura com pretos não produz alteração, como valor todas as cores são zero.
 - Subtracção: Este modo de renderização simplesmente subtrair os valores do pixel de uma camada à outra. Em caso de valores negativos, o preto é exibido.


12.3.2 Menu Etiquetas


The  Labels core application provides smart labeling for vector point, line and polygon layers, and it only requires a few parameters. This new application also supports on-the-fly transformed layers. The core functions of the application have been redesigned. In QGIS, there are a number of other features that improve the labeling. The following menus have been created for labeling the vector layers:

- Texto
- Formatação
- Buffer
- Fundo
- Sombra
- Posicionamento

- Renderização

Vamos ver como os novos menus podem ser usados para várias camadas vectoriais. **Labeling point layers**

Inicie o QGIS e carregue uma camada vectorial de pontos. Active a camada na legenda e clique no ícone  Opções de Rotulagem da Camada no menu da barra de ferramentas do QGIS.

The first step is to activate the *Label this layer with* checkbox and select an attribute column to use for labeling. Click  if you want to define labels based on expressions - See [labeling_with_expressions](#).

The following steps describe a simple labeling without using the *Data defined override* functions, which are situated next to the drop-down menus.

You can define the text style in the *Text* menu (see [Figure_labels_1](#)). Use the *Type case* option to influence the text rendering. You have the possibility to render the text 'All uppercase', 'All lowercase' or 'Capitalize first letter'. Use the blend modes to create effects known from graphics programs (see [blend_modes](#)).


In the *Formatting* menu, you can define a character for a line break in the labels with the 'Wrap on character' function. Use the *Formatted numbers* option to format the numbers in an attribute table. Here, decimal places may be inserted. If you enable this option, three decimal places are initially set by default.

To create a buffer, just activate the *Draw text buffer* checkbox in the *Buffer* menu. The buffer color is variable. Here, you can also use blend modes (see [blend_modes](#)).

If the *color buffer's fill* checkbox is activated, it will interact with partially transparent text and give mixed color transparency results. Turning off the buffer fill fixes that issue (except where the interior aspect of the buffer's stroke intersects with the text's fill) and also allows you to make outlined text.


In the *Background* menu, you can define with *Size X* and *Size Y* the shape of your background. Use *Size type* to insert an additional 'Buffer' into your background. The buffer size is set by default here. The background then consists of the buffer plus the background in *Size X* and *Size Y*. You can set a *Rotation* where you can choose between 'Sync with label', 'Offset of label' and 'Fixed'. Using 'Offset of label' and 'Fixed', you can rotate the background. Define an *Offset X,Y* with *X* and *Y* values, and the background will be shifted. When applying *Radius X,Y*, the background gets rounded corners. Again, it is possible to mix the background with the underlying layers in the map canvas using the *Blend mode* (see [blend_modes](#)).

Use the *Shadow* menu for a user-defined *Drop shadow*. The drawing of the background is very variable. Choose between 'Lowest label component', 'Text', 'Buffer' and 'Background'. The *Offset* angle depends on the orientation of the label. If you choose the *Use global shadow* checkbox, then the zero point of the angle is always oriented to the north and doesn't depend on the orientation of the label. You can influence the appearance of the shadow with the *Blur radius*. The higher the number, the softer the shadows. The appearance of the drop shadow can also be altered by choosing a blend mode (see [blend_modes](#)).

Choose the *Placement* menu for the label placement and the labeling priority. Using the  *Offset from point* setting, you now have the option to use *Quadrants* to place your label. Additionally, you can alter the angle of the label placement with the *Rotation* setting. Thus, a placement in a certain quadrant with a certain rotation is possible.

In the *Rendering* menu, you can define label and feature options. Under *Label options*, you find the scale-based visibility setting now. You can prevent QGIS from rendering only selected labels with the *Show all labels for this layer (including colliding labels)* checkbox. Under *Feature options*, you can define whether every part of a multipart feature is to be labeled. It's possible to define whether the number of features to be labeled is limited and to *Discourage labels from covering features*.

Labeling line layers

The first step is to activate the *Label this layer* checkbox in the *Label settings* tab and select an attribute column to use for labeling. Click  if you want to define labels based on expressions - See [labeling_with_expressions](#).

After that, you can define the text style in the *Text* menu. Here, you can use the same settings as for point layers.

Also, in the *Formatting* menu, the same settings as for point layers are possible.

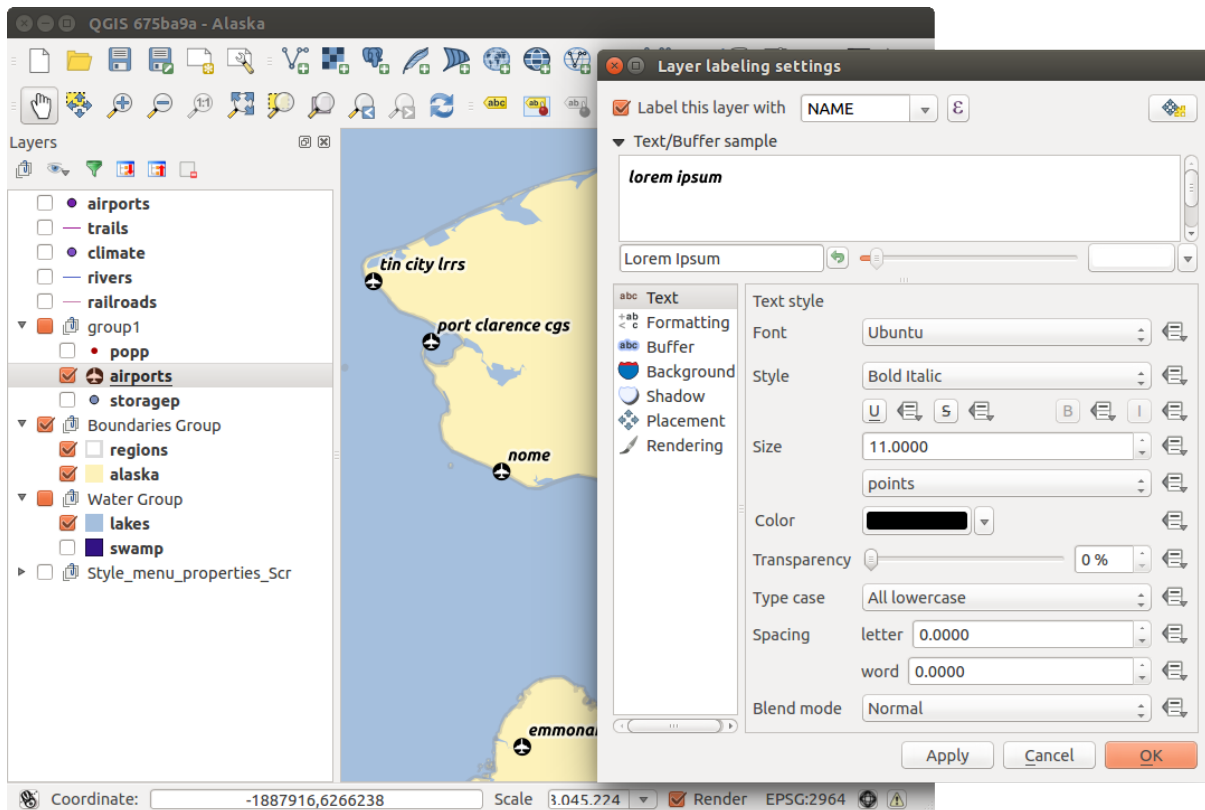



Figura 12.20: Smart labeling of vector point layers 

The *Buffer* menu has the same functions as described in section [labeling_point_layers](#).

The *Background* menu has the same entries as described in section [labeling_point_layers](#).

Also, the *Shadow* menu has the same entries as described in section [labeling_point_layers](#).


In the *Placement* menu, you find special settings for line layers. The label can be placed *Parallel*, *Curved* or *Horizontal*. With the *Parallel* and *Curved* option, you can define the position *Above line*, *On line* and *Below line*. It's possible to select several options at once. In that case, QGIS will look for the optimal position of the label. Remember that here you can also use the line orientation for the position of the label. Additionally, you can define a *Maximum angle between curved characters* when selecting the *Curved* option (see [Figure_labels_2](#)).

You can set up a minimum distance for repeating labels. Distance can be in mm or in map units.

Some Placement setup will display more options, for example, *Curved* and *Parallel* Placements will allow the user to set up the position of the label (above, below or on the line), *distance* from the line and for *Curved*, the user can also setup inside/outside max angle between curved label.

The *Rendering* menu has nearly the same entries as for point layers. In the *Feature options*, you can now *Suppress labeling of features smaller than*.

Labeling polygon layers

The first step is to activate the *Label this layer* checkbox and select an attribute column to use for labeling. Click  if you want to define labels based on expressions - See [labeling_with_expressions](#).

In the *Text* menu, define the text style. The entries are the same as for point and line layers.

The *Formatting* menu allows you to format multiple lines, also similar to the cases of point and line layers.

As with point and line layers, you can create a text buffer in the *Buffer* menu.

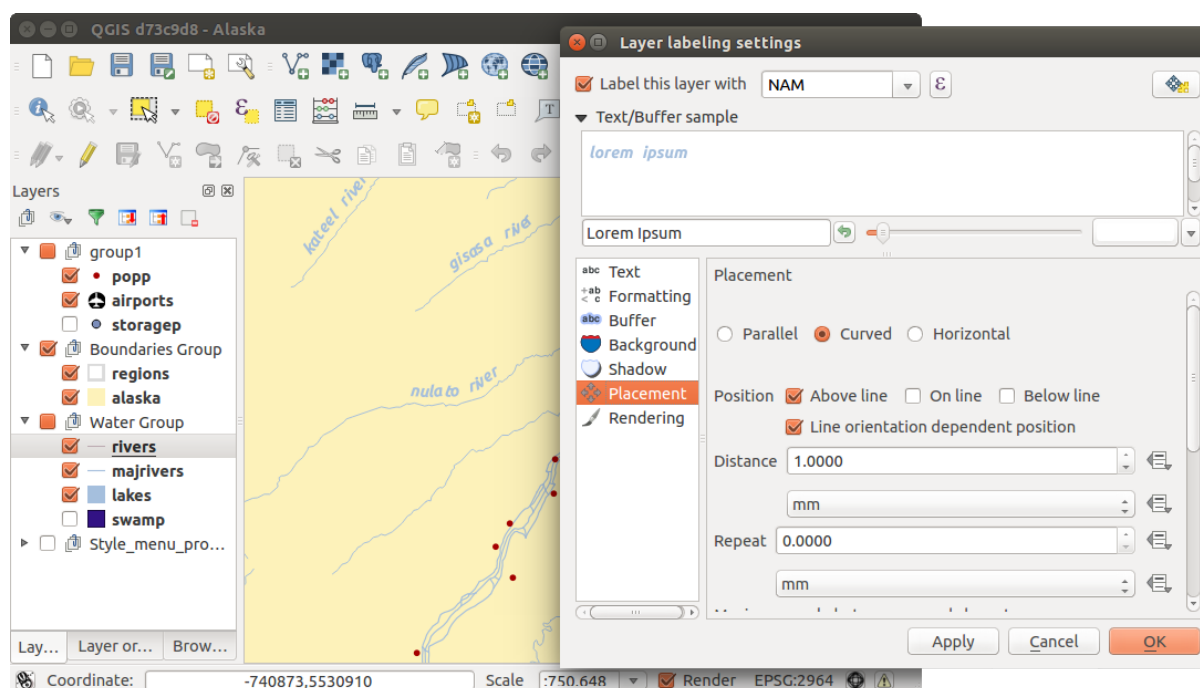



Figura 12.21: Smart labeling of vector line layers 

Use the *Background* menu to create a complex user-defined background for the polygon layer. You can use the menu also as with the point and line layers.



as entradas no menu *Sombra* são as mesmas para as camadas de pontos e linhas.

In the *Placement* menu, you find special settings for polygon layers (see [Figure_labels_3](#)). *Offset from centroid*, *Horizontal (slow)*, *Around centroid*, *Free* and *Using perimeter* are possible.

In the *Offset from centroid* settings, you can specify if the centroid is of the *visible polygon* or *whole polygon*. That means that either the centroid is used for the polygon you can see on the map or the centroid is determined for the whole polygon, no matter if you can see the whole feature on the map. You can place your label with the quadrants here, and define offset and rotation. The *Around centroid* setting makes it possible to place the label around the centroid with a certain distance. Again, you can define *visible polygon* or *whole polygon* for the centroid. With the *Using perimeter* settings, you can define a position and a distance for the label. For the position, *Above line*, *On line*, *Below line* and *Line orientation dependent position* are possible.

Related to the choose of Label Placement, several options will appear. As for Point Placement you can choose the distance for the polygon outline, repeat the label around the polygon perimeter.

The entries in the *Rendering* menu are the same as for line layers. You can also use *Suppress labeling of features smaller than* in the *Feature options*. **Define labels based on expressions**

QGIS allows to use expressions to label features. Just click the  icon in the  Labels menu of the properties dialog. In [figure_labels_4](#) you see a sample expression to label the alaska regions with name and area size, based on the field 'NAME_2', some descriptive text and the function '\$area()' in combination with 'format_number()' to make it look nicer.

Expression based labeling is easy to work with. All you have to take care of is, that you need to combine all elements (strings, fields and functions) with a string concatenation sign '||' and that fields a written in "double quotes" and strings in 'single quotes'. Let's have a look at some examples:

```
# label based on two fields 'name' and 'place' with a comma as separator
"name" || ', ' || "place"
```

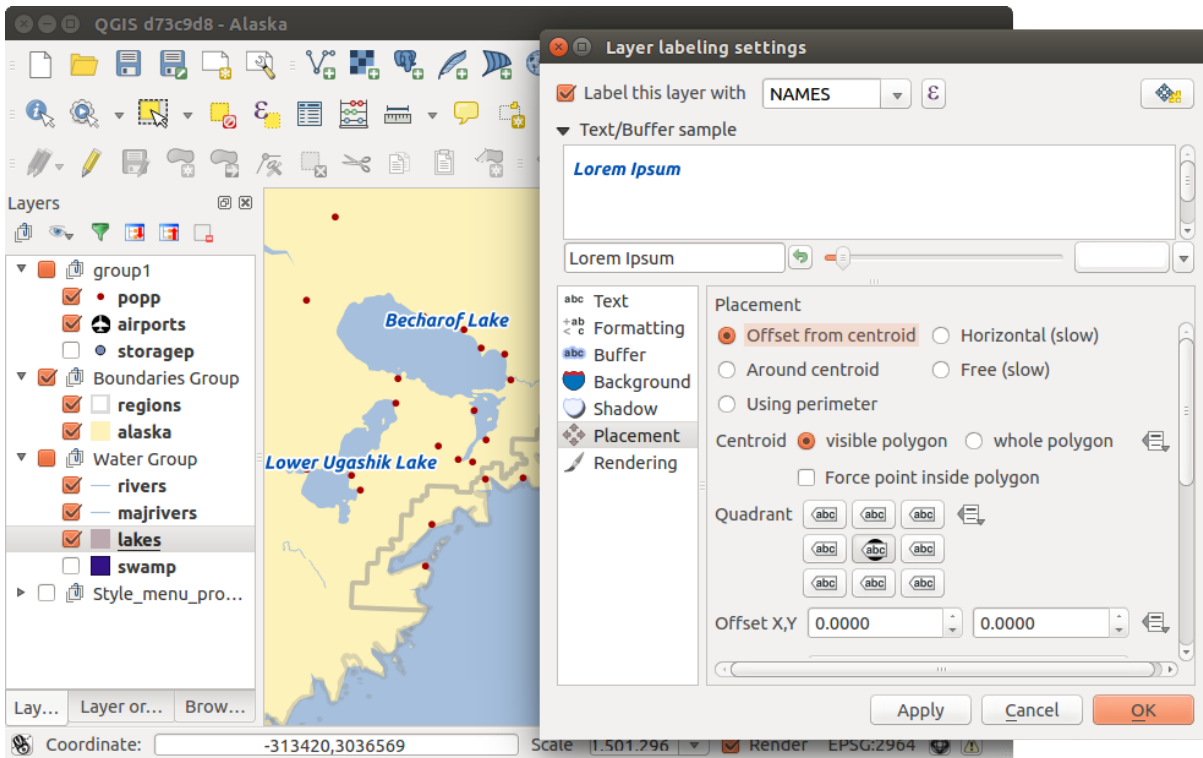


Figura 12.22: Smart labeling of vector polygon layers 🐧

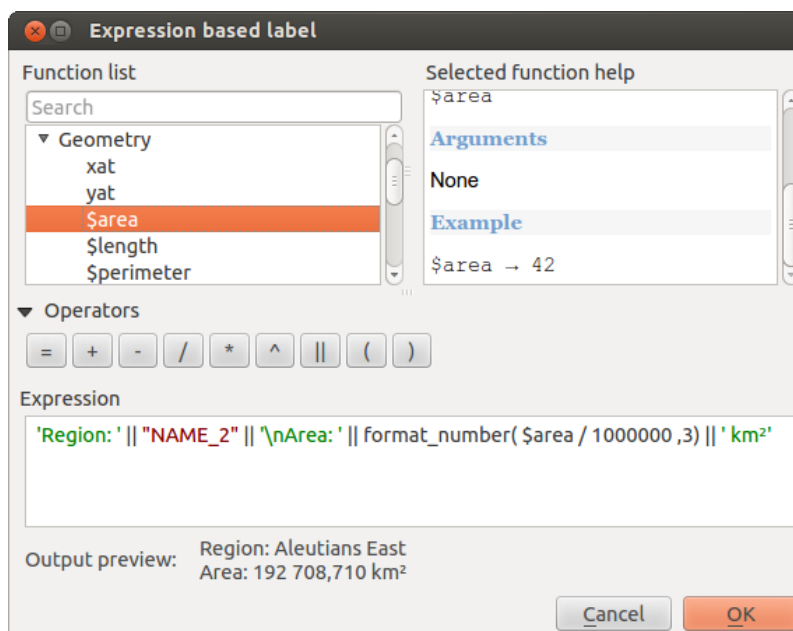


Figura 12.23: Using expressions for labeling 🐧

```

-> John Smith, Paris

# label based on two fields 'name' and 'place' separated by comma
'My name is ' || "name" || 'and I live in ' || "place"

-> My name is John Smith and I live in Paris

# label based on two fields 'name' and 'place' with a descriptive text
# and a line break (\n)
'My name is ' || "name" || '\nI live in ' || "place"

-> My name is John Smith
    I live in Paris

# create a multi-line label based on a field and the $area function
# to show the place name and its area size based on unit meter.
'The area of ' || "place" || 'has a size of ' || $area || 'm²'

-> The area of Paris has a size of 105000000 m²


# create a CASE ELSE condition. If the population value in field
# population is <= 50000 it is a town, otherwise a city.
'This place is a ' || CASE WHEN "population <= 50000" THEN 'town' ELSE 'city' END




-> This place is a town

```





As you can see in the expression builder, you have hundreds of functions available to create simple and very complex expressions to label your data in QGIS. See [Expressions](#) chapter for more information and example on expressions.

Using data-defined override for labeling

With the data-defined override functions, the settings for the labeling are overridden by entries in the attribute table. You can activate and deactivate the function with the right-mouse button. Hover over the symbol and you see the information about the data-defined override, including the current definition field. We now describe an example using the data-defined override function for the  Move label function (see [figure_labels_5](#)).

1. Importe o `lakes.shp` a partir da amostra do conjunto de dados do QGIS.
2. Double-click the layer to open the Layer Properties. Click on *Labels* and *Placement*. Select  *Offset from centroid*.
3. Look for the *Data defined* entries. Click the  icon to define the field type for the *Coordinate*. Choose 'xlabel' for X and 'ylabel' for Y. The icons are now highlighted in yellow.
4. Ampliar ao lago.
5. Go to the Label toolbar and click the  icon. Now you can shift the label manually to another position (see [figure_labels_6](#)). The new position of the label is saved in the 'xlabel' and 'ylabel' columns of the attribute table.

12.3.3 Menu Campos

 Within the *Fields* menu, the field attributes of the selected dataset can be manipulated. The buttons  *New Column* and  *Delete Column* can be used when the dataset is in  *Editing mode*.

Editor Widget

Within the *Fields* menu, you also find an **edit widget** column. This column can be used to define values or a range of values that are allowed to be added to the specific attribute table column. If you click on the **[edit widget]**

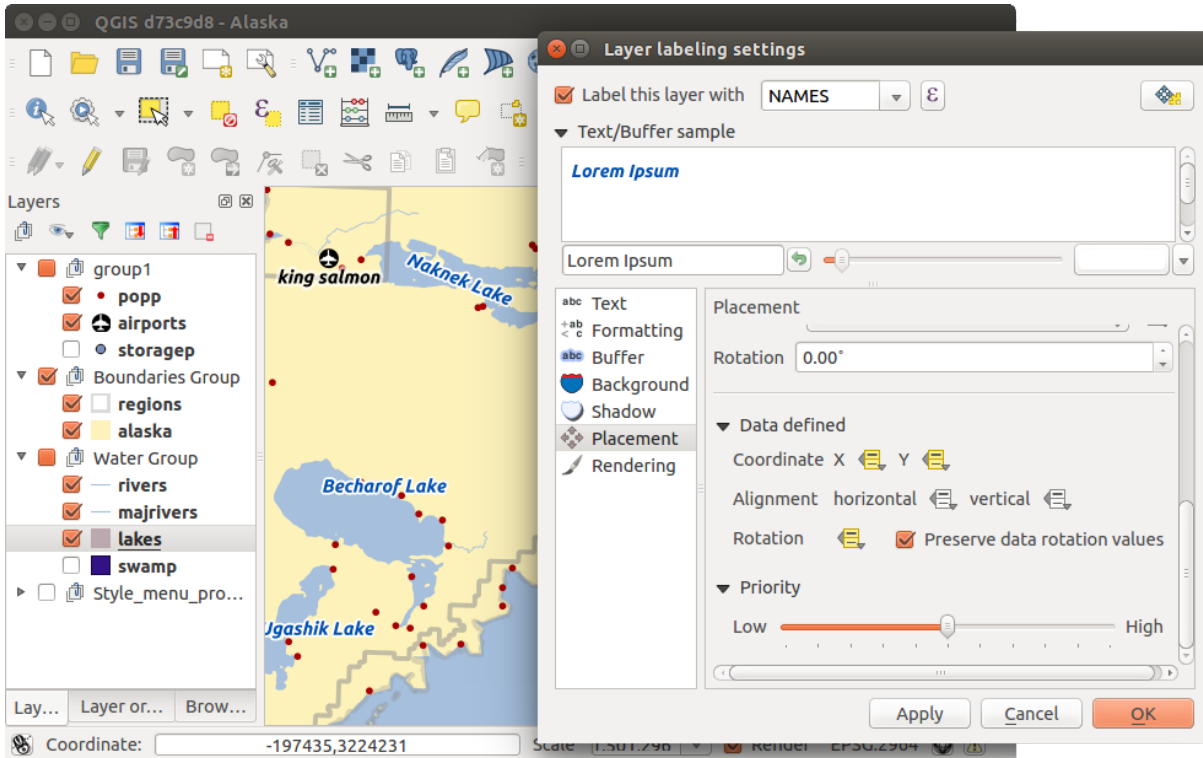


Figura 12.24: Labeling of vector polygon layers with data-defined override 🐧

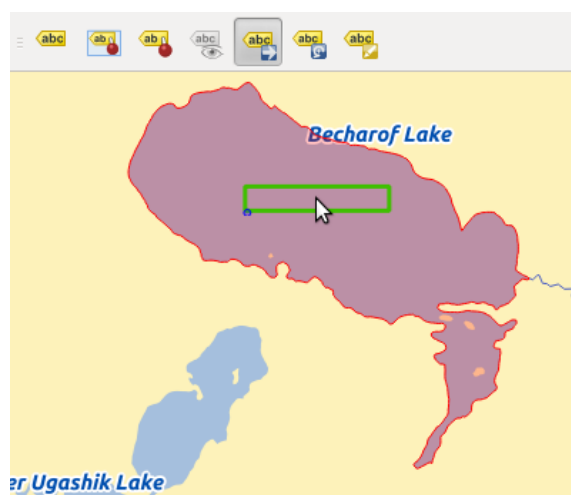


Figura 12.25: Mover etiquetas 🐧

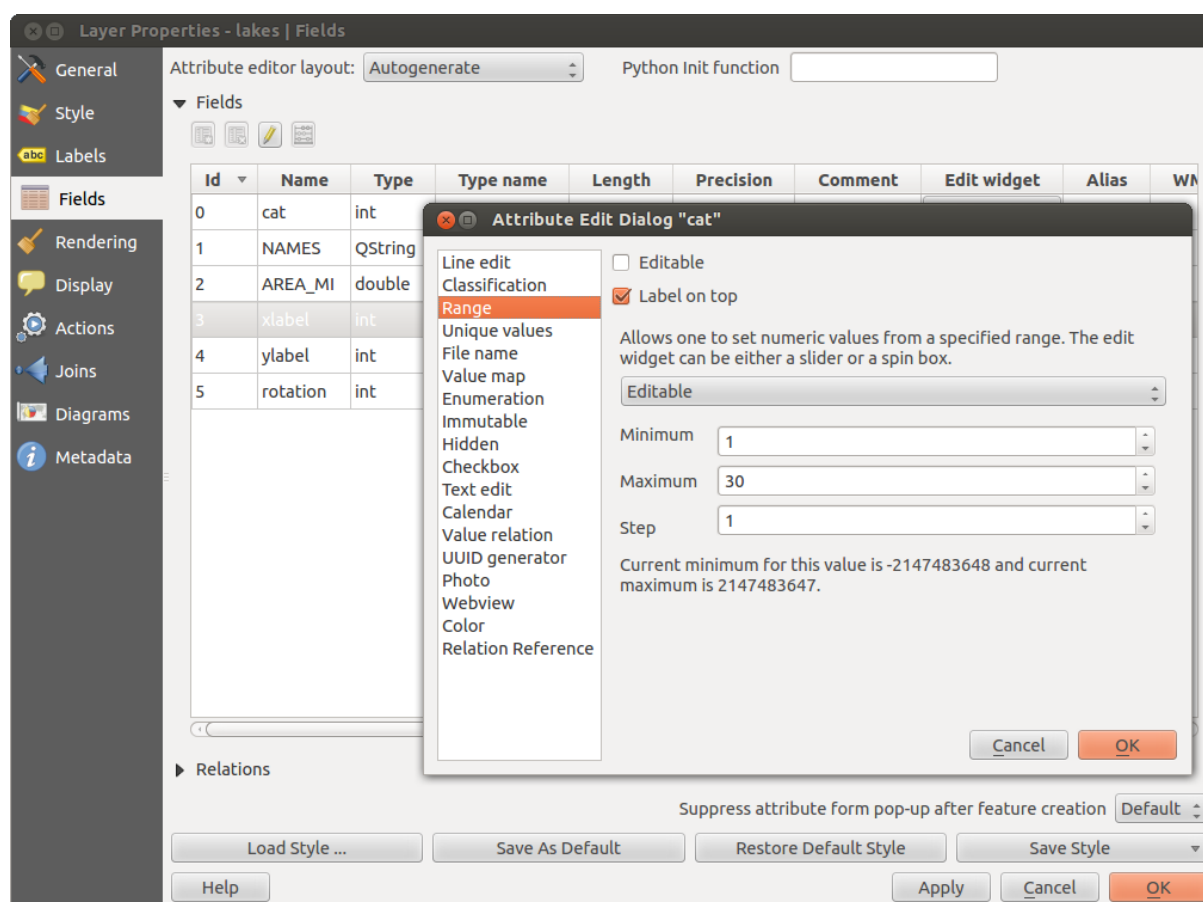




Figura 12.26: Janela de diálogo para seleccionar um widget de edição para um atributo da coluna 🐧

button, a dialog opens, where you can define different widgets. These widgets are:

- **Checkbox:** Displays a checkbox, and you can define what attribute is added to the column when the checkbox is activated or not.
- **Classification:** Displays a combo box with the values used for classification, if you have chosen 'unique value' as legend type in the *Style* menu of the properties dialog.
- **Color:** Displays a color button allowing user to choose a color from the color dialog window.
- **Date/Time:** Displays a line fields which can opens a calendar widget to enter a date, a time or both. Column type must be text. You can select a custom format, pop-up a calendar, etc.
- **Enumeration:** Opens a combo box with values that can be used within the columns type. This is currently only supported by the PostgreSQL provider.
- **Nome do ficheiro:** Simplifica a selecção ao adicionar um diálogo de escolha do ficheiro.
- **Hidden:** A hidden attribute column is invisible. The user is not able to see its contents.
- **Fotografia:** Campo que contem o nome do ficheiro da imagem. A largura e altura do campo podem se definidos.
- **Range:** Allows you to set numeric values from a specific range. The edit widget can be either a slider or a spin box.
- **Relation Reference:** This widget lets you embed the feature form of the referenced layer on the feature form of the actual layer. See *Creating one to many relations*.
- **Text edit (default):** This opens a text edit field that allows simple text or multiple lines to be used. If you choose multiple lines you can also choose html content.
- **Unique values:** You can select one of the values already used in the attribute table. If 'Editable' is activated, a line edit is shown with autocompletion support, otherwise a combo box is used.
- **Gerador UUID:** Gera um campo UUID de leitura (Identificador Único Universal), se estiver vazio.
- **Value map:** A combo box with predefined items. The value is stored in the attribute, the description is shown in the combo box. You can define values manually or load them from a layer or a CSV file.
- **Value Relation:** Offers values from a related table in a combobox. You can select layer, key column and value column.
- **Webview:** Field contains a URL. The width and height of the field is variable.

With the **Attribute editor layout**, you can now define built-in forms for data entry jobs (see [figure_fields_2](#)).

Choose 'Drag and drop designer' and an attribute column. Use the  icon to create a category that will then be shown during the digitizing session (see [figure_fields_3](#)). The next step will be to assign the relevant fields to the category with the  icon. You can create more categories and use the same fields again. When creating a new category, QGIS will insert a new tab for the category in the built-in form.

Other options in the dialog are 'Autogenerate' and 'Provide ui-file'. 'Autogenerate' just creates editors for all fields and tabulates them. The 'Provide ui-file' option allows you to use complex dialogs made with the Qt-Designer. Using a UI-file allows a great deal of freedom in creating a dialog. For detailed information, see <http://nathanw.net/2011/09/05/qgis-tips-custom-feature-forms-with-python-logic/>.

QGIS dialogs can have a Python function that is called when the dialog is opened. Use this function to add extra logic to your dialogs. An example is (in module MyForms.py):

```
def open(dialog, layer, feature) :
    geom = feature.geometry()
    control = dialog.findChild(QWidget, "My line edit")
```

Reference in Python Init Function like so: MyForms.open

MyForms.py must live on PYTHONPATH, in .qgis2/python, or inside the project folder.

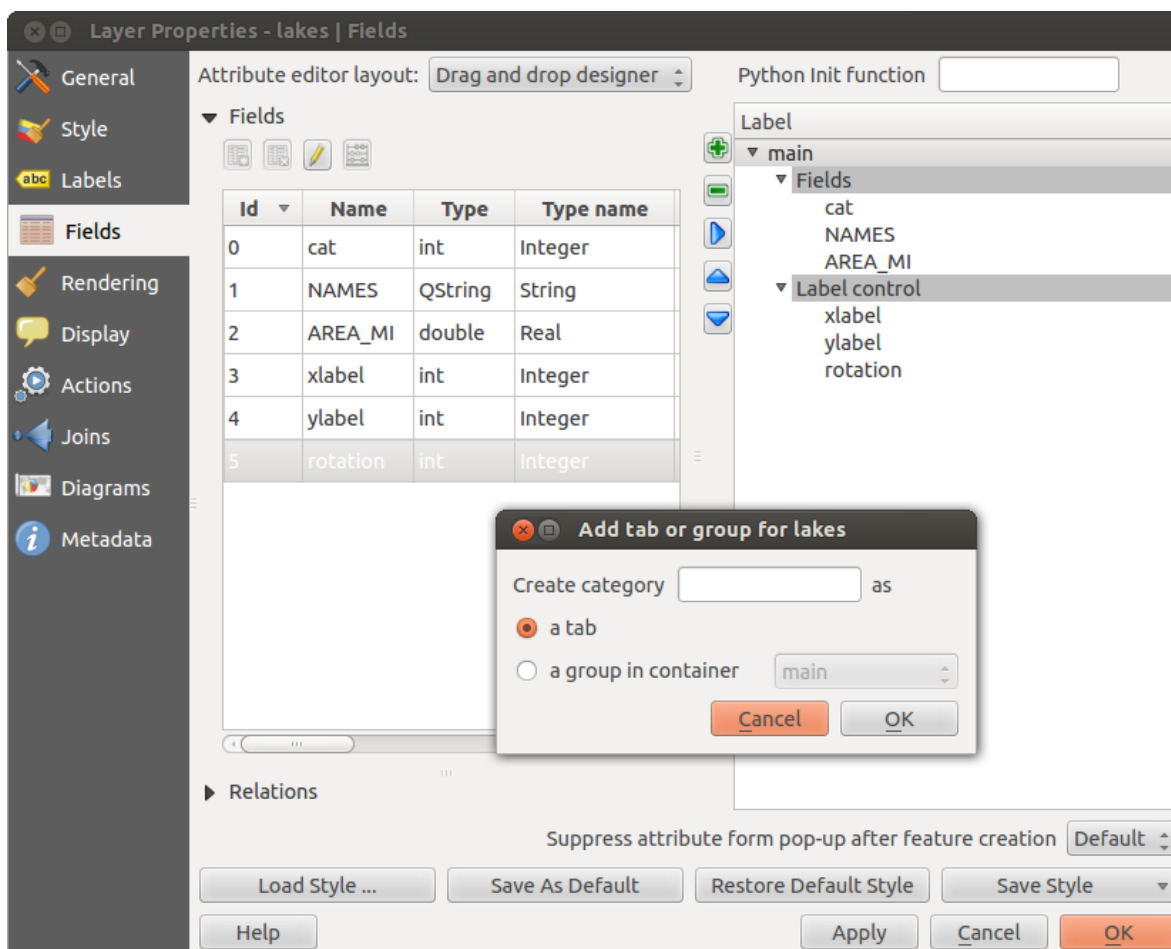


Figura 12.27: Janela de diálogo para criar categorias com o **Layout do editor de atributos**

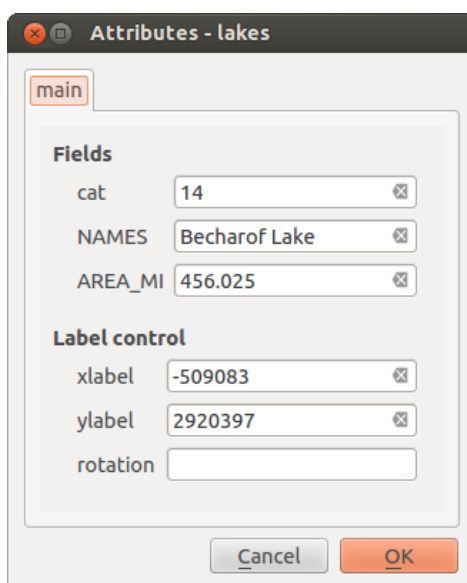


Figura 12.28: Resulting built-in form in a data entry session

12.3.4 Menu Geral



Use this menu to make general settings for the vector layer. There are several options available:

Informação da camada

- Muda o nome de exibição da camada em *exibir como*
- Define a *Fonte da camada* da camada vectorial
- Define the *Data source encoding* to define provider-specific options and to be able to read the file

Sistema de Referência de Coordenadas

- *Specify* the coordinate reference system. Here, you can view or change the projection of the specific vector layer.
- Criar *Índice Espacial* (apenas para formatos OGR suportados)
- *Atualizar Extensões* da camada
- Veja ou altere a projecção de uma camada vectorial específica, clicando em `guilabel:Especificar ...`



Scale dependent visibility

- You can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale. The scale can also be set by the **[Current]** buttons.

Subconjunto de elementos

- With the **[Query Builder]** button, you can create a subset of the features in the layer that will be visualized (also refer to section *Ferramenta de Consulta*).

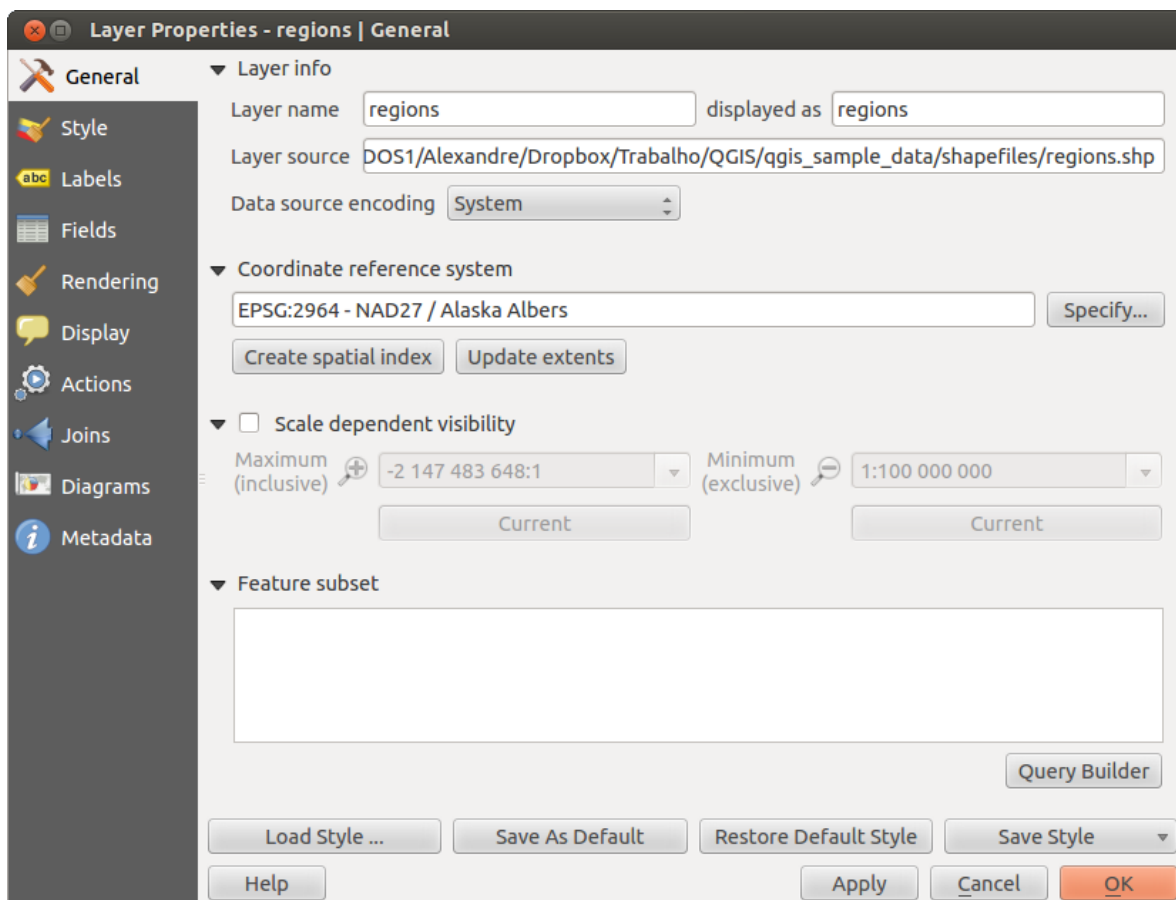





Figura 12.29: Menu geral na janela de diálogo das propriedades dos vectores 

12.3.5 Menu Renderização

QGIS 2.2 introduces support for on-the-fly feature generalisation. This can improve rendering times when drawing many complex features at small scales. This feature can be enabled or disabled in the layer settings using the  *Simplify geometry* option. There is also a new global setting that enables generalisation by default for newly added layers (see section *Opções*). **Note:** Feature generalisation may introduce artefacts into your rendered output in some cases. These may include slivers between polygons and inaccurate rendering when using offset-based symbol layers.

12.3.6 Menu Mostrar

 This menu is specifically created for Map Tips. It includes a new feature: Map Tip display text in HTML. While you can still choose a *Field* to be displayed when hovering over a feature on the map, it is now possible to insert HTML code that creates a complex display when hovering over a feature. To activate Map Tips, select the menu option *View* → *MapTips*. Figure Display 1 shows an example of HTML code.

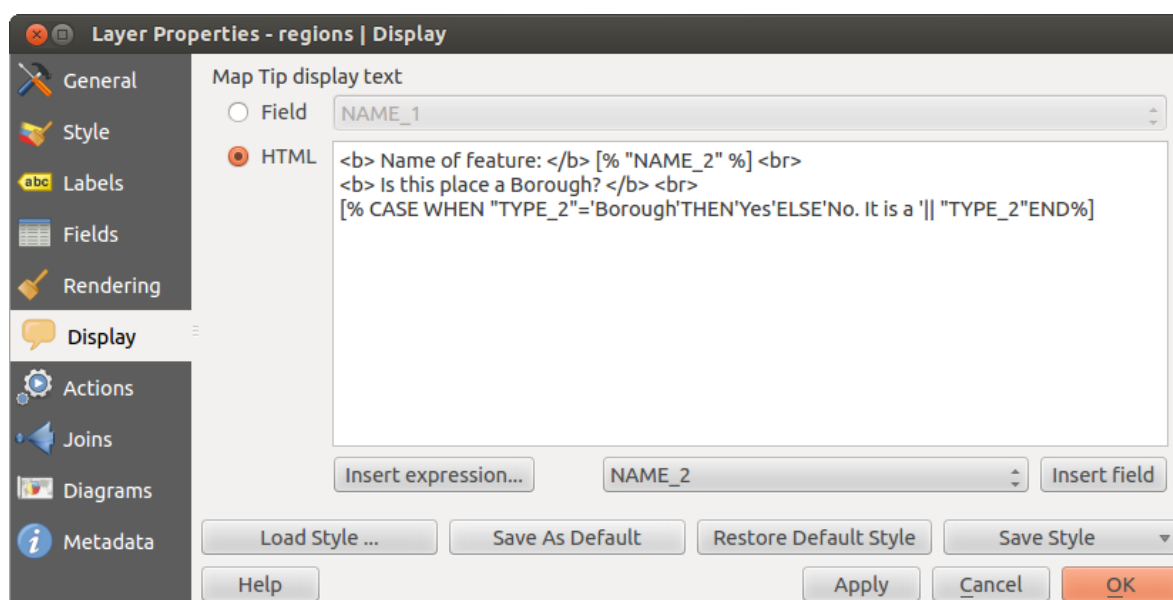



Figura 12.30: Código HTML para as dicas de mapa 


12.3.7 Ações

 QGIS provides the ability to perform an action based on the attributes of a feature. This can be used to perform any number of actions, for example, running a program with arguments built from the attributes of a feature or passing parameters to a web reporting tool.

Actions are useful when you frequently want to run an external application or view a web page based on one or more values in your vector layer. They are divided into six types and can be used like this:

- Acções do tipo Genérico, Mac, Windows, e Unix começam um processo externo.
- Acções Python executam expressões Python.
- Generic and Python actions are visible everywhere.
- Mac, Windows and Unix actions are visible only on the respective platform (i.e., you can define three 'Edit' actions to open an editor and the users can only see and execute the one 'Edit' action for their platform to run the editor).



Figura 12.31: Dicas de mapa feitas com código HTML 

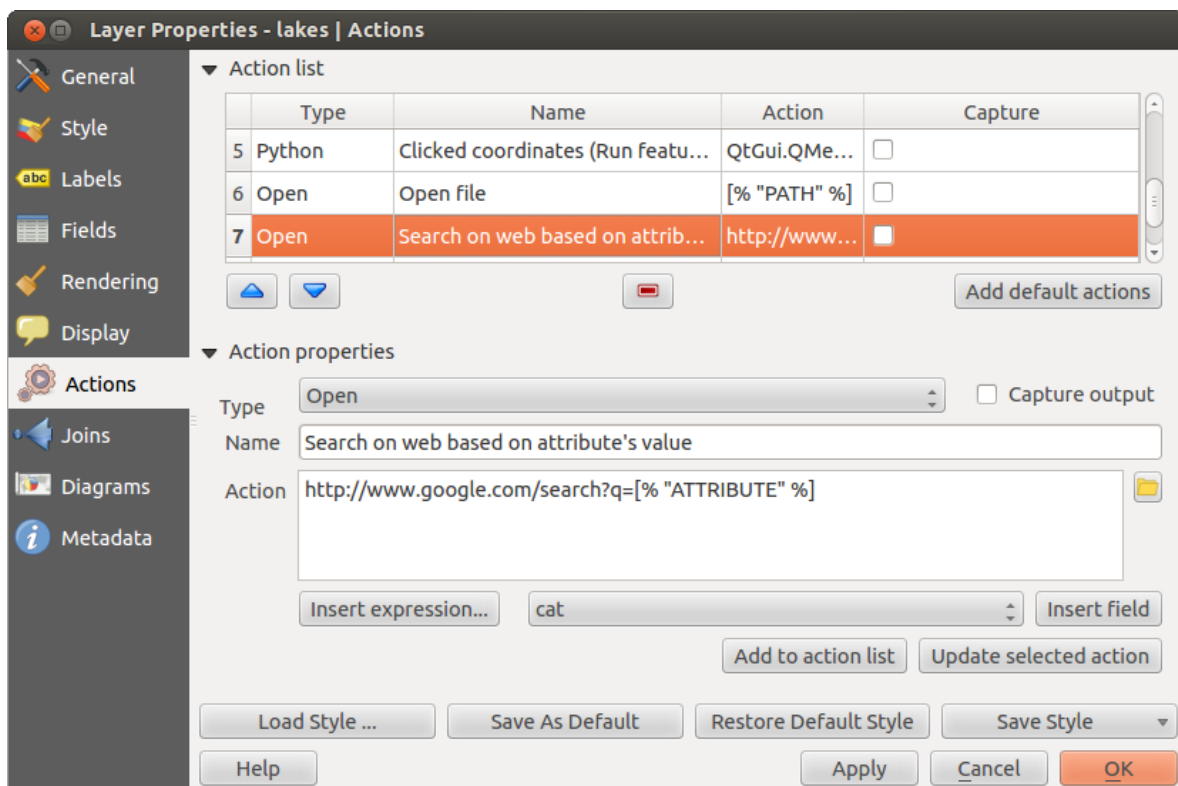



Figura 12.32: Overview action dialog with some sample actions 

There are several examples included in the dialog. You can load them by clicking on **[Add default actions]**. One example is performing a search based on an attribute value. This concept is used in the following discussion.

Definindo Acções

Attribute actions are defined from the vector *Layer Properties* dialog. To define an action, open the vector *Layer Properties* dialog and click on the *Actions* menu. Go to the *Action properties*. Select 'Generic' as type and provide a descriptive name for the action. The action itself must contain the name of the application that will be executed when the action is invoked. You can add one or more attribute field values as arguments to the application. When the action is invoked, any set of characters that start with a % followed by the name of a field will be replaced by the value of that field. The special characters %% will be replaced by the value of the field that was selected from the identify results or attribute table (see [using_actions](#) below). Double quote marks can be used to group text into a single argument to the program, script or command. Double quotes will be ignored if preceded by a backslash.

If you have field names that are substrings of other field names (e.g., `coll` and `coll0`), you should indicate that by surrounding the field name (and the % character) with square brackets (e.g., `[%coll0]`). This will prevent the `%coll0` field name from being mistaken for the `%coll` field name with a 0 on the end. The brackets will be removed by QGIS when it substitutes in the value of the field. If you want the substituted field to be surrounded by square brackets, use a second set like this: `[[%coll0]]`.

Using the *Identify Features* tool, you can open the *Identify Results* dialog. It includes a (*Derived*) item that contains information relevant to the layer type. The values in this item can be accessed in a similar way to the other fields by preceding the derived field name with `(Derived) .`. For example, a point layer has an X and Y field, and the values of these fields can be used in the action with `%(Derived) .X` and `%(Derived) .Y`. The derived attributes are only available from the *Identify Results* dialog box, not the *Attribute Table* dialog box.




Dois exemplo de acções são exibidos em baixo:

- `konqueror http://www.google.com/search?q=%nam`
- `konqueror http://www.google.com/search?q=%%`



In the first example, the web browser `konqueror` is invoked and passed a URL to open. The URL performs a Google search on the value of the `nam` field from our vector layer. Note that the application or script called by the action must be in the path, or you must provide the full path. To be certain, we could rewrite the first example as: `/opt/kde3/bin/konqueror http://www.google.com/search?q=%nam`. This will ensure that the `konqueror` application will be executed when the action is invoked.

The second example uses the %% notation, which does not rely on a particular field for its value. When the action is invoked, the %% will be replaced by the value of the selected field in the identify results or attribute table.

Usando Acções

Actions can be invoked from either the *Identify Results* dialog, an *Attribute Table* dialog or from *Run Feature Action* (recall that these dialogs can be opened by clicking  Identify Features or  Open Attribute Table or  Run Feature Action). To invoke an action, right click on the record and choose the action from the pop-up menu. Actions are listed in the popup menu by the name you assigned when defining the action. Click on the action you wish to invoke.

If you are invoking an action that uses the %% notation, right-click on the field value in the *Identify Results* dialog or the *Attribute Table* dialog that you wish to pass to the application or script.

Here is another example that pulls data out of a vector layer and inserts it into a file using `bash` and the `echo` command (so it will only work on  or perhaps ). The layer in question has fields for a species name `taxon_name`, latitude `lat` and longitude `long`. We would like to be able to make a spatial selection of localities and export these field values to a text file for the selected record (shown in yellow in the QGIS map area). Here is the action to achieve this:

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

Após seleccionar umas localidades e executado a acção para cada um, a abertura do ficheiro irá mostrar algo como isto:

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
```

```
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

As an exercise, we can create an action that does a Google search on the `lakes` layer. First, we need to determine the URL required to perform a search on a keyword. This is easily done by just going to Google and doing a simple search, then grabbing the URL from the address bar in your browser. From this little effort, we see that the format is `http://google.com/search?q=qgis`, where `QGIS` is the search term. Armed with this information, we can proceed:

1. Garanta que a camada “`lakes`” está carregada.
2. Open the *Layer Properties* dialog by double-clicking on the layer in the legend, or right-click and choose *Properties* from the pop-up menu.
3. Clique no menu *Ações*.
4. Introduza um nome para a acção, por exemplo `Google Search`.
5. For the action, we need to provide the name of the external program to run. In this case, we can use `Firefox`. If the program is not in your path, you need to provide the full path.
6. Following the name of the external application, add the URL used for doing a Google search, up to but not including the search term: `http://google.com/search?q=`
7. The text in the *Action* field should now look like this: `firefox http://google.com/search?q=`
8. Click on the drop-down box containing the field names for the `lakes` layer. It’s located just to the left of the **[Insert Field]** button.
9. From the drop-down box, select ‘`NAMES`’ and click **[Insert Field]**.
10. O seu texto da acção será algo como isto:

```
firefox http://google.com/search?q=%NAMES
```

11. Para finalizar a acção, clique no botão **[Adicionar à lista de acções]**

This completes the action, and it is ready to use. The final text of the action should look like this:

```
firefox http://google.com/search?q=%NAMES
```

We can now use the action. Close the *Layer Properties* dialog and zoom in to an area of interest. Make sure the `lakes` layer is active and identify a lake. In the result box you’ll now see that our action is visible:

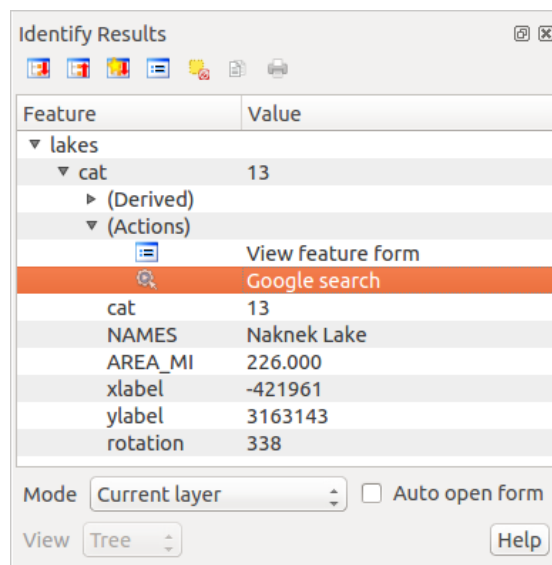



Figura 12.33: Seleccione o elemento e escolha a acção 

When we click on the action, it brings up Firefox and navigates to the URL <http://www.google.com/search?q=Tustumena>. It is also possible to add further attribute fields to the action. Therefore, you can add a + to the end of the action text, select another field and click on **[Insert Field]**. In this example, there is just no other field available that would make sense to search for.

You can define multiple actions for a layer, and each will show up in the *Identify Results* dialog.

There are all kinds of uses for actions. For example, if you have a point layer containing locations of images or photos along with a file name, you could create an action to launch a viewer to display the image. You could also use actions to launch web-based reports for an attribute field or combination of fields, specifying them in the same way we did in our Google search example.

Podemos também efectuar exemplos mais complexo, por exemplo, usando acções **Python**.

Usually, when we create an action to open a file with an external application, we can use absolute paths, or eventually relative paths. In the second case, the path is relative to the location of the external program executable file. But what about if we need to use relative paths, relative to the selected layer (a file-based one, like a shapefile or SpatiaLite)? The following code will do the trick:

```
command = "firefox";
imagerelpath = "images_test/test_image.jpg";
layer = qgis.utils.iface.activeLayer();
import os.path;
layerpath = layer.source() if layer.providerType() == 'ogr'
    else (qgis.core.QgsDataSourceURI(layer.source()).database()
    if layer.providerType() == 'spatialite' else None);
path = os.path.dirname(str(layerpath));
image = os.path.join(path, imagerelpath);
import subprocess;
subprocess.Popen( [command, image ] );
```

We just have to remember that the action is one of type *Python* and the *command* and *imagerelpath* variables must be changed to fit our needs.

But what about if the relative path needs to be relative to the (saved) project file? The code of the Python action would be:

```
command="firefox";
imagerelpath="images/test_image.jpg";
projectpath=qgis.core.QgsProject.instance().fileName();
import os.path; path=os.path.dirname(str(projectpath)) if projectpath != '' else None;
image=os.path.join(path, imagerelpath);
import subprocess;
subprocess.Popen( [command, image ] );
```



Another Python action example is the one that allows us to add new layers to the project. For instance, the following examples will add to the project respectively a vector and a raster. The names of the files to be added to the project and the names to be given to the layers are data driven (*filename* and *layername* are column names of the table of attributes of the vector where the action was created):

```
qgis.utils.iface.addVectorLayer('/yourpath/[% "filename" %].shp', '[% "layername" %]',
    'ogr')
```

Para adicionar um raster (uma imagem TIF neste exemplo), ficará:

```
qgis.utils.iface.addRasterLayer('/yourpath/[% "filename" %].tif', '[% "layername" %]
')
```

12.3.8 Menu União

 The *Joins* menu allows you to join a loaded attribute table to a loaded vector layer. After clicking , the *Add vector join* dialog appears. As key columns, you have to define a join layer you want to connect with the

target vector layer. Then, you have to specify the join field that is common to both the join layer and the target layer. Now you can also specify a subset of fields from the joined layer based on the checkbox *Choose which fields are joined*. As a result of the join, all information from the join layer and the target layer are displayed in the attribute table of the target layer as joined information. If you specified a subset of fields only these fields are displayed in the attribute table of the target layer.

QGIS currently has support for joining non-spatial table formats supported by OGR (e.g., CSV, DBF and Excel), delimited text and the PostgreSQL provider (see [figure_joins_1](#)).

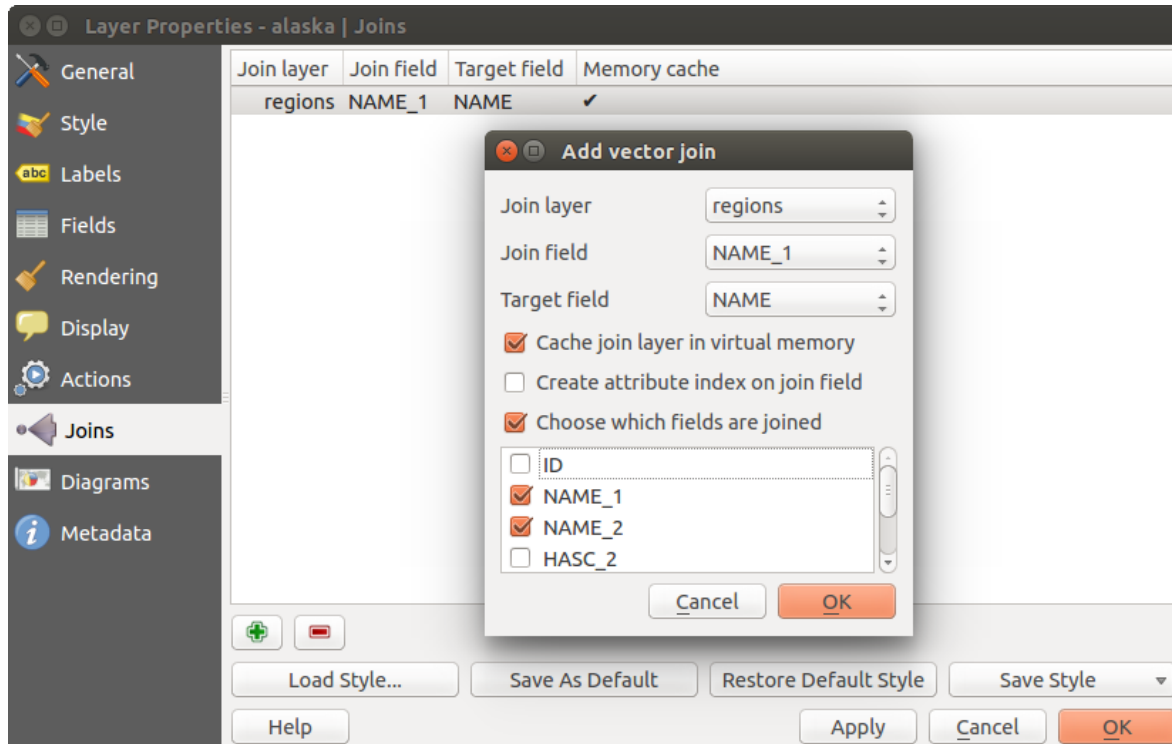


Figura 12.34: Une uma tabela de atributos com uma tabela de um vector existente 🐧

Additionally, the add vector join dialog allows you to:

- *Cache join layer in virtual memory*
- *Create attribute index on the join field*

12.3.9 Menu Diagramas



The *Diagrams* menu allows you to add a graphic overlay to a vector layer (see [figure_diagrams_1](#)).

The current core implementation of diagrams provides support for pie charts, text diagrams and histograms.

The menu is divided into four tabs: *Appearance*, *Size*, *Position* and *Options*.

In the cases of the text diagram and pie chart, text values of different data columns are displayed one below the other with a circle or a box and dividers. In the *Size* tab, diagram size is based on a fixed size or on linear scaling according to a classification attribute. The placement of the diagrams, which is done in the *Position* tab, interacts with the new labeling, so position conflicts between diagrams and labels are detected and solved. In addition, chart positions can be fixed manually.

We will demonstrate an example and overlay on the Alaska boundary layer a text diagram showing temperature data from a climate vector layer. Both vector layers are part of the QGIS sample dataset (see section *Amostra de Dados*).

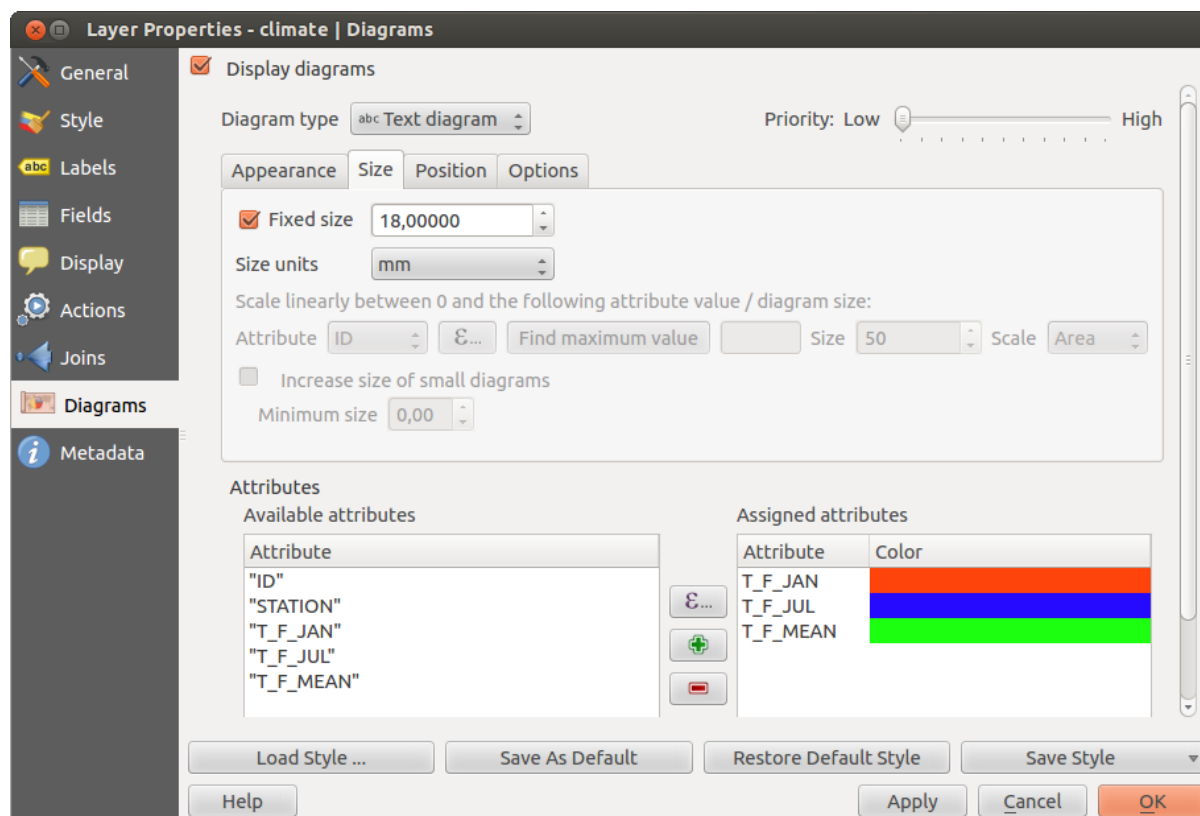






Figura 12.35: Vector properties dialog with diagram menu 

1. First, click on the  icon, browse to the QGIS sample dataset folder, and load the two vector shape layers `alaska.shp` and `climate.shp`.
2. Faça duplo clique na camada `climate` na legenda do mapa para abrir a janela das *Propriedades da Camada*
3. Click on the *Diagrams* menu, activate *Display diagrams*, and from the *Diagram type*  combo box, select 'Text diagram'.
4. In the *Appearance* tab, we choose a light blue as background color, and in the *Size* tab, we set a fixed size to 18 mm.
5. In the *Position* tab, placement could be set to 'Around Point'.
6. In the diagram, we want to display the values of the three columns `T_F_JAN`, `T_F_JUL` and `T_F_MEAN`. First select `T_F_JAN` as *Attributes* and click the  button, then `T_F_JUL`, and finally `T_F_MEAN`.
7. Agora clique [**Aplicar**] para exibir o diagrama na janela principal do QGIS.
8. You can adapt the chart size in the *Size* tab. Deactivate the *Fixed size* and set the size of the diagrams on the basis of an attribute with the [**Find maximum value**] button and the *Size* menu. If the diagrams appear too small on the screen, you can activate the *Increase size of small diagrams* checkbox and define the minimum size of the diagrams.
9. Change the attribute colors by double clicking on the color values in the *Assigned attributes* field. [Figure_diagrams_2](#) gives an idea of the result.
10. Finalmente, clique [**Ok**].

Remember that in the *Position* tab, a *Data defined position* of the diagrams is possible. Here, you can use attributes to define the position of the diagram. You can also set a scale-dependent visibility in the *Appearance* tab.

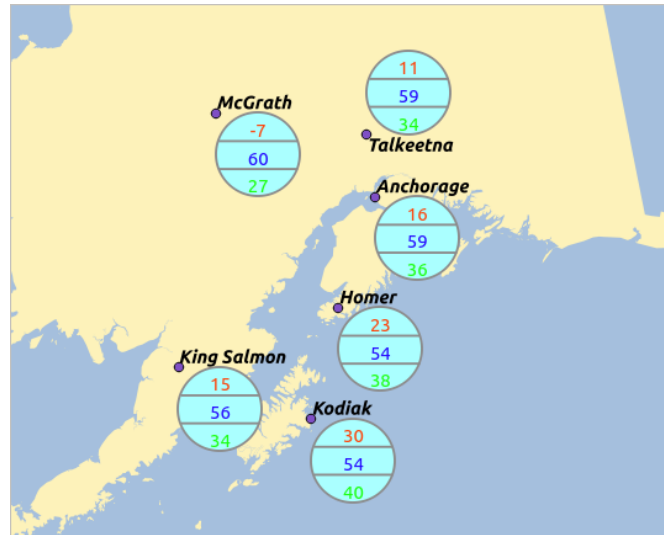




Figura 12.36: Diagrama a partir da informação da temperatura sobreposta no mapa 

The size and the attributes can also be an expression. Use the  button to add an expression. See [Expressions](#) chapter for more information and example.

12.3.10 Menu Metadados





The *Metadata* menu consists of *Description*, *Attribution*, *MetadataURL* and *Properties* sections.

In the *Properties* section, you get general information about the layer, including specifics about the type and location, number of features, feature type, and editing capabilities. The *Extents* table provides you with layer extent information and the *Layer Spatial Reference System*, which is information about the CRS of the layer. This is a quick way to get information about the layer.

Additionally, you can add or edit a title and abstract for the layer in the *Description* section. It's also possible to define a *Keyword list* here. These keyword lists can be used in a metadata catalogue. If you want to use a title from an XML metadata file, you have to fill in a link in the *DataUrl* field. Use *Attribution* to get attribute data from an XML metadata catalogue. In *MetadataUrl*, you can define the general path to the XML metadata catalogue. This information will be saved in the QGIS project file for subsequent sessions and will be used for QGIS server.

12.4 Expressions

The **Expressions** feature are available through the field calculator or the add a new column button in the attribute table or the Field tab in the Layer properties ; through the graduated, categorized and rule-based rendering in the Style tab of the Layer properties ; through the expression-based labeling  in the  Labeling core application ; through the feature selection and through the diagram tab of the Layer properties.

There are powerful way to manipulate attribute value in order to dynamically change the final value in order to change the geometry style, the content of the label, the value for diagram, select some feature or create virtual column.

12.4.1 Functions List

The **Function List** contains functions as well as fields and values. View the help function in the **Selected Function Help**. In **Expression** you see the calculation expressions you create with the **Function List**. For the most

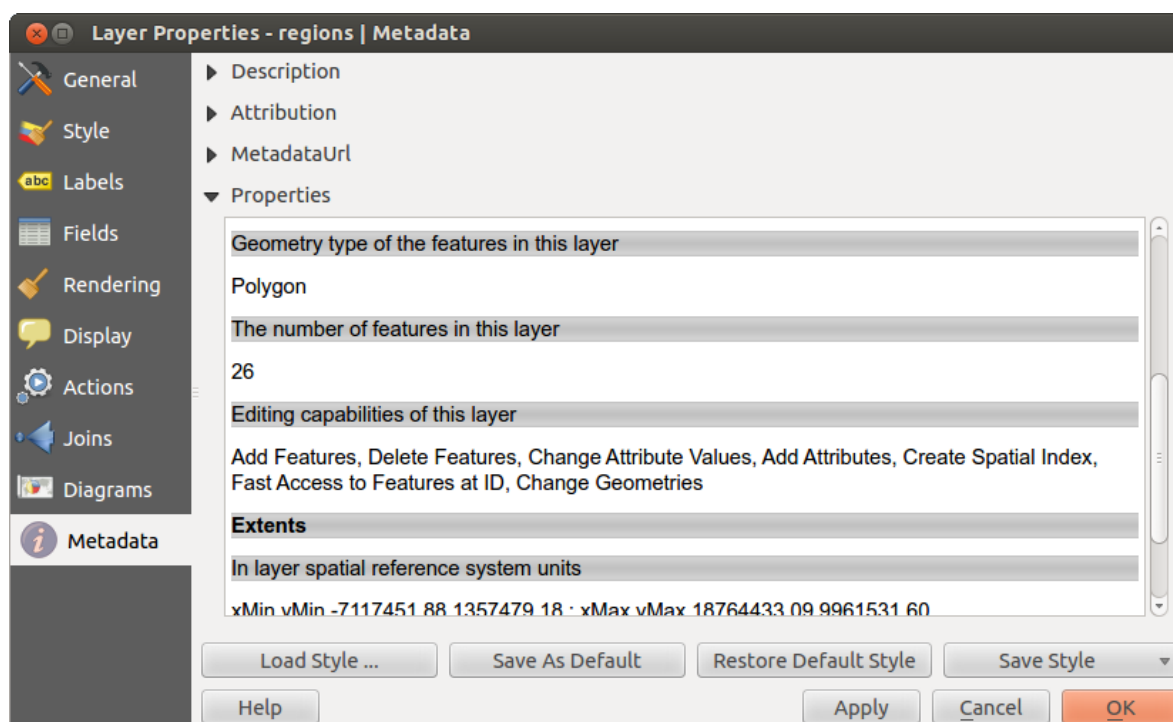


Figura 12.37: Janela de diálogo do menu de metadados das propriedades das camadas vectoriais 

commonly used operators, see **Operators**.

In the **Function List**, click on *Fields and Values* to view all attributes of the attribute table to be searched. To add an attribute to the Field calculator **Expression** field, double click its name in the *Fields and Values* list. Generally, you can use the various fields, values and functions to construct the calculation expression, or you can just type it into the box. To display the values of a field, you just right click on the appropriate field. You can choose between *Load top 10 unique values* and *Load all unique values*. On the right side, the **Field Values** list opens with the unique values. To add a value to the Field calculator **Expression** box, double click its name in the **Field Values** list.

The *Operators*, *Math*, *Conversions*, *String*, *Geometry* and *Record* groups provide several functions. In *Operators*, you find mathematical operators. Look in *Math* for mathematical functions. The *Conversions* group contains functions that convert one data type to another. The *String* group provides functions for data strings. In the *Geometry* group, you find functions for geometry objects. With *Record* group functions, you can add a numeration to your data set. To add a function to the Field calculator **Expression** box, click on the > and then double click the function.

Operators

This group contains operators (e.g., +, -, *).

a + b	a plus b
a - b	a minus b
a * b	a multiplied by b
a / b	a divided by b
a % b	a modulo b (for example, 7 % 2 = 1, or 2 fits into 7 three times with remainder 1)
a ^ b	a power b (for example, 2^2=4 or 2^3=8)
a = b	a and b are equal
a > b	a is larger than b
a < b	a is smaller than b
a <> b	a and b are not equal
a != b	a and b are not equal

a <= b	a is less than or equal to b
a >= b	a is larger than or equal to b
a ~ b	a matches the regular expression b
+ a	positive sign
- a	negative value of a
	joins two values together into a string 'Hello' ' world'
LIKE	returns 1 if the string matches the supplied pattern
ILIKE	returns 1 if the string matches case-insensitive the supplied pattern (ILIKE can be used instead of LIKE to make the match case-insensitive)
IS	returns 1 if a is the same as b
OR	returns 1 when condition a or b is true
AND	returns 1 when condition a and b are true
NOT	returns 1 if a is not the same as b
column name "column name"	value of the field column name, take care to not be confused with simple quote, see below
'string'	a string value, take care to not be confused with double quote, see above
NULL	null value
a IS NULL	a has no value
a IS NOT NULL	a has a value
a IN (value[,value])	a is below the values listed
a NOT IN (value[,value])	a is not below the values listed

Some example:

- Joins a string and a value from a column name:


```
'My feature's id is: ' || "gid"
```
- Test if the “description” attribute field starts with the ‘Hello’ string in the value (note the position of the % character):


```
"description" LIKE 'Hello%'
```

Conditionals

This group contains functions to handle conditional checks in expressions.

CASE	evaluates multiple expressions and returns a result
CASE ELSE	evaluates multiple expressions and returns a result
coalesce	returns the first non-NULL value from the expression list
regexp_match	returns true if any part of a string matches the supplied regular expression

Some example:

- Send back a value if the first condition is true, else another value:


```
CASE WHEN "software" LIKE '%QGIS%' THEN 'QGIS' ELSE 'Other'
```

Mathematical Functions

This group contains math functions (e.g., square root, sin and cos).

sqrt(a)	square root of a
abs	returns the absolute value of a number
sin(a)	sine of a

cos(a)	cosine of a
tan(a)	tangent of a
asin(a)	arcsin of a
acos(a)	arccos of a
atan(a)	arctan of a
atan2(y,x)	arctan of y/x using the signs of the two arguments to determine the quadrant of the result
exp	exponential of a value
ln	value of the natural logarithm of the passed expression
log10	value of the base 10 logarithm of the passed expression
log	value of the logarithm of the passed value and base
round	round to number of decimal places
rand	random integer within the range specified by the minimum and maximum argument (inclusive)
randf	random float within the range specified by the minimum and maximum argument (inclusive)
max	largest value in a set of values
min	smallest value in a set of values
clamp	restricts an input value to a specified range
scale_linear	transforms a given value from an input domain to an output range using linear interpolation
scale_exp	transforms a given value from an input domain to an output range using an exponential curve
floor	rounds a number downwards
ceil	rounds a number upwards
\$pi	pi as value for calculations

Conversions

This group contains functions to convert one data type to another (e.g., string to integer, integer to string).

toint	converts a string to integer number
toreal	converts a string to real number
tostring	converts number to string
todatetime	converts a string into Qt data time type
todate	converts a string into Qt data type
totime	converts a string into Qt time type
tointerval	converts a string to an interval type (can be used to take days, hours, months, etc. off a date)

Date and Time Functions

This group contains functions for handling date and time data.

\$now	current date and time
age	difference between two dates
year	extract the year part from a date, or the number of years from an interval
month	extract the month part from a date, or the number of months from an interval

week	extract the week number from a date, or the number of weeks from an interval
day	extract the day from a date, or the number of days from an interval
hour	extract the hour from a datetime or time, or the number of hours from an interval
minute	extract the minute from a datetime or time, or the number of minutes from an interval
second	extract the second from a datetime or time, or the number of minutes from an interval

Some example:

- Get the month and the year of today in the format “10/2014”

```
month($now) || '/' || year($now)
```

String Functions

This group contains functions that operate on strings (e.g., that replace, convert to upper case).

lower	convert string a to lower case
upper	convert string a to upper case
title	converts all words of a string to title case (all words lower case with leading capital letter)
trim	removes all leading and trailing white space (spaces, tabs, etc.) from a string
wordwrap	returns a string wrapped to a maximum/minimum number of characters
length	length of string a
replace	returns a string with the supplied string replaced
regexp_replace(a,this,that)	returns a string with the supplied regular expression replaced
regexp_substr	returns the portion of a string which matches a supplied regular expression
substr(*a*,from,len)	returns a part of a string
concat	concatenates several strings to one
strpos	returns the index of a regular expression in a string
left	returns a substring that contains the n leftmost characters of the string
right	returns a substring that contains the n rightmost characters of the string
rpad	returns a string with supplied width padded using the fill character
lpad	returns a string with supplied width padded using the fill character
format	formats a string using supplied arguments
format_number	returns a number formatted with the locale separator for thousands (also truncates the number to the number of supplied places)
format_date	formats a date type or string into a custom string format

Color Functions

This group contains functions for manipulating colors.

color_rgb	returns a string representation of a color based on its red, green, and blue components
color_rgba	returns a string representation of a color based on its red, green, blue, and alpha (transparency) components
ramp_color	returns a string representing a color from a color ramp
color_hsl	returns a string representation of a color based on its hue, saturation, and lightness attributes
color_hsla	returns a string representation of a color based on its hue, saturation, lightness and alpha (transparency) attributes
color_hsv	returns a string representation of a color based on its hue, saturation, and value attributes
color_hsva	returns a string representation of a color based on its hue, saturation, value and alpha (transparency) attributes
color_cmyk	returns a string representation of a color based on its cyan, magenta, yellow and black components
color_cmyka	returns a string representation of a color based on its cyan, magenta, yellow, black and alpha (transparency) components

Geometry Functions

This group contains functions that operate on geometry objects (e.g., length, area).

\$geometry	returns the geometry of the current feature (can be used for processing with other functions)
\$area	returns the area size of the current feature
\$length	returns the length size of the current feature
\$perimeter	returns the perimeter length of the current feature
\$x	returns the x coordinate of the current feature
\$y	returns the y coordinate of the current feature
xat	retrieves the nth x coordinate of the current feature. n given as a parameter of the function
yat	retrieves the nth y coordinate of the current feature. n given as a parameter of the function
xmin	returns the minimum x coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
xmax	returns the maximum x coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
ymin	returns the minimum y coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
ymax	returns the maximum y coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
geomFromWKT	returns a geometry created from a well-known text (WKT) representation
geomFromGML	returns a geometry from a GML representation of geometry
bbox	
disjoint	returns 1 if the geometries do not share any space together
intersects	returns 1 if the geometries spatially intersect (share any portion of space) and 0 if they don't
touches	returns 1 if the geometries have at least one point in common, but their interiors do not intersect
crosses	returns 1 if the supplied geometries have some, but not all, interior points in common
contains	returns true if and only if no points of b lie in the exterior of a, and at least one point of the interior of b lies in the interior of a

overlaps	returns 1 if the geometries share space, are of the same dimension, but are not completely contained by each other
within	returns 1 if geometry a is completely inside geometry b
buffer	returns a geometry that represents all points whose distance from this geometry is less than or equal to distance
centroid	returns the geometric center of a geometry
bounds	returns a geometry which represents the bounding box of an input geometry. Calculations are in the Spatial Reference System of this Geometry.
bounds_width	returns the width of the bounding box of a geometry. Calculations are in the Spatial Reference System of this Geometry.
bounds_height	returns the height of the bounding box of a geometry. Calculations are in the Spatial Reference System of this Geometry.
convexHull	returns the convex hull of a geometry (this represents the minimum convex geometry that encloses all geometries within the set)
difference	returns a geometry that represents that part of geometry a that does not intersect with geometry b
distance	returns the minimum distance (based on spatial ref) between two geometries in projected units
intersection	returns a geometry that represents the shared portion of geometry a and geometry b
symDifference	returns a geometry that represents the portions of a and b that do not intersect
combine	returns the combination of geometry a and geometry b
union	returns a geometry that represents the point set union of the geometries
geomToWKT	returns the well-known text (WKT) representation of the geometry without SRID metadata

Record Functions

This group contains functions that operate on record identifiers.

\$rownum	returns the number of the current row
\$id	returns the feature id of the current row
\$currentfeature	returns the current feature being evaluated. This can be used with the 'attribute' function to evaluate attribute values from the current feature.
\$scale	returns the current scale of the map canvas
\$uuid	generates a Universally Unique Identifier (UUID) for each row. Each UUID is 38 characters long.
getFeature	returns the first feature of a layer matching a given attribute value.
attribute	returns the value of a specified attribute from a feature.
\$map	returns the id of the current map item if the map is being drawn in a composition, or "canvas" if the map is being drawn within the main QGIS window.

Fields and Values

Contains a list of fields from the layer. Sample values can also be accessed via right-click.

Select the field name from the list, then right-click to access a context menu with options to load sample values from the selected field.

Fields name should be double-quoted. Values or string should be simple-quoted.

12.5 Editando

QGIS supports various capabilities for editing OGR, SpatiaLite, PostGIS, MSSQL Spatial and Oracle Spatial vector layers and tables.

Note: The procedure for editing GRASS layers is different - see section *Digitalizando e editando as camadas vectoriais GRASS* for details.

Tip: Edições Simultâneas




This version of QGIS does not track if somebody else is editing a feature at the same time as you are. The last person to save their edits wins.

12.5.1 Configurando a Tolerância de Atracção e Raio de Pesquisa

Antes de podermos editar os vértices, necessitamos de configurar a tolerância e o raio de pesquisa para um valor que nos permita uma edição ideal das geometrias da camada vectorial.

Tolerância de Atracção

Snapping tolerance is the distance QGIS uses to search for the closest vertex and/or segment you are trying to connect to when you set a new vertex or move an existing vertex. If you aren't within the snapping tolerance, QGIS will leave the vertex where you release the mouse button, instead of snapping it to an existing vertex and/or segment. The snapping tolerance setting affects all tools that work with tolerance.

1. A general, project-wide snapping tolerance can be defined by choosing *Settings* →  *Options*. On Mac, go to *QGIS* →  *Preferences...*. On Linux: *Edit* →  *Options*. In the *Digitizing* tab, you can select between 'to vertex', 'to segment' or 'to vertex and segment' as default snap mode. You can also define a default snapping tolerance and a search radius for vertex edits. The tolerance can be set either in map units or in pixels. The advantage of choosing pixels is that the snapping tolerance doesn't have to be changed after zoom operations. In our small digitizing project (working with the Alaska dataset), we define the snapping units in feet. Your results may vary, but something on the order of 300 ft at a scale of 1:10000 should be a reasonable setting.
2. A layer-based snapping tolerance can be defined by choosing *Settings* → (or *File* →) *Snapping options...* to enable and adjust snapping mode and tolerance on a layer basis (see [figure_edit_1](#)).

Note that this layer-based snapping overrides the global snapping option set in the *Digitizing* tab. So, if you need to edit one layer and snap its vertices to another layer, then enable snapping only on the `snap to` layer, then decrease the global snapping tolerance to a smaller value. Furthermore, snapping will never occur to a layer that is not checked in the snapping options dialog, regardless of the global snapping tolerance. So be sure to mark the checkbox for those layers that you need to snap to.

Raio de pesquisa

Search radius is the distance QGIS uses to search for the closest vertex you are trying to move when you click on the map. If you aren't within the search radius, QGIS won't find and select any vertex for editing, and it will pop up an annoying warning to that effect. Snap tolerance and search radius are set in map units or pixels, so you may find you need to experiment to get them set right. If you specify too big of a tolerance, QGIS may snap to the

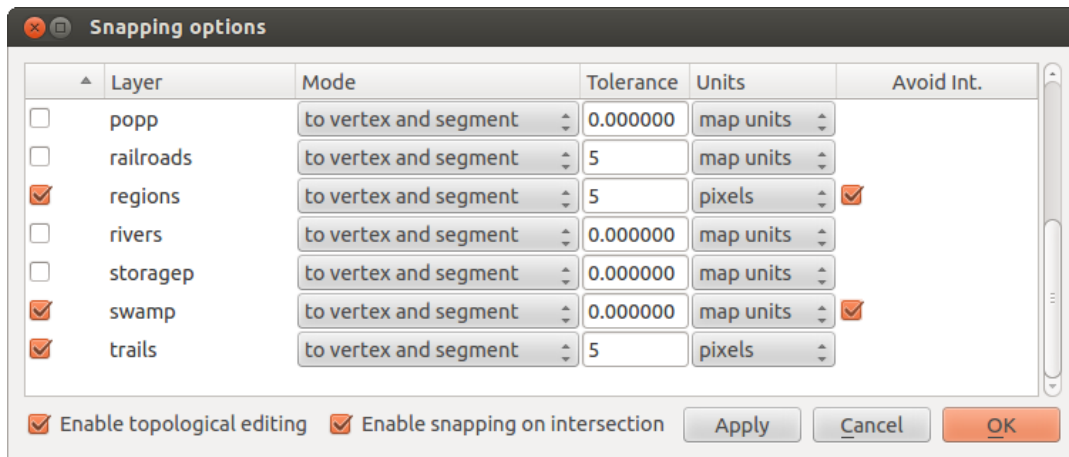







Figura 12.38: Edit snapping options on a layer basis 

wrong vertex, especially if you are dealing with a large number of vertices in close proximity. Set search radius too small, and it won't find anything to move.


The search radius for vertex edits in layer units can be defined in the *Digitizing* tab under *Settings* →  *Options*. This is the same place where you define the general, project- wide snapping tolerance.

12.5.2 Ampliando e Movendo

Antes de editar uma camada, deve fazer uma ampliação à zona da área de interesse. Isto evita que espere enquanto os marcadores dos vértices são renderizados em toda a camada.

Além de usar os ícones  mover mapa e  aproximar /  afastar na barra de ferramentas com o rato, a navegação pode também ser feita com a roda do rato, espaço e as teclas de direcção.

Ampliando e movendo com a roda do rato

While digitizing, you can press the mouse wheel to pan inside of the main window, and you can roll the mouse wheel to zoom in and out on the map. For zooming, place the mouse cursor inside the map area and roll it forward (away from you) to zoom in and backwards (towards you) to zoom out. The mouse cursor position will be the center of the zoomed area of interest. You can customize the behavior of the mouse wheel zoom using the *Map tools* tab under the *Settings* →  *Options* menu.

Movendo a direcção com as setas do teclado


Panning the map during digitizing is possible with the arrow keys. Place the mouse cursor inside the map area, and click on the right arrow key to pan east, left arrow key to pan west, up arrow key to pan north, and down arrow key to pan south.

You can also use the space bar to temporarily cause mouse movements to pan the map. The PgUp and PgDown keys on your keyboard will cause the map display to zoom in or out without interrupting your digitizing session.


12.5.3 Edição Topológica

Besides layer-based snapping options, you can also define topological functionalities in the *Snapping options...* dialog in the *Settings* (or *File*) menu. Here, you can define *Enable topological editing*, and/or for polygon layers, you can activate the column *Avoid Int.*, which avoids intersection of new polygons.


Activar edição topológica

The option  *Enable topological editing* is for editing and maintaining common boundaries in polygon mosaics. QGIS ‘detects’ a shared boundary in a polygon mosaic, so you only have to move the vertex once, and QGIS will take care of updating the other boundary.

Evitar intersecções de novos polígonos

The second topological option in the  *Avoid Int.* column, called *Avoid intersections of new polygons*, avoids overlaps in polygon mosaics. It is for quicker digitizing of adjacent polygons. If you already have one polygon, it is possible with this option to digitize the second one such that both intersect, and QGIS then cuts the second polygon to the common boundary. The advantage is that you don’t have to digitize all vertices of the common boundary.













Activar atracção nas intersecções

Another option is to use  *Enable snapping on intersection*. It allows you to snap on an intersection of background layers, even if there’s no vertex on the intersection.


12.5.4 Digitalizar uma camada existente


By default, QGIS loads layers read-only. This is a safeguard to avoid accidentally editing a layer if there is a slip of the mouse. However, you can choose to edit any layer as long as the data provider supports it, and the underlying data source is writable (i.e., its files are not read-only).

In general, tools for editing vector layers are divided into a digitizing and an advanced digitizing toolbar, described in section *Digitalização Avançada*. You can select and unselect both under *View → Toolbars →*. Using the basic digitizing tools, you can perform the following functions:


Ícone	Finalidade	Ícone	Finalidade
	Edições actuais		Alternar edição
	Adicionando Elementos: Capturar Ponto		Adicionando Elementos: Capturar Linha
	Adicionando Elementos: Capturar Polígono		Mover Elemento
	Ferramenta de Nós		Apagar Seleccionados
	Cortar Elementos		Copiar Elementos
	Colar Elementos		Guardar edições da camada

Edição da Tabela: Edição básica da camada vectorial pela barra de ferramentas




All editing sessions start by choosing the  *Toggle editing* option. This can be found in the context menu after right clicking on the legend entry for a given layer.

Alternatively, you can use the *Toggle Editing*  *Toggle editing* button from the digitizing toolbar to start or stop the editing mode. Once the layer is in edit mode, markers will appear at the vertices, and additional tool buttons on the editing toolbar will become available.

Tip: Guardar Regularmente

Lembre-se de  *Guardar Edições da Camada* regularmente. Isto irá também verificar que a sua fonte de dados aceita todas as alterações.

Adicionando Elementos

Pode usar os ícones  Adicionar Elemento,  Adicionar Elemento ou  Adicionar Elemento da barra de ferramentas para por o cursor QGIS no modo de digitalização.

Para cada elemento, primeiro digitaliza a geometria, e de seguida introduz os atributos. Para digitalizar a geometria, clique com o botão direito do rato na área do mapa para criar o primeiro ponto do seu novo elemento.

Para linhas e polígonos, mantenha o clique com o botão direito do rato para cada ponto adicional que pretende capturar. Quando acabar de adicionar os pontos, clique com o direito do rato em qualquer sítio da área do mapa para confirmar a finalização da introdução da geometria desse elemento.

The attribute window will appear, allowing you to enter the information for the new feature. [Figure_edit_2](#) shows setting attributes for a fictitious new river in Alaska. In the *Digitizing* menu under the *Settings* → *Options* menu, you can also activate *Suppress attributes pop-up windows after each created feature* and *Reuse last entered attribute values*.

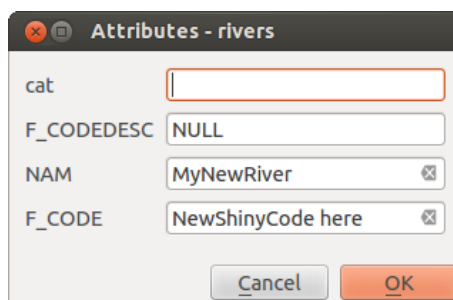






Figura 12.39: Introduza os Valores dos atributos na Janela após a digitalização do novo elemento vectorial 

With the  Move Feature(s) icon on the toolbar, you can move existing features.

Tip: Tipos de Valores de Atributo


For editing, the attribute types are validated during entry. Because of this, it is not possible to enter a number into a text column in the dialog *Enter Attribute Values* or vice versa. If you need to do so, you should edit the attributes in a second step within the *Attribute table* dialog.


Edições Actuais

This feature allows the digitization of multiple layers. Choose  *Save for Selected Layers* to save all changes you made in multiple layers. You also have the opportunity to  *Rollback for Selected Layers*, so that the digitization may be withdrawn for all selected layers. If you want to stop editing the selected layers,  *Cancel for Selected Layer(s)* is an easy way.


The same functions are available for editing all layers of the project.

Ferramenta de Nós


For shapefile-based layers as well as SpatialLite, PostgreSQL/PostGIS, MSSQL Spatial, and Oracle Spatial tables, the  Node Tool provides manipulation capabilities of feature vertices similar to CAD programs. It is possible to simply select multiple vertices at once and to move, add or delete them altogether. The node tool also works with 'on the fly' projection turned on, and it supports the topological editing feature. This tool is, unlike other tools in QGIS, persistent, so when some operation is done, selection stays active for this feature and tool. If the node tool is unable to find any features, a warning will be displayed.



It is important to set the property *Settings* →  *Options* → *Digitizing* → *Search Radius*: to a number greater than zero (i.e., 10). Otherwise, QGIS will not be able to tell which vertex is being edited.

Tip: Marcadores de Vértice

The current version of QGIS supports three kinds of vertex markers: ‘Semi-transparent circle’, ‘Cross’ and ‘None’. To change the marker style, choose  *Options* from the *Settings* menu, click on the *Digitizing* tab and select the appropriate entry.


Operações Básicas

Comece por activar a  Ferramenta de Nós e seleccione um elemento clicando em cima de um, As caixas vermelhos irão aparecer em cada vértice deste elemento.

- **Selecting vertices:** You can select vertices by clicking on them one at a time, by clicking on an edge to select the vertices at both ends, or by clicking and dragging a rectangle around some vertices. When a vertex is selected, its color changes to blue. To add more vertices to the current selection, hold down the `Ctrl` key while clicking. Hold down `Ctrl` or `Shift` when clicking to toggle the selection state of vertices (vertices that are currently unselected will be selected as usual, but also vertices that are already selected will become unselected).
- **Adding vertices:** To add a vertex, simply double click near an edge and a new vertex will appear on the edge near to the cursor. Note that the vertex will appear on the edge, not at the cursor position; therefore, it should be moved if necessary.
- **Eliminar vértices:** Depois de seleccionar os vértices para eliminar, clique na tecla `Delete`. Note-se que não pode usar a  Node Tool para eliminar um elemento completo; QGIS vai assegurar que mantém o número mínimo de vértices necessários para o tipo de elemento que está a trabalhar. Para eliminar um elemento completo, utilize a ferramenta  Delete Selected.
- **Mover vértices:** Selecione todos os vértices que pretende mover. Clique num vértice seleccionado ou na extremidade e arraste na direção que pretende mover. Todos os vértices seleccionados serão movidos em conjunto. Se a ferramenta snapping está ativa, toda a seleção pode saltar para o vértice ou linha mais próximos.

Each change made with the node tool is stored as a separate entry in the Undo dialog. Remember that all operations support topological editing when this is turned on. On-the-fly projection is also supported, and the node tool provides tooltips to identify a vertex by hovering the pointer over it.

Cortando, Copiando e Colando Elementos




Selected features can be cut, copied and pasted between layers in the same QGIS project, as long as destination layers are set to  Toggle editing beforehand.

Features can also be pasted to external applications as text. That is, the features are represented in CSV format, with the geometry data appearing in the OGC Well-Known Text (WKT) format.

However, in this version of QGIS, text features from outside QGIS cannot be pasted to a layer within QGIS. When would the copy and paste function come in handy? Well, it turns out that you can edit more than one layer at a time and copy/paste features between layers. Why would we want to do this? Say we need to do some work on a new layer but only need one or two lakes, not the 5,000 on our `big_lakes` layer. We can create a new layer and use copy/paste to plop the needed lakes into it.

As an example, we will copy some lakes to a new layer:

1. Carregue a camada que quer copiar a partir (camada de origem)
2. Carregue ou crie a camada que quer copiar para (camada de destino)
3. Começar a editar a camada de destino



4. Active a camada de origem clicando nela na legenda
5. Use a ferramenta  Seleccione Elemento Único para seleccionar os elemento(s) na camada fonte
6. Clique na ferramenta  Copiar Elementos
7. Active a camada de destino clicando na legenda
8. Clique na ferramenta  Colar Elementos
9. Parar a edição e guardar as alterações



What happens if the source and target layers have different schemas (field names and types are not the same)? QGIS populates what matches and ignores the rest. If you don't care about the attributes being copied to the target layer, it doesn't matter how you design the fields and data types. If you want to make sure everything - the feature and its attributes - gets copied, make sure the schemas match.

Tip: Congruência dos Elementos Colados



If your source and destination layers use the same projection, then the pasted features will have geometry identical to the source layer. However, if the destination layer is a different projection, then QGIS cannot guarantee the geometry is identical. This is simply because there are small rounding-off errors involved when converting between projections.

Apagando os Elementos Seleccionados

Se quisermos apagar o polígono todo, podemos fazê-lo seleccionando o polígono usando a ferramenta  Seleccionador de Elemento Único. Pode seleccionar múltiplos elementos para apagar. Uma vez feita a selecção do conjunto, use a ferramenta  Apagar Seleccionados para apagar os elementos.

The  Cut Features tool on the digitizing toolbar can also be used to delete features. This effectively deletes the feature but also places it on a “spatial clipboard”. So, we cut the feature to delete. We could then use the  Paste Features tool to put it back, giving us a one-level undo capability. Cut, copy, and paste work on the currently selected features, meaning we can operate on more than one at a time.

Guardando as Camadas Editadas

When a layer is in editing mode, any changes remain in the memory of QGIS. Therefore, they are not committed/saved immediately to the data source or disk. If you want to save edits to the current layer but want to continue editing without leaving the editing mode, you can click the  Save Layer Edits button. When you turn editing mode off with  Toggle editing (or quit QGIS for that matter), you are also asked if you want to save your changes or discard them.

If the changes cannot be saved (e.g., disk full, or the attributes have values that are out of range), the QGIS in-memory state is preserved. This allows you to adjust your edits and try again.

Tip: Integridade dos dados



É sempre boa ideia fazer cópias de segurança da sua fonte de dados antes de começar a editar. Enquanto os autores do QGIS fizeram todo o esforço para preservar a integridade dos seus dados, nós não oferecemos garantia neste sentido.

12.5.5 Digitalização Avançada

Ícone	Finalidade	Ícone	Finalidade
	Retroceder		Retomar
	Rodar Elemento(s)		Simplificar elemento
	Adicionar Anél		Adicionar Parte
	Fill Ring		Apagar Anél
	Apagar Parte		Refazer elementos
	Curva de Afastamento		Dividir Elementos
	Split Parts		Juntar Elementos Seleccionados
	Juntar Atributos dos Elementos Seleccionados		Rodar Símbolos de Pontos

Tabela de edição avançada: Barra de ferramentas de edição avançada de camadas vectoriais

Retroceder e Retomar

The  Undo and  Redo tools allows you to undo or redo vector editing operations. There is also a dockable widget, which shows all operations in the undo/redo history (see [Figure_edit_3](#)). This widget is not displayed by default; it can be displayed by right clicking on the toolbar and activating the Undo/Redo checkbox. Undo/Redo is however active, even if the widget is not displayed.

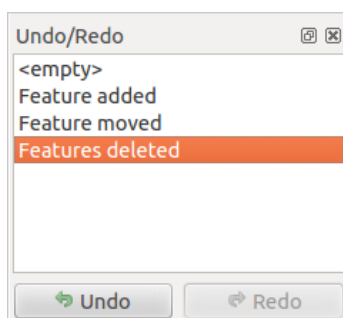





Figura 12.40: Refazer e desfazer passos de digitalização 🐧

When Undo is hit, the state of all features and attributes are reverted to the state before the reverted operation happened. Changes other than normal vector editing operations (for example, changes done by a plugin), may or may not be reverted, depending on how the changes were performed.

To use the undo/redo history widget, simply click to select an operation in the history list. All features will be reverted to the state they were in after the selected operation.


Rodar Elemento(s)

Use  Rotate Feature(s) to rotate one or multiple selected features in the map canvas. You first need to select the features and then press the  Rotate Feature(s) icon. The centroid of the feature(s) appears and will be the rotation anchor point. If you selected multiple features, the rotation anchor point will be the common center of the features. Press and drag the left mouse button in the desired direction to rotate the selected features.


It's also possible to create a user-defined rotation anchor point around which the selected feature will rotate. Select the features to rotate and activate the  Rotate Feature(s) tool. Press and hold the `Ctrl` button and move the mouse

pointer (without pressing the mouse button) to the place where you want the rotation anchor to be moved. Release the `Ctrl` button when the desired rotation anchor point is reached. Now, press and drag the left mouse button in the desired direction to rotate the selected feature(s).


Simplificar elemento

The  Simplify Feature tool allows you to reduce the number of vertices of a feature, as long as the geometry doesn't change and geometry type is not a multi geometry. First, select a feature. It will be highlighted by a red rubber band and a slider will appear. Moving the slider, the red rubber band will change its shape to show how the feature is being simplified. Click **[OK]** to store the new, simplified geometry. If a feature cannot be simplified (e.g. multi-polygons), a message will appear.

Adicionar Anél



You can create ring polygons using the  Add Ring icon in the toolbar. This means that inside an existing area, it is possible to digitize further polygons that will occur as a 'hole', so only the area between the boundaries of the outer and inner polygons remains as a ring polygon.

Adicionar Parte


You can  add part polygons to a selected multipolygon. The new part polygon must be digitized outside the selected multi-polygon.

Fill Ring


You can use the  Fill Ring function to add a ring to a polygon and add a new feature to the layer at the same time.

Thus you need not first use the  Add Ring icon and then the  Add feature function anymore.


Apagar Anél

The  Delete Ring tool allows you to delete ring polygons inside an existing area. This tool only works with polygon layers. It doesn't change anything when it is used on the outer ring of the polygon. This tool can be used on polygon and multi-polygon features. Before you select the vertices of a ring, adjust the vertex edit tolerance.

Apagar Parte

The  Delete Part tool allows you to delete parts from multifeatures (e.g., to delete polygons from a multi-polygon feature). It won't delete the last part of the feature; this last part will stay untouched. This tool works with all multi-part geometries: point, line and polygon. Before you select the vertices of a part, adjust the vertex edit tolerance.

Refazer elementos


You can reshape line and polygon features using the  Reshape Features icon on the toolbar. It replaces the line or polygon part from the first to the last intersection with the original line. With polygons, this can sometimes lead



to unintended results. It is mainly useful to replace smaller parts of a polygon, not for major overhauls, and the reshape line is not allowed to cross several polygon rings, as this would generate an invalid polygon.

Por exemplo, pode editar o limite de um polígono com esta ferramenta. Primeiro, clique na área interna do polígono junto do ponto onde pretende adicionar o novo vértice. Depois, atravesse o limite e adicione os vértices no exterior do polígono. Para finalizar, clique com o botão direito na área interna do polígono. A ferramenta vai adicionar um nó automaticamente no ponto onde a linha atravessa o limite. Também é possível remover parte da área do polígono, começa-se a nova linha no exterior do polígono, adicionam-se vértices no interior e termina-se a linha no exterior do polígono com um clique no botão direito.

Note: The reshape tool may alter the starting position of a polygon ring or a closed line. So, the point that is represented 'twice' will not be the same any more. This may not be a problem for most applications, but it is something to consider.


Curvas de Afastamento

The  Offset Curve tool creates parallel shifts of line layers. The tool can be applied to the edited layer (the geometries are modified) or also to background layers (in which case it creates copies of the lines / rings and adds them to the the edited layer). It is thus ideally suited for the creation of distance line layers. The displacement is shown at the bottom left of the taskbar.

To create a shift of a line layer, you must first go into editing mode and then select the feature. You can make the  Offset Curve tool active and drag the cross to the desired distance. Your changes may then be saved with the  Save Layer Edits tool.

QGIS options dialog (Digitizing tab then **Curve offset tools** section) allows you to configure some parameters like **Join style**, **Quadrant segments**, **Miter limit**.


Dividir Elementos

Pode dividir elementos usando o ícone  Dividir Elementos da barra de ferramentas. Apenas desenhe uma linha ao longo do elemento que quer dividir.



Split parts

In QGIS 2.0 it is now possible to split the parts of a multi part feature so that the number of parts is increased. Just draw a line across the part you want to split using the  Split Parts icon.


Juntar elementos seleccionados

The  Merge Selected Features tool allows you to merge features that have common boundaries. A new dialog will allow you to choose which value to choose between each selected features or select a function (Minimum, Maximum, Median, Sum, Skip Attribute) to use for each column.

Juntar os atributos dos elementos seleccionados

The  Merge Attributes of Selected Features tool allows you to merge attributes of features with common boundaries and attributes without merging their boundaries. First, select several features at once. Then press the  Merge Attributes of Selected Features button. Now QGIS asks you which attributes are to be applied to all selected objects. As a result, all selected objects have the same attribute entries.

Rodar Símbolos de Pontos

 Rotate Point Symbols allows you to change the rotation of point symbols in the map canvas. You must first define a rotation column from the attribute table of the point layer in the *Advanced* menu of the *Style* menu of the *Layer Properties*. Also, you will need to go into the ‘SVG marker’ and choose *Data defined properties ...*. Activate *Angle* and choose ‘rotation’ as field. Without these settings, the tool is inactive.

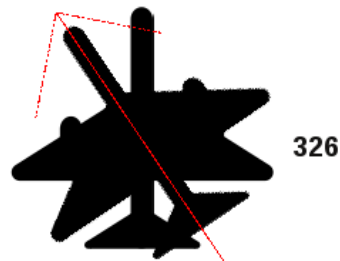



Figura 12.41: Rodar Símbolos de Pontos 


To change the rotation, select a point feature in the map canvas and rotate it, holding the left mouse button pressed. A red arrow with the rotation value will be visualized (see [Figure_edit_4](#)). When you release the left mouse button again, the value will be updated in the attribute table.

Note: Se segurar a tecla `Ctrl` premido, a rotação irá ser feita em passos de 15 graus.





12.5.6 Criando novas camadas Vectoriais

QGIS allows you to create new shapefile layers, new SpatialLite layers, and new GPX layers. Creation of a new GRASS layer is supported within the GRASS plugin. Please refer to section [Criando uma nova camada vectorial GRASS](#) for more information on creating GRASS vector layers.

Criando uma nova camada Shapefile

To create a new shape layer for editing, choose *New* →  *New Shapefile Layer...* from the *Layer* menu. The *New Vector Layer* dialog will be displayed as shown in [Figure_edit_5](#). Choose the type of layer (point, line or polygon) and the CRS (coordinate reference system).

Note that QGIS does not yet support creation of 2.5D features (i.e., features with X,Y,Z coordinates).

To complete the creation of the new shapefile layer, add the desired attributes by clicking on the [**Add to attributes list**] button and specifying a name and type for the attribute. A first ‘id’ column is added as default but can be removed, if not wanted. Only *Type: real* , *Type: integer* , *Type: string*  and *Type: date*  attributes are supported. Additionally and according to the attribute type, you can also define the width and precision of the new attribute column. Once you are happy with the attributes, click [**OK**] and provide a name for the shapefile. QGIS will automatically add a `.shp` extension to the name you specify. Once the layer has been created, it will be added to the map, and you can edit it in the same way as described in section [Digitalizar uma camada existente](#) above.

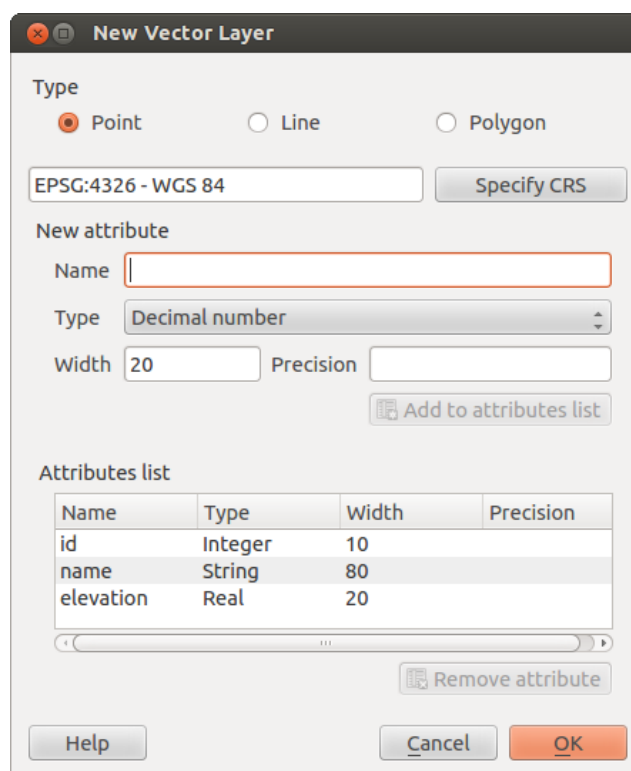




Figura 12.42: Janela de criação de uma nova camada Shapefile 🐧

Criando uma nova camada SpatialLite


Para criar uma nova camada SpatialLite para edição, escolha *Novo* →  *Nova Camada SpatialLite...* do menu *Camada*. O diálogo *Nova Camada SpatialLite* irá aparecer como é mostrado na [Figure_edit_6](#).


The first step is to select an existing SpatialLite database or to create a new SpatialLite database. This can be done with the browse button  to the right of the database field. Then, add a name for the new layer, define the layer type, and specify the coordinate reference system with [**Specify CRS**]. If desired, you can select *Create an autoincrementing primary key*.

To define an attribute table for the new SpatialLite layer, add the names of the attribute columns you want to create with the corresponding column type, and click on the [**Add to attribute list**] button. Once you are happy with the attributes, click [**OK**]. QGIS will automatically add the new layer to the legend, and you can edit it in the same way as described in section [Digitalizar uma camada existente](#) above.

Further management of SpatialLite layers can be done with the DB Manager. See [Módulo Gestor BD](#).

Criando uma nova camada GPX

To create a new GPX file, you need to load the GPS plugin first. *Plugins* →  *Plugin Manager...* opens the Plugin Manager Dialog. Activate the *GPS Tools* checkbox.

When this plugin is loaded, choose *New* →  *Create new GPX Layer...* from the *Layer* menu. In the *Save new GPX file as* dialog, you can choose where to save the new GPX layer.

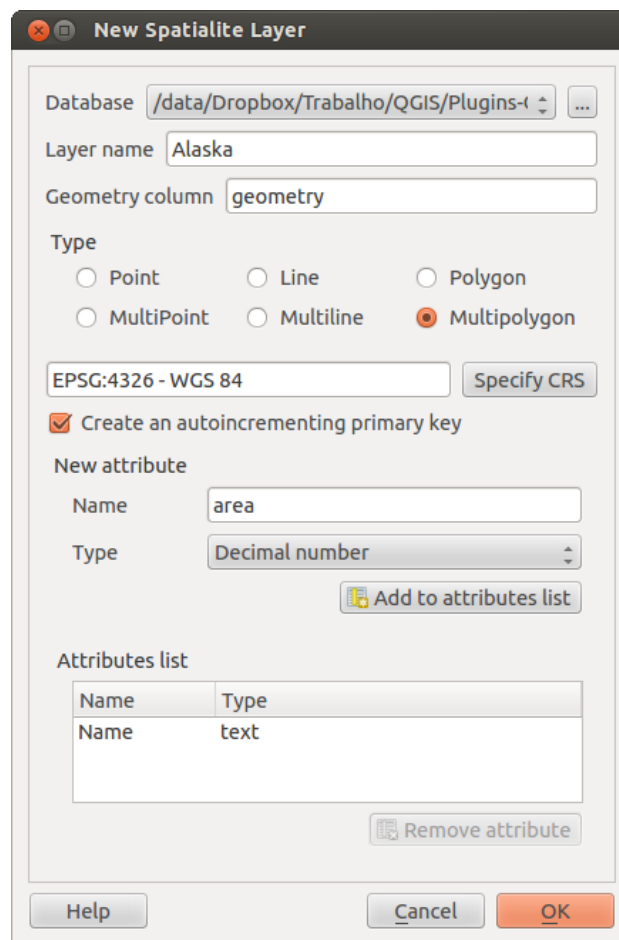






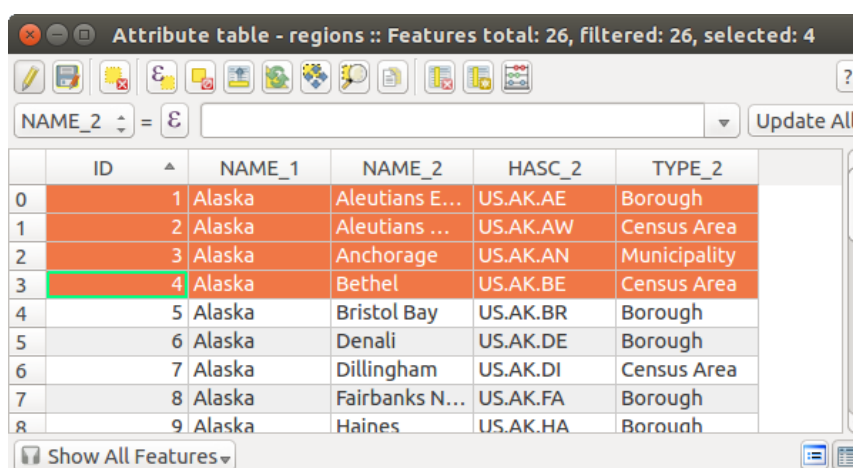
Figura 12.43: Janela de criação de uma nova camada SpatialLite 

12.5.7 Trabalhando com a Tabela de Atributos


The attribute table displays features of a selected layer. Each row in the table represents one map feature, and each column contains a particular piece of information about the feature. Features in the table can be searched, selected, moved or even edited.

To open the attribute table for a vector layer, make the layer active by clicking on it in the map legend area. Then, from the main *Layer* menu, choose  *Open Attribute Table*. It is also possible to right click on the layer and choose  *Open Attribute Table* from the drop-down menu, and to click on the  *Open Attribute Table* button in the Attributes toolbar.

This will open a new window that displays the feature attributes for the layer (*figure_attributes_1*). The number of features and the number of selected features are shown in the attribute table title.



	ID	NAME_1	NAME_2	HASC_2	TYPE_2
0	1	Alaska	Aleutians E...	US.AK.AE	Borough
1	2	Alaska	Aleutians ...	US.AK.AW	Census Area
2	3	Alaska	Anchorage	US.AK.AN	Municipality
3	4	Alaska	Bethel	US.AK.BE	Census Area
4	5	Alaska	Bristol Bay	US.AK.BR	Borough
5	6	Alaska	Denali	US.AK.DE	Borough
6	7	Alaska	Dillingham	US.AK.DI	Census Area
7	8	Alaska	Fairbanks N...	US.AK.FA	Borough
8	9	Alaska	Haines	US.AK.HA	Borough

Figura 12.44: Tabela de Atributos para a camada regiões 



Seleccionando elementos na tabela de atributos


Cada linha selecionada na tabela de atributos representa os atributos de um determinado elemento da camada. Se o conjunto de elementos selecionados na janela principal é alterado, a seleção também é atualizada na tabela de atributos. Da mesma forma, se um conjunto de linhas selecionadas na tabela de atributos é modificada, o conjunto de elementos selecionado na janela principal será atualizado.

As linhas podem ser selecionadas ao clicar no número da linha, do lado esquerdo desta. Podem ser marcadas **Linhas múltiplas** manter a tecla *Ctrl* primida. Pode ser feita uma **seleção contínua** se manter primida a tecla *Shift* e clicar na entrada de várias linha, do lado esquerdo. Todas as linhas entre a posição atual do cursor e a linha clicada serão selecionados. Ao mover a posição do cursor na tabela de atributos, ao clicar um célula na tabela, não tem influência na seleção de linhas. Modificar a seleção na tela principal não altera a posição do cursor na tabela de atributos.

A tabela pode ser ordenada por qualquer coluna, clicando no cabeçalho da coluna. Uma pequena seta indica a forma de ordenação (apontar para baixo significa valores descendentes do topo da linha, apontar para cima significa valores descendentes do topo da linha).

For a **simple search by attributes** on only one column, choose the *Column filter* → from the menu in the bottom left corner. Select the field (column) on which the search should be performed from the drop-down menu, and hit the **[Apply]** button. Then, only the matching features are shown in the attribute table.

To make a selection, you have to use the  *Select features using an Expression* icon on top of the attribute table. 

Select features using an Expression allows you to define a subset of a table using a *Function List* like in the  *Field Calculator* (see *Calculadora de Campos*). The query result can then be saved as a new vector layer. For example, if you want to find regions that are boroughs from *regions.shp* of the QGIS sample data, you have to open the *Fields and*

Values menu and choose the field that you want to query. Double-click the field 'TYPE_2' and also **[Load all unique values]** . From the list, choose and double-click 'Borough'. In the *Expression* field, the following query appears:













```
"TYPE_2" = 'Borough'
```


Here you can also use the *Function list* → *Recent (Selection)* to make a selection that you used before. The expression builder remembers the last 20 used expressions.

The matching rows will be selected, and the total number of matching rows will appear in the title bar of the attribute table, as well as in the status bar of the main window. For searches that display only selected features on the map, use the Query Builder described in section *Ferramenta de Consulta*.



To show selected records only, use *Show Selected Features* from the menu at the bottom left.

The other buttons at the top of the attribute table window provide the following functionality:


-  Toggle editing mode to edit single values and to enable functionalities described below (also with Ctrl+E)
-  Save Edits (also with Ctrl+S)
-  Unselect all (also with Ctrl+U)
-  Move selected to top (also with Ctrl+T)
-  Invert selection (also with Ctrl+R)
-  Copy selected rows to clipboard (also with Ctrl+C)
-  Zoom map to the selected rows (also with Ctrl+J)
-  Pan map to the selected rows (also with Ctrl+P)
-  Delete selected features (also with Ctrl+D)
-  New Column for PostGIS layers and for OGR layers with GDAL version >= 1.6 (also with Ctrl+W)
-  Delete Column for PostGIS layers and for OGR layers with GDAL version >= 1.9 (also with Ctrl+L)
-  Open field calculator (also with Ctrl+I)

Below these buttons is the Field Calculator bar, which allows calculations to be quickly applied attributes visible in the table. This bar uses the same expressions as the  Field Calculator (see *Calculadora de Campos*).

Tip: Ignorar geometria WKT

If you want to use attribute data in external programs (such as Excel), use the  Copy selected rows to clipboard button. You can copy the information without vector geometries if you deactivate *Settings* → *Options* → *Data sources* menu  *Copy geometry in WKT representation from attribute table*.

Guardar elementos seleccionados como nova camada


The selected features can be saved as any OGR-supported vector format and also transformed into another coordinate reference system (CRS). Just open the right mouse menu of the layer and click on *Save as* to define the name of the output file, its format and CRS (see section *Legenda do Mapa*). To save the selection ensure that the  *Save only selected features* is selected. It is also possible to specify OGR creation options within the dialog.

Paste into new layer

Features that are on the clipboard may be pasted into a new layer. To do this, first make a layer editable. Select some features, copy them to the clipboard, and then paste them into a new layer using *Edit* → *Paste Features as* and choosing *New vector layer* or *New memory layer*.

This applies to features selected and copied within QGIS and also to features from another source defined using well-known text (WKT).

Trabalhando com tabelas de atributos não espaciais

QGIS allows you also to load non-spatial tables. This currently includes tables supported by OGR and delimited text, as well as the PostgreSQL, MSSQL and Oracle provider. The tables can be used for field lookups or just generally browsed and edited using the table view. When you load the table, you will see it in the legend field. It can be opened with the  Open Attribute Table tool and is then editable like any other layer attribute table.

As an example, you can use columns of the non-spatial table to define attribute values, or a range of values that are allowed, to be added to a specific vector layer during digitizing. Have a closer look at the edit widget in section *Menu Campos* to find out more.

12.5.8 Creating one to many relations


Relations are a technique often used in databases. The concept is, that features (rows) of different layers (tables) can belong to each other.

As an example you have a layer with all regions of alaska (polygon) which provides some attributes about its name and region type and a unique id (which acts as primary key).

Foreign keys

Then you get another point layer or table with information about airports that are located in the regions and you also want to keep track of these. If you want to add them to the region layer, you need to create a one to many relation using foreign keys, because there are several airports in most regions.



Figura 12.45: Alaska region with airports 

In addition to the already existing attributes in the airports attribute table another field `fk_region` which acts as a foreign key (if you have a database, you will probably want to define a constraint on it).

This field `fk_region` will always contain an id of a region. It can be seen like a pointer to the region it belongs to. And you can design a custom edit form for the editing and QGIS takes care about the setup. It works with different providers (so you can also use it with shape and csv files) and all you have to do is to tell QGIS the relations between your tables.

Layers

QGIS makes no difference between a table and a vector layer. Basically, a vector layer is a table with a geometry. So can add your table as a vector layer. To demonstrate you can load the 'region' shapefile (with geometries) and the 'airport' csv table (without geometries) and a foreign key (`fk_region`) to the layer region. This means, that each airport belongs to exactly one region while each region can have any number of airports (a typical one to many relation).

Definition (Relation Manager)

The first thing we are going to do is to let QGIS know about the relations between the layer. This is done in *Settings* → *Project Properties*. Open the *Relations* menu and click on *Add*.

- **name** is going to be used as a title. It should be a human readable string, describing, what the relation is used for. We will just call say "Airports" in this case.
- **referencing layer** is the one with the foreign key field on it. In our case this is the airports layer
- **referencing field** will say, which field points to the other layer so this is `fk_region` in this case
- **referenced layer** is the one with the primary key, pointed to, so here it is the regions layer
- **referenced field** is the primary key of the referenced layer so it is `ID`
- **id** will be used for internal purposes and has to be unique. You may need it to build custom forms once this is supported. If you leave it empty, one will be generated for you but you can assign one yourself to get one that is easier to handle.

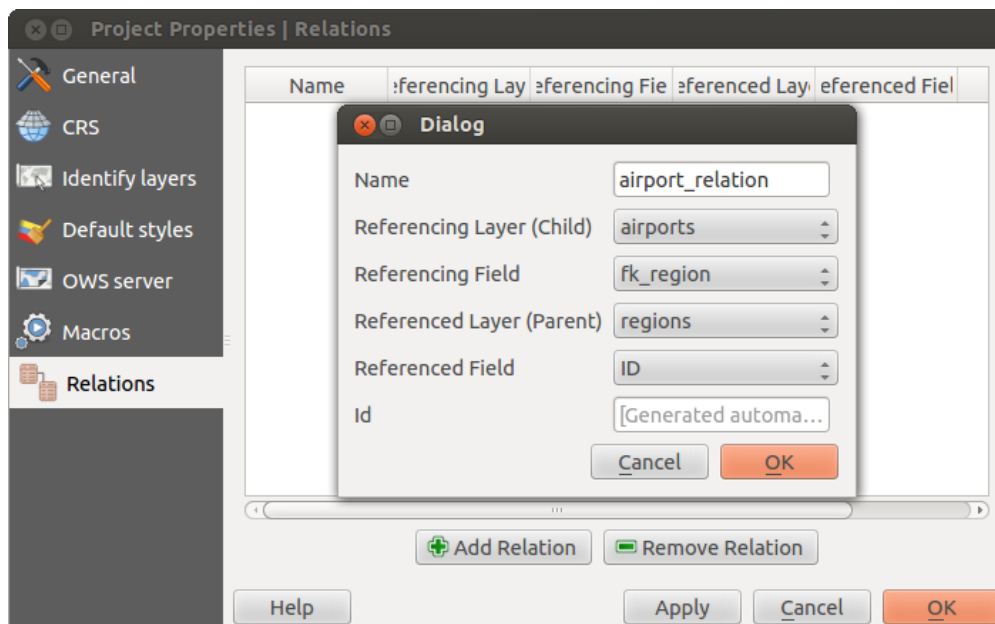


Figura 12.46: Relation Manager 

Forms

Now that QGIS knows about the relation, it will be used to improve the forms it generates. As we did not change the default form method (autogenerated) it will just add a new widget in our form. So let's select the layer region in the legend and use the identify tool. Depending on your settings, the form might open directly or you will have to choose to open it in the identification dialog under actions.

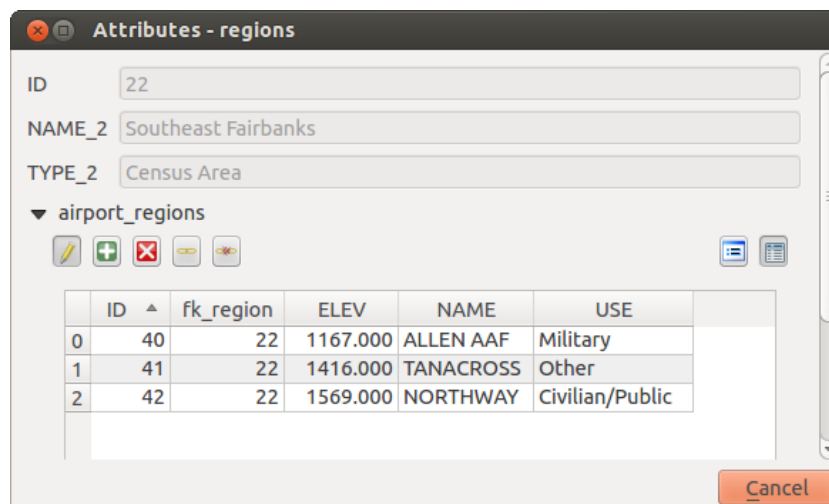







Figura 12.47: Identification dialog regions with relation to airports 🐧

As you can see, the airports assigned to this particular region are all shown in a table. And there are also some buttons available. Let's review them shortly

- The  button is for toggling the edit mode. Be aware that it toggles the edit mode of the airport layer, although we are in the feature form of a feature from the region layer. But the table is representing features of the airport layer.
- The  button will add a new feature to the airport layer. And it will assign the new airport to the current region by default.
- The  button will delete the selected airport permanently.
- The  symbol will open a new dialog where you can select any existing airport which will then be assigned to the current region. This may be handy if you created the airport on the wrong region by accident.
- The  symbol will unlink the selected airport from the current region, leaving them unassigned (the foreign key is set to NULL) effectively.
- The two buttons to the right switch between table view and form view where the later let's you view all the airports in their respective form.

If you work on the airport table, a new widget type is available which lets you embed the feature form of the referenced region on the feature form of the airports. It can be used when you open the layer properties of the airports table, switch to the *Fields* menu and change the widget type of the foreign key field 'fk_region' to Relation Reference.

If you look at the feature dialog now, you will see, that the form of the region is embedded inside the airports form and will even have a combobox, which allows you to assign the current airport to another region.

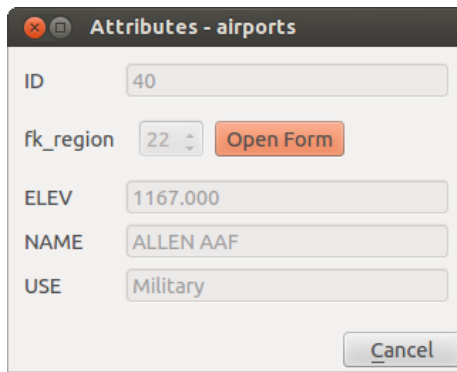



Figura 12.48: Identification dialog airport with relation to regions 

12.6 Ferramenta de Consulta

The Query Builder allows you to define a subset of a table using a SQL-like WHERE clause and to display the result in the main window. The query result can then be saved as a new vector layer.

12.6.1 Consulta

Open the **Query Builder** by opening the Layer Properties and going to the *General* menu. Under *Feature subset*, click on the **[Query Builder]** button to open the *Query builder*. For example, if you have a *regions* layer with a *TYPE_2* field, you could select only regions that are *borough* in the *Provider specific filter expression* box of the Query Builder. [Figure_attributes_2](#) shows an example of the Query Builder populated with the *regions.shp* layer from the QGIS sample data. The Fields, Values and Operators sections help you to construct the SQL-like query.

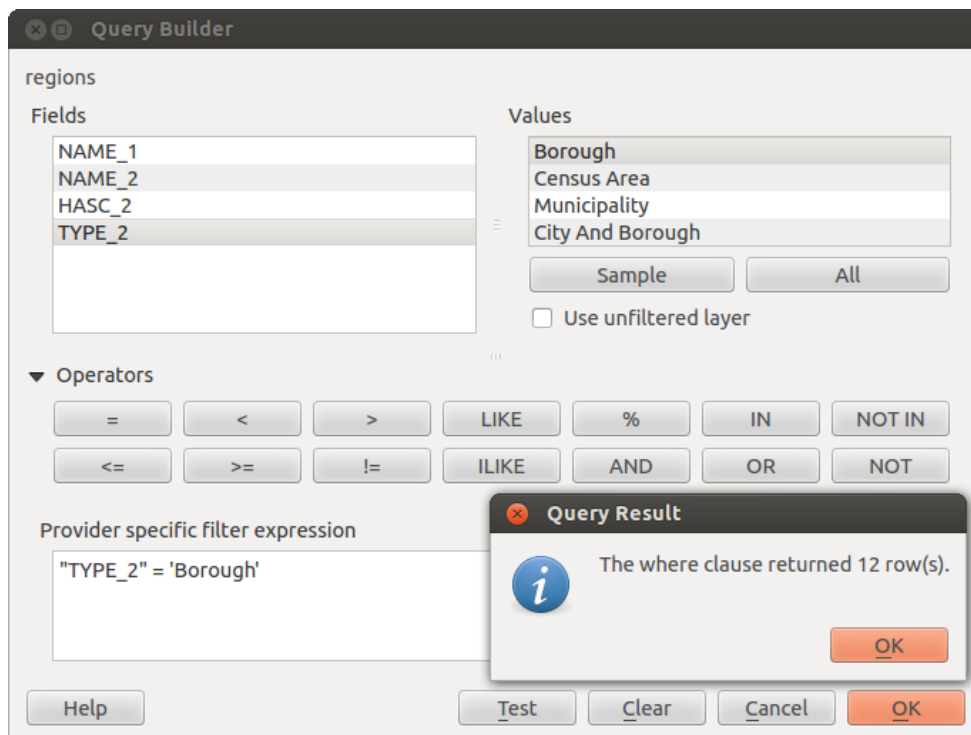


Figura 12.49: Ferramenta de Consulta

The **Fields list** contains all attribute columns of the attribute table to be searched. To add an attribute column to

the SQL WHERE clause field, double click its name in the Fields list. Generally, you can use the various fields, values and operators to construct the query, or you can just type it into the SQL box.

The **Values list** lists the values of an attribute table. To list all possible values of an attribute, select the attribute in the Fields list and click the **[all]** button. To list the first 25 unique values of an attribute column, select the attribute column in the Fields list and click the **[Sample]** button. To add a value to the SQL WHERE clause field, double click its name in the Values list.


The **Operators section** contains all usable operators. To add an operator to the SQL WHERE clause field, click the appropriate button. Relational operators ($=$, $>$, $<$, ...), string comparison operator (LIKE), and logical operators (AND, OR, ...) are available.

The **[Test]** button shows a message box with the number of features satisfying the current query, which is useful in the process of query construction. The **[Clear]** button clears the text in the SQL WHERE clause text field. The **[OK]** button closes the window and selects the features satisfying the query. The **[Cancel]** button closes the window without changing the current selection.

QGIS treats the resulting subset acts as if it where the entire layer. For example if you applied the filter above for 'Borough', you can not display, query, save or edit Anorage, because that is a 'Manicpality' and therefore not part of the subset.

The only exception is that unless your layer is part of a database, using a subset will prevent you from editing the layer.

12.7 Calculadora de Campos

The  Field Calculator button in the attribute table allows you to perform calculations on the basis of existing attribute values or defined functions, for instance, to calculate length or area of geometry features. The results can be written to a new attribute field, a virtual field, or they can be used to update values in an existing field.

Tip: Virtual Fields

- Virtual fields are not permanent and are not saved.
 - To make a field virtual it must be done when the field is made.
-

The field calculator is now available on any layer that supports edit. When you click on the field calculator icon the dialog opens (see [figure_attributes_3](#)). If the layer is not in edit mode, a warning is displayed and using the field calculator will cause the layer to be put in edit mode before the calculation is made.




The quick field calculation bar in top of the attribute table is only visible if the layer is editable.

In quick field calculation bar, you first select the existing field name then open the expression dialog to create your expression or write it directly in the field then click on **Update All** button.

In the field calculator dialog, you first must select whether you want to only update selected features, create a new attribute field where the results of the calculation will be added or update an existing field.

Se optar por adicionar um novo campo, você precisa digitar um nome de campo, o tipo de campo (inteiro, real ou texto), a largura total do campo, e a precisão do campo (veja [figure_attributes_3](#)). Por exemplo, se você escolher uma largura de campo de 10 e uma precisão de 3, significa que você tem 6 números antes do ponto, então o ponto e mais 3 indicações para a precisão.

A short example illustrates how the field calculator works. We want to calculate the length in km of the `railroads` layer from the QGIS sample dataset:

1. Carregue a Shapefile `railroads.shp` no QGIS e pressione  `Abrir Tabela de Atributos`.
2. Clique no  `Alternar o modo de edição` e abra a janela da  `Calculadora de Campos`.

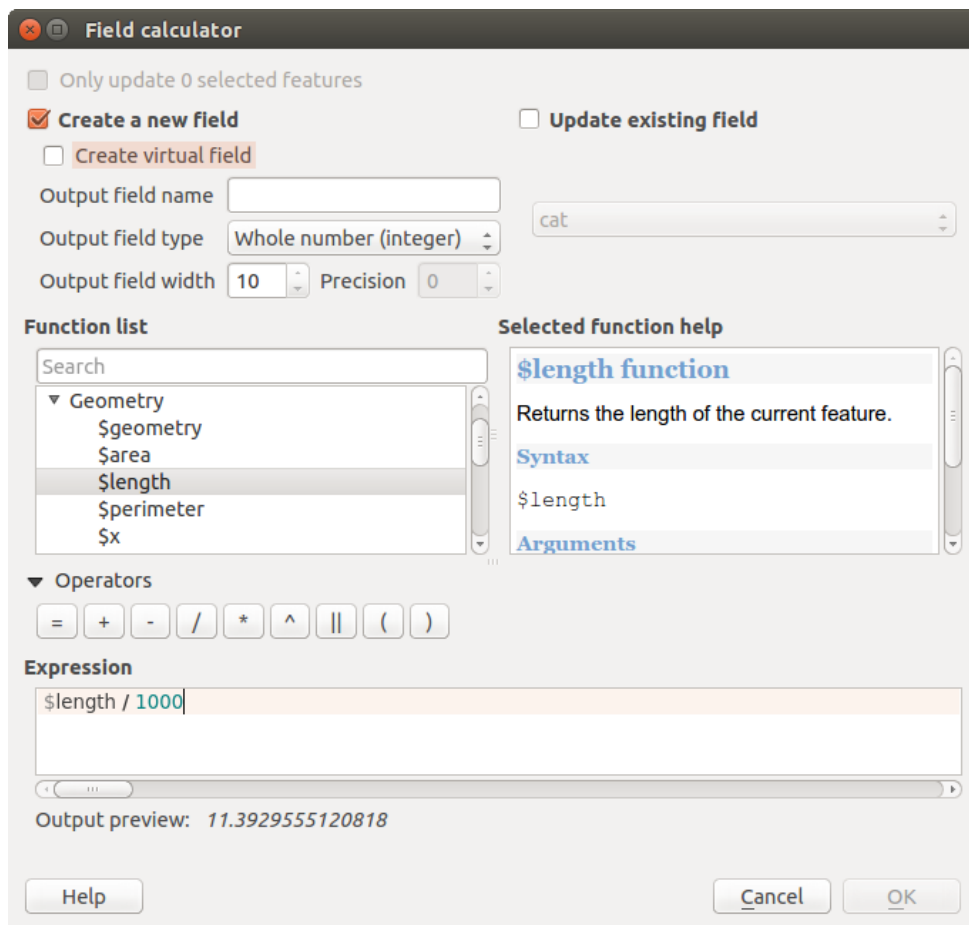



Figura 12.50: Calculadora de Campos

3. Selecciona a caixa de verificação  *Criar novo campo* para guardar os cálculos no novo campo.
4. Adicione comprimento como campo de saída do nome, “real como o tipo de campo de saída e defina o campo de comprimento de saída de 10 com Precisão 3.
5. agora faça duplo clique na função \$length no grupo :guilabel:‘Geometria’ para adicionar à caixa de expressões da Calculadora de campos.
6. Complete a expressão introduzindo “/ 1000” na caixa de expressões da Calculadora de campos e clique [OK].
7. You can now find a new field length in the attribute table.

The available functions are listed in [Expressions](#) chapter.

Trabalhando com Informação Matricial

13.1 A trabalhar com Dados Matriciais

This section describes how to visualize and set raster layer properties. QGIS uses the GDAL library to read and write raster data formats, including ArcInfo Binary Grid, ArcInfo ASCII Grid, GeoTIFF, ERDAS IMAGINE, and many more. GRASS raster support is supplied by a native QGIS data provider plugin. The raster data can also be loaded in read mode from zip and gzip archives into QGIS.

As of the date of this document, more than 100 raster formats are supported by the GDAL library (see GDAL-SOFTWARE-SUITE in *Literatura e Referências Web*). A complete list is available at http://www.gdal.org/formats_list.html.

Note: Not all of the listed formats may work in QGIS for various reasons. For example, some require external commercial libraries, or the GDAL installation of your OS may not have been built to support the format you want to use. Only those formats that have been well tested will appear in the list of file types when loading a raster into QGIS. Other untested formats can be loaded by selecting the [GDAL] All files (*) filter.

A utilização de dados raster com GRASS está descrita na secção *Integração GRASS SIG*.

13.1.1 O que são dados raster?

Raster data in GIS are matrices of discrete cells that represent features on, above or below the earth's surface. Each cell in the raster grid is the same size, and cells are usually rectangular (in QGIS they will always be rectangular). Typical raster datasets include remote sensing data, such as aerial photography, or satellite imagery and modelled data, such as an elevation matrix.

Unlike vector data, raster data typically do not have an associated database record for each cell. They are geocoded by pixel resolution and the *x/y* coordinate of a corner pixel of the raster layer. This allows QGIS to position the data correctly in the map canvas.

QGIS makes use of georeference information inside the raster layer (e.g., GeoTiff) or in an appropriate world file to properly display the data.

13.1.2 Carregar dados raster no QGIS

Raster layers are loaded either by clicking on the  Add Raster Layer icon or by selecting the *Layer* →  Add Raster Layer menu option. More than one layer can be loaded at the same time by holding down the **Ctrl** or **Shift** key and clicking on multiple items in the *Open a GDAL Supported Raster Data Source* dialog.

Once a raster layer is loaded in the map legend, you can click on the layer name with the right mouse button to select and activate layer-specific features or to open a dialog to set raster properties for the layer.

Opções do botão direito do rato para camadas raster

- *Aproximar à Extensão da Camada*
- *Ampliar à Melhor Escala (100%)*
- *Esticar Usando o Enquadramento Actual*
- *Adicionar ao Enquadramento*
- *Remover*
- *Duplicar*
- *Definir SRC da Camada*
- *Definir SRC do projecto a partir da Camada*
- *Guardar como ...*
- *Propriedades*
- *Renomear*
- *Copiar Estilo*
- *Adicionar Novo Grupo*
- *Expandir Tudo*
- *Ocultar Tudo*
- *Actualizar Ordem de Desenho*

13.2 Janela de Propriedades do Projecto

Para visualizar e definir as propriedades da camada raster, dê um duplo clique no nome da camada na legenda do mapa, ou clique com o botão direito no nome da camada e escolha: *Propriedades* a partir do menu de contexto. Irá abrir o diálogo *Propriedades da Camada Raster* (see [figure_raster_1](#)).

Existem vários menus na caixa de diálogo:

- Separador: *Geral*
- *Gestor de Estilo*
- *Transparência*
- *Piramides*
- *Histograma*
- *Metadados*

13.2.1 Menu Geral

Informação da camada

O menu *General* apresenta informações básicas sobre a imagem seleccionada, incluindo o caminho da origem da camada, o nome de exibição na legenda (que pode ser modificado), e o número de colunas, linhas e valores nulos do raster.

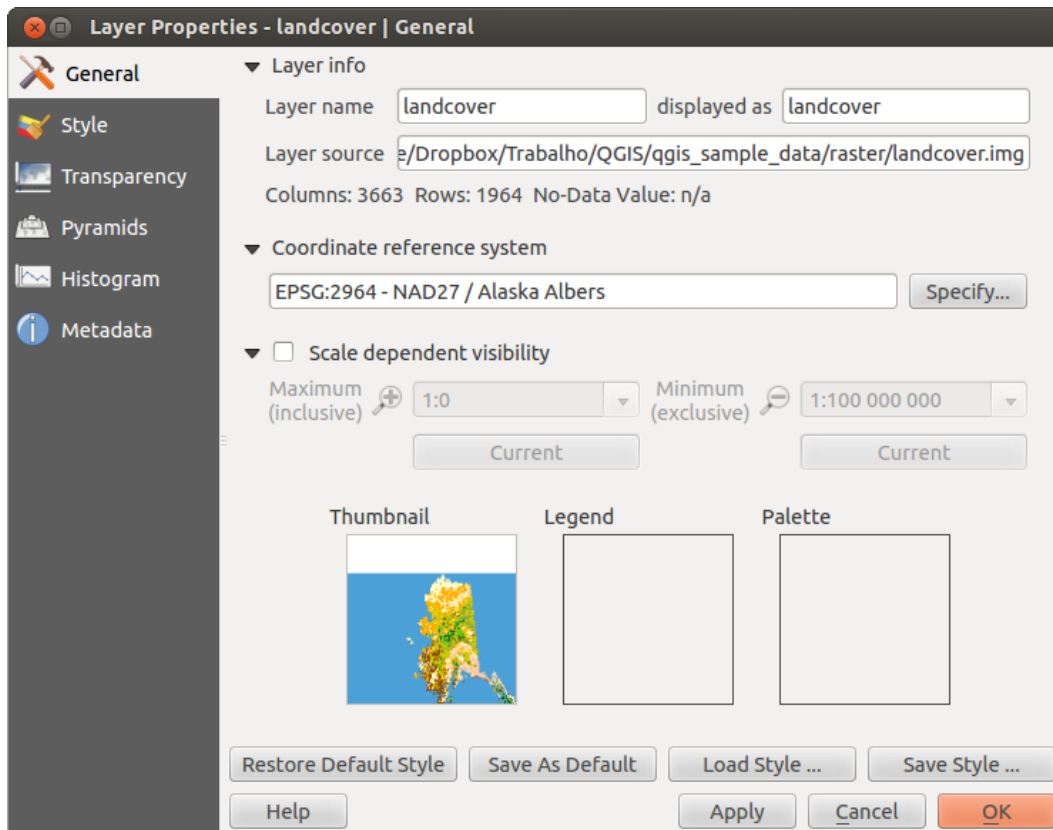


Figura 13.1: Janela das Propriedades da Camada Raster 🐧

Sistema de Referência de Coordenadas

Aqui, pode encontrar a informação do sistema de referência de coordenadas (SRC) impressos numa sequência PROJ.4. Se essa configuração não estiver correta, ele pode ser modificada clicando no botão **[Specify]**.

Escala dependente da visibilidade

Além disso a visibilidade dependente da escala pode ser vista neste guia. Você terá que verificar a caixa de seleção e definir uma escala adequada, onde seus dados serão exibidos na tela do mapa.

Na parte inferior, pode ver uma miniatura da camada, a simbologia da legenda e a paleta.

13.2.2 Estilos

Renderizar banda

QGIS Oferece quatro diferentes: *guiabel:Tipos de renderização*. A renderização a escolher depende do tipo de dados.

1. Multibanda cor - se o arquivo vem como multibanda, com várias bandas (e.g., usado para imagens de satélite com várias bandas)
2. Paleta - se o ficheiro de banda simples vem com a paleta indexada (e.g., usado em mapas topográficos digitais)
3. A banda simples cinza - (one band of) a imagem será processada como cinza; QGIS vai escolher esse representante, se o arquivo não tem nem multibandas nem uma paleta indexada nem uma paleta contínua (por exemplo, usado no mapa de relevo sombreado)

4. Banda Simples de Pseudocor - é possível a renderização de ficheiros com uma paleta continua ou de cor (e.g., usada num mapa de altitude)

Cor multibanda

Com o renderizador da cor multibanda, as três bandas da imagem pode ser renderizada, pela banda que representa o componente vermelho, verde ou azul, que será usado para criar uma imagem colorida. Pode escolher vários: *Contrast enhancement methods*: ‘No enhancement’, ‘Stretch to MinMax’, ‘Stretch and clip to MinMax’ and ‘Clip to min max’.

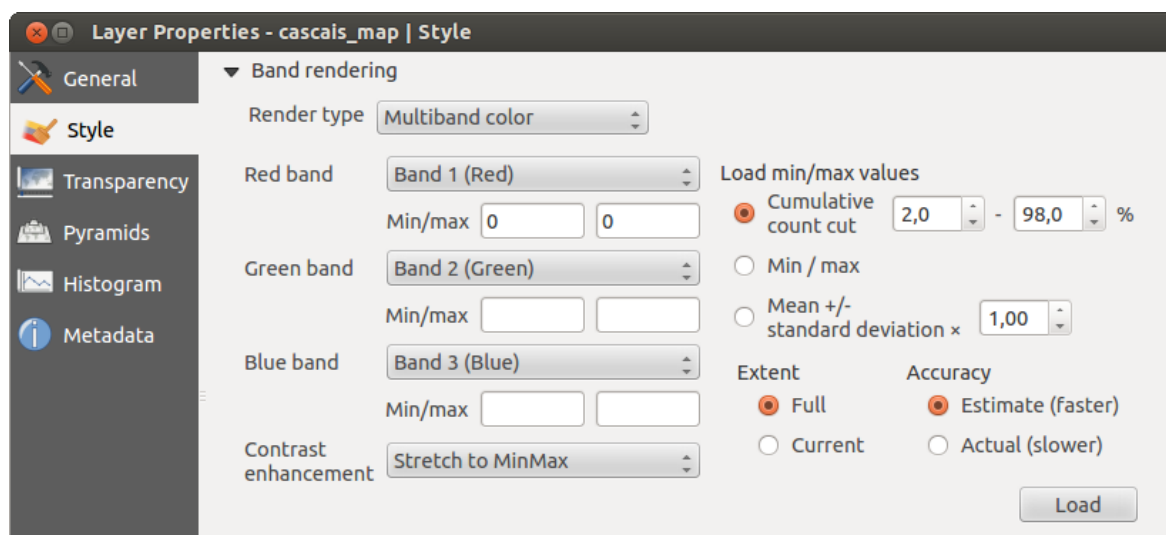


Figura 13.2: Renderização Raster - Cor Multibanda 

Esta seleção oferece-lhe uma ampla gama de opções para modificar a aparência da sua camada. Em primeiro lugar, tem que ter o intervalo de dados a partir da sua imagem. Isto pode ser feito escolhendo a *Extent* and pressing [Load]. QGIS can *Estimate (faster)* the *Min* and *Max* values of the bands or use the *Actual (slower)* *Accuracy*.

Now you can scale the colors with the help of the *Load min/max values* section. A lot of images have a few very low and high data. These outliers can be eliminated using the *Cumulative count cut* setting. The standard data range is set from 2% to 98% of the data values and can be adapted manually. With this setting, the gray character of the image can disappear. With the scaling option *Min/max*, QGIS creates a color table with all of the data included in the original image (e.g., QGIS creates a color table with 256 values, given the fact that you have 8 bit bands). You can also calculate your color table using the *Mean +/- standard deviation x* . Then, only the values within the standard deviation or within multiple standard deviations are considered for the color table. This is useful when you have one or two cells with abnormally high values in a raster grid that are having a negative impact on the rendering of the raster.

Todos os cálculos podem ser feitos através da *Atual* extensão.

Tip: Visualização de uma Banda Simples e Multibanda raster

Se quiser ver uma única banda de uma imagem multibanda (por exemplo, vermelho), pode pensar que iria definir o verde e faixas azuis para “Not Set”. Mas esta não é a maneira correta. Para apresentar a banda vermelha, defina o tipo de imagem para ‘Banda simples cinza’, em seguida, selecione vermelha como a banda para usar a Cinza.

Paletizada

This is the standard render option for singleband files that already include a color table, where each pixel value is assigned to a certain color. In that case, the palette is rendered automatically. If you want to change colors assigned to certain values, just double-click on the color and the *Select color* dialog appears. Also, in QGIS 2.2. it’s now possible to assign a label to the color values. The label appears in the legend of the raster layer then.

Melhorar contraste

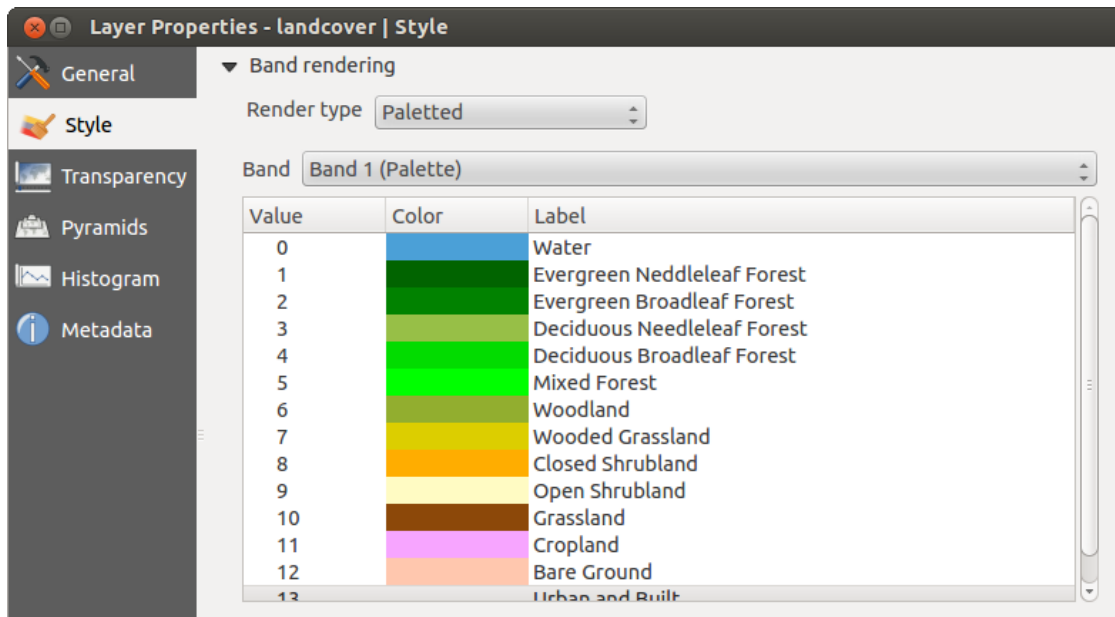


Figura 13.3: Renderização Raster - Paletizada 🐧

Note: When adding GRASS rasters, the option *Contrast enhancement* will always be set automatically to *stretch to min max*, regardless of if this is set to another value in the QGIS general options.

Banda cinza simples

This renderer allows you to render a single band layer with a *Color gradient*: 'Black to white' or 'White to black'. You can define a *Min* and a *Max* value by choosing the *Extent* first and then pressing [Load]. QGIS can *Estimate (faster)* the *Min* and *Max* values of the bands or use the *Actual (slower)* Accuracy.

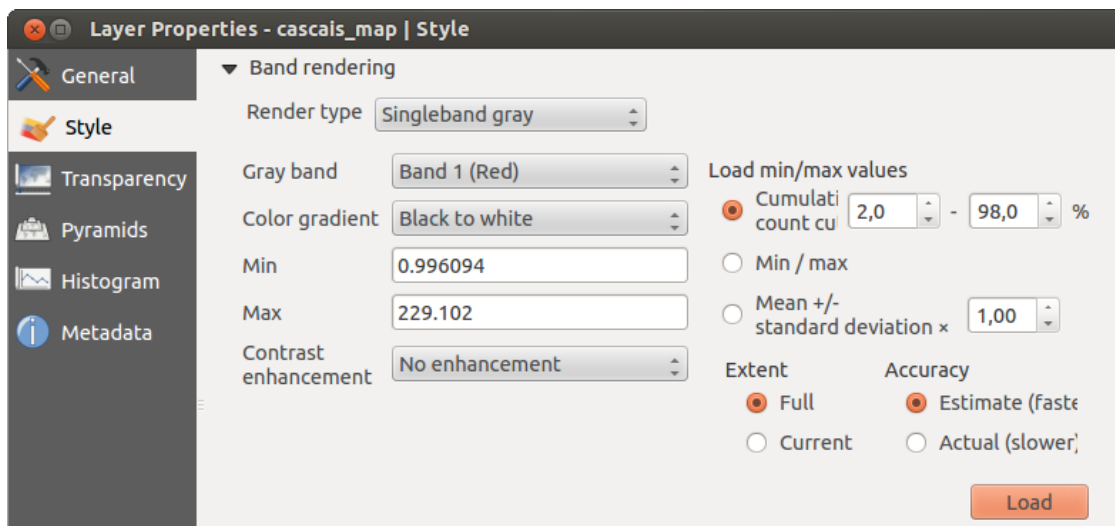


Figura 13.4: Renderização Raster - Banda simples cinza 🐧

With the *Load min/max values* section, scaling of the color table is possible. Outliers can be eliminated using the *Cumulative count cut* setting. The standard data range is set from 2% to 98% of the data values and can be adapted manually. With this setting, the gray character of the image can disappear. Further settings can be made with *Min/max* and *Mean +/- standard deviation x 1,00*. While the first one creates a color table with all of the data included in the original image, the second creates a color table that only considers values within

the standard deviation or within multiple standard deviations. This is useful when you have one or two cells with abnormally high values in a raster grid that are having a negative impact on the rendering of the raster.

Banda de cor falsa simples

Esta é uma opção de representação para arquivos de uma banda, que incluem um mapa de cores contínuo. Aqui também se pode criar mapas de cores individuais para uma banda. Três tipos de interpolação de cores estão

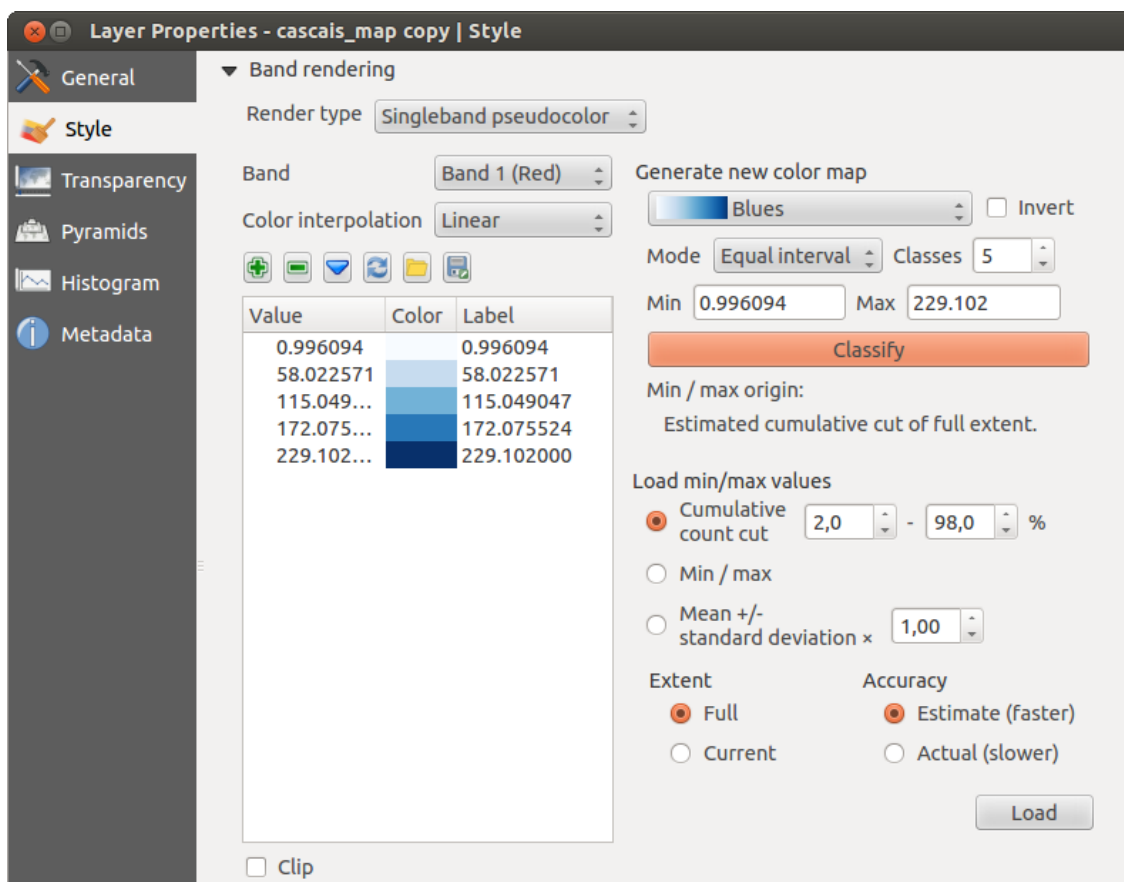











Figura 13.5: Renderização Raster - Banda simples pseudocor 

disponíveis:

1. Discreto
2. Linear
3. Exacto

In the left block, the button  Add values manually adds a value to the individual color table. The button  Remove selected row deletes a value from the individual color table, and the  Sort colormap items button sorts the color table according to the pixel values in the value column. Double clicking on the value column lets you insert a specific value. Double clicking on the color column opens the dialog *Change color*, where you can select a color to apply on that value. Further, you can also add labels for each color, but this value won't be displayed when you use the identify feature tool. You can also click on the button  Load color map from band, which tries to load the table from the band (if it has any). And you can use the buttons  Load color map from file or  Export color map to file to load an existing color table or to save the defined color table for other sessions.

In the right block, *Generate new color map* allows you to create newly categorized color maps. For the *Classification mode*  'Equal interval', you only need to select the *number of classes*  and press the button *Classify*. You can invert the colors of the color map by clicking the *Invert* checkbox. In the case of the *Mode*  'Continuous', QGIS creates classes automatically depending on the *Min* and *Max*. Defining *Min/Max* values

can be done with the help of the *Load min/max values* section. A lot of images have a few very low and high data. These outliers can be eliminated using the *Cumulative count cut* setting. The standard data range is set from 2% to 98% of the data values and can be adapted manually. With this setting, the gray character of the image can disappear. With the scaling option *Min/max*, QGIS creates a color table with all of the data included in the original image (e.g., QGIS creates a color table with 256 values, given the fact that you have 8 bit bands). You can also calculate your color table using the *Mean +/- standard deviation x 1,00*. Then, only the values within the standard deviation or within multiple standard deviations are considered for the color table.

Renderização Cor

For every *Band rendering*, a *Color rendering* is possible.

You can also achieve special rendering effects for your raster file(s) using one of the blending modes (see *Janela das Propriedades da Camada Vectorial*).

Further settings can be made in modifying the *Brightness*, the *Saturation* and the *Contrast*. You can also use a *Grayscale* option, where you can choose between ‘By lightness’, ‘By luminosity’ and ‘By average’. For one hue in the color table, you can modify the ‘Strength’.

Reamostragem

The *Resampling* option makes its appearance when you zoom in and out of an image. Resampling modes can optimize the appearance of the map. They calculate a new gray value matrix through a geometric transformation.

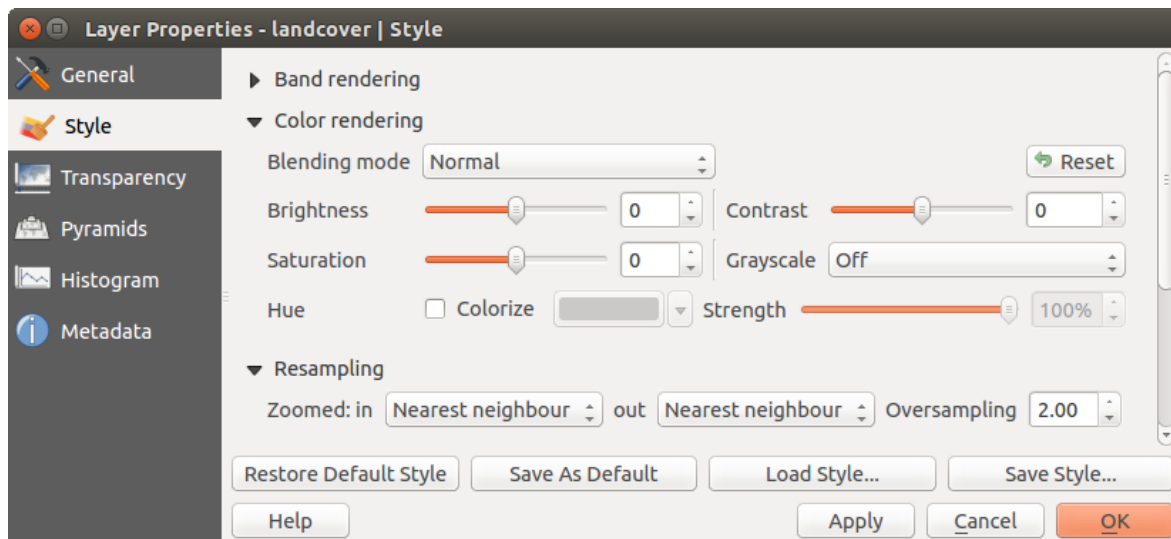



Figura 13.6: Renderização Raster - Reamostragem 

When applying the ‘Nearest neighbour’ method, the map can have a pixelated structure when zooming in. This appearance can be improved by using the ‘Bilinear’ or ‘Cubic’ method, which cause sharp features to be blurred. The effect is a smoother image. This method can be applied, for instance, to digital topographic raster maps.


13.2.3 Menu Transparência

QGIS has the ability to display each raster layer at a different transparency level. Use the transparency slider  to indicate to what extent the underlying layers (if any) should be visible though the current raster layer. This is very useful if you like to overlay more than one raster layer (e.g., a shaded relief map overlaid by a classified raster map). This will make the look of the map more three dimensional.



Additionally, you can enter a raster value that should be treated as *NODATA* in the *Additional no data value* menu.

Uma forma ainda mais flexível para personalizar a transparência pode ser feito no: guilabel: *seção de opções de transparência personalizado*. A transparência de cada pixel pode ser definido aqui.

As an example, we want to set the water of our example raster file `landcover.tif` to a transparency of 20%. The following steps are necessary:

1. Carregar o ficheiro raster:ficheiro:*landcover.tif*.
2. Open the *Properties* dialog by double-clicking on the raster name in the legend, or by right-clicking and choosing *Properties* from the pop-up menu.
3. Seleccionar *Transparência* menu
4. No menu *Transparencia da banda*, escolha 'Nenhum'.
5. Clique no botão  :sup: 'Add values manually'. Vai aparecer uma linha nova na lista de pixels.
6. Entre o valor dos raster na coluna 'De' e 'Para' (usamos 0 aqui), e ajuste a transparência a 20%.
7. Pressione no botão [**Aplicar**] e olhe para o mapa

Pode repetir os passos 5 e 6 para ajustar mais valores com a transparência personalizada.

As you can see, it is quite easy to set custom transparency, but it can be quite a lot of work. Therefore, you can use the button  Export to file to save your transparency list to a file. The button  Import from file loads your transparency settings and applies them to the current raster layer.

13.2.4 Menu Pirâmides

Large resolution raster layers can slow navigation in QGIS. By creating lower resolution copies of the data (pyramids), performance can be considerably improved, as QGIS selects the most suitable resolution to use depending on the level of zoom.

Deverá ter acesso à edição no directório onde os dados originais são armazenados para construir pirâmides.




Podem ser usados vários métodos de re-amostragem para calcular as pirâmides:

- Vizinho mais próximo
- Média
- Gauss
- Cúbico
- moda
- Nenhum

If you choose 'Internal (if possible)' from the *Overview format* menu, QGIS tries to build pyramids internally. You can also choose 'External' and 'External (Erdas Imagine)'.

Note que o cálculo de peirâmides pode modificar o arquivo original de dados, e uma vez criado, não pode ser apagado. Se desejar preservar uma versão 'sem pirâmides' do seu raster, faça uma cópia de segurança antes do cálculo das mesmas.

13.2.5 Menu Histograma

The *Histogram* menu allows you to view the distribution of the bands or colors in your raster. The histogram is generated automatically when you open the *Histogram* menu. All existing bands will be displayed together. You can save the histogram as an image with the  button. With the *Visibility* option in the  *Prefs/Actions* menu, you can display histograms of the individual bands. You will need to select the option  *Show selected band*. The *Min/max options* allow you to 'Always show min/max markers', to 'Zoom to min/max' and to 'Update style to min/max'. With the *Actions* option, you can 'Reset' and 'Recompute histogram' after you have chosen the *Min/max options*.

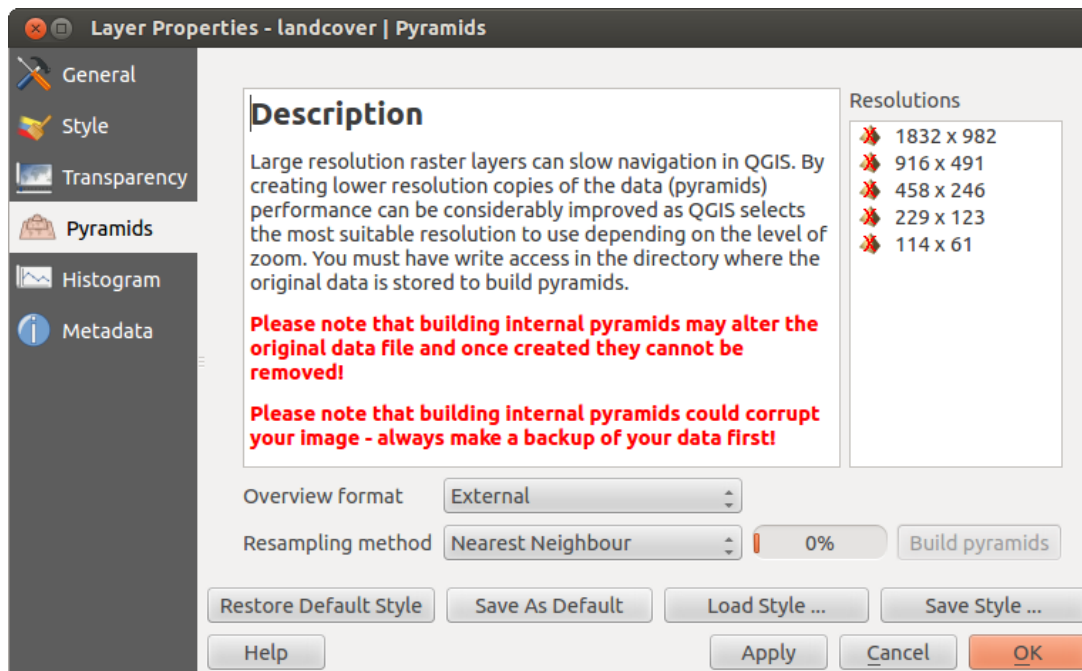


Figura 13.7: Menu Pirâmides 🐧

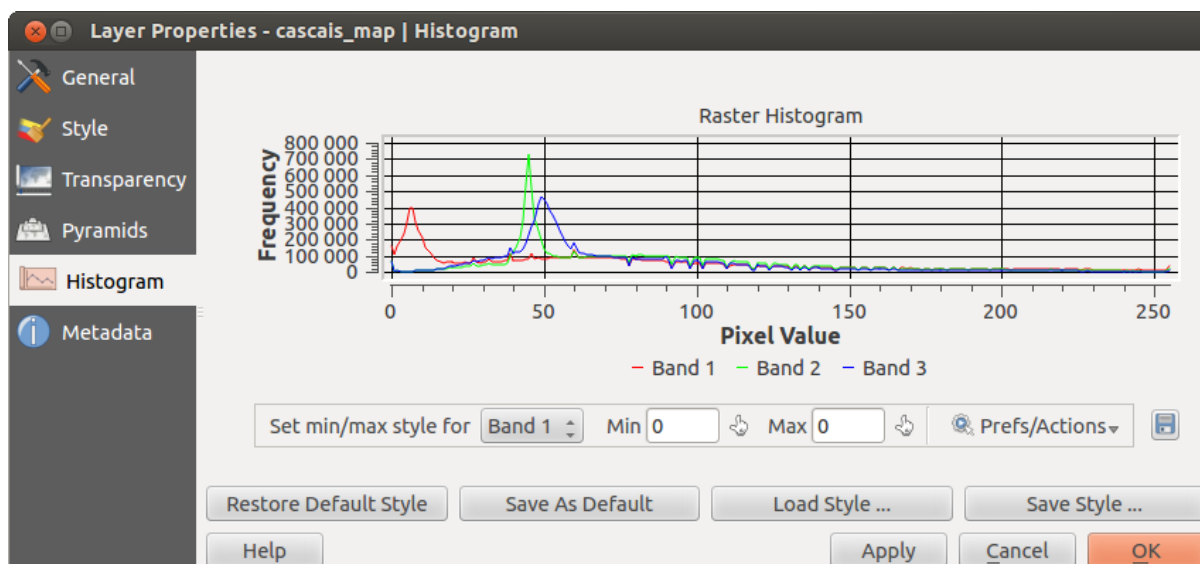


Figura 13.8: Histograma Raster 🐧

13.2.6 Menu Metadados

The *Metadata* menu displays a wealth of information about the raster layer, including statistics about each band in the current raster layer. From this menu, entries may be made for the *Description*, *Attribution*, *MetadataUrl* and *Properties*. In *Properties*, statistics are gathered on a 'need to know' basis, so it may well be that a given layer's statistics have not yet been collected.

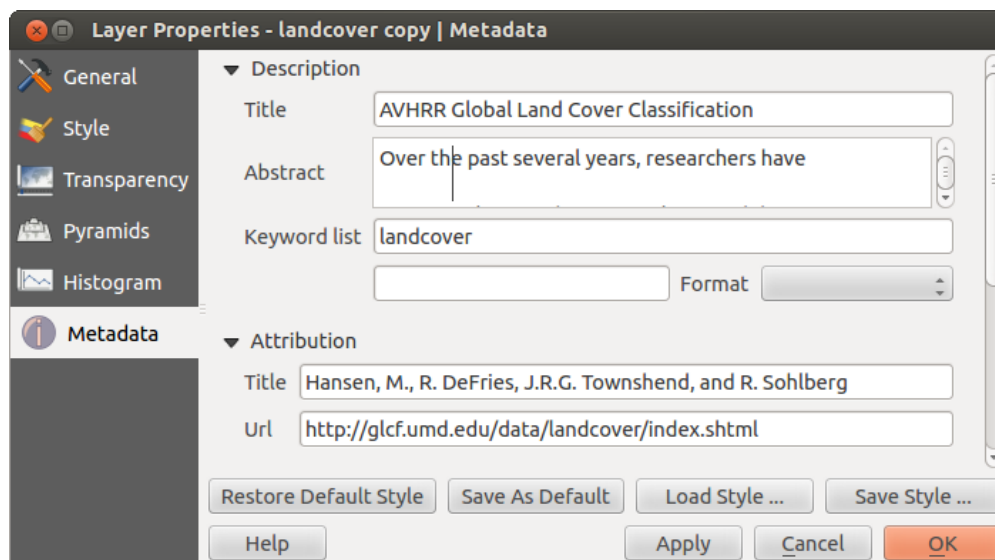


Figura 13.9: Metadados Raster 

13.3 Calculadora Matricial

The *Raster Calculator* in the *Raster* menu allows you to perform calculations on the basis of existing raster pixel values (see [figure_raster_10](#)). The results are written to a new raster layer with a GDAL-supported format.

The **Raster bands** list contains all loaded raster layers that can be used. To add a raster to the raster calculator expression field, double click its name in the Fields list. You can then use the operators to construct calculation expressions, or you can just type them into the box.

In the **Result layer** section, you will need to define an output layer. You can then define the extent of the calculation area based on an input raster layer, or based on X,Y coordinates and on columns and rows, to set the resolution of the output layer. If the input layer has a different resolution, the values will be resampled with the nearest neighbor algorithm.

The **Operators** section contains all available operators. To add an operator to the raster calculator expression box, click the appropriate button. Mathematical calculations (+, -, *, ...) and trigonometric functions (sin, cos, tan, ...) are available. Stay tuned for more operators to come!

With the *Add result to project* checkbox, the result layer will automatically be added to the legend area and can be visualized.

13.3.1 Exemplos

Convert elevation values from meters to feet

Creating an elevation raster in feet from a raster in meters, you need to use the conversion factor for meters to feet: 3.28. The expression is:

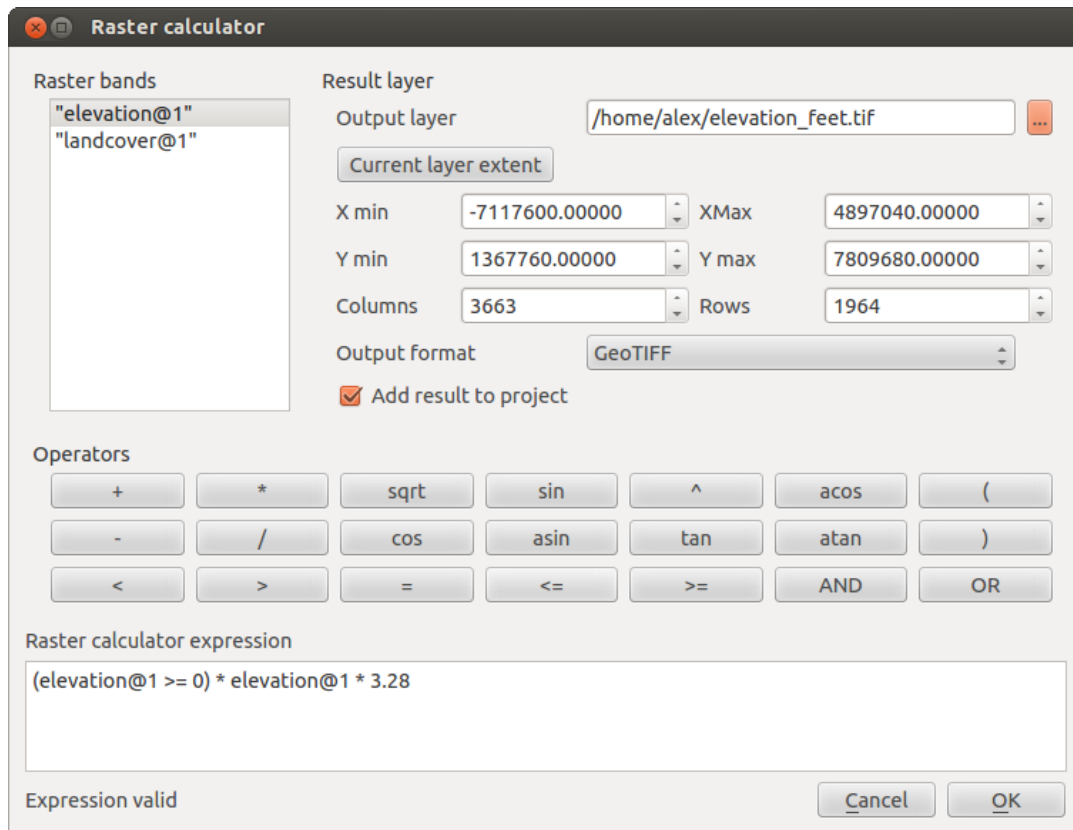


Figura 13.10: Calculador Matricial 🐧

```
"elevation@1" * 3.28
```

Usando uma máscara

If you want to mask out parts of a raster – say, for instance, because you are only interested in elevations above 0 meters – you can use the following expression to create a mask and apply the result to a raster in one step.

```
("elevation@1" >= 0) * "elevation@1"
```

In other words, for every cell greater than or equal to 0, set its value to 1. Otherwise set it to 0. This creates the mask on the fly.

If you want to classify a raster – say, for instance into two elevation classes, you can use the following expression to create a raster with two values 1 and 2 in one step.

```
("elevation@1" < 50) * 1 + ("elevation@1" >= 50) * 2
```

In other words, for every cell less than 50 set its value to 1. For every cell greater than or equal 50 set its value to 2.

.

Trabalhando com dados OGC

14.1 QGIS como Cliente de Dados OGC

O Consórcio Geoespacial Aberto (OGC), é uma organização internacional como mais de 300 organizações em todo mundo do tipo comerciais, sem fins lucrativos e de investigação. Os membros desenvolvem e implementam padrões para os conteúdos e serviços geoespaciais, processamento e troca de dados SIG.

Descrevendo um modelo básico de dados para elementos geográficos e um número crescente de especificações estão desenvolvidos para servir necessidades específicas para localização interoperável e tecnologia geoespacial, incluindo o SIG. Mais informação é encontrada em <http://www.opengeospatial.org/>.

As especificações OGC importantes suportadas pelo QGIS são:

- **WMS** — Serviço de Mapas Web (*Cliente WMS/WMTS*)
- **WMTS** — Serviço de Mosaicos de Mapa Web (*Cliente WMS/WMTS*)
- **WFS** — Serviços de Elementos Web (*WFS e WFS-T Cliente*)
- **WFS-T** — Serviços de Elementos Web - Transacionais (*WFS e WFS-T Cliente*)
- **WCS** — Serviços de Cobertura Web (*WCS Cliente*)
- **SFS** — Elementos Simples para SQL (*PostGIS Layers*)
- **GML** — Linguagem de Marcadores Geográfico

Os serviços OGC estão a ser crescentemente usados para troca de dados geoespaciais de diferentes implementações SIG e armazenamento de dados. O QGIS consegue lidar com as especificações em baixo como cliente, sendo **SFS** (através do suporte do fornecedor PostgreSQL/ PostGIS, veja Secção *PostGIS Layers*).

14.1.1 Cliente WMS/WMTS

Visão Global do Suporte WMS

O QGIS actualmente pode funcionar como cliente WMS que entende servidores WMS 1.1, 1.1.1 e 1.3. Foi particularmente testado contra serviços públicos de acesso como os DEMIS.

Os servidores WMS funcionam através de pedidos pelo cliente (ex.: QGIS) para mapas matriciais com uma dada extensão, conjunto de camadas, estilos de simbolização, e transparência. O servidor WMS de seguida consulta as suas fontes de dados locais, matricializa num mapa e manda de volta ao cliente em formato matricial. Para o QGIS isto tipicamente vem em JPEG ou PNG.

WMS é um serviço REST genérico (Estado de Transferência Representativo) mais que um Serviço Web completamente fundido. Como tal, pode realmente obter os URLs gerados pelo QGIS e usá-los num navegador da web para recuperar as mesmas imagens que o QGIS usa internamente. Isto pode ser útil para resolver problemas, uma

vez que existem várias marcas de servidores WMS no mercado e todos eles têm a sua própria interpretação da norma WMS.

As camadas WMS podem ser adicionadas facilmente, desde que conheça o acesso URL para o servidor WMS, tenha uma ligação de serviço a esse servidor, e o servidor compreenda HTTP como um mecanismos de transporte de dados.

Visão Global do Suporte WMTS

O QGIS pode também agir como um cliente WMTS. O WMTS é um padrão OGC para a distribuição de conjuntos de mosaicos de dados geoespaciais. Isto é uma forma mais rápida e mais eficiente para distribuir dados que o WMS porque com o WMTS os conjuntos de mosaicos são pré-gerados e o cliente apenas faz pedidos de transmissão dos mosaicos e não os produz. Um pedido típico do WMS envolve a geração e transmissão de dados. Um exemplo conhecido de um padrão não-OGC para a visualização de mosaicos de dados geoespaciais é o Google Maps.

De maneira a exibir os dados a várias escalas perto do que o utilizador queira ver, os conjuntos de mosaicos WMTS são produzidos em vários diferentes níveis de escala e são disponibilizados para o cliente SIG a seu pedido.

Este diagrama ilustra o conceito dos conjuntos de mosaicos

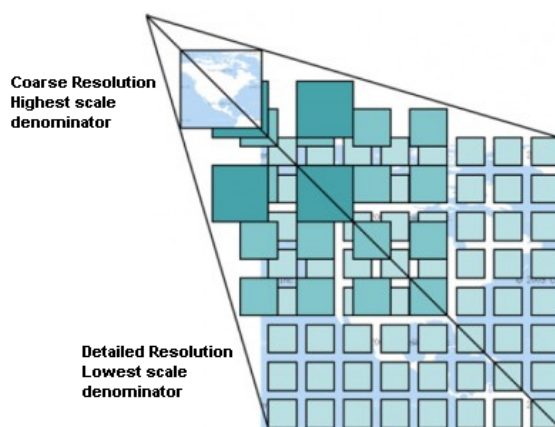


Figura 14.1: Conceito dos conjuntos de mosaicos do WMTS

Os dois tipos de interfaces WMTS que o QGIS suporta são via Key-Value-Pairs (KVP) e RESTful. Estas duas interfaces são diferentes e necessita especificá-los ao QGIS de forma diferente.

1) In order to access a **WMTS KVP** service, a QGIS user must open the WMS/WMTS interface and add the following string to the URL of the WMTS tile service:

```
"?SERVICE=WMTS&REQUEST=GetCapabilities"
```

Um exemplo deste tipo de endereço é

```
http://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&request=GetCapabilities
```

A camada topo2 funciona lindamente para testar neste WMTS. Adicionando esta linha e texto indica que é um serviço web WMTS que deverá ser usado em vez de o serviço WMS.

2. O serviço **WMTS RESTful** torna-se uma forma diferente, é um URL simples, a forma recomendada pela OGC é:

```
{WMTSBaseURL}/1.0.0/WMTSCapabilities.xml
```


Este formato ajuda-a a reconhecer que isto é um endereço RESTful. O WMTS RESTful é acedido no QGIS adicionando simplesmente o endereço na configuração do WMS no campo do URL da forma. Como exemplo para um mapa base Austríaco deste tipo de endereço é <http://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml>.

Note: Pode encontrar alguns serviços antigos chamados de WMS-C. Esses serviços são muito semelhantes ao WMTS com a mesma finalidade mas trabalham ligeiramente diferente). Pode gerir-los da mesma forma que faz nos serviços WMTS. Apenas adicione `?tiled=true` no final do url. Veja http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification para mais informações sobre esta especificação.

Quando lê o WMTS pode muitas vezes pensar em WMS-C.

Seleccionando os Servidores WMS/WMTS


Na primeira vez que usa o elemento WMS no QGIS, não existem servidores definidos.

Comece clicando no botão  Adicionar camada WMS na barra de ferramentas, ou através do menu *Camada* → *Adicionar Camada WMS*

A janela *Adicionar Camada(s) do Servidor* para adicionar camadas dos servidores WMS aparecem. Pode adicionar alguns servidores para brincar clicando no botão **[Adicionar servidores padrões]** . Isto irá adicionar dois servidores WMS demonstração para que possa usar, os servidores WMS do DM Solutions Group e Lizardtech. Para definir uma novo servidor WMS no separador *Camadas*, seleccione o botão **[Novo]** . De seguida introduza os parâmetros de ligação que deseja para o Servidor WMS, como está listado na *table_OGC_1*:

Nome	Um nome para esta ligação. Este nome será usado para a lista de Ligações do Servidor para que possa distingui-la de outros Servidores WMS.
URL	URL do servidor que fornece os dados. Isto deverá ser um nome de alojamento válido – o mesmo formato que irá usar para abrir a ligação telnet ou o ping a um alojamento.
Utilizador	Nome de Utilizador para aceder a um Servidor WMS protegido. Este parâmetro é opcional.
Palavra-chave	Palavra-chave para autenticação básica no Servidor WMS. Este parâmetro é opcional.
Ignorar GetMap URI	<input checked="" type="checkbox"/> <i>Ignorar GetMap URI reportado nas capacidades.</i> Use um URI dado em cima para o campo URL.
Ignorar GetFeatureInfo URI	<input checked="" type="checkbox"/> <i>Ignorar GetMap URI reportado nas capacidades,</i> use um URI dado em cima para o campo URL.

Tabela 1 OGC : Parâmetros de Ligação do WMS

Se precisar de configurar um servidor proxy para receber serviços WMS da internet, pode adicionar o seu servidor proxy nas opções. Escolha o menu *Configurações* → *Opções* e clique no separador *Rede & Proxy*. De seguida pode adicionar as suas configurações de proxy e activá-las configurando o *Use o proxy para acesso web*. Certifique-se que seleccionou o tipo proxy correcto da lista de menu *Tipo de Proxy* .

Uma vez a nova ligação do Servidor WMS for criada, será preservada para sessões futuras do QGIS.

Tip: Ligar URLs dos Servidores WMS

Certifique-se, que quando introduzir o URL do servidor WMS, seja o URL base. Por exemplo, não deve ter fragmentos como `request=GetCapabilities` ou `version=1.0.0` no seu URL.

Carregando as camadas WMS/WMTS

Uma vez preenchido os parâmetros com sucesso pode usar o botão **[Ligar]** para responder às capacidades do servidor seleccionado. Isto inclui a codificação da imagem, Camadas, Estilos de Camadas, e Projecções. Uma vez que isto é uma operação de rede, a velocidade de resposta depende da qualidade da ligação da sua rede ao servidor WMS. Enquanto faz a transferência de dados do servidor WMS, o progresso da transferência é visualizada no canto inferior esquerdo da janela do WMS.

O seu ecrã irá ficar parecido um pouco como *figure_OGR_1*, que mostra a resposta fornecida pelo servidor WMS do DM Solutions Group.

Codificação da Imagem

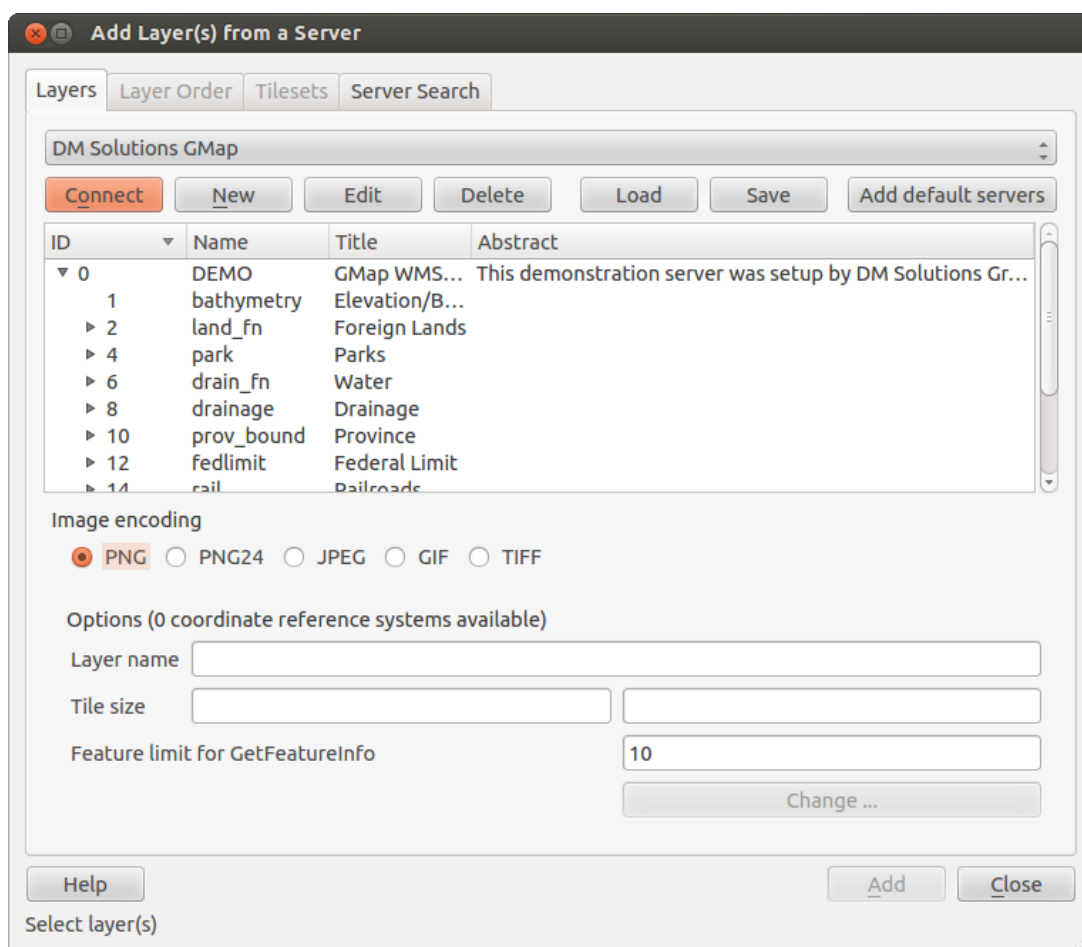


Figura 14.2: Janela para adicionar servidores WMS, mostra as camadas disponíveis 🐧

A secção *Codificação de Imagem* agora apresenta uma lista de formatos suportados pelo cliente e servidor. Escolha de acordo com as exigências de precisão da sua imagem.

Tip: Codificação da Imagem

Tipicamente irá encontrar servidores WMS que oferecem a escolha da codificação de imagem JPEG ou PNG. O JPEG é um formato de perda de compressão, enquanto que o PNG reprodução fielmente os dados brutos do matricial.

Use JPEG se espera dados WMS para fotografia na natureza e/ou não se importa da perda de alguma qualidade na imagem. Este típico trade.off reduz em 5 vezes o requerimento de transferência comparado ao PNG.

Use o PNG se necessitar representações precisas dos dados originais, e não se importa dos requerimentos de transferência de dados.

Opções

O campo Opções fornece um campo de texto onde pode adicionar *Nome da Camada* para a camada WMS. Este nome irá ser apresentado na legenda após carregamento da camada.

Em baixo do nome da camada pode definir *Tamanho do mosaico*, se quiser configurar os tamanhos do mosaico (ex.: 256x256) para dividir o pedido do WMS em pedidos múltiplos.

O *Limite de elementos para GetFeatureInfo* define quais os elementos do servidor para consulta.

Se seleccionou um WMS da lista, um capo com a projecção padrão, fornecida pelo mapserver, aparece. Se o botão [**Change...**] está activo, pode clicar e mudar para a projecção padrão do WMS para outro SRC fornecido pelo Servidor WMS.

Ordem das Camadas

O separador *Ordem de Camada* tem uma lista das camadas seleccionadas disponíveis do WMS actual ligado. Poderá reparar que algumas camadas expansíveis, isto significa que a camada pode ser exibida na escolha dos estilos de imagem.

Pode seleccionar várias camadas de uma só vez, mas apenas um estilo de imagem por camada. Quando várias camadas são seleccionadas, elas serão combinadas no Servidor WMS e transmitido ao QGIS de uma só vez.

Tip: Ordenação das Camadas WMS

As camadas WMS renderizadas por um servidor são sobrepostas na ordem da lista da secção de Camadas, de cima para baixo da lista. Se quiser alterar a ordem de sobreposição, pode usar o separador *Ordem de camadas*.

Transparência

Nesta versão do QGIS, a configuração *Transparência Global* da *Propriedades da Camada* está codificado para estar sempre ligado, quando disponível.

Tip: Transparência da Camada WMS

Disponibilidade da transparência da imagem do WMS depende da codificação da imagem usada: PNG e GIF suportam transparência enquanto que o JPEG deixa como não suportado.

Sistema de Coordenadas Referência

O Sistema de Coordenadas Referência (SRC) é a terminologia OGC para uma projecção QGIS.

Cada Camada WMS pode ser apresentada em múltiplos SRC, dependendo da capacidade do servidor WMS.

Para escolher o SRC seleccione [**Alterar...**] e a janela semelhante à da Figura Projecção 3 na *Trabalhando com Projecções* aparecerá. A diferença principal com a versão do WMS do ecrã é que neste apenas os SRC suportados apareceram no Servidor WMS.

Pesquisa de servidor

Dentro do QGIS pode pesquisar por servidores WMS. *Figure_OGC_2* mostra o separador *Pesquisar Servidor* com a janela *Adicionar Camada(s) do Servidor*.

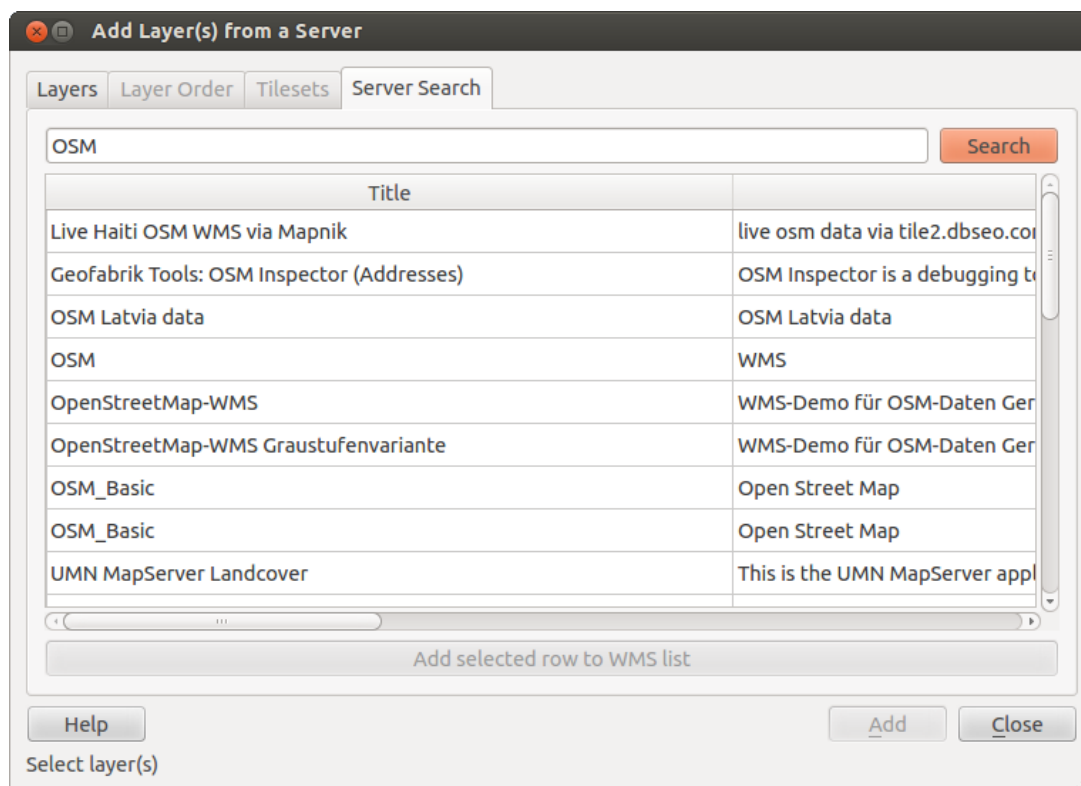


Figura 14.3: Janela de pesquisa de servidores WMS depois de algumas palavras-chave 🐧

Como pode ver é possível introduzir uma cadeia de texto de pesquisa no campo de texto e pressionar o botão **[Pesquisar]**. Após pouco tempo o resultado da pesquisa será preenchido na lista debaixo do campo de texto. Procure a lista de resultados dentro da tabela. Para visualizar os resultados, seleccione uma entrada da tabela, prima o botão **[Adicionar a linha seleccionada à lista WMS]** e mude para o separador *Camadas*. O QGIS automaticamente actualiza a sua lista de servidores e o resultado seleccionado da pesquisa está disponível na lista de servidores WMS guardados no separador *Camadas*. Apenas precisará de pedir a lista de camadas clicando no botão **[Ligar]**. Esta opção é especialmente útil quando quer pesquisar mapas por palavras-chave específicas.

Basicamente esta opção é um frontend para a API do <http://geopole.org>.


Mosaicos

Ao usar Serviços WMTS (WMS em cache) como


```
http://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?
service=WMTS&request=GetCapabilities
```

pode procurar através do separador *Conjuntos de Mosaicos* dados pelo servidor. Informação adicional como o tamanho do mosaico, formatos e SRC suportados estão na lista desta tabela. Em combinação com esta característica pode usar a escala da quadrícula do *Configurações* → *Painéis* (KDE e Windows) ou *Ver* → *Painéis* (Gnome e MacOSX) e de seguida escolha *Escala da quadrícula*, que dá escalas disponíveis do servidor de mosaicos com uma barra de deslizamento acoplada.

Usando a Ferramenta Identificar

Uma vez adicionado o servidor WMS, e se qualquer camada do servidor WMS é consultável, pode usar a ferramenta  para seleccionar o píxel do enquadramento do mapa. A consulta é feita ao servidor WMS para cada selecção feita. Os resultados da consulta vêm na forma de um texto plano. A formatação desse texto depende do servidor WMS particular usado. **Seleção do formato**

Se múltiplos formatos são suportados pelo servidor, uma caixa de combinação com os formatos suportados é automaticamente adicionado aos resultados da janela de identificação e o formato seleccionado irá armazenar no projecto para a camada. **Suporte do formato GML**

A ferramenta  suporta respostas do Servidor WMS (GetFeatureInfo) no formato GML (é chamado de Elemento no GUI do QGIS neste contexto). Se o formato “Elemento” for suportado pelo servidor e seleccionado, os resultados da ferramenta Identificar são elementos vectoriais como normais camadas vectoriais. Quando um elemento é seleccionado na árvore, é destacada no mapa e pode ser copiada para a área de transferência e colada noutra camada vectorial. Veja o exemplo da instalação em baixo do UMN Mapserver para suportar o formato GML GetFeatureInfo.

```
# in layer METADATA add which fields should be included and define geometry (example):

"gml_include_items"    "all"
"ows_geometries"      "mygeom"
"ows_mygeom_type"     "polygon"

# Then there are two possibilities/formats available, see a) and b):

# a) basic (output is generated by Mapserver and does not contain XSD)
# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "application/vnd.ogc.gml,text/html"

# b) using OGR (output is generated by OGR, it is send as multipart and contains XSD)
# in MAP define OUTPUTFORMAT (example):
OUTPUTFORMAT
  NAME "OGRGML"
  MIMETYPE "ogr/gml"
  DRIVER "OGR/GML"
  FORMATOPTION "FORM=multipart"
END

# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "OGRGML,text/html"
```

Propriedades de Visualização

Uma vez ter adicionado o servidor WMS, pode ver as suas propriedades clicando com o direito do rato na legenda, e seleccionando *Propriedades*. **Separador de Metadados**

O separador *Metadados* exibem a riqueza de informação sobre o servidor WMS, geralmente recolhidos a partir da declaração de de Capacidades devolvidos a partir desse servidor. Muitas das definições podem ser removidas através da leitura dos padrões WMS (veja OPEN-GEOSPATIAL-CONSORTIUM *Literatura e Referências Web*), mas aqui estão algumas definições uteis:

- **Propriedades do Servidor**

- **Versão do WMS** — Versão do WMS suportada pelo servidor.
- **Formatos de Imagem** — A lista de MIME-types que o servidor pode responder com o desenho do mapa. O QGIS formata qualquer que seja as bibliotecas Qt subjacentes que foram construídas, que é pelo menos tipicamente a `image/png` e `image/jpeg`.
- **Formatos de Identidade** — A lista dos MIME-types do servidor pode responder quando usa a ferramenta identificar. Actualmente o QGIS suporta o tipo `text-plain`.

- **Propriedades da Camada**

- **Seleccionado** — Querendo ou não esta camada seleccionada quando o seu servidor foi adicionado a este projecto.
- **Visível** — Seja ou não essa camada esta é seleccionada como visível na legenda. (Ainda não é usado nesta versão do QGIS.)
- **Pode Identificar** — Camada que retorna ou não resultados através do uso da ferramenta Identificar.
- **Pode ser Transparente** — Quando uma camada pode ou não pode ser renderizada com transparência. Esta versão do QGIS irá sempre usar a transparência se esta é `Sim` e a codificação da imagem suportar transparência
- **Permite Aproximar** — Quando uma camada pode ou não pode ser ampliada neste servidor. Esta versão do QGIS assume que todas as camadas WMS têm esta definida como `Sim`. Camadas deficientes podem ser renderizadas de forma estranha.
- **Contagem em cascata** — Os servidores WMS pode funcionar como proxy para outros servidores WMS para obter dados matriciais para uma camada. Esta entrada mostra quantas vezes o pedido para esta camada é avançada por peer nos servidores WMS para o resultado.
- **Largura Fixa, Altura Fixa** — Quando uma camada pode ou não pode ter dimensões de pixels da fonte fixadas. Esta versão QGIS assume que todas as camadas WMS têm esta definição como nada. Camadas deficientes podem ser renderizadas de forma estranha.
- **Caixa de Enquadramento WGS 84** — A caixa de enquadramento desta camada, nas coordenadas WGS 84. Alguns servidores WMS não configuram esta situação correctamente (ex.: usam as coordenadas UTM). Se for esse o caso, então a vista inicial desta camada pode ser renderizada com uma aparência 'afastada' pelo QGIS. O webmaster do WMS deve informar deste erro, que pode ser conhecido como elementos WMS XML `LatLonBoundingBox`, `EX_GeographicBoundingBox` ou `CRS:84 BoundingBox`.
- **SRC disponíveis** — As projecções desta camada podem ser renderizadas pelo servidor WMS. Os mesmos estão listados num formato nativo WMS.
- **Disponível em estilo** — Os estilos da imagem desta camada podem ser renderizadas pelo o servidor WMS.

Mostra a legenda gráfica do WMS no tabela de conteúdos e compositor

O fornecedor de dados WMS do QGIS permite exibir o gráfico da legenda na tabela de conteúdos da lista da camada e no compositor do mapa. A legenda WMS irá ser exibida apenas se o servidor WMS tiver a capacidade `GetLegendGraphic` e a camada tiver um url `getCapability` especificado, portanto adicionalmente terá de seleccionar um estilo para a camada.

Se a `legendGraphic` estiver disponível, é exibido em baixo da camada. é pequeno e tem de clicar nele para este ser aberto na sua dimensão real (devido à limitação da arquitectura do `QgsLegendInterface`). Ao clicar na legenda da camada irá ser aberto uma moldura com a legenda na sua resolução completa.


No compositor de impressão, a legenda irá ser integrada na sua dimensão original (transferida). A resolução do gráfico da legenda pode ser configurado nas propriedades do item em Legenda -> WMS `LegendGraphic` para coincidir com os seus requisitos de impressão.

A legenda irá exibir informação de contexto baseada na sua escala actual. A legenda WMS será exibida apenas se o WMS tiver capacidade `GetLegendGraphic` e a camada tiver um url `getCapability` específico, possa que possa seleccionar um estilo.

Limitações do WMS Cliente

Nem todas as funcionalidades possíveis do cliente WMS foram incluídas nesta versão do QGIS. Seguem-se algumas das mais notáveis excepções .

Editando as Configurações da Camada WMS

Uma vez completo o procedimento  Adicionar camada WMS, não existe forma para alterar as configurações. Uma forma alternativa é apagar a camada completamente e começar de novo.

Servidores WMS que Requerem Autenticação

Actualmente o acesso público e serviços WMS protegidos são suportados. Os servidores WMS protegidos podem ser acedidos pela autenticação pública. Pode adicionar credenciais (opcionais) quando adiciona o servidor WMS. Veja a secção *Seleccionando os Servidores WMS/WMTS* para detalhes.


Tip: Accedendo a camadas protegidas OGC

Se necessitar de proteger camadas com outros métodos seguros além da autenticação básica, pode usar o InteProxy como proxy transparente, que suporta vários métodos de suporte. Mais informação pode ser encontrada no manual InteProxy no sítio na internet <http://inteproxy.wald.intevation.org>.

Tip: lqgl WMS Mapserver

Desde a Versão 1.7. o QGIS tem uma implementação própria do Masserver WMS 1.3.0. Leia mais sobre este assunto no capítulo *QGIS com Servidor de Dados OGC*.

14.1.2 WCS Cliente

 O Serviço Web de Cobertura (WCS) fornecem acesso aos dados matriciais de forma a serem úteis à renderização cliente, como entrada a modelos científicos, e para outros clientes. O WCS pode ser comprado ao WFS e ao WMS. Como as instâncias WMS e WFS, o WCS permite aos clientes escolher porções de informação guardada de servidores baseados condicionantes espaciais e outros critérios de consulta.

O QGIS tem um fornecedor WCS nativo e suporta a versão 1.0 e 1.1 (que são significativamente diferentes), mas actualmente dá-se preferência ao 1.0, porque o 1.1 tem vários problemas (ex.: cada servidor implementa de formas diferentes com várias particularidades).

O fornecedor WCS nativo gere todos pedidos de rede e usa as configurações padrões de rede do QGIS (especialmente o proxy). É também possível seleccionar o modo de cache ('sempre cache', 'preferir cache', 'preferir rede', 'sempre rede') e fornece também suporte à selecção de posição do tempo se o domínio temporal é oferecido pelo servidor.



14.1.3 WFS e WFS-T Cliente

No QGIS, a camada WFS comporta-se de forma semelhante a uma camada vectorial. Pode identificar e seleccionar elementos e ver a tabela de atributos. Desde o QGIS 1.6 a edição WFS-T é também suportada.

De uma forma geral adicionar uma camada WFS é muito semelhante ao procedimento usado com o WMS. A diferença é que não existe servidores padrões definidos, portanto nós teremos de adicionar o nosso.

Carregando uma camada WFS

Como exemplo nós usaremos o servidor WFS DM Solutions e exibiremos uma camada, O URL é: http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap

1. Clique na ferramenta  Adicionar Camada WFS na barra de ferramentas Camadas, a janela *Adicionar Camada WFS do Servidor* aparecerá.
2. Clique em [Novo].
3. Introduza 'DM Solutions' como nome.
4. Introduza o URL (veja em cima).
5. Clique [OK].
6. Escolha 'DM Solutions' da lista *Ligações do Servidor* .

7. Clique [**Ligar**].
8. Espere pela lista de camadas a ser preenchida.
9. Selecione a camada *Parques* da lista.
10. Clique [**Aplicar**] para adicionar a camada ao mapa.

Repare que as configurações proxy que definiu nas suas preferências são também reconhecidas.

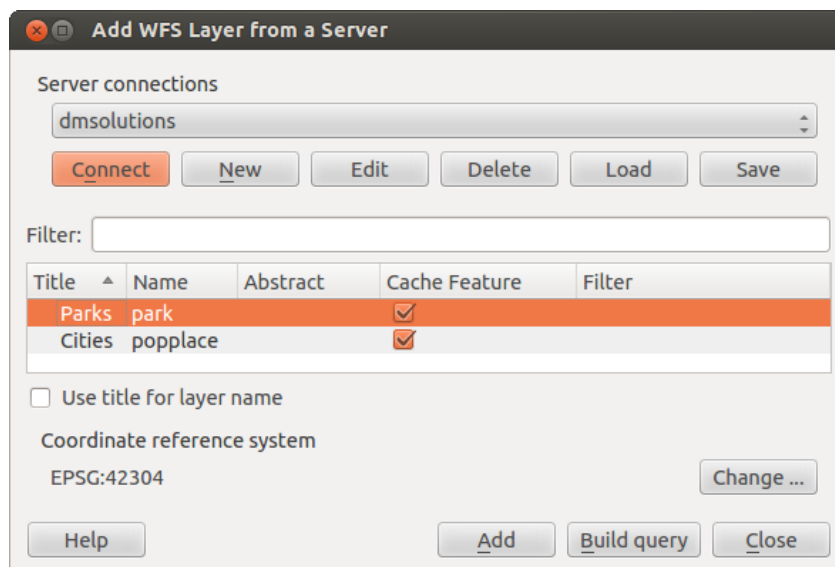


Figura 14.4: Adicionando uma camada WFS 

Será avisado do progresso de transferência que pode ser visualizado na janela principal do QGIS no canto inferior esquerdo. Uma vez a camada carregada, pode identificar e seleccionar uma província ou duas e ver na tabela de atributos.

Apenas o WFS 1.0.0 é suportado. Até este momento não houve muitos testes contra versões WFS implementadas em outros servidores WFS. Se encontrar problemas com outros servidores WFS, por favor não hesite em contactar a equipa de desenvolvimento. Por favor dirija-se à Secção *Ajuda e Suporte* para mais informação sobre as listas de discussões.

Tip: Encontrando Servidores WFS

Pode encontrar servidores WFS adicionais usando o Google ou outro motor de busca favorito. Existe um número de listas com URLs públicos, alguns têm manutenção outros não.

14.2 QGIS com Servidor de Dados OGC

QGIS Server is an open source WMS 1.3, WFS 1.0.0 and WCS 1 1.1.1 implementation that, in addition, implements advanced cartographic features for thematic mapping. The QGIS Server is a FastCGI/CGI (Common Gateway Interface) application written in C++ that works together with a web server (e.g., Apache, Lighttpd). It is funded by the EU projects Orchestra, Sany and the city of Uster in Switzerland.

QGIS Server uses QGIS as back end for the GIS logic and for map rendering. Furthermore, the Qt library is used for graphics and for platform-independent C++ programming. In contrast to other WMS software, the QGIS Server uses cartographic rules as a configuration language, both for the server configuration and for the user-defined cartographic rules.

As QGIS desktop and QGIS Server use the same visualization libraries, the maps that are published on the web look the same as in desktop GIS.

In one of the following manuals, we will provide a sample configuration to set up a QGIS Server. For now, we recommend to read one of the following URLs to get more information:

- http://karlinapp.ethz.ch/qgis_wms/
- http://hub.qgis.org/projects/quantum-gis/wiki/QGIS_Server_Tutorial
- <http://linfiniti.com/2010/08/qgis-mapserver-a-wms-server-for-the-masses/>

14.2.1 Instalação simples no Debian Squeeze

At this point, we will give a short and simple sample installation how-to for Debian Squeeze. Many other OSs provide packages for QGIS Server, too. If you have to build it all from source, please refer to the URLs above.

Apart from QGIS and QGIS Server, you need a web server, in our case apache2. You can install all packages with aptitude or apt-get install together with other necessary dependency packages. After installation, you should test to confirm that the web server and QGIS Server work as expected. Make sure the apache server is running with /etc/init.d/apache2 start. Open a web browser and type URL: http://localhost. If apache is up, you should see the message 'It works!'.

Now we test the QGIS Server installation. The qgis_mapserv.fcgi is available at /usr/lib/cgi-bin/qgis_mapserv.fcgi and provides a standard WMS that shows the state boundaries of Alaska. Add the WMS with the URL http://localhost/cgi-bin/qgis_mapserv.fcgi as described in *Seleccionando os Servidores WMS/WMTS*.

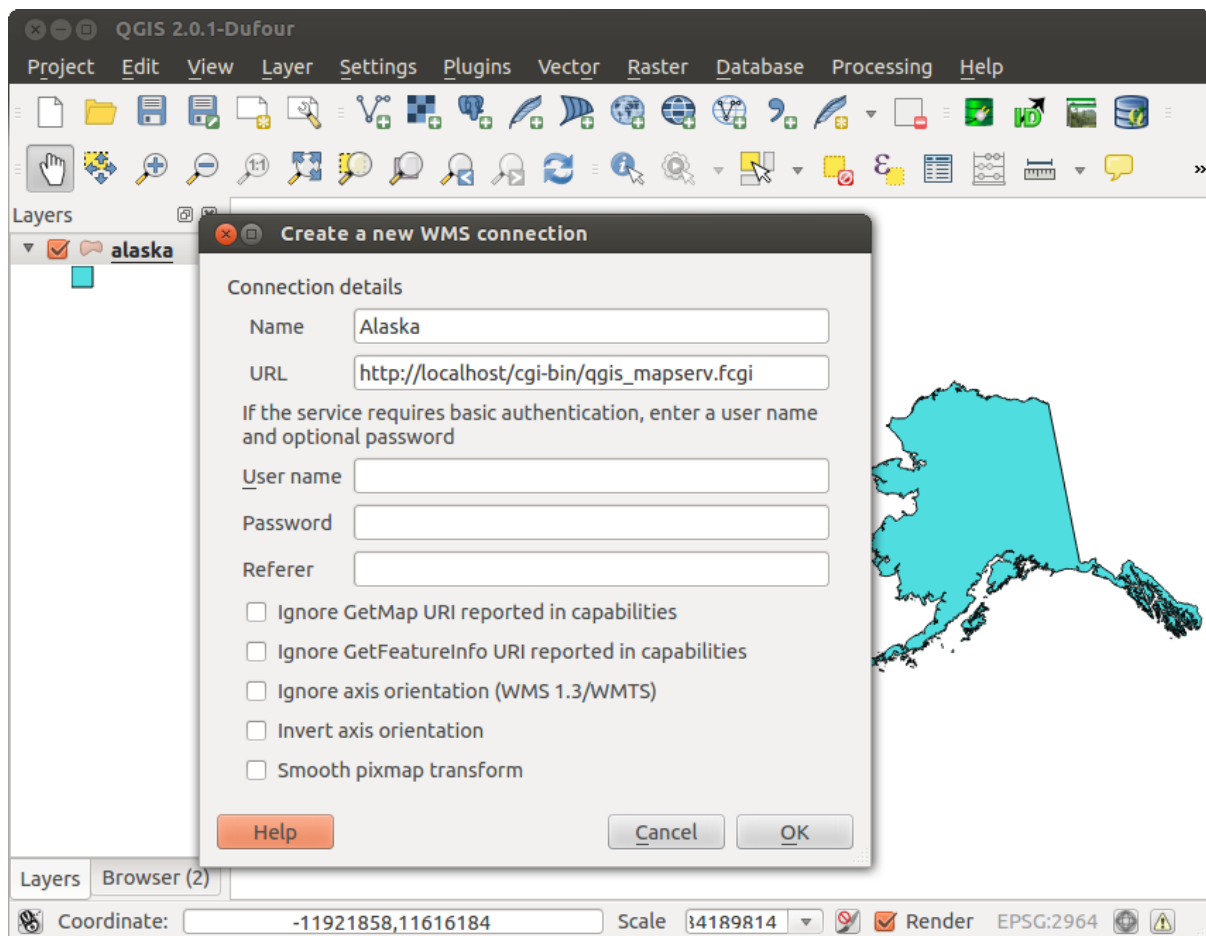


Figura 14.5: WMS padrão com os limites dos EUA incluído no Servidor QGIS (KDE) 🐧

14.2.2 Creating a WMS/WFS/WCS from a QGIS project

To provide a new QGIS Server WMS, WFS or WCS, we have to create a QGIS project file with some data. Here, we use the 'Alaska' shapefile from the QGIS sample dataset. Define the colors and styles of the layers in QGIS and the project CRS, if not already defined.

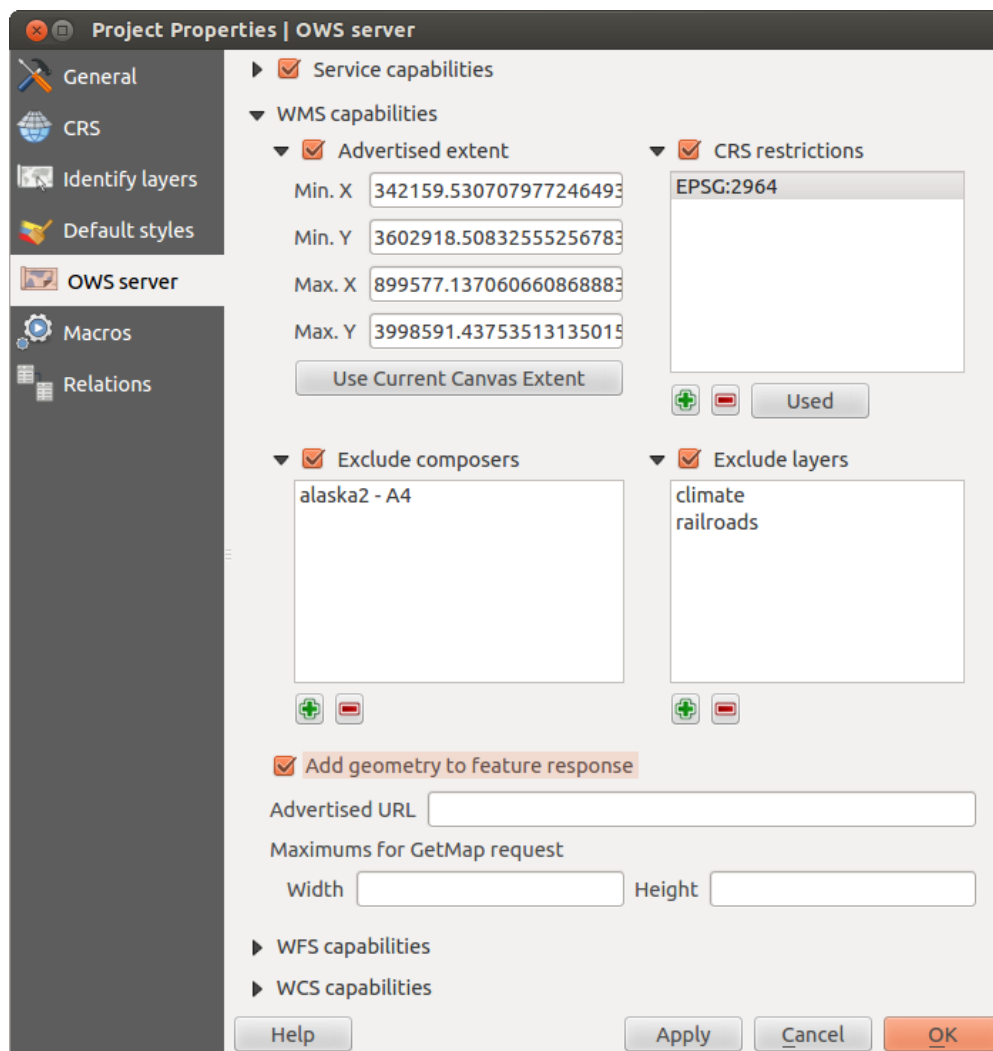




Figura 14.6: Definições para um QGIS Server WMS/WFS/WCS project (KDE)


Then, go to the *OWS Server* menu of the *Project* → *Project Properties* dialog and provide some information about the OWS in the fields under *Service Capabilities*. This will appear in the *GetCapabilities* response of the WMS, WFS or WCS. If you don't check *Service capabilities*, QGIS Server will use the information given in the `wms_metadata.xml` file located in the `cgi-bin` folder.

WMS capabilities

In the *WMS capabilities* section, you can define the extent advertised in the WMS *GetCapabilities* response by entering the minimum and maximum X and Y values in the fields under *Advertised extent*. Clicking *Use Current Canvas Extent* sets these values to the extent currently displayed in the QGIS map canvas. By checking *CRS restrictions*, you can restrict in which coordinate reference systems (CRS) QGIS Server will offer to render maps. Use the  button below to select those CRS from the Coordinate Reference System Selector, or click *Used* to add the CRS used in the QGIS project to the list.

If you have print compositors defined in your project, they will be listed in the *GetCapabilities* response, and they can be used by the *GetPrint* request to create prints, using one of the print compositor layouts as a template. This

is a QGIS-specific extension to the WMS 1.3.0 specification. If you want to exclude any print composer from being published by the WMS, check *Exclude composers* and click the  button below. Then, select a print composer from the *Select print composer* dialog in order to add it to the excluded composers list.

If you want to exclude any layer or layer group from being published by the WMS, check *Exclude Layers* and click the  button below. This opens the *Select restricted layers and groups* dialog, which allows you to choose the layers and groups that you don't want to be published. Use the *Shift* or *Ctrl* key if you want to select multiple entries at once.

You can receive requested GetFeatureInfo as plain text, XML and GML. Default is XML, text or GML format depends the output format chosen for the GetFeatureInfo request.

If you wish, you can check *Add geometry to feature response*. This will include in the GetFeatureInfo response the geometries of the features in a text format. If you want QGIS Server to advertise specific request URLs in the WMS GetCapabilities response, enter the corresponding URL in the *Advertised URL* field. Furthermore, you can restrict the maximum size of the maps returned by the GetMap request by entering the maximum width and height into the respective fields under *Maximums for GetMap request*.

If one of your layers uses the Map Tip display (i.e. to show text using expressions) this will be listed inside the GetFeatureInfo output. If the layer uses a Value Map for one of his attributes, also this information will be shown in the GetFeatureInfo output.

WFS capabilities

In the *WFS capabilities* area, you can select the layers that you want to publish as WFS, and specify if they will allow the update, insert and delete operations. If you enter a URL in the *Advertised URL* field of the *WFS capabilities* section, QGIS Server will advertise this specific URL in the WFS GetCapabilities response.

WCS capabilities

In the *WCS capabilities* area, you can select the layers that you want to publish as WCS. If you enter a URL in the *Advertised URL* field of the *WCS capabilities* section, QGIS Server will advertise this specific URL in the WCS GetCapabilities response.

Now, save the session in a project file `alaska.qgs`. To provide the project as a WMS/WFS, we create a new folder `/usr/lib/cgi-bin/project` with admin privileges and add the project file `alaska.qgs` and a copy of the `qgis_mapserv.fcgi` file - that's all.

Now we test our project WMS, WFS and WCS. Add the WMS, WFS and WCS as described in *Carregando as camadas WMS/WMTS, WFS e WFS-T Cliente* and *WCS Cliente* to QGIS and load the data. The URL is:

```
http://localhost/cgi-bin/project/qgis_mapserv.fcgi
```

Afinando o seu OWS

For vector layers, the *Fields* menu of the *Layer* → *Properties* dialog allows you to define for each attribute if it will be published or not. By default, all the attributes are published by your WMS and WFS. If you want a specific attribute not to be published, uncheck the corresponding checkbox in the *WMS* or *WFS* column.

You can overlay watermarks over the maps produced by your WMS by adding text annotations or SVG annotations to the project file. See section *Annotation Tools* in *Ferramentas gerais* for instructions on creating annotations. For annotations to be displayed as watermarks on the WMS output, the *Fixed map position* check box in the *Annotation text* dialog must be unchecked. This can be accessed by double clicking the annotation while one of the annotation tools is active. For SVG annotations, you will need either to set the project to save absolute paths (in the *General* menu of the *Project* → *Project Properties* dialog) or to manually modify the path to the SVG image in a way that it represents a valid relative path.

Os parâmetros extra suportados pelo pedido WMS GetMap

In the WMS GetMap request, QGIS Server accepts a couple of extra parameters in addition to the standard parameters according to the OGC WMS 1.3.0 specification:

- Parâmetro **MAPA**: Semelhante ao MapServer, o parâmetro **MAPA** pode ser usado para especificar o caminho para o ficheiro de projecto QGIS. Pode especificar um caminho absoluto ou relativo para a localização do executável do servidor (`qgis_mapserv.fcgi`). Se não estiver especificado, o Servidor QGIS procura por ficheiros `.qgs` no directório onde o executável do servidor está localizado.

Exemplo:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?\nREQUEST=GetMap&MAP=/home/qgis/mymap.qgs&...
```

- Parâmetro **DPI**: O parâmetro **DPI** pode ser usado para especificar o pedido de saída da resolução.

Exemplo:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?REQUEST=GetMap&DPI=300&...
```

- Parâmetro **OPACIDADE**: A opacidade pode se definida numa camada ou ao nível do grupo. O intervalo de valores permitido vai de 0 (totalmente transparente) a 255 (totalmente opaco).

Exemplo:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?\nREQUEST=GetMap&LAYERS=mylayer1,mylayer2&OPACITIES=125,200&...
```

QGIS Server logging

To log requests send to server, set the following environment variables:

- **QGIS_SERVER_LOG_FILE**: Specify path and filename. Make sure that server has proper permissions for writing to file. File should be created automatically, just send some requests to server. If it's not there, check permissions.
- **QGIS_SERVER_LOG_LEVEL**: Specify desired log level. Available values are:
 - 0 INFO (log all requests),
 - 1 WARNING,
 - 2 CRITICAL (log just critical errors, suitable for production purposes).

Exemplo:

```
SetEnv QGIS_SERVER_LOG_FILE /var/tmp/qgislog.txt\nSetEnv QGIS_SERVER_LOG_LEVEL 0
```

Note

- When using Fcgid module use `FcgidInitialEnv` instead of `SetEnv`!
- Server logging is enabled also if executable is compiled in release mode.

Environment variables

- **QGIS_OPTIONS_PATH**: The variable specifies path to directory with settings. It works the same ways as QGIS application `-optionspath` option. It is looking for settings file in `<QGIS_OPTIONS_PATH>/QGIS/QGIS2.ini`. For exaple, to set QGIS server on Apache to use `/path/to/config/QGIS/QGIS2.ini` settings file, add to Apache config:

```
SetEnv QGIS_OPTIONS_PATH "/path/to/config/"
```

Trabalhando com dados GPS



15.1 Módulo GPS



15.1.1 O que é o GPS?

GPS, Sistema de Posicionamento Global, é um sistema baseado em satélites que permite qualquer um com um receptor GPS encontrar a sua posição exacta em qualquer parte do mundo. É usado para auxiliar a navegação, por exemplo, aviões, barcas e por caminhantes. O receptor GPS usa sinais a partir dos satélites para calcular a latitude e longitude e (por vezes) a elevação. A maioria dos receptores também têm a capacidade de armazenar localizações (conhecidos como **sítios de interesses**), sequências de localizações que faz uma **rota** planeada e um registo de trilho ou **trilho** dos movimentos do receptores ao longo do tempo. Os sítios de interesse, rotas e trilhos são três tipos básicos de elementos dos dados GPS. O QGIS exhibe sítios de interesse em camadas do tipo ponto enquanto que as rotas e trilhos são exibidos em camadas do tipo linha.


15.1.2 Carregando dados GPS a partir de um ficheiro

Existem dezenas de diferentes formatos de ficheiro para armazenar dados GPS. O formato que o QGIS usa é chamado de GPX (GPS eXchange format), que é um formato padrão de intercâmbio que pode conter um número variado de sítios de interesse, rotas e trilhos no mesmo ficheiro.

Para carregar um ficheiro GPX primeiro necessita de carregar o módulo. *Módulos* →  *Gestor de Módulos...* abre a janela do Gestor de Módulos. Active a caixa de verificação  *Ferramentas GPS*. Quando o módulo é carregado dois botões com o desenho de um pequeno GPS de mão é exibido na caixa de ferramentas:

-  Criar nova camada GPX
-  Ferramentas GPS

Para trabalhar com os dados GPS nós fornecemos um exemplo de ficheiro GPX que está disponível no conjunto de dados amostra QGIS `qgis_sample_data/gps/national_monuments.gpx`. Veja a secção *Amostra de Dados* para mais informação sobre a amostra de dados.

1. Selecione *Vector* → *GPS* → *Ferramentas GPS* ou clique no ícone  Ferramentas GPS na barra de ferramentas e abra o separador *Carregar ficheiro GPX* (veja [figure_GPS_1](#)).
2. Pesquisa na pasta `qgis_sample_data/gps/`, seleccione o ficheiro GPX `national_monuments.gpx` e clique [**Abrir**].

Use o botão [**Pesquisar...**] para seleccionar o ficheiro GPX, e de seguida use as caixas de verificação para seleccionar os tipos de elementos que quer carregar do ficheiro GPX. Cada tipo de elemento será carregado em camadas separadas quando clica em [**OK**]. O ficheiro `national_monuments.gpx` apenas contém sítios de interesse.

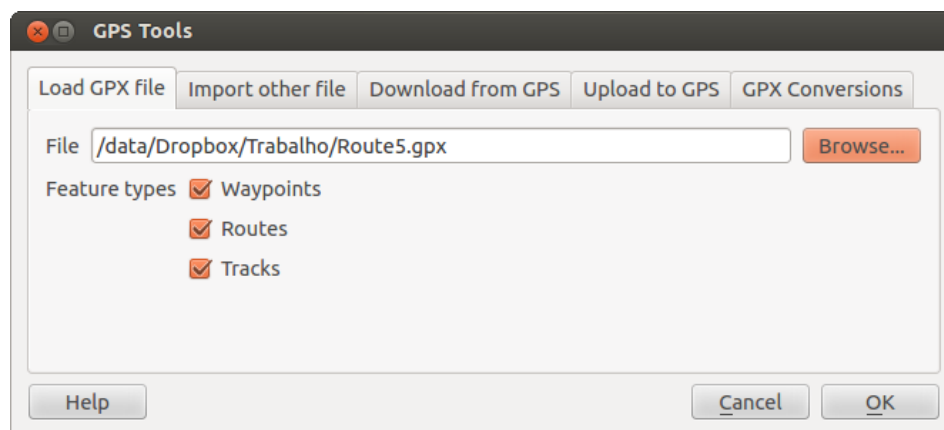


Figura 15.1: Janela das Ferramentas GPS 

Note: Os dispositivos GPS permitem armazenar os dados em diferentes sistemas de coordenadas. Quando é feita a transferência do ficheiro GPX (a partir do seu dispositivo GPS ou de um sítio na internet) e de seguida carregado no QGIS, tenha atenção que os dados armazenados no ficheiro GPX usa WGS 84 (latitude/longitude). O QGIS tem em conta isso e é a especificação oficial do GPX. Veja <http://www.topografix.com/GPX/1/1/>.

15.1.3 GPSTabel

Uma vez que o QGIS usa ficheiros GPX você necessita de uma forma de converter os outros formatos de ficheiros GPS para GPX. Isto pode ser feito para vários formatos usando o programa gratuito GPSTabel, que está disponível em <http://www.gpsbabel.org>. Este programa também transfere os dados GPS entre o seu computador e o dispositivo GPS. O QGIS usa o GPSTabel para este tipo de coisas, portanto é recomendado que você o instale. Contudo, se apenas quer carregar dados GPS a partir de ficheiros GPX não irá necessitar dele. A versão 1.2.3 do GPSTabel é conhecido por trabalhar com o QGIS, mas deverá conseguir usar versão mais recentes sem problemas.

15.1.4 Importando dados GOS


Para importar dados GPS a partir de um ficheiro que não é ficheiro GPX, pode usar a ferramenta *Importar outro ficheiro* na janela das Ferramentas GPS. Aqui selecciona o ficheiro que quer importar (e o tipo de ficheiro), que tipo de elemento quer importar, onde quer armazenar o ficheiro GPX convertido e que nome da nova camada irá ter. Tenha atenção que nem todos os formatos de dados GPS irão suportar os três tipos de elementos, portanto para muitos formatos apenas poderá escolher entre um ou dois tipos.

15.1.5 Transferindo dados GPS a partir do dispositivo

O QGIS pode usar o GPSTabel para transferir dados directamente de um dispositivo GPS como camadas vectoriais. Para isso podemos usar o separador *Transferir do GPS* da janela de Ferramentas GPS (veja a [Figure_GPS_2](#)). Aqui, selecciona o tipo de dispositivo GPS, a porta que está ligado a (ou o usb se o seu GPS suporta-lo), o tipo de elemento que quer transferir, o ficheiro GPX onde quer armazenar os dados, e o nome da nova camada.

O tipo de dispositivo que selecciona no menu do dispositivo GPS determina como o GPSTabel tentará comunicar com o seu dispositivo GPS. Se nenhum dos tipos disponíveis não trabalha com o seu dispositivo GPS pode criar um novo tipo (veja secção [Definindo novos tipos de dispositivos](#)).

A porta pode ser um nome de ficheiro ou outro nome que o seu sistema operativo usa como referência a uma porta física no seu computador no qual o dispositivo GPS está ligado. Pode ser um simples USB, para dispositivos GPS com USB.

-  No Linux é algo do género como `/dev/ttyS0` ou `/dev/ttyS1`.

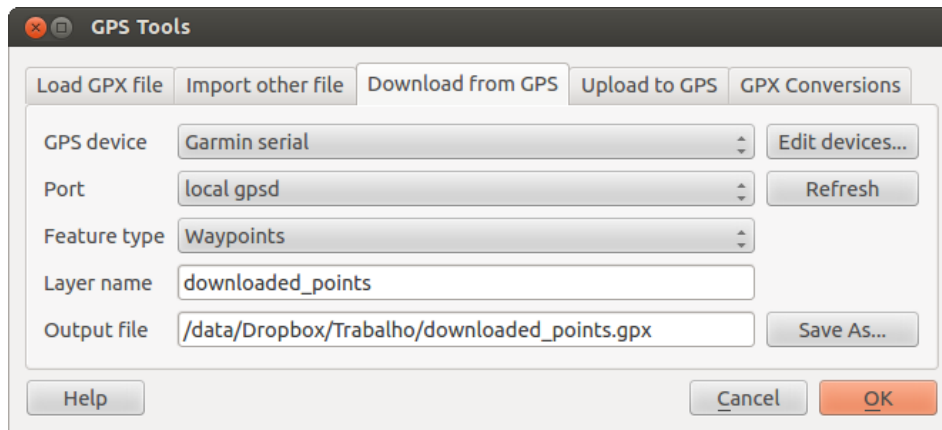


Figura 15.2: A ferramenta de transferência

-  No Windows é COM1 ou COM2.

Quando clica em [OK] os dados serão transferidos a partir do dispositivo e aparecerá como uma camada no QGIS.

15.1.6 Uploading os dados GPS para o dispositivo

Pode também fazer upload dos dados directamente de uma camada vectorial no QGIS para um dispositivo GPS usando o separador *Upload to GPS* da janela de Ferramentas GPS. Para fazer isto necessita simplesmente de seleccionar a camada que quer fazer o upload (que tem de ser uma camada GPX), o seu tipo de dispositivo GPS, e a porta (ou usb) a que está ligado. Como a ferramenta de transferência pode especificar o novo tipo de dispositivo se não estiver na lista de dispositivos.

Esta ferramenta é muito útil na combinação com as capacidades de edição vectorial do QGIS. Permite-nos carregar um mapa, criar sítios de interesse e rotas, e de seguida fazer um upload e usá-los no seu dispositivo GPS.

15.1.7 Definindo novos tipos de dispositivos

Existem vários diferentes tipos de dispositivos GPS. Os programadores QGIS não podem testá-los todos, portanto se tiver um que não funciona com nenhum tipo de dispositivo da lista das ferramentas *Transferir do GPS* e *Upload do GPS* você pode definir o seu próprio tipo de dispositivo. Você faz isto usando o editor de dispositivos GPS, onde pode começar clicando no botão [Editar dispositivos] nos separadores de transferência ou upload.

Para definir um novo dispositivo apenas clique no botão [Novo Dispositivo], introduza um nome, um comando de transferência e um comando de upload para o seu dispositivo, e clique no botão [Actualizar dispositivo]. O nome aparecerá na lista do menu de dispositivos na janela de upload e transferência, e pode ser qualquer cadeia de texto. O comando de transferência é o comando que é usado para transferir os dados a partir do dispositivo para um ficheiro GPX. Isto provavelmente será um comando GPSTool, mas pode usar qualquer outra linha de comando para criar um ficheiro GPX. O QGIS irá substituir as palavras-chaves %type, %in, e %out quando correr o comando.

%type será substituído por -w se está a transferir sítios de interesse, -r se está a transferir rotas e -t se está a transferir trilhos. Estes são as opções da linha de comandos que informam o GPSTool que tipo de elemento irá transferir.

%in será substituído por o nome da porta que escolheu na janela de transferência e o %out será substituído pelo nome que escolheu para o ficheiro GPX dos dados transferidos onde serão armazenados. Portanto se criar um tipo de dispositivo com o comando de transferência `gpsbabel %type -i garmin -o gpx %in %out` (isto actualmente é o comando de transferência para o tipos de dispositivo pré-definido 'Garmin Serial') e de seguida use-o para transferir os sítios de interesse a partir da porta `/dev/ttyS0` para o ficheiro `output.gpx`, o QGIS irá substituir as palavras-chave e correrá o comando `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx`.

O comando de upload é o comando que é usado para fazer upload de dados para o dispositivo. As mesmas palavras-chave são usadas, mas %in é agora substituída pelo o nome do ficheiro GPX para a camada que vai ser feita um upload, e %out é substituída pelo o nome da porta.

Pode aprender mais sobre o GPSTabel e as suas opções de linhas de comandos em <http://www.gpsbabel.org>.

Após ter criado um novo tipo de dispositivo irá aparecer na lista de dispositivos para as ferramentas de transferência e upload.

15.1.8 Transferência de pontos/trilhos a partir de unidades GPS

Como está descrito em secções anteriores do QGIS ele usa o GPSTabel para transferir pontos/trilhos directamente para o projecto. O QGIS abre uma caixa com o perfil pré-definido para transferir a partir de dispositivos Garmin. Infelizmente existe um erro que não permite criar outros perfis, portanto por agora só pode transferir directamente via USB com aparelhos Garmin usando as Ferramentas GPS.

Garmin GPSMAP 60cs

MS Windows

Instale os controladores USB do Garmin a partir do http://www8.garmin.com/support/download_details.jsp?id=591

Ligue o aparelho. Abra as Ferramentas GPS e use `type=garmin serial e port=usb`: Preencha os campos *Nome da camada* e *Ficheiro de saída*. Poderá ter problemas algumas vezes a guardar numa pasta, usando algo como `c:\temp` funcionará.

Ubuntu/Mint GNU/Linux

É preciso ter em conta as permissões do dispositivo, como é descrito em https://wiki.openstreetmap.org/wiki/USB_Garmin_on_GNU/Linux. Pode tentar criar um ficheiro `/etc/udev/rules.d/51-garmin.rules` contendo esta regra

```
ATTRS{idVendor}=="091e", ATTRS{idProduct}=="0003", MODE="666"
```

Depois disto é necessário ter a certeza que o módulo kernel do `garmin_gps` não está carregado

```
rmmod garmin_gps
```

e de seguida pode usar as Ferramentas GPS. Infelizmente, parece existir um [bug #7182](#) e normalmente o QGIS pára várias vezes antes da operação trabalhar bem.

BTGP-38KM datalogger (apenas com Bluetooth)

MS Windows

O erro atrás referido não permite que possa transferir os dados a partir do QGIS, por isso é necessário usar o GPSTabel a partir da linha de comandos ou usá-lo através da sua interacção. O comando que o executa é

```
gpsbabel -t -i skytraq,baud=9600,initbaud=9600 -f COM9 -o gpx -F C:/GPX/aaa.gpx
```

Ubuntu/Mint GNU/Linux

Use o mesmo comando (ou configurações se usa o GUI do GPSTabel) como no Windows. No Linux aparecerá uma mensagem como

```
skytraq: Too many read errors on serial port
```

É uma questão de desligá-lo e ligá-lo no datalogger e voltar a tentar.

BlueMax GPS-4044 datalogger (com BT e USB)

MS Windows

Note: É necessário instalar os controladores antes de usar o Windows 7. Veja o site do fabricante para a transferência correcta.

Transferindo com o GPSTabel, com o USB e BT retorna sempre um erro como

```
gpsbabel -t -i mtk -f COM12 -o gpx -F C:/temp/test.gpx
mtk_logger: Can't create temporary file data.bin
Error running gpsbabel: Process exited unsuccessfully with code 1
```

Ubuntu/Mint GNU/Linux

Com USB

Após ter ligado o cabo use o comando `dmesg` para perceber que porta está a ser usada, por exemplo `/dev/ttyACM3`. De seguida como é habitual use o GPSTabel a partir do CLI ou GUI


```
gpsbabel -t -i mtk -f /dev/ttyACM3 -o gpx -F /home/user/bluemax.gpx
```

Com Bluetooth





Use o Gestor de Dispositivos BlueMan para emparelhar os dispositivos e torná-los disponíveis através do sistema de portas, e de seguida corra o GPSTabel

```
gpsbabel -t -i mtk -f /dev/rfcomm0 -o gpx -F /home/user/bluemax_bt.gpx
```

15.2 Live GPS tracking

To activate live GPS tracking in QGIS, you need to select *Settings* → *Panels*  *GPS information*. You will get a new docked window on the left side of the canvas.


There are four possible screens in this GPS tracking window:

-  GPS position coordinates and an interface for manually entering vertices and features
-  GPS signal strength of satellite connections
-  GPS polar screen showing number and polar position of satellites
-  GPS options screen (see [figure_gps_options](#))

With a plugged-in GPS receiver (has to be supported by your operating system), a simple click on [**Connect**] connects the GPS to QGIS. A second click (now on [**Disconnect**]) disconnects the GPS receiver from your computer. For GNU/Linux, `gpsd` support is integrated to support connection to most GPS receivers. Therefore, you first have to configure `gpsd` properly to connect QGIS to it.

Warning: If you want to record your position to the canvas, you have to create a new vector layer first and switch it to editable status to be able to record your track.

15.2.1 Position and additional attributes

 If the GPS is receiving signals from satellites, you will see your position in latitude, longitude and altitude together with additional attributes.

GPS Information

Add Polygon

Add track point

Connect

Latitude

Longitude

Altitude

Time of fix

Speed

Direction

HDOP

VDOP

PDOP

H accuracy

V accuracy

Mode

Dimensions


Quality

Status

Satellites

Figura 15.3: GPS tracking position and additional attributes 🐧

15.2.2 GPS signal strength

 Here, you can see the signal strength of the satellites you are receiving signals from.

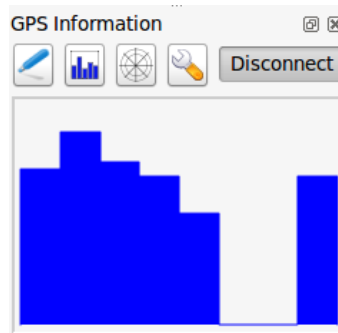



Figura 15.4: GPS tracking signal strength 

15.2.3 GPS polar window

 If you want to know where in the sky all the connected satellites are, you have to switch to the polar screen. You can also see the ID numbers of the satellites you are receiving signals from.

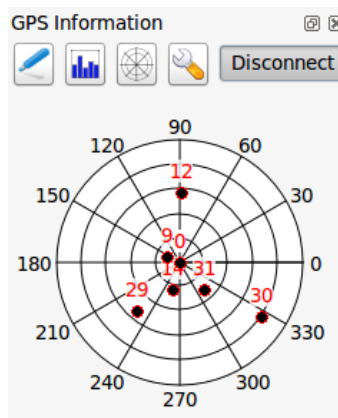




Figura 15.5: GPS tracking polar window 

15.2.4 GPS options

 In case of connection problems, you can switch between:

- *Autodetect*
- *Internal*
- *Serial device*
- *gpsd* (selecting the Host, Port and Device your GPS is connected to)

A click on [**Connect**] again initiates the connection to the GPS receiver.

You can activate *Automatically save added features* when you are in editing mode. Or you can activate *Automatically add points* to the map canvas with a certain width and color.

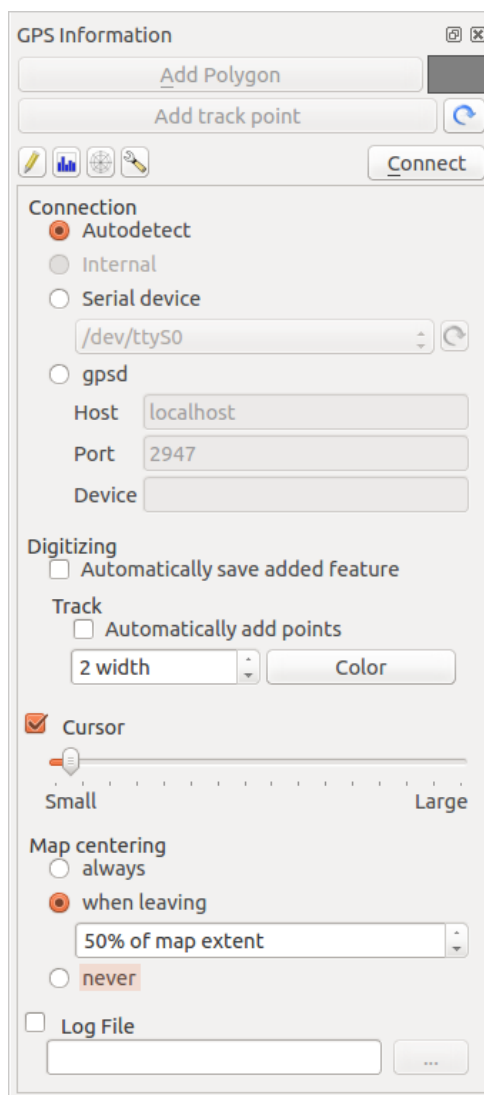




Figura 15.6: GPS tracking options window 

Activating *Cursor*, you can use a slider  to shrink and grow the position cursor on the canvas.

Activating *Map centering* allows you to decide in which way the canvas will be updated. This includes ‘always’, ‘when leaving’, if your recorded coordinates start to move out of the canvas, or ‘never’, to keep map extent.

Finally, you can activate *Log file* and define a path and a file where log messages about the GPS tracking are logged.

If you want to set a feature manually, you have to go back to  *Position* and click on [**Add Point**] or [**Add track point**].

15.2.5 Connect to a Bluetooth GPS for live tracking


With QGIS you can connect a Bluetooth GPS for field data collection. To perform this task you need a GPS Bluetooth device and a Bluetooth receiver on your computer.

At first you must let your GPS device be recognized and paired to the computer. Turn on the GPS, go to the Bluetooth icon on your notification area and search for a New Device.

On the right side of the Device selection mask make sure that all devices are selected so your GPS unit will probably appear among those available. In the next step a serial connection service should be available, select it and click on [**Configure**] button.

Remember the number of the COM port assigned to the GPS connection as resulting by the Bluetooth properties.

After the GPS has been recognized, make the pairing for the connection. Usually the authorization code is 0000.

Now open *GPS information* panel and switch to  *GPS options* screen. Select the COM port assigned to the GPS connection and click the [**Connect**]. After a while a cursor indicating your position should appear.

If QGIS can't receive GPS data, then you should restart your GPS device, wait 5-10 seconds then try to connect again. Usually this solution work. If you receive again a connection error make sure you don't have another Bluetooth receiver near you, paired with the same GPS unit.

15.2.6 Using GPSPMAP 60cs

MS Windows

Easiest way to make it work is to use a middleware (freeware, not open) called *GPSPGate*.

Launch the program, make it scan for GPS devices (works for both USB and BT ones) and then in QGIS just click [**Connect**] in the Live tracking panel using the *Autodetect* mode.

Ubuntu/Mint GNU/Linux

As for Windows the easiest way is to use a server in the middle, in this case *GPSPD*, so

```
sudo apt-get install gpsd
```

Then load the *garmin_gps* kernel module

```
sudo modprobe garmin_gps
```

And then connect the unit. Then check with *dmesg* the actual device being used by the unit, for example */dev/ttyUSB0*. Now you can launch *gpsd*

```
gpsd /dev/ttyUSB0
```


And finally connect with the QGIS live tracking tool.

15.2.7 Using BTGP-38KM datalogger (only Bluetooth)

Using GPSD (under Linux) or GPSTool (under Windows) is effortless.

15.2.8 Using BlueMax GPS-4044 datalogger (both BT and USB)

MS Windows

The live tracking works for both USB and BT modes, by using GPSTool or even without it, just use the  *Autodetect* mode, or point the tool the right port.

Ubuntu/Mint GNU/Linux

For USB

The live tracking works both with GPSD

```
gpsd /dev/ttyACM3
```

or without it, by connecting the QGIS live tracking tool directly to the device (for example `/dev/ttyACM3`).

For Bluetooth

The live tracking works both with GPSD











```
gpsd /dev/rfcomm0
```

or without it, by connecting the QGIS live tracking tool directly to the device (for example `/dev/rfcomm0`).


Integração GRASS SIG

The GRASS plugin provides access to GRASS GIS databases and functionalities (see GRASS-PROJECT in *Literatura e Referências Web*). This includes visualizing GRASS raster and vector layers, digitizing vector layers, editing vector attributes, creating new vector layers and analysing GRASS 2-D and 3-D data with more than 400 GRASS modules.

In this section, we'll introduce the plugin functionalities and give some examples of managing and working with GRASS data. The following main features are provided with the toolbar menu when you start the GRASS plugin, as described in section [sec_starting_grass](#):

-  Abrir conjunto de mapas
-  Novo conjunto de mapas
-  Fechar conjunto de mapas
-  Adicionar camada vectorial GRASS
-  Adicionar camada raster GRASS
-  Criar nova camada GRASS
-  Editar camada vectorial GRASS
-  Abrir ferramentas GRASS
-  Exibir a extensão actual do GRASS
-  Editar a extensão actual do GRASS








16.1 Iniciando o módulo GRASS

To use GRASS functionalities and/or visualize GRASS vector and raster layers in QGIS, you must select and load the GRASS plugin with the Plugin Manager. Therefore, go to the menu *Plugins* →  *Manage Plugins*, select *GRASS* and click [OK].

You can now start loading raster and vector layers from an existing GRASS LOCATION (see section [sec_load_grassdata](#)). Or, you can create a new GRASS LOCATION with QGIS (see section [Criando uma nova LOCALIZAÇÃO GRASS](#)) and import some raster and vector data (see section [Importando dados para uma LOCALIZAÇÃO GRASS](#)) for further analysis with the GRASS Toolbox (see section [The GRASS Toolbox](#)).

16.2 Carregando as camadas raster e vectoriais GRASS

With the GRASS plugin, you can load vector or raster layers using the appropriate button on the toolbar menu. As an example, we will use the QGIS Alaska dataset (see section *Amostra de Dados*). It includes a small sample GRASS LOCATION with three vector layers and one raster elevation map.

1. Create a new folder called `grassdata`, download the QGIS 'Alaska' dataset `qgis_sample_data.zip` from <http://download.osgeo.org/qgis/data/> and unzip the file into `grassdata`.
2. Start QGIS.
3. If not already done in a previous QGIS session, load the GRASS plugin clicking on *Plugins* →  *Manage Plugins* and activate  GRASS. The GRASS toolbar appears in the QGIS main window.
4. Na barra de ferramentas GRASS, clique no ícone  *Abrir conjunto de mapas* para iniciar o assistente de instalação do *CONJUNTO DE DADOS*.
5. For `gisdbase`, browse and select or enter the path to the newly created folder `grassdata`.
6. Poderá agora ser capaz de seleccionar a *LOCALIZAÇÃO*  `alaska` e o guilabel: *CONJUNTO DE DADOS*  `demo`.
7. Clique [OK]. Repare que algumas das ferramentas na barra de ferramentas GRASS que estão desactivadas agora estão activas.
8. Clique no  *Adicionar camada raster GRASS*, escolha o nome do mapa `gtopo30` e cliquem em [OK]. A camada de elevação irá ser visualizada.
9. Click on  *Add GRASS vector layer*, choose the map name `alaska` and click [OK]. The Alaska boundary vector layer will be overlaid on top of the `gtopo30` map. You can now adapt the layer properties as described in chapter *Janela das Propriedades da Camada Vectorial* (e.g., change opacity, fill and outline color).
10. Also load the other two vector layers, `rivers` and `airports`, and adapt their properties.

As you see, it is very simple to load GRASS raster and vector layers in QGIS. See the following sections for editing GRASS data and creating a new LOCATION. More sample GRASS LOCATIONS are available at the GRASS website at <http://grass.osgeo.org/download/sample-data/>.

Tip: Carregamento de Dados GRASS

If you have problems loading data or QGIS terminates abnormally, check to make sure you have loaded the GRASS plugin properly as described in section *Iniciando o módulo GRASS*.

16.3 LOCALIZAÇÃO GRASS e CONJUNTO DE MAPAS

GRASS data are stored in a directory referred to as GISDBASE. This directory, often called `grassdata`, must be created before you start working with the GRASS plugin in QGIS. Within this directory, the GRASS GIS data are organized by projects stored in subdirectories called *LOCATIONS*. Each *LOCATION* is defined by its coordinate system, map projection and geographical boundaries. Each *LOCATION* can have several *MAPSETS* (subdirectories of the *LOCATION*) that are used to subdivide the project into different topics or subregions, or as workspaces for individual team members (see Neteler & Mitsova 2008 in *Literatura e Referências Web*). In order to analyze vector and raster layers with GRASS modules, you must import them into a GRASS *LOCATION*. (This is not strictly true – with the GRASS modules `r.external` and `v.external` you can create read-only links to external GDAL/OGR-supported datasets without importing them. But because this is not the usual way for beginners to work with GRASS, this functionality will not be described here.)

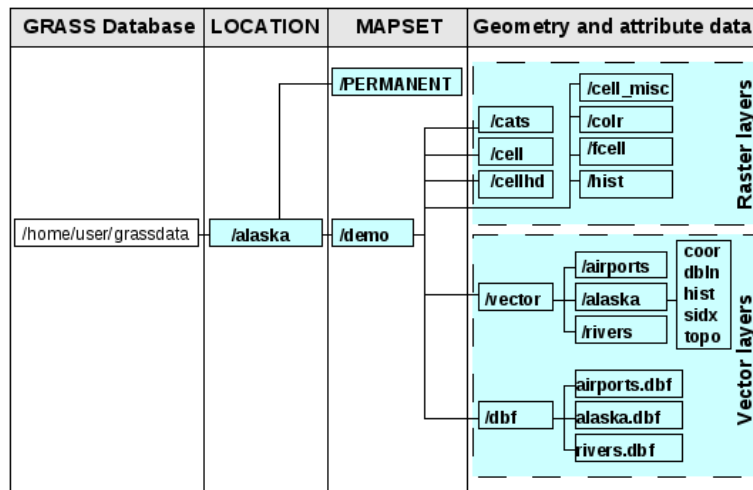




Figura 16.1: Dados GRASS na LOCALIZAÇÃO alaska

16.3.1 Criando uma nova LOCALIZAÇÃO GRASS

As an example, here is how the sample GRASS LOCATION alaska, which is projected in Albers Equal Area projection with unit feet was created for the QGIS sample dataset. This sample GRASS LOCATION alaska will be used for all examples and exercises in the following GRASS-related sections. It is useful to download and install the dataset on your computer (see *Amostra de Dados*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. Visualize the alaska.shp shapefile (see section *Loading a Shapefile*) from the QGIS Alaska dataset (see *Amostra de Dados*).
3. Na barra de ferramentas GRASS, clique no ícone  Novo conjunto de mapas para iniciar o assistente de instalação do CONJUNTO DE DADOS.
4. Select an existing GRASS database (GISDBASE) folder grassdata, or create one for the new LOCATION using a file manager on your computer. Then click [Next].
5. Nós podemos usar este assistente para criar um novo CONJUNTO DE MAPAS dentro de uma LOCALIZAÇÃO existente (veja secção *Adicionando um novo CONJUNTO DE MAPAS*) ou para criar juntamente um nova LOCALIZAÇÃO. Selecciona Criar nova localização (veja *figure_grass_location_2*).
6. Enter a name for the LOCATION – we used ‘alaska’ – and click [Next].
7. Defina a projecção clicando no botão Projecção para activar a lista de projecção.
8. We are using Albers Equal Area Alaska (feet) projection. Since we happen to know that it is represented by the EPSG ID 2964, we enter it in the search box. (Note: If you want to repeat this process for another LOCATION and projection and haven’t memorized the EPSG ID, click on the  CRS Status icon in the lower right-hand corner of the status bar (see section *Trabalhando com Projecções*)).
9. In Filter, insert 2964 to select the projection.
10. Clique [Seguinte].
11. To define the default region, we have to enter the LOCATION bounds in the north, south, east, and west directions. Here, we simply click on the button [Set current lqgl extent], to apply the extent of the loaded layer alaska.shp as the GRASS default region extent.
12. Clique [Seguinte].
13. We also need to define a MAPSET within our new LOCATION (this is necessary when creating a new LOCATION). You can name it whatever you like - we used ‘demo’. GRASS automatically creates a special

MAPSET called PERMANENT, designed to store the core data for the project, its default spatial extent and coordinate system definitions (see Neteler & Mitasova 2008 in *Literatura e Referências Web*).

14. Verifique o sumário para ter a certeza que está correcto e clique em [**Concluído**].
15. The new LOCATION, 'alaska', and two MAPSETs, 'demo' and 'PERMANENT', are created. The currently opened working set is 'demo', as you defined.
16. Repare que algumas das ferramentas na barra de ferramentas GRASS que estão desactivadas agora estão activas.

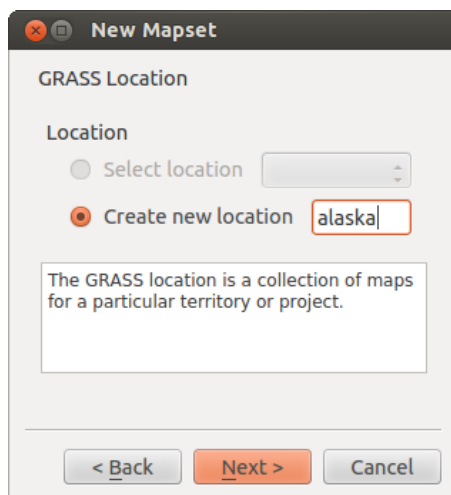



Figura 16.2: Criando uma nova LOCALIZAÇÃO GRASS ou um novo CONJUNTO DE MAPAS no QGIS

If that seemed like a lot of steps, it's really not all that bad and a very quick way to create a LOCATION. The LOCATION 'alaska' is now ready for data import (see section *Importando dados para uma LOCALIZAÇÃO GRASS*). You can also use the already-existing vector and raster data in the sample GRASS LOCATION 'alaska', included in the QGIS 'Alaska' dataset *Amostra de Dados*, and move on to section *O modelo de dados vectoriais do GRASS*.

16.3.2 Adicionando um novo CONJUNTO DE MAPAS

A user has write access only to a GRASS MAPSET he or she created. This means that besides access to your own MAPSET, you can read maps in other users' MAPSETs (and they can read yours), but you can modify or remove only the maps in your own MAPSET.



All MAPSETs include a WIND file that stores the current boundary coordinate values and the currently selected raster resolution (see Neteler & Mitasova 2008 in *Literatura e Referências Web*, and section *A ferramenta da região GRASS*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. Na barra de ferramentas GRASS, clique no ícone  Novo conjunto de mapas para iniciar o assistente de instalação do *CONJUNTO DE DADOS*.
3. Select the GRASS database (GISDBASE) folder `grassdata` with the LOCATION 'alaska', where we want to add a further MAPSET called 'test'.
4. Clique [**Seguinte**].
5. Podemos usar este assistente para criar um novo CONJUNTO DE MAPAS dentro de uma LOCALIZAÇÃO existente ou criar uma nova LOCALIZAÇÃO tudo junto. Clique no botão de rádio *Seleccionar localização* (veja *figure_grass_location_2*) e clique [**Próximo**].
6. Enter the name `test` for the new MAPSET. Below in the wizard, you see a list of existing MAPSETs and corresponding owners.

7. Clique [**Seguinte**], e verifique o sumário para ter a certeza que está tudo correcto e clique em [**Concluído**].

16.4 Importando dados para uma LOCALIZAÇÃO GRASS

This section gives an example of how to import raster and vector data into the ‘alaska’ GRASS LOCATION provided by the QGIS ‘Alaska’ dataset. Therefore, we use the landcover raster map `landcover.img` and the vector GML file `lakes.gml` from the QGIS ‘Alaska’ dataset (see *Amostra de Dados*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. Na barra de ferramentas GRASS, clique no ícone  `Abrir CONJUNTO DE MAPAS` para trazer o assistente de `CONJUNTO DE MAPAS`.
3. Select as GRASS database the folder `grassdata` in the QGIS Alaska dataset, as LOCATION ‘alaska’, as MAPSET ‘demo’ and click [**OK**].
4. Agora clique no ícone  `Abrir ferramentas GRASS`. O diálogo da Caixa de Ferramentas GRASS (veja secção *The GRASS Toolbox*) aparece.
5. To import the raster map `landcover.img`, click the module `r.in.gdal` in the *Modules Tree* tab. This GRASS module allows you to import GDAL-supported raster files into a GRASS LOCATION. The module dialog for `r.in.gdal` appears.
6. Browse to the folder `raster` in the QGIS ‘Alaska’ dataset and select the file `landcover.img`.
7. As raster output name, define `landcover_grass` and click [**Run**]. In the *Output* tab, you see the currently running GRASS command `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass`.
8. When it says **Successfully finished**, click [**View output**]. The `landcover_grass` raster layer is now imported into GRASS and will be visualized in the QGIS canvas.
9. To import the vector GML file `lakes.gml`, click the module `v.in.ogr` in the *Modules Tree* tab. This GRASS module allows you to import OGR-supported vector files into a GRASS LOCATION. The module dialog for `v.in.ogr` appears.
10. Browse to the folder `gml` in the QGIS ‘Alaska’ dataset and select the file `lakes.gml` as OGR file.
11. As vector output name, define `lakes_grass` and click [**Run**]. You don’t have to care about the other options in this example. In the *Output* tab you see the currently running GRASS command `v.in.ogr -o dsn=/path/to/lakes.gml output=lakes_grass`.
12. When it says **Successfully finished**, click [**View output**]. The `lakes_grass` vector layer is now imported into GRASS and will be visualized in the QGIS canvas.

16.5 O modelo de dados vectoriais do GRASS

É importante perceber previamente o modelo de dados vectorial GRASS antes da digitalização.

O GRASS usa um modelo topológico vectorial.

Isto significa que as áreas não estão representadas como polígonos fechados, mas por um ou mais limites. Um limite entre duas áreas adjacentes é digitalizada apenas uma vez, e é partilhada por ambas as áreas. Os limites devem estar ligados e fechados sem buracos. Uma área é identificada (e rotulada) pelo **centróide** da área.

Besides boundaries and centroids, a vector map can also contain points and lines. All these geometry elements can be mixed in one vector and will be represented in different so-called ‘layers’ inside one GRASS vector map. So in GRASS, a layer is not a vector or raster map but a level inside a vector layer. This is important to distinguish carefully. (Although it is possible to mix geometry elements, it is unusual and, even in GRASS, only used in special cases such as vector network analysis. Normally, you should prefer to store different geometry elements in different layers.)

It is possible to store several ‘layers’ in one vector dataset. For example, fields, forests and lakes can be stored in one vector. An adjacent forest and lake can share the same boundary, but they have separate attribute tables. It is also possible to attach attributes to boundaries. An example might be the case where the boundary between a lake and a forest is a road, so it can have a different attribute table.

The ‘layer’ of the feature is defined by the ‘layer’ inside GRASS. ‘Layer’ is the number which defines if there is more than one layer inside the dataset (e.g., if the geometry is forest or lake). For now, it can be only a number. In the future, GRASS will also support names as fields in the user interface.

Attributes can be stored inside the GRASS LOCATION as dBase or SQLite3 or in external database tables, for example, PostgreSQL, MySQL, Oracle, etc.


Os atributos nas tabelas da base de dados estão ligados aos elementos de geometria usando o valor ‘categoria’.

‘Categoria’ (chave, ID) é um inteiro anexado às primitivas da geometria, e é usado como ligação a uma coluna de chave na tabela da base de dados.

Tip: Aprendendo o Modelo Vectorial GRASS

The best way to learn the GRASS vector model and its capabilities is to download one of the many GRASS tutorials where the vector model is described more deeply. See <http://grass.osgeo.org/documentation/manuals/> for more information, books and tutorials in several languages.

16.6 Criando uma nova camada vectorial GRASS


To create a new GRASS vector layer with the GRASS plugin, click the  Create new GRASS vector toolbar icon. Enter a name in the text box, and you can start digitizing point, line or polygon geometries following the procedure described in section *Digitalizando e editando as camadas vectoriais GRASS*.

In GRASS, it is possible to organize all sorts of geometry types (point, line and area) in one layer, because GRASS uses a topological vector model, so you don’t need to select the geometry type when creating a new GRASS vector. This is different from shapefile creation with QGIS, because shapefiles use the Simple Feature vector model (see section *Criando novas camadas Vectoriais*).

Tip: Criando uma tabela de atributos para uma nova camada vectorial GRASS

Se desejar atribuir atributos aos seus elementos de geometria digitalizados, tenha a certeza que criou uma tabela de atributos com as colunas antes de começar a digitalizar (veja [figure_grass_digitizing_5](#)).

16.7 Digitalizando e editando as camadas vectoriais GRASS

The digitizing tools for GRASS vector layers are accessed using the  Edit GRASS vector layer icon on the toolbar. Make sure you have loaded a GRASS vector and it is the selected layer in the legend before clicking on the edit tool. Figure [figure_grass_digitizing_2](#) shows the GRASS edit dialog that is displayed when you click on the edit tool. The tools and settings are discussed in the following sections.

Tip: Digitalizando polígonos no GRASS

If you want to create a polygon in GRASS, you first digitize the boundary of the polygon, setting the mode to ‘No category’. Then you add a centroid (label point) into the closed boundary, setting the mode to ‘Next not used’. The reason for this is that a topological vector model links the attribute information of a polygon always to the centroid and not to the boundary.

Barra de Ferramentas

In [figure_grass_digitizing_1](#), you see the GRASS digitizing toolbar icons provided by the GRASS plugin. Table [table_grass_digitizing_1](#) explains the available functionalities.



Figura 16.3: Barra de Ferramentas Digitalização GRASS

Ícone	Ferramenta	Finalidade
	Novo Ponto	Digitalizar um novo ponto
	Nova Linha	Digitalizar nova linha
	Novo Limite	Digitalizar novo limite (finalizar seleccionando uma nova ferramenta)
	Novo Centróide	Digitalizar um novo centróide (rótulo com a área existente)
	Mover vértice	Mover um vértice de uma linha existente ou limite e identificar nova posição
	Adicionar vértice	Adicionar um novo vértice a uma linha existente
	Apagar vértice	Apagar vértice de uma linha existente (confirme o vértice seleccionado clicando com outro clique)
	Mover elemento	Mover o limite seleccionado, linha, ponto ou centróide e clique na nova posição
	Dividir linha	Split an existing line into two parts
	Apagar elemento	Apagar um limite existente, linha, ponto ou centróide (confirme o elemento seleccionado com outro clique)
	Editar atributos	Editar os atributos do elemento seleccionado (note que um elemento pode representar mais elementos, veja acima)
	Fechar	Feche a sessão e guarde o estado actual (reconstrução da topologia depois)

Tabela GRASS Digitalização 1: Ferramentas de Digitalização GRASS

Separador Categoria

O separador *Categoria* permite definir a forma de como os valores categoria serão atribuídos a um novo elemento de geometria.

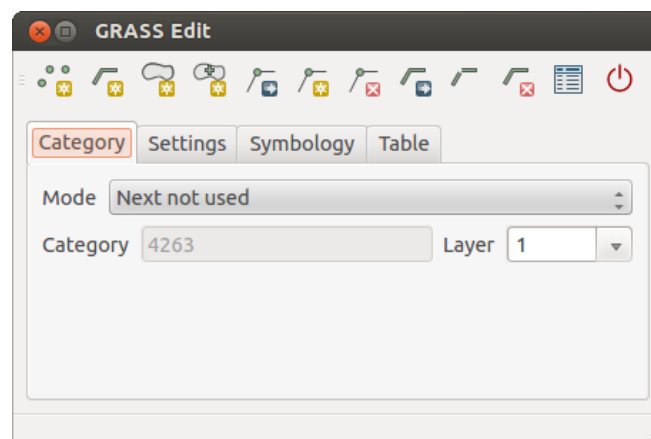


Figura 16.4: Separador de Digitalização de Categorias

- **Mode:** The category value that will be applied to new geometry elements.
 - Next not used - Apply next not yet used category value to geometry element.

- Manual entry - Manually define the category value for the geometry element in the 'Category' entry field.
- No category - Do not apply a category value to the geometry element. This is used, for instance, for area boundaries, because the category values are connected via the centroid.
- **Category** - The number (ID) that is attached to each digitized geometry element. It is used to connect each geometry element with its attributes.
- **Field (layer)** - Each geometry element can be connected with several attribute tables using different GRASS geometry layers. The default layer number is 1.

Tip: Creating an additional GRASS 'layer' with lqgl

If you would like to add more layers to your dataset, just add a new number in the 'Field (layer)' entry box and press return. In the Table tab, you can create your new table connected to your new layer.

Separador das Configurações

The *Settings* tab allows you to set the snapping in screen pixels. The threshold defines at what distance new points or line ends are snapped to existing nodes. This helps to prevent gaps or dangles between boundaries. The default is set to 10 pixels.

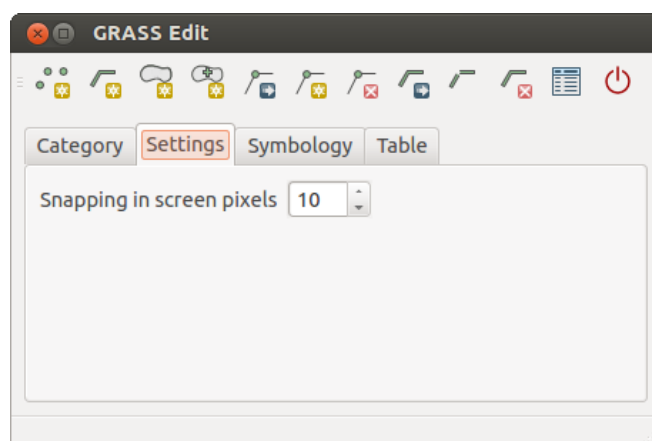


Figura 16.5: Separador de Configurações de Digitalização GRASS

Separador da Simbologia

The *Symbology* tab allows you to view and set symbology and color settings for various geometry types and their topological status (e.g., closed / opened boundary).

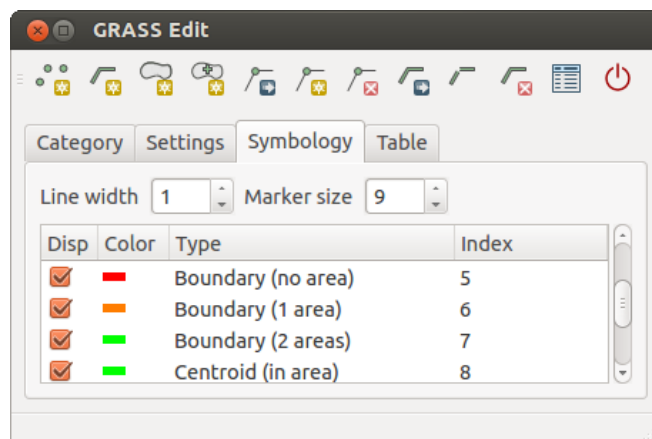


Figura 16.6: GRASS Digitizing Symbology Tab

Separador Tabela

The *Table* tab provides information about the database table for a given 'layer'. Here, you can add new columns to an existing attribute table, or create a new database table for a new GRASS vector layer (see section *Criando uma nova camada vectorial GRASS*).

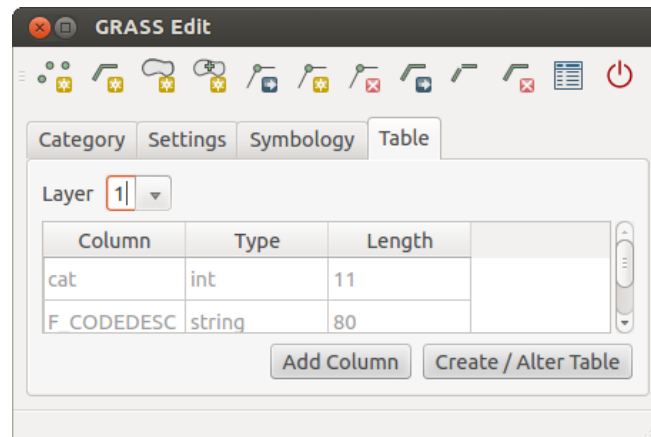



Figura 16.7: Separador de Digitalização da Tabela


Tip: Editar Permissões GRASS

You must be the owner of the GRASS MAPSET you want to edit. It is impossible to edit data layers in a MAPSET that is not yours, even if you have write permission.

16.8 A ferramenta da região GRASS


The region definition (setting a spatial working window) in GRASS is important for working with raster layers. Vector analysis is by default not limited to any defined region definitions. But all newly created rasters will have the spatial extension and resolution of the currently defined GRASS region, regardless of their original extension and resolution. The current GRASS region is stored in the `$LOCATION/$MAPSET/WIND` file, and it defines north, south, east and west bounds, number of columns and rows, horizontal and vertical spatial resolution.

It is possible to switch on and off the visualization of the GRASS region in the QGIS canvas using the  `Display current GRASS region` button.

With the  `Edit current GRASS region` icon, you can open a dialog to change the current region and the symbology of the GRASS region rectangle in the QGIS canvas. Type in the new region bounds and resolution, and click **[OK]**. The dialog also allows you to select a new region interactively with your mouse on the QGIS canvas. Therefore, click with the left mouse button in the QGIS canvas, open a rectangle, close it using the left mouse button again and click **[OK]**.

The GRASS module `g.region` provides a lot more parameters to define an appropriate region extent and resolution for your raster analysis. You can use these parameters with the GRASS Toolbox, described in section *The GRASS Toolbox*.

16.9 The GRASS Toolbox

The  `Open GRASS Tools` box provides GRASS module functionalities to work with data inside a selected GRASS LOCATION and MAPSET. To use the GRASS Toolbox you need to open a LOCATION and MAPSET that you have write permission for (usually granted, if you created the MAPSET). This is necessary, because new raster or vector layers created during analysis need to be written to the currently selected LOCATION and MAPSET.

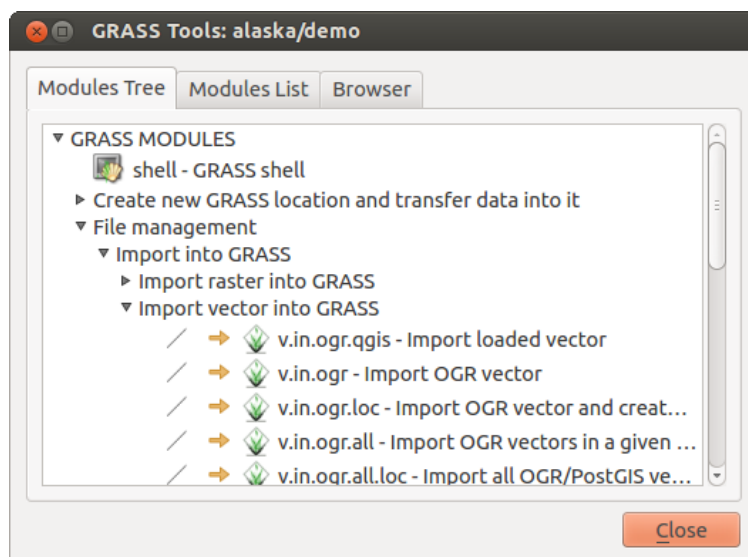


Figura 16.8: Árvore de Módulos e Caixa de Ferramentas GRASS 

16.9.1 Trabalhando com os módulos do GRASS

The GRASS shell inside the GRASS Toolbox provides access to almost all (more than 300) GRASS modules in a command line interface. To offer a more user-friendly working environment, about 200 of the available GRASS modules and functionalities are also provided by graphical dialogs within the GRASS plugin Toolbox.

A complete list of GRASS modules available in the graphical Toolbox in QGIS version 2.6 is available in the GRASS wiki at http://grass.osgeo.org/wiki/GRASS-QGIS_relevant_module_list.

It is also possible to customize the GRASS Toolbox content. This procedure is described in section *Personalizar a Caixa de Ferramentas GRASS*.

As shown in [figure_grass_toolbox_1](#), you can look for the appropriate GRASS module using the thematically grouped *Modules Tree* or the searchable *Modules List* tab.

By clicking on a graphical module icon, a new tab will be added to the Toolbox dialog, providing three new sub-tabs: *Options*, *Output* and *Manual*.

Opções

The *Options* tab provides a simplified module dialog where you can usually select a raster or vector layer visualized in the QGIS canvas and enter further module-specific parameters to run the module.

The provided module parameters are often not complete to keep the dialog clear. If you want to use further module parameters and flags, you need to start the GRASS shell and run the module in the command line.

A new feature since QGIS 1.8 is the support for a *Show Advanced Options* button below the simplified module dialog in the *Options* tab. At the moment, it is only added to the module `v.in.ascii` as an example of use, but it will probably be part of more or all modules in the GRASS Toolbox in future versions of QGIS. This allows you to use the complete GRASS module options without the need to switch to the GRASS shell.

Ficheiro de Saída

The *Output* tab provides information about the output status of the module. When you click the **[Run]** button, the module switches to the *Output* tab and you see information about the analysis process. If all works well, you will finally see a `Successfully finished` message.

Manual

The *Manual* tab shows the HTML help page of the GRASS module. You can use it to check further module parameters and flags or to get a deeper knowledge about the purpose of the module. At the end of each module manual page, you see further links to the `Main Help index`, the `Thematic index` and the `Full index`. These links provide the same information as the module `g.manual`.

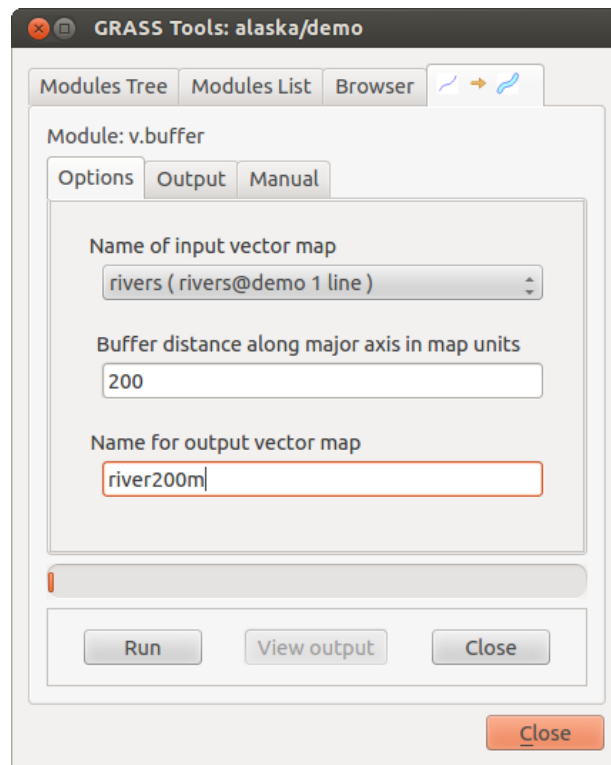


Figura 16.9: Opções dos Módulos da Caixa de Ferramentas do GRASS 🐧

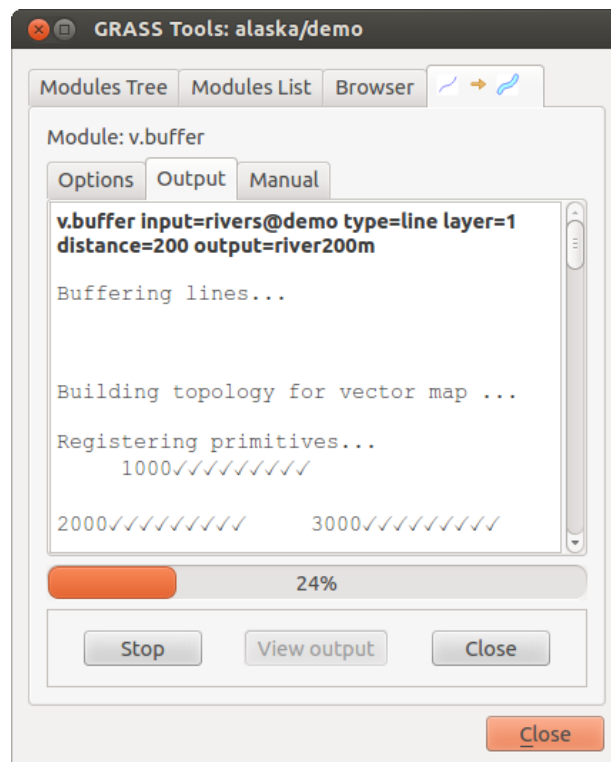


Figura 16.10: GRASS Toolbox Module Output 🐧

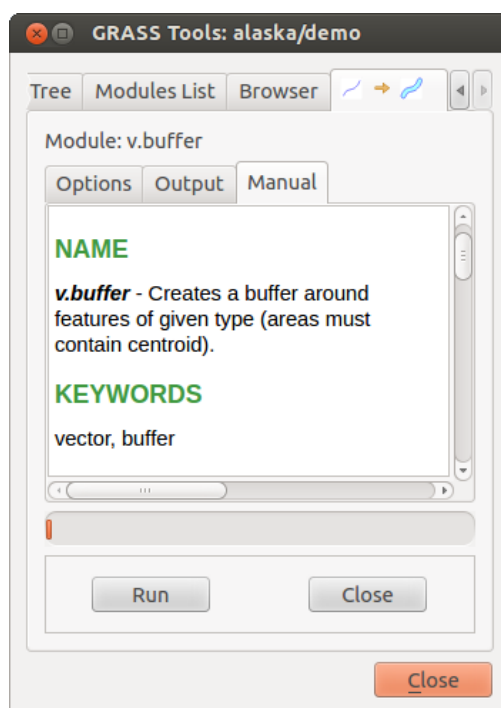


Figura 16.11: GRASS Toolbox Module Manual 

Tip: Exibir os resultados imediatamente




Se quiser exibir os seus resultados do cálculo imediatamente no seu enquadramento do mapa, pode usar o botão ‘Ver ficheiro de saída’ no fundo do separador do módulo.

16.9.2 Exemplos de módulos GRASS

Os seguintes exemplos irão demonstrar o poder de alguns módulos GRASS.

Criando linhas de contorno

The first example creates a vector contour map from an elevation raster (DEM). Here, it is assumed that you have the Alaska LOCATION set up as explained in section *Importando dados para uma LOCALIZAÇÃO GRASS*.

- First, open the location by clicking the  Open mapset button and choosing the Alaska location.
- Now load the gtopo30 elevation raster by clicking  Add GRASS raster layer and selecting the gtopo30 raster from the demo location.
- Now open the Toolbox with the  Open GRASS tools button.
- In the list of tool categories, double-click *Raster* → *Surface Management* → *Generate vector contour lines*.
- Now a single click on the tool **r.contour** will open the tool dialog as explained above (see *Trabalhando com os módulos do GRASS*). The gtopo30 raster should appear as the *Name of input raster*.
- Type into the *Increment between Contour levels* the value 100. (This will create contour lines at intervals of 100 meters.)
- Type into the *Name for output vector map* the name `ctour_100`.

- Click **[Run]** to start the process. Wait for several moments until the message `Successfully finished` appears in the output window. Then click **[View Output]** and **[Close]**.

Since this is a large region, it will take a while to display. After it finishes rendering, you can open the layer properties window to change the line color so that the contours appear clearly over the elevation raster, as in *Janela das Propriedades da Camada Vectorial*.

Next, zoom in to a small, mountainous area in the center of Alaska. Zooming in close, you will notice that the contours have sharp corners. GRASS offers the **v.generalize** tool to slightly alter vector maps while keeping their overall shape. The tool uses several different algorithms with different purposes. Some of the algorithms (i.e., Douglas Peucker and Vertex Reduction) simplify the line by removing some of the vertices. The resulting vector will load faster. This process is useful when you have a highly detailed vector, but you are creating a very small-scale map, so the detail is unnecessary.

Tip: **A ferramenta de simplificação

Note that the QGIS fTools plugin has a *Simplify geometries* → tool that works just like the GRASS **v.generalize** Douglas-Peucker algorithm.

However, the purpose of this example is different. The contour lines created by `r.contour` have sharp angles that should be smoothed. Among the **v.generalize** algorithms, there is Chaiken's, which does just that (also Hermite splines). Be aware that these algorithms can **add** additional vertices to the vector, causing it to load even more slowly.

- Open the GRASS Toolbox and double-click the categories *Vector* → *Develop map* → *Generalization*, then click on the **v.generalize** module to open its options window.
- Check that the 'ctour_100' vector appears as the *Name of input vector*.
- From the list of algorithms, choose Chaiken's. Leave all other options at their default, and scroll down to the last row to enter in the field *Name for output vector map* 'ctour_100_smooth', and click **[Run]**.
- The process takes several moments. Once `Successfully finished` appears in the output windows, click **[View output]** and then **[Close]**.
- You may change the color of the vector to display it clearly on the raster background and to contrast with the original contour lines. You will notice that the new contour lines have smoother corners than the original while staying faithful to the original overall shape.

Tip: Outros usos para o `r.contour`

The procedure described above can be used in other equivalent situations. If you have a raster map of precipitation data, for example, then the same method will be used to create a vector map of isohyetal (constant rainfall) lines.

Creating a Hillshade 3-D effect

Several methods are used to display elevation layers and give a 3-D effect to maps. The use of contour lines, as shown above, is one popular method often chosen to produce topographic maps. Another way to display a 3-D effect is by hillshading. The hillshade effect is created from a DEM (elevation) raster by first calculating the slope and aspect of each cell, then simulating the sun's position in the sky and giving a reflectance value to each cell. Thus, you get sun-facing slopes lighted; the slopes facing away from the sun (in shadow) are darkened.

- Begin this example by loading the `gtopo30` elevation raster. Start the GRASS Toolbox, and under the Raster category, double-click to open *Spatial analysis* → *Terrain analysis*.
- De seguida clique em **r.shaded.relief** para abrir o módulo.
- Altere o *ângulo do azimuth* 270 para 315.
- Introduza `gtopo30_shade` para o novo raster de ensombramento, e clique **[Executar]**.
- Quando o processo concluir, adicione o raster de ensombramento ao mapa, Deverá vê-lo numa escala de cinzentos.

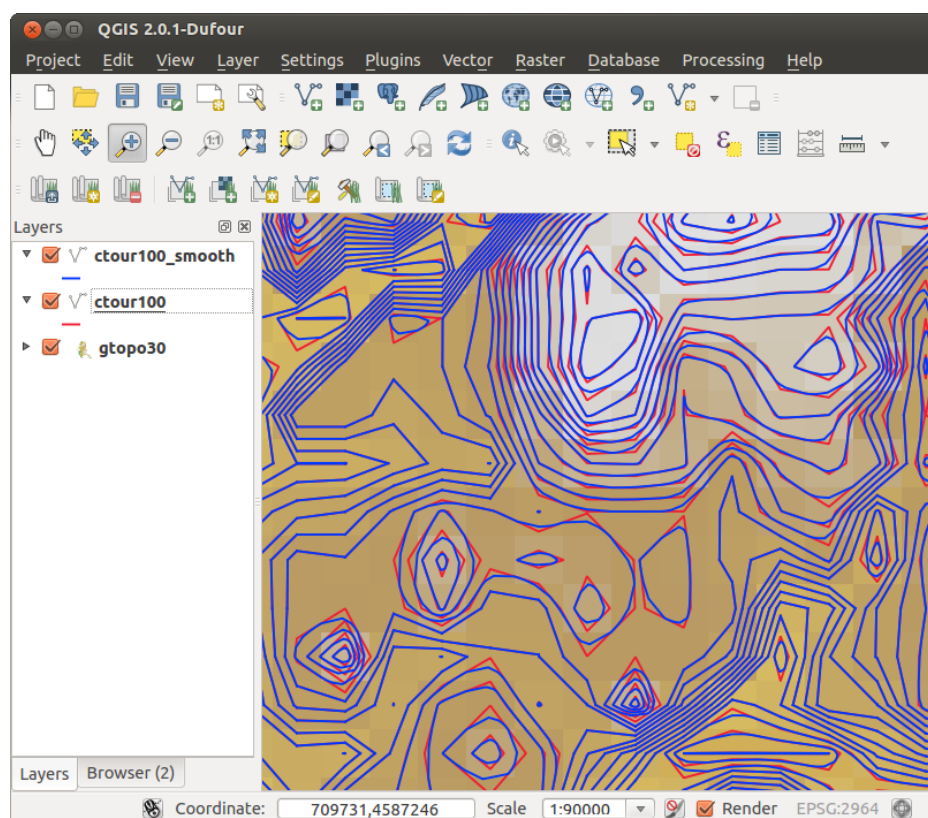



Figura 16.12: GRASS module v.generalize to smooth a vector map 

- To view both the hillshading and the colors of the `gtopo30` together, move the hillshade map below the `gtopo30` map in the table of contents, then open the *Properties* window of `gtopo30`, switch to the *Transparency* tab and set its transparency level to about 25%.

You should now have the `gtopo30` elevation with its colormap and transparency setting displayed **above** the grayscale hillshade map. In order to see the visual effects of the hillshading, turn off the `gtopo30_shade` map, then turn it back on.

Usando a linha de comandos do GRASS

The GRASS plugin in QGIS is designed for users who are new to GRASS and not familiar with all the modules and options. As such, some modules in the Toolbox do not show all the options available, and some modules do not appear at all. The GRASS shell (or console) gives the user access to those additional GRASS modules that do not appear in the Toolbox tree, and also to some additional options to the modules that are in the Toolbox with the simplest default parameters. This example demonstrates the use of an additional option in the **r.shaded.relief** module that was shown above.

The module **r.shaded.relief** can take a parameter `zmult`, which multiplies the elevation values relative to the X-Y coordinate units so that the hillshade effect is even more pronounced.

- Load the `gtopo30` elevation raster as above, then start the GRASS Toolbox and click on the GRASS shell. In the shell window, type the command `r.shaded.relief map=gtopo30 shade=gtopo30_shade2 azimuth=315 zmult=3` and press **[Enter]**.
- After the process finishes, shift to the *Browse* tab and double-click on the new `gtopo30_shade2` raster to display it in QGIS.
- As explained above, move the shaded relief raster below the `gtopo30` raster in the table of contents, then check the transparency of the colored `gtopo30` layer. You should see that the 3-D effect stands out more strongly compared with the first shaded relief map.

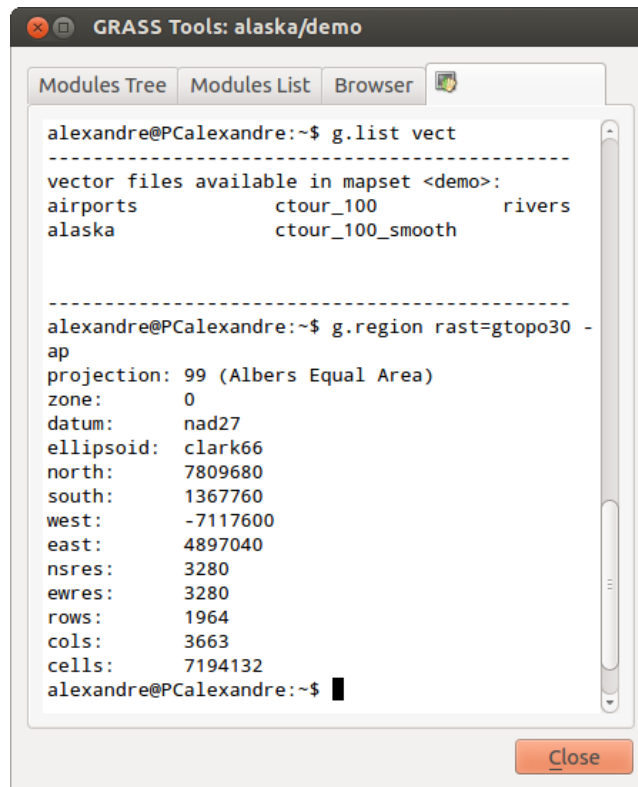



Figure 16.13: The GRASS shell, r.shaded.relief module 

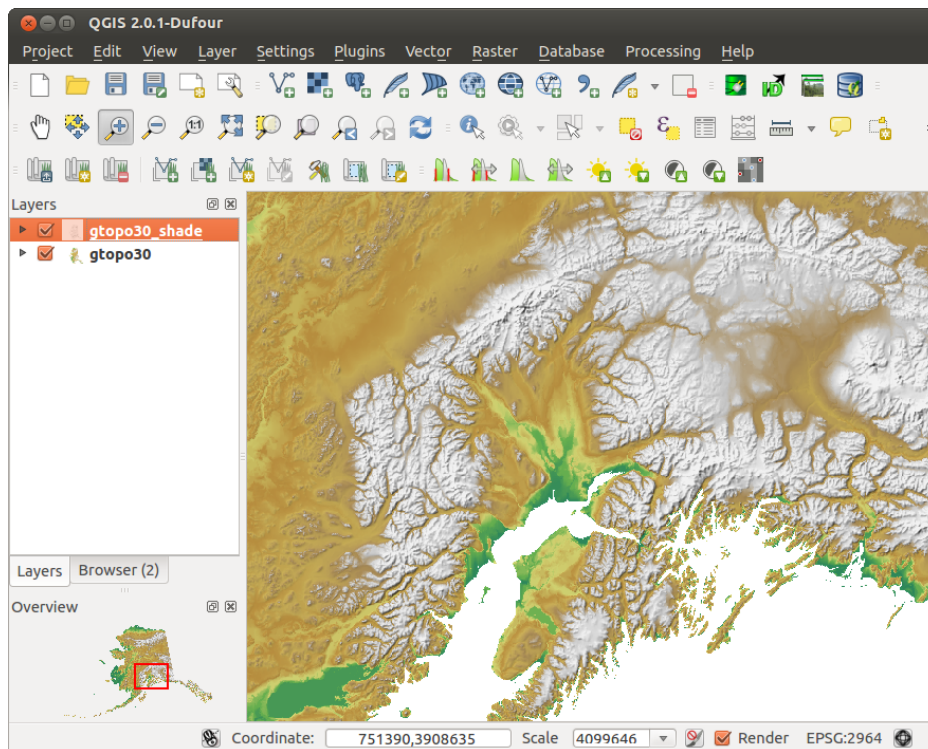



Figure 16.14: Displaying shaded relief created with the GRASS module r.shaded.relief 

Estatísticas Raster num mapa vectorial

The next example shows how a GRASS module can aggregate raster data and add columns of statistics for each polygon in a vector map.







- Again using the Alaska data, refer to *Importando dados para uma LOCALIZAÇÃO GRASS* to import the trees shapefile from the `shapefiles` directory into GRASS.
- Now an intermediate step is required: centroids must be added to the imported trees map to make it a complete GRASS area vector (including both boundaries and centroids).
- From the Toolbox, choose *Vector* → *Manage features*, and open the module **v.centroids**.
- Introduza como *mapa vectorial de saída* ‘forest_areas’ e execute o módulo.
- Now load the `forest_areas` vector and display the types of forests - deciduous, evergreen, mixed - in different colors: In the layer *Properties* window, *Symbology* tab, choose from *Legend type*  ‘Unique value’ and set the *Classification field* to ‘VEGDESC’. (Refer to the explanation of the symbology tab in *Estilos* of the vector section.)
- Next, reopen the GRASS Toolbox and open *Vector* → *Vector update* by other maps.
- Click on the **v.rast.stats** module. Enter `gtopo30` and `forest_areas`.
- Only one additional parameter is needed: Enter *column prefix* `elev`, and click **[Run]**. This is a computationally heavy operation, which will run for a long time (probably up to two hours).
- Finally, open the `forest_areas` attribute table, and verify that several new columns have been added, including `elev_min`, `elev_max`, `elev_mean`, etc., for each forest polygon.



16.9.3 Trabalhando com a pesquisa da LOCALIZAÇÃO GRASS

Another useful feature inside the GRASS Toolbox is the GRASS LOCATION browser. In *figure_grass_module_7*, you can see the current working LOCATION with its MAPSETS.

In the left browser windows, you can browse through all MAPSETS inside the current LOCATION. The right browser window shows some meta-information for selected raster or vector layers (e.g., resolution, bounding box, data source, connected attribute table for vector data, and a command history).

The toolbar inside the *Browser* tab offers the following tools to manage the selected LOCATION:

-  *Adicionar mapa seleccionado ao enquadramento*
-  *Copiar mapa seleccionado*
-  *Renomear mapa seleccionado*
-  *Apagar mapa seleccionado*
-  *Definir a região actual do mapa seleccionado*
-  *Actualizar a janela de pesquisa*

The  *Rename selected map* and  *Delete selected map* only work with maps inside your currently selected MAPSET. All other tools also work with raster and vector layers in another MAPSET.

16.9.4 Personalizar a Caixa de Ferramentas GRASS

Nearly all GRASS modules can be added to the GRASS Toolbox. An XML interface is provided to parse the pretty simple XML files that configure the modules’ appearance and parameters inside the Toolbox.

A sample XML file for generating the module `v.buffer` (`v.buffer.qgm`) looks like this:

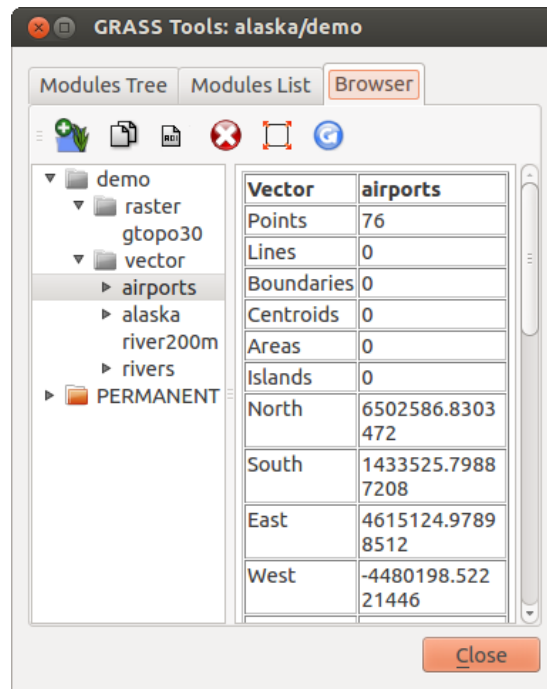


Figura 16.15: Pesquisa na LOCALIZAÇÃO GRASS 🐧

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE qgisgrassmodule SYSTEM "http://mrcc.com/qgisgrassmodule.dtd">

<qgisgrassmodule label="Vector buffer" module="v.buffer">
  <option key="input" typeoption="type" layeroption="layer" />
  <option key="buffer"/>
  <option key="output" />
</qgisgrassmodule>
```

The parser reads this definition and creates a new tab inside the Toolbox when you select the module. A more detailed description for adding new modules, changing a module's group, etc., can be found on the QGIS wiki at http://hub.qgis.org/projects/quantum-gis/wiki/Adding_New_Tools_to_the_GRASS_Toolbox.

Infraestrutura do Processamento QGIS

17.1 Introdução

This chapter introduces the QGIS processing framework, a geoprocessing environment that can be used to call native and third-party algorithms from QGIS, making your spatial analysis tasks more productive and easy to accomplish.

In the following sections, we will review how to use the graphical elements of this framework and make the most out of each one of them.

There are four basic elements in the framework GUI, which are used to run algorithms for different purposes. Choosing one tool or another will depend on the kind of analysis that is to be performed and the particular characteristics of each user and project. All of them (except for the batch processing interface, which is called from the toolbox, as we will see) can be accessed from the *Processing* menu item. (You will see more than four entries. The remaining ones are not used to execute algorithms and will be explained later in this chapter.)

- A caixa de ferramentas. O elemento principal do GUI, é usado para executar um algoritmo único ou correr um processo batch baseado nesse algoritmo.

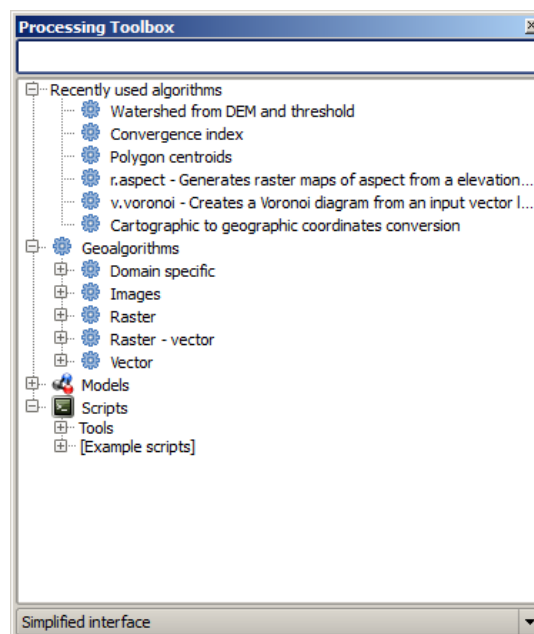


Figura 17.1: Caixa de Ferramentas Processamento 

- The graphical modeler. Several algorithms can be combined graphically using the modeler to define a workflow, creating a single process that involves several subprocesses.

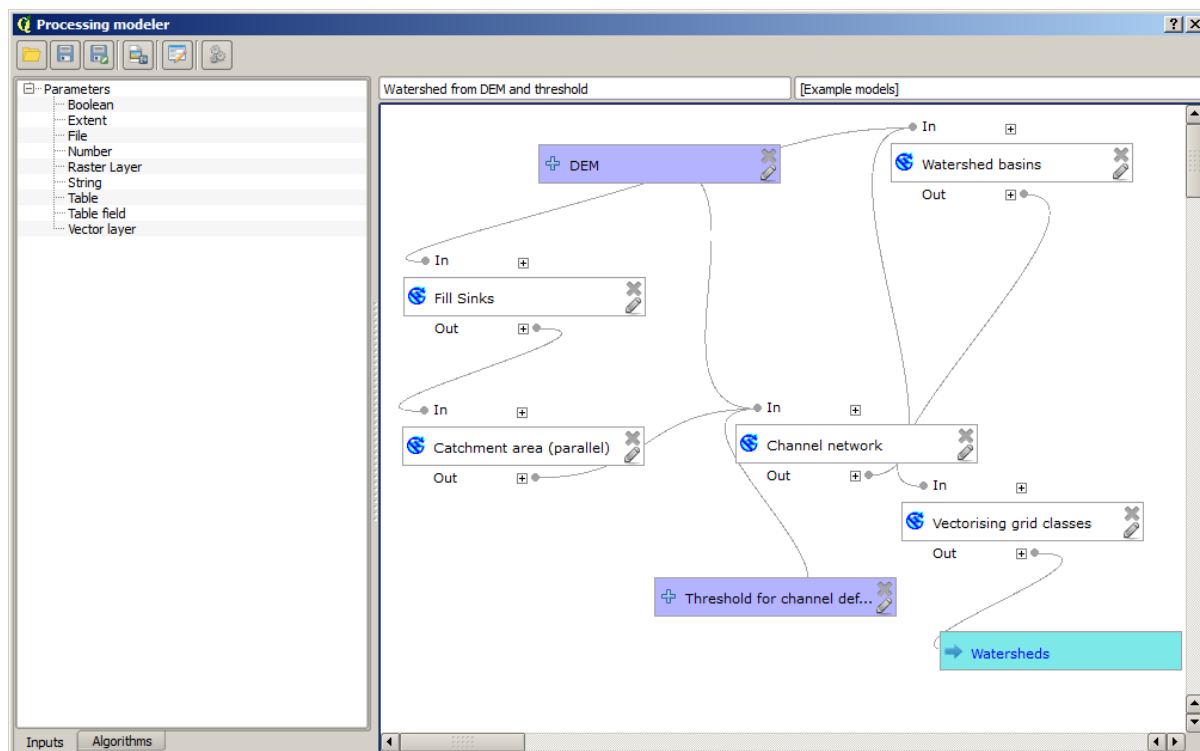


Figura 17.2: Modelador do Processamento

- The history manager. All actions performed using any of the aforementioned elements are stored in a history file and can be later easily reproduced using the history manager.
- A interface de processamento batch. Esta interface permite que possa executar processos batch e automaticamente a execução de um único algoritmo num múltiplo conjunto de dados.

In the following sections, we will review each one of these elements in detail.

17.2 A caixa de ferramentas

The *Toolbox* is the main element of the processing GUI, and the one that you are more likely to use in your daily work. It shows the list of all available algorithms grouped in different blocks, and it is the access point to run them, whether as a single process or as a batch process involving several executions of the same algorithm on different sets of inputs.

A caixa de ferramentas contém todos os algoritmos disponíveis, divididos em grupos pré-definidos. Todos esses grupos podem ser encontrados numa árvore única com uma entrada que chama *Geoalgoritmos*

Additionally, two more entries are found, namely *Models* and *Scripts*. These include user-created algorithms, and they allow you to define your own workflows and processing tasks. We will devote a full section to them a bit later.

In the upper part of the toolbox, you will find a text box. To reduce the number of algorithms shown in the toolbox and make it easier to find the one you need, you can enter any word or phrase on the text box. Notice that, as you type, the number of algorithms in the toolbox is reduced to just those that contain the text you have entered in their names.

In the lower part, you will find a box that allows you to switch between the simplified algorithm list (the one explained above) and the advanced list. If you change to the advanced mode, the toolbox will look like this:

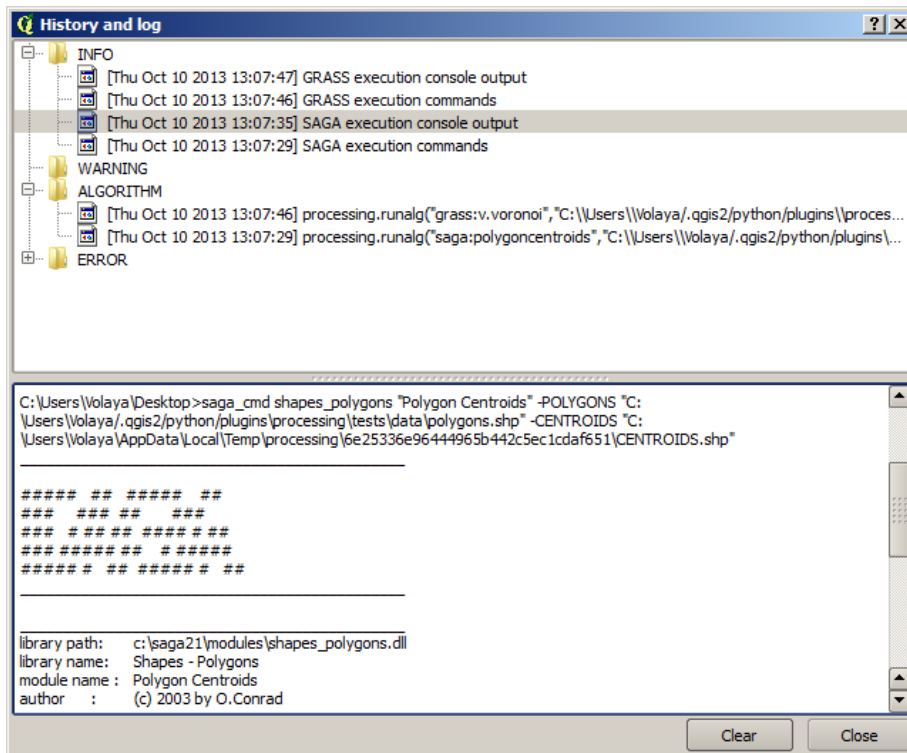


Figura 17.3: Histórico do Processamento

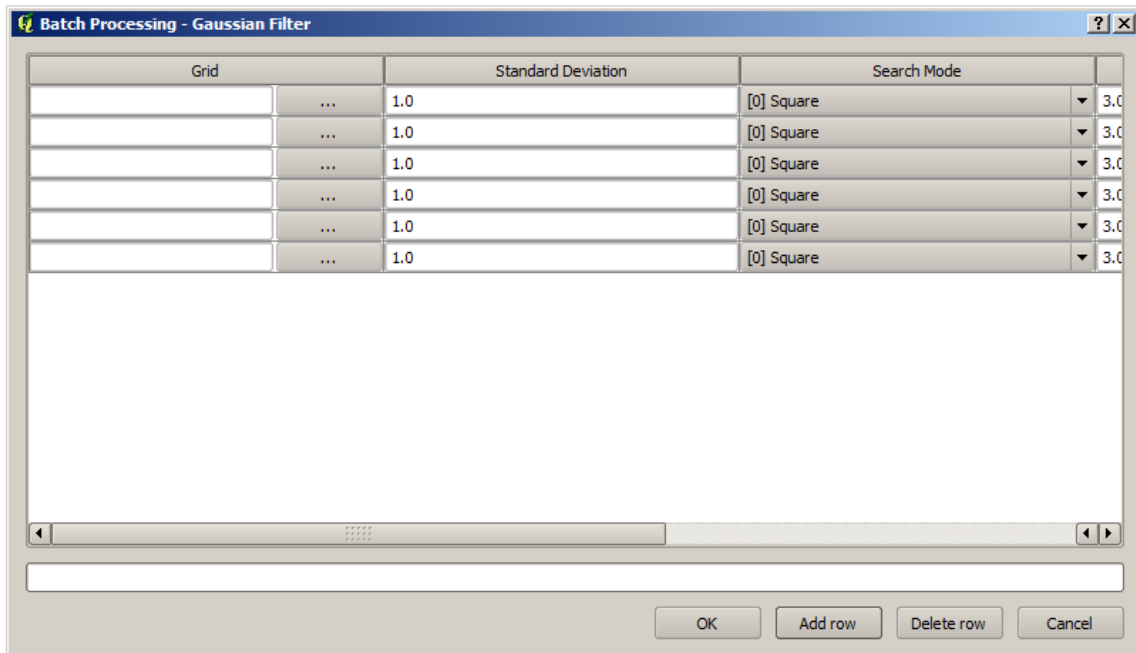


Figura 17.4: Interface de Processamento Batch

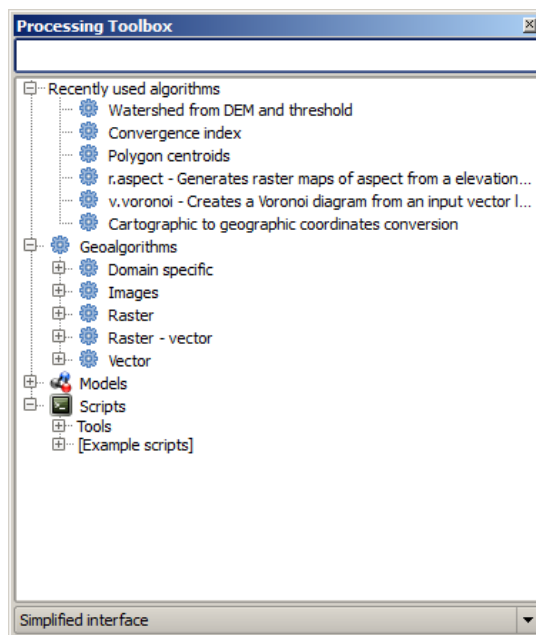


Figura 17.5: Caixa de Ferramentas Processamento

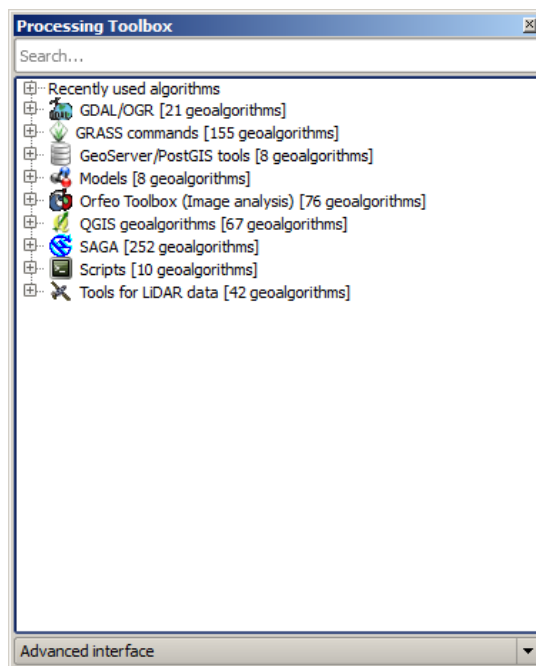


Figura 17.6: Caixa de Ferramentas Processamento (modo avançado)

In the advanced view, each group represents a so-called ‘algorithm provider’, which is a set of algorithms coming from the same source, for instance, from a third-party application with geoprocessing capabilities. Some of these groups represent algorithms from third-party applications like SAGA, GRASS or R, while others contain algorithms directly coded as part of the processing plugin, not relying on any additional software.

This view is recommended to those users who have a certain knowledge of the applications that are backing the algorithms, since they will be shown with their original names and groups.

Also, some additional algorithms are available only in the advanced view, such as LiDAR tools and scripts based on the R statistical computing software, among others. Independent QGIS plugins that add new algorithms to the toolbox will only be shown in the advanced view.

Particularmente, a vista simplificada contém algoritmo dos seguintes fornecedores:

- GRASS
- SAGA
- OTB
- Algoritmos nativos QGIS

In the case of running QGIS under Windows, these algorithms are fully-functional in a fresh installation of QGIS, and they can be run without requiring any additional installation. Also, running them requires no prior knowledge of the external applications they use, making them more accessible for first-time users.

If you want to use an algorithm not provided by any of the above providers, switch to the advanced mode by selecting the corresponding option at the bottom of the toolbox.

Para executar um algoritmo, faça duplo clique no seu nome na caixa de ferramentas.

17.2.1 O diálogo do algoritmo

Once you double-click on the name of the algorithm that you want to execute, a dialog similar to that in the figure below is shown (in this case, the dialog corresponds to the SAGA ‘Convergence index’ algorithm).

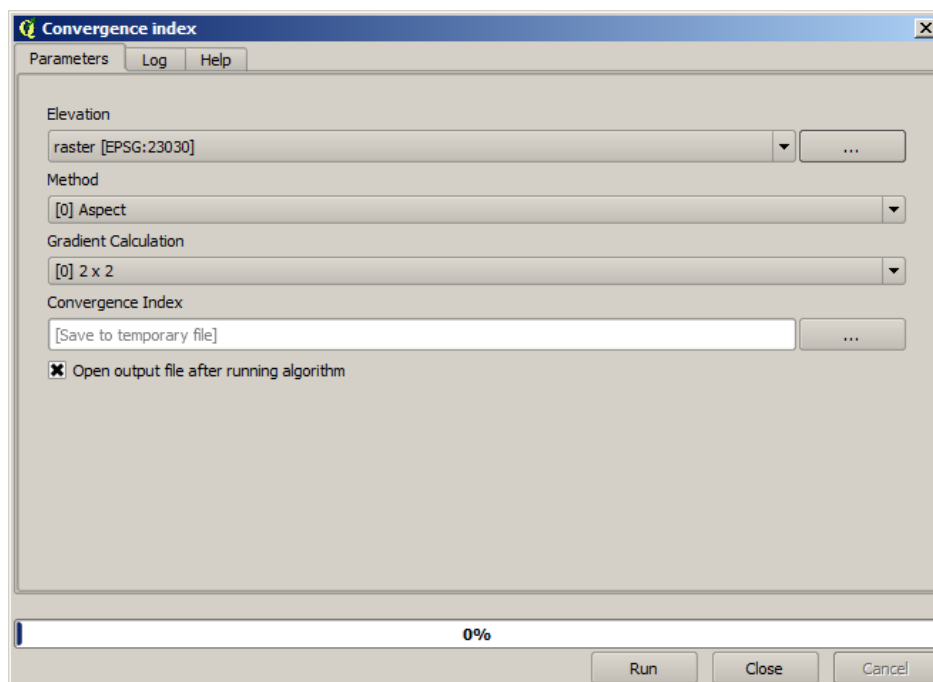


Figura 17.7: Diálogo dos Parâmetros

This dialog is used to set the input values that the algorithm needs to be executed. It shows a table where input values and configuration parameters are to be set. It of course has a different content, depending on the require-

ments of the algorithm to be executed, and is created automatically based on those requirements. On the left side, the name of the parameter is shown. On the right side, the value of the parameter can be set.

Although the number and type of parameters depend on the characteristics of the algorithm, the structure is similar for all of them. The parameters found in the table can be of one of the following types.

- A raster layer, to select from a list of all such layers available (currently opened) in QGIS. The selector contains as well a button on its right-hand side, to let you select filenames that represent layers currently not loaded in QGIS.
- A vector layer, to select from a list of all vector layers available in QGIS. Layers not loaded in QGIS can be selected as well, as in the case of raster layers, but only if the algorithm does not require a table field selected from the attributes table of the layer. In that case, only opened layers can be selected, since they need to be open so as to retrieve the list of field names available.

Irá ver um botão por cada seleccionador de camada vectorial, como é exibido na figura em baixo.

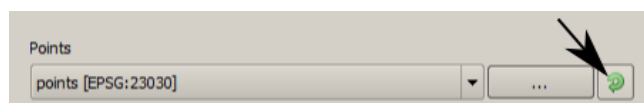


Figura 17.8: Botão de interacção vectorial

If the algorithm contains several of them, you will be able to toggle just one of them. If the button corresponding to a vector input is toggled, the algorithm will be executed iteratively on each one of its features, instead of just once for the whole layer, producing as many outputs as times the algorithm is executed. This allows for automating the process when all features in a layer have to be processed separately.

- A table, to select from a list of all available in QGIS. Non-spatial tables are loaded into QGIS like vector layers, and in fact they are treated as such by the program. Currently, the list of available tables that you will see when executing an algorithm that needs one of them is restricted to tables coming from files in dBase (.dbf) or Comma-Separated Values (.csv) formats.
- Uma opção, a escolher de uma lista de selecção de uma lista de opções possíveis.
- A numerical value, to be introduced in a text box. You will find a button by its side. Clicking on it, you will see a dialog that allows you to enter a mathematical expression, so you can use it as a handy calculator. Some useful variables related to data loaded into QGIS can be added to your expression, so you can select a value derived from any of these variables, such as the cell size of a layer or the northernmost coordinate of another one.

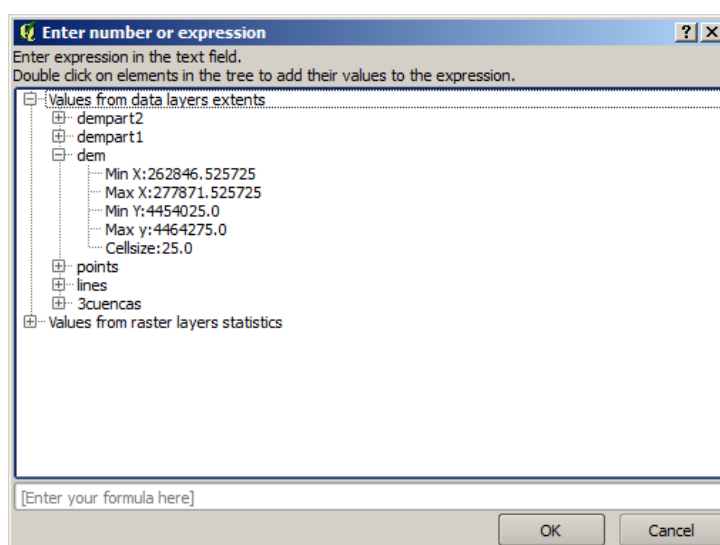


Figura 17.9: Seleccionador de Números

- Um intervalo, com valores min e máx para serem introduzidos em duas caixas de texto.

- Uma cadeia de texto, a ser introduzida na caixa de texto.
- Um campo, para escolher a partir de uma tabela de atributos de uma camada vectorial ou uma tabela única de outro parâmetro.
- A coordinate reference system. You can type the EPSG code directly in the text box, or select it from the CRS selection dialog that appears when you click on the button on the right-hand side.
- An extent, to be entered by four numbers representing its x_{min} , x_{max} , y_{min} , y_{max} limits. Clicking on the button on the right-hand side of the value selector, a pop-up menu will appear, giving you two options: to select the value from a layer or the current canvas extent, or to define it by dragging directly onto the map canvas.

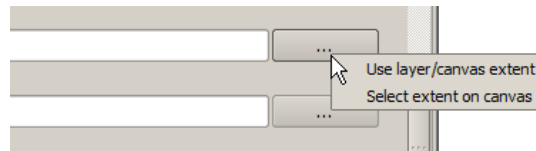


Figura 17.10: Seleccionador de Extensão

Se seleccionar a primeira opção, irá ver uma janela igual a próxima.

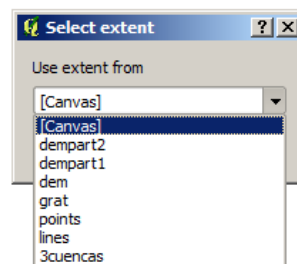


Figura 17.11: Lista de Extensão

Se seleccionar o segundo, os parâmetros da janela irão esconder-se, para que possa clicar e arrastar para o enquadramento. Uma vez definido o rectângulo seleccionado, o diálogo irá reaparecer, contendo os valores na caixa de texto da extensão.

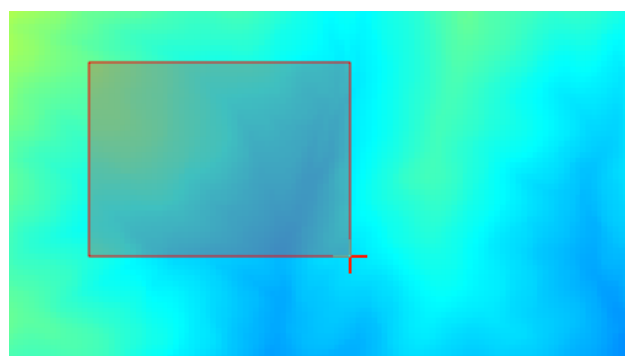


Figura 17.12: Arrastamento para Extensão

- A list of elements (whether raster layers, vector layers or tables), to select from the list of such layers available in QGIS. To make the selection, click on the small button on the left side of the corresponding row to see a dialog like the following one.
- Uma pequena tabela para ser editada pelo utilizador. Estes são usados para definir os parâmetros como tabelas lookup ou kernels de convolução, entre outros.

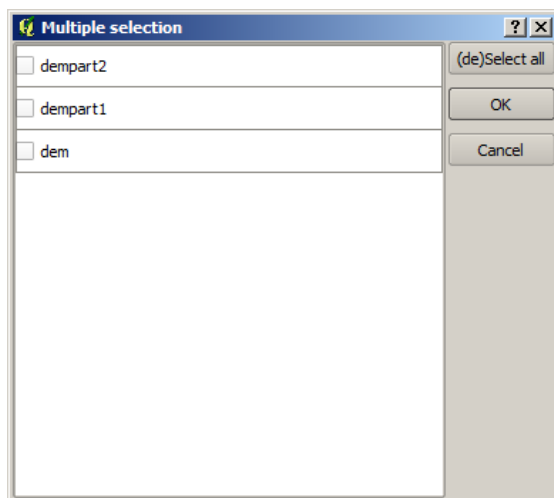


Figura 17.13: Múltipla Seleção

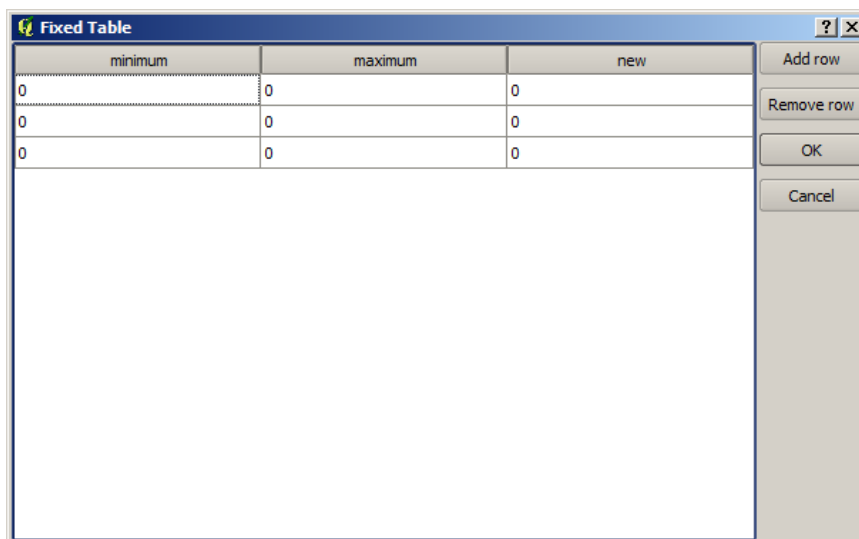


Figura 17.14: Tabela Fixa

Clique no botão do lado direito para ver a tabela e editar os seus valores.

Depending on the algorithm, the number of rows can be modified or not by using the buttons on the right side of the window.

You will find a **[Help]** tab in the the parameters dialog. If a help file is available, it will be shown, giving you more information about the algorithm and detailed descriptions of what each parameter does. Unfortunately, most algorithms lack good documentation, but if you feel like contributing to the project, this would be a good place to start.

Uma nota nas projecções

Algorithms run from the processing framework — this is also true of most of the external applications whose algorithms are exposed through it. Do not perform any reprojection on input layers and assume that all of them are already in a common coordinate system and ready to be analyzed. Whenever you use more than one layer as input to an algorithm, whether vector or raster, it is up to you to make sure that they are all in the same coordinate system.

Note that, due to QGIS's on-the-fly reprojecting capabilities, although two layers might seem to overlap and match, that might not be true if their original coordinates are used without reprojecting them onto a common coordinate system. That reprojection should be done manually, and then the resulting files should be used as input to the algorithm. Also, note that the reprojection process can be performed with the algorithms that are available in the processing framework itself.

By default, the parameters dialog will show a description of the CRS of each layer along with its name, making it easy to select layers that share the same CRS to be used as input layers. If you do not want to see this additional information, you can disable this functionality in the processing configuration dialog, unchecking the *Show CRS* option.

If you try to execute an algorithm using as input two or more layers with unmatching CRSs, a warning dialog will be shown.

Pode continuar a executar o algoritmo, mas tenha atenção que na maioria dos casos irá produzir resultados errados, tais como, camadas vazias devido à falta de sobreposição das camadas usadas como ficheiros de entrada.

17.2.2 Objecto de dados gerados por algoritmos

Objectos de dados gerado por um algoritmo podem ser dos seguintes tipos:

- Uma camada raster
- Uma camada vectorial
- Uma tabela
- Um ficheiro HTML (usado para ficheiro de saída de texto e gráficos)

These are all saved to disk, and the parameters table will contain a text box corresponding to each one of these outputs, where you can type the output channel to use for saving it. An output channel contains the information needed to save the resulting object somewhere. In the most usual case, you will save it to a file, but the architecture allows for any other way of storing it. For instance, a vector layer can be stored in a database or even uploaded to a remote server using a WFS-T service. Although solutions like these are not yet implemented, the processing framework is prepared to handle them, and we expect to add new kinds of output channels in a near future.

To select an output channel, just click on the button on the right side of the text box. That will open a save file dialog, where you can select the desired file path. Supported file extensions are shown in the file format selector of the dialog, depending on the kind of output and the algorithm.

The format of the output is defined by the filename extension. The supported formats depend on what is supported by the algorithm itself. To select a format, just select the corresponding file extension (or add it, if you are directly typing the file path instead). If the extension of the file path you entered does not match any of the supported formats, a default extension (usually `.dbf`` for tables, `.tif` for raster layers and `.shp` for vector layers) will

be appended to the file path, and the file format corresponding to that extension will be used to save the layer or table.

If you do not enter any filename, the result will be saved as a temporary file in the corresponding default file format, and it will be deleted once you exit QGIS (take care with that, in case you save your project and it contains temporary layers).

You can set a default folder for output data objects. Go to the configuration dialog (you can open it from the *Processing* menu), and in the *General* group, you will find a parameter named *Output folder*. This output folder is used as the default path in case you type just a filename with no path (i.e., `myfile.shp`) when executing an algorithm.

When running an algorithm that uses a vector layer in iterative mode, the entered file path is used as the base path for all generated files, which are named using the base name and appending a number representing the index of the iteration. The file extension (and format) is used for all such generated files.

Apart from raster layers and tables, algorithms also generate graphics and text as HTML files. These results are shown at the end of the algorithm execution in a new dialog. This dialog will keep the results produced by any algorithm during the current session, and can be shown at any time by selecting *Processing* → *Results viewer* from the QGIS main menu.

Some external applications might have files (with no particular extension restrictions) as output, but they do not belong to any of the categories above. Those output files will not be processed by QGIS (opened or included into the current QGIS project), since most of the time they correspond to file formats or elements not supported by QGIS. This is, for instance, the case with LAS files used for LiDAR data. The files get created, but you won't see anything new in your QGIS working session.

For all the other types of output, you will find a checkbox that you can use to tell the algorithm whether to load the file once it is generated by the algorithm or not. By default, all files are opened.

Optional outputs are not supported. That is, all outputs are created. However, you can uncheck the corresponding checkbox if you are not interested in a given output, which essentially makes it behave like an optional output (in other words, the layer is created anyway, but if you leave the text box empty, it will be saved to a temporary file and deleted once you exit QGIS).

17.2.3 Configurando a infraestrutura do processamento

Como foi mencionado, o menu de configuração dá acesso a um novo diálogo onde pode configurar a forma como o algoritmo trabalha. Os parâmetros de configuração são estruturados em blocos separados que podem ser seleccionados no lado esquerdo do diálogo.

Juntamente com o que já foi mencionada da entrada *Pasta de saída*, o bloco *Geral* contém parâmetros para configuração de estilos de renderização padrão para camadas de saída (ou seja, camadas geradas pela utilização de algoritmo de qualquer componente da infraestrutura do GUI). Basta criar um estilo que quer usar no QGIS, guarda-lo num ficheiro, e de seguida introduzir um caminho para o ficheiro nas configurações para que sejam usado pelos algoritmos. Cada vez que a camada for carregada pelo SEXTANTE e adicionada ao enquadramento do QGIS, esta será renderizada com esse estilo.

Os estilos de renderização podem ser configurados individualmente para cada algoritmo e cada um para os seus ficheiros de saída. Apenas clique com o direito do rato no nome do algoritmo na caixa de ferramentas e seleccione *Editar estilos de renderização*. Irá ver um diálogo como o que é exibido a seguir.

Seleccione o ficheiro de estilo (`.qml`) que quer para cada ficheiro de saída e pressione **[OK]**.

Os outros parâmetros de configuração no grupo *Geral* estão em baixo referidos:

- *Use filename as layer name*. The name of each resulting layer created by an algorithm is defined by the algorithm itself. In some cases, a fixed name might be used, meaning that the same output name will be used, no matter which input layer is used. In other cases, the name might depend on the name of the input layer or some of the parameters used to run the algorithm. If this checkbox is checked, the name will be taken from the output filename instead. Notice that, if the output is saved to a temporary file, the filename of this temporary file is usually a long and meaningless one intended to avoid collision with other already existing filenames.

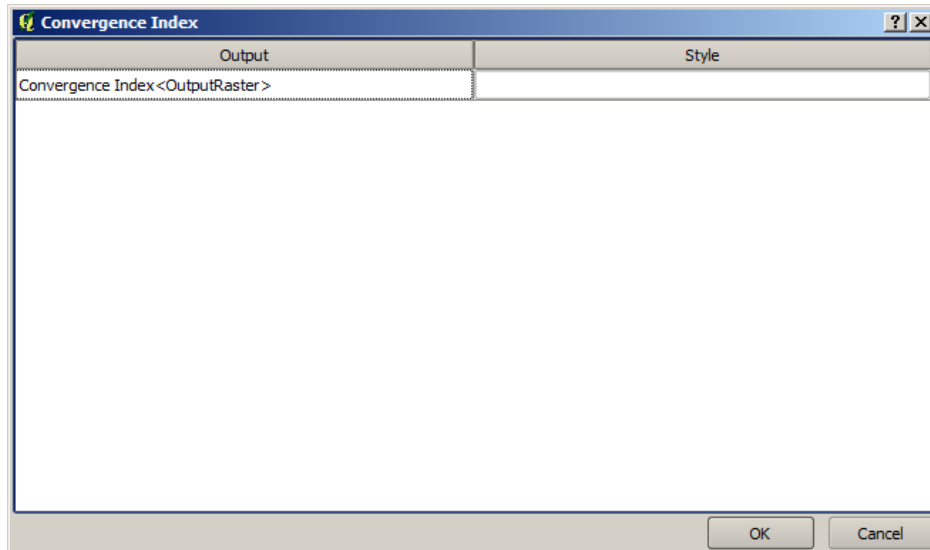



Figura 17.15: Estilos de Renderização 

- *Use only selected features.* If this option is selected, whenever a vector layer is used as input for an algorithm, only its selected features will be used. If the layer has no selected features, all features will be used.
- *Pre-execution script file* and *Post-execution script file.* These parameters refer to scripts written using the processing scripting functionality, and are explained in the section covering scripting and the console.

Apart from the *General* block in the settings dialog, you will also find a block for algorithm providers. Each entry in this block contains an *Activate* item that you can use to make algorithms appear or not in the toolbox. Also, some algorithm providers have their own configuration items, which we will explain later when covering particular algorithm providers.

17.3 O modelador gráfico

O *modelador gráfico* permite que possa criar modelos complexos usando uma simples interface fácil-de-usar. Quando se trabalha com SIG, a maioria das operações de análises não são isoladas, mas fazem parte de uma cadeia de operações. Usando o modelador gráfico, essa cadeia de processos pode ser agregada num único processo, sendo mais fácil e mais conveniente, mais tarde, a sua execução num conjunto de diferentes dados de entrada. Não importa o número de etapas e os diferentes algoritmos envolvidos, o modelo é executado como um único algoritmo, poupando assim tempo e esforço, especialmente para modelos maiores.

O modelador pode ser aberto a partir do menu processamento.

O modelador tem uma área de trabalho onde a estrutura do modelo e o seu fluxo de trabalho são representados como está exibido. Na parte esquerda da janela, um painel com dois separadores podem ser usados para adicionar novos elementos ao modelo.

A criação de um modelo envolve dois passos:

1. *Definição dos dados entrada necessários.* Estes dados de entrada serão adicionados na janela de parâmetros, para que o utilizador possa configurar os seus valores quando executa o modelo. O modelo por si é um algoritmo, portanto os parâmetros da janela é gerado automaticamente como acontece em todos os algoritmos disponíveis na infraestrutura de processamento.
2. *Definição do fluxo de trabalho.* Usando os dados de entrada do modelo, o fluxo de trabalho é definido adicionando algoritmos e seleccionando como vão usar esses ficheiros de entrada ou de saída gerados por outros algoritmos que já existem no modelo

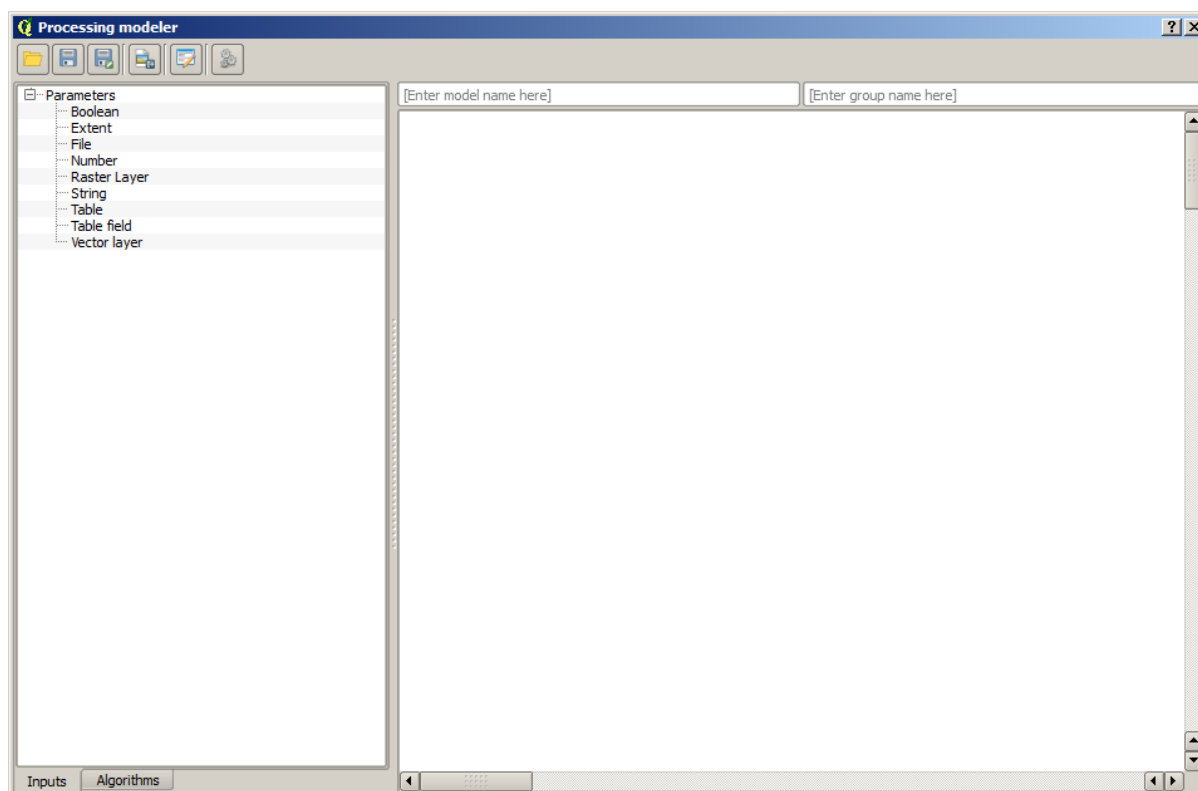



Figura 17.16: Modelador 

17.3.1 Definição das entradas

O primeiro passo para criar um modelo é definir as entradas que necessita. Os seguintes elementos são encontrados no separador *Entradas* no lado esquerdo da janela do modelador:

- Camada matricial
- Camada vectorial
- Cadeia de texto
- Campo da tabela
- Tabela
- Extensão
- Número
- Booleano
- Ficheiro

Faça duplo clique em qualquer um, e será exibida uma janela de diálogo para definir as suas características. Dependendo do parâmetro, o diálogo irá conter apenas um elemento básico (a descrição, que será o que o utilizador irá ver quando executar o modelo) ou outros mais. Por exemplo, quando adiciona um valor numérico, como aparece na próxima figura, além da descrição do parâmetro é necessário definir o valor por defeito e o intervalo de valores válidos.

Para cada entrada adicionada, um novo elemento é adicionado ao enquadramento do modelador.

Pode também adicionar ficheiros de entrada arrastando o tipo de entrada a partir de uma lista e largar no enquadramento do modelador, na posição onde quer que fique.

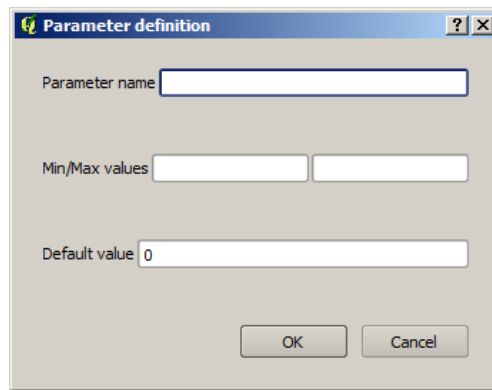


Figura 17.17: Parâmetros do Modelo



Figura 17.18: Parâmetros do Modelo

17.3.2 Definição do fluxo de trabalho

Uma vez as entradas definidas, é tempo de definir os algoritmos que vamos aplicar. Os algoritmos podem ser encontrados no separador *Algoritmos*, agrupados na mesma forma que estão na caixa de ferramentas.

A aparência da caixa de ferramentas tem dois modos: simplificado e avançado. Contudo, não existe nenhum elemento para trocar entre as vistas no modelador, e pode fazê-lo na caixa de ferramentas. O modo que é seleccionado na caixa de ferramentas é aquele que será usado para a lista de algoritmos no modelador.

Para adicionar um algoritmo ao modelo, clique duas vezes no seu nome. Um diálogo de execução irá aparecer, com um conteúdo similar ao que é encontrado no painel de execução que é exibido aquando da execução do algoritmo a partir da caixa de ferramentas. O algoritmo que é exibido corresponde ao algoritmo SAGA ‘Convergence index’, o mesmo que viu na secção dedicada na caixa de ferramentas.

Como pode ver, existem algumas diferenças. Em vez da caixa de saída do ficheiro ser usada para configurar o caminho do ficheiro para as camadas e tabelas de saída, é usado apenas uma caixa de texto simples. Se a camada gerada pelo algoritmo é apenas um resultado temporário, este será usado como o ficheiro de entrada de outro algoritmo e não será mantido como resultado final, mas não edite essa caixa de texto. Introduzindo alguma coisa nela, significa que o resultado é final, e o texto que forneceu será a descrição para o ficheiro de saída, que será aquele que o utilizador irá ver quando executar o modelo.

A selecção do valor para cada parâmetro é também um pouco diferente, uma vez que existem diferenças importantes entre o contexto do modelador e a caixa de ferramentas. Vamos ver como é introduzido os valores para cada tipo de parâmetro.

- Camadas (raster e vector) e tabelas. Estes são seleccionados de uma lista, mas só em caso dos valores possíveis serem camadas ou tabelas carregadas no QGIS, a lista dos dados de entrada do modelo do tipo correspondente, ou outras camadas ou tabelas geradas por algoritmos que já tenham sido adicionadas ao modelo.
- Valores numéricos. Valores literais podem ser introduzidos directamente na caixa de texto. Mas esta caixa de texto é também uma lista que pode ser usada para seleccionar qualquer valor numérico de entrada do modelo. Nesse caso, o parâmetro irá tomar o valor introduzido pelo utilizador quando executado o modelo.
- Cadeia de texto. Como no caso dos valores numéricos, as cadeias de texto podem ser introduzidas, ou uma cadeia de texto de entrada pode ser seleccionada.
- Campo da Tabela. Os campos de uma tabela origem ou camada não podem ser conhecidas na altura do seu desenho, uma vez que dependem da selecção do utilizador cada vez que o modelo é executado. Para definir o valor para este parâmetro, introduza o nome do campo directamente na caixa de texto, ou usa a

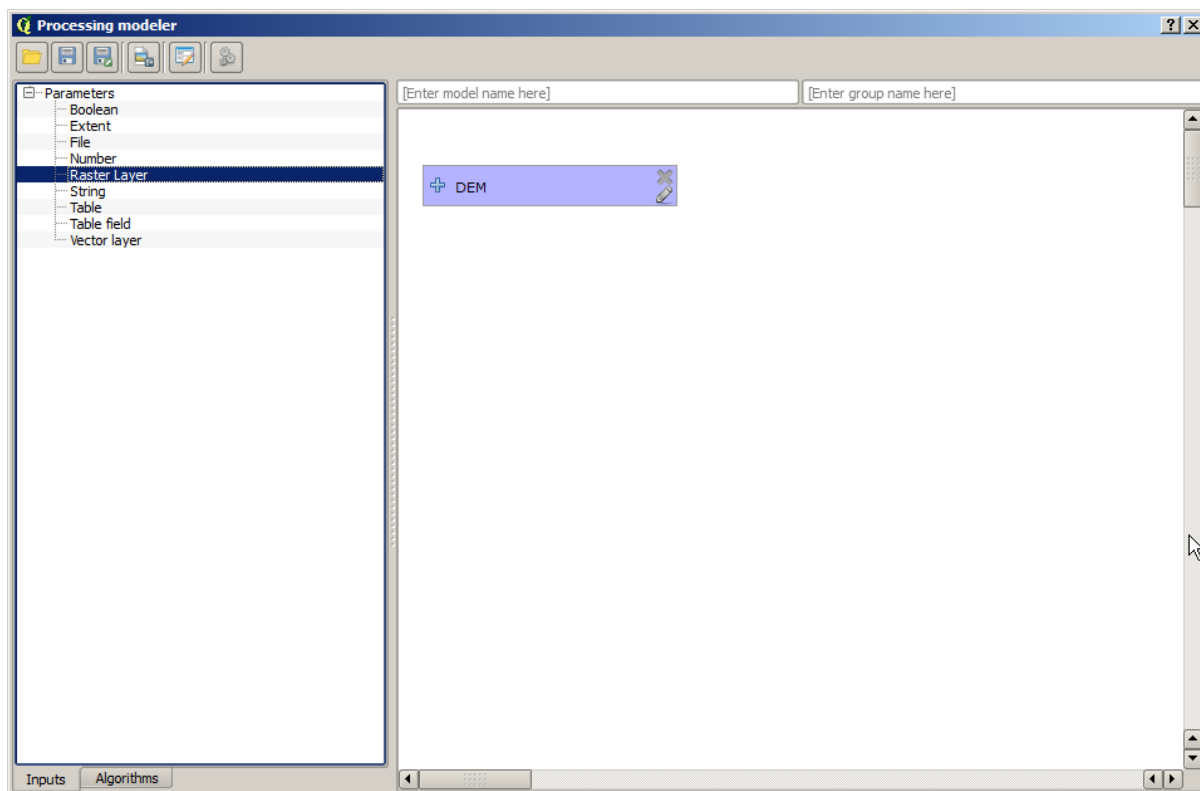


Figura 17.19: Parâmetros do Modelo

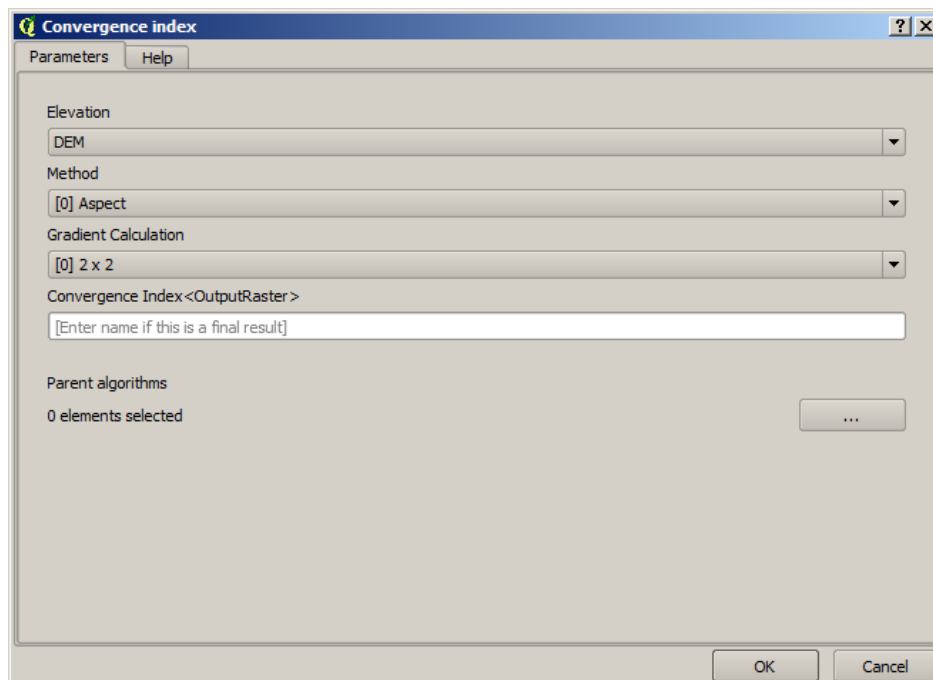


Figura 17.20: Parâmetros do Modelo

lista para seleccionar a entrada do campo de tabela que já está adicionada no modelo. A validade de um campo seleccionado irá ser verificado no seu processo de execução.

Em todos os casos, irá encontrar um parâmetro adicional denominado de *Algoritmos Parent* que não está disponível quando é chamado o algoritmo a partir da caixa de ferramentas. Este parâmetro permite que possa definir a ordem de como os algoritmos são executados, definindo explicitamente um algoritmo como parent do actual, que irá força-lo a executá-lo antes.

Quando usa um ficheiro de saída de um algoritmo prévio como ficheiro de entrada do seu algoritmo, isso implica definir o antigo como parent do actual (e posiciona a seta correspondente no enquadramento do modelador), Contudo, em alguns casos o algoritmo pode depender de outro mesmo que não use um objecto de saída a partir dele (por exemplo, um algoritmo que executa uma instância SQL numa base de dados PostGIS e outra que importa uma camada para a mesma base de dados). Nesse caso, apenas seleccione no parâmetro *Algoritmos Parent* e eles irão ser executados na ordem correcta.

Uma vez todos os parâmetros estiver relacionados a valores válidos, clique no botão **[OK]** e o algoritmo irá ser adicionado ao enquadramento. Será ligado a todos os outros elementos do enquadramento, sejam eles algoritmos ou ficheiros de entrada, o que irá fornecer objectos que foram usados como ficheiros de entrada para esse algoritmo.

Os elementos podem ser arrastados para diferentes posições dentro do enquadramento, para alterar a forma como é exibida a estrutura do módulo e tornar mais claro e intuitivo. As ligações entre os elementos são actualizados automaticamente. Pode ampliar ou afastar usando a roda do rato.

Pode correr o seu algoritmo em qualquer altura clicando no botão **[Executar]**. Contudo, se quiser usá-lo a partir da caixa de ferramentas, é necessário guarda-lo e fechar o diálogo do modelador, para permitir que a caixa de ferramentas actualize os seus conteúdos.

17.3.3 Guardando e e carregando os modelos

Use o botão **[Guardar]** para guardar o modelo actual e o **[Abrir]** para abrir qualquer modelo anteriormente guardado. Os modelos são guardados com a extensão: `.model`. Se o modelo for previamente guardado da janela do modelador, não lhe irá ser solicitado por um nome de ficheiro, uma vez que já existe um ficheiro associado ao modelo, e será usado.

Antes de guardar o modelo, necessita de introduzir o nome e o grupo do modelo, usando as caixas de texto para parte superior da janela.

Os modelos guardados na pasta `modelos` (a pasta padrão quando é pedido o nome do ficheiro para guardar o modelo) irá aparecer na caixa de ferramentas na ligação correspondente. Quando a caixa de ferramentas é chamada, ele procura na pasta `modelos` por ficheiros com a extensão `:file:*.model` e carrega os modelos guardados. Uma vez que o modelo é um próprio algoritmo, este pode ser adicionado na caixa de ferramentas como qualquer outro algoritmo.

A pasta de modelos pode ser configurada a partir do diálogo de configuração do processamento, no grupo *Modelador*

Os modelos carregados a partir da pasta `modelos` aparecem na caixa de ferramentas, mas também na árvore de algoritmos do separador *Algoritmos* da janela do modelador. Isto significa que pode incorporar o modelo como parte de um modelo maior, tal como adiciona qualquer outro algoritmo.

Em alguns casos, um modelo pode não ser carregado porque nem todos os algoritmos incluídos estão disponíveis. Se tiver usado um dado algoritmo como parte do seu modelo, este deverá estar disponível (isto é, deverá aparecer na caixa de ferramentas) de forma a carregar o modelo. Desactivando um fornecedor de algoritmos nas configurações do da janela do processamento poderá desactivar o modelador, o que pode trazer problemas quando carrega os modelos. Tenha isso em atenção quando tiver problemas a carregar ou a executar modelos.

17.3.4 Editando um modelo

Pode editar o modelo que criou, redefinindo o fluxo de trabalho e as relações entre os algoritmos e os dados de entrada que defeni no modelo.

Se clicar com o direito do rato num algoritmo na área que representa o modelo, irá ver o menu contexto como é mostrado a seguir:

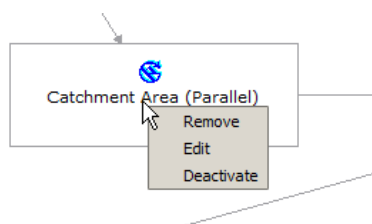


Figura 17.21: Clique direito do Modelador

Seleccionando a opção *Remover* irá fazer com que o algoritmo para ser removido. Um algoritmo só pode ser removido apenas se existir outros algoritmos dependentes dele. Isto é, se não estiver a ser usado um ficheiro de saída de um algoritmo usado como ficheiro de entrada diferente. Se tentar remover um algoritmo que tenha outros que dependa dele, uma mensagem de aviso igual à que vê aqui em baixo:

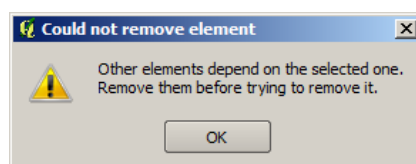


Figura 17.22: Não pode apagar o Algoritmo

Seleccionando a opção *Editar* ou fazendo simplesmente duplo clique no ícone do algoritmo, irá ser exibido o diálogo de parâmetros do algoritmo, para que possa mudar os ficheiros de entrada e os valores do parâmetro. Tenha em atenção que nem todos os elementos disponíveis no modelo aparecerão neste caso como ficheiros de entrada disponível. Camadas ou valores gerados num passo mais avançado no fluxo de trabalho definido pelo modelo não irá estar disponível se causar dependências em ciclo.

Selecione novos valores e de seguida clique no botão **[OK]** como comum. As ligações entre os elementos do modelo irão alterar de acordo com o enquadramento do modelador.

17.3.5 Editando os ficheiros de ajuda do modelo e a meta-informação

Pode documentar os seus modelos a partir do próprio modelador. Basta apenas clicar no botão **[Editar ajuda do modelo]** e o diálogo como aquele que irá aparecer.

No lado direito irá ver uma página simples HTML, criada usando a descrição dos parâmetros de entrada e de saída do algoritmo, juntamente com itens adicionais como uma descrição geral do modelo ou o seu autor. A primeira vez que abre o editor da ajuda, todas as ajudas estão vazias, mas pode editá-los usando os elementos do lado esquerdo do diálogo. Selecione um elemento na parte superior e escreva a descrição na caixa de texto em baixo.

A ajuda do Modelo é guardada num ficheiro na mesma pasta ao pé do seu respectivo modelo. Não tem de ser preocupar como guardá-lo, uma vez ser feito automaticamente.

17.3.6 Sobre os algoritmos disponíveis

Poderá notar que alguns algoritmos podem ser executados a partir da caixa de ferramentas mas não aparecem na lista de algoritmos disponíveis quando está a desenhar o modelo. Para ser incluído no modelo, o algoritmo deve ter uma semântica correcta, assim como ligações apropriadas a outros fluxos de trabalho. Se um algoritmo não tiver uma boa semântica bem definida (por exemplo, se o número de camadas de saída não são conhecidas), portanto não é possível usá-lo dentro do modelo, e não aparece na lista na janela do diálogo do modelador.

Adicionalmente, irá ver algoritmos no modelador que não se encontram na caixa de ferramentas. Esses algoritmos são para ser usados exclusivamente como parte do modelo, e eles não têm interesse noutra contexto diferente.

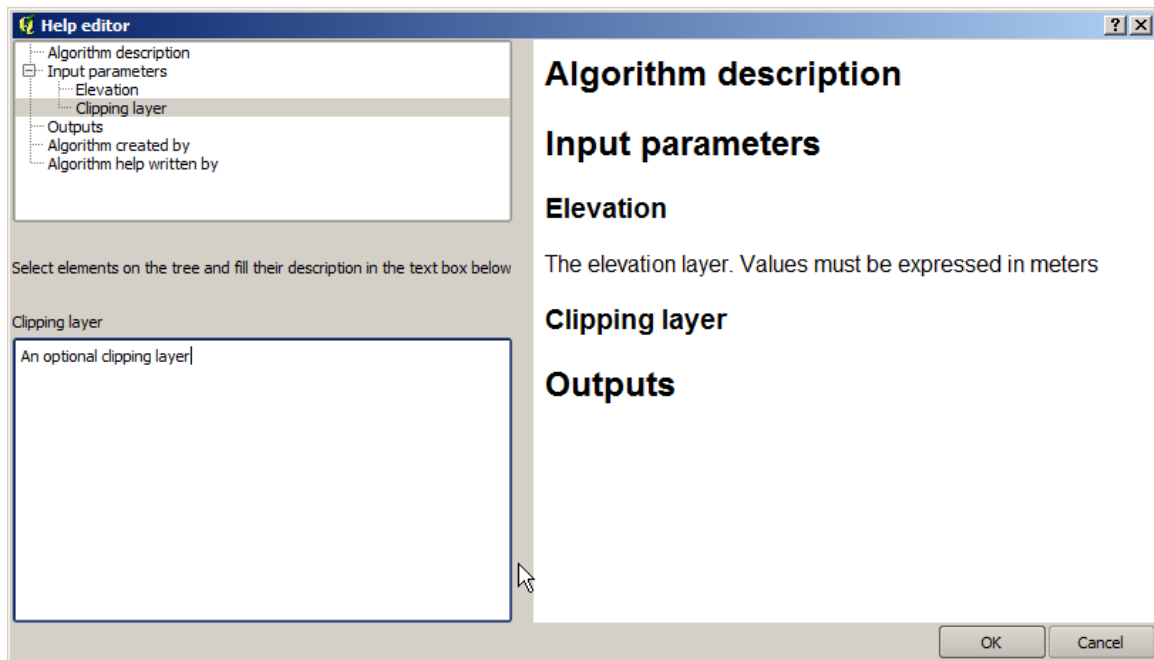


Figura 17.23: Edição da Ajuda

O algoritmo ‘Calculadora’ é um exemplo disso. É apenas uma simples calculadora aritmética que pode usar para modificar valores numéricos (introduzidos pelo utilizador ou gerados por outro tipo de algoritmo). Estas ferramentas são realmente úteis dentro do modelo, mas fora do contexto, não fazem sentido.

17.4 A interface do processamento em lote

17.4.1 Introdução

Todos os algoritmos (incluindo os modelos) podem ser executados como um processo em lote. Quer dizer, que podem ser executados não apenas com um mas com conjunto de ficheiros de entradas, executando o algoritmo quantas vezes forem necessárias. Isto é útil quando processamos grandes quantidades de dados, uma vez que não é necessário lançar o algoritmo a partir da caixa de ferramentas.

Para executar um algoritmo como um processamento em lote, clique com o direito do rato na caixa de ferramentas e seleccione a opção *Executar como processamento em lote* no balão de opções que irá aparecer.

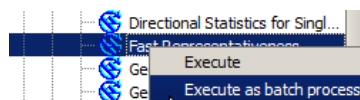


Figura 17.24: Clique do botão direito do rato do processamento em lote

17.4.2 Os parâmetros da tabela

A execução de um processamento em lote é semelhante à execução única de um algoritmo. Os valores do parâmetro têm de ser definidos, mas neste caso nós necessitamos um conjunto de parâmetros, uma para cada vez que o algoritmo é executado. Os valores são introduzidos usando uma tabela como é mostrada a seguir.

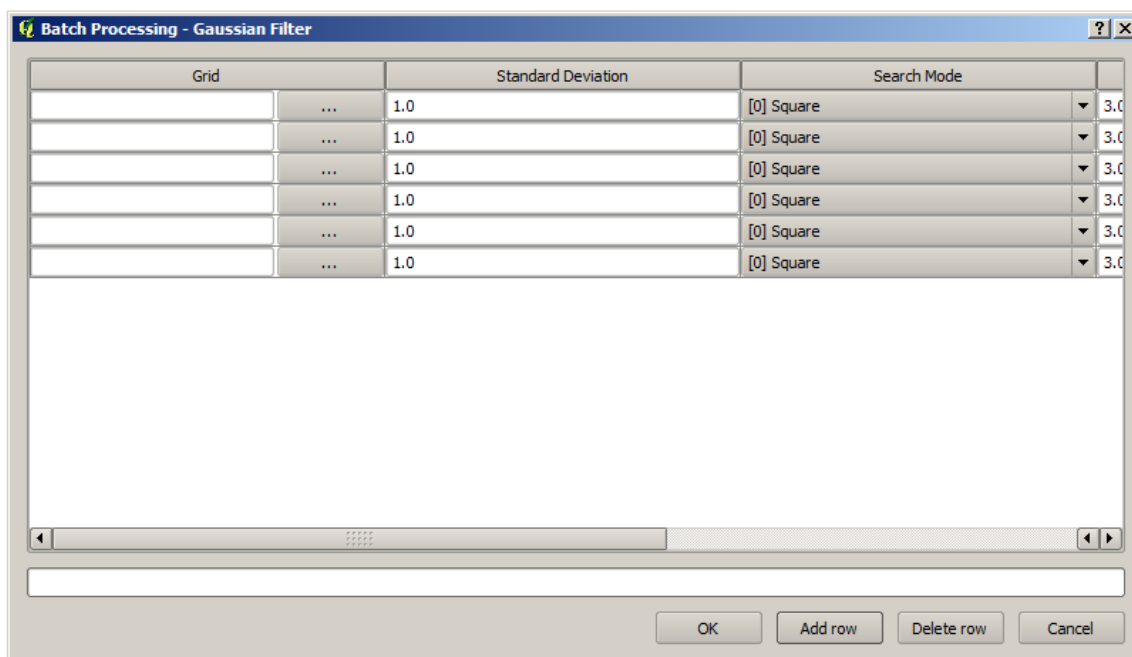


Figura 17.25: Processamento em Lote

Cada linha desta tabela representa uma única execução do algoritmo, e cada célula contém o valor de um dos parâmetros. É semelhante aos parâmetros do diálogo que vê quando executa o algoritmo a partir da caixa de ferramentas, mas com uma diferente disposição.


Por defeito, a tabela contém apenas duas linhas. Pode adicionar ou remover linhas usando os botões na parte inferior da janela.

Uma vez o tamanho da tabela ser configurado, terá de ser preenchido com os valores desejados.

17.4.3 Preenchendo os parâmetros da tabela

Para a maioria dos parâmetros, a configuração de um valor é trivial. Apenas introduza o valor ou selecione-o a partir de uma lista de opções disponíveis dependendo do tipo de parâmetro.

As principais diferenças são encontradas para parâmetros que representam camadas ou tabelas, e para caminhos de ficheiros de saída. Relativamente às camadas e tabelas de entrada, quando o algoritmo é executado como parte de um processamento em lote esses objectos de dados de entrada são usados directamente dos ficheiros, e não de um conjunto deles já abertos no QGIS. Por essa razão, qualquer algoritmo pode ser executado como um processamento em lote mesmo se os objectos de dados estiverem todos abertos e o algoritmo não possa ser executado a partir da caixa de ferramentas.

O nome do ficheiro para os objectos de dados de entrada são introduzidos directamente ou, mais convenientemente através do clique no botão  no lado direito da célula, que exhibe um típico diálogo de escolha de ficheiros. Podem ser seleccionados de uma vez ficheiros múltiplos. Se o parâmetro de entrada representar um objecto único de dados e vários ficheiros são seleccionados, cada um deles serão postos numa linha separada, adicionando novos se necessário. Se representa múltiplas entradas, todos os ficheiros seleccionados irão ser adicionados a uma célula única, separada por ponto e vírgulas (;).

Os dados de saída são sempre guardados num ficheiro e, contrariamente à execução do algoritmo na caixa de ferramentas, não é permitido guardar temporariamente. Pode introduzir o nome directamente ou usar o diálogo de escolha do ficheiro que aparece quando é clicado o botão anexo.

Um vez seleccionado o ficheiro, um novo diálogo será exibido para permitir que faça o auto-preenchimento das outras células na mesma coluna (o mesmo parâmetro).

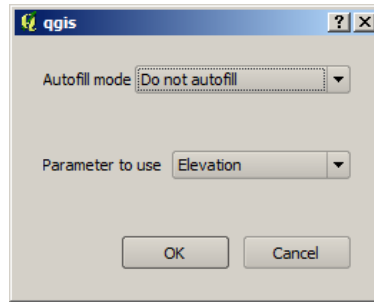


Figura 17.26: Guardar o Processamento em Lote

Se o valor padrão (‘Não efectuar auto-preenchimento’) é seleccionado, irá apenas por o nome de ficheiro seleccionado na célula seleccionada da tabela de parâmetros. Se alguma das outras opções estiver seleccionada, todas as células abaixo da seleccionada irão automaticamente ser preenchidas baseando-se no critério definido. Desta forma, é mais fácil preencher a tabela, e o processamento em lote pode ser definido com menor esforço.

O preenchimento automático pode ser feito de forma simples adicionando números correlativos ao caminho do ficheiro seleccionado, ou anexando o valor de outro campo na mesma linha. Isto é particularmente útil para dar o nome ao objecto de dados de saída de acordo com os dados de entrada.

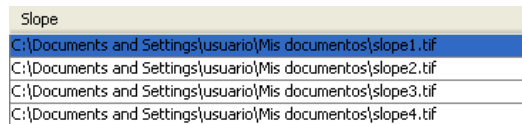



Figura 17.27: Caminho do Ficheiro do Processamento em Lote 

17.4.4 Executando o processo em lote

Após a introdução de todos os valores necessários pode executar o processamento em lote, apenas clicando em **[OK]**. O progresso global da tarefa de processamento será exibido na barra de progresso na parte inferior do diálogo.

17.5 Usando os algoritmos do processamento a partir da consola

The console allows advanced users to increase their productivity and perform complex operations that cannot be performed using any of the other GUI elements of the processing framework. Models involving several algorithms can be defined using the command-line interface, and additional operations such as loops and conditional sentences can be added to create more flexible and powerful workflows.

There is not a processing console in QGIS, but all processing commands are available instead from the QGIS built-in Python console. That means that you can incorporate those commands into your console work and connect processing algorithms to all the other features (including methods from the QGIS API) available from there.

O código que pode executar a partir da consola Python, mesmo quando não chama nenhum método específico do processamento, pode ser convertido num novo algoritmo que pode mais tarde chamar da caixa de ferramentas, o modelador gráfico ou qualquer outro componente, como faz para outro algoritmo. De facto, alguns algoritmos que encontra na caixa de ferramentas são scripts simples.

In this section, we will see how to use processing algorithms from the QGIS Python console, and also how to write algorithms using Python.

17.5.1 Chamando os algoritmos a partir da consola Python

A primeira coisa que tem de fazer é importar as funções do processamento com a seguinte linha:

```
>>> import processing
```

Now, there is basically just one (interesting) thing you can do with that from the console: execute an algorithm. That is done using the `runalg()` method, which takes the name of the algorithm to execute as its first parameter, and then a variable number of additional parameters depending on the requirements of the algorithm. So the first thing you need to know is the name of the algorithm to execute. That is not the name you see in the toolbox, but rather a unique command-line name. To find the right name for your algorithm, you can use the `algslist()` method. Type the following line in your console:

```
>>> processing.algslist()
```

Irá ver algo como isto.

```
Accumulated Cost (Anisotropic)----->saga:accumulatedcost (anisotropic)
Accumulated Cost (Isotropic)----->saga:accumulatedcost (isotropic)
Add Coordinates to points----->saga:addcoordinatestopoints
Add Grid Values to Points----->saga:addgridvaluestopoints
Add Grid Values to Shapes----->saga:addgridvaluestoshapes
Add Polygon Attributes to Points----->saga:addpolygonattributestopoints
Aggregate----->saga:aggregate
Aggregate Point Observations----->saga:aggregatepointobservations
Aggregation Index----->saga:aggregationindex
Analytical Hierarchy Process----->saga:analyticalhierarchyprocess
Analytical Hillshading----->saga:analyticalhillshading
Average With Mask 1----->saga:averagewithmask1
Average With Mask 2----->saga:averagewithmask2
Average With Threshold 1----->saga:averagewiththreshold1
Average With Threshold 2----->saga:averagewiththreshold2
Average With Threshold 3----->saga:averagewiththreshold3
B-Spline Approximation----->saga:b-splineapproximation
...
```

Esta é a lista de todos os algoritmos disponíveis, ordenados alfabeticamente, juntamente com os seus nomes da linha de comandos correspondentes.

You can use a string as a parameter for this method. Instead of returning the full list of algorithms, it will only display those that include that string. If, for instance, you are looking for an algorithm to calculate slope from a DEM, type `algslist("slope")` to get the following result:

```
DTM Filter (slope-based)----->saga:dtmfilter (slope-based)
Downslope Distance Gradient----->saga:downslopedistancegradient
Relative Heights and Slope Positions----->saga:relativeheightsandslopepositions
Slope Length----->saga:sloplength
Slope, Aspect, Curvature----->saga:slopeaspectcurvature
Upslope Area----->saga:upslopearea
Vegetation Index[slope based]----->saga:vegetationindex[slopebased]
```

Este resultado pode mudar dependendo dos algoritmos que estão disponíveis.

Agora é mais fácil encontrar o algoritmo que procura e o seu nome da linha de comandos, neste caso `saga:slopeaspectcurvature`.

Once you know the command-line name of the algorithm, the next thing to do is to determine the right syntax to execute it. That means knowing which parameters are needed and the order in which they have to be passed when calling the `runalg()` method. There is a method to describe an algorithm in detail, which can be used to get a list of the parameters that an algorithm requires and the outputs that it will generate. To get this information, you can use the `alghelp(name_of_the_algorithm)` method. Use the command-line name of the algorithm, not the full descriptive name.

Calling the method with `saga:slopeaspectcurvature` as parameter, you get the following description:

```
>>> processing.alghelp("saga:slopeaspectcurvature")
ALGORITHM: Slope, Aspect, Curvature
  ELEVATION <ParameterRaster>
  METHOD <ParameterSelection>
  SLOPE <OutputRaster>
  ASPECT <OutputRaster>
  CURV <OutputRaster>
  HCURV <OutputRaster>
  VCURV <OutputRaster>
```

Agora tem tudo o que necessita de correr qualquer algoritmo. Como já tínhamos mencionado, existe apenas um comando para executar algoritmos: `runalg()`. A sua sintaxe é como está descrito a seguir:

```
>>> processing.runalg(name_of_the_algorithm, param1, param2, ..., paramN,
  Output1, Output2, ..., OutputN)
```

A lista de parâmetros e ficheiros de saída para adicionar dependem do algoritmo que quer correr, e é exactamente a lista que o método `alghelp()` lhe dá, na mesma ordem que é exibido.

Depending on the type of parameter, values are introduced differently. The next list gives a quick review of how to introduce values for each type of input parameter:

- Raster Layer, Vector Layer or Table. Simply use a string with the name that identifies the data object to use (the name it has in the QGIS Table of Contents) or a filename (if the corresponding layer is not opened, it will be opened but not added to the map canvas). If you have an instance of a QGIS object representing the layer, you can also pass it as parameter. If the input is optional and you do not want to use any data object, use `None`.
- Selecção. Se algum algoritmo tiver um parâmetro de selecção, o valor desse parâmetro deve ser introduzido usando um valor inteiro. Para saber as opções disponíveis, pode usar o comando `algotptions()`, como é exibido no seguinte exemplo:

```
>>> processing.algotptions("saga:slopeaspectcurvature")
METHOD (Method)
  0 - [0] Maximum Slope (Travis et al. 1975)
  1 - [1] Maximum Triangle Slope (Tarboton 1997)
  2 - [2] Least Squares Fitted Plane (Horn 1981, Costa-Cabral & Burgess 1996)
  3 - [3] Fit 2.Degree Polynom (Bauer, Rohdenburg, Bork 1985)
  4 - [4] Fit 2.Degree Polynom (Heerdegen & Beran 1982)
  5 - [5] Fit 2.Degree Polynom (Zevenbergen & Thorne 1987)
  6 - [6] Fit 3.Degree Polynom (Haralick 1983)
```

In this case, the algorithm has one such parameter, with seven options. Notice that ordering is zero-based.

- Multiple input. The value is a string with input descriptors separated by semicolons (;). As in the case of single layers or tables, each input descriptor can be the data object name, or its file path.
- Campo da Tabela de XXX. Use uma cadeia de texto com o nome do campo a usar. O parâmetro é caso sensível.
- Fixed Table. Type the list of all table values separated by commas (,) and enclosed between quotes ("). Values start on the upper row and go from left to right. You can also use a 2-D array of values representing the table.
- SRC. Introduza o número do código ESPG do SRC desejado.
- Extensão. Deve usar uma cadeia de texto com `xmin`, `xmax`, `ymin` e `ymax` valores separados por vírgulas (,).

Os parâmetros booleanos, de ficheiro, cadeia de texto e numéricos não necessitam de explicações adicionais.

Input parameters such as strings, booleans, or numerical values have default values. To use them, specify `None` in the corresponding parameter entry.

For output data objects, type the file path to be used to save it, just as it is done from the toolbox. If you want to save the result to a temporary file, use `None`. The extension of the file determines the file format. If you enter

a file extension not supported by the algorithm, the default file format for that output type will be used, and its corresponding extension appended to the given file path.

Unlike when an algorithm is executed from the toolbox, outputs are not added to the map canvas if you execute that same algorithm from the Python console. If you want to add an output to the map canvas, you have to do it yourself after running the algorithm. To do so, you can use QGIS API commands, or, even easier, use one of the handy methods provided for such tasks.

The `runalg` method returns a dictionary with the output names (the ones shown in the algorithm description) as keys and the file paths of those outputs as values. You can load those layers by passing the corresponding file paths to the `load()` method.

17.5.2 Funções adicionais para a manipulação dos dados

Além das funções usadas para chamar os algoritmos, importar o pacote `processamento` irá também importar algumas funções adicionais que facilitará o trabalho dos dados, particularmente os dados vectoriais. Estas funções de conveniência que envolvem alguma funcionalidade a partir da API do QGIS, usualmente com uma sintaxe menos complexa. Estas funções devem ser usadas quando são programados novos algoritmos, para tornar mais fácil a operação com os dados de entrada.

Below is a list of some of these commands. More information can be found in the classes under the `processing/tools` package, and also in the example scripts provided with QGIS.

- `getObject(obj)`: Returns a QGIS object (a layer or table) from the passed object, which can be a filename or the name of the object in the QGIS Table of Contents.
- `values(layer, fields)`: Returns the values in the attributes table of a vector layer, for the passed fields. Fields can be passed as field names or as zero-based field indices. Returns a dict of lists, with the passed field identifiers as keys. It considers the existing selection.
- `features(layer)`: Returns an iterator over the features of a vector layer, considering the existing selection.
- `uniqueValues(layer, field)`: Returns a list of unique values for a given attribute. Attributes can be passed as a field name or a zero-based field index. It considers the existing selection.

17.5.3 Criando scripts e correndo-os a partir da caixa de ferramentas

You can create your own algorithms by writing the corresponding Python code and adding a few extra lines to supply additional information needed to define the semantics of the algorithm. You can find a *Create new script* menu under the *Tools* group in the *Script* algorithms block of the toolbox. Double-click on it to open the script editing dialog. That's where you should type your code. Saving the script from there in the `scripts` folder (the default folder when you open the save file dialog) with `.py` extension will automatically create the corresponding algorithm.

O nome do algoritmo (aquele que irá ver na caixa de ferramentas) é criado a partir do nome do ficheiro, removendo a extensão e substituindo os hífens inferiores com espaços em branco.

Let's have a look at the following code, which calculates the Topographic Wetness Index (TWI) directly from a DEM.

```
##dem=raster
##twi=output
ret_slope = processing.runalg("saga:slopeaspectcurvature", dem, 0, None,
                             None, None, None, None)
ret_area = processing.runalg("saga:catchmentarea(mass-fluxmethod)", dem,
                             0, False, False, False, False, None, None, None, None, None)
processing.runalg("saga:topographicwetnessindex(twi)", ret_slope['SLOPE'],
                 ret_area['AREA'], None, 1, 0, twi)
```


As you can see, the calculation involves three algorithms, all of them coming from SAGA. The last one calculates the TWI, but it needs a slope layer and a flow accumulation layer. We do not have these layers, but since we have the DEM, we can calculate them by calling the corresponding SAGA algorithms.

A parte do código onde este processamento tem lugar não é difícil de perceber se leu-o as secções anteriores deste capítulo. Contudo, as primeiras linhas, necessitam de uma explicação adicional. Eles fornecem a informação que é necessária para tornar o código num algoritmo que possa ser corrido a partir qualquer componente do GUI, como por exemplo a caixa de ferramentas ou o modelador gráfico.

These lines start with a double Python comment symbol (##) and have the following structure:

```
[parameter_name]=[parameter_type] [optional_values]
```

Here is a list of all the parameter types that are supported in processing scripts, their syntax and some examples.

- `raster`. A raster layer.
- `vector`. A vector layer.
- `table`. A table.
- `number`. A numerical value. A default value must be provided. For instance, `depth=number 2.4`.
- `string`. A text string. As in the case of numerical values, a default value must be added. For instance, `name=string Victor`.
- `boolean`. A boolean value. Add `True` or `False` after it to set the default value. For example, `verbose=boolean True`.
- `múltiplos raster`. Um conjunto de camadas rasters de entrada.
- `vectores múltiplos`. Um conjunto de camadas vectoriais de entrada.
- `campo`. Um campo da tabela de atributos de uma camada vectorial. O nome da camada tem de ser adicionada depois da etiqueta `campo`. Por exemplo, se declarou um ficheiro de entrada vectorial com `mylayer=vector`, poderá usar `myfield=field mylayer` para adicionar o campo a partir dessa camada como parâmetro.
- `folder`. A folder.
- `file`. A filename.

O nome do parâmetro é o nome que será exibido ao utilizador quando executa o algoritmo, e também o nome da variável a usar no código do script. O valor introduzido pelo utilizador para esse parâmetro será atribuído à variável com esse nome.

When showing the name of the parameter to the user, the name will be edited to improve its appearance, replacing low hyphens with spaces. So, for instance, if you want the user to see a parameter named `A_numerical_value`, you can use the variable name `A_numerical_value`.

Layers and table values are strings containing the file path of the corresponding object. To turn them into a QGIS object, you can use the `processing.getObjectFromUri()` function. Multiple inputs also have a string value, which contains the file paths to all selected object, separated by semicolons (;).

Os ficheiros de saída são definidos numa maneira semelhante, usando as seguintes etiquetas:

- `raster de saída`
- `vector de saída`
- `tabela de saída`
- `html de saída`
- `ficheiro de saída`
- `número de saída`
- `cadeia de texto de saída`

The value assigned to the output variables is always a string with a file path. It will correspond to a temporary file path in case the user has not entered any output filename.

When you declare an output, the algorithm will try to add it to QGIS once it is finished. That is why, although the `runalg()` method does not load the layers it produces, the final TWI layer will be loaded (using the case of our previous example), since it is saved to the file entered by the user, which is the value of the corresponding output.

Do not use the `load()` method in your script algorithms, just when working with the console line. If a layer is created as output of an algorithm, it should be declared as such. Otherwise, you will not be able to properly use the algorithm in the modeler, since its syntax (as defined by the tags explained above) will not match what the algorithm really creates.

Hidden outputs (numbers and strings) do not have a value. Instead, you have to assign a value to them. To do so, just set the value of a variable with the name you used to declare that output. For instance, if you have used this declaration,

```
##average=output number
```

a linha seguinte irá configurar o valor de saída para 5:

```
average = 5
```

Em adição às etiquetas para os parâmetros e ficheiros de saída, pode também definir o grupo onde o algoritmo será exibido, usando a etiqueta `group`.

If your algorithm takes a long time to process, it is a good idea to inform the user. You have a global named `progress` available, with two possible methods: `setText(text)` and `setPercentage(percent)` to modify the progress text and the progress bar.

Several examples are provided. Please check them to see real examples of how to create algorithms using the processing framework classes. You can right-click on any script algorithm and select *Edit script* to edit its code or just to see it.

17.5.4 Documentando os seus scripts

As in the case of models, you can create additional documentation for your scripts, to explain what they do and how to use them. In the script editing dialog, you will find an **[Edit script help]** button. Click on it and it will take you to the help editing dialog. Check the section about the graphical modeler to know more about this dialog and how to use it.

Help files are saved in the same folder as the script itself, adding the `.help` extension to the filename. Notice that you can edit your script's help before saving the script for the first time. If you later close the script editing dialog without saving the script (i.e., you discard it), the help content you wrote will be lost. If your script was already saved and is associated to a filename, saving the help content is done automatically.

17.5.5 Pré- e pós-execução de encaixe da script

Os scripts podem também ser usados para definir um encaixe de pré- e pós-execução que correm antes e depois do algoritmo correr. Isto pode ser usado para automatizar tarefas que serão executados quando qualquer algoritmo é executado.

A sintaxe é idêntica à sintaxe explicada em cima, mas uma variável global `alg` está disponível, representando o algoritmo que foi (ou está prestes a ser) executado.

In the *General* group of the processing configuration dialog, you will find two entries named *Pre-execution script file* and *Post-execution script file* where the filename of the scripts to be run in each case can be entered.

17.6 Gestão do histórico

17.6.1 O histórico do processamento

Cada vez que executa um algoritmo, a informação do processo é armazenado no gestor histórico. Juntamente com os parâmetros usados, a data e o tempo de execução são também guardados.

This way, it is easy to track and control all the work that has been developed using the processing framework, and easily reproduce it.

The history manager is a set of registry entries grouped according to their date of execution, making it easier to find information about an algorithm executed at any particular moment.

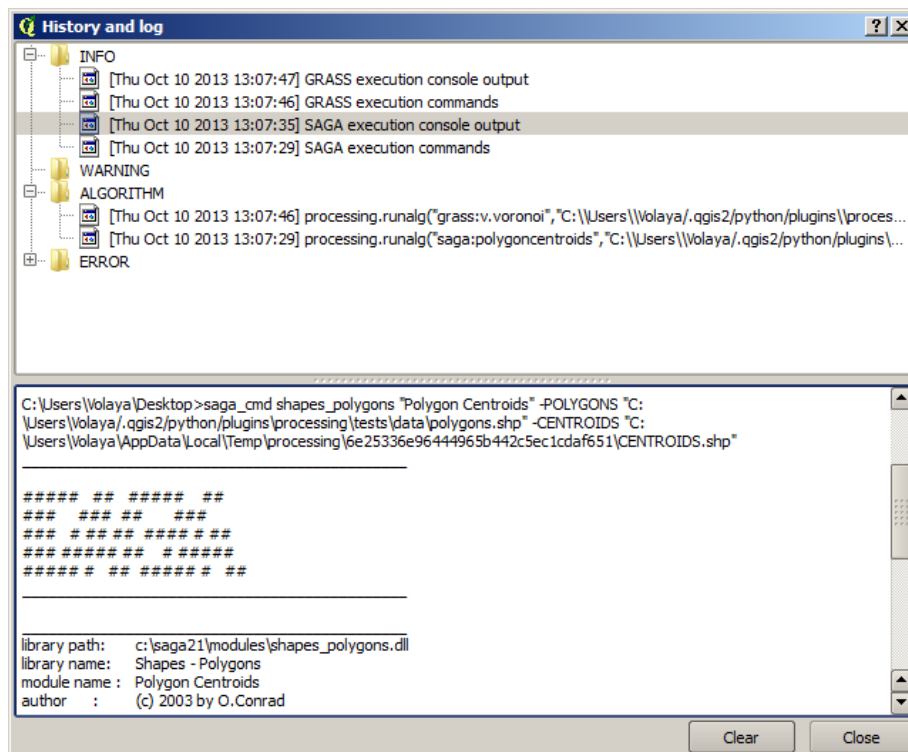



Figura 17.28: Histórico 

A informação do processo é mantida como uma expressão de linha de comandos, mesmo se o algoritmo seja lançado a partir da barra de ferramentas. Isto permite ser útil para aqueles que querem aprender como se usa a interface da linha de comandos, uma vez que podem chamar o algoritmo usando a barra de ferramentas e verificar o gestor histórico para ver como o mesmo algoritmo pode ser chamado a partir da linha de comandos.

Apart from browsing the entries in the registry, you can also re-execute processes by simply double-clicking on the corresponding entry.

Along with recording algorithm executions, the processing framework communicates with the user by means of the other groups of the registry, namely *Errors*, *Warnings* and *Information*. In case something is not working properly, having a look at the *Errors* might help you to see what is happening. If you get in contact with a developer to report a bug or error, the information in that group will be very useful for her or him to find out what is going wrong.

Third-party algorithms are usually executed by calling their command-line interfaces, which communicate with the user via the console. Although that console is not shown, a full dump of it is stored in the *Information* group each time you run one of those algorithms. If, for instance, you are having problems executing a SAGA algorithm, look for an entry named 'SAGA execution console output' to check all the messages generated by SAGA and try to find out where the problem is.

Some algorithms, even if they can produce a result with the given input data, might add comments or additional information to the *Warning* block if they detect potential problems with the data, in order to warn you. Make sure you check those messages if you are having unexpected results.

17.7 Writing new Processing algorithms as python scripts

You can create your own algorithms by writing the corresponding Python code and adding a few extra lines to supply additional information needed to define the semantics of the algorithm. You can find a *Create new script* menu under the *Tools* group in the *Script* algorithms block of the toolbox. Double-click on it to open the script edition dialog. That's where you should type your code. Saving the script from there in the `scripts` folder (the default one when you open the save file dialog), with `.py` extension, will automatically create the corresponding algorithm.

The name of the algorithm (the one you will see in the toolbox) is created from the filename, removing its extension and replacing low hyphens with blank spaces.

Let's have the following code, which calculates the Topographic Wetness Index (TWI) directly from a DEM

```
##dem=raster
##twi=output raster
ret_slope = processing.runalg("saga:slopeaspectcurvature", dem, 0, None,
                             None, None, None, None)
ret_area = processing.runalg("saga:catchmentarea", dem,
                             0, False, False, False, False, None, None, None, None, None)
processing.runalg("saga:topographicwetnessindextwi", ret_slope['SLOPE'],
                 ret_area['AREA'], None, 1, 0, twi)
```

As you can see, it involves 3 algorithms, all of them coming from SAGA. The last one of them calculates the TWI, but it needs a slope layer and a flow accumulation layer. We do not have these ones, but since we have the DEM, we can calculate them calling the corresponding SAGA algorithms.

The part of the code where this processing takes place is not difficult to understand if you have read the previous chapter. The first lines, however, need some additional explanation. They provide the information that is needed to turn your code into an algorithm that can be run from any of the GUI components, like the toolbox or the graphical modeler.

These lines start with a double Python comment symbol (`##`) and have the following structure

```
[parameter_name]=[parameter_type] [optional_values]
```

Here is a list of all the parameter types that are supported in processing scripts, their syntax and some examples.

- `raster`. A raster layer
- `vector`. A vector layer
- `table`. A table
- `number`. A numerical value. A default value must be provided. For instance, `depth=number 2.4`
- `string`. A text string. As in the case of numerical values, a default value must be added. For instance, `name=string Victor`
- `longstring`. Same as `string`, but a larger text box will be shown, so it is better suited for long strings, such as for a script expecting a small code snippet.
- `boolean`. A boolean value. Add `True` or `False` after it to set the default value. For example, `verbose=boolean True`.
- `multiple raster`. A set of input raster layers.
- `multiple vector`. A set of input vector layers.

- `field`. A field in the attributes table of a vector layer. The name of the layer has to be added after the `field` tag. For instance, if you have declared a vector input with `mylayer=vector`, you could use `myfield=field mylayer` to add a field from that layer as parameter.
- `folder`. A folder
- `file`. A filename
- `crs`. A Coordinate Reference System

The parameter name is the name that will be shown to the user when executing the algorithm, and also the variable name to use in the script code. The value entered by the user for that parameter will be assigned to a variable with that name.

When showing the name of the parameter to the user, the name will be edited to improve its appearance, replacing low hyphens with spaces. So, for instance, if you want the user to see a parameter named `A_numerical_value`, you can use the variable name `A_numerical_value`.

Layers and tables values are strings containing the filepath of the corresponding object. To turn them into a QGIS object, you can use the `processing.getObjectFromUri()` function. Multiple inputs also have a string value, which contains the filepaths to all selected objects, separated by semicolons (;).

Outputs are defined in a similar manner, using the following tags:

- `output raster`
- `output vector`
- `output table`
- `output html`
- `output file`
- `output number`
- `output string`
- `output extent`

The value assigned to the output variables is always a string with a filepath. It will correspond to a temporary filepath in case the user has not entered any output filename.

In addition to the tags for parameters and outputs, you can also define the group under which the algorithm will be shown, using the `group` tag.

The last tag that you can use in your script header is `##nomodeler`. Use that when you do not want your algorithm to be shown in the modeler window. This should be used for algorithms that do not have a clear syntax (for instance, if the number of layers to be created is not known in advance, at design time), which make them unsuitable for the graphical modeler

17.8 Handing data produced by the algorithm

When you declare an output representing a layer (raster, vector or table), the algorithm will try to add it to QGIS once it is finished. That is the reason why, although the `runalg()` method does not load the layers it produces, the final `TWI` layer will be loaded, since it is saved to the file entered by the user, which is the value of the corresponding output.

Do not use the `load()` method in your script algorithms, but just when working with the console line. If a layer is created as output of an algorithm, it should be declared as such. Otherwise, you will not be able to properly use the algorithm in the modeler, since its syntax (as defined by the tags explained above) will not match what the algorithm really creates.

Hidden outputs (numbers and strings) do not have a value. Instead, it is you who has to assign a value to them. To do so, just set the value of a variable with the name you used to declare that output. For instance, if you have used this declaration,

```
##average=output number
```

the following line will set the value of the output to 5:

```
average = 5
```

17.9 Communicating with the user

If your algorithm takes a long time to process, it is a good idea to inform the user. You have a global named `progress` available, with two available methods: `setText(text)` and `setPercentage(percent)` to modify the progress text and the progress bar.

If you have to provide some information to the user, not related to the progress of the algorithm, you can use the `setInfo(text)` method, also from the `progress` object.

If your script has some problem, the correct way of propagating it is to raise an exception of type `GeoAlgorithmExecutionException()`. You can pass a message as argument to the constructor of the exception. Processing will take care of handling it and communicating with the user, depending on where the algorithm is being executed from (toolbox, modeler, Python console...)

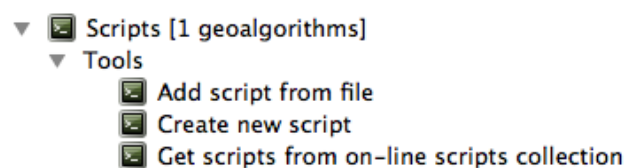
17.10 Documenting your scripts

As in the case of models, you can create additional documentation for your script, to explain what they do and how to use them. In the script editing dialog you will find a **[Edit script help]** button. Click on it and it will take you to the help editing dialog. Check the chapter about the graphical modeler to know more about this dialog and how to use it.

Help files are saved in the same folder as the script itself, adding the `.help` extension to the filename. Notice that you can edit your script's help before saving it for the first time. If you later close the script editing dialog without saving the script (i.e. you discard it), the help content you wrote will be lost. If your script was already saved and is associated to a filename, saving is done automatically.

17.11 Example scripts

Several examples are available in the on-line collection of scripts, which you can access by selecting the *Get script from on-line script collection* tool under the *Scripts/tools* entry in the toolbox.



Please, check them to see real examples of how to create algorithms using the processing framework classes. You can right-click on any script algorithm and select *Edit script* to edit its code or just to see it.

17.12 Best practices for writing script algorithms

Here's a quick summary of ideas to consider when creating your script algorithms and, especially, if you want to share with other QGIS users. Following these simple rules will ensure consistency across the different Processing elements such as the toolbox, the modeler or the batch processing interface.

- Do not load resulting layers. Let Processing handle your results and load your layers if needed.

- Always declare the outputs your algorithm creates. Avoid things such as declaring one output and then using the destination filename set for that output to create a collection of them. That will break the correct semantics of the algorithm and make it impossible to use it safely in the modeler. If you have to write an algorithm like that, make sure you add the `##nomodeler` tag.
- Do not show message boxes or use any GUI element from the script. If you want to communicate with the user, use the `setInfo()` method or throw an `GeoAlgorithmExecutionException`
- As a rule of thumb, do not forget that your algorithm might be executed in a context other than the Processing toolbox.

17.13 Pre- and post-execution script hooks

Scripts can also be used to set pre- and post-execution hooks that are run before and after an algorithm is run. This can be used to automate tasks that should be performed whenever an algorithm is executed.

The syntax is identical to the syntax explained above, but an additional global variable named `alg` is available, representing the algorithm that has just been (or is about to be) executed.

In the *General* group of the processing config dialog you will find two entries named *Pre-execution script file* and *Post-execution script file* where the filename of the scripts to be run in each case can be entered.

17.14 Configurando as aplicações externas

The processing framework can be extended using additional applications. Currently, SAGA, GRASS, OTB (Orfeo Toolbox) and R are supported, along with some other command-line applications that provide spatial data analysis functionalities. Algorithms relying on an external application are managed by their own algorithm provider.

This section will show you how to configure the processing framework to include these additional applications, and it will explain some particular features of the algorithms based on them. Once you have correctly configured the system, you will be able to execute external algorithms from any component like the toolbox or the graphical modeler, just like you do with any other geospatial algorithm.

By default, all algorithms that rely on an external application not shipped with QGIS are not enabled. You can enable them in the configuration dialog. Make sure that the corresponding application is already installed in your system. Enabling an algorithm provider without installing the application it needs will cause the algorithms to appear in the toolbox, but an error will be thrown when you try to execute them.

This is because the algorithm descriptions (needed to create the parameters dialog and provide the information needed about the algorithm) are not included with each application, but with QGIS instead. That is, they are part of QGIS, so you have them in your installation even if you have not installed any other software. Running the algorithm, however, needs the application binaries to be installed in your system.

17.14.1 Uma nota para utilizadores Windows

If you are not an advanced user and you are running QGIS on Windows, you might not be interested in reading the rest of this chapter. Make sure you install QGIS in your system using the standalone installer. That will automatically install SAGA, GRASS and OTB in your system and configure them so they can be run from QGIS. All the algorithms in the simplified view of the toolbox will be ready to be run without needing any further configuration. If installing through OSGeo4W application, make sure you select for installation SAGA and OTB as well.

If you want to know more about how these providers work, or if you want to use some algorithms not included in the simplified toolbox (such as R scripts), keep on reading.

17.14.2 Uma nota para os formatos dos ficheiros

When using an external software, opening a file in QGIS does not mean that it can be opened and processed as well in that other software. In most cases, other software can read what you have opened in QGIS, but in some cases, that might not be true. When using databases or uncommon file formats, whether for raster or vector layers, problems might arise. If that happens, try to use well-known file formats that you are sure are understood by both programs, and check the console output (in the history and log dialog) to know more about what is going wrong.

Usando as camadas raster GRASS, por exemplo, um dos casos em que pode ter problema e não seja possível completar o seu trabalho é se chamar um algoritmo externo usando uma camada como ficheiro de entrada. Por esta razão, essas camadas não irão aparecer como disponíveis para os algoritmos.

You should, however, find no problems at all with vector layers, since QGIS automatically converts from the original file format to one accepted by the external application before passing the layer to it. This adds extra processing time, which might be significant if the layer has a large size, so do not be surprised if it takes more time to process a layer from a DB connection than it does to process one of a similar size stored in a shapefile.

Providers not using external applications can process any layer that you can open in QGIS, since they open it for analysis through QGIS.

Regarding output formats, all formats supported by QGIS as output can be used, both for raster and vector layers. Some providers do not support certain formats, but all can export to common raster layer formats that can later be transformed by QGIS automatically. As in the case of input layers, if this conversion is needed, that might increase the processing time.

If the extension of the filename specified when calling an algorithm does not match the extension of any of the formats supported by QGIS, then a suffix will be added to set a default format. In the case of raster layers, the `.tif` extension is used, while `.shp` is used for vector layers.

17.14.3 Uma nota para as selecções da camada vectorial

External applications may also be made aware of the selections that exist in vector layers within QGIS. However, that requires rewriting all input vector layers, just as if they were originally in a format not supported by the external application. Only when no selection exists, or the *Use only selected features* option is not enabled in the processing general configuration, can a layer be directly passed to an external application.

Noutros casos, exportando apenas os elementos seleccionados é necessário, o que em alguns casos os tempos de execução serão mais longos.

SAGA

SAGA algorithms can be run from QGIS if you have SAGA installed in your system and you configure the processing framework properly so it can find SAGA executables. In particular, the SAGA command-line executable is needed to run SAGA algorithms.

If you are running Windows, both the stand-alone installer and the OSGeo4W installer include SAGA along with QGIS, and the path is automatically configured, so there is no need to do anything else.

If you have installed SAGA yourself (remember, you need version 2.1), the path to the SAGA executable must be configured. To do this, open the configuration dialog. In the *SAGA* block, you will find a setting named *SAGA Folder*. Enter the path to the folder where SAGA is installed. Close the configuration dialog, and now you are ready to run SAGA algorithms from QGIS.

If you are running Linux, SAGA binaries are not included with SEXTANTE, so you have to download and install the software yourself. Please check the SAGA website for more information. SAGA 2.1 is needed.

In this case, there is no need to configure the path to the SAGA executable, and you will not see those folders. Instead, you must make sure that SAGA is properly installed and its folder is added to the `PATH` environment variable. Just open a console and type `saga_cmd` to check that the system can find where the SAGA binaries are located.

17.14.4 Sobre as limitações do sistema de grelhas do SAGA

Most SAGA algorithms that require several input raster layers require them to have the same grid system. That is, they must cover the same geographic area and have the same cell size, so their corresponding grids match. When calling SAGA algorithms from QGIS, you can use any layer, regardless of its cell size and extent. When multiple raster layers are used as input for a SAGA algorithm, QGIS resamples them to a common grid system and then passes them to SAGA (unless the SAGA algorithm can operate with layers from different grid systems).

The definition of that common grid system is controlled by the user, and you will find several parameters in the SAGA group of the settings window to do so. There are two ways of setting the target grid system:

- Setting it manually. You define the extent by setting the values of the following parameters:
 - *Reamostragem do X min*
 - *Reamostragem do X máx*
 - *Reamostragem do Y min*
 - *Reamostragem do Y máx*
 - *Reamostragem do tamanho da célula*

Tenha em atenção que o QGIS irá reamostrar as camadas de entrada para essa extensão, mesmo que não se sobreponham.

- Setting it automatically from input layers. To select this option, just check the *Use min covering grid system for resampling* option. All the other settings will be ignored and the minimum extent that covers all the input layers will be used. The cell size of the target layer is the maximum of all cell sizes of the input layers.

Para algoritmos que não usam camadas raster múltiplas, ou para aquelas que não necessitam de um único sistema de grelha de entrada, não será feita uma reamostragem antes de chamar o SAGA, e esses parâmetros não serão usados.

17.14.5 Limitações para camadas multi-banda

Unlike QGIS, SAGA has no support for multi-band layers. If you want to use a multiband layer (such as an RGB or multispectral image), you first have to split it into single-banded images. To do so, you can use the ‘SAGA/Grid - Tools/Split RGB image’ algorithm (which creates three images from an RGB image) or the ‘SAGA/Grid - Tools/Extract band’ algorithm (to extract a single band).

17.14.6 Limitations in cell size

SAGA assumes that raster layers have the same cell size in the X and Y axis. If you are working with a layer with different values for horizontal and vertical cell size, you might get unexpected results. In this case, a warning will be added to the processing log, indicating that an input layer might not be suitable to be processed by SAGA.

17.14.7 Registando

When QGIS calls SAGA, it does so using its command-line interface, thus passing a set of commands to perform all the required operations. SAGA shows its progress by writing information to the console, which includes the percentage of processing already done, along with additional content. This output is filtered and used to update the progress bar while the algorithm is running.

Both the commands sent by QGIS and the additional information printed by SAGA can be logged along with other processing log messages, and you might find them useful to track in detail what is going on when QGIS runs a SAGA algorithm. You will find two settings, namely *Log console output* and *Log execution commands*, to activate that logging mechanism.

A maioria dos outros fornecedores que usam uma aplicação externa e chamam a partir da linha de comandos têm opções semelhantes, portanto irá encontrar noutros sítios da lista de configurações do processamento.

R. Creating R scripts

R integration in QGIS is different from that of SAGA in that there is not a predefined set of algorithms you can run (except for a few examples). Instead, you should write your scripts and call R commands, much like you would do from R, and in a very similar manner to what we saw in the section dedicated to processing scripts. This section shows you the syntax to use to call those R commands from QGIS and how to use QGIS objects (layers, tables) in them.

The first thing you have to do, as we saw in the case of SAGA, is to tell QGIS where your R binaries are located. You can do this using the *R folder* entry in the processing configuration dialog. Once you have set that parameter, you can start creating and executing your own R scripts.

Uma vez mais, isto é diferente no Linux, e só tem de ter certeza que a pasta do R está incluída na variável de ambiente PATH. Se conseguir iniciar o R, apenas introduza R na consola, e estará pronto a começar.

To add a new algorithm that calls an R function (or a more complex R script that you have developed and you would like to have available from QGIS), you have to create a script file that tells the processing framework how to perform that operation and the corresponding R commands to do so.

R script files have the extension `.rsx`, and creating them is pretty easy if you just have a basic knowledge of R syntax and R scripting. They should be stored in the R scripts folder. You can set this folder in the *R settings group* (available from the processing settings dialog), just like you do with the folder for regular processing scripts.

Let's have a look at a very simple script file, which calls the R method `spsample` to create a random grid within the boundary of the polygons in a given polygon layer. This method belongs to the `maptools` package. Since almost all the algorithms that you might like to incorporate into QGIS will use or generate spatial data, knowledge of spatial packages like `maptools` and, especially, `sp`, is mandatory.

```
##polyg=vector
##numpoints=number 10
##output=output vector
##sp=group
pts=spsample(polyg,numpoints,type="random")
output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```

The first lines, which start with a double Python comment sign (`##`), tell QGIS the inputs of the algorithm described in the file and the outputs that it will generate. They work with exactly the same syntax as the SEXTANTE scripts that we have already seen, so they will not be described here again.

When you declare an input parameter, QGIS uses that information for two things: creating the user interface to ask the user for the value of that parameter and creating a corresponding R variable that can later be used as input for R commands.

In the above example, we are declaring an input of type `vector` named `polyg`. When executing the algorithm, QGIS will open in R the layer selected by the user and store it in a variable also named `polyg`. So, the name of a parameter is also the name of the variable that we can use in R for accessing the value of that parameter (thus, you should avoid using reserved R words as parameter names).

Spatial elements such as vector and raster layers are read using the `readOGR()` and `brick()` commands (you do not have to worry about adding those commands to your description file – QGIS will do it), and they are stored as `Spatial*DataFrame` objects. Table fields are stored as strings containing the name of the selected field.

Tables are opened using the `read.csv()` command. If a table entered by the user is not in CSV format, it will be converted prior to importing it into R.

Additionally, raster files can be read using the `readGDAL()` command instead of `brick()` by using the `##userreadgdal`.

If you are an advanced user and do not want QGIS to create the object representing the layer, you can use the `##passfilename` tag to indicate that you prefer a string with the filename instead. In this case, it is up to you to open the file before performing any operation on the data it contains.

Com a informação em cima, podemos agora perceber a primeira linha do nosso primeiro script exemplo (a primeira linha que não começa com o comentário Python).

```
pts=spsample(polyg,numpoints,type="random")
```

A variável `polyg` já contém o objecto `SpatialPolygonsDataFrame`, portanto pode ser usado para chamar o método `spsample`, tal como o `numpoints`, que indica o número de pontos a ser adicionados à grelha de amostra criada.

Since we have declared an output of type vector named `out`, we have to create a variable named `out` and store a `Spatial*DataFrame` object in it (in this case, a `SpatialPointsDataFrame`). You can use any name for your intermediate variables. Just make sure that the variable storing your final result has the same name that you used to declare it, and that it contains a suitable value.

In this case, the result obtained from the `spsample` method has to be converted explicitly into a `SpatialPointsDataFrame` object, since it is itself an object of class `ppp`, which is not a suitable class to be returned to QGIS.

If your algorithm generates raster layers, the way they are saved will depend on whether or not you have used the `#dontuserasterpackage` option. In you have used it, layers are saved using the `writeGDAL()` method. If not, the `writeRaster()` method from the `raster` package will be used.

Se usou a opção `#passfilename`, os ficheiros de saída são gerados usando o pacote `raster` (com `writeRaster()`), mesmo que não seja usado nos ficheiros de entrada.

If your algorithm does not generate any layer, but rather a text result in the console instead, you have to indicate that you want the console to be shown once the execution is finished. To do so, just start the command lines that produce the results you want to print with the `>` ('greater') sign. The output of all other lines will not be shown. For instance, here is the description file of an algorithm that performs a normality test on a given field (column) of the attributes of a vector layer:

```
##layer=vector
##field=field layer
##nortest=group
library(nortest)
>lillie.test(layer[[field]])
```

The output of the last line is printed, but the output of the first is not (and neither are the outputs from other command lines added automatically by QGIS).

Se o seu algoritmo criar qualquer tipo de gráficos (usando o método `plot()`), adicione a seguinte linha:

```
##showplots
```

This will cause QGIS to redirect all R graphical outputs to a temporary file, which will be opened once R execution has finished.

Tanto os gráficos como os resultados da consola serão exibidos no gestor de resultados do processamento.

For more information, please check the script files provided with SEXTANTE. Most of them are rather simple and will greatly help you understand how to create your own scripts.

Note: `rgdal` and `maptools` libraries are loaded by default, so you do not have to add the corresponding `library()` commands (you just have to make sure that those two packages are installed in your R distribution). However, other additional libraries that you might need have to be explicitly loaded. Just add the necessary commands at the beginning of your script. You also have to make sure that the corresponding packages are installed in the R distribution used by QGIS. The processing framework will not take care of any package installation. If you run a script that requires a package that is not installed, the execution will fail, and SEXTANTE will try to detect which packages are missing. You must install those missing libraries manually before you can run the algorithm.

GRASS

Configurar o GRASS não é muito diferente de configurar o SAGA. Primeiro, o caminho para a pasta GRASS tem de ser definida, mas apenas se estiver a correr o Windows. Adicionalmente, um interpretador shell, (normalmente

o `msys.exe`, que pode ser encontrado na maioria das distribuições do GRASS para o Windows) tem de ser definido e o seu caminho configurado.

By default, the processing framework tries to configure its GRASS connector to use the GRASS distribution that ships along with QGIS. This should work without problems in most systems, but if you experience problems, you might have to configure the GRASS connector manually. Also, if you want to use a different GRASS installation, you can change that setting and point to the folder where the other version is installed. GRASS 6.4 is needed for algorithms to work correctly.

Se está a trabalhar em Linux, só necessita de ter a certeza que o GRASS está correctamente instalado, e pode ser executado sem problema a partir da consola.

GRASS algorithms use a region for calculations. This region can be defined manually using values similar to the ones found in the SAGA configuration, or automatically, taking the minimum extent that covers all the input layers used to execute the algorithm each time. If the latter approach is the behaviour you prefer, just check the *Use min covering region* option in the GRASS configuration parameters.

The last parameter that has to be configured is related to the mapset. A mapset is needed to run GRASS, and the processing framework creates a temporary one for each execution. You have to specify if the data you are working with uses geographical (lat/lon) coordinates or projected ones.

GDAL

No additional configuration is needed to run GDAL algorithms. Since they are already incorporated into QGIS, the algorithms can infer their configuration from it.



Orfeo Toolbox

Orfeo Toolbox (OTB) algorithms can be run from QGIS if you have OTB installed in your system and you have configured QGIS properly, so it can find all necessary files (command-line tools and libraries).

As in the case of SAGA, OTB binaries are included in the stand-alone installer for Windows, but they are not included if you are running Linux, so you have to download and install the software yourself. Please check the OTB website for more information.

Once OTB is installed, start QGIS, open the processing configuration dialog and configure the OTB algorithm provider. In the *Orfeo Toolbox (image analysis)* block, you will find all settings related to OTB. First, ensure that algorithms are enabled.

Then, configure the path to the folder where OTB command-line tools and libraries are installed:

-  Usually *OTB applications folder* points to `/usr/lib/otb/applications` and *OTB command line tools folder* is `/usr/bin`.
-  If you use the OSGeo4W installer, then install `otb-bin` package and enter `C:\OSGeo4W\apps\orfeotoolbox\applications` as *OTB applications folder* and `C:\OSGeo4W\bin` as *OTB command line tools folder*. These values should be configured by default, but if you have a different OTB installation, configure them to the corresponding values in your system.

TauDEM

To use this provider, you need to install TauDEM command line tools.

17.14.8 Windows

Please visit the [TauDEM homepage](#) for installation instructions and precompiled binaries for 32-bit and 64-bit systems. **IMPORTANT:** You need TauDEM 5.0.6 executables. Version 5.2 is currently not supported.

17.14.9 Linux

There are no packages for most Linux distributions, so you should compile TauDEM by yourself. As TauDEM uses MPICH2, first install it using your favorite package manager. Alternatively, TauDEM works fine with Open MPI, so you can use it instead of MPICH2.

Download TauDEM 5.0.6 [source code](#) and extract the files in some folder.

Open the `linearpart.h` file, and after line

```
#include "mpi.h"
```

add a new line with

```
#include <stdint.h>
```

e irá obter

```
#include "mpi.h"
#include <stdint.h>
```

Save the changes and close the file. Now open `tiffIO.h`, find line `#include "stdint.h"` and replace quotes (" ") with `<>`, so you'll get

```
#include <stdint.h>
```

Save the changes and close the file. Create a build directory and cd into it

```
mkdir build
cd build
```

Configure your build with the command

```
CXX=mpicxx cmake -DCMAKE_INSTALL_PREFIX=/usr/local ..
```

e de seguida compile

```
make
```

Finally, to install TauDEM into `/usr/local/bin`, run

```
sudo make install
```

.

17.15 The QGIS Commander

Processing includes a practical tool that allows you to run algorithms without having to use the toolbox, but just by typing the name of the algorithm you want to run.

This tool is known as the *QGIS commander*, and it is just a simple text box with autocompletion where you type the command you want to run.

The Commander is started from the *Analysis* menu or, more practically, by pressing `Shift + Ctrl + M` (you can change that default keyboard shortcut in the QGIS configuration if you prefer a different one). Apart from executing Processing algorithms, the Commander gives you access to most of the functionality in QGIS, which means that it gives you a practical and efficient way of running QGIS tasks and allows you to control QGIS with reduced usage of buttons and menus.

Moreover, the Commander is configurable, so you can add your custom commands and have them just a few keystrokes away, making it a powerful tool to help you become more productive in your daily work with QGIS.

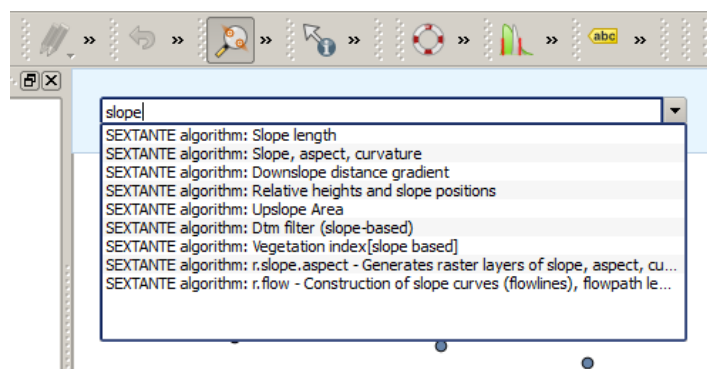


Figura 17.29: The QGIS Commander

17.15.1 Comandos disponíveis

The commands available in the Commander fall in the following categories:

- Processing algorithms. These are shown as Processing algorithm: <name of the algorithm>.
- Menu items. These are shown as Menu item: <menu entry text>. All menus items available from the QGIS interface are available, even if they are included in a submenu.
- Python functions. You can create short Python functions that will be then included in the list of available commands. They are shown as Function: <function name>.

To run any of the above, just start typing and then select the corresponding element from the list of available commands that appears after filtering the whole list of commands with the text you have entered.

No caso de chamar uma função Python, pode seleccionar uma entrada da lista, que tem o prefixo de Função: (para a instância, Função: removeall), ou apenas escreva directamente o nome da função ('removeall no exemplo anterior). Não existe necessidade de adicionar parêntesis após o nome da função.

17.15.2 Criando funções personalizadas

As funções personalizadas são adicionadas ao introduzir o código Python correspondente no ficheiro `commands.py` que pode ser encontrado `.qgis/sextante/commander` directory na pasta do utilizador. É apenas um ficheiro Python simples onde pode adicionar as funções que necessita.

The file is created with a few example functions the first time you open the Commander. If you haven't launched the Commander yet, you can create the file yourself. To edit the commands file, use your favorite text editor. You can also use a built-in editor by calling the `edit` command from the Commander. It will open the editor with the commands file, and you can edit it directly and then save your changes.

Por exemplo, pode adicionar a seguinte função, que remove todas as camadas:

```
from qgis.gui import *

def removeall():
    mapreg = QgsMapLayerRegistry.instance()
    mapreg.removeAllMapLayers()
```

Once you have added the function, it will be available in the Commander, and you can invoke it by typing `removeall`. There is no need to do anything apart from writing the function itself.

Functions can receive parameters. Add `*args` to your function definition to receive arguments. When calling the function from the Commander, parameters have to be passed separated by spaces.

Aqui está um exemplo de uma função que carrega uma camada e que tome como parâmetro o nome do ficheiro da camada para carregar.

```
import processing
```

```
def load(*args):  
    processing.load(args[0])
```

If you want to load the layer in `/home/myuser/points.shp`, type `load /home/myuser/points.shp` in the Commander text box.

.

Processing providers and algorithms

18.1 GDAL algorithm provider

GDAL (Geospatial Data Abstraction Library) is a translator library for raster and vector geospatial data formats.

18.1.1 GDAL analysis

Aspect

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Use Zevenbergen&Thorne formula (instead of the Horn's one) [boolean] <put parameter description here>

Default: *False*

Return trigonometric angle (instead of azimuth) [boolean] <put parameter description here>

Default: *False*

Return 0 for flat (instead of -9999) [boolean] <put parameter description here>

Default: *False*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:aspect', input, band, compute_edges, zevenbergen, trig_angle, zero_flat)
```

See also

Color relief

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Color configuration file [file] <put parameter description here>

Matching mode [selection] <put parameter description here>

Options:

- 0 — “0,0,0,0” RGBA
- 1 — Exact color
- 2 — Nearest color

Default: *0*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:colorrelief', input, band, compute_edges, color_table, match_mode, out)
```

See also

Fill nodata

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Search distance [number] <put parameter description here>

Default: *100*

Smooth iterations [number] <put parameter description here>

Default: *0*

Band to operate on [number] <put parameter description here>

Default: *1*

Validity mask [raster] Optional.

<put parameter description here>

Do not use default validity mask [boolean] <put parameter description here>

Default: *False*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:fillnodata', input, distance, iterations, band, mask, no_default_mask,
```

See also

Grid (Moving average)

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Z field [tablefield: numeric] Optional.

<put parameter description here>

Radius 1 [number] <put parameter description here>

Default: *0.0*

Radius 2 [number] <put parameter description here>

Default: *0.0*

Min points [number] <put parameter description here>

Default: *0.0*

Angle [number] <put parameter description here>

Default: *0.0*

Nodata [number] <put parameter description here>

Default: 0.0

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:gridaverage', input, z_field, radius_1, radius_2, min_points, angle, n
```

See also

Grid (Data metrics)

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Z field [tablefield: numeric] Optional.

<put parameter description here>

Metrics [selection] <put parameter description here>

Options:

- 0 — Minimum
- 1 — Maximum
- 2 — Range

- 3 — Count
- 4 — Average distance
- 5 — Average distance between points

Default: 0

Radius 1 [number] <put parameter description here>

Default: 0.0

Radius 2 [number] <put parameter description here>

Default: 0.0

Min points [number] <put parameter description here>

Default: 0.0

Angle [number] <put parameter description here>

Default: 0.0

Nodata [number] <put parameter description here>

Default: 0.0

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:griddatametrics', input, z_field, metric, radius_1, radius_2, min_point
```

See also

Grid (Inverse distance to a power)

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Z field [tablefield: numeric] Optional.

<put parameter description here>

Power [number] <put parameter description here>

Default: 2.0

Smoothing [number] <put parameter description here>

Default: 0.0

Radius 1 [number] <put parameter description here>

Default: 0.0

Radius 2 [number] <put parameter description here>

Default: 0.0

Max points [number] <put parameter description here>

Default: 0.0

Min points [number] <put parameter description here>

Default: 0.0

Angle [number] <put parameter description here>

Default: 0.0

Nodata [number] <put parameter description here>

Default: 0.0

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32

- 10 — CFloat64

Default: 5

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:gridinvdist', input, z_field, power, smothing, radius_1, radius_2, max.
```

See also

Grid (Nearest neighbor)

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Z field [tablefield: numeric] Optional.

<put parameter description here>

Radius 1 [number] <put parameter description here>

Default: 0.0

Radius 2 [number] <put parameter description here>

Default: 0.0

Angle [number] <put parameter description here>

Default: 0.0

Nodata [number] <put parameter description here>

Default: 0.0

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32

- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:gridnearestneighbor', input, z_field, radius_1, radius_2, angle, nodata)
```

See also

Hillshade

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: 1

Compute edges [boolean] <put parameter description here>

Default: *False*

Use Zevenbergen&Thorne formula (instead of the Horn's one) [boolean] <put parameter description here>

Default: *False*

Z factor (vertical exaggeration) [number] <put parameter description here>

Default: 1.0

Scale (ratio of vert. units to horiz.) [number] <put parameter description here>

Default: 1.0

Azimuth of the light [number] <put parameter description here>

Default: 315.0

Altitude of the light [number] <put parameter description here>

Default: 45.0

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:hillshade', input, band, compute_edges, zevenbergen, z_factor, scale, a
```

See also

Near black

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

How far from black (white) [number] <put parameter description here>

Default: *15*

Search for nearly white pixels instead of nearly black [boolean] <put parameter description here>

Default: *False*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:nearblack', input, near, white, output)
```

See also

Proximity (raster distance)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Values [string] <put parameter description here>

Default: *(not set)*

Dist units [selection] <put parameter description here>

Options:

- 0 — GEO
- 1 — PIXEL

Default: 0

Max dist (negative value to ignore) [number] <put parameter description here>

Default: -1

No data (negative value to ignore) [number] <put parameter description here>

Default: -1

Fixed buf val (negative value to ignore) [number] <put parameter description here>

Default: -1

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:proximity', input, values, units, max_dist, nodata, buf_val, rtype, out)
```

See also

Roughness

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: 1

Compute edges [boolean] <put parameter description here>

Default: *False*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:roughness', input, band, compute_edges, output)
```

See also

Sieve

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Threshold [number] <put parameter description here>

Default: 2

Pixel connection [selection] <put parameter description here>

Options:

- 0 — 4
- 1 — 8

Default: 0

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:sieve', input, threshold, connections, output)
```

See also

Slope

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Use Zevenbergen&Thorne formula (instead of the Horn's one) [boolean] <put parameter description here>

Default: *False*

Slope expressed as percent (instead of degrees) [boolean] <put parameter description here>

Default: *False*

Scale (ratio of vert. units to horiz.) [number] <put parameter description here>

Default: *1.0*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalgr:slope', input, band, compute_edges, zevenbergen, as_percent, scale, out)
```

See also

TPI (Topographic Position Index)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:tpitopographicpositionindex', input, band, compute_edges, output)
```

See also

TRI (Terrain Ruggedness Index)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:triterrainruggednessindex', input, band, compute_edges, output)
```

See also

.

18.1.2 GDAL conversion

gdal2xyz

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Outputs

Output file [table] <put output description here>

Console usage

```
processing.runalg('gdalogr:gdal2xyz', input, band, output)
```

See also

PCT to RGB

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band to convert [selection] <put parameter description here>

Options:

- 0 — 1
- 1 — 2
- 2 — 3
- 3 — 4
- 4 — 5
- 5 — 6
- 6 — 7
- 7 — 8
- 8 — 9
- 9 — 10
- 10 — 11
- 11 — 12
- 12 — 13
- 13 — 14
- 14 — 15
- 15 — 16
- 16 — 17
- 17 — 18
- 18 — 19
- 19 — 20
- 20 — 21

- 21 — 22
- 22 — 23
- 23 — 24
- 24 — 25

Default: 0

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:pcttorgb', input, nband, output)
```

See also

Polygonize (raster to vector)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Output field name [string] <put parameter description here>

Default: *DN*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:polygonize', input, field, output)
```

See also

Rasterize (vector to raster)

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Attribute field [tablefield: any] <put parameter description here>

Write values inside an existing raster layer (*) [boolean] <put parameter description here>

Default: *False*

Set output raster size (ignored if above option is checked) [selection] <put parameter description here>

Options:

- 0 — Output size in pixels
- 1 — Output resolution in map units per pixel

Default: *1*

Horizontal [number] <put parameter description here>

Default: *100.0*

Vertical [number] <put parameter description here>

Default: *100.0*

Raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: *0*

Outputs

Output layer: mandatory to choose an existing raster layer if the (*) option is selected
<put output description here>

Console usage

```
processing.runalg('gdalogr:rasterize', input, field, writeover, dimensions, width, height, rtype,
```


See also

RGB to PCT

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Number of colors [number] <put parameter description here>

Default: 2

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:rgbtocpct', input, ncolors, output)
```

See also

Translate (convert format)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Set the size of the output file (In pixels or %) [number] <put parameter description here>

Default: 100

Output size is a percentage of input size [boolean] <put parameter description here>

Default: *True*

Nodata value, leave as none to take the nodata value from input [string] <put parameter description here>

Default: *none*

Expand [selection] <put parameter description here>

Options:

- 0 — none
- 1 — gray
- 2 — rgb

- 3 — rgba

Default: 0

Output projection for output file [leave blank to use input projection] [crs]
<put parameter description here>

Default: *None*

Subset based on georeferenced coordinates [extent] <put parameter description here>

Default: *0,1,0,1*

Copy all subdatasets of this file to individual output files [boolean] <put parameter description here>

Default: *False*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:translate', input, outsize, outsize_perc, no_data, expand, srs, projwin)
```

See also

.

18.1.3 GDAL extraction

Clip raster by extent

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Nodata value, leave as none to take the nodata value from input [string] <put parameter description here>

Default: *none*

Clipping extent [extent] <put parameter description here>

Default: *0,1,0,1*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:cliprasterbyextent', input, no_data, projwin, extra, output)
```

See also

Clip raster by mask layer

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Mask layer [vector: polygon] <put parameter description here>

Nodata value, leave as none to take the nodata value from input [string] <put parameter description here>

Default: *none*

Create and output alpha band [boolean] <put parameter description here>

Default: *False*

Keep resolution of output raster [boolean] <put parameter description here>

Default: *False*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:cliprasterbymasklayer', input, mask, no_data, alpha_band, keep_resolut
```

See also

Contour

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Interval between contour lines [number] <put parameter description here>

Default: *10.0*

Attribute name (if not set, no elevation attribute is attached) [string]

Optional.

<put parameter description here>

Default: *ELEV*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output file for contour lines (vector) [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:contour', input_raster, interval, field_name, extra, output_vector)
```

See also

.

18.1.4 GDAL miscellaneous

Build Virtual Raster

Description

<put algorithm description here>

Parameters

Input layers [**multipleinput: rasters**] <put parameter description here>

Resolution [**selection**] <put parameter description here>

Options:

- 0 — average
- 1 — highest
- 2 — lowest

Default: *0*

Layer stack [**boolean**] <put parameter description here>

Default: *True*

Allow projection difference [**boolean**] <put parameter description here>

Default: *False*

Outputs

Output layer [**raster**] <put output description here>

Console usage

```
processing.runalg('gdalogr:buildvirtualraster', input, resolution, separate, proj_difference, outp
```

See also

Merge

Description

<put algorithm description here>

Parameters

Input layers [**multipleinput: rasters**] <put parameter description here>

Grab pseudocolor table from first layer [**boolean**] <put parameter description here>

Default: *False*

Layer stack [**boolean**] <put parameter description here>

Default: *False*

Output raster type [**selection**] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output layer [**raster**] <put output description here>

Console usage

```
processing.runalg('gdalogr:merge', input, pct, separate, rtype, output)
```

See also

Build overviews (pyramids)

Description

<put algorithm description here>

Parameters

Input layer [**raster**] <put parameter description here>

Overview levels [**string**] <put parameter description here>

Default: 2 4 8 16

Remove all existing overviews [boolean] <put parameter description here>

Default: *False*

Resampling method [selection] <put parameter description here>

Options:

- 0 — nearest
- 1 — average
- 2 — gauss
- 3 — cubic
- 4 — average_mp
- 5 — average_magphase
- 6 — mode

Default: *0*

Overview format [selection] <put parameter description here>

Options:

- 0 — Internal (if possible)
- 1 — External (GTiff .ovr)
- 2 — External (ERDAS Imagine .aux)

Default: *0*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:overviews', input, levels, clean, resampling_method, format)
```

See also

Information

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Suppress GCP info [boolean] <put parameter description here>

Default: *False*

Suppress metadata info [boolean] <put parameter description here>

Default: *False*

Outputs

Layer information [html] <put output description here>

Console usage

```
processing.runalg('gdalorg:rasterinfo', input, nogcp, nometadata, output)
```

See also

.

18.1.5 GDAL projections

Extract projection

Description

<put algorithm description here>

Parameters

Input file [raster] <put parameter description here>

Create also .prj file [boolean] <put parameter description here>

Default: *False*

Outputs

Console usage

```
processing.runalg('gdalorg:extractprojection', input, prj_file)
```

See also

Warp (reproject)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Source SRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Destination SRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Output file resolution in target georeferenced units (leave 0 for no change) [number]
 <put parameter description here>

Default: *0.0*

Resampling method [selection] <put parameter description here>

Options:

- 0 — near
- 1 — bilinear
- 2 — cubic
- 3 — cubicspline
- 4 — lanczos

Default: *0*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: *5*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdal:warpreproject', input, source_srs, dest_srs, tr, method, extra, rtype,
```

See also

.

18.1.6 OGR conversion

Convert format

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Destination Format [selection] <put parameter description here>

Options:

- 0 — ESRI Shapefile
- 1 — GeoJSON
- 2 — GeoRSS
- 3 — SQLite
- 4 — GMT
- 5 — MapInfo File
- 6 — INTERLIS 1
- 7 — INTERLIS 2
- 8 — GML
- 9 — Geoconcept
- 10 — DXF
- 11 — DGN
- 12 — CSV
- 13 — BNA
- 14 — S57
- 15 — KML
- 16 — GPX
- 17 — PGDump
- 18 — GPSTrackMaker
- 19 — ODS
- 20 — XLSX
- 21 — PDF

Default: 0

Creation Options [string] Optional.

<put parameter description here>

Default: (*not set*)

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:convertformat', input_layer, format, options, output_layer)
```

See also

.

18.1.7 OGR geoprocessing

Clip vectors by extent

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Clip extent [extent] <put parameter description here>

Default: *0,1,0,1*

Additional creation Options [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:clipvectorsbyextent', input_layer, clip_extent, options, output_layer)
```

See also

Clip vectors by polygon

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Clip layer [vector: polygon] <put parameter description here>

Additional creation Options [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:clipvectorsbypolygon', input_layer, clip_layer, options, output_layer)
```

See also

.

18.1.8 OGR miscellaneous

Execute SQL

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

SQL [string] <put parameter description here>

Default: *(not set)*

Outputs

SQL result [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:executesql', input, sql, output)
```

See also**Import Vector into PostGIS database (available connections)****Description**

<put algorithm description here>

Parameters

Database (connection name) [selection] <put parameter description here>

Options:

- 0 — local

Default: 0

Input layer [vector: any] <put parameter description here>

Output geometry type [selection] <put parameter description here>

Options:

- 0 —
- 1 — NONE
- 2 — GEOMETRY
- 3 — POINT
- 4 — LINESTRING
- 5 — POLYGON
- 6 — GEOMETRYCOLLECTION
- 7 — MULTIPOINT
- 8 — MULTIPOLYGON
- 9 — MULTILINESTRING

Default: 5

Input CRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Output CRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Schema name [string] Optional.

<put parameter description here>

Default: *public*

Table name, leave blank to use input name [string] Optional.

<put parameter description here>

Default: *(not set)*

Primary Key [string] Optional.

<put parameter description here>

Default: *id*

Geometry column name [string] Optional.

<put parameter description here>

Default: *geom*

Vector dimensions [selection] <put parameter description here>

Options:

- 0 — 2
- 1 — 3

Default: *0*

Distance tolerance for simplification [string] Optional.

<put parameter description here>

Default: *(not set)*

Maximum distance between 2 nodes (densification) [string] Optional.

<put parameter description here>

Default: *(not set)*

Select features by extent (defined in input layer CRS) [extent] <put parameter description here>

Default: *0,1,0,1*

Clip the input layer using the above (rectangle) extent [boolean] <put parameter description here>

Default: *False*

Select features using a SQL "WHERE" statement (Ex: column="value") [string] Optional.

<put parameter description here>

Default: *(not set)*

Group "n" features per transaction (Default: 20000) [string] Optional.

<put parameter description here>

Default: *(not set)*

Overwrite existing table? [boolean] <put parameter description here>

Default: *True*

Append to existing table? [boolean] <put parameter description here>

Default: *False*

Append and add new fields to existing table? [boolean] <put parameter description here>

Default: *False*

Do not launder columns/table name/s? [boolean] <put parameter description here>

Default: *False*

Do not create Spatial Index? [boolean] <put parameter description here>

Default: *False*

Continue after a failure, skipping the failed feature [boolean] <put parameter description here>

Default: *False*

Additional creation options [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Console usage

```
processing.runalg('gdalogr:importvectorintopostgisdatabaseavailableconnections', database, input_
```

See also

Import Vector into PostGIS database (new connection)

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Output geometry type [selection] <put parameter description here>

Options:

- 0 —
- 1 — NONE
- 2 — GEOMETRY
- 3 — POINT
- 4 — LINESTRING
- 5 — POLYGON
- 6 — GEOMETRYCOLLECTION
- 7 — MULTIPOINT
- 8 — MULTIPOLYGON
- 9 — MULTILINESTRING

Default: 5

Input CRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Output CRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Host [string] <put parameter description here>

Default: *localhost*

Port [string] <put parameter description here>

Default: *5432*

Username [string] <put parameter description here>

Default: *(not set)*

Database Name [string] <put parameter description here>

Default: *(not set)*

Password [string] <put parameter description here>

Default: *(not set)*

Schema name [string] Optional.

<put parameter description here>

Default: *public*

Table name, leave blank to use input name [string] Optional.

<put parameter description here>

Default: *(not set)*

Primary Key [string] Optional.

<put parameter description here>

Default: *id*

Geometry column name [string] Optional.

<put parameter description here>

Default: *geom*

Vector dimensions [selection] <put parameter description here>

Options:

- 0 — 2
- 1 — 3

Default: *0*

Distance tolerance for simplification [string] Optional.

<put parameter description here>

Default: *(not set)*

Maximum distance between 2 nodes (densification) [string] Optional.

<put parameter description here>

Default: *(not set)*

Select features by extent (defined in input layer CRS) [extent] <put parameter description here>

Default: *0,1,0,1*

Clip the input layer using the above (rectangle) extent [boolean] <put parameter description here>

Default: *False*

Select features using a SQL "WHERE" statement (Ex: column="value") [string] Optional.

<put parameter description here>

Default: *(not set)*

Group "n" features per transaction (Default: 20000) [string] Optional.

<put parameter description here>

Default: *(not set)*

Overwrite existing table? [boolean] <put parameter description here>

Default: *True*

Append to existing table? [boolean] <put parameter description here>

Default: *False*

Append and add new fields to existing table? [boolean] <put parameter description here>

Default: *False*

Do not launder columns/table name/s? [boolean] <put parameter description here>

Default: *False*

Do not create Spatial Index? [boolean] <put parameter description here>

Default: *False*

Continue after a failure, skipping the failed feature [boolean] <put parameter description here>

Default: *False*

Additional creation options [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Console usage

```
processing.runalg('gdalogr:importvectorintopostgisdatabasewconnection', input_layer, gtype, s_s
```

See also

Information

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Outputs

Layer information [html] <put output description here>

Console usage

```
processing.runalg('gdalogr:information', input, output)
```

See also

.

18.2 LAStools

LAStools is a collection of highly efficient, multicore command line tools for LiDAR data processing.

18.2.1 las2las_filter

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

input LAS/LAZ file [**file**] Optional.

<put parameter description here>

filter (by return, classification, flags) [**selection**] <put parameter description here>

Options:

- 0 — —
- 1 — keep_last
- 2 — keep_first
- 3 — keep_middle
- 4 — keep_single
- 5 — drop_single
- 6 — keep_double
- 7 — keep_class 2
- 8 — keep_class 2 8
- 9 — keep_class 8
- 10 — keep_class 6
- 11 — keep_class 9
- 12 — keep_class 3 4 5
- 13 — keep_class 2 6
- 14 — drop_class 7
- 15 — drop_withheld

Default: 0

second filter (by return, classification, flags) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — keep_last
- 2 — keep_first
- 3 — keep_middle
- 4 — keep_single
- 5 — drop_single
- 6 — keep_double
- 7 — keep_class 2
- 8 — keep_class 2 8
- 9 — keep_class 8
- 10 — keep_class 6
- 11 — keep_class 9
- 12 — keep_class 3 4 5
- 13 — keep_class 2 6
- 14 — drop_class 7
- 15 — drop_withheld

Default: 0

filter (by coordinate, intensity, GPS time, ...) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — clip_x_above
- 2 — clip_x_below
- 3 — clip_y_above
- 4 — clip_y_below
- 5 — clip_z_above
- 6 — clip_z_below
- 7 — drop_intensity_above
- 8 — drop_intensity_below
- 9 — drop_gps_time_above
- 10 — drop_gps_time_below
- 11 — drop_scan_angle_above
- 12 — drop_scan_angle_below
- 13 — keep_point_source
- 14 — drop_point_source
- 15 — drop_point_source_above

- 16 — drop_point_source_below
- 17 — keep_user_data
- 18 — drop_user_data
- 19 — drop_user_data_above
- 20 — drop_user_data_below
- 21 — keep_every_nth
- 22 — keep_random_fraction
- 23 — thin_with_grid

Default: 0

value for filter (by coordinate, intensity, GPS time, ...) [string] <put parameter description here>

Default: *(not set)*

second filter (by coordinate, intensity, GPS time, ...) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — clip_x_above
- 2 — clip_x_below
- 3 — clip_y_above
- 4 — clip_y_below
- 5 — clip_z_above
- 6 — clip_z_below
- 7 — drop_intensity_above
- 8 — drop_intensity_below
- 9 — drop_gps_time_above
- 10 — drop_gps_time_below
- 11 — drop_scan_angle_above
- 12 — drop_scan_angle_below
- 13 — keep_point_source
- 14 — drop_point_source
- 15 — drop_point_source_above
- 16 — drop_point_source_below
- 17 — keep_user_data
- 18 — drop_user_data
- 19 — drop_user_data_above
- 20 — drop_user_data_below
- 21 — keep_every_nth
- 22 — keep_random_fraction
- 23 — thin_with_grid

Default: 0

value for second filter (by coordinate, intensity, GPS time, ...) [string] <put parameter description here>

Default: *(not set)*

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:las2lasfilter', verbose, input_laslaz, filter_return_class_flags1, ...)
```

See also

18.2.2 las2las_project

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

source projection [selection] <put parameter description here>

Options:

- 0 — —
- 1 — utm
- 2 — sp83
- 3 — sp27
- 4 — longlat
- 5 — latlong

Default: *0*

source utm zone [selection] <put parameter description here>

Options:

- 0 — —
- 1 — 1 (north)
- 2 — 2 (north)
- 3 — 3 (north)
- 4 — 4 (north)
- 5 — 5 (north)
- 6 — 6 (north)

- 7 — 7 (north)
- 8 — 8 (north)
- 9 — 9 (north)
- 10 — 10 (north)
- 11 — 11 (north)
- 12 — 12 (north)
- 13 — 13 (north)
- 14 — 14 (north)
- 15 — 15 (north)
- 16 — 16 (north)
- 17 — 17 (north)
- 18 — 18 (north)
- 19 — 19 (north)
- 20 — 20 (north)
- 21 — 21 (north)
- 22 — 22 (north)
- 23 — 23 (north)
- 24 — 24 (north)
- 25 — 25 (north)
- 26 — 26 (north)
- 27 — 27 (north)
- 28 — 28 (north)
- 29 — 29 (north)
- 30 — 30 (north)
- 31 — 31 (north)
- 32 — 32 (north)
- 33 — 33 (north)
- 34 — 34 (north)
- 35 — 35 (north)
- 36 — 36 (north)
- 37 — 37 (north)
- 38 — 38 (north)
- 39 — 39 (north)
- 40 — 40 (north)
- 41 — 41 (north)
- 42 — 42 (north)
- 43 — 43 (north)
- 44 — 44 (north)
- 45 — 45 (north)

- 46 — 46 (north)
- 47 — 47 (north)
- 48 — 48 (north)
- 49 — 49 (north)
- 50 — 50 (north)
- 51 — 51 (north)
- 52 — 52 (north)
- 53 — 53 (north)
- 54 — 54 (north)
- 55 — 55 (north)
- 56 — 56 (north)
- 57 — 57 (north)
- 58 — 58 (north)
- 59 — 59 (north)
- 60 — 60 (north)
- 61 — 1 (south)
- 62 — 2 (south)
- 63 — 3 (south)
- 64 — 4 (south)
- 65 — 5 (south)
- 66 — 6 (south)
- 67 — 7 (south)
- 68 — 8 (south)
- 69 — 9 (south)
- 70 — 10 (south)
- 71 — 11 (south)
- 72 — 12 (south)
- 73 — 13 (south)
- 74 — 14 (south)
- 75 — 15 (south)
- 76 — 16 (south)
- 77 — 17 (south)
- 78 — 18 (south)
- 79 — 19 (south)
- 80 — 20 (south)
- 81 — 21 (south)
- 82 — 22 (south)
- 83 — 23 (south)
- 84 — 24 (south)

- 85 — 25 (south)
- 86 — 26 (south)
- 87 — 27 (south)
- 88 — 28 (south)
- 89 — 29 (south)
- 90 — 30 (south)
- 91 — 31 (south)
- 92 — 32 (south)
- 93 — 33 (south)
- 94 — 34 (south)
- 95 — 35 (south)
- 96 — 36 (south)
- 97 — 37 (south)
- 98 — 38 (south)
- 99 — 39 (south)
- 100 — 40 (south)
- 101 — 41 (south)
- 102 — 42 (south)
- 103 — 43 (south)
- 104 — 44 (south)
- 105 — 45 (south)
- 106 — 46 (south)
- 107 — 47 (south)
- 108 — 48 (south)
- 109 — 49 (south)
- 110 — 50 (south)
- 111 — 51 (south)
- 112 — 52 (south)
- 113 — 53 (south)
- 114 — 54 (south)
- 115 — 55 (south)
- 116 — 56 (south)
- 117 — 57 (south)
- 118 — 58 (south)
- 119 — 59 (south)
- 120 — 60 (south)

Default: 0

source state plane code [selection] <put parameter description here>

Options:

- 0 — —
- 1 — AK_10
- 2 — AK_2
- 3 — AK_3
- 4 — AK_4
- 5 — AK_5
- 6 — AK_6
- 7 — AK_7
- 8 — AK_8
- 9 — AK_9
- 10 — AL_E
- 11 — AL_W
- 12 — AR_N
- 13 — AR_S
- 14 — AZ_C
- 15 — AZ_E
- 16 — AZ_W
- 17 — CA_I
- 18 — CA_II
- 19 — CA_III
- 20 — CA_IV
- 21 — CA_V
- 22 — CA_VI
- 23 — CA_VII
- 24 — CO_C
- 25 — CO_N
- 26 — CO_S
- 27 — CT
- 28 — DE
- 29 — FL_E
- 30 — FL_N
- 31 — FL_W
- 32 — GA_E
- 33 — GA_W
- 34 — HI_1
- 35 — HI_2
- 36 — HI_3
- 37 — HI_4
- 38 — HI_5

- 39 — IA_N
- 40 — IA_S
- 41 — ID_C
- 42 — ID_E
- 43 — ID_W
- 44 — IL_E
- 45 — IL_W
- 46 — IN_E
- 47 — IN_W
- 48 — KS_N
- 49 — KS_S
- 50 — KY_N
- 51 — KY_S
- 52 — LA_N
- 53 — LA_S
- 54 — MA_I
- 55 — MA_M
- 56 — MD
- 57 — ME_E
- 58 — ME_W
- 59 — MI_C
- 60 — MI_N
- 61 — MI_S
- 62 — MN_C
- 63 — MN_N
- 64 — MN_S
- 65 — MO_C
- 66 — MO_E
- 67 — MO_W
- 68 — MS_E
- 69 — MS_W
- 70 — MT_C
- 71 — MT_N
- 72 — MT_S
- 73 — NC
- 74 — ND_N
- 75 — ND_S
- 76 — NE_N
- 77 — NE_S

- 78 — NH
- 79 — NJ
- 80 — NM_C
- 81 — NM_E
- 82 — NM_W
- 83 — NV_C
- 84 — NV_E
- 85 — NV_W
- 86 — NY_C
- 87 — NY_E
- 88 — NY_LI
- 89 — NY_W
- 90 — OH_N
- 91 — OH_S
- 92 — OK_N
- 93 — OK_S
- 94 — OR_N
- 95 — OR_S
- 96 — PA_N
- 97 — PA_S
- 98 — PR
- 99 — RI
- 100 — SC_N
- 101 — SC_S
- 102 — SD_N
- 103 — SD_S
- 104 — St.Croix
- 105 — TN
- 106 — TX_C
- 107 — TX_N
- 108 — TX_NC
- 109 — TX_S
- 110 — TX_SC
- 111 — UT_C
- 112 — UT_N
- 113 — UT_S
- 114 — VA_N
- 115 — VA_S
- 116 — VT

- 117 — WA_N
- 118 — WA_S
- 119 — WI_C
- 120 — WI_N
- 121 — WI_S
- 122 — WV_N
- 123 — WV_S
- 124 — WY_E
- 125 — WY_EC
- 126 — WY_W
- 127 — WY_WC

Default: 0

target projection [selection] <put parameter description here>

Options:

- 0 — —
- 1 — utm
- 2 — sp83
- 3 — sp27
- 4 — longlat
- 5 — latlong

Default: 0

target utm zone [selection] <put parameter description here>

Options:

- 0 — —
- 1 — 1 (north)
- 2 — 2 (north)
- 3 — 3 (north)
- 4 — 4 (north)
- 5 — 5 (north)
- 6 — 6 (north)
- 7 — 7 (north)
- 8 — 8 (north)
- 9 — 9 (north)
- 10 — 10 (north)
- 11 — 11 (north)
- 12 — 12 (north)
- 13 — 13 (north)
- 14 — 14 (north)
- 15 — 15 (north)

- 16 — 16 (north)
- 17 — 17 (north)
- 18 — 18 (north)
- 19 — 19 (north)
- 20 — 20 (north)
- 21 — 21 (north)
- 22 — 22 (north)
- 23 — 23 (north)
- 24 — 24 (north)
- 25 — 25 (north)
- 26 — 26 (north)
- 27 — 27 (north)
- 28 — 28 (north)
- 29 — 29 (north)
- 30 — 30 (north)
- 31 — 31 (north)
- 32 — 32 (north)
- 33 — 33 (north)
- 34 — 34 (north)
- 35 — 35 (north)
- 36 — 36 (north)
- 37 — 37 (north)
- 38 — 38 (north)
- 39 — 39 (north)
- 40 — 40 (north)
- 41 — 41 (north)
- 42 — 42 (north)
- 43 — 43 (north)
- 44 — 44 (north)
- 45 — 45 (north)
- 46 — 46 (north)
- 47 — 47 (north)
- 48 — 48 (north)
- 49 — 49 (north)
- 50 — 50 (north)
- 51 — 51 (north)
- 52 — 52 (north)
- 53 — 53 (north)
- 54 — 54 (north)

- 55 — 55 (north)
- 56 — 56 (north)
- 57 — 57 (north)
- 58 — 58 (north)
- 59 — 59 (north)
- 60 — 60 (north)
- 61 — 1 (south)
- 62 — 2 (south)
- 63 — 3 (south)
- 64 — 4 (south)
- 65 — 5 (south)
- 66 — 6 (south)
- 67 — 7 (south)
- 68 — 8 (south)
- 69 — 9 (south)
- 70 — 10 (south)
- 71 — 11 (south)
- 72 — 12 (south)
- 73 — 13 (south)
- 74 — 14 (south)
- 75 — 15 (south)
- 76 — 16 (south)
- 77 — 17 (south)
- 78 — 18 (south)
- 79 — 19 (south)
- 80 — 20 (south)
- 81 — 21 (south)
- 82 — 22 (south)
- 83 — 23 (south)
- 84 — 24 (south)
- 85 — 25 (south)
- 86 — 26 (south)
- 87 — 27 (south)
- 88 — 28 (south)
- 89 — 29 (south)
- 90 — 30 (south)
- 91 — 31 (south)
- 92 — 32 (south)
- 93 — 33 (south)

- 94 — 34 (south)
- 95 — 35 (south)
- 96 — 36 (south)
- 97 — 37 (south)
- 98 — 38 (south)
- 99 — 39 (south)
- 100 — 40 (south)
- 101 — 41 (south)
- 102 — 42 (south)
- 103 — 43 (south)
- 104 — 44 (south)
- 105 — 45 (south)
- 106 — 46 (south)
- 107 — 47 (south)
- 108 — 48 (south)
- 109 — 49 (south)
- 110 — 50 (south)
- 111 — 51 (south)
- 112 — 52 (south)
- 113 — 53 (south)
- 114 — 54 (south)
- 115 — 55 (south)
- 116 — 56 (south)
- 117 — 57 (south)
- 118 — 58 (south)
- 119 — 59 (south)
- 120 — 60 (south)

Default: 0

target state plane code [selection] <put parameter description here>

Options:

- 0 — —
- 1 — AK_10
- 2 — AK_2
- 3 — AK_3
- 4 — AK_4
- 5 — AK_5
- 6 — AK_6
- 7 — AK_7
- 8 — AK_8

- 9 — AK_9
- 10 — AL_E
- 11 — AL_W
- 12 — AR_N
- 13 — AR_S
- 14 — AZ_C
- 15 — AZ_E
- 16 — AZ_W
- 17 — CA_I
- 18 — CA_II
- 19 — CA_III
- 20 — CA_IV
- 21 — CA_V
- 22 — CA_VI
- 23 — CA_VII
- 24 — CO_C
- 25 — CO_N
- 26 — CO_S
- 27 — CT
- 28 — DE
- 29 — FL_E
- 30 — FL_N
- 31 — FL_W
- 32 — GA_E
- 33 — GA_W
- 34 — HI_1
- 35 — HI_2
- 36 — HI_3
- 37 — HI_4
- 38 — HI_5
- 39 — IA_N
- 40 — IA_S
- 41 — ID_C
- 42 — ID_E
- 43 — ID_W
- 44 — IL_E
- 45 — IL_W
- 46 — IN_E
- 47 — IN_W

- 48 — KS_N
- 49 — KS_S
- 50 — KY_N
- 51 — KY_S
- 52 — LA_N
- 53 — LA_S
- 54 — MA_I
- 55 — MA_M
- 56 — MD
- 57 — ME_E
- 58 — ME_W
- 59 — MI_C
- 60 — MI_N
- 61 — MI_S
- 62 — MN_C
- 63 — MN_N
- 64 — MN_S
- 65 — MO_C
- 66 — MO_E
- 67 — MO_W
- 68 — MS_E
- 69 — MS_W
- 70 — MT_C
- 71 — MT_N
- 72 — MT_S
- 73 — NC
- 74 — ND_N
- 75 — ND_S
- 76 — NE_N
- 77 — NE_S
- 78 — NH
- 79 — NJ
- 80 — NM_C
- 81 — NM_E
- 82 — NM_W
- 83 — NV_C
- 84 — NV_E
- 85 — NV_W
- 86 — NY_C

- 87 — NY_E
- 88 — NY_LI
- 89 — NY_W
- 90 — OH_N
- 91 — OH_S
- 92 — OK_N
- 93 — OK_S
- 94 — OR_N
- 95 — OR_S
- 96 — PA_N
- 97 — PA_S
- 98 — PR
- 99 — RI
- 100 — SC_N
- 101 — SC_S
- 102 — SD_N
- 103 — SD_S
- 104 — St.Croix
- 105 — TN
- 106 — TX_C
- 107 — TX_N
- 108 — TX_NC
- 109 — TX_S
- 110 — TX_SC
- 111 — UT_C
- 112 — UT_N
- 113 — UT_S
- 114 — VA_N
- 115 — VA_S
- 116 — VT
- 117 — WA_N
- 118 — WA_S
- 119 — WI_C
- 120 — WI_N
- 121 — WI_S
- 122 — WV_N
- 123 — WV_S
- 124 — WY_E
- 125 — WY_EC

- 126 — WY_W
- 127 — WY_WC

Default: 0

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:las2lasproject', verbose, input_laslaz, source_projection, source_u
```

See also

18.2.3 las2las_transform

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

transform (coordinates) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — translate_x
- 2 — translate_y
- 3 — translate_z
- 4 — scale_x
- 5 — scale_y
- 6 — scale_z
- 7 — clamp_z_above
- 8 — clamp_z_below

Default: 0

value for transform (coordinates) [string] <put parameter description here>

Default: *(not set)*

second transform (coordinates) [selection] <put parameter description here>

Options:

- 0 — —

- 1 — translate_x
- 2 — translate_y
- 3 — translate_z
- 4 — scale_x
- 5 — scale_y
- 6 — scale_z
- 7 — clamp_z_above
- 8 — clamp_z_below

Default: 0

value for second transform (coordinates) [string] <put parameter description here>

Default: *(not set)*

transform (intensities, scan angles, GPS times, ...) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — scale_intensity
- 2 — translate_intensity
- 3 — clamp_intensity_above
- 4 — clamp_intensity_below
- 5 — scale_scan_angle
- 6 — translate_scan_angle
- 7 — translate_gps_time
- 8 — set_classification
- 9 — set_user_data
- 10 — set_point_source
- 11 — scale_rgb_up
- 12 — scale_rgb_down
- 13 — repair_zero_returns

Default: 0

value for transform (intensities, scan angles, GPS times, ...) [string] <put parameter description here>

Default: *(not set)*

second transform (intensities, scan angles, GPS times, ...) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — scale_intensity
- 2 — translate_intensity
- 3 — clamp_intensity_above
- 4 — clamp_intensity_below

- 5 — scale_scan_angle
- 6 — translate_scan_angle
- 7 — translate_gps_time
- 8 — set_classification
- 9 — set_user_data
- 10 — set_point_source
- 11 — scale_rgb_up
- 12 — scale_rgb_down
- 13 — repair_zero_returns

Default: 0

value for second transform (intensities, scan angles, GPS times, ...) [string]
 <put parameter description here>

Default: *(not set)*

operations (first 7 need an argument) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — set_point_type
- 2 — set_point_size
- 3 — set_version_minor
- 4 — set_version_major
- 5 — start_at_point
- 6 — stop_at_point
- 7 — remove_vlr
- 8 — auto_reoffset
- 9 — week_to_adjusted
- 10 — adjusted_to_week
- 11 — scale_rgb_up
- 12 — scale_rgb_down
- 13 — remove_all_vlrs
- 14 — remove_extra
- 15 — clip_to_bounding_box

Default: 0

argument for operation [string] <put parameter description here>

Default: *(not set)*

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:las2lastransform', verbose, input_laslaz, transform_coordinate1, tr
```

See also

18.2.4 las2txt

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

parse_string [string] <put parameter description here>

Default: *xyz*

Outputs

Output ASCII file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:las2txt', verbose, input_laslaz, parse_string, output)
```

See also

18.2.5 lasindex

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

is mobile or terrestrial LiDAR (not airborne) [boolean] <put parameter description here>

Default: *False*

Outputs

Console usage

```
processing.runalg('lidartools:lasindex', verbose, input_laslaz, mobile_or_terrestrial)
```

See also

18.2.6 lasinfo

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

Outputs

Output ASCII file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:lasinfo', verbose, input_laslaz, output)
```

See also

18.2.7 lasmerge

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

files are flightlines [boolean] <put parameter description here>

Default: *True*

input LAS/LAZ file [file] Optional.

<put parameter description here>

2nd file [file] Optional.

<put parameter description here>

3rd file [file] Optional.

<put parameter description here>

4th file [file] Optional.

<put parameter description here>

5th file [file] Optional.

<put parameter description here>

6th file [file] Optional.

<put parameter description here>

7th file [file] Optional.

<put parameter description here>

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:lasmerge', verbose, files_are_flightlines, input_laslaz, file2, file3)
```

See also

18.2.8 lasprecision

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

Outputs

Output ASCII file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:lasprecision', verbose, input_laslaz, output)
```


See also

18.2.9 lasquery

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

area of interest [**extent**] <put parameter description here>

Default: *0,1,0,1*

Outputs

Console usage

```
processing.runalg('lidartools:lasquery', verbose, aoi)
```

See also

18.2.10 lasvalidate

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

input LAS/LAZ file [**file**] Optional.

<put parameter description here>

Outputs

Output XML file [**file**] <put output description here>

Console usage

```
processing.runalg('lidartools:lasvalidate', verbose, input_laslaz, output)
```

See also

18.2.11 laszip

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

input LAS/LAZ file [**file**] Optional.

<put parameter description here>

only report size [**boolean**] <put parameter description here>

Default: *False*

Outputs

output LAS/LAZ file [**file**] <put output description here>

Console usage

```
processing.runalg('lidartools:laszip', verbose, input_laslaz, report_size, output_laslaz)
```

See also

18.2.12 txt2las

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

Input ASCII file [**file**] Optional.

<put parameter description here>

parse lines as [**string**] <put parameter description here>

Default: *xyz*

skip the first n lines [**number**] <put parameter description here>

Default: *0*

resolution of x and y coordinate [**number**] <put parameter description here>

Default: *0.01*

resolution of z coordinate [number] <put parameter description here>

Default: *0.01*

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:txt2las', verbose, input, parse_string, skip, scale_factor_xy, scale_factor_z)
```

See also

.

18.3 Modeler Tools

18.3.1 Calculator

Description

<put algorithm description here>

Parameters

Formula [string] <put parameter description here>

Default: *(not set)*

dummy [number] <put parameter description here>

Default: *0.0*

dummy [number] <put parameter description here>

Default: *0.0*

dummy [number] <put parameter description here>

Default: *0.0*

dummy [number] <put parameter description here>

Default: *0.0*

dummy [number] <put parameter description here>

Default: *0.0*

dummy [number] <put parameter description here>

Default: *0.0*

dummy [number] <put parameter description here>

Default: *0.0*

dummy [number] <put parameter description here>

Default: *0.0*

dummy [number] <put parameter description here>

Default: 0.0

dummy [number] <put parameter description here>

Default: 0.0

Outputs

Result [number] <put output description here>

Console usage

```
processing.runalg('modelertools:calculator', formula, number0, number1, number2, number3, number4,
```

See also

18.3.2 Raster layer bounds

Description

<put algorithm description here>

Parameters

Layer [raster] <put parameter description here>

Outputs

min X [number] <put output description here>

max X [number] <put output description here>

min Y [number] <put output description here>

max Y [number] <put output description here>

Extent [extent] <put output description here>

Console usage

```
processing.runalg('modelertools:rasterlayerbounds', layer)
```

See also

18.3.3 Vector layer bounds

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Outputs

min X [number] <put output description here>

max X [number] <put output description here>

min Y [number] <put output description here>

max Y [number] <put output description here>

Extent [extent] <put output description here>

Console usage

```
processing.runalg('modelertools:vectorlayerbounds', layer)
```

See also

.

18.4 OrfeoToolbox algorithm provider

Orfeo ToolBox (OTB) is an open source library of image processing algorithms. OTB is based on the medical image processing library ITK and offers particular functionalities for remote sensing image processing in general and for high spatial resolution images in particular. Targeted algorithms for high resolution optical images (Pleiades, SPOT, QuickBird, WorldView, Landsat, Ikonos), hyperspectral sensors (Hyperion) or SAR (TerraSarX, ERS, Palsar) are available.

Note: Please remember that Processing contains only the interface description, so you need to install OTB by yourself and configure Processing properly.

.

18.4.1 Calibration

Optical calibration

Description

<put algorithm description here>

Parameters

Input [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Calibration Level [selection] <put parameter description here>

Options:

- 0 — toa

Default: *0*

Convert to milli reflectance [boolean] <put parameter description here>

Default: *True*

Clamp of reflectivity values between [0, 100] [boolean] <put parameter description here>

Default: *True*

Relative Spectral Response File [file] Optional.

<put parameter description here>

Outputs

Output [raster] <put output description here>

Console usage

```
processing.runalg('otb:opticalcalibration', -in, -ram, -level, -milli, -clamp, -rsr, -out)
```

See also

.

18.4.2 Feature extrcation

BinaryMorphologicalOperation (closing)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — closing

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

`processing.runalg('otb:binarymorphologicaloperationclosing', -in, -channel, -ram, -structype, -st`

See also

BinaryMorphologicalOperation (dilate)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — dilate

Default: *0*

Foreground Value [number] <put parameter description here>

Default: *1*

Background Value [number] <put parameter description here>

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:binarymorphologicaloperationdilate', -in, -channel, -ram, -structype, -stru
```

See also

BinaryMorphologicalOperation (erode)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — erode

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:binarymorphologicaloperationerode', -in, -channel, -ram, -structype, -stru
```

See also

BinaryMorphologicalOperation (opening)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — opening

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:binarymorphologicaloperationopening', -in, -channel, -ram, -structype, -st
```

See also

EdgeExtraction (gradient)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Edge feature [selection] <put parameter description here>

Options:

- 0 — gradient

Default: 0

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:edgeextractiongradient', -in, -channel, -ram, -filter, -out)
```

See also

EdgeExtraction (sobel)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: 1

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Edge feature [selection] <put parameter description here>

Options:

- 0 — sobel

Default: 0

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:edgeextractionsobel', -in, -channel, -ram, -filter, -out)
```

See also

EdgeExtraction (touzi)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Edge feature [selection] <put parameter description here>

Options:

- 0 — touzi

Default: *0*

The Radius [number] <put parameter description here>

Default: *1*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:edgeextractiontouzi', -in, -channel, -ram, -filter, -filter.touzi.xradius,
```

See also

GrayScaleMorphologicalOperation (closing)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — closing

Default: 0

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationclosing', -in, -channel, -ram, -structype, ...)
```

See also

GrayScaleMorphologicalOperation (dilate)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: 1

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: 0

The Structuring Element Radius [number] <put parameter description here>

Default: 5

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — dilate

Default: 0

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationdilate', -in, -channel, -ram, -structype, -s
```

See also

GrayScaleMorphologicalOperation (erode)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — erode

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationerode', -in, -channel, -ram, -structype, -s
```

See also

GrayScaleMorphologicalOperation (opening)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — opening

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationopening', -in, -channel, -ram, -structype, ...
```

See also

Haralick Texture Extraction

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

X Radius [number] <put parameter description here>

Default: *2*

Y Radius [number] <put parameter description here>

Default: 2

X Offset [number] <put parameter description here>

Default: 1

Y Offset [number] <put parameter description here>

Default: 1

Image Minimum [number] <put parameter description here>

Default: 0

Image Maximum [number] <put parameter description here>

Default: 255

Histogram number of bin [number] <put parameter description here>

Default: 8

Texture Set Selection [selection] <put parameter description here>

Options:

- 0 — simple
- 1 — advanced
- 2 — higher

Default: 0

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:haralicktextureextraction', -in, -channel, -ram, -parameters.xrad, -parameter
```

See also

Line segment detection

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

No rescaling in [0, 255] [boolean] <put parameter description here>

Default: *True*

Outputs

Output **Detected lines** [**vector**] <put output description here>

Console usage

```
processing.runalg('otb:linesegmentdetection', -in, -norescale, -out)
```

See also

Local Statistic Extraction

Description

<put algorithm description here>

Parameters

Input Image [**raster**] <put parameter description here>

Selected Channel [**number**] <put parameter description here>

Default: *1*

Available RAM (Mb) [**number**] <put parameter description here>

Default: *128*

Neighborhood radius [**number**] <put parameter description here>

Default: *3*

Outputs

Feature Output Image [**raster**] <put output description here>

Console usage

```
processing.runalg('otb:localstatisticextraction', -in, -channel, -ram, -radius, -out)
```

See also

Multivariate alteration detector

Description

<put algorithm description here>

Parameters

Input Image 1 [raster] <put parameter description here>

Input Image 2 [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Change Map [raster] <put output description here>

Console usage

```
processing.runalg('otb:multivariatealterationdetector', -in1, -in2, -ram, -out)
```

See also

Radiometric Indices

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Blue Channel [number] <put parameter description here>

Default: *1*

Green Channel [number] <put parameter description here>

Default: *1*

Red Channel [number] <put parameter description here>

Default: *1*

NIR Channel [number] <put parameter description here>

Default: *1*

Mir Channel [number] <put parameter description here>

Default: *1*

Available Radiometric Indices [selection] <put parameter description here>

Options:

- 0 — ndvi
- 1 — tndvi
- 2 — rvi

- 3 — savi
- 4 — tsavi
- 5 — msavi
- 6 — msavi2
- 7 — gemi
- 8 — ipvi
- 9 — ndwi
- 10 — ndwi2
- 11 — mndwi
- 12 — ndpi
- 13 — ndti
- 14 — ri
- 15 — ci
- 16 — bi
- 17 — bi2

Default: 0

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:radiometricindices', -in, -ram, -channels.blue, -channels.green, -channels
```

See also

.

18.4.3 Geometry

Image Envelope

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Sampling Rate [number] <put parameter description here>

Default: 0

Projection [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output Vector Data [vector] <put output description here>

Console usage

```
processing.runalg('otb:imageenvelope', -in, -sr, -proj, -out)
```

See also

OrthoRectification (epsg)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Output Cartographic Map Projection [selection] <put parameter description here>

Options:

- 0 — epsg

Default: *0*

EPSG Code [number] <put parameter description here>

Default: *4326*

Parameters estimation modes [selection] <put parameter description here>

Options:

- 0 — autosize
- 1 — autospacing

Default: *0*

Default pixel value [number] <put parameter description here>

Default: *0*

Default elevation [number] <put parameter description here>

Default: *0*

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Resampling grid spacing [number] <put parameter description here>

Default: 4

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:orthorectificationeps', -io.in, -map, -map.epsg.code, -outputs.mode, -outp
```

See also

OrthoRectification (fit-to-ortho)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Parameters estimation modes [selection] <put parameter description here>

Options:

- 0 — orthofit

Default: 0

Model ortho-image [raster] Optional.

<put parameter description here>

Default pixel value [number] <put parameter description here>

Default: 0

Default elevation [number] <put parameter description here>

Default: 0

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Resampling grid spacing [number] <put parameter description here>

Default: 4

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:orthorectificationfittoortho', -io.in, -outputs.mode, -outputs.ortho, -outp
```

See also

OrthoRectification (lambert-WGS84)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Output Cartographic Map Projection [selection] <put parameter description here>

Options:

- 0 — lambert2
- 1 — lambert93
- 2 — wgs

Default: 0

Parameters estimation modes [selection] <put parameter description here>

Options:

- 0 — autosize
- 1 — autospacing

Default: 0

Default pixel value [number] <put parameter description here>

Default: 0

Default elevation [number] <put parameter description here>

Default: 0

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Resampling grid spacing [number] <put parameter description here>

Default: 4

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:orthorectificationlambertwgs84', -io.in, -map, -outputs.mode, -outputs.def
```

See also

OrthoRectification (utm)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Output Cartographic Map Projection [selection] <put parameter description here>

Options:

- 0 — utm

Default: 0

Zone number [number] <put parameter description here>

Default: 31

Northern Hemisphere [boolean] <put parameter description here>

Default: *True*

Parameters estimation modes [selection] <put parameter description here>

Options:

- 0 — autosize
- 1 — autospacing

Default: 0

Default pixel value [number] <put parameter description here>

Default: 0

Default elevation [number] <put parameter description here>

Default: 0

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Resampling grid spacing [number] <put parameter description here>

Default: 4

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:orthorectificationutm', -io.in, -map, -map.utm.zone, -map.utm.northhem, -o
```

See also

Pansharpening (bayes)

Description

<put algorithm description here>

Parameters

Input PAN Image [raster] <put parameter description here>

Input XS Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — bayes

Default: 0

Weight [number] <put parameter description here>

Default: 0.9999

S coefficient [number] <put parameter description here>

Default: 1

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:pansharpeningbayes', -inp, -inxs, -method, -method.bayes.lambda, -method.b
```

See also

Pansharpening (lmvm)

Description

<put algorithm description here>

Parameters

Input PAN Image [raster] <put parameter description here>

Input XS Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — lmvm

Default: 0

X radius [number] <put parameter description here>

Default: 3

Y radius [number] <put parameter description here>

Default: 3

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:pansharpeninglmvm', -inp, -inxs, -method, -method.lmvm.radiusx, -method.lm
```


See also

Pansharpening (rcs)

Description

<put algorithm description here>

Parameters

Input PAN Image [raster] <put parameter description here>

Input XS Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — rcs

Default: 0

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:pansharpeningrcs', -inp, -inxs, -method, -ram, -out)
```

See also

RigidTransformResample (id)

Description

<put algorithm description here>

Parameters

Input image [raster] <put parameter description here>

Type of transformation [selection] <put parameter description here>

Options:

- 0 — id

Default: 0

X scaling [number] <put parameter description here>

Default: 1

Y scaling [number] <put parameter description here>

Default: 1

Interpolation [selection] <put parameter description here>

Options:

- 0 — nn
- 1 — linear
- 2 — bco

Default: 2

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:rigidtransformresampleid', -in, -transform.type, -transform.type.id.scalex
```

See also

RigidTransformResample (rotation)

Description

<put algorithm description here>

Parameters

Input image [raster] <put parameter description here>

Type of transformation [selection] <put parameter description here>

Options:

- 0 — rotation

Default: 0

Rotation angle [number] <put parameter description here>

Default: 0

X scaling [number] <put parameter description here>

Default: 1

Y scaling [number] <put parameter description here>

Default: 1

Interpolation [selection] <put parameter description here>

Options:

- 0 — nn
- 1 — linear
- 2 — bco

Default: 2

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:rigidtransformresamplerotation', -in, -transform.type, -transform.type.rot
```

See also

RigidTransformResample (translation)

Description

<put algorithm description here>

Parameters

Input image [raster] <put parameter description here>

Type of transformation [selection] <put parameter description here>

Options:

- 0 — translation

Default: 0

The X translation (in physical units) [number] <put parameter description here>

Default: 0

The Y translation (in physical units) [number] <put parameter description here>

Default: 0

X scaling [number] <put parameter description here>

Default: 1

Y scaling [number] <put parameter description here>

Default: 1

Interpolation [selection] <put parameter description here>

Options:

- 0 — nn
- 1 — linear
- 2 — bco

Default: 2

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:rigidtransformresampletranslation', -in, -transform.type, -transform.type.
```

See also

Superimpose sensor

Description

<put algorithm description here>

Parameters

Reference input [raster] <put parameter description here>

The image to reproject [raster] <put parameter description here>

Default elevation [number] <put parameter description here>

Default: 0

Spacing of the deformation field [number] <put parameter description here>

Default: 4

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:superimposesensor', -inr, -inm, -elev.default, -lms, -interpolator, -inter
```

See also

.

18.4.4 Image filtering

DimensionalityReduction (ica)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — ica

Default: *0*

number of iterations [number] <put parameter description here>

Default: *20*

Give the increment weight of W in [0, 1] [number] <put parameter description here>

Default: *1*

Number of Components [number] <put parameter description here>

Default: *0*

Normalize [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

“Inverse Output Image“ [raster] <put output description here>

Transformation matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:dimensionalityreductionica', -in, -method, -method.ica.iter, -method.ica.m
```

See also

DimensionalityReduction (maf)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — maf

Default: 0

Number of Components. [number] <put parameter description here>

Default: 0

Normalize. [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

Transformation matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:dimensionalityreductionmaf', -in, -method, -nbcomp, -normalize, -out, -out
```

See also

DimensionalityReduction (napca)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — napca

Default: 0

Set the x radius of the sliding window. [number] <put parameter description here>

Default: 1

Set the y radius of the sliding window. [number] <put parameter description here>

Default: 1

Number of Components. [number] <put parameter description here>

Default: 0

Normalize. [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

“**Inverse Output Image**“ [raster] <put output description here>

Transformation matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:dimensionalityreductionnapca', -in, -method, -method.napca.radiusx, -method
```

See also

DimensionalityReduction (pca)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — pca

Default: 0

Number of Components. [number] <put parameter description here>

Default: 0

Normalize. [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

“ **Inverse Output Image**“ [raster] <put output description here>

Transformation matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:dimensionalityreductionpca', -in, -method, -nbcomp, -normalize, -out, -out.
```

See also

Mean Shift filtering (can be used as Exact Large-Scale Mean-Shift segmentation, step 1)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Spatial radius [number] <put parameter description here>

Default: *5*

Range radius [number] <put parameter description here>

Default: *15*

Mode convergence threshold [number] <put parameter description here>

Default: *0.1*

Maximum number of iterations [number] <put parameter description here>

Default: *100*

Range radius coefficient [number] <put parameter description here>

Default: *0*

Mode search. [boolean] <put parameter description here>

Default: *True*

Outputs

Filtered output [raster] <put output description here>

Spatial image [raster] <put output description here>

Console usage

```
processing.runalg('otb:meanshiftfilteringcanbeusedasexactlargescalemeanshiftsegmentationstep1', -
```

See also

Smoothing (anidif)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Smoothing Type [selection] <put parameter description here>

Options:

- 0 — anidif

Default: *2*

Time Step [number] <put parameter description here>

Default: *0.125*

Nb Iterations [number] <put parameter description here>

Default: *10*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:smoothinganidif', -in, -ram, -type, -type.anidif.timestep, -type.anidif.nb
```

See also

Smoothing (gaussian)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Smoothing Type [selection] <put parameter description here>

Options:

- 0 — gaussian

Default: 2

Radius [number] <put parameter description here>

Default: 2

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:smoothinggaussian', -in, -ram, -type, -type.gaussian.radius, -out)
```

See also

Smoothing (mean)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Smoothing Type [selection] <put parameter description here>

Options:

- 0 — mean

Default: 2

Radius [number] <put parameter description here>

Default: 2

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:smoothingmean', -in, -ram, -type, -type.mean.radius, -out)
```

See also

.

18.4.5 Image manipulation

ColorMapping (continuous)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Operation [selection] <put parameter description here>

Options:

- 0 — labeltocolour

Default: *0*

Color mapping method [selection] <put parameter description here>

Options:

- 0 — continuous

Default: *0*

Look-up tables [selection] <put parameter description here>

Options:

- 0 — red
- 1 — green
- 2 — blue
- 3 — grey
- 4 — hot
- 5 — cool
- 6 — spring
- 7 — summer
- 8 — autumn
- 9 — winter
- 10 — copper

- 11 — jet
- 12 — hsv
- 13 — overunder
- 14 — relief

Default: 0

Mapping range lower value [number] <put parameter description here>

Default: 0

Mapping range higher value [number] <put parameter description here>

Default: 255

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:colormappingcontinuous', -in, -ram, -op, -method, -method.continuous.lut, ...
```

See also

ColorMapping (custom)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Operation [selection] <put parameter description here>

Options:

- 0 — labeltcolor

Default: 0

Color mapping method [selection] <put parameter description here>

Options:

- 0 — custom

Default: 0

Look-up table file [file] <put parameter description here>

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:colormappingcustom', -in, -ram, -op, -method, -method.custom.lut, -out)
```

See also

ColorMapping (image)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Operation [selection] <put parameter description here>

Options:

- 0 — labeltocolor

Default: *0*

Color mapping method [selection] <put parameter description here>

Options:

- 0 — image

Default: *0*

Support Image [raster] <put parameter description here>

NoData value [number] <put parameter description here>

Default: *0*

lower quantile [number] <put parameter description here>

Default: *2*

upper quantile [number] <put parameter description here>

Default: *2*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:colormappingimage', -in, -ram, -op, -method, -method.image.in, -method.ima
```

See also

ColorMapping (optimal)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Operation [selection] <put parameter description here>

Options:

- 0 — labeltocolor

Default: *0*

Color mapping method [selection] <put parameter description here>

Options:

- 0 — optimal

Default: *0*

Background label [number] <put parameter description here>

Default: *0*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:colormappingoptimal', -in, -ram, -op, -method, -method.optimal.background,
```

See also

ExtractROI (fit)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Extraction mode [selection] <put parameter description here>

Options:

- 0 — fit

Default: 0

Reference image [raster] <put parameter description here>

Default elevation [number] <put parameter description here>

Default: 0

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:extractroi', -in, -ram, -mode, -mode.fit.ref, -mode.fit.elev.default, -
```

See also

ExtractROI (standard)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Extraction mode [selection] <put parameter description here>

Options:

- 0 — standard

Default: 0

Start X [number] <put parameter description here>

Default: 0

Start Y [number] <put parameter description here>

Default: 0

Size X [number] <put parameter description here>

Default: 0

Size Y [number] <put parameter description here>

Default: 0

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:extractroistandard', -in, -ram, -mode, -startx, -starty, -sizex, -sizey, -o
```

See also

Images Concatenation

Description

<put algorithm description here>

Parameters

Input images list [multipleinput: rasters] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:imagesconcatenation', -il, -ram, -out)
```

See also

Image Tile Fusion

Description

<put algorithm description here>

Parameters

Input Tile Images [**multipleinput: rasters**] <put parameter description here>

Number of tile columns [**number**] <put parameter description here>

Default: *0*

Number of tile rows [**number**] <put parameter description here>

Default: *0*

Outputs

Output Image [**raster**] <put output description here>

Console usage

```
processing.runalg('otb:imagetilefusion', -il, -cols, -rows, -out)
```

See also

Read image information

Description

<put algorithm description here>

Parameters

Input Image [**raster**] <put parameter description here>

Display the OSSIM keywordlist [**boolean**] <put parameter description here>

Default: *True*

GCPs Id [**string**] <put parameter description here>

Default: *None*

GCPs Info [**string**] <put parameter description here>

Default: *None*

GCPs Image Coordinates [**string**] <put parameter description here>

Default: *None*

GCPs Geographic Coordinates [**string**] <put parameter description here>

Default: *None*

Outputs

Console usage

```
processing.runalg('otb:readimageinformation', -in, -keywordlist, -gcp.ids, -gcp.info, -gcp.imcoord
```

See also

Rescale Image

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Output min value [number] <put parameter description here>

Default: 0

Output max value [number] <put parameter description here>

Default: 255

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:rescaleimage', -in, -ram, -outmin, -outmax, -out)
```

See also

Split Image

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output Image [file] <put output description here>

Console usage

```
processing.runalg('otb:splitimage', -in, -ram, -out)
```

See also

.

18.4.6 Learning

Classification Map Regularization

Description

<put algorithm description here>

Parameters

Input classification image [raster] <put parameter description here>

Structuring element radius (in pixels) [number] <put parameter description here>

Default: *1*

Multiple majority: Undecided(X)/Original [boolean] <put parameter description here>

Default: *True*

Label for the NoData class [number] <put parameter description here>

Default: *0*

Label for the Undecided class [number] <put parameter description here>

Default: *0*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Output regularized image [raster] <put output description here>

Console usage

```
processing.runalg('otb:classificationmapregularization', -io.in, -ip.radius, -ip.suvbool, -ip.nod
```

See also

ComputeConfusionMatrix (raster)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Ground truth [selection] <put parameter description here>

Options:

- 0 — raster

Default: 0

Input reference image [raster] <put parameter description here>

Value for nodata pixels [number] <put parameter description here>

Default: 0

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:computeconfusionmatrixraster', -in, -ref, -ref.raster.in, -nodatalabel, -r
```

See also

ComputeConfusionMatrix (vector)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Ground truth [selection] <put parameter description here>

Options:

- 0 — vector

Default: 0

Input reference vector data [file] <put parameter description here>

Field name [string] Optional.

<put parameter description here>

Default: *Class*

Value for nodata pixels [number] <put parameter description here>

Default: 0

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:computeconfusionmatrixvector', -in, -ref, -ref.vector.in, -ref.vector.field)
```

See also

Compute Images second order statistics

Description

<put algorithm description here>

Parameters

Input images [multipleinput: rasters] <put parameter description here>

Background Value [number] <put parameter description here>

Default: *0.0*

Outputs

Output XML file [file] <put output description here>

Console usage

```
processing.runalg('otb:computeimagessecondorderstatistics', -il, -bv, -out)
```

See also

FusionOfClassifications (dempstershafer)

Description

<put algorithm description here>

Parameters

Input classifications [multipleinput: rasters] <put parameter description here>

Fusion method [selection] <put parameter description here>

Options:

- 0 — dempstershafer

Default: 0

Confusion Matrices [**multipleinput: files**] <put parameter description here>

Mass of belief measurement [**selection**] <put parameter description here>

Options:

- 0 — precision
- 1 — recall
- 2 — accuracy
- 3 — kappa

Default: 0

Label for the NoData class [**number**] <put parameter description here>

Default: 0

Label for the Undecided class [**number**] <put parameter description here>

Default: 0

Outputs

The output classification image [**raster**] <put output description here>

Console usage

```
processing.runalg('otb:fusionofclassificationsdempstershafer', -il, -method, -method.dempstershafer)
```

See also

FusionOfClassifications (majorityvoting)

Description

<put algorithm description here>

Parameters

Input classifications [**multipleinput: rasters**] <put parameter description here>

Fusion method [**selection**] <put parameter description here>

Options:

- 0 — majorityvoting

Default: 0

Label for the NoData class [**number**] <put parameter description here>

Default: 0

Label for the Undecided class [**number**] <put parameter description here>

Default: 0

Outputs

The output classification image [raster] <put output description here>

Console usage

```
processing.runalg('otb:fusionofclassificationsmajorityvoting', -il, -method, -nodatalabel, -undec
```

See also

Image Classification

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Input Mask [raster] Optional.

<put parameter description here>

Model file [file] <put parameter description here>

Statistics file [file] Optional.

<put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:imageclassification', -in, -mask, -model, -imstat, -ram, -out)
```

See also

SOM Classification

Description

<put algorithm description here>

Parameters

InputImage [raster] <put parameter description here>

ValidityMask [raster] Optional.

<put parameter description here>

TrainingProbability [number] <put parameter description here>

Default: 1

TrainingSetSize [number] <put parameter description here>

Default: 0

StreamingLines [number] <put parameter description here>

Default: 0

SizeX [number] <put parameter description here>

Default: 32

SizeY [number] <put parameter description here>

Default: 32

NeighborhoodX [number] <put parameter description here>

Default: 10

NeighborhoodY [number] <put parameter description here>

Default: 10

NumberIteration [number] <put parameter description here>

Default: 5

BetaInit [number] <put parameter description here>

Default: 1

BetaFinal [number] <put parameter description here>

Default: 0.1

InitialValue [number] <put parameter description here>

Default: 0

Available RAM (Mb) [number] <put parameter description here>

Default: 128

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

OutputImage [raster] <put output description here>

SOM Map [raster] <put output description here>

Console usage

```
processing.runalg('otb:somclassification', -in, -vm, -tp, -ts, -sl, -sx, -sy, -nx, -ny, -ni, -bi,
```


See also

TrainImagesClassifier (ann)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: 0

Maximum training sample size per class [**number**] <put parameter description here>

Default: 1000

Maximum validation sample size per class [**number**] <put parameter description here>

Default: 1000

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: True

Training and validation sample ratio [**number**] <put parameter description here>

Default: 0.5

Name of the discrimination field [**string**] <put parameter description here>

Default: Class

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — ann

Default: 0

Train Method Type [**selection**] <put parameter description here>

Options:

- 0 — reg
- 1 — back

Default: 0

Number of neurons in each intermediate layer [**string**] <put parameter description here>

Default: None

Neuron activation function type [**selection**] <put parameter description here>

Options:

- 0 — ident
- 1 — sig

- 2 — gau

Default: 1

Alpha parameter of the activation function [number] <put parameter description here>

Default: 1

Beta parameter of the activation function [number] <put parameter description here>

Default: 1

Strength of the weight gradient term in the BACKPROP method [number] <put parameter description here>

Default: 0.1

Strength of the momentum term (the difference between weights on the 2 previous iterations) [number] <put parameter description here>

Default: 0.1

Initial value Delta_0 of update-values Delta_{ij} in RPROP method [number] <put parameter description here>

Default: 0.1

Update-values lower limit Delta_{min} in RPROP method [number] <put parameter description here>

Default: 1e-07

Termination criteria [selection] <put parameter description here>

Options:

- 0 — iter
- 1 — eps
- 2 — all

Default: 2

Epsilon value used in the Termination criteria [number] <put parameter description here>

Default: 0.01

Maximum number of iterations used in the Termination criteria [number] <put parameter description here>

Default: 1000

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierann', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

See also

TrainImagesClassifier (bayes)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: 0

Maximum training sample size per class [**number**] <put parameter description here>

Default: 1000

Maximum validation sample size per class [**number**] <put parameter description here>

Default: 1000

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: True

Training and validation sample ratio [**number**] <put parameter description here>

Default: 0.5

Name of the discrimination field [**string**] <put parameter description here>

Default: Class

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — bayes

Default: 0

set user defined seed [**number**] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [**file**] <put output description here>

Output model [**file**] <put output description here>

Console usage

`processing.runalg('otb:trainimagesclassifierbayes', -io.il, -io.vd, -io.imstat, -elev.default, -s`

See also

TrainImagesClassifier (boost)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: 0

Maximum training sample size per class [**number**] <put parameter description here>

Default: 1000

Maximum validation sample size per class [**number**] <put parameter description here>

Default: 1000

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: True

Training and validation sample ratio [**number**] <put parameter description here>

Default: 0.5

Name of the discrimination field [**string**] <put parameter description here>

Default: Class

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — boost

Default: 0

Boost Type [**selection**] <put parameter description here>

Options:

- 0 — discrete
- 1 — real
- 2 — logit
- 3 — gentle

Default: 1

Weak count [**number**] <put parameter description here>

Default: 100

Weight Trim Rate [**number**] <put parameter description here>

Default: 0.95

Maximum depth of the tree [number] <put parameter description here>

Default: *1*

set user defined seed [number] <put parameter description here>

Default: *0*

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierboost', -io.il, -io.vd, -io.imstat, -elev.default, -s
```

See also

TrainImagesClassifier (dt)

Description

<put algorithm description here>

Parameters

Input Image List [multipleinput: rasters] <put parameter description here>

Input Vector Data List [multipleinput: any vectors] <put parameter description here>

Input XML image statistics file [file] Optional.

<put parameter description here>

Default elevation [number] <put parameter description here>

Default: *0*

Maximum training sample size per class [number] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [number] <put parameter description here>

Default: *1000*

On edge pixel inclusion [boolean] <put parameter description here>

Default: *True*

Training and validation sample ratio [number] <put parameter description here>

Default: *0.5*

Name of the discrimination field [string] <put parameter description here>

Default: *Class*

Classifier to use for the training [selection] <put parameter description here>

Options:

- 0 — dt

Default: 0

Maximum depth of the tree [number] <put parameter description here>

Default: 65535

Minimum number of samples in each node [number] <put parameter description here>

Default: 10

Termination criteria for regression tree [number] <put parameter description here>

Default: 0.01

Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal
<put parameter description here>

Default: 10

K-fold cross-validations [number] <put parameter description here>

Default: 10

Set UseSelfRule flag to false [boolean] <put parameter description here>

Default: True

Set TruncatePrunedTree flag to false [boolean] <put parameter description here>

Default: True

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierdt', -io.il, -io.vd, -io.imstat, -elev.default, -samp
```

See also

TrainImagesClassifier (gbt)

Description

<put algorithm description here>

Parameters

Input Image List [multipleinput: rasters] <put parameter description here>

Input Vector Data List [multipleinput: any vectors] <put parameter description here>

Input XML image statistics file [file] Optional.

<put parameter description here>

Default elevation [number] <put parameter description here>

Default: *0*

Maximum training sample size per class [number] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [number] <put parameter description here>

Default: *1000*

On edge pixel inclusion [boolean] <put parameter description here>

Default: *True*

Training and validation sample ratio [number] <put parameter description here>

Default: *0.5*

Name of the discrimination field [string] <put parameter description here>

Default: *Class*

Classifier to use for the training [selection] <put parameter description here>

Options:

- 0 — gbt

Default: *0*

Number of boosting algorithm iterations [number] <put parameter description here>

Default: *200*

Regularization parameter [number] <put parameter description here>

Default: *0.01*

Portion of the whole training set used for each algorithm iteration [number]

<put parameter description here>

Default: *0.8*

Maximum depth of the tree [number] <put parameter description here>

Default: *3*

set user defined seed [number] <put parameter description here>

Default: *0*

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifiergbt', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

See also

TrainImagesClassifier (knn)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: 0

Maximum training sample size per class [**number**] <put parameter description here>

Default: 1000

Maximum validation sample size per class [**number**] <put parameter description here>

Default: 1000

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: *True*

Training and validation sample ratio [**number**] <put parameter description here>

Default: 0.5

Name of the discrimination field [**string**] <put parameter description here>

Default: *Class*

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — knn

Default: 0

Number of Neighbors [**number**] <put parameter description here>

Default: 32

set user defined seed [**number**] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [**file**] <put output description here>

Output model [**file**] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierknn', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

See also

TrainImagesClassifier (libsvm)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: *0*

Maximum training sample size per class [**number**] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [**number**] <put parameter description here>

Default: *1000*

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: *True*

Training and validation sample ratio [**number**] <put parameter description here>

Default: *0.5*

Name of the discrimination field [**string**] <put parameter description here>

Default: *Class*

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — libsvm

Default: *0*

SVM Kernel Type [**selection**] <put parameter description here>

Options:

- 0 — linear
- 1 — rbf
- 2 — poly
- 3 — sigmoid

Default: *0*

Cost parameter C [number] <put parameter description here>

Default: *1*

Parameters optimization [boolean] <put parameter description here>

Default: *True*

set user defined seed [number] <put parameter description here>

Default: *0*

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierlibsvm', -io.il, -io.vd, -io.imstat, -elev.default, -
```

See also

TrainImagesClassifier (rf)

Description

<put algorithm description here>

Parameters

Input Image List [multipleinput: rasters] <put parameter description here>

Input Vector Data List [multipleinput: any vectors] <put parameter description here>

Input XML image statistics file [file] Optional.

<put parameter description here>

Default elevation [number] <put parameter description here>

Default: *0*

Maximum training sample size per class [number] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [number] <put parameter description here>

Default: *1000*

On edge pixel inclusion [boolean] <put parameter description here>

Default: *True*

Training and validation sample ratio [number] <put parameter description here>

Default: *0.5*

Name of the discrimination field [string] <put parameter description here>

Default: *Class*

Classifier to use for the training [selection] <put parameter description here>

Options:

- 0 — rf

Default: 0

Maximum depth of the tree [number] <put parameter description here>

Default: 5

Minimum number of samples in each node [number] <put parameter description here>

Default: 10

Termination Criteria for regression tree [number] <put parameter description here>

Default: 0

Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal <put parameter description here>

Default: 10

Size of the randomly selected subset of features at each tree node [number] <put parameter description here>

Default: 0

Maximum number of trees in the forest [number] <put parameter description here>

Default: 100

Sufficient accuracy (OOB error) [number] <put parameter description here>

Default: 0.01

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierrf', -io.il, -io.vd, -io.imstat, -elev.default, -samp
```

See also

TrainImagesClassifier (svm)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: *0*

Maximum training sample size per class [**number**] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [**number**] <put parameter description here>

Default: *1000*

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: *True*

Training and validation sample ratio [**number**] <put parameter description here>

Default: *0.5*

Name of the discrimination field [**string**] <put parameter description here>

Default: *Class*

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — svm

Default: *0*

SVM Model Type [**selection**] <put parameter description here>

Options:

- 0 — csvc
- 1 — nusvc
- 2 — oneclass

Default: *0*

SVM Kernel Type [**selection**] <put parameter description here>

Options:

- 0 — linear
- 1 — rbf
- 2 — poly
- 3 — sigmoid

Default: *0*

Cost parameter C [**number**] <put parameter description here>

Default: *1*

Parameter nu of a SVM optimization problem (NU_SVC / ONE_CLASS) [**number**] <put parameter description here>

Default: *0*

Parameter coef0 of a kernel function (POLY / SIGMOID) [number] <put parameter description here>

Default: 0

Parameter gamma of a kernel function (POLY / RBF / SIGMOID) [number] <put parameter description here>

Default: 1

Parameter degree of a kernel function (POLY) [number] <put parameter description here>

Default: 1

Parameters optimization [boolean] <put parameter description here>

Default: *True*

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifiersvm', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

See also

Unsupervised KMeans image classification

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Validity Mask [raster] Optional.

<put parameter description here>

Training set size [number] <put parameter description here>

Default: 100

Number of classes [number] <put parameter description here>

Default: 5

Maximum number of iterations [number] <put parameter description here>

Default: 1000

Convergence threshold [number] <put parameter description here>

Default: *0.0001*

Outputs

Output Image [raster] <put output description here>

Centroid filename [file] <put output description here>

Console usage

```
processing.runalg('otb:unsupervisedkmeansimageclassification', -in, -ram, -vm, -ts, -nc, -maxit,
```

See also

.

18.4.7 Miscellaneous

Band Math

Description

<put algorithm description here>

Parameters

Input image list [multipleinput: rasters] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Expression [string] <put parameter description here>

Default: *None*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:bandmath', -il, -ram, -exp, -out)
```

See also

ComputeModulusAndPhase-one (OneEntry)

Description

<put algorithm description here>

Parameters

Number Of inputs [selection] <put parameter description here>

Options:

- 0 — one

Default: 0

Input image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Modulus [raster] <put output description here>

Phase [raster] <put output description here>

Console usage

```
processing.runalg('otb:computemodulusandphaseoneoneentry', -nbininput, -nbininput.one.in, -ram, -mod,
```

See also

ComputeModulusAndPhase-two (TwoEntries)

Description

<put algorithm description here>

Parameters

Number Of inputs [selection] <put parameter description here>

Options:

- 0 — two

Default: 0

Real part input [raster] <put parameter description here>

Imaginary part input [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Modulus [raster] <put output description here>

Phase [raster] <put output description here>

Console usage

```
processing.runalg('otb:computemodulusandphasetwoentries', -nbinput, -nbinput.two.re, -nbinput.t
```

See also

Images comparaison

Description

<put algorithm description here>

Parameters

Reference image [raster] <put parameter description here>

Reference image channel [number] <put parameter description here>

Default: *1*

Measured image [raster] <put parameter description here>

Measured image channel [number] <put parameter description here>

Default: *1*

Start X [number] <put parameter description here>

Default: *0*

Start Y [number] <put parameter description here>

Default: *0*

Size X [number] <put parameter description here>

Default: *0*

Size Y [number] <put parameter description here>

Default: *0*

Outputs

Console usage

```
processing.runalg('otb:imagescomparaison', -ref.in, -ref.channel, -meas.in, -meas.channel, -roi.st
```

See also

Image to KMZ Export

Description

<put algorithm description here>

Parameters

Input image [raster] <put parameter description here>

Tile Size [number] <put parameter description here>

Default: *512*

Image logo [raster] Optional.

<put parameter description here>

Image legend [raster] Optional.

<put parameter description here>

Default elevation [number] <put parameter description here>

Default: *0*

Outputs

Output .kmz product [file] <put output description here>

Console usage

```
processing.runalg('otb:imagetokmzexport', -in, -tilesize, -logo, -legend, -elev.default, -out)
```

See also

.

18.4.8 Segmentation

Connected Component Segmentation

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Mask expression [string] Optional.

<put parameter description here>

Default: *None*

Connected Component Expression [string] <put parameter description here>

Default: *None*

Minimum Object Size [number] <put parameter description here>

Default: *2*

OBIA Expression [string] Optional.

<put parameter description here>

Default: *None*

Default elevation [number] <put parameter description here>

Default: *0*

Outputs

Output Shape [vector] <put output description here>

Console usage

```
processing.runalg('otb:connectedcomponentsegmentation', -in, -mask, -expr, -minsize, -obia, -elev
```

See also

Exact Large-Scale Mean-Shift segmentation, step 2

Description

<put algorithm description here>

Parameters

Filtered image [raster] <put parameter description here>

Spatial image [raster] Optional.

<put parameter description here>

Range radius [number] <put parameter description here>

Default: *15*

Spatial radius [number] <put parameter description here>

Default: *5*

Minimum Region Size [number] <put parameter description here>

Default: *0*

Size of tiles in pixel (X-axis) [number] <put parameter description here>

Default: *500*

Size of tiles in pixel (Y-axis) [number] <put parameter description here>

Default: *500*

Directory where to write temporary files [file] Optional.

<put parameter description here>

Temporary files cleaning [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:exactlargescalemeanshiftsegmentationstep2', -in, -inpos, -ranger, -spatial,
```

See also

Exact Large-Scale Mean-Shift segmentation, step 3 (optional)

Description

<put algorithm description here>

Parameters

Input image [raster] <put parameter description here>

Segmented image [raster] <put parameter description here>

Minimum Region Size [number] <put parameter description here>

Default: *50*

Size of tiles in pixel (X-axis) [number] <put parameter description here>

Default: *500*

Size of tiles in pixel (Y-axis) [number] <put parameter description here>

Default: *500*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:exactlargescalemeanshiftsegmentationstep3optional', -in, -inseg, -minsize,
```

See also

Exact Large-Scale Mean-Shift segmentation, step 4

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Segmented image [raster] <put parameter description here>

Size of tiles in pixel (X-axis) [number] <put parameter description here>

Default: *500*

Size of tiles in pixel (Y-axis) [number] <put parameter description here>

Default: *500*

Outputs

Output GIS vector file [vector] <put output description here>

Console usage

```
processing.runalg('otb:exactlargescalemeanshiftsegmentationstep4', -in, -inseg, -tilesizex, -tile
```

See also

Hoover compare segmentation

Description

<put algorithm description here>

Parameters

Input ground truth [raster] <put parameter description here>

Input machine segmentation [raster] <put parameter description here>

Background label [number] <put parameter description here>

Default: *0*

Overlapping threshold [number] <put parameter description here>

Default: *0.75*

Correct detection score [number] <put parameter description here>

Default: *0.0*

Over-segmentation score [number] <put parameter description here>

Default: *0.0*

Under-segmentation score [number] <put parameter description here>

Default: *0.0*

Missed detection score [number] <put parameter description here>

Default: *0.0*

Outputs

Colored ground truth output [raster] <put output description here>

Colored machine segmentation output [raster] <put output description here>

Console usage

```
processing.runalg('otb:hoovercomparesegmentation', -ingt, -inms, -bg, -th, -rc, -rf, -ra, -rm, -o
```

See also

Segmentation (cc)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Segmentation algorithm [selection] <put parameter description here>

Options:

- 0 — cc

Default: 0

Condition [string] <put parameter description here>

Default: *None*

Processing mode [selection] <put parameter description here>

Options:

- 0 — vector

Default: 0

Writing mode for the output vector file [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: 0

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [**boolean**] <put parameter description here>

Default: *True*

Minimum object size [**number**] <put parameter description here>

Default: *1*

Simplify polygons [**number**] <put parameter description here>

Default: *0.1*

Layer name [**string**] <put parameter description here>

Default: *layer*

Geometry index field name [**string**] <put parameter description here>

Default: *DN*

Tiles size [**number**] <put parameter description here>

Default: *1024*

Starting geometry index [**number**] <put parameter description here>

Default: *1*

OGR options for layer creation [**string**] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [**vector**] <put output description here>

Console usage

```
processing.runalg('otb:segmentationcc', -in, -filter, -filter.cc.expr, -mode, -mode.vector.outmode)
```

See also

Segmentation (edison)

Description

<put algorithm description here>

Parameters

Input Image [**raster**] <put parameter description here>

Segmentation algorithm [**selection**] <put parameter description here>

Options:

- 0 — edison

Default: *0*

Spatial radius [**number**] <put parameter description here>

Default: *5*

Range radius [number] <put parameter description here>

Default: *15*

Minimum region size [number] <put parameter description here>

Default: *100*

Scale factor [number] <put parameter description here>

Default: *1*

Processing mode [selection] <put parameter description here>

Options:

- 0 — vector

Default: *0*

Writing mode for the output vector file [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: *0*

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [boolean] <put parameter description here>

Default: *True*

Minimum object size [number] <put parameter description here>

Default: *1*

Simplify polygons [number] <put parameter description here>

Default: *0.1*

Layer name [string] <put parameter description here>

Default: *layer*

Geometry index field name [string] <put parameter description here>

Default: *DN*

Tiles size [number] <put parameter description here>

Default: *1024*

Starting geometry index [number] <put parameter description here>

Default: *1*

OGR options for layer creation [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [**vector**] <put output description here>

Console usage

```
processing.runalg('otb:segmentationedison', -in, -filter, -filter.edison.spatialr, -filter.edison
```

See also

Segmentation (meanshift)

Description

<put algorithm description here>

Parameters

Input Image [**raster**] <put parameter description here>

Segmentation algorithm [**selection**] <put parameter description here>

Options:

- 0 — meanshift

Default: 0

Spatial radius [**number**] <put parameter description here>

Default: 5

Range radius [**number**] <put parameter description here>

Default: 15

Mode convergence threshold [**number**] <put parameter description here>

Default: 0.1

Maximum number of iterations [**number**] <put parameter description here>

Default: 100

Minimum region size [**number**] <put parameter description here>

Default: 100

Processing mode [**selection**] <put parameter description here>

Options:

- 0 — vector

Default: 0

Writing mode for the output vector file [**selection**] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw

- 3 — ulu

Default: *0*

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [boolean] <put parameter description here>

Default: *True*

Minimum object size [number] <put parameter description here>

Default: *1*

Simplify polygons [number] <put parameter description here>

Default: *0.1*

Layer name [string] <put parameter description here>

Default: *layer*

Geometry index field name [string] <put parameter description here>

Default: *DN*

Tiles size [number] <put parameter description here>

Default: *1024*

Starting geometry index [number] <put parameter description here>

Default: *1*

OGR options for layer creation [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [vector] <put output description here>

Console usage

```
processing.runalg('otb:segmentationmeanshift', -in, -filter, -filter.meanshift.spatialr, -filter.
```

See also

Segmentation (mprofiles)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Segmentation algorithm [selection] <put parameter description here>

Options:

- 0 — mprofiles

Default: 0

Profile Size [number] <put parameter description here>

Default: 5

Initial radius [number] <put parameter description here>

Default: 1

Radius step. [number] <put parameter description here>

Default: 1

Threshold of the final decision rule [number] <put parameter description here>

Default: 1

Processing mode [selection] <put parameter description here>

Options:

- 0 — vector

Default: 0

Writing mode for the output vector file [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: 0

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [boolean] <put parameter description here>

Default: *True*

Minimum object size [number] <put parameter description here>

Default: 1

Simplify polygons [number] <put parameter description here>

Default: 0.1

Layer name [string] <put parameter description here>

Default: *layer*

Geometry index field name [string] <put parameter description here>

Default: *DN*

Tiles size [number] <put parameter description here>

Default: *1024*

Starting geometry index [number] <put parameter description here>

Default: *1*

OGR options for layer creation [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [vector] <put output description here>

Console usage

```
processing.runalg('otb:segmentationmprofiles', -in, -filter, -filter.mprofiles.size, -filter.mpro
```

See also

Segmentation (watershed)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Segmentation algorithm [selection] <put parameter description here>

Options:

- 0 — watershed

Default: *0*

Depth Threshold [number] <put parameter description here>

Default: *0.01*

Flood Level [number] <put parameter description here>

Default: *0.1*

Processing mode [selection] <put parameter description here>

Options:

- 0 — vector

Default: *0*

Writing mode for the output vector file [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: 0

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [boolean] <put parameter description here>

Default: *True*

Minimum object size [number] <put parameter description here>

Default: 1

Simplify polygons [number] <put parameter description here>

Default: 0.1

Layer name [string] <put parameter description here>

Default: *layer*

Geometry index field name [string] <put parameter description here>

Default: *DN*

Tiles size [number] <put parameter description here>

Default: 1024

Starting geometry index [number] <put parameter description here>

Default: 1

OGR options for layer creation [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [vector] <put output description here>

Console usage

```
processing.runalg('otb:segmentationwatershed', -in, -filter, -filter.watershed.threshold, -filter
```

See also

.

18.4.9 Stereo

Stereo Framework

Description

<put algorithm description here>

Parameters

Input images list [**multipleinput: rasters**] <put parameter description here>

Couples list [**string**] Optional.

<put parameter description here>

Default: *None*

Image channel used for the block matching [**number**] <put parameter description here>

Default: *1*

Default elevation [**number**] <put parameter description here>

Default: *0*

Output resolution [**number**] <put parameter description here>

Default: *1*

NoData value [**number**] <put parameter description here>

Default: *-32768*

Method to fuse measures in each DSM cell [**selection**] <put parameter description here>

Options:

- 0 — max
- 1 — min
- 2 — mean
- 3 — acc

Default: *0*

Parameters estimation modes [**selection**] <put parameter description here>

Options:

- 0 — fit
- 1 — user

Default: *0*

Upper Left X [**number**] <put parameter description here>

Default: *0.0*

Upper Left Y [**number**] <put parameter description here>

Default: *0.0*

Size X [**number**] <put parameter description here>

Default: *0*

Size Y [number] <put parameter description here>

Default: 0

Pixel Size X [number] <put parameter description here>

Default: 0.0

Pixel Size Y [number] <put parameter description here>

Default: 0.0

Output Cartographic Map Projection [selection] <put parameter description here>

Options:

- 0 — utm
- 1 — lambert2
- 2 — lambert93
- 3 — wgs
- 4 — epsg

Default: 3

Zone number [number] <put parameter description here>

Default: 31

Northern Hemisphere [boolean] <put parameter description here>

Default: *True*

EPSG Code [number] <put parameter description here>

Default: 4326

Step of the deformation grid (in pixels) [number] <put parameter description here>

Default: 16

Sub-sampling rate for epipolar grid inversion [number] <put parameter description here>

Default: 10

Block-matching metric [selection] <put parameter description here>

Options:

- 0 — ssdmean
- 1 — ssd
- 2 — ncc
- 3 — lp

Default: 0

p value [number] <put parameter description here>

Default: 1

Radius of blocks for matching filter (in pixels) [number] <put parameter description here>

Default: 2

Minimum altitude offset (in meters) [number] <put parameter description here>

Default: -20

Maximum altitude offset (in meters) [number] <put parameter description here>

Default: *20*

Use bijection consistency in block matching strategy [boolean] <put parameter description here>

Default: *True*

Use median disparities filtering [boolean] <put parameter description here>

Default: *True*

Correlation metric threshold [number] <put parameter description here>

Default: *0.6*

Input left mask [raster] Optional.

<put parameter description here>

Input right mask [raster] Optional.

<put parameter description here>

Discard pixels with low local variance [number] <put parameter description here>

Default: *50*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Output DSM [raster] <put output description here>

Console usage

```
processing.runalg('otb:stereoframework', -input.il, -input.co, -input.channel, -elev.default, -out
```

See also

.

18.4.10 Vector

Concatenate

Description

<put algorithm description here>

Parameters

Input VectorDatas to concatenate [multipleinput: any vectors] <put parameter description here>

Outputs

Concatenated VectorData [**vector**] <put output description here>

Console usage

```
processing.runalg('otb:concatenate', -vd, -out)
```

See also

.

18.5 QGIS algorithm provider

QGIS algorithm provider implements various analysis and geoprocessing operations using mostly only QGIS API. So almost all algorithms from this provider will work “out of the box” without any additional configuration.

This provider incorporates fTools functionality, some algorithms from mmQGIS plugin and also adds its own algorithms.

.

18.5.1 Database

Import into PostGIS

Description

<put algorithm description here>

Parameters

Layer to import [**vector: any**] <put parameter description here>

Database (connection name) [**selection**] <put parameter description here>

Options:

- 0 — local

Default: 0

Schema (schema name) [**string**] <put parameter description here>

Default: *public*

Table to import to (leave blank to use layer name) [**string**] <put parameter description here>

Default: *(not set)*

Primary key field [**tablefield: any**] Optional.

<put parameter description here>

Geometry column [**string**] <put parameter description here>

Default: *geom*

Overwrite [boolean] <put parameter description here>

Default: *True*

Create spatial index [boolean] <put parameter description here>

Default: *True*

Convert field names to lowercase [boolean] <put parameter description here>

Default: *True*

Drop length constraints on character fields [boolean] <put parameter description here>

Default: *False*

Outputs

Console usage

```
processing.runalg('qgis:importintopostgis', input, database, schema, tablename, primary_key, geom
```

See also

PostGIS execute SQL

Description

<put algorithm description here>

Parameters

Database [string] <put parameter description here>

Default: *(not set)*

SQL query [string] <put parameter description here>

Default: *(not set)*

Outputs

Console usage

```
processing.runalg('qgis:postgisexecutesql', database, sql)
```

See also

.

18.5.2 Raster general

Set style for raster layer

Description

<put algorithm description here>

Parameters

Raster layer [**raster**] <put parameter description here>

Style file [**file**] <put parameter description here>

Outputs

Styled layer [**raster**] <put output description here>

Console usage

```
processing.runalg('qgis:setstyleforrasterlayer', input, style)
```

See also

.

18.5.3 Raster

Hypsometric curves

Description

Calculate hypsometric curves for features of polygon layer and save them as CSV file for further processing.

Parameters

DEM to analyze [**raster**] DEM to use for calculating altitudes.

Boundary layer [**vector: polygon**] Polygonal vector layer with boundaries of areas used to calculate hypsometric curves.

Step [**number**] Distance between curves.

Default: *100.0*

Use % of area instead of absolute value [**boolean**] Write area percentage to “Area” field of the CSV file instead of absolute area value.

Default: *False*

Outputs

Output directory [directory] Directory where output will be saved. For each feature from input vector layer CSV file with area and altitude values will be created.

File name consists of prefix `hystogram_` followed by layer name and feature ID.

Console usage

```
processing.runalg('qgis:hypsometriccurves', input_dem, boundary_layer, step, use_percentage, output_directory)
```

See also

Raster layer statistics

Description

Calculates basic statistics of the raster layer.

Parameters

Input layer [raster] Raster to analyze.

Outputs

Statistics [html] Analysis results in HTML format.

Minimum value [number] Minimum cell value.

Maximum value [number] Maximum cell value.

Sum [number] Sum of all cells values.

Mean value [number] Mean cell value.

valid cells count [number] Number of cell with data.

No-data cells count [number] Number of NODATA cells.

Standard deviation [number] Standard deviation of cells values.

Console usage

```
processing.runalg('qgis:rasterlayerstatistics', input_raster, output_html_file)
```

See also

Zonal Statistics

Description

Calculates some statistics values for pixels of input raster inside certain zones, defined as polygon layer.

Following values calculated for each zone:

- minimum

- maximum
- sum
- count
- mean
- standard deviation
- number of unique values
- range
- variance

Parameters

Raster layer [**raster**] Raster to analyze.

Raster band [**number**] Number of raster band to analyze.

Default: *1*

Vector layer containing zones [**vector: polygon**] Layer with zones boundaries.

Output column prefix [**string**] Prefix for output fields.

Default: *_*

Load whole raster in memory [**boolean**] Determines if raster band will be loaded in memory (**True**) or readed by chunks (**False**). Useful only when disk IO or raster scanning inefficiencies are your limiting factor.

Default: *True*

Outputs

Output layer [**vector**] The resulting layer. Basically this is same layer as zones layer with new columns containing statistics added.

Console usage

```
processing.runalg('qgis:zonalstatistics', input_raster, raster_band, input_vector, column_prefix,
```

See also

.

18.5.4 Table

Frequency analysis

Description

<put algorithm description here>

Parameters

input [vector: any] <put parameter description here>

fields [string] <put parameter description here>

Default: *(not set)*

Outputs

output [table] <put output description here>

Console usage

```
processing.runalg('qgis:frequencyanalysis', input, fields, output)
```

See also

.

18.5.5 Vector analysis

Count points in polygon

Description

Counts the number of points present in each feature of a polygon layer.

Parameters

Polygons [vector: polygon] Polygons layer.

Points [vector: point] Points layer.

Count field name [string] The name of the attribute table column containing the points number.

Default: *NUMPOINTS*

Outputs

Result [vector] Resulting layer with the attribute table containing the new column of the points count.

Console usage

```
processing.runalg('qgis:countpointsinpolygon', polygons, points, field, output)
```

See also

Count points in polygon (weighted)

Description

Counts the number of points in each feature of a polygon layer and calculates the mean of the selected field for each feature of the polygon layer. These values will be added to the attribute table of the resulting polygon layer.

Parameters

Polygons [vector: polygon] Polygons layer.

Points [vector: point] Points layer.

Weight field [tablefield: any] Weight field of the points attribute table.

Count field name [string] Name of the column for the new weighted field.

Default: *NUMPOINTS*

Outputs

Result [vector] The resulting polygons layer.

Console usage

```
processing.runalg('qgis:countpointsinpolygonweighted', polygons, points, weight, field, output)
```

See also

Count unique points in polygon

Description

Counts the number of unique values of a points in a polygons layer. Creates a new polygons layer with an extra column in the attribute table containing the count of unique values for each feature.

Parameters

Polygons [vector: polygon] Polygons layer.

Points [vector: point] Points layer.

Class field [tablefield: any] Points layer column name of the unique value chosen.

Count field name [string] Column name containing the count of unique values in the resulting polygons layer.

Default: *NUMPOINTS*

Outputs

Result [vector] The resulting polygons layer.

Console usage

```
processing.runalg('qgis:countuniquepointsinpolygon', polygons, points, classfield, field, output)
```

See also

Distance matrix

Description

<put algorithm description here>

Parameters

Input point layer [**vector: point**] <put parameter description here>

Input unique ID field [**tablefield: any**] <put parameter description here>

Target point layer [**vector: point**] <put parameter description here>

Target unique ID field [**tablefield: any**] <put parameter description here>

Output matrix type [**selection**] <put parameter description here>

Options:

- 0 — Linear ($N \times k \times 3$) distance matrix
- 1 — Standard ($N \times T$) distance matrix
- 2 — Summary distance matrix (mean, std. dev., min, max)

Default: 0

Use only the nearest (k) target points [**number**] <put parameter description here>

Default: 0

Outputs

Distance matrix [**table**] <put output description here>

Console usage

```
processing.runalg('qgis:distancematrix', input_layer, input_field, target_layer, target_field, ma
```

See also

Distance to nearest hub

Description

<put algorithm description here>

Parameters

Source points layer [vector: any] <put parameter description here>

Destination hubs layer [vector: any] <put parameter description here>

Hub layer name attribute [tablefield: any] <put parameter description here>

Output shape type [selection] <put parameter description here>

Options:

- 0 — Point
- 1 — Line to hub

Default: 0

Measurement unit [selection] <put parameter description here>

Options:

- 0 — Meters
- 1 — Feet
- 2 — Miles
- 3 — Kilometers
- 4 — Layer units

Default: 0

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:distancetonearesthub', points, hubs, field, geometry, unit, output)
```

See also

Generate points (pixel centroids) along line

Description

<put algorithm description here>

Parameters

Raster layer [raster] <put parameter description here>

Vector layer [vector: line] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:generatepointspixelcentroidsalongline', input_raster, input_vector, output)
```

See also

Generate points (pixel centroids) inside polygons

Description

<put algorithm description here>

Parameters

Raster layer [raster] <put parameter description here>

Vector layer [vector: polygon] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:generatepointspixelcentroidsinsidepolygons', input_raster, input_vector, output)
```

See also

Hub lines

Description

Creates hub and spoke diagrams with lines drawn from points on the `Spoke Point` layer to matching points in the `Hub Point` layer. Determination of which hub goes with each point is based on a match between the `Hub ID field` on the hub points and the `Spoke ID field` on the spoke points.

Parameters

Hub point layer [vector: any] <put parameter description here>

Hub ID field [tablefield: any] <put parameter description here>

Spoke point layer [vector: any] <put parameter description here>

Spoke ID field [tablefield: any] <put parameter description here>

Outputs

Output [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:hublines', hubs, hub_field, spokes, spoke_field, output)
```

See also

Mean coordinate(s)

Description

Calculates the mean of the coordinates of a layer starting from a field of the attribute table.

Parameters

Input layer [**vector: any**] <put parameter description here>

Weight field [**tablefield: numeric**] Optional.

Field to use if you want to perform a weighted mean.

Unique ID field [**tablefield: numeric**] Optional.

Unique field on which the calculation of the mean will be made.

Outputs

Result [**vector**] The resulting points layer.

Console usage

```
processing.runalg('qgis:meancoordinates', points, weight, uid, output)
```

See also

Nearest neighbour analysis

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Outputs

Result [**html**] <put output description here>

Observed mean distance [**number**] <put output description here>

Expected mean distance [**number**] <put output description here>

Nearest neighbour index [**number**] <put output description here>

Number of points [number] <put output description here>

Z-Score [number] <put output description here>

Console usage

```
processing.runalg('qgis:nearestneighbouranalysis', points, output)
```

See also

Sum line lengths

Description

<put algorithm description here>

Parameters

Lines [vector: line] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Lines length field name [string] <put parameter description here>

Default: *LENGTH*

Lines count field name [string] <put parameter description here>

Default: *COUNT*

Outputs

Result [vector] <put output description here>

Console usage

```
processing.runalg('qgis:sumlinelengths', lines, polygons, len_field, count_field, output)
```

See also

.

18.5.6 Vector creation

Create grid

Description

Creates a grid.

Parameters

Grid type [selection] Grid type.

Options:

- 0 — Rectangle (line)
- 1 — Rectangle (polygon)
- 2 — Diamond (polygon)
- 3 — Hexagon (polygon)

Default: *0*

Width [number] Horizontal extent of the grid.

Default: *360.0*

Height [number] Vertical extent of the grid.

Default: *180.0*

Horizontal spacing [number] X-axes spacing between the lines.

Default: *10.0*

Vertical spacing [number] Y-axes spacing between the lines.

Default: *10.0*

Center X [number] X-coordinate of the grid center.

Default: *0.0*

Center Y [number] Y-coordinate of the grid center.

Default: *0.0*

Output CRS [crs] Coordinate reference system for grid.

Default: *EPSG:4326*

Outputs

Output [vector] The resulting grid layer (lines or polygons).

Console usage

```
processing.runalg('qgis:creategrid', type, width, height, hspacing, vspacing, centerx, centery, c
```

See also

Points layer from table

Description

Creates points layer from geometryless table with columns that contain point coordinates.

Parameters

Input layer [table] Input table

X field [tablefield: any] Table column containing the X coordinate.

Y field [tablefield: any] Table column containing the Y coordinate.

Target CRS [crs] Coordinate reference system to use for layer.

Default: *EPSG:4326*

Outputs

Output layer [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:pointslayerfromtable', input, xfield, yfield, target_crs, output)
```

See also

Points to path

Description

<put algorithm description here>

Parameters

Input point layer [vector: point] <put parameter description here>

Group field [tablefield: any] <put parameter description here>

Order field [tablefield: any] <put parameter description here>

Date format (if order field is DateTime) [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Paths [vector] <put output description here>

Directory [directory] <put output description here>

Console usage

```
processing.runalg('qgis:pointstopath', vector, group_field, order_field, date_format, output_line)
```

See also

Random points along line

Description

<put algorithm description here>

Parameters

Input layer [vector: line] <put parameter description here>

Number of points [number] <put parameter description here>

Default: 1

Minimum distance [number] <put parameter description here>

Default: 0.0

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsalongline', vector, point_number, min_distance, output)
```

See also

Random points in extent

Description

<put algorithm description here>

Parameters

Input extent [extent] <put parameter description here>

Default: 0,1,0,1

Points number [number] <put parameter description here>

Default: 1

Minimum distance [number] <put parameter description here>

Default: 0.0

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsinextent', extent, point_number, min_distance, output)
```

See also

Random points in layer bounds

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon] <put parameter description here>

Points number [number] <put parameter description here>

Default: 1

Minimum distance [number] <put parameter description here>

Default: 0.0

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsinlayerbounds', vector, point_number, min_distance, output)
```

See also

Random points inside polygons (fixed)

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon] <put parameter description here>

Sampling strategy [selection] <put parameter description here>

Options:

- 0 — Points count
- 1 — Points density

Default: 0

Number or density of points [number] <put parameter description here>

Default: *1.0*

Minimum distance [number] <put parameter description here>

Default: *0.0*

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsinsidepolygonsfixed', vector, strategy, value, min_distance, ...)
```

See also

Random points inside polygons (variable)

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon] <put parameter description here>

Sampling strategy [selection] <put parameter description here>

Options:

- 0 — Points count
- 1 — Points density

Default: *0*

Number field [tablefield: numeric] <put parameter description here>

Minimum distance [number] <put parameter description here>

Default: *0.0*

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsinsidepolygonsvariable', vector, strategy, field, min_distance, ...)
```


See also

Regular points

Description

<put algorithm description here>

Parameters

Input extent [extent] <put parameter description here>

Default: *0,1,0,1*

Point spacing/count [number] <put parameter description here>

Default: *0.0001*

Initial inset from corner (LH side) [number] <put parameter description here>

Default: *0.0*

Apply random offset to point spacing [boolean] <put parameter description here>

Default: *False*

Use point spacing [boolean] <put parameter description here>

Default: *True*

Outputs

Regular points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:regularpoints', extent, spacing, inset, randomize, is_spacing, output)
```

See also

Vector grid

Description

<put algorithm description here>

Parameters

Grid extent [extent] <put parameter description here>

Default: *0,1,0,1*

X spacing [number] <put parameter description here>

Default: *0.0001*

Y spacing [number] <put parameter description here>

Default: *0.0001*

Grid type [selection] <put parameter description here>

Options:

- 0 — Output grid as polygons
- 1 — Output grid as lines

Default: 0

Outputs

Grid [vector] <put output description here>

Console usage

```
processing.runalg('qgis:vectorgrid', extent, step_x, step_y, type, output)
```

See also

.

18.5.7 Vector general

Delete duplicate geometries

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:deleteduplicategeometries', input, output)
```

See also

Join attributes by location

Description

<put algorithm description here>

Parameters

Target vector layer [**vector: any**] <put parameter description here>

Join vector layer [**vector: any**] <put parameter description here>

Attribute summary [**selection**] <put parameter description here>

Options:

- 0 — Take attributes of the first located feature
- 1 — Take summary of intersecting features

Default: 0

Statistics for summary (comma separated) [**string**] <put parameter description here>

Default: *sum,mean,min,max,median*

Output table [**selection**] <put parameter description here>

Options:

- 0 — Only keep matching records
- 1 — Keep all records (including non-matching target records)

Default: 0

Outputs

Output layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:joinattributesbylocation', target, join, summary, stats, keep, output)
```

See also

Join attributes table

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

Input layer 2 [**table**] <put parameter description here>

Table field [**tablefield: any**] <put parameter description here>

Table field 2 [**tablefield: any**] <put parameter description here>

Outputs

Output layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:joinattributetable', input_layer, input_layer_2, table_field, table_fiel
```

See also

Merge vector layers

Description

<put algorithm description here>

Parameters

Input layer 1 [vector: any] <put parameter description here>

Input layer 2 [vector: any] <put parameter description here>

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:mergevectorlayers', layer1, layer2, output)
```

See also

Polygon from layer extent

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Calculate extent for each feature separately [boolean] <put parameter description here>

Default: *False*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:polygonfromlayerextent', input_layer, by_feature, output)
```

See also

Reproject layer

Description

Reprojects a vector layer in a different CRS.

Parameters

Input layer [vector: any] Layer to reproject.

Target CRS [crs] Destination coordinate reference system.

Default: *EPSG:4326*

Outputs

Reprojected layer [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:reprojectlayer', input, target_crs, output)
```

See also

Save selected features

Description

Saves the selected features as a new layer.

Parameters

Input layer [vector: any] Layer to process.

Outputs

Output layer with selected features [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:savesselectedfeatures', input_layer, output_layer)
```

See also

Set style for vector layer

Description

<put algorithm description here>

Parameters

Vector layer [vector: any] <put parameter description here>

Style file [file] <put parameter description here>

Outputs

Styled layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:setstyleforvectorlayer', input, style)
```

See also

Snap points to grid

Description

<put algorithm description here>

Parameters

Input Layer [vector: any] <put parameter description here>

Horizontal spacing [number] <put parameter description here>

Default: *0.1*

Vertical spacing [number] <put parameter description here>

Default: *0.1*

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:snappointstogrid', input, hspacing, vspacing, output)
```

See also

Split vector layer

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Unique ID field [tablefield: any] <put parameter description here>

Outputs

Output directory [directory] <put output description here>

Console usage

```
processing.runalg('qgis:splitvectorlayer', input, field, output)
```

See also

.

18.5.8 Vector geometry

Concave hull

Description

<put algorithm description here>

Parameters

Input point layer [vector: point] <put parameter description here>

Threshold (0-1, where 1 is equivalent with Convex Hull) [number] <put parameter description here>

Default: *0.3*

Allow holes [boolean] <put parameter description here>

Default: *True*

Split multipart geometry into singleparts geometries [boolean] <put parameter description here>

Default: *False*

Outputs

Concave hull [vector] <put output description here>

Console usage

```
processing.runalg('qgis:concavehull', input, alpha, holes, no_multigeometry, output)
```

See also

Convert geometry type

Description

Converts a geometry type to another one.

Parameters

Input layer [vector: any] Layer in input.

New geometry type [selection] Type of conversion to perform.

Options:

- 0 — Centroids
- 1 — Nodes
- 2 — Linestrings
- 3 — Multilinestrings
- 4 — Polygons

Default: 0

Outputs

Output [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:convertgeometrytype', input, type, output)
```

See also

Convex hull

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

Field (optional, only used if creating convex hulls by classes) [**tablefield: any**]

Optional.

<put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — Create single minimum convex hull
- 1 — Create convex hulls based on field

Default: 0

Outputs

Convex hull [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:convexhull', input, field, method, output)
```

See also

Create points along lines

Description

<put algorithm description here>

Parameters

lines [**vector: any**] <put parameter description here>

distance [**number**] <put parameter description here>

Default: 1

startpoint [**number**] <put parameter description here>

Default: 0

endpoint [**number**] <put parameter description here>

Default: 0

Outputs

output [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:createpointsalonglines', lines, distance, startpoint, endpoint, output)
```

See also

Delaunay triangulation

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Outputs

Delaunay triangulation [vector] <put output description here>

Console usage

```
processing.runalg('qgis:delaunaytriangulation', input, output)
```

See also

Densify geometries given an interval

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Interval between Vertices to add [number] <put parameter description here>

Default: 1.0

Outputs

Densified layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:densifygeometriesgivenaninterval', input, interval, output)
```

See also

Densify geometries

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Vertices to add [number] <put parameter description here>

Default: 1

Outputs

Densified layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:densifygeometries', input, vertices, output)
```

See also

Dissolve

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Dissolve all (do not use field) [boolean] <put parameter description here>

Default: *True*

Unique ID field [tablefield: any] Optional.

<put parameter description here>

Outputs

Dissolved [vector] <put output description here>

Console usage

```
processing.runalg('qgis:dissolve', input, dissolve_all, field, output)
```

See also

Eliminate sliver polygons

Description

<put algorithm description here>

Parameters

Input layer [**vector: polygon**] <put parameter description here>

Use current selection in input layer (works only if called from toolbox) [**boolean**]
<put parameter description here>

Default: *False*

Selection attribute [**tablefield: any**] <put parameter description here>

Comparison [**selection**] <put parameter description here>

Options:

- 0 — ==
- 1 — !=
- 2 — >
- 3 — >=
- 4 — <
- 5 — <=
- 6 — begins with
- 7 — contains

Default: *0*

Value [**string**] <put parameter description here>

Default: *0*

Merge selection with the neighbouring polygon with the [**selection**] <put parameter description here>

Options:

- 0 — Largest area
- 1 — Smallest Area
- 2 — Largest common boundary

Default: *0*

Outputs

Cleaned layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:eliminatesliverpolygons', input, keepselection, attribute, comparison, co
```

See also

Explode lines

Description

<put algorithm description here>

Parameters

Input layer [vector: line] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:explodelines', input, output)
```

See also

Extract nodes

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:extractnodes', input, output)
```

See also

Fill holes

Description

<put algorithm description here>

Parameters

Polygons [vector: any] <put parameter description here>

Max area [number] <put parameter description here>

Default: *100000*

Outputs

Results [vector] <put output description here>

Console usage

```
processing.runalg('qgis:fillholes', polygons, max_area, results)
```

See also

Fixed distance buffer

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Distance [number] <put parameter description here>

Default: *10.0*

Segments [number] <put parameter description here>

Default: *5*

Dissolve result [boolean] <put parameter description here>

Default: *False*

Outputs

Buffer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:fixeddistancebuffer', input, distance, segments, dissolve, output)
```

See also

Keep n biggest parts

Description

<put algorithm description here>

Parameters

Polygons [vector: polygon] <put parameter description here>

To keep [number] <put parameter description here>

Default: 1

Outputs

Results [vector] <put output description here>

Console usage

```
processing.runalg('qgis:keepnbiggestparts', polygons, to_keep, results)
```

See also

Lines to polygons

Description

<put algorithm description here>

Parameters

Input layer [vector: line] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:linestopolygons', input, output)
```

See also

Multipart to singleparts

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:multiparttosingleparts', input, output)
```

See also

Points displacement

Description

Moves overlapped points at small distance, that they all become visible. The result is very similar to the output of the “Point displacement” renderer but it is permanent.

Parameters

Input layer [vector: point] Layer with overlapped points.

Displacement distance [number] Desired displacement distance **NOTE:** displacement distance should be in same units as layer.

Default: *0.00015*

Horizontal distribution for two point case [boolean] Controls distribution direction in case of two overlapped points. If *True* points will be distributed horizontally, otherwise they will be distributed vertically.

Default: *True*

Outputs

Output layer [vector] The resulting layer with shifted overlapped points.

Console usage

```
processing.runalg('qgis:pointsdisplacement', input_layer, distance, horizontal, output_layer)
```


See also

Polygon centroids

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:polygoncentroids', input_layer, output_layer)
```

See also

Polygonize

Description

<put algorithm description here>

Parameters

Input layer [vector: line] <put parameter description here>

Keep table structure of line layer [boolean] <put parameter description here>

Default: *False*

Create geometry columns [boolean] <put parameter description here>

Default: *True*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:polygonize', input, fields, geometry, output)
```

See also

Polygons to lines

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:polygonstolines', input, output)
```

See also

Simplify geometries

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Tolerance [number] <put parameter description here>

Default: *1.0*

Outputs

Simplified layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:simplifygeometries', input, tolerance, output)
```

See also

Singleparts to multipart

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Unique ID field [tablefield: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:singlepartstomultipart', input, field, output)
```

See also

Variable distance buffer

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Distance field [tablefield: any] <put parameter description here>

Segments [number] <put parameter description here>

Default: 5

Dissolve result [boolean] <put parameter description here>

Default: *False*

Outputs

Buffer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:variabledistancebuffer', input, field, segments, dissolve, output)
```

See also

Voronoi polygons

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Buffer region [number] <put parameter description here>

Default: *0.0*

Outputs

Voronoi polygons [vector] <put output description here>

Console usage

```
processing.runalg('qgis:voronoipolygons', input, buffer, output)
```

See also

.

18.5.9 Vector overlay

Clip

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Clip layer [vector: any] <put parameter description here>

Outputs

Clipped [vector] <put output description here>

Console usage

```
processing.runalg('qgis:clip', input, overlay, output)
```

See also

Difference

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Difference layer [vector: any] <put parameter description here>

Outputs

Difference [vector] <put output description here>

Console usage

```
processing.runalg('qgis:difference', input, overlay, output)
```

See also

Intersection

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Intersect layer [vector: any] <put parameter description here>

Outputs

Intersection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:intersection', input, input2, output)
```

See also

Line intersections

Description

<put algorithm description here>

Parameters

Input layer [vector: line] <put parameter description here>

Intersect layer [vector: line] <put parameter description here>

Input unique ID field [tablefield: any] <put parameter description here>

Intersect unique ID field [tablefield: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:lineintersections', input_a, input_b, field_a, field_b, output)
```

See also

Symmetrical difference

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Difference layer [vector: any] <put parameter description here>

Outputs

Symmetrical difference [vector] <put output description here>

Console usage

```
processing.runalg('qgis:symmetricaldifference', input, overlay, output)
```

See also

Union

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Input layer 2 [vector: any] <put parameter description here>

Outputs

Union [vector] <put output description here>

Console usage

```
processing.runalg('qgis:union', input, input2, output)
```

See also

.

18.5.10 Vector selection

Extract by attribute

Description

<put algorithm description here>

Parameters

Input Layer [vector: any] <put parameter description here>

Selection attribute [tablefield: any] <put parameter description here>

Operator [selection] <put parameter description here>

Options:

- 0 — =
- 1 — !=
- 2 — >
- 3 — >=
- 4 — <
- 5 — <=
- 6 — begins with

- 7 — contains

Default: 0

Value [string] <put parameter description here>

Default: *(not set)*

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:extractbyattribute', input, field, operator, value, output)
```

See also

Extract by location

Description

<put algorithm description here>

Parameters

Layer to select from [vector: any] <put parameter description here>

Additional layer (intersection layer) [vector: any] <put parameter description here>

Include input features that touch the selection features [boolean] <put parameter description here>

Default: *False*

Include input features that overlap/cross the selection features [boolean] <put parameter description here>

Default: *False*

Include input features completely within the selection features [boolean] <put parameter description here>

Default: *False*

Outputs

Selection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:extractbylocation', input, intersect, touches, overlaps, within, output)
```


See also

Random extract

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: 0

Number/percentage of selected features [**number**] <put parameter description here>

Default: 10

Outputs

Selection [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:randomextract', input, method, number, output)
```

See also

Random extract within subsets

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

ID Field [**tablefield: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: 0

Number/percentage of selected features [number] <put parameter description here>

Default: *10*

Outputs

Selection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randomextractwithinsubsets', input, field, method, number, output)
```

See also

Random selection

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: *0*

Number/percentage of selected features [number] <put parameter description here>

Default: *10*

Outputs

Selection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randomselection', input, method, number)
```

See also

Random selection within subsets

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

ID Field [**tablefield: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: 0

Number/percentage of selected features [**number**] <put parameter description here>

Default: 10

Outputs

Selection [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:randomselectionwithinsubsets', input, field, method, number)
```

See also

Select by attribute

Description

Selects and saves as new layer all features from input layer that satisfy condition.

NOTE: algorithm is case-sensitive (“qgis” is different from “Qgis” and “QGIS”)

Parameters

Input Layer [**vector: any**] Layer to process.

Selection attribute [**tablefield: any**] Field on which perform the selection.

Operator [**selection**] Comparison operator.

Options:

- 0 — =
- 1 — !=
- 2 — >
- 3 — >=
- 4 — <
- 5 — <=
- 6 — begins with
- 7 — contains

Default: 0

Value [string] Value to compare.

Default: *(not set)*

Outputs

Output [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:selectbyattribute', input, field, operator, value, output)
```

See also

Select by expression

Description

<put algorithm description here>

Parameters

Input Layer [vector: any] <put parameter description here>

Expression [string] <put parameter description here>

Default: *(not set)*

Modify current selection by [selection] <put parameter description here>

Options:

- 0 — creating new selection
- 1 — adding to current selection
- 2 — removing from current selection

Default: 0

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:selectbyexpression', layername, expression, method)
```

See also

Select by location

Description

<put algorithm description here>

Parameters

Layer to select from [vector: any] <put parameter description here>

Additional layer (intersection layer) [vector: any] <put parameter description here>

Include input features that touch the selection features [boolean] <put parameter description here>

Default: *False*

Include input features that overlap/cross the selection features [boolean] <put parameter description here>

Default: *False*

Include input features completely within the selection features [boolean] <put parameter description here>

Default: *False*

Modify current selection by [selection] <put parameter description here>

Options:

- 0 — creating new selection
- 1 — adding to current selection
- 2 — removing from current selection

Default: *0*

Outputs

Selection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:selectbylocation', input, intersect, touches, overlaps, within, method)
```

See also

.

18.5.11 Vector table

Add autoincremental field

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:addautoincrementalfield', input, output)
```

See also

Add field to attributes table

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Field name [string] <put parameter description here>

Default: *(not set)*

Field type [selection] <put parameter description here>

Options:

- 0 — Integer
- 1 — Float
- 2 — String

Default: *0*

Field length [number] <put parameter description here>

Default: *10*

Field precision [number] <put parameter description here>

Default: *0*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:addfieldtoattributetable', input_layer, field_name, field_type, field_length)
```

See also

Advanced Python field calculator

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Result field name [string] <put parameter description here>

Default: *NewField*

Field type [selection] <put parameter description here>

Options:

- 0 — Integer
- 1 — Float
- 2 — String

Default: *0*

Field length [number] <put parameter description here>

Default: *10*

Field precision [number] <put parameter description here>

Default: *0*

Global expression [string] Optional.

<put parameter description here>

Default: *(not set)*

Formula [string] <put parameter description here>

Default: *value =*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:advancedpythonfieldcalculator', input_layer, field_name, field_type, field...
```

See also

Basic statistics for numeric fields

Description

<put algorithm description here>

Parameters

Input vector layer [**vector: any**] <put parameter description here>

Field to calculate statistics on [**tablefield: numeric**] <put parameter description here>

Outputs

Statistics for numeric field [**html**] <put output description here>

Coefficient of Variation [**number**] <put output description here>

Minimum value [**number**] <put output description here>

Maximum value [**number**] <put output description here>

Sum [**number**] <put output description here>

Mean value [**number**] <put output description here>

Count [**number**] <put output description here>

Range [**number**] <put output description here>

Median [**number**] <put output description here>

Number of unique values [**number**] <put output description here>

Standard deviation [**number**] <put output description here>

Console usage

```
processing.runalg('qgis:basicstatisticsfornumericfields', input_layer, field_name, output_html_fi...
```

See also

Basic statistics for text fields

Description

<put algorithm description here>

Parameters

Input vector layer [vector: any] <put parameter description here>

Field to calculate statistics on [tablefield: string] <put parameter description here>

Outputs

Statistics for text field [html] <put output description here>

Minimum length [number] <put output description here>

Maximum length [number] <put output description here>

Mean length [number] <put output description here>

Count [number] <put output description here>

Number of empty values [number] <put output description here>

Number of non-empty values [number] <put output description here>

Number of unique values [number] <put output description here>

Console usage

```
processing.runalg('qgis:basicstatisticsfortextfields', input_layer, field_name, output_html_file)
```

See also

Create equivalent numerical field

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Class field [tablefield: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:createequivalentnumericalfield', input, field, output)
```

See also

Delete column

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

Field to delete [**tablefield: any**] <put parameter description here>

Outputs

Output [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:deletecolumn', input, column, output)
```

See also

Export/Add geometry columns

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

Calculate using [**selection**] <put parameter description here>

Options:

- 0 — Layer CRS
- 1 — Project CRS
- 2 — Ellipsoidal

Default: 0

Outputs

Output layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:exportaddgeometrycolumns', input, calc_method, output)
```

See also

Field calculator

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Result field name [string] <put parameter description here>

Default: *(not set)*

Field type [selection] <put parameter description here>

Options:

- 0 — Float
- 1 — Integer
- 2 — String
- 3 — Date

Default: *0*

Field length [number] <put parameter description here>

Default: *10*

Field precision [number] <put parameter description here>

Default: *3*

Create new field [boolean] <put parameter description here>

Default: *True*

Formula [string] <put parameter description here>

Default: *(not set)*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:fieldcalculator', input_layer, field_name, field_type, field_length, field...
```

See also

List unique values

Description

Lists unique values of an attribute table field and counts their number.

Parameters

Input layer [**vector: any**] Layer to analyze.

Target field [**tablefield: any**] Field to analyze.

Outputs

Unique values [**html**] Analysis results in HTML format.

Total unique values [**number**] Total number of unique values in given field.

Unique values [**string**] List of all unique values in given field.

Console usage

```
processing.runalg('qgis:listuniquevalues', input_layer, field_name, output)
```

See also

Number of unique values in classes

Description

<put algorithm description here>

Parameters

input [**vector: any**] <put parameter description here>

class field [**tablefield: any**] <put parameter description here>

value field [**tablefield: any**] <put parameter description here>

Outputs

output [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:numberofuniquevaluesinclasses', input, class_field, value_field, output)
```

See also

Statistics by categories

Description

<put algorithm description here>

Parameters

Input vector layer [vector: any] <put parameter description here>

Field to calculate statistics on [tablefield: numeric] <put parameter description here>

Field with categories [tablefield: any] <put parameter description here>

Outputs

Statistics [table] <put output description here>

Console usage

```
processing.runalg('qgis:statisticsbycategories', input_layer, values_field_name, categories_field)
```

See also

Text to float

Description

<put algorithm description here>

Parameters

Input Layer [vector: any] <put parameter description here>

Text attribute to convert to float [tablefield: string] <put parameter description here>

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:texttofloat', input, field, output)
```

See also

.

18.6 R algorithm provider

R also called GNU S, is a strongly functional language and environment to statistically explore data sets, make many graphical displays of data from custom data sets

Note: Please remember that Processing contains only R scripts, so you need to install R by yourself and configure Processing properly.

18.6.1 Basic statistics

Frequency table

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Outputs

R Console Output [html] <put output description here>

Console usage

```
processing.runalg('r:frequencytable', layer, field, r_console_output)
```

See also

Teste Kolmogrov-Smirnov

Descrição

<ponha a descrição do algoritmo aqui>

Parâmetros

Camada [vector: qualquer] <ponha o parâmetro do algoritmo aqui>

Campo [tablefield: qualquer] <ponha o parâmetro do algoritmo aqui>

Saídas

Ficheiro de Saída da Linha de comandos R [html] <ponha a descrição de saída aqui>

Uso da Linha de Comandos

```
processing.runalg('r:kolmogrovsmirnovtest', layer, field, r_console_output)
```

Veja também

Summary statistics

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Outputs

R Console Output [html] <put output description here>

Console usage

```
processing.runalg('r:summarystatistics', layer, field, r_console_output)
```

See also

.

18.6.2 Home range

Characteristic hull method

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Outputs

Home_ranges [vector] <put output description here>

Console usage

```
processing.runalg('r:characteristichullmethod', layer, field, home_ranges)
```

See also

Kernel h ref

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Grid [number] <put parameter description here>

Default: *10.0*

Percentage [number] <put parameter description here>

Default: *10.0*

Folder [directory] Optional.

<put parameter description here>

Outputs

Home_ranges [vector] <put output description here>

Console usage

```
processing.runalg('r:kernelhref', layer, field, grid, percentage, folder, home_ranges)
```

See also

Minimum convex polygon

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Percentage [number] <put parameter description here>

Default: *10.0*

Field [tablefield: any] <put parameter description here>

Outputs

Home_ranges [vector] <put output description here>

Console usage

```
processing.runalg('r:minimumconvexpolygon', layer, percentage, field, home_ranges)
```

See also

Single-linkage cluster analysis

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Percentage [number] <put parameter description here>

Default: *10.0*

Outputs

R Plots [html] <put output description here>

Home_ranges [vector] <put output description here>

Console usage

```
processing.runalg('r:singlelinkageclusteranalysis', layer, field, percentage, rplots, home_ranges)
```

See also

.

18.6.3 Point pattern

F function

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Nsim [number] <put parameter description here>

Default: *10.0*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:ffunction', layer, nsim, rplots)
```

See also

G function

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Nsim [number] <put parameter description here>

Default: *10.0*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:gfunction', layer, nsim, rplots)
```

See also

Monte-Carlo spatial randomness

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Simulations [number] <put parameter description here>

Default: *100.0*

Optional plot name [string] <put parameter description here>

Default: *(not set)*

Outputs

R Plots [html] <put output description here>

R Console Output [html] <put output description here>

Console usage

```
processing.runalg('r:montecarlospatialrandomness', layer, simulations, optional_plot_name, rplots)
```

See also

Quadrat analysis

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Outputs

R Plots [html] <put output description here>

R Console Output [html] <put output description here>

Console usage

```
processing.runalg('r:quadratanalysis', layer, rplots, r_console_output)
```

See also

Random sampling grid

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Size [number] <put parameter description here>

Default: *10.0*

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('r:randomsamplinggrid', layer, size, output)
```

See also

Regular sampling grid

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Size [number] <put parameter description here>

Default: *10.0*

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('r:regularsamplinggrid', layer, size, output)
```

See also

Relative distribution (distance covariate)

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Covariate [vector: any] <put parameter description here>

Covariate name [string] <put parameter description here>

Default: *mandatory_covariate_name_(no_spaces)*

x label [string] <put parameter description here>

Default: *(not set)*

Plot name [string] <put parameter description here>

Default: *(not set)*

Legend position [string] <put parameter description here>

Default: *float*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:relativedistributiondistancecovariate', layer, covariate, covariate_name, x_
```

See also

Relative distribution (raster covariate)

Description

<put algorithm description here>

Parameters

points [vector: any] <put parameter description here>

covariate [raster] <put parameter description here>

covariate name [string] <put parameter description here>

Default: *mandatory_covariate_name_(no_spaces)*

x label [string] <put parameter description here>

Default: *(not set)*

plot name [string] <put parameter description here>

Default: *(not set)*

legend position [string] <put parameter description here>

Default: *float*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:relativedistributionrastercovariate', points, covariate, covariate_name, x_l
```

See also

Ripley - Rasson spatial domain

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('r:ripleyrassonspatialdomain', layer, output)
```

See also

.

18.6.4 Raster processing

Advanced raster histogram

Description

<put algorithm description here>

Parameters

Layer [raster] <put parameter description here>

Dens or Hist [string] <put parameter description here>

Default: *Hist*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:advancedrasterhistogram', layer, dens_or_hist, rplots)
```

See also

Raster histogram

Description

<put algorithm description here>

Parameters

Layer [raster] <put parameter description here>

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:rasterhistogram', layer, rplots)
```

See also

.

18.6.5 Vector processing

Histogram

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:histogram', layer, field, rplots)
```

See also

.

18.7 SAGA algorithm provider

SAGA (System for Automated Geoscientific Analyses) is a free, hybrid, cross-platform GIS software. SAGA provides many geoscientific methods which are bundled in so-called module libraries.

Note: Please remember that Processing contains only the interface description, so you need to install SAGA by yourself and configure Processing properly.

18.7.1 Geostatistics

Directional statistics for single grid

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Points [**vector: any**] Optional.

<put parameter description here>

Direction [**Degree**] [**number**] <put parameter description here>

Default: *0.0*

Tolerance [**Degree**] [**number**] <put parameter description here>

Default: *0.0*

Maximum Distance [**Cells**] [**number**] <put parameter description here>

Default: *0*

Distance Weighting [**selection**] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *1.0*

Outputs

Arithmetic Mean [raster] <put output description here>

Difference from Arithmetic Mean [raster] <put output description here>

Minimum [raster] <put output description here>

Maximum [raster] <put output description here>

Range [raster] <put output description here>

Variance [raster] <put output description here>

Standard Deviation [raster] <put output description here>

Mean less Standard Deviation [raster] <put output description here>

Mean plus Standard Deviation [raster] <put output description here>

Deviation from Arithmetic Mean [raster] <put output description here>

Percentile [raster] <put output description here>

Directional Statistics for Points [vector] <put output description here>

Console usage

```
processing.runalg('saga:directionalstatisticsforsinglegrid', grid, points, direction, tolerance, n)
```

See also

Fast representativeness

Description

<put algorithm description here>

Parameters

Input [raster] <put parameter description here>

Level of Generalisation [number] <put parameter description here>

Default: *16*

Outputs

Output [raster] <put output description here>

Output Lod [raster] <put output description here>

Output Seeds [raster] <put output description here>

Console usage

```
processing.runalg('saga:fastrepresentativeness', input, lod, result, result_lod, seeds)
```

See also

Geographically weighted multiple regression (points/grids)

Description

<put algorithm description here>

Parameters

Predictors [multipleinput: rasters] <put parameter description here>

Output of Regression Parameters [boolean] <put parameter description here>

Default: *True*

Points [vector: point] <put parameter description here>

Dependent Variable [tablefield: any] <put parameter description here>

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *1.0*

Search Range [selection] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [number] <put parameter description here>

Default: 100

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Outputs

Regression [raster] <put output description here>

Coefficient of Determination [raster] <put output description here>

Regression Parameters [raster] <put output description here>

Residuals [vector] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedmultipleregressionpointsgrids', predictors, parameter)
```

See also

Geographically weighted multiple regression (points)

Description

<put algorithm description here>

Parameters

Points [vector: any] <put parameter description here>

Dependent Variable [tablefield: any] <put parameter description here>

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1.0

Search Range [selection] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [number] <put parameter description here>

Default: 100

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Outputs

Regression [vector] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedmultipleregressionpoints', points, dependent, distan
```

See also

Geographically weighted multiple regression

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Dependent Variable [tablefield: any] <put parameter description here>

Target Grids [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1

Search Range [selection] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [number] <put parameter description here>

Default: 100

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Quality [raster] <put output description here>

Intercept [raster] <put output description here>

Quality [raster] <put output description here>

Intercept [raster] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedmultipleregression', points, dependent, target, dist
```

See also

Geographically weighted regression (points/grid)

Description

<put algorithm description here>

Parameters

Predictor [raster] <put parameter description here>

Points [vector: point] <put parameter description here>

Dependent Variable [tablefield: any] <put parameter description here>

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: True

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1.0

Search Range [selection] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [number] <put parameter description here>

Default: 0

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Outputs

Regression [raster] <put output description here>

Coefficient of Determination [raster] <put output description here>

Intercept [raster] <put output description here>

Slope [raster] <put output description here>

Residuals [vector] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedregressionpointsgrid', predictor, points, dependent,
```

See also

Geographically weighted regression

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Dependent Variable [**tablefield: any**] <put parameter description here>

Predictor [**tablefield: any**] <put parameter description here>

Target Grids [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Distance Weighting [**selection**] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [**number**] <put parameter description here>

Default: *0*

Inverse Distance Offset [**boolean**] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [**number**] <put parameter description here>

Default: *0.0*

Search Range [**selection**] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: *0*

Search Radius [**number**] <put parameter description here>

Default: *100*

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Quality [raster] <put output description here>

Intercept [raster] <put output description here>

Slope [raster] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedregression', points, dependent, predictor, target, 0)
```

See also

Global moran's i for grids

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Case of contiguity [selection] <put parameter description here>

Options:

- 0 — [0] Rook
 - 1 — [1] Queen
- Default: 0

Outputs

Result [table] <put output description here>

Console usage

```
processing.runalg('saga:globalmoransiforgrids', grid, contiguity, result)
```

See also

Minimum distance analysis

Description

Performs a complete distance analysis of a point layer:

- minimum distance of points
- maximum distance of points
- average distance of all the points
- standard deviation of the distance
- duplicated points

Parameters

Points [vector: point] Layer to analyze.

Outputs

Minimum Distance Analysis [table] The resulting table.

Console usage

```
processing.runalg('saga:minimumdistanceanalysis', points, table)
```

See also

Multi-band variation

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Radius [**Cells**] [**number**] <put parameter description here>

Default: *1*

Distance Weighting [**selection**] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [**number**] <put parameter description here>

Default: *1*

Inverse Distance Offset [**boolean**] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [**number**] <put parameter description here>

Default: *1.0*

Outputs

Mean Distance [**raster**] <put output description here>

Standard Deviation [**raster**] <put output description here>

Distance [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:multibandvariation', bands, radius, distance_weighting_weighting, distance_offset)
```

See also

Multiple regression analysis (grid/grids)

Description

<put algorithm description here>

Parameters

Dependent [**raster**] <put parameter description here>

Grids [**multipleinput: rasters**] <put parameter description here>

Grid Interpolation [**selection**] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Include X Coordinate [boolean] <put parameter description here>

Default: *True*

Include Y Coordinate [boolean] <put parameter description here>

Default: *True*

Method [selection] <put parameter description here>

Options:

- 0 — [0] include all
- 1 — [1] forward
- 2 — [2] backward
- 3 — [3] stepwise

Default: 0

P in [number] <put parameter description here>

Default: 5

P out [number] <put parameter description here>

Default: 5

Outputs

Regression [raster] <put output description here>

Residuals [raster] <put output description here>

Details: Coefficients [table] <put output description here>

Details: Model [table] <put output description here>

Details: Steps [table] <put output description here>

Console usage

```
processing.runalg('saga:multipleregressionanalysisgridgrids', dependent, grids, interpol, coord_x,
```

See also

Multiple regression analysis (points/grids)

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Shapes [**vector: any**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Grid Interpolation [**selection**] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Include X Coordinate [**boolean**] <put parameter description here>

Default: *True*

Include Y Coordinate [**boolean**] <put parameter description here>

Default: *True*

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] include all
- 1 — [1] forward
- 2 — [2] backward
- 3 — [3] stepwise

Default: 0

P in [**number**] <put parameter description here>

Default: 5

P out [**number**] <put parameter description here>

Default: 5

Outputs

Details: **Coefficients** [**table**] <put output description here>

Details: **Model** [**table**] <put output description here>

Details: **Steps** [**table**] <put output description here>

Residuals [**vector**] <put output description here>

Regression [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:multipleregressionanalysispointsgrids', grids, shapes, attribute, interpo
```

See also

Polynomial regression

Description

<put algorithm description here>

Parameters

Points [**vector: any**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Polynom [**selection**] <put parameter description here>

Options:

- 0 — [0] simple planar surface
- 1 — [1] bi-linear saddle
- 2 — [2] quadratic surface
- 3 — [3] cubic surface
- 4 — [4] user defined

Default: 0

Maximum X Order [**number**] <put parameter description here>

Default: 4

Maximum Y Order [**number**] <put parameter description here>

Default: 4

Maximum Total Order [**number**] <put parameter description here>

Default: 4

Trend Surface [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Output extent [**extent**] <put parameter description here>

Default: 0,1,0,1

Cellsize [**number**] <put parameter description here>

Default: 100.0

Outputs

Residuals [**vector**] <put output description here>

Grid [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:polynomialregression', points, attribute, polynom, xorder, yorder, torder)
```

See also

Radius of variance (grid)

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Standard Deviation [number] <put parameter description here>

Default: *1.0*

Maximum Search Radius (cells) [number] <put parameter description here>

Default: *20*

Type of Output [selection] <put parameter description here>

Options:

- 0 — [0] Cells
- 1 — [1] Map Units

Default: *0*

Outputs

Variance Radius [raster] <put output description here>

Console usage

```
processing.runalg('saga:radiusofvariancegrid', input, variance, radius, output, result)
```

See also

Regression analysis

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Shapes [vector: any] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Grid Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Regression Function [selection] <put parameter description here>

Options:

- 0 — [0] $Y = a + b * X$ (linear)
- 1 — [1] $Y = a + b / X$
- 2 — [2] $Y = a / (b - X)$
- 3 — [3] $Y = a * X^b$ (power)
- 4 — [4] $Y = a e^{(b * X)}$ (exponential)
- 5 — [5] $Y = a + b * \ln(X)$ (logarithmic)

Default: 0

Outputs

Regression [raster] <put output description here>

Residuals [vector] <put output description here>

Console usage

```
processing.runalg('saga:regressionanalysis', grid, shapes, attribute, interpol, method, regression)
```

See also

Representativeness

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Radius (Cells) [number] <put parameter description here>

Default: *10*

Exponent [number] <put parameter description here>

Default: *1*

Outputs

Representativeness [raster] <put output description here>

Console usage

```
processing.runalg('saga:representativeness', input, radius, exponent, result)
```

See also

Residual analysis

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Radius (Cells) [number] <put parameter description here>

Default: *7*

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *1.0*

Outputs

Mean Value [raster] <put output description here>
Difference from Mean Value [raster] <put output description here>
Standard Deviation [raster] <put output description here>
Value Range [raster] <put output description here>
Minimum Value [raster] <put output description here>
Maximum Value [raster] <put output description here>
Deviation from Mean Value [raster] <put output description here>
Percentile [raster] <put output description here>

Console usage

```
processing.runalg('saga:residualanalysis', grid, radius, distance_weighting_weighting, distance_w
```

See also

Spatial point pattern analysis

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>
Vertex Distance [Degree] [number] <put parameter description here>
Default: 5

Outputs

Mean Centre [vector] <put output description here>
Standard Distance [vector] <put output description here>
Bounding Box [vector] <put output description here>

Console usage

```
processing.runalg('saga:spatialpointpatternanalysis', points, step, centre, stddist, bbox)
```

See also

Statistics for grids

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Outputs

Arithmetic Mean [**raster**] <put output description here>

Minimum [**raster**] <put output description here>

Maximum [**raster**] <put output description here>

Variance [**raster**] <put output description here>

Standard Deviation [**raster**] <put output description here>

Mean less Standard Deviation [**raster**] <put output description here>

Mean plus Standard Deviation [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:statisticsforgrids', grids, mean, min, max, var, stddev, stddevlo, stddevhi)
```

See also

Variogram cloud

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Maximum Distance [**number**] <put parameter description here>

Default: *0.0*

Skip Number [**number**] <put parameter description here>

Default: *1*

Outputs

Variogram Cloud [**table**] <put output description here>

Console usage

```
processing.runalg('saga:variogramcloud', points, field, distmax, nskip, result)
```

See also

Variogram surface

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Number of Distance Classes [number] <put parameter description here>

Default: 10

Skip Number [number] <put parameter description here>

Default: 1

Outputs

Number of Pairs [raster] <put output description here>

Variogram Surface [raster] <put output description here>

Covariance Surface [raster] <put output description here>

Console usage

```
processing.runalg('saga:variogramsurface', points, field, distcount, nskip, count, variance, covar
```

See also

Zonal grid statistics

Description

<put algorithm description here>

Parameters

Zone Grid [raster] <put parameter description here>

Categorical Grids [multipleinput: rasters] Optional.

<put parameter description here>

Grids to analyse [multipleinput: rasters] Optional.

<put parameter description here>

Aspect [raster] Optional.

<put parameter description here>

Short Field Names [boolean] <put parameter description here>

Default: *True*

Outputs

Zonal Statistics [table] <put output description here>

Console usage

```
processing.runalg('saga:zonalgridstatistics', zones, catlist, statlist, aspect, shortnames, outta
```

See also

.

18.7.2 Grid analysis

Accumulated cost (anisotropic)

Description

<put algorithm description here>

Parameters

Cost Grid [raster] <put parameter description here>

Direction of max cost [raster] <put parameter description here>

Destination Points [raster] <put parameter description here>

k factor [number] <put parameter description here>

Default: *1*

Threshold for different route [number] <put parameter description here>

Default: *0*

Outputs

Accumulated Cost [raster] <put output description here>

Console usage

```
processing.runalg('saga:accumulatedcostanisotropic', cost, direction, points, k, threshold, accco
```

See also

Accumulated cost (isotropic)

Description

<put algorithm description here>

Parameters

Cost Grid [raster] <put parameter description here>

Destination Points [raster] <put parameter description here>

Threshold for different route [number] <put parameter description here>

Default: 0.0

Outputs

Accumulated Cost [raster] <put output description here>

Closest Point [raster] <put output description here>

Console usage

```
processing.runalg('saga:accumulatedcostisotropic', cost, points, threshold, acccost, closestpt)
```

See also

Aggregation index

Description

<put algorithm description here>

Parameters

Input Grid [raster] <put parameter description here>

Max. Number of Classes [number] <put parameter description here>

Default: 5

Outputs

Result [table] <put output description here>

Console usage

```
processing.runalg('saga:aggregationindex', input, maxnumclass, result)
```

See also

Analytical hierarchy process

Description

<put algorithm description here>

Parameters

Input Grids [**multipleinput: rasters**] <put parameter description here>

Pairwise Comparisons Table [**table**] <put parameter description here>

Outputs

Output Grid [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:analyticalhierarchyprocess', grids, table, output)
```

See also

Cross-classification and tabulation

Description

<put algorithm description here>

Parameters

Input Grid 1 [**raster**] <put parameter description here>

Input Grid 2 [**raster**] <put parameter description here>

Max. Number of Classes [**number**] <put parameter description here>

Default: 5

Outputs

Cross-Classification Grid [**raster**] <put output description here>

Cross-Tabulation Table [**table**] <put output description here>

Console usage

```
processing.runalg('saga:crossclassificationandtabulation', input, input2, maxnumclass, resultgrid)
```

See also

Fragmentation (alternative)

Description

<put algorithm description here>

Parameters

Classification [raster] <put parameter description here>

Class Identifier [number] <put parameter description here>

Default: *1*

Neighborhood Min [number] <put parameter description here>

Default: *1*

Neighborhood Max [number] <put parameter description here>

Default: *1*

Level Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] average
- 1 — [1] multiplicative

Default: *0*

Add Border [boolean] <put parameter description here>

Default: *True*

Connectivity Weighting [number] <put parameter description here>

Default: *1.1*

Minimum Density [Percent] [number] <put parameter description here>

Default: *10*

Minimum Density for Interior Forest [Percent] [number] <put parameter description here>

Default: *99*

Search Distance Increment [number] <put parameter description here>

Default: *0.0*

Density from Neighbourhood [boolean] <put parameter description here>

Default: *True*

Outputs

Density [Percent] [raster] <put output description here>

Connectivity [Percent] [raster] <put output description here>

Fragmentation [raster] <put output description here>

Summary [table] <put output description here>

Console usage

```
processing.runalg('saga:fragmentationalternative', classes, class, neighborhood_min, neighborhood_max)
```

See also

Fragmentation classes from density and connectivity

Description

<put algorithm description here>

Parameters

Density [Percent] [raster] <put parameter description here>

Connectivity [Percent] [raster] <put parameter description here>

Add Border [boolean] <put parameter description here>

Default: *True*

Connectivity Weighting [number] <put parameter description here>

Default: *0*

Minimum Density [Percent] [number] <put parameter description here>

Default: *10*

Minimum Density for Interior Forest [Percent] [number] <put parameter description here>

Default: *99*

Outputs

Fragmentation [raster] <put output description here>

Console usage

```
processing.runalg('saga:fragmentationclassesfromdensityandconnectivity', density, connectivity, border)
```

See also

Fragmentation (standard)

Description

<put algorithm description here>

Parameters

Classification [raster] <put parameter description here>

Class Identifier [number] <put parameter description here>

Default: 1

Neighborhood Min [number] <put parameter description here>

Default: 1

Neighborhood Max [number] <put parameter description here>

Default: 3

Level Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] average
- 1 — [1] multiplicative

Default: 0

Add Border [boolean] <put parameter description here>

Default: *True*

Connectivity Weighting [number] <put parameter description here>

Default: 1.1

Minimum Density [Percent] [number] <put parameter description here>

Default: 10

Minimum Density for Interior Forest [Percent] [number] <put parameter description here>

Default: 99

Neighborhood Type [selection] <put parameter description here>

Options:

- 0 — [0] square
- 1 — [1] circle

Default: 0

Include diagonal neighbour relations [boolean] <put parameter description here>

Default: *True*

Outputs

Density [Percent] [raster] <put output description here>

Connectivity [Percent] [raster] <put output description here>

Fragmentation [raster] <put output description here>

Summary [table] <put output description here>

Console usage

```
processing.runalg('saga:fragmentationstandard', classes, class, neighborhood_min, neighborhood_max)
```

See also

Layer of extreme value

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] Maximum
- 1 — [1] Minimum

Default: 0

Outputs

Result [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:layerofextremevalue', grids, criteria, result)
```

See also

Least cost paths

Description

<put algorithm description here>

Parameters

Source Point (s) [**vector: point**] <put parameter description here>

Accumulated cost [**raster**] <put parameter description here>

Values [**multipleinput: rasters**] Optional.

<put parameter description here>

Outputs

Profile (points) [**vector**] <put output description here>

Profile (lines) [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:leastcostpaths', source, dem, values, points, line)
```

See also

Ordered Weighted Averaging

Description

<put algorithm description here>

Parameters

Input Grids [**multipleinput: rasters**] <put parameter description here>

Weights [**fixedtable**] <put parameter description here>

Outputs

Output Grid [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:orderedweightedaveraging', grids, weights, output)
```

See also

Pattern analysis

Description

<put algorithm description here>

Parameters

Input Grid [**raster**] <put parameter description here>

Size of Analysis Window [**selection**] <put parameter description here>

Options:

- 0 — [0] 3 X 3
- 1 — [1] 5 X 5
- 2 — [2] 7 X 7

Default: 0

Max. Number of Classes [**number**] <put parameter description here>

Default: 0

Outputs

Relative Richness [raster] <put output description here>

Diversity [raster] <put output description here>

Dominance [raster] <put output description here>

Fragmentation [raster] <put output description here>

Number of Different Classes [raster] <put output description here>

Center Versus Neighbours [raster] <put output description here>

Console usage

```
processing.runalg('saga:patternanalysis', input, winsize, maxnumclass, relative, diversity, domin
```

See also

Soil texture classification

Description

<put algorithm description here>

Parameters

Sand [raster] Optional.

<put parameter description here>

Silt [raster] Optional.

<put parameter description here>

Clay [raster] Optional.

<put parameter description here>

Outputs

Soil Texture [raster] <put output description here>

Sum [raster] <put output description here>

Console usage

```
processing.runalg('saga:soiltextureclassification', sand, silt, clay, texture, sum)
```

See also

.

18.7.3 Grid calculus

Function

Description

<put algorithm description here>

Parameters

xmin [number] <put parameter description here>

Default: 0.0

xmax [number] <put parameter description here>

Default: 0.0

ymin [number] <put parameter description here>

Default: 0.0

ymax [number] <put parameter description here>

Default: 0.0

Formula [string] <put parameter description here>

Default: (not set)

Outputs

Function [raster] <put output description here>

Console usage

```
processing.runalg('saga:function', xmin, xmax, ymin, ymax, formul, result)
```

See also

Fuzzify

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

A [number] <put parameter description here>

Default: 0.0

B [number] <put parameter description here>

Default: 0.0

C [number] <put parameter description here>

Default: *0.0*

D [number] <put parameter description here>

Default: *0.0*

Membership Function Type [selection] <put parameter description here>

Options:

- 0 — [0] linear
- 1 — [1] sigmoidal
- 2 — [2] j-shaped

Default: *0*

Adjust to Grid [boolean] <put parameter description here>

Default: *True*

Outputs

Fuzzified Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:fuzzify', input, a, b, c, d, type, autofit, output)
```

See also

Fuzzy intersection (and)

Description

<put algorithm description here>

Parameters

Grids [multipleinput: rasters] <put parameter description here>

Operator Type [selection] <put parameter description here>

Options:

- 0 — [0] min(a, b) (non-interactive)
- 1 — [1] a * b
- 2 — [2] max(0, a + b - 1)

Default: *0*

Outputs

Intersection [raster] <put output description here>

Console usage

```
processing.runalg('saga:fuzzyintersectionand', grids, type, and)
```

See also

Fuzzy union (or)

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Operator Type [**selection**] <put parameter description here>

Options:

- 0 — [0] $\max(a, b)$ (non-interactive)
- 1 — [1] $a + b - a * b$
- 2 — [2] $\min(1, a + b)$

Default: 0

Outputs

Union [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:fuzzyunionor', grids, type, or)
```

See also

Geometric figures

Description

Draws simple geometric figures.

Parameters

Cell Count [**number**] Number of cells to use.

Default: 0

Cell Size [**number**] Size of the single cell.

Default: 0

Figure [selection] Type of the figure.

Options:

- 0 — [0] Cone (up)
- 1 — [1] Cone (down)
- 2 — [2] Plane

Default: 0

Direction of Plane [Degree] [number] Rotation factor in degrees.

Default: 0

Outputs

Result [raster] The resulting layer.

Console usage

```
processing.runalg('saga:geometricfigures', cell_count, cell_size, figure, plane, result)
```

See also

Gradient vector from cartesian to polar coordinates

Description

<put algorithm description here>

Parameters

X Component [raster] <put parameter description here>

Y Component [raster] <put parameter description here>

Polar Angle Units [selection] <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: 0

Polar Coordinate System [selection] <put parameter description here>

Options:

- 0 — [0] mathematical
- 1 — [1] geographical
- 2 — [2] user defined

Default: 0

User defined Zero Direction [number] <put parameter description here>

Default: 0.0

User defined Orientation [selection] <put parameter description here>

Options:

- 0 — [0] clockwise
- 1 — [1] counterclockwise

Default: 0

Outputs

Direction [raster] <put output description here>

Length [raster] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorfromcartesiantopolarcoordinates', dx, dy, units, system, sy
```

See also

Gradient vector from polar to cartesian coordinates

Description

<put algorithm description here>

Parameters

Direction [raster] <put parameter description here>

Length [raster] <put parameter description here>

Polar Angle Units [selection] <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: 0

Polar Coordinate System [selection] <put parameter description here>

Options:

- 0 — [0] mathematical
- 1 — [1] geographical
- 2 — [2] user defined

Default: 0

User defined Zero Direction [number] <put parameter description here>

Default: 0.0

User defined Orientation [selection] <put parameter description here>

Options:

- 0 — [0] clockwise

- 1 — [1] counterclockwise

Default: 0

Outputs

X Component [raster] <put output description here>

Y Component [raster] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorfrompolarcoordinates', dir, len, units, system, ...)
```

See also

Grid difference

Description

Creates a new grid layer as the result of the difference between two other grid layers.

Parameters

A [raster] First layer.

B [raster] Second layer.

Outputs

Difference (A - B) [raster] The resulting layer.

Console usage

```
processing.runalg('saga:griddifference', a, b, c)
```

See also

Grid division

Description

Creates a new grid layer as the result of the division between two other grid layers.

Parameters

Dividend [raster] First layer.

Divisor [raster] Second layer.

Outputs

Quotient [raster] The resulting layer.

Console usage

```
processing.runalg('saga:griddivision', a, b, c)
```

See also

Grid normalisation

Description

Normalises the grid values according to minimum and maximum values chosen.

Parameters

Grid [raster] Grid to normalize.

Target Range (min) [number] Minimum value.

Default: *0*

Target Range (max) [number] Maximum value.

Default: *1*

Outputs

Normalised Grid [raster] The resulting layer.

Console usage

```
processing.runalg('saga:gridnormalisation', input, range_min, range_max, output)
```

See also

Grids product

Description

<put algorithm description here>

Parameters

Grids [multipleinput: rasters] <put parameter description here>

Outputs

Product [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridsproduct', grids, result)
```

See also

Grids sum

Description

Creates a new grid layer as the result of the sum of two or more grid layers.

Parameters

Grids [multipleinput: rasters] Grid layers to sum

Outputs

Sum [raster] The resulting layer.

Console usage

```
processing.runalg('saga:gridssum', grids, result)
```

See also

Grid standardisation

Description

Standardises the grid layer values.

Parameters

Grid [raster] Grid to process.

Stretch Factor [number] stretching factor.

Default: *1.0*

Outputs

Standardised Grid [raster] The resulting layer.

Console usage

```
processing.runalg('saga:gridstandardisation', input, stretch, output)
```

See also

Grid volume

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Count Only Above Base Level
- 1 — [1] Count Only Below Base Level
- 2 — [2] Subtract Volumes Below Base Level
- 3 — [3] Add Volumes Below Base Level

Default: 0

Base Level [number] <put parameter description here>

Default: 0.0

Outputs

Console usage

```
processing.runalg('saga:gridvolume', grid, method, level)
```

See also

Metric conversions

Description

Performs numerical conversions of the grid values.

Parameters

Grid [raster] Grid to process.

Conversion [selection] Conversion type.

Options:

- 0 — [0] radians to degree
- 1 — [1] degree to radians
- 2 — [2] Celsius to Fahrenheit
- 3 — [3] Fahrenheit to Celsius

Default: 0

Outputs

Converted Grid [raster] The resulting layer.

Console usage

```
processing.runalg('saga:metricconversions', grid, conversion, conv)
```

See also

Polynomial trend from grids

Description

<put algorithm description here>

Parameters

Dependent Variables [multipleinput: rasters] <put parameter description here>

Independent Variable (per Grid and Cell) [multipleinput: rasters] Optional.

<put parameter description here>

Independent Variable (per Grid) [fixedtable] <put parameter description here>

Type of Approximated Function [selection] <put parameter description here>

Options:

- 0 — [0] first order polynom (linear regression)
- 1 — [1] second order polynom
- 2 — [2] third order polynom
- 3 — [3] fourth order polynom
- 4 — [4] fifth order polynom

Default: 0

Outputs

Polynomial Coefficients [raster] <put output description here>

Coefficient of Determination [raster] <put output description here>

Console usage

```
processing.runalg('saga:polynomialtrendfromgrids', grids, y_grids, y_table, polynom, parms, quality)
```

See also

Random field

Description

Generates a random grid layer.

Parameters

Width (Cells) [number] Width of the layer in cells.

Default: *100*

Height (Cells) [number] Height of the layer in cells.

Default: *100*

Cellsize [number] Cell size to use.

Default: *100.0*

West [number] West coordinate of the bottom-left corner of the grid.

Default: *0.0*

South [number] South coordinate of the bottom-left corner of the grid.

Default: *0.0*

Method [selection] Statistical method used for the calculation.

Options:

- 0 — [0] Uniform
- 1 — [1] Gaussian

Default: *0*

Range Min [number] Minimum cell value to use.

Default: *0.0*

Range Max [number] Maximum cell value to use.

Default: *1.0*

Arithmetic Mean [number] Mean of all the cell values to use.

Default: *0.0*

Standard Deviation [number] Standard deviation of all the cell values to use.

Default: *1.0*

Outputs

Random Field [raster] The resulting layer.

Console usage

```
processing.runalg('saga:randomfield', nx, ny, cellsize, xmin, ymin, method, range_min, range_max,
```


See also

Random terrain generation

Description

<put algorithm description here>

Parameters

Radius (cells) [number] <put parameter description here>

Default: *10*

Iterations [number] <put parameter description here>

Default: *10*

Target Dimensions [selection] <put parameter description here>

Options:

- 0 — [0] User defined

Default: *0*

Grid Size [number] <put parameter description here>

Default: *1.0*

Cols [number] <put parameter description here>

Default: *100*

Rows [number] <put parameter description here>

Default: *100*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:randomterraingeneration', radius, iterations, target_type, user_cell_size)
```

See also

Raster calculator

Description

<put algorithm description here>

Parameters

Main input layer [raster] <put parameter description here>

Additional layers [multipleinput: rasters] Optional.

<put parameter description here>

Formula [string] <put parameter description here>

Default: *(not set)*

Outputs

Result [raster] <put output description here>

Console usage

```
processing.runalg('saga:rastercalculator', grids, xgrids, formula, result)
```

See also

.

18.7.4 Grid filter

Dtm filter (slope-based)

Description

<put algorithm description here>

Parameters

Grid to filter [raster] <put parameter description here>

Search Radius [number] <put parameter description here>

Default: 2

Approx. Terrain Slope [number] <put parameter description here>

Default: 30.0

Use Confidence Interval [boolean] <put parameter description here>

Default: *True*

Outputs

Bare Earth [raster] <put output description here>

Removed Objects [raster] <put output description here>

Console usage

```
processing.runalg('saga:dtmfilterslopebased', input, radius, terrainslope, stddev, ground, nongro
```

See also

Filter clumps

Description

<put algorithm description here>

Parameters

Input Grid [raster] <put parameter description here>

Min. Size [number] <put parameter description here>

Default: 10

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:filterclumps', grid, threshold, output)
```

See also

Gaussian filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Standard Deviation [number] <put parameter description here>

Default: 1

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Search Radius [number] <put parameter description here>

Default: 3

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gaussianfilter', input, sigma, mode, radius, result)
```

See also

Laplacian filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] standard kernel 1
- 1 — [1] standard kernel 2
- 2 — [2] Standard kernel 3
- 3 — [3] user defined kernel

Default: 0

Standard Deviation (Percent of Radius) [number] <put parameter description here>

Default: 0

Radius [number] <put parameter description here>

Default: 1

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] square
- 1 — [1] circle

Default: 0

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:laplacianfilter', input, method, sigma, radius, mode, result)
```

See also

Majority filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Radius [number] <put parameter description here>

Default: 1

Threshold [Percent] [number] <put parameter description here>

Default: 0

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:majorityfilter', input, mode, radius, threshold, result)
```

See also

Morphological filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Radius [number] <put parameter description here>

Default: *1*

Method [selection] <put parameter description here>

Options:

- 0 — [0] Dilation
- 1 — [1] Erosion
- 2 — [2] Opening
- 3 — [3] Closing

Default: *0*

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:morphologicalfilter', input, mode, radius, method, result)
```

See also

Multi direction lee filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Estimated Noise (absolute) [number] <put parameter description here>

Default: *1.0*

Estimated Noise (relative) [number] <put parameter description here>

Default: *1.0*

Weighted [boolean] <put parameter description here>

Default: *True*

Method [selection] <put parameter description here>

Options:

- 0 — [0] noise variance given as absolute value
- 1 — [1] noise variance given relative to mean standard deviation
- 2 — [2] original calculation (Ringeler)

Default: *0*

Outputs

Filtered Grid [raster] <put output description here>

Minimum Standard Deviation [raster] <put output description here>

Direction of Minimum Standard Deviation [raster] <put output description here>

Console usage

```
processing.runalg('saga:multidirectionleefilter', input, noise_abs, noise_rel, weighted, method, ...)
```

See also

Rank filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Radius [number] <put parameter description here>

Default: 1

Rank [Percent] [number] <put parameter description here>

Default: 50

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:rankfilter', input, mode, radius, rank, result)
```

See also

Simple filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Filter [selection] <put parameter description here>

Options:

- 0 — [0] Smooth
- 1 — [1] Sharpen
- 2 — [2] Edge

Default: 0

Radius [number] <put parameter description here>

Default: 2

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:simplefilter', input, mode, method, radius, result)
```

See also

User defined filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Filter Matrix [table] Optional.

<put parameter description here>

Default Filter Matrix (3x3) [fixedtable] <put parameter description here>

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:userdefinedfilter', input, filter, filter_3x3, result)
```

See also

.

18.7.5 Grid gridding

Inverse distance weighted

Description

Inverse distance grid interpolation from irregular distributed points.

Parameters

Points [**vector:** **point**] <put parameter description here>

Attribute [**tablefield:** **any**] <put parameter description here>

Target Grid [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Distance Weighting [**selection**] <put parameter description here>

Options:

- 0 — [0] inverse distance to a power
- 1 — [1] linearly decreasing within search radius
- 2 — [2] exponential weighting scheme
- 3 — [3] gaussian weighting scheme

Default: 0

Inverse Distance Power [**number**] <put parameter description here>

Default: 2

Exponential and Gaussian Weighting Bandwidth [**number**] <put parameter description here>

Default: 1

Search Range [**selection**] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [**number**] <put parameter description here>

Default: 100.0

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of points
- 1 — [1] all points

Default: 0

Maximum Number of Points [number] <put parameter description here>

Default: 10

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:inversedistanceweighted', shapes, field, target, weighting, power, bandwi
```

See also

Kernel density estimation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Weight [tablefield: any] <put parameter description here>

Radius [number] <put parameter description here>

Default: 10

Kernel [selection] <put parameter description here>

Options:

- 0 — [0] quartic kernel
- 1 — [1] gaussian kernel

Default: 0

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:kerneldensityestimation', points, population, radius, kernel, target, out)
```

See also

Modified quadratic shepard

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Quadratic Neighbors [number] <put parameter description here>

Default: 13

Weighting Neighbors [number] <put parameter description here>

Default: 19

Left [number] <put parameter description here>

Default: 0.0

Right [number] <put parameter description here>

Default: 0.0

Bottom [number] <put parameter description here>

Default: *0.0*

Top [number] <put parameter description here>

Default: *0.0*

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:modifiedquadraticshepard', shapes, field, target, quadratic_neighbors, wei
```

See also

Natural neighbour

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Sibson [boolean] <put parameter description here>

Default: *True*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:naturalneighbour', shapes, field, target, sibson, output_extent, user_size)
```

See also

Nearest neighbour

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:nearestneighbour', shapes, field, target, output_extent, user_size, user_size)
```

See also

Shapes to grid

Description

<put algorithm description here>

Parameters

Shapes [vector: any] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Method for Multiple Values [selection] <put parameter description here>

Options:

- 0 — [0] first
- 1 — [1] last
- 2 — [2] minimum
- 3 — [3] maximum
- 4 — [4] mean

Default: 0

Method for Lines [selection] <put parameter description here>

Options:

- 0 — [0] thin
- 1 — [1] thick

Default: 0

Preferred Target Grid Type [selection] <put parameter description here>

Options:

- 0 — [0] Integer (1 byte)
- 1 — [1] Integer (2 byte)
- 2 — [2] Integer (4 byte)
- 3 — [3] Floating Point (4 byte)
- 4 — [4] Floating Point (8 byte)

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:shapestogrid', input, field, multiple, line_type, grid_type, output_extent)
```

See also

Triangulation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:triangulation', shapes, field, target, output_extent, user_size, user_grid)
```

See also

.

18.7.6 Grid spline

B-spline approximation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Resolution [number] <put parameter description here>

Default: 1.0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:bsplineapproximation', shapes, field, target, level, output_extent, user_
```

See also

Cubic spline approximation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Minimal Number of Points [number] <put parameter description here>

Default: 3

Maximal Number of Points [number] <put parameter description here>

Default: 20

Points per Square [number] <put parameter description here>

Default: 5

Tolerance [number] <put parameter description here>

Default: 140.0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:cubicsplineapproximation', shapes, field, target, npmin, npmax, nppc, k, c
```

See also

Multilevel b-spline interpolation (from grid)

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Method [selection] <put parameter description here>

Options:

- 0 — [0] without B-spline refinement
- 1 — [1] with B-spline refinement

Default: 0

Threshold Error [number] <put parameter description here>

Default: 0.0001

Maximum Level [number] <put parameter description here>

Default: 11.0

Data Type [selection] <put parameter description here>

Options:

- 0 — [0] same as input grid
- 1 — [1] floating point

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:multilevelbsplineinterpolationfromgrid', gridpoints, target, method, epsi
```

See also

Multilevel b-spline interpolation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Method [selection] <put parameter description here>

Options:

- 0 — [0] without B-spline refinement
- 1 — [1] with B-spline refinement

Default: 0

Threshold Error [number] <put parameter description here>

Default: 0.0001

Maximum Level [number] <put parameter description here>

Default: 11.0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:multilevelbsplineinterpolation', shapes, field, target, method, epsilon, .
```

See also

Thin plate spline (global)

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Regularisation [number] <put parameter description here>

Default: *0.0*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:thinplatesplineglobal', shapes, field, target, regul, output_extent, user.
```

See also

Thin plate spline (local)

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Regularisation [number] <put parameter description here>

Default: 0.0001

Search Radius [number] <put parameter description here>

Default: 100.0

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Points Selection [selection] <put parameter description here>

Options:

- 0 — [0] all points in search radius
- 1 — [1] maximum number of points

Default: 0

Maximum Number of Points [number] <put parameter description here>

Default: 10

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:thinplatesplinelocal', shapes, field, target, regul, radius, mode, select,
```

See also

Thin plate spline (tin)

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Regularisation [number] <put parameter description here>

Default: 0.0

Neighbourhood [selection] <put parameter description here>

Options:

- 0 — [0] immediate
- 1 — [1] level 1
- 2 — [2] level 2

Default: 0

Add Frame [boolean] <put parameter description here>

Default: True

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:thinplatesplinetin', shapes, field, target, regul, level, frame, output_e
```

See also

.

18.7.7 Grid tools

Aggregate

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Aggregation Size [**number**] <put parameter description here>

Default: 3

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] Sum
- 1 — [1] Min
- 2 — [2] Max

Default: 0

Outputs

Console usage

```
processing.runalg('saga:aggregate', input, size, method)
```

See also

Change grid values

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Replace Condition [**selection**] <put parameter description here>

Options:

- 0 — [0] Grid value equals low value
- 1 — [1] Low value < grid value < high value

- 2 — [2] Low value \leq grid value $<$ high value

Default: 0

Lookup Table [fixedtable] <put parameter description here>

Outputs

Changed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:changegridvalues', grid_in, method, lookup, grid_out)
```

See also

Close gaps

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Mask [raster] Optional.

<put parameter description here>

Tension Threshold [number] <put parameter description here>

Default: 0.1

Outputs

Changed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:closegaps', input, mask, threshold, result)
```

See also

Close gaps with spline

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Mask [raster] Optional.

<put parameter description here>

Only Process Gaps with Less Cells [number] <put parameter description here>

Default: *0*

Maximum Points [number] <put parameter description here>

Default: *1000*

Number of Points for Local Interpolation [number] <put parameter description here>

Default: *10*

Extended Neighbourhood [boolean] <put parameter description here>

Default: *True*

Neighbourhood [selection] <put parameter description here>

Options:

- 0 — [0] Neumann
- 1 — [1] Moore

Default: *0*

Radius (Cells) [number] <put parameter description here>

Default: *0*

Relaxation [number] <put parameter description here>

Default: *0.0*

Outputs

Closed Gaps Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:closegapswithspline', grid, mask, maxgapcells, maxpoints, localpoints, ex
```

See also

Close one cell gaps

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Outputs

Changed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:closeonecellgaps', input, result)
```

See also

Convert data storage type

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Data storage type [selection] <put parameter description here>

Options:

- 0 — [0] bit
- 1 — [1] unsigned 1 byte integer
- 2 — [2] signed 1 byte integer
- 3 — [3] unsigned 2 byte integer
- 4 — [4] signed 2 byte integer
- 5 — [5] unsigned 4 byte integer
- 6 — [6] signed 4 byte integer
- 7 — [7] 4 byte floating point number
- 8 — [8] 8 byte floating point number

Default: 0

Outputs

Converted Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:convertdatastoragetype', input, type, output)
```

See also

Crop to data

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Outputs

Cropped layer [raster] <put output description here>

Console usage

```
processing.runalg('saga:croptodata', input, output)
```

See also

Grid buffer

Description

<put algorithm description here>

Parameters

Features Grid [raster] <put parameter description here>

Distance [number] <put parameter description here>

Default: *1000*

Buffer Distance [selection] <put parameter description here>

Options:

- 0 — [0] Fixed
- 1 — [1] Cell value

Default: *0*

Outputs

Buffer Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridbuffer', features, dist, buffertype, buffer)
```

See also

Grid masking

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Mask [raster] <put parameter description here>

Outputs

Masked Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridmasking', grid, mask, masked)
```

See also

Grid orientation

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Copy
- 1 — [1] Flip
- 2 — [2] Mirror
- 3 — [3] Invert

Default: 0

Outputs

Changed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridorientation', input, method, result)
```

See also

Grid proximity buffer

Description

<put algorithm description here>

Parameters

Source Grid [raster] <put parameter description here>

Buffer distance [number] <put parameter description here>

Default: *500.0*

Equidistance [number] <put parameter description here>

Default: *100.0*

Outputs

Distance Grid [raster] <put output description here>

Allocation Grid [raster] <put output description here>

Buffer Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridproximitybuffer', source, dist, ival, distance, alloc, buffer)
```

See also

Grid shrink/expand

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Operation [selection] <put parameter description here>

Options:

- 0 — [0] Shrink
- 1 — [1] Expand

Default: 0

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Radius [number] <put parameter description here>

Default: 1

Method [selection] <put parameter description here>

Options:

- 0 — [0] min
- 1 — [1] max
- 2 — [2] mean
- 3 — [3] majority

Default: 0

Outputs

Result Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridshrinkexpand', input, operation, mode, radius, method_expand, result)
```

See also

Invert data/no-data

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Outputs

Result [raster] <put output description here>

Console usage

```
processing.runalg('saga:invertdatanodata', input, output)
```

See also

Merge raster layers

Description

<put algorithm description here>

Parameters

Grids to Merge [**multipleinput: rasters**] <put parameter description here>

Preferred data storage type [**selection**] <put parameter description here>

Options:

- 0 — [0] 1 bit
- 1 — [1] 1 byte unsigned integer
- 2 — [2] 1 byte signed integer
- 3 — [3] 2 byte unsigned integer
- 4 — [4] 2 byte signed integer
- 5 — [5] 4 byte unsigned integer
- 6 — [6] 4 byte signed integer
- 7 — [7] 4 byte floating point
- 8 — [8] 8 byte floating point

Default: 0

Interpolation [**selection**] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Overlapping Cells [**selection**] <put parameter description here>

Options:

- 0 — [0] mean value
- 1 — [1] first value in order of grid list

Default: 0

Outputs

Merged Grid [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:mergerasterlayers', grids, type, interpol, overlap, merged)
```

See also

Patching

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Patch Grid [raster] <put parameter description here>

Interpolation Method [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Outputs

Completed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:patching', original, additional, interpolation, completed)
```

See also

Proximity grid

Description

<put algorithm description here>

Parameters

Features [raster] <put parameter description here>

Outputs

Distance [raster] <put output description here>

Direction [raster] <put output description here>

Allocation [raster] <put output description here>

Console usage

```
processing.runalg('saga:proximitygrid', features, distance, direction, allocation)
```

See also

Reclassify grid values

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] single
- 1 — [1] range
- 2 — [2] simple table

Default: 0

old value (for single value change) [number] <put parameter description here>

Default: 0.0

new value (for single value change) [number] <put parameter description here>

Default: 1.0

operator (for single value change) [selection] <put parameter description here>

Options:

- 0 — [0] =
- 1 — [1] <
- 2 — [2] <=
- 3 — [3] >=
- 4 — [4] >

Default: 0

minimum value (for range) [number] <put parameter description here>

Default: 0.0

maximum value (for range) [number] <put parameter description here>

Default: *1.0*

new value (for range) [number] <put parameter description here>

Default: *2.0*

operator (for range) [selection] <put parameter description here>

Options:

- 0 — [0] <=
- 1 — [1] <

Default: *0*

Lookup Table [fixedtable] <put parameter description here>

operator (for table) [selection] <put parameter description here>

Options:

- 0 — [0] min <= value < max
- 1 — [1] min <= value <= max
- 2 — [2] min < value <= max
- 3 — [3] min < value < max

Default: *0*

replace no data values [boolean] <put parameter description here>

Default: *True*

new value for no data values [number] <put parameter description here>

Default: *0.0*

replace other values [boolean] <put parameter description here>

Default: *True*

new value for other values [number] <put parameter description here>

Default: *0.0*

Outputs

Reclassified Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:reclassifygridvalues', input, method, old, new, soperator, min, max, rnew,
```

See also

Resampling

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Preserve Data Type [boolean] <put parameter description here>

Default: *True*

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Interpolation Method (Scale Up) [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation
- 5 — [5] Mean Value
- 6 — [6] Mean Value (cell area weighted)
- 7 — [7] Minimum Value
- 8 — [8] Maximum Value
- 9 — [9] Majority

Default: *0*

Interpolation Method (Scale Down) [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: *0*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:resampling', input, keep_type, target, scale_up_method, scale_down_method)
```

See also

Sort grid

Description

<put algorithm description here>

Parameters

Input Grid [raster] <put parameter description here>

Down sort [boolean] <put parameter description here>

Default: *True*

Outputs

Sorted Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:sortgrid', grid, down, output)
```

See also

Split RGB bands

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Outputs

Output R band layer [raster] <put output description here>

Output G band layer [raster] <put output description here>

Output B band layer [raster] <put output description here>

Console usage

```
processing.runalg('saga:splitrgbbands', input, r, g, b)
```

See also

Threshold buffer

Description

<put algorithm description here>

Parameters

Features Grid [raster] <put parameter description here>

Value Grid [raster] <put parameter description here>

Threshold Grid [raster] Optional.

<put parameter description here>

Threshold [number] <put parameter description here>

Default: *0.0*

Threshold Type [selection] <put parameter description here>

Options:

- 0 — [0] Absolute
- 1 — [1] Relative from cell value

Default: *0*

Outputs

Buffer Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:thresholdbuffer', features, value, thresholdgrid, threshold, thresholdtype)
```

See also

.

18.7.8 Grid visualization

Histogram surface

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] rows
- 1 — [1] columns
- 2 — [2] circle

Default: 0

Outputs

Histogram [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:histogramsurface', grid, method, hist)
```

See also

Rgb composite

Description

<put algorithm description here>

Parameters

R [**raster**] <put parameter description here>

G [**raster**] <put parameter description here>

B [**raster**] <put parameter description here>

Method for R value [**selection**] <put parameter description here>

Options:

- 0 — 0 - 255
- 1 — Rescale to 0 - 255
- 2 — User defined rescale
- 3 — Percentiles
- 4 — Percentage of standard deviation

Default: 0

Method for G value [**selection**] <put parameter description here>

Options:

- 0 — 0 - 255
- 1 — Rescale to 0 - 255

- 2 — User defined rescale
- 3 — Percentiles
- 4 — Percentage of standard deviation

Default: 0

Method for B value [selection] <put parameter description here>

Options:

- 0 — 0 - 255
- 1 — Rescale to 0 - 255
- 2 — User defined rescale
- 3 — Percentiles
- 4 — Percentage of standard deviation

Default: 0

Rescale Range for RED min [number] <put parameter description here>

Default: 0

Rescale Range for RED max [number] <put parameter description here>

Default: 255

Percentiles Range for RED max [number] <put parameter description here>

Default: 1

Percentiles Range for RED max [number] <put parameter description here>

Default: 99

Percentage of standard deviation for RED [number] <put parameter description here>

Default: 150.0

Rescale Range for GREEN min [number] <put parameter description here>

Default: 0

Rescale Range for GREEN max [number] <put parameter description here>

Default: 255

Percentiles Range for GREEN max [number] <put parameter description here>

Default: 1

Percentiles Range for GREEN max [number] <put parameter description here>

Default: 99

Percentage of standard deviation for GREEN [number] <put parameter description here>

Default: 150.0

Rescale Range for BLUE min [number] <put parameter description here>

Default: 0

Rescale Range for BLUE max [number] <put parameter description here>

Default: 255

Percentiles Range for BLUE max [number] <put parameter description here>

Default: 1

Percentiles Range for BLUE max [number] <put parameter description here>

Default: *99*

Percentage of standard deviation for BLUE [number] <put parameter description here>

Default: *150.0*

Outputs

Output RGB [raster] <put output description here>

Console usage

```
processing.runalg('saga:rgbcomposite', grid_r, grid_g, grid_b, r_method, g_method, b_method, r_ra
```

See also

.

18.7.9 Imagery classification

Change detection

Description

<put algorithm description here>

Parameters

Initial State [raster] <put parameter description here>

Look-up Table [table] Optional.

<put parameter description here>

Value [tablefield: any] <put parameter description here>

Value (Maximum) [tablefield: any] <put parameter description here>

Name [tablefield: any] <put parameter description here>

Final State [raster] <put parameter description here>

Look-up Table [table] Optional.

<put parameter description here>

Value [tablefield: any] <put parameter description here>

Value (Maximum) [tablefield: any] <put parameter description here>

Name [tablefield: any] <put parameter description here>

Report Unchanged Classes [boolean] <put parameter description here>

Default: *True*

Output as... [selection] <put parameter description here>

Options:

- 0 — [0] cells
- 1 — [1] percent
- 2 — [2] area

Default: 0

Outputs

Changes [raster] <put output description here>

Changes [table] <put output description here>

Console usage

```
processing.runalg('saga:changedetection', initial, ini_lut, ini_lut_min, ini_lut_max, ini_lut_name)
```

See also

Cluster analysis for grids

Description

<put algorithm description here>

Parameters

Grids [multipleinput: rasters] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Iterative Minimum Distance (Forgy 1965)
- 1 — [1] Hill-Climbing (Rubin 1967)
- 2 — [2] Combined Minimum Distance / Hillclimbing

Default: 0

Clusters [number] <put parameter description here>

Default: 5

Normalise [boolean] <put parameter description here>

Default: *True*

Old Version [boolean] <put parameter description here>

Default: *True*

Outputs

Clusters [raster] <put output description here>

Statistics [table] <put output description here>

Console usage

```
processing.runalg('saga:clusteranalysisforgrids', grids, method, ncluster, normalise, oldversion,
```

See also

Supervised classification

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Training Areas [**vector: polygon**] <put parameter description here>

Class Identifier [**tablefield: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] Binary Encoding
- 1 — [1] Parallelepiped
- 2 — [2] Minimum Distance
- 3 — [3] Mahalanobis Distance
- 4 — [4] Maximum Likelihood
- 5 — [5] Spectral Angle Mapping
- 6 — [6] Winner Takes All

Default: *0*

Normalise [**boolean**] <put parameter description here>

Default: *True*

Distance Threshold [**number**] <put parameter description here>

Default: *0.0*

Probability Threshold (Percent) [**number**] <put parameter description here>

Default: *0.0*

Probability Reference [**selection**] <put parameter description here>

Options:

- 0 — [0] absolute
- 1 — [1] relative

Default: *0*

Spectral Angle Threshold (Degree) [**number**] <put parameter description here>

Default: *0.0*

Outputs

Class Information [table] <put output description here>

Classification [raster] <put output description here>

Quality [raster] <put output description here>

Console usage

```
processing.runalg('saga:supervisedclassification', grids, roi, roi_id, method, normalise, thresho
```

See also

.

18.7.10 Imagery RGA

Fast region growing algorithm

Description

<put algorithm description here>

Parameters

Input Grids [multipleinput: rasters] <put parameter description here>

Seeds Grid [raster] <put parameter description here>

Smooth Rep [raster] Optional.

<put parameter description here>

Outputs

Segmente [raster] <put output description here>

Mean [raster] <put output description here>

Console usage

```
processing.runalg('saga:fastregiongrowingalgorithm', input, start, rep, result, mean)
```

See also

.

18.7.11 Imagery segmentation

Grid skeletonization

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Standard
- 1 — [1] Hilditch's Algorithm
- 2 — [2] Channel Skeleton

Default: 0

Initialisation [selection] <put parameter description here>

Options:

- 0 — [0] Less than
- 1 — [1] Greater than

Default: 0

Threshold (Init.) [number] <put parameter description here>

Default: 0.0

Convergence [number] <put parameter description here>

Default: 3.0

Outputs

Skeleton [raster] <put output description here>

Skeleton [vector] <put output description here>

Console usage

```
processing.runalg('saga:gridskeletonization', input, method, init_method, init_threshold, converg
```

See also

Seed generation

Description

<put algorithm description here>

Parameters

Features [**multipleinput: rasters**] <put parameter description here>

Bandwidth (Cells) [**number**] <put parameter description here>

Default: 2

Type of Surface [**selection**] <put parameter description here>

Options:

- 0 — [0] smoothed surface
- 1 — [1] variance (a)
- 2 — [2] variance (b)

Default: 0

Extraction of... [**selection**] <put parameter description here>

Options:

- 0 — [0] minima
- 1 — [1] maxima
- 2 — [2] minima and maxima

Default: 0

Feature Aggregation [**selection**] <put parameter description here>

Options:

- 0 — [0] additive
- 1 — [1] multiplicative

Default: 0

Normalized [**boolean**] <put parameter description here>

Default: *True*

Outputs

Surface [**raster**] <put output description here>

Seeds Grid [**raster**] <put output description here>

Seeds [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:seedgeneration', grids, factor, type_surface, type_seeds, type_merge, norm
```

See also

Simple region growing

Description

<put algorithm description here>

Parameters

Seeds [**raster**] <put parameter description here>

Features [**multipleinput: rasters**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] feature space and position
- 1 — [1] feature space

Default: *0*

Neighbourhood [**selection**] <put parameter description here>

Options:

- 0 — [0] 4 (von Neumann)
- 1 — [1] 8 (Moore)

Default: *0*

Variance in Feature Space [**number**] <put parameter description here>

Default: *1.0*

Variance in Position Space [**number**] <put parameter description here>

Default: *1.0*

Threshold - Similarity [**number**] <put parameter description here>

Default: *0.0*

Refresh [**boolean**] <put parameter description here>

Default: *True*

Leaf Size (for Speed Optimisation) [**number**] <put parameter description here>

Default: *256*

Outputs

Segments [**raster**] <put output description here>

Similarity [**raster**] <put output description here>

Seeds [**table**] <put output description here>

Console usage

```
processing.runalg('saga:simpleregiongrowing', seeds, features, method, neighbour, sig_1, sig_2, t
```

See also

Watershed segmentation

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Output [selection] <put parameter description here>

Options:

- 0 — [0] Seed Value
- 1 — [1] Segment ID

Default: 0

Method [selection] <put parameter description here>

Options:

- 0 — [0] Minima
- 1 — [1] Maxima

Default: 0

Join Segments based on Threshold Value [selection] <put parameter description here>

Options:

- 0 — [0] do not join
- 1 — [1] seed to saddle difference
- 2 — [2] seeds difference

Default: 0

Threshold [number] <put parameter description here>

Default: 0

Allow Edge Pixels to be Seeds [boolean] <put parameter description here>

Default: *True*

Borders [boolean] <put parameter description here>

Default: *True*

Outputs

Segments [raster] <put output description here>

Seed Points [vector] <put output description here>

Borders [raster] <put output description here>

Console usage

```
processing.runalg('saga:watershedsegmentation', grid, output, down, join, threshold, edge, bborder)
```

See also

.

18.7.12 Imagery tools

Vegetation index[distance based]

Description

<put algorithm description here>

Parameters

Near Infrared Band [raster] <put parameter description here>

Red Band [raster] <put parameter description here>

Slope of the soil line [number] <put parameter description here>

Default: *0.0*

Intercept of the soil line [number] <put parameter description here>

Default: *0.0*

Outputs

PVI (Richardson and Wiegand) [raster] <put output description here>

PVI (Perry & Lautenschlager) [raster] <put output description here>

PVI (Walther & Shabaani) [raster] <put output description here>

PVI (Qi, et al) [raster] <put output description here>

Console usage

```
processing.runalg('saga:vegetationindexdistancebased', nir, red, slope, intercept, pvi, pvi1, pvi2)
```

See also

Vegetation index[slope based]

Description

<put algorithm description here>

Parameters

Near Infrared Band [raster] <put parameter description here>

Red Band [raster] <put parameter description here>

Outputs

Normalized Difference Vegetation Index [raster] <put output description here>

Ratio Vegetation Index [raster] <put output description here>

Transformed Vegetation Index [raster] <put output description here>

Corrected Transformed Vegetation Index [raster] <put output description here>

Thiam's Transformed Vegetation Index [raster] <put output description here>

Normalized Ratio Vegetation Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:vegetationindexslopebased', nir, red, ndvi, ratio, tvi, ctvi, ttvi, nratio)
```

See also

.

18.7.13 Kriging

Ordinary kriging (global)

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Create Variance Grid [boolean] <put parameter description here>

Default: *True*

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Variogram Model [selection] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression
- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: *0*

Block Kriging [boolean] <put parameter description here>

Default: *True*

Block Size [number] <put parameter description here>

Default: *100*

Logarithmic Transformation [boolean] <put parameter description here>

Default: *True*

Nugget [number] <put parameter description here>

Default: *0.0*

Sill [number] <put parameter description here>

Default: *0.0*

Range [number] <put parameter description here>

Default: *0.0*

Linear Regression [number] <put parameter description here>

Default: *1.0*

Exponential Regression [number] <put parameter description here>

Default: *0.1*

Power Function - A [number] <put parameter description here>

Default: *1.0*

Power Function - B [number] <put parameter description here>

Default: *0.5*

Grid Size [number] <put parameter description here>

Default: *1.0*

Fit Extent [boolean] <put parameter description here>

Default: *True*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Outputs

Grid [raster] <put output description here>

Variance [raster] <put output description here>

Console usage

```
processing.runalg('saga:ordinarykrigingglobal', shapes, field, bvariance, target, model, block, d
```

See also

Ordinary kriging

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Create Variance Grid [**boolean**] <put parameter description here>

Default: *True*

Target Grid [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Variogram Model [**selection**] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression
- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: *0*

Block Kriging [**boolean**] <put parameter description here>

Default: *True*

Block Size [**number**] <put parameter description here>

Default: *100*

Logarithmic Transformation [**boolean**] <put parameter description here>

Default: *True*

Nugget [**number**] <put parameter description here>

Default: *0.0*

Sill [**number**] <put parameter description here>

Default: *10.0*

Range [**number**] <put parameter description here>

Default: *100.0*

Linear Regression [**number**] <put parameter description here>

Default: *1.0*

Exponential Regression [**number**] <put parameter description here>

Default: *0.1*

Power Function - A [**number**] <put parameter description here>

Default: *1*

Power Function - B [**number**] <put parameter description here>

Default: *0.5*

Maximum Search Radius (map units) [**number**] <put parameter description here>

Default: *1000.0*

Min. Number of m_Points [**number**] <put parameter description here>

Default: *4*

Max. Number of m_Points [**number**] <put parameter description here>

Default: *20*

Grid Size [**number**] <put parameter description here>

Default: *1.0*

Fit Extent [**boolean**] <put parameter description here>

Default: *True*

Output extent [**extent**] <put parameter description here>

Default: *0,1,0,1*

Outputs

Grid [**raster**] <put output description here>

Variance [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:ordinarykriging', shapes, field, bvariance, target, model, block, dblock,
```

See also

Universal kriging (global)

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Create Variance Grid [**boolean**] <put parameter description here>

Default: *True*

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Variogram Model [selection] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression
- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: 0

Block Kriging [boolean] <put parameter description here>

Default: *True*

Block Size [number] <put parameter description here>

Default: 100

Logarithmic Transformation [boolean] <put parameter description here>

Default: *True*

Nugget [number] <put parameter description here>

Default: 0.0

Sill [number] <put parameter description here>

Default: 0.0

Range [number] <put parameter description here>

Default: 0.0

Linear Regression [number] <put parameter description here>

Default: 1

Exponential Regression [number] <put parameter description here>

Default: 0.5

Power Function - A [number] <put parameter description here>

Default: 1.0

Power Function - B [number] <put parameter description here>

Default: 0.1

Grids [multipleinput: rasters] <put parameter description here>

Grid Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation

- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: *0*

Grid Size [number] <put parameter description here>

Default: *1.0*

Fit Extent [boolean] <put parameter description here>

Default: *True*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Outputs

Grid [raster] <put output description here>

Variance [raster] <put output description here>

Console usage

```
processing.runalg('saga:universalkrigingglobal', shapes, field, bvariance, target, model, block, ...)
```

See also

Universal kriging

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Create Variance Grid [boolean] <put parameter description here>

Default: *True*

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Variogram Model [selection] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression

- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: 0

Block Kriging [boolean] <put parameter description here>

Default: *True*

Block Size [number] <put parameter description here>

Default: 100

Logarithmic Transformation [boolean] <put parameter description here>

Default: *True*

Nugget [number] <put parameter description here>

Default: 0.0

Sill [number] <put parameter description here>

Default: 0.0

Range [number] <put parameter description here>

Default: 0.0

Linear Regression [number] <put parameter description here>

Default: 1.0

Exponential Regression [number] <put parameter description here>

Default: 0.1

Power Function - A [number] <put parameter description here>

Default: 1

Power Function - B [number] <put parameter description here>

Default: 0.5

Grids [multipleinput: rasters] <put parameter description here>

Grid Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Min.Number of m_Points [number] <put parameter description here>

Default: 4

Max. Number of m_Points [number] <put parameter description here>

Default: 20

Maximum Search Radius (map units) [number] <put parameter description here>

Default: 1000.0

Grid Size [number] <put parameter description here>

Default: *1.0*

Fit Extent [boolean] <put parameter description here>

Default: *True*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Outputs

Grid [raster] <put output description here>

Variance [raster] <put output description here>

Console usage

```
processing.runalg('saga:universalkriging', shapes, field, bvariance, target, model, block, dblock,
```

See also

.

18.7.14 Shapes grid

Add grid values to points

Description

Creates a new vector layer as a result of the union of a points layer with the interpolated value of one or more base background grid layer(s). This way, the new layer created will have a new column in the attribute table that reflects the interpolated value of the background grid.

Parameters

Points [vector: point] Input layer.

Grids [multipleinput: rasters] Background grid layer(s)

Interpolation [selection] interpolation method to use.

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: *0*

Outputs

Result [vector] The resulting layer.

Console usage

```
processing.runalg('saga:addgridvaluestopoints', shapes, grids, interpol, result)
```

See also

Add grid values to shapes

Description

<put algorithm description here>

Parameters

Shapes [vector: any] <put parameter description here>

Grids [multipleinput: rasters] <put parameter description here>

Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Outputs

Result [vector] <put output description here>

Console usage

```
processing.runalg('saga:addgridvaluestoshapes', shapes, grids, interpol, result)
```

See also

Clip grid with polygon

Description

<put algorithm description here>

Parameters

Input [raster] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Outputs

Output [raster] <put output description here>

Console usage

```
processing.runalg('saga:clipgridwithpolygon', input, polygons, output)
```

See also

Contour lines from grid

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Minimum Contour Value [number] <put parameter description here>

Default: *0.0*

Maximum Contour Value [number] <put parameter description here>

Default: *10000.0*

Equidistance [number] <put parameter description here>

Default: *100.0*

Outputs

Contour Lines [vector] <put output description here>

Console usage

```
processing.runalg('saga:contourlinesfromgrid', input, zmin, zmax, zstep, contour)
```

See also

Gradient vectors from directional components

Description

<put algorithm description here>

Parameters

X Component [raster] <put parameter description here>

Y Component [raster] <put parameter description here>

Step [number] <put parameter description here>

Default: *1*

Size Range Min [number] <put parameter description here>

Default: *25.0*

Size Range Max [number] <put parameter description here>

Default: *100.0*

Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] nearest neighbour
- 1 — [1] mean value

Default: *0*

Style [selection] <put parameter description here>

Options:

- 0 — [0] simple line
- 1 — [1] arrow
- 2 — [2] arrow (centered to cell)

Default: *0*

Outputs

Gradient Vectors [vector] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorsfromdirectionalcomponents', x, y, step, size_min, size_max)
```

See also

Gradient vectors from direction and length

Description

<put algorithm description here>

Parameters

Direction [raster] <put parameter description here>

Length [raster] <put parameter description here>

Step [number] <put parameter description here>

Default: 1

Size Range Min [number] <put parameter description here>

Default: 25.0

Size Range Max [number] <put parameter description here>

Default: 100.0

Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] nearest neighbour
- 1 — [1] mean value

Default: 0

Style [selection] <put parameter description here>

Options:

- 0 — [0] simple line
- 1 — [1] arrow
- 2 — [2] arrow (centered to cell)

Default: 0

Outputs

Gradient Vectors [vector] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorsfromdirectionandlength', dir, len, step, size_min, size_max)
```

See also

Gradient vectors from surface

Description

<put algorithm description here>

Parameters

Surface [raster] <put parameter description here>

Step [number] <put parameter description here>

Default: 1

Size Range Min [number] <put parameter description here>

Default: 25.0

Size Range Max [number] <put parameter description here>

Default: 100.0

Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] nearest neighbour
- 1 — [1] mean value

Default: 0

Style [selection] <put parameter description here>

Options:

- 0 — [0] simple line
- 1 — [1] arrow
- 2 — [2] arrow (centered to cell)

Default: 0

Outputs

Gradient Vectors [vector] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorsfromsurface', surface, step, size_min, size_max, aggr, sty
```

See also

Grid statistics for polygons

Description

<put algorithm description here>

Parameters

Grids [multipleinput: rasters] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Number of Cells [boolean] <put parameter description here>

Default: *True*

Minimum [boolean] <put parameter description here>

Default: *True*

Maximum [boolean] <put parameter description here>

Default: *True*

Range [boolean] <put parameter description here>

Default: *True*

Sum [boolean] <put parameter description here>

Default: *True*

Mean [boolean] <put parameter description here>

Default: *True*

Variance [boolean] <put parameter description here>

Default: *True*

Standard Deviation [boolean] <put parameter description here>

Default: *True*

Quantiles [number] <put parameter description here>

Default: *0*

Outputs

Statistics [vector] <put output description here>

Console usage

```
processing.runalg('saga:gridstatisticsforpolygons', grids, polygons, count, min, max, range, sum,
```

See also

Grid values to points (randomly)

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Frequency [number] <put parameter description here>

Default: *100*

Outputs

Points [vector] <put output description here>

Console usage

```
processing.runalg('saga:gridvaluestopointsrandomly', grid, freq, points)
```

See also

Grid values to points

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Polygons [**vector: any**] Optional.

<put parameter description here>

Exclude NoData Cells [**boolean**] <put parameter description here>

Default: *True*

Type [**selection**] <put parameter description here>

Options:

- 0 — [0] nodes
- 1 — [1] cells

Default: *0*

Outputs

Shapes [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:gridvaluestopoints', grids, polygons, nodata, type, shapes)
```

See also

Local minima and maxima

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Outputs

Minima [**vector**] <put output description here>

Maxima [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:localminimaandmaxima', grid, minima, maxima)
```

See also

Vectorising grid classes

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Class Selection [selection] <put parameter description here>

Options:

- 0 — [0] one single class specified by class identifier
- 1 — [1] all classes

Default: 0

Class Identifier [number] <put parameter description here>

Default: 0

Vectorised class as... [selection] <put parameter description here>

Options:

- 0 — [0] one single (multi-)polygon object
- 1 — [1] each island as separated polygon

Default: 0

Outputs

Polygons [vector] <put output description here>

Console usage

```
processing.runalg('saga:vectorisinggridclasses', grid, class_all, class_id, split, polygons)
```

See also

.

18.7.15 Shapes lines

Convert points to line(s)

Description

Converts points to lines.

Parameters

Points [vector: point] Points to convert.

Order by... [tablefield: any] Lines will be ordered following this field.

Separate by... [tablefield: any] Lines will be grouped according to this field.

Outputs

Lines [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertpointstolines', points, order, separate, lines)
```

See also

Convert polygons to lines

Description

Creates lines from polygons.

Parameters

Polygons [vector: polygon] Layer to process.

Outputs

Lines [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertpolygonstolines', polygons, lines)
```

See also

Line dissolve

Description

<put algorithm description here>

Parameters

Lines [vector: any] <put parameter description here>

1. **Attribute** [tablefield: any] <put parameter description here>
2. **Attribute** [tablefield: any] <put parameter description here>

3. Attribute [tablefield: any] <put parameter description here>

Dissolve... [selection] <put parameter description here>

Options:

- 0 — [0] lines with same attribute value(s)
- 1 — [1] all lines

Default: 0

Outputs

Dissolved Lines [vector] <put output description here>

Console usage

```
processing.runalg('saga:linedissolve', lines, field_1, field_2, field_3, all, dissolved)
```

See also

Line-polygon intersection

Description

<put algorithm description here>

Parameters

Lines [vector: line] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Output [selection] <put parameter description here>

Options:

- 0 — [0] one multi-line per polygon
- 1 — [1] keep original line attributes

Default: 0

Outputs

Intersection [vector] <put output description here>

Console usage

```
processing.runalg('saga:linepolygonintersection', lines, polygons, method, intersect)
```

See also

Line properties

Description

Calculates some information on each line of the layer.

Parameters

Lines [**vector: line**] Layer to analyze.

Number of Parts [**boolean**] Determines whether to calculate number of segments in line.

Default: *True*

Number of Vertices [**boolean**] Determines whether to calculate number of vertices in line.

Default: *True*

Length [**boolean**] Determines whether to calculate total line length.

Default: *True*

Outputs

Lines with Property Attributes [**vector**] The resulting layer.

Console usage

```
processing.runalg('saga:lineproperties', lines, bparts, bpoints, blength, output)
```

See also

Line simplification

Description

Simplifies the geometry of a lines layer.

Parameters

Lines [**vector: line**] Layer to process.

Tolerance [**number**] Simplification tolerance.

Default: *1.0*

Outputs

Simplified Lines [**vector**] The resulting layer.

Console usage

```
processing.runalg('saga:linesimplification', lines, tolerance, output)
```

See also

.

18.7.16 Shapes points

Add coordinates to points

Description

Adds the X and Y coordinates of feature in the attribute table of input layer.

Parameters

Points [**vector: point**] Input layer.

Outputs

Output [**vector**] Resulting layer with the updated attribute table.

Console usage

```
processing.runalg('saga:addcoordinatestopoints', input, output)
```

See also

Add polygon attributes to points

Description

Adds the specified field of the polygons layer to the attribute table of the points layer. The new attributes added for each point depend on the value of the background polygon layer.

Parameters

Points [**vector: point**] Points layer.

Polygons [**vector: polygon**] Background polygons layer.

Attribute [**tablefield: any**] Attribute of the polygons layer that will be added to the points layer.

Outputs

Result [**vector**] The resulting layer.

Console usage

```
processing.runalg('saga:addpolygonattributestopoints', input, polygons, field, output)
```

See also

Aggregate point observations

Description

<put algorithm description here>

Parameters

Reference Points [**vector: any**] <put parameter description here>

ID [**tablefield: any**] <put parameter description here>

Observations [**table**] <put parameter description here>

X [**tablefield: any**] <put parameter description here>

Y [**tablefield: any**] <put parameter description here>

Track [**tablefield: any**] <put parameter description here>

Date [**tablefield: any**] <put parameter description here>

Time [**tablefield: any**] <put parameter description here>

Parameter [**tablefield: any**] <put parameter description here>

Maximum Time Span (Seconds) [**number**] <put parameter description here>

Default: *60.0*

Maximum Distance [**number**] <put parameter description here>

Default: *0.002*

Outputs

Aggregated [**table**] <put output description here>

Console usage

```
processing.runalg('saga:aggregatepointobservations', reference, reference_id, observations, x, y,
```

See also

Clip points with polygons

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Add Attribute to Clipped Points [tablefield: any] <put parameter description here>

Clipping Options [selection] <put parameter description here>

Options:

- 0 — [0] one layer for all points
- 1 — [1] separate layer for each polygon

Default: 0

Outputs

Clipped Points [vector] <put output description here>

Console usage

```
processing.runalg('saga:clippointswithpolygons', points, polygons, field, method, clips)
```

See also

Convert lines to points

Description

Converts lines layer into a points.

Parameters

Lines [vector: line] Lines layer to convert.

Insert Additional Points [boolean] Determines whether to add additional nodes or not.

Default: *True*

Insert Distance [number] Distance between the additional points.

Default: *1.0*

Outputs

Points [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertlinestopoints', lines, add, dist, points)
```

See also

Convert multipoints to points

Description

<put algorithm description here>

Parameters

Multipoints [**vector: point**] <put parameter description here>

Outputs

Points [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:convertmultipointstopoints', multipoints, points)
```

See also

Convex hull

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Hull Construction [**selection**] <put parameter description here>

Options:

- 0 — [0] one hull for all shapes
- 1 — [1] one hull per shape
- 2 — [2] one hull per shape part

Default: 0

Outputs

Convex Hull [**vector**] <put output description here>

Minimum Bounding Box [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:convexhull', shapes, polypoints, hulls, boxes)
```

See also

Distance matrix

Description

Generates a distance matrix between each point of the input layer. A unique ID will be created in the first row of the resulting matrix (symmetric matrix), while every other cell reflects the distance between the points.

Parameters

Points [vector: point] Input layer.

Outputs

Distance Matrix Table [table] The resulting table.

Console usage

```
processing.runalg('saga:distancematrix', points, table)
```

See also

Fit n points to shape

Description

<put algorithm description here>

Parameters

Shapes [vector: polygon] <put parameter description here>

Number of points [number] <put parameter description here>

Default: 10

Outputs

Points [vector] <put output description here>

Console usage

```
processing.runalg('saga:fitnpointstoshape', shapes, numpoints, points)
```

See also

Points filter

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Radius [**number**] <put parameter description here>

Default: *1*

Minimum Number of Points [**number**] <put parameter description here>

Default: *0*

Maximum Number of Points [**number**] <put parameter description here>

Default: *0*

Quadrants [**boolean**] <put parameter description here>

Default: *True*

Filter Criterion [**selection**] <put parameter description here>

Options:

- 0 — [0] keep maxima (with tolerance)
- 1 — [1] keep minima (with tolerance)
- 2 — [2] remove maxima (with tolerance)
- 3 — [3] remove minima (with tolerance)
- 4 — [4] remove below percentile
- 5 — [5] remove above percentile

Default: *0*

Tolerance [**number**] <put parameter description here>

Default: *0.0*

Percentile [**number**] <put parameter description here>

Default: *50*

Outputs

Filtered Points [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:pointsfilter', points, field, radius, minnum, maxnum, quadrants, method, t
```


See also

Points thinning

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Resolution [**number**] <put parameter description here>

Default: *1.0*

Outputs

Thinned Points [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:pointsthinning', points, field, resolution, thinned)
```

See also

Remove duplicate points

Description

<put algorithm description here>

Parameters

Points [**vector: any**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Point to Keep [**selection**] <put parameter description here>

Options:

- 0 — [0] first point
- 1 — [1] last point
- 2 — [2] point with minimum attribute value
- 3 — [3] point with maximum attribute value

Default: *0*

Numeric Attribute Values [**selection**] <put parameter description here>

Options:

- 0 — [0] take value from the point to be kept

- 1 — [1] minimum value of all duplicates
- 2 — [2] maximum value of all duplicates
- 3 — [3] mean value of all duplicates

Default: 0

Outputs

Result [vector] <put output description here>

Console usage

```
processing.runalg('saga:removeduplicatepoints', points, field, method, numeric, result)
```

See also

Separate points by direction

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Number of Directions [number] <put parameter description here>

Default: 4

Tolerance (Degree) [number] <put parameter description here>

Default: 5

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('saga:separatepointsbydirection', points, directions, tolerance, output)
```

See also

.

18.7.17 Shapes polygons

Convert lines to polygons

Description

Converts lines to polygons.

Parameters

Lines [vector: line] Lines to convert.

Outputs

Polygons [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertlinestopolygons', lines, polygons)
```

See also

Convert polygon/line vertices to points

Description

Converts the line or polygon vertices into points.

Parameters

Shapes [vector: any] Layer to process.

Outputs

Points [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertpolygonlineverticestopoints', shapes, points)
```

See also

Polygon centroids

Description

Calculates the centroids of polygons.

Parameters

Polygons [vector: polygon] Input layer.

Centroids for each part [boolean] Determines whether centroids should be calculated for each part of multipart polygon or not.

Default: *True*

Outputs

Centroids [vector] The resulting layer.

Console usage

```
processing.runalg('saga:polygoncentroids', polygons, method, centroids)
```

See also

Polygon dissolve

Description

<put algorithm description here>

Parameters

Polygons [vector: polygon] <put parameter description here>

1. **Attribute** [tablefield: any] Optional.

<put parameter description here>

2. **Attribute** [tablefield: any] Optional.

<put parameter description here>

3. **Attribute** [tablefield: any] Optional.

<put parameter description here>

Dissolve... [selection] <put parameter description here>

Options:

- 0 — [0] polygons with same attribute value
- 1 — [1] all polygons
- 2 — [2] polygons with same attribute value (keep inner boundaries)
- 3 — [3] all polygons (keep inner boundaries)

Default: *0*

Outputs

Dissolved Polygons [vector] <put output description here>

Console usage

```
processing.runalg('saga:polygondissolve', polygons, field_1, field_2, field_3, dissolve, dissolve)
```

See also

Polygon-line intersection

Description

<put algorithm description here>

Parameters

Polygons [vector: **polygon**] <put parameter description here>

Lines [vector: **line**] <put parameter description here>

Outputs

Intersection [vector] <put output description here>

Console usage

```
processing.runalg('saga:polygonlineintersection', polygons, lines, intersect)
```

See also

Polygon parts to separate polygons

Description

<put algorithm description here>

Parameters

Polygons [vector: **polygon**] <put parameter description here>

Ignore Lakes [boolean] <put parameter description here>

Default: *True*

Outputs

Polygon Parts [vector] <put output description here>

Console usage

```
processing.runalg('saga:polygonpartstoseparatepolygons', polygons, lakes, parts)
```

See also

Polygon properties

Description

<put algorithm description here>

Parameters

Polygons [**vector: polygon**] <put parameter description here>

Number of Parts [**boolean**] <put parameter description here>

Default: *True*

Number of Vertices [**boolean**] <put parameter description here>

Default: *True*

Perimeter [**boolean**] <put parameter description here>

Default: *True*

Area [**boolean**] <put parameter description here>

Default: *True*

Outputs

Polygons with Property Attributes [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:polygonproperties', polygons, bparts, bpoints, blength, barea, output)
```

See also

Polygon shape indices

Description

Calculates spatial statistics for polygons. This includes:

- area
- perimeter
- perimeter / area
- perimeter / square root of the area
- maximum distance
- maximum distance / area
- maximum distance / square root of the area
- shape index

Parameters

Shapes [vector: polygon] Layer to analyze.

Outputs

Shape Index [vector] The resulting layer.

Console usage

```
processing.runalg('saga:polygonshapeindices', shapes, index)
```

See also

Polygons to edges and nodes

Description

Extracts boundaries and nodes of polygons in separate files.

Parameters

Polygons [vector: polygon] Input layer.

Outputs

Edges [vector] Resulting line layer with polygons boundaries.

Nodes [vector] Resulting line layer with polygons nodes.

Console usage

```
processing.runalg('saga:polygonstoedgesandnodes', polygons, edges, nodes)
```

See also

.

18.7.18 Shapes tools

Create graticule

Description

Creates a grid.

Parameters

Extent [**vector: any**] Optional.

Grid will be created according to the selected layer.

Output extent [**extent**] Extent of the grid.

Default: *0,1,0,1*

Division Width [**number**] X-axes spacing between the lines.

Default: *1.0*

Division Height [**number**] Y-axes spacing between the lines.

Default: *1.0*

Type [**selection**] Geometry type of the resulting grid.

Options:

- 0 — [0] Lines
- 1 — [1] Rectangles

Default: *0*

Outputs

Graticule [**vector**] The resulting layer.

Console usage

```
processing.runalg('saga:creategraticule', extent, output_extent, distx, disty, type, graticule)
```

See also

Cut shapes layer

Description

<put algorithm description here>

Parameters

Vector layer to cut [**vector: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] completely contained
- 1 — [1] intersects
- 2 — [2] center

Default: *0*

Cutting polygons [**vector: any**] <put parameter description here>

Outputs

Result [vector] <put output description here>

Extent [vector] <put output description here>

Console usage

```
processing.runalg('saga:cutshapeslayer', shapes, method, polygons_polygons, cut, extent)
```

See also

Get shapes extents

Description

Creates polygons according to the extent of the input layer features.

Parameters

Shapes [vector: any] Input layer.

Parts [boolean] Determines whether create polygon for each feature (`True`) or just create single polygon for whole layer (`False`).

Default: *True*

Outputs

Extents [vector] The resulting layer.

Console usage

```
processing.runalg('saga:getshapesextents', shapes, parts, extents)
```

See also

Merge shapes layers

Description

Merges two or more input layer into a unique resulting layer. You can merge together only layer of the same type (polygons with polygons, lines with lines, points with points).

The attribute table of the resulting layer will include only the attributes of the first input layer. Two additional columns will be added: one corresponding to the ID of every merged layer and the other one corresponding to the original name of the merged layer.

Parameters

Main Layer [vector: any] Initial layer.

Additional Layers [multipleinput: any vectors] Optional.

Layer(s) to merge with.

Outputs

Merged Layer [vector] The resulting layer.

Console usage

```
processing.runalg('saga:mergeshapelayers', main, layers, out)
```

See also

Polar to cartesian coordinates

Description

<put algorithm description here>

Parameters

Polar Coordinates [vector: any] <put parameter description here>

Exaggeration [tablefield: any] <put parameter description here>

Exaggeration Factor [number] <put parameter description here>

Default: *1*

Radius [number] <put parameter description here>

Default: *6371000.0*

Degree [boolean] <put parameter description here>

Default: *True*

Outputs

Cartesian Coordinates [vector] <put output description here>

Console usage

```
processing.runalg('saga:polartocartesiancoordinates', polar, f_exagg, d_exagg, radius, degree, ca
```

See also

Quadtree structure to shapes

Description

<put algorithm description here>

Parameters

Shapes [**vector: any**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Outputs

Polygons [**vector**] <put output description here>

Lines [**vector**] <put output description here>

Duplicated Points [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:quadtreestructuretoshapes', shapes, attribute, polygons, lines, points)
```

See also

Shapes buffer

Description

Creates buffer around features based on fixed distance or distance field.

Parameters

Shapes [**vector: any**] Input layer.

Buffer Distance [**selection**] Buffering method.

Options:

- 0 — [0] fixed value
- 1 — [1] attribute field

Default: 0

Buffer Distance (Fixed) [**number**] Buffer distance for “fixed value” method.

Default: 100.0

Buffer Distance (Attribute) [**tablefield: any**] Name of the distance field for “attribute field” method.

Scaling Factor for Attribute Value [**number**] <put parameter description here>

Default: 1.0

Number of Buffer Zones [number] Number of buffer(s) to generate.

Default: *1.0*

Circle Point Distance [Degree] [number] Smoothness of the buffer borders: great numbers means rough borders.

Default: *5.0*

Dissolve Buffers [boolean] Determines whether to dissolve results or not.

Default: *True*

Outputs

Buffer [vector] The resulting layer.

Console usage

```
processing.runalg('saga:shapesbuffer', shapes, buf_type, buf_dist, buf_field, buf_scale, buf_zones)
```

See also

Split shapes layer randomly

Description

Splits the input layer randomly in two parts.

Parameters

Shapes [vector: any] Layer to split.

Split ratio (%) [number] Split ratio between the resulting layers.

Default: *50*

Outputs

Group A [vector] First resulting layer.

Group B [vector] Second resulting layer.

Console usage

```
processing.runalg('saga:splitshapeslayerrandomly', shapes, percent, a, b)
```

See also

Transform shapes

Description

<put algorithm description here>

Parameters

Shapes [**vector: any**] <put parameter description here>

dX [**number**] <put parameter description here>

Default: *0.0*

dY [**number**] <put parameter description here>

Default: *0.0*

Angle [**number**] <put parameter description here>

Default: *0.0*

Scale Factor X [**number**] <put parameter description here>

Default: *1.0*

Scale Factor Y [**number**] <put parameter description here>

Default: *1.0*

X [**number**] <put parameter description here>

Default: *0.0*

Y [**number**] <put parameter description here>

Default: *0.0*

Outputs

Output [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:transformshapes', in, dx, dy, angle, scalex, scaley, anchorx, anchory, out)
```

See also

.

18.7.19 Shapes transect

Transect through polygon shapefile

Description

<put algorithm description here>

Parameters

Line Transect (s) [**vector: line**] <put parameter description here>

Theme [**vector: any**] <put parameter description here>

Theme Field [**tablefield: any**] <put parameter description here>

Outputs

Result table [table] <put output description here>

Console usage

```
processing.runalg('saga:transectthroughpolygonshapefile', transect, theme, theme_field, transect_
```

See also

.

18.7.20 Simulation fire

Fire risk analysis

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Fuel Model [raster] <put parameter description here>

Wind Speed [raster] <put parameter description here>

Wind Direction [raster] <put parameter description here>

Dead Fuel Moisture 1H [raster] <put parameter description here>

Dead Fuel Moisture 10H [raster] <put parameter description here>

Dead Fuel Moisture 100H [raster] <put parameter description here>

Herbaceous Fuel Moisture [raster] <put parameter description here>

Wood Fuel Moisture [raster] <put parameter description here>

Value [raster] Optional.

<put parameter description here>

Base Probability [raster] Optional.

<put parameter description here>

Number of Events [number] <put parameter description here>

Default: *1000*

Fire Length [number] <put parameter description here>

Default: *100*

Outputs

Danger [raster] <put output description here>

Compound Probability [raster] <put output description here>

Priority Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:fireriskanalysis', dem, fuel, windspd, winddir, m1h, m10h, m100h, mherb, m
```

See also

Simulation

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Fuel Model [raster] <put parameter description here>

Wind Speed [raster] <put parameter description here>

Wind Direction [raster] <put parameter description here>

Dead Fuel Moisture 1H [raster] <put parameter description here>

Dead Fuel Moisture 10H [raster] <put parameter description here>

Dead Fuel Moisture 100H [raster] <put parameter description here>

Herbaceous Fuel Moisture [raster] <put parameter description here>

Wood Fuel Moisture [raster] <put parameter description here>

Ignition Points [raster] <put parameter description here>

Update View [boolean] <put parameter description here>

Default: *True*

Outputs

Time [raster] <put output description here>

Flame Length [raster] <put output description here>

Intensity [raster] <put output description here>

Console usage

```
processing.runalg('saga:simulation', dem, fuel, windspd, winddir, m1h, m10h, m100h, mherb, mwood,
```

See also

.

18.7.21 Simulation hydrology

Overland flow - kinematic wave d8

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Gauges [vector: any] Optional.

<put parameter description here>

Simulation Time [h] [number] <put parameter description here>

Default: 24

Simulation Time Step [h] [number] <put parameter description here>

Default: 0.1

Manning's Roughness [number] <put parameter description here>

Default: 0.03

Max. Iterations [number] <put parameter description here>

Default: 100

Epsilon [number] <put parameter description here>

Default: 0.0001

Precipitation [selection] <put parameter description here>

Options:

- 0 — [0] Homogenous
- 1 — [1] Above Elevation
- 2 — [2] Left Half

Default: 0

Threshold Elevation [number] <put parameter description here>

Default: 0.0

Outputs

Runoff [raster] <put output description here>

Flow at Gauges [table] <put output description here>

Console usage

```
processing.runalg('saga:overlandflowkinematicwaved8', dem, gauges, time_span, time_step, roughnes
```

See also

Water retention capacity

Description

<put algorithm description here>

Parameters

Plot Holes [vector: any] <put parameter description here>

DEM [raster] <put parameter description here>

Outputs

Final Parameters [vector] <put output description here>

Water Retention Capacity [raster] <put output description here>

Console usage

```
processing.runalg('saga:waterretentioncapacity', shapes, dem, output, retention)
```

See also

.

18.7.22 Table calculus

Fill gaps in records

Description

<put algorithm description here>

Parameters

Table [table] <put parameter description here>

Order [tablefield: any] <put parameter description here>

Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbour
- 1 — [1] Linear

- 2 — [2] Spline

Default: 0

Outputs

Table without Gaps [table] <put output description here>

Console usage

```
processing.runalg('saga:fillgapsinrecords', table, order, method, nogaps)
```

See also

Principle components analysis

Description

<put algorithm description here>

Parameters

Table [table] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] correlation matrix
- 1 — [1] variance-covariance matrix
- 2 — [2] sums-of-squares-and-cross-products matrix

Default: 0

Number of Components [number] <put parameter description here>

Default: 3

Outputs

Principle Components [table] <put output description here>

Console usage

```
processing.runalg('saga:principlecomponentsanalysis', table, method, nfirst, pca)
```

See also

Running average

Description

<put algorithm description here>

Parameters

Input [table] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Number of Records [number] <put parameter description here>

Default: 10

Outputs

Output [table] <put output description here>

Console usage

```
processing.runalg('saga:runningaverage', input, field, count, output)
```

See also

.

18.7.23 Table tools

Change date format

Description

Converts the date format of the input layer.

Parameters

Table [table] Input table.

Date Field [tablefield: any] Attribute the date.

Input Format [selection] Input date format.

Options:

- 0 — [0] dd.mm.yy
- 1 — [1] yy.mm.dd
- 2 — [2] dd:mm:yy
- 3 — [3] yy:mm:dd
- 4 — [4] ddmmyyyy, fix size
- 5 — [5] yyyymmdd, fix size
- 6 — [6] ddmmyy, fix size
- 7 — [7] yymmdd, fix size
- 8 — [8] Julian Day

Default: 0

Output Format [selection] Output date format.

Options:

- 0 — [0] dd.mm.yy
- 1 — [1] yy.mm.dd
- 2 — [2] dd:mm:yy
- 3 — [3] yy:mm:dd
- 4 — [4] ddmmyyyy, fix size
- 5 — [5] yyyyymmdd, fix size
- 6 — [6] ddmmyy, fix size
- 7 — [7] yymmdd, fix size
- 8 — [8] Julian Day

Default: 0

Outputs

Output [table] The resulting table.

Console usage

```
processing.runalg('saga:changedateformat', table, field, fmt_in, fmt_out, output)
```

See also

Change time format

Description

Converts the time format of the input layer.

Parameters

Table [table] Input table.

Time Field [tablefield: any] Attribute with time.

Input Format [selection] Input time format.

Options:

- 0 — [0] hh.mm.ss
- 1 — [1] hh:mm:ss
- 2 — [2] hhmmss, fix size
- 3 — [3] hours
- 4 — [4] minutes
- 5 — [5] seconds

Default: 0

Output Format [selection] Output time format.

Options:

- 0 — [0] hh.mm.ss
- 1 — [1] hh:mm:ss
- 2 — [2] hhmmss, fix size
- 3 — [3] hours
- 4 — [4] minutes
- 5 — [5] seconds

Default: 0

Outputs

Output [table] The resulting table.

Console usage

```
processing.runalg('saga:changetimeformat', table, field, fmt_in, fmt_out, output)
```

See also

.

18.7.24 Terrain channels

Channel network and drainage basins

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Threshold [number] <put parameter description here>

Default: 5.0

Outputs

Flow Direction [raster] <put output description here>

Flow Connectivity [raster] <put output description here>

Strahler Order [raster] <put output description here>

Drainage Basins [raster] <put output description here>

Channels [vector] <put output description here>

Drainage Basins [vector] <put output description here>

Junctions [vector] <put output description here>

Console usage

```
processing.runalg('saga:channelnetworkanddrainagebasins', dem, threshold, direction, connection, c
```

See also

Channel network

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Flow Direction [raster] Optional.

<put parameter description here>

Initiation Grid [raster] <put parameter description here>

Initiation Type [selection] <put parameter description here>

Options:

- 0 — [0] Less than
- 1 — [1] Equals
- 2 — [2] Greater than

Default: 0

Initiation Threshold [number] <put parameter description here>

Default: 0.0

Divergence [raster] Optional.

<put parameter description here>

Tracing: Max. Divergence [number] <put parameter description here>

Default: 10

Tracing: Weight [raster] Optional.

<put parameter description here>

Min. Segment Length [number] <put parameter description here>

Default: 10

Outputs

Channel Network [raster] <put output description here>

Channel Direction [raster] <put output description here>

Channel Network [vector] <put output description here>

Console usage

```
processing.runalg('saga:channelnetwork', elevation, sinkroute, init_grid, init_method, init_value)
```

See also

Overland flow distance to channel network

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Channel Network [raster] <put parameter description here>

Flow Algorithm [selection] <put parameter description here>

Options:

- 0 — [0] D8
- 1 — [1] MFD

Default: 0

Outputs

Overland Flow Distance [raster] <put output description here>

Vertical Overland Flow Distance [raster] <put output description here>

Horizontal Overland Flow Distance [raster] <put output description here>

Console usage

```
processing.runalg('saga:overlandflowdistancetochannelnetwork', elevation, channels, method, distanceto)
```

See also

Strahler order

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Outputs

Strahler Order [raster] <put output description here>

Console usage

```
processing.runalg('saga:strahlerorder', dem, strahler)
```

See also

Vertical distance to channel network

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Channel Network [raster] <put parameter description here>

Tension Threshold [Percentage of Cell Size] [number] <put parameter description here>

Default: *1*

Keep Base Level below Surface [boolean] <put parameter description here>

Default: *True*

Outputs

Vertical Distance to Channel Network [raster] <put output description here>

Channel Network Base Level [raster] <put output description here>

Console usage

```
processing.runalg('saga:verticaldistancetochannelnetwork', elevation, channels, threshold, nounde
```

See also

Watershed basins

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Channel Network [raster] <put parameter description here>

Sink Route [raster] Optional.

<put parameter description here>

Min. Size [number] <put parameter description here>

Default: 0

Outputs

Watershed Basins [raster] <put output description here>

Console usage

```
processing.runalg('saga:watershedbasins', elevation, channels, sinkroute, minsize, basins)
```

See also

.

18.7.25 Terrain hydrology

Burn stream network into dem

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Streams [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] simply decrease cell's value by epsilon
- 1 — [1] lower cell's value to neighbours minimum value minus epsilon

Default: 0

Epsilon [number] <put parameter description here>

Default: 1.0

Outputs

Processed DEM [raster] <put output description here>

Console usage

```
processing.runalg('saga:burnstreamnetworkintodem', dem, stream, method, epsilon, burn)
```

See also

Catchment area (flow tracing)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Sink Routes [raster] Optional.

<put parameter description here>

Weight [raster] Optional.

<put parameter description here>

Material [raster] Optional.

<put parameter description here>

Target [raster] Optional.

<put parameter description here>

Step [number] <put parameter description here>

Default: *1*

Method [selection] <put parameter description here>

Options:

- 0 — [0] Rho 8
- 1 — [1] Kinematic Routing Algorithm
- 2 — [2] DEMON

Default: *0*

DEMON - Min. DQV [number] <put parameter description here>

Default: *0.0*

Flow Correction [boolean] <put parameter description here>

Default: *True*

Outputs

Catchment Area [raster] <put output description here>

Catchment Height [raster] <put output description here>

Catchment Slope [raster] <put output description here>

Total accumulated Material [raster] <put output description here>

Accumulated Material from _left_ side [raster] <put output description here>

Accumulated Material from _right_ side [raster] <put output description here>

Console usage

```
processing.runalg('saga:catchmentareafLOWtracing', elevation, sinkroute, weight, material, target,
```

See also

Catchment area (recursive)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Sink Routes [raster] Optional.

<put parameter description here>

Weight [raster] Optional.

<put parameter description here>

Material [raster] Optional.

<put parameter description here>

Target [raster] Optional.

<put parameter description here>

Step [number] <put parameter description here>

Default: 1

Target Areas [raster] Optional.

<put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Rho 8
- 2 — [2] Deterministic Infinity
- 3 — [3] Multiple Flow Direction

Default: 0

Convergence [number] <put parameter description here>

Default: 1.1

Outputs

Catchment Area [raster] <put output description here>

Catchment Height [raster] <put output description here>

Catchment Slope [raster] <put output description here>

Total accumulated Material [raster] <put output description here>

Accumulated Material from _left_ side [raster] <put output description here>

Accumulated Material from _right_ side [raster] <put output description here>

Flow Path Length [raster] <put output description here>

Console usage

```
processing.runalg('saga:catchmentarearecursive', elevation, sinkroute, weight, material, target, ...)
```

See also

Catchment Area

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Rho 8
- 2 — [2] Braunschweiger Reliefmodell
- 3 — [3] Deterministic Infinity
- 4 — [4] Multiple Flow Direction
- 5 — [5] Multiple Triangular Flow Directon

Default: 0

Outputs

Catchment Area [raster] <put output description here>

Console usage

```
processing.runalg('saga:catchmentarea', elevation, method, carea)
```

See also

Cell balance

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Parameter [raster] Optional.

<put parameter description here>

Default Weight [number] <put parameter description here>

Default: *1.0*

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Multiple Flow Direction

Default: *0*

Outputs

Cell Balance [raster] <put output description here>

Console usage

```
processing.runalg('saga:cellbalance', dem, weights, weight, method, balance)
```

See also

Edge contamination

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Outputs

Edge Contamination [raster] <put output description here>

Console usage

```
processing.runalg('saga:edgecontamination', dem, contamination)
```

See also

Fill Sinks

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Minimum Slope [Degree] [number] <put parameter description here>

Default: *0.01*

Outputs

Filled DEM [raster] <put output description here>

Console usage

```
processing.runalg('saga:fillsinks', dem, minslope, result)
```

See also

Fill sinks (wang & liu)

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Minimum Slope [Degree] [number] <put parameter description here>

Default: *0.01*

Outputs

Filled DEM [raster] <put output description here>

Flow Directions [raster] <put output description here>

Watershed Basins [raster] <put output description here>

Console usage

```
processing.runalg('saga:fillsinkswangliu', elev, minslope, filled, fdir, wshed)
```

See also

Fill sinks xxi (wang & liu)

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Minimum Slope [Degree] [number] <put parameter description here>

Default: *0.01*

Outputs

Filled DEM [raster] <put output description here>

Console usage

```
processing.runalg('saga:fillsinksxxlwangliu', elev, minslope, filled)
```

See also

Flat detection

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Flat Area Values [selection] <put parameter description here>

Options:

- 0 — [0] elevation
- 1 — [1] enumeration

Default: *0*

Outputs

No Flats [raster] <put output description here>

Flat Areas [raster] <put output description here>

Console usage

```
processing.runalg('saga:flatdetection', dem, flat_output, noflats, flats)
```

See also

Flow path length

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Seeds [raster] Optional.

<put parameter description here>

Seeds Only [boolean] <put parameter description here>

Default: *True*

Flow Routing Algorithm [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8 (D8)
- 1 — [1] Multiple Flow Direction (FD8)

Default: *0*

Convergence (FD8) [number] <put parameter description here>

Default: *1.1*

Outputs

Flow Path Length [raster] <put output description here>

Console usage

```
processing.runalg('saga:flowpathlength', elevation, seed, seeds_only, method, convergence, length)
```


See also

Flow width and specific catchment area

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Total Catchment Area (TCA) [raster] Optional.

<put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Multiple Flow Direction (Quinn et al. 1991)
- 2 — [2] Aspect

Default: 0

Outputs

Flow Width [raster] <put output description here>

Specific Catchment Area (SCA) [raster] <put output description here>

Console usage

```
processing.runalg('saga:flowwidthandspecificcatchmentarea', dem, tca, method, width, sca)
```

See also

Lake flood

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Seeds [raster] <put parameter description here>

Absolute Water Levels [boolean] <put parameter description here>

Default: *True*

Outputs

Lake [raster] <put output description here>

Surface [raster] <put output description here>

Console usage

```
processing.runalg('saga:lakeflood', elev, seeds, level, outdepth, outlevel)
```

See also

Ls factor

Description

<put algorithm description here>

Parameters

Slope [raster] <put parameter description here>

Catchment Area [raster] <put parameter description here>

Area to Length Conversion [selection] <put parameter description here>

Options:

- 0 — [0] no conversion (areas already given as specific catchment area)
- 1 — [1] 1 / cell size (specific catchment area)
- 2 — [2] square root (catchment length)

Default: 0

Method (LS) [selection] <put parameter description here>

Options:

- 0 — [0] Moore et al. 1991
- 1 — [1] Desmet & Govers 1996
- 2 — [2] Bochner & Selige 2006

Default: 0

Rill/Interrill Erosivity [number] <put parameter description here>

Default: 0.0

Stability [selection] <put parameter description here>

Options:

- 0 — [0] stable
- 1 — [1] instable (thawing)

Default: 0

Outputs

LS Factor [raster] <put output description here>

Console usage

```
processing.runalg('saga:lsfactor', slope, area, conv, method, erosivity, stability, ls)
```

See also

Saga wetness index

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

t [number] <put parameter description here>

Default: *10*

Outputs

Catchment area [raster] <put output description here>

Catchment slope [raster] <put output description here>

Modified catchment area [raster] <put output description here>

Wetness index [raster] <put output description here>

Console usage

```
processing.runalg('saga:sagawetnessindex', dem, t, c, gn, cs, sb)
```

See also

Sink drainage route detection

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Threshold [boolean] <put parameter description here>

Default: *True*

Threshold Height [number] <put parameter description here>

Default: *100.0*

Outputs

Sink Route [raster] <put output description here>

Console usage

```
processing.runalg('saga:sinkdrainageroutedetection', elevation, threshold, thrsheight, sinkroute)
```

See also

Sink removal

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Sink Route [raster] Optional.

<put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deepen Drainage Routes
- 1 — [1] Fill Sinks

Default: *0*

Threshold [boolean] <put parameter description here>

Default: *True*

Threshold Height [number] <put parameter description here>

Default: *100.0*

Outputs

Preprocessed DEM [raster] <put output description here>

Console usage

```
processing.runalg('saga:sinkremoval', dem, sinkroute, method, threshold, thrsheight, dem_preproc)
```

See also

Slope length

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Outputs

Slope Length [raster] <put output description here>

Console usage

```
processing.runalg('saga:slopelength', dem, length)
```

See also

Stream power index

Description

<put algorithm description here>

Parameters

Slope [raster] <put parameter description here>

Catchment Area [raster] <put parameter description here>

Area Conversion [selection] <put parameter description here>

Options:

- 0 — [0] no conversion (areas already given as specific catchment area)
- 1 — [1] 1 / cell size (pseudo specific catchment area)

Default: 0

Outputs

Stream Power Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:streampowerindex', slope, area, conv, spi)
```

See also

Topographic wetness index (twi)

Description

<put algorithm description here>

Parameters

Slope [raster] <put parameter description here>

Catchment Area [raster] <put parameter description here>

Transmissivity [raster] Optional.

<put parameter description here>

Area Conversion [selection] <put parameter description here>

Options:

- 0 — [0] no conversion (areas already given as specific catchment area)
- 1 — [1] 1 / cell size (pseudo specific catchment area)

Default: 0

Method (TWI) [selection] <put parameter description here>

Options:

- 0 — [0] Standard
- 1 — [1] TOPMODEL

Default: 0

Outputs

Topographic Wetness Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:topographicwetnessindextwi', slope, area, trans, conv, method, twi)
```

See also

Upslope Area

Description

<put algorithm description here>

Parameters

Target Area [raster] Optional.

<put parameter description here>

Target X coordinate [number] <put parameter description here>

Default: *0.0*

Target Y coordinate [number] <put parameter description here>

Default: *0.0*

Elevation [raster] <put parameter description here>

Sink Routes [raster] Optional.

<put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Deterministic Infinity
- 2 — [2] Multiple Flow Direction

Default: *0*

Convergence [number] <put parameter description here>

Default: *1.1*

Outputs

Upslope Area [raster] <put output description here>

Console usage

```
processing.runalg('saga:upslopearea', target, target_pt_x, target_pt_y, elevation, sinkroute, met
```

See also

.

18.7.26 Terrain lighting

Analytical hillshading

Description

<put algorithm description here>

Parameters

Elevation [**raster**] <put parameter description here>

Shading Method [**selection**] <put parameter description here>

Options:

- 0 — [0] Standard
- 1 — [1] Standard (max. 90Degree)
- 2 — [2] Combined Shading
- 3 — [3] Ray Tracing

Default: 0

Azimuth [**Degree**] [**number**] <put parameter description here>

Default: 315.0

Declination [**Degree**] [**number**] <put parameter description here>

Default: 45.0

Exaggeration [**number**] <put parameter description here>

Default: 4.0

Outputs

Analytical Hillshading [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:analyticalhillshading', elevation, method, azimuth, declination, exaggeration)
```

See also

Sky view factor

Description

<put algorithm description here>

Parameters

Elevation [**raster**] <put parameter description here>

Maximum Search Radius [**number**] <put parameter description here>

Default: 10000

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] multi scale
- 1 — [1] sectors

Default: 0

Multi Scale Factor [number] <put parameter description here>

Default: 3

Number of Sectors [number] <put parameter description here>

Default: 8

Outputs

Visible Sky [raster] <put output description here>

Sky View Factor [raster] <put output description here>

Sky View Factor (Simplified) [raster] <put output description here>

Terrain View Factor [raster] <put output description here>

Console usage

```
processing.runalg('saga:skyviewfactor', dem, maxradius, method, level_inc, ndirs, visible, svf, s
```

See also

Topographic correction

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Original Image [raster] <put parameter description here>

Azimuth [number] <put parameter description here>

Default: 180.0

Height [number] <put parameter description here>

Default: 45.0

Method [selection] <put parameter description here>

Options:

- 0 — [0] Cosine Correction (Teillet et al. 1982)
- 1 — [1] Cosine Correction (Civco 1989)
- 2 — [2] Minnaert Correction
- 3 — [3] Minnaert Correction with Slope (Riano et al. 2003)
- 4 — [4] Minnaert Correction with Slope (Law & Nichol 2004)
- 5 — [5] C Correction
- 6 — [6] Normalization (after Civco, modified by Law & Nichol)

Default: 0

Minnaert Correction [number] <put parameter description here>

Default: *0.5*

Maximum Cells (C Correction Analysis) [number] <put parameter description here>

Default: *1000*

Value Range [selection] <put parameter description here>

Options:

- 0 — [0] 1 byte (0-255)
- 1 — [1] 2 byte (0-65535)

Default: *0*

Outputs

Corrected Image [raster] <put output description here>

Console usage

```
processing.runalg('saga:topographiccorrection', dem, original, azi, hgt, method, minnaert, maxcel.
```

See also

.

18.7.27 Terrain morphometry

Convergence index

Description

Calculates an index of convergence/divergence regarding to overland flow. By its meaning it is similar to plan or horizontal curvature, but gives much smoother results. The calculation uses the aspects of surrounding cells, i.e. it looks to which degree surrounding cells point to the center cell. The result is given as percentages, negative values correspond to convergent, positive to divergent flow conditions. Minus 100 would be like a peak of a cone, plus 100 a pit, and 0 an even slope.

Parameters

Elevation [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Aspect
- 1 — [1] Gradient

Default: *0*

Gradient Calculation [selection] <put parameter description here>

Options:

- 0 — [0] 2 x 2

- 1 — [1] 3 x 3

Default: 0

Outputs

Convergence Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:convergenceindex', elevation, method, neighbours, result)
```

See also

- Koethe, R. / Lehmeier, F. (1996): 'SARA, System zur Automatischen Relief-Analyse', Benutzerhandbuch, 2. Auflage [Geogr. Inst. Univ. Goettingen, unpublished]

Convergence index (search radius)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Radius [Cells] [number] <put parameter description here>

Default: 10

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1

Gradient [boolean] <put parameter description here>

Default: *True*

Difference [selection] <put parameter description here>

Options:

- 0 — [0] direction to the center cell
- 1 — [1] center cell's aspect direction

Default: 0

Outputs

Convergence Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:convergenceindexsearchradius', elevation, radius, distance_weighting_weight)
```

See also

Curvature classification

Description

<put algorithm description here>

Parameters

Plan Curvature [raster] <put parameter description here>

Profile Curvature [raster] <put parameter description here>

Threshold for plane [number] <put parameter description here>

Default: 0.001

Outputs

Curvature Classification [raster] <put output description here>

Console usage

```
processing.runalg('saga:curvatureclassification', cplan, cprof, threshold, class)
```

See also

Diurnal anisotropic heating

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Alpha Max (Degree) [number] <put parameter description here>

Default: *202.5*

Outputs

Diurnal Anisotropic Heating [raster] <put output description here>

Console usage

```
processing.runalg('saga:diurnalanisotropicheating', dem, alpha_max, dah)
```

See also

Downslope distance gradient

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Vertical Distance [number] <put parameter description here>

Default: *10*

Output [selection] <put parameter description here>

Options:

- 0 — [0] distance
- 1 — [1] gradient (tangens)
- 2 — [2] gradient (degree)

Default: *0*

Outputs

Gradient [raster] <put output description here>

Gradient Difference [raster] <put output description here>

Console usage

```
processing.runalg('saga:downslopedistancegradient', dem, distance, output, gradient, difference)
```

See also

Effective air flow heights

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Wind Direction [raster] Optional.

<put parameter description here>

Wind Speed [raster] Optional.

<put parameter description here>

Constant Wind Direction [Degree] [number] <put parameter description here>

Default: *135*

Old Version [boolean] <put parameter description here>

Default: *True*

Search Distance [km] [number] <put parameter description here>

Default: *300*

Acceleration [number] <put parameter description here>

Default: *1.5*

Use Pyramids with New Version [boolean] <put parameter description here>

Default: *True*

Lee Factor [number] <put parameter description here>

Default: *0.5*

Luv Factor [number] <put parameter description here>

Default: *1.0*

Wind Direction Units [selection] <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: *0*

Wind Speed Scale Factor [number] <put parameter description here>

Default: *1.0*

Outputs

Effective Air Flow Heights [raster] <put output description here>

Console usage

```
processing.runalg('saga:effectiveairflowheights', dem, dir, len, dir_const, oldver, maxdist, acce
```

See also

Hypsometry

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Number of Classes [number] <put parameter description here>

Default: *100.0*

Sort [selection] <put parameter description here>

Options:

- 0 — [0] up
- 1 — [1] down

Default: *0*

Classification Constant [selection] <put parameter description here>

Options:

- 0 — [0] height
- 1 — [1] area

Default: *0*

Use Z-Range [boolean] <put parameter description here>

Default: *True*

Z-Range Min [number] <put parameter description here>

Default: *0.0*

Z-Range Max [number] <put parameter description here>

Default: *1000.0*

Outputs

Hypsometry [table] <put output description here>

Console usage

```
processing.runalg('saga:hypsometry', elevation, count, sorting, method, bzrange, zrange_min, zran
```

See also

Land surface temperature

Description

<put algorithm description here>

Parameters

Elevation [m] [raster] <put parameter description here>

Short Wave Radiation [kW/m2] [raster] <put parameter description here>

Leaf Area Index [raster] <put parameter description here>

Elevation at Reference Station [m] [number] <put parameter description here>

Default: 0.0

Temperature at Reference Station [Deg.Celsius] [number] <put parameter description here>

Default: 0.0

Temperature Gradient [Deg.Celsius/km] [number] <put parameter description here>

Default: 6.5

C Factor [number] <put parameter description here>

Default: 1.0

Outputs

Land Surface Temperature [Deg.Celsius] [raster] <put output description here>

Console usage

```
processing.runalg('saga:landsurfacetemperature', dem, swr, lai, z_reference, t_reference, t_gradient)
```

See also

Mass balance index

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Vertical Distance to Channel Network [raster] Optional.

<put parameter description here>

T Slope [number] <put parameter description here>

Default: *15.0*

T Curvature [number] <put parameter description here>

Default: *0.01*

T Vertical Distance to Channel Network [number] <put parameter description here>

Default: *15.0*

Outputs

Mass Balance Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:massbalanceindex', dem, hrel, tslope, tcurve, threl, mbi)
```

See also

Morphometric protection index

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Radius [number] <put parameter description here>

Default: *2000.0*

Outputs

Protection Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:morphometricprotectionindex', dem, radius, protection)
```

See also

Multiresolution index of valley bottom flatness (mrvbf)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Initial Threshold for Slope [number] <put parameter description here>

Default: *16*

Threshold for Elevation Percentile (Lowness) [number] <put parameter description here>

Default: *0.4*

Threshold for Elevation Percentile (Upness) [number] <put parameter description here>

Default: *0.35*

Shape Parameter for Slope [number] <put parameter description here>

Default: *4.0*

Shape Parameter for Elevation Percentile [number] <put parameter description here>

Default: *3.0*

Update Views [boolean] <put parameter description here>

Default: *True*

Classify [boolean] <put parameter description here>

Default: *True*

Maximum Resolution (Percentage) [number] <put parameter description here>

Default: *100*

Outputs

MRVBF [raster] <put output description here>

MRRTF [raster] <put output description here>

Console usage

```
processing.runalg('saga:multiresolutionindexofvalleybottomflatnessmrvbf', dem, t_slope, t_pctl_v,
```

See also

Real area calculation

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Outputs

Real Area Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:realareacalculation', dem, area)
```

See also

Relative heights and slope positions

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

w [number] <put parameter description here>

Default: *0.5*

t [number] <put parameter description here>

Default: *10.0*

e [number] <put parameter description here>

Default: *2.0*

Outputs

Slope Height [raster] <put output description here>

Valley Depth [raster] <put output description here>

Normalized Height [raster] <put output description here>

Standardized Height [raster] <put output description here>

Mid-Slope Positon [raster] <put output description here>

Console usage

```
processing.runalg('saga:relativeheightsandslopepositions', dem, w, t, e, ho, hu, nh, sh, ms)
```

See also

Slope, aspect, curvature

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Maximum Slope (Travis et al. 1975)
- 1 — [1] Maximum Triangle Slope (Tarboton 1997)
- 2 — [2] Least Squares Fitted Plane (Horn 1981, Costa-Cabral & Burgess 1996)
- 3 — [3] Fit 2.Degree Polynom (Bauer, Rohdenburg, Bork 1985)
- 4 — [4] Fit 2.Degree Polynom (Heerdegen & Beran 1982)
- 5 — [5] Fit 2.Degree Polynom (Zevenbergen & Thorne 1987)
- 6 — [6] Fit 3.Degree Polynom (Haralick 1983)

Default: 5

Outputs

Slope [raster] <put output description here>

Aspect [raster] <put output description here>

Curvature [raster] <put output description here>

Plan Curvature [raster] <put output description here>

Profile Curvature [raster] <put output description here>

Console usage

```
processing.runalg('saga:slopeaspectcurvature', elevation, method, slope, aspect, curv, hcurv, vcurv)
```

See also

Surface specific points

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Mark Highest Neighbour
- 1 — [1] Opposite Neighbours
- 2 — [2] Flow Direction
- 3 — [3] Flow Direction (up and down)

- 4 — [4] Peucker & Douglas

Default: 0

Threshold [number] <put parameter description here>

Default: 2.0

Outputs

Result [raster] <put output description here>

Console usage

```
processing.runalg('saga:surfacespecificpoints', elevation, method, threshold, result)
```

See also

Terrain ruggedness index (tri)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Radius (Cells) [number] <put parameter description here>

Default: 1

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1.0

Outputs

Terrain Ruggedness Index (TRI) [raster] <put output description here>

Console usage

```
processing.runalg('saga:terrainruggednessindextri', dem, radius, distance_weighting_weighting, di
```

See also

Topographic position index (tpi)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Standardize [boolean] <put parameter description here>

Default: *True*

Min Radius [number] <put parameter description here>

Default: *0.0*

Max Radius [number] <put parameter description here>

Default: *100.0*

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *75.0*

Outputs

Topographic Position Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:topographicpositionindextpi', dem, standard, radius_min, radius_max, dist
```

See also

Tpi based landform classification

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Min Radius A [number] <put parameter description here>

Default: 0

Max Radius A [number] <put parameter description here>

Default: 100

Min Radius B [number] <put parameter description here>

Default: 0

Max Radius B [number] <put parameter description here>

Default: 1000

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: True

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 75.0

Outputs

Landforms [raster] <put output description here>

Console usage

```
processing.runalg('saga:tpibasedlandformclassification', dem, radius_a_min, radius_a_max, radius_b_min, radius_b_max, distance_weighting, inverse_distance_weighting_power, inverse_distance_offset, gaussian_and_exponential_weighting_bandwidth)
```

See also

Vector ruggedness measure (vrm)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Radius (Cells) [number] <put parameter description here>

Default: *1*

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *1*

Outputs

Vector Terrain Ruggedness (VRM) [raster] <put output description here>

Console usage

```
processing.runalg('saga:vectorruggednessmeasurevrm', dem, radius, distance_weighting_weighting, d
```

See also

Wind effect

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Wind Direction [raster] Optional.

<put parameter description here>

Wind Speed [raster] Optional.

<put parameter description here>

Constant Wind Direction [Degree] [number] <put parameter description here>

Default: *135*

Old Version [boolean] <put parameter description here>

Default: *True*

Search Distance [km] [number] <put parameter description here>

Default: *300.0*

Acceleration [number] <put parameter description here>

Default: *1.5*

Use Pyramids [boolean] <put parameter description here>

Default: *True*

Wind Direction Units [selection] <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: *0*

Wind Speed Scale Factor [number] <put parameter description here>

Default: *1.0*

Outputs

Wind Effect [raster] <put output description here>

Windward Effect [raster] <put output description here>

Leeward Effect [raster] <put output description here>

Console usage

```
processing.runalg('saga:windeffect', dem, dir, len, dir_const, oldver, maxdist, accel, pyramids, o
```

See also

.

18.7.28 Terrain profiles

Cross profiles

Description

<put algorithm description here>

Parameters

DEM [**raster**] <put parameter description here>

Lines [**vector: line**] <put parameter description here>

Profile Distance [**number**] <put parameter description here>

Default: *10.0*

Profile Length [**number**] <put parameter description here>

Default: *10.0*

Profile Samples [**number**] <put parameter description here>

Default: *10.0*

Outputs

Cross Profiles [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:crossprofiles', dem, lines, dist_line, dist_profile, num_profile, profile)
```

See also

Profile from points table

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Input [**table**] <put parameter description here>

X [**tablefield: any**] <put parameter description here>

Y [**tablefield: any**] <put parameter description here>

Outputs

Result [**table**] <put output description here>

Console usage

```
processing.runalg('saga:profilefrompointstable', grid, table, x, y, result)
```

See also

Profiles from lines

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Values [multipleinput: rasters] Optional.

<put parameter description here>

Lines [vector: line] <put parameter description here>

Name [tablefield: any] <put parameter description here>

Each Line as new Profile [boolean] <put parameter description here>

Default: *True*

Outputs

Profiles [vector] <put output description here>

Profiles [vector] <put output description here>

Console usage

```
processing.runalg('saga:profilesfromlines', dem, values, lines, name, split, profile, profiles)
```

See also

.

18.8 TauDEM algorithm provider

TauDEM (Terrain Analysis Using Digital Elevation Models) is a set of Digital Elevation Model (DEM) tools for the extraction and analysis of hydrologic information from topography as represented by a DEM. This is software developed at Utah State University (USU) for hydrologic digital elevation model analysis and watershed delineation.

TauDEM is distributed as a set of standalone command line executable programs for a Windows and source code for compiling and use on other systems.

Note: Please remember that Processing contains only the interface description, so you need to install TauDEM 5.0.6 by yourself and configure Processing properly.

Documentation for TauDEM algorithms derived from official [TauDEM documentation](#)

18.8.1 Basic Grid Analysis

D8 Contributing Area

Description

Calculates a grid of contributing areas using the single direction D8 flow model. The contribution of each grid cell is taken as one (or when the optional weight grid is used, the value from the weight grid). The contributing area for each grid cell is taken as its own contribution plus the contribution from upslope neighbors that drain in to it according to the D8 flow model.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D8 flow model) of them are in the domain to be evaluated.

By default, the tool checks for edge contamination. This is defined as the possibility that a contributing area value may be underestimated due to grid cells outside of the domain not being counted. This occurs when drainage is inwards from the boundaries or areas with “no data” values for elevation. The algorithm recognizes this and reports “no data” for the contributing area. It is common to see streaks of “no data” values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge contamination checking may be turned off in cases where you know this is not an issue or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Parameters

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Outlets Shapefile [vector: point] Optional.

A point shapefile defining the outlets of interest. If this input file is used, only the cells upslope of these outlet cells are considered to be within the domain being evaluated.

Weight Grid [raster] Optional.

A grid giving contribution to flow for each cell. These contributions (also sometimes referred to as weights or loadings) are used in the contributing area accumulation. If this input file is not used, the contribution to flow will assumed to be one for each grid cell.

Check for edge contamination [boolean] A flag that indicates whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a contributing area value may be underestimated due to the fact that grid cells outside of the domain have not been evaluated. This occurs when drainage is inwards from the boundaries or areas with NODATA values for elevation. The algorithm recognizes this and reports NODATA for the impated cells. It is common to see streaks of NODATA values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of available data. Edge contamination checking may be turned off in cases where you know this is not an issue, or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Default: *True*

Outputs

D8 Contributing Area Grid [raster] A grid of contributing area values calculated as the cells own contribution plus the contribution from upslope neighbors that drain in to it according to the D8 flow model.

Console usage

```
processing.runalg('taudem:d8contributingarea', -p, -o, -wg, -nc, -ad8)
```

See also

D8 Flow Directions

Description

Creates 2 grids. The first contains the flow direction from each grid cell to one of its adjacent or diagonal neighbors, calculated using the direction of steepest descent. The second contain the slope, as evaluated in the direction of steepest descent, and is reported as drop/distance, i.e. tan of the angle. Flow direction is reported as NODATA for any grid cell adjacent to the edge of the DEM domain, or adjacent to a NODATA value in the DEM. In flat areas, flow directions are assigned away from higher ground and towards lower ground using the method of Garbrecht and Martz (1997). The D8 flow direction algorithm may be applied to a DEM that has not had its pits filled, but it will then result in NODATA values for flow direction and slope at the lowest point of each pit.

D8 Flow Direction Coding:

- 1 — East
- 2 — Northeast
- 3 — North
- 4 — Northwest
- 5 — West
- 6 — Southwest
- 7 — South
- 8 — Southeast

The flow direction routing across flat areas is performed according to the method described by Garbrecht, J. and L. W. Martz, (1997), “The Assignment of Drainage Direction Over Flat Surfaces in Raster Digital Elevation Models”, *Journal of Hydrology*, 193: 204-213.

Parameters

Pit Filled Elevation Grid [raster] A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed. Pits are low elevation areas in digital elevation models (DEMs) that are completely surrounded by higher terrain. They are generally taken to be artifacts of the digitation process that interfere with the processing of flow across DEMs. So they are removed by raising their elevation to the point where they just drain off the domain. This step is not essential if you have reason to believe that the pits in your DEM are real. If a few pits actually exist and so should not be removed, while at the same time others are believed to be artifacts that need to be removed, the actual pits should have NODATA elevation values inserted at their lowest point. NODATA values serve to define edges of the domain in the flow field, and elevations are only raised to where flow is off an edge, so an internal NODATA value will stop a pit from being removed, if necessary.

Outputs

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope.

D8 Slope Grid [raster] A grid giving slope in the D8 flow direction. This is measured as drop/distance.

Console usage

```
processing.runalg('taudem:d8flowdirections', -fel, -p, -sd8)
```

See also

D-Infinity Contributing Area

Description

Calculates a grid of specific catchment area which is the contributing area per unit contour length using the multiple flow direction D-infinity approach. D-infinity flow direction is defined as steepest downward slope on planar triangular facets on a block centered grid. The contribution at each grid cell is taken as the grid cell length (or when the optional weight grid input is used, from the weight grid). The contributing area of each grid cell is then taken as its own contribution plus the contribution from upslope neighbors that have some fraction draining to it according to the D-infinity flow model. The flow from each cell either all drains to one neighbor, if the angle falls along a cardinal ($0, \pi/2, \pi, 3\pi/2$) or ordinal ($\pi/4, 3\pi/4, 5\pi/4, 7\pi/4$) direction, or is on an angle falling between the direct angle to two adjacent neighbors. In the latter case the flow is proportioned between these two neighbor cells according to how close the flow direction angle is to the direct angle to those cells. The contour length used here is the grid cell size. The resulting units of the specific catchment area are length units the same as those of the grid cell size.

When the optional weight grid is not used, the result is reported in terms of specific catchment area, the upslope area per unit contour length, taken here as the number of cells times grid cell length (cell area divided by cell length). This assumes that grid cell length is the effective contour length, in the definition of specific catchment area and does not distinguish any difference in contour length dependent upon the flow direction. When the optional weight grid is used, the result is reported directly as a summation of weights, without any scaling.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D-infinity flow model) of them are in the domain to be evaluated.

By default, the tool checks for edge contamination. This is defined as the possibility that a contributing area value may be underestimated due to grid cells outside of the domain not being counted. This occurs when drainage is inwards from the boundaries or areas with “no data” values for elevation. The algorithm recognizes this and reports “no data” for the contributing area. It is common to see streaks of “no data” values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge contamination checking may be turned off in cases where you know it is not an issue or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Parameters

D-Infinity Flow Direction Grid [raster] A grid of flow directions based on the D-infinity flow method using the steepest slope of a triangular facet. Flow direction is determined as the direction of the steepest downward slope on the 8 triangular facets of a 3x3 block centered grid. Flow direction is encoded as an angle in radians, counter-clockwise from east as a continuous (floating point) quantity between 0 and 2π . The resulting flow in a grid is then usually interpreted as being proportioned between the two neighboring cells that define the triangular facet with the steepest downward slope.

Outlets Shapefile [vector: point] Optional.

A point shapefile defining the outlets of interest. If this input file is used, only the cells upslope of these outlet cells are considered to be within the domain being evaluated.

Weight Grid [raster] Optional.

A grid giving contribution to flow for each cell. These contributions (also sometimes referred to as weights or loadings) are used in the contributing area accumulation. If this input file is not used, the result is reported in terms of specific catchment area (the upslope area per unit contour length) taken as the number of cells times grid cell length (cell area divided by cell length).

Check for edge contamination [boolean] A flag that indicates whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a contributing area value may be underestimated due to the fact that grid cells outside of the domain have not been evaluated. This occurs when drainage is inwards from the boundaries or areas with NODATA values for elevation. The algorithm recognizes this and reports NODATA for the impated cells. It is common to see streaks of NODATA values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of available data. Edge contamination checking may be turned off in cases where you know this is not an issue, or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Default: *True*

Outputs

D-Infinity Specific Catchment Area Grid [raster] A grid of specific catchment area which is the contributing area per unit contour length using the multiple flow direction D-infinity approach. The contributing area of each grid cell is then taken as its own contribution plus the contribution from upslope neighbors that have some fraction draining to it according to the D-infinity flow model.

Console usage

```
processing.runalg('taudem:dinfinitecontributingarea', -ang, -o, -wg, -nc, -sca)
```

See also

D-Infinity Flow Directions

Description

Assigns a flow direction based on the D-infinity flow method using the steepest slope of a triangular facet (Tarboton, 1997, "A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models", Water Resources Research, 33(2): 309-319). Flow direction is defined as steepest downward slope on planar triangular facets on a block centered grid. Flow direction is encoded as an angle in radians counter-clockwise from east as a continuous (floating point) quantity between 0 and 2π . The flow direction angle is determined as the direction of the steepest downward slope on the eight triangular facets formed in a 3 x 3 grid cell window centered on the grid cell of interest. The resulting flow in a grid is then usually interpreted as being proportioned between the two neighboring cells that define the triangular facet with the steepest downward slope.

A block-centered representation is used with each elevation value taken to represent the elevation of the center of the corresponding grid cell. Eight planar triangular facets are formed between each grid cell and its eight neighbors. Each of these has a downslope vector which when drawn outwards from the center may be at an angle that lies within or outside the 45 degree ($\pi/4$ radian) angle range of the facet at the center point. If the slope vector

angle is within the facet angle, it represents the steepest flow direction on that facet. If the slope vector angle is outside a facet, the steepest flow direction associated with that facet is taken along the steepest edge. The slope and flow direction associated with the grid cell is taken as the magnitude and direction of the steepest downslope vector from all eight facets. Slope is measured as drop/distance, i.e. \tan of the slope angle.

In the case where no slope vectors are positive (downslope), the flow direction is set using the method of Garbrecht and Martz (1997) for the determination of flow across flat areas. This makes flat areas drain away from high ground and towards low ground. The flow path grid to enforce drainage along existing streams is an optional input, and if used, takes precedence over elevations for the setting of flow directions.

The D-infinity flow direction algorithm may be applied to a DEM that has not had its pits filled, but it will then result in “no data” values for the D-infinity flow direction and slope associated with the lowest point of the pit.

Parameters

Pit Filled Elevation Grid [raster] A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed.

Outputs

D-Infinity Flow Directions Grid [raster] A grid of flow directions based on the D-infinity flow method using the steepest slope of a triangular facet. Flow direction is determined as the direction of the steepest downward slope on the 8 triangular facets of a 3x3 block centered grid. Flow direction is encoded as an angle in radians, counter-clockwise from east as a continuous (floating point) quantity between 0 and 2π . The resulting flow in a grid is then usually interpreted as being proportioned between the two neighboring cells that define the triangular facet with the steepest downward slope.

D-Infinity Slope Grid [raster] A grid of slope evaluated using the D-infinity method described in Tarboton, D. G., (1997), “A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models”, *Water Resources Research*, 33(2): 309-319. This is the steepest outwards slope on one of eight triangular facets centered at each grid cell, measured as drop/distance, i.e. \tan of the slope angle.

Console usage

```
processing.runalg('taudem:dinfiniteflowdirections', -fel, -ang, -slp)
```

See also

Grid Network

Description

Creates 3 grids that contain for each grid cell: 1) the longest path, 2) the total path, and 3) the Strahler order number. These values are derived from the network defined by the D8 flow model.

The longest upslope length is the length of the flow path from the furthest cell that drains to each cell. The total upslope path length is the length of the entire grid network upslope of each grid cell. Lengths are measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal.

Strahler order is defined as follows: A network of flow paths is defined by the D8 Flow Direction grid. Source flow paths have a Strahler order number of one. When two flow paths of different order join the order of the downstream flow path is the order of the highest incoming flow path. When two flow paths of equal order join the downstream flow path order is increased by 1. When more than two flow paths join the downstream flow path order is calculated as the maximum of the highest incoming flow path order or the second highest incoming flow path order + 1. This generalizes the common definition to cases where more than two flow paths join at a point.

Where the optional mask grid and threshold value are input, the function is evaluated only considering grid cells that lie in the domain with mask grid value greater than or equal to the threshold value. Source (first order) grid cells are taken as those that do not have any other grid cells from inside the domain draining in to them, and only when two of these flow paths join is order propagated according to the ordering rules. Lengths are also only evaluated counting paths within the domain greater than or equal to the threshold.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D8 flow model) of them are in the domain to be evaluated.

Parameters

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Outlets Shapefile [vector: point] Optional.

A point shapefile defining the outlets of interest. If this input file is used, only the cells upslope of these outlet cells are considered to be within the domain being evaluated.

Mask Grid [raster] Optional.

A grid that is used to determine the domain to be analyzed. If the mask grid value \geq mask threshold (see below), then the cell will be included in the domain. While this tool does not have an edge contamination flag, if edge contamination analysis is needed, then a mask grid from a function like “**D8 Contributing Area**” that does support edge contamination can be used to achieve the same result.

Mask Threshold [number] This input parameter is used in the calculation mask grid value \geq mask threshold to determine if the grid cell is in the domain to be analyzed.

Default: *100*

Outputs

Longest Upslope Length Grid [raster] A grid that gives the length of the longest upslope D8 flow path terminating at each grid cell. Lengths are measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal.

Total Upslope Length Grid [raster] The total upslope path length is the length of the entire D8 flow grid network upslope of each grid cell. Lengths are measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal.

Strahler Network Order Grid [raster] A grid giving the Strahler order number for each cell. A network of flow paths is defined by the D8 Flow Direction grid. Source flow paths have a Strahler order number of one. When two flow paths of different order join the order of the downstream flow path is the order of the highest incoming flow path. When two flow paths of equal order join the downstream flow path order is increased by 1. When more than two flow paths join the downstream flow path order is calculated as the maximum of the highest incoming flow path order or the second highest incoming flow path order + 1. This generalizes the common definition to cases where more than two flow paths join at a point.

Console usage

```
processing.runalg('taudem:gridnetwork', d8_flow_dir_grid, outlets_shape, mask_grid, threshold, lo
```

See also

Pit Remove

Description

Identifies all pits in the DEM and raises their elevation to the level of the lowest pour point around their edge. Pits are low elevation areas in digital elevation models (DEMs) that are completely surrounded by higher terrain. They are generally taken to be artifacts that interfere with the routing of flow across DEMs, so are removed by raising their elevation to the point where they drain off the edge of the domain. The pour point is the lowest point on the boundary of the “watershed” draining to the pit. This step is not essential if you have reason to believe that the pits in your DEM are real. If a few pits actually exist and so should not be removed, while at the same time others are believed to be artifacts that need to be removed, the actual pits should have NODATA elevation values inserted at their lowest point. NODATA values serve to define edges in the domain, and elevations are only raised to where flow is off an edge, so an internal NODATA value will stop a pit from being removed, if necessary.

Parameters

Elevation Grid [raster] A digital elevation model (DEM) grid to serve as the base input for the terrain analysis and stream delineation.

Outputs

Pit Removed Elevation Grid [raster] A grid of elevation values with pits removed so that flow is routed off of the domain.

Console usage

```
processing.runalg('taudem:pitremove', -z, -fel)
```

See also

.

18.8.2 Specialized Grid Analysis

D8 Distance To Streams

Description

Computes the horizontal distance to stream for each grid cell, moving downslope according to the D8 flow model, until a stream grid cell is encountered.

Parameters

D8 Flow Direction Grid [raster] This input is a grid of flow directions that are encoded using the D8 method where all flow from a cells goes to a single neighboring cell in the direction of steepest descent. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Stream Raster Grid [raster] A grid indicating streams. Such a grid can be created by several of the tools in the “**Stream Network Analysis**” toolset. However, the tools in the “**Stream Network Analysis**” toolset only create grids with a value of 0 for no stream, or 1 for stream cells. This tool can also accept grids with values greater than 1, which can be used in conjunction with the `Threshold` parameter to determine the location of streams. This allows Contributing Area grids to be used to define streams as well as the normal Stream Raster grids. This grid expects integer (long integer) values and any non-integer values will be truncated to an integer before being evaluated.

Threshold [number] This value acts as threshold on the `Stream Raster Grid` to determine the location of streams. Cells with a `Stream Raster Grid` value greater than or equal to the `Threshold` value are interpreted as streams.

Default: 50

Outputs

Output Distance to Streams [raster] A grid giving the horizontal distance along the flow path as defined by the D8 Flow Directions Grid to the streams in the Stream Raster Grid.

Console usage

```
processing.runalg('taudem:d8distancetostreams', -p, -src, -thresh, -dist)
```

See also

D-Infinity Avalanche Runout

Description

Identifies an avalanche’s affected area and the flow path length to each cell in that affected area. All cells downslope from each source area cell, up to the point where the slope from the source to the affected area is less than a threshold angle called the Alpha Angle can be in the affected area. This tool uses the D-infinity multiple flow direction method for determining flow direction. This will likely cause very small amounts of flow to be dispersed to some downslope cells that might overstate the affected area, so a threshold proportion can be set to avoid this excess dispersion. The flow path length is the distance from the cell in question to the source cell that has the highest angle.

All points downslope from the source area are potentially in the affected area, but not beyond a point where the slope from the source to the affected area is less than a threshold angle called the Alpha Angle.

Slope is to be measured using the straight line distance from source point to evaluation point.

It makes more physical sense to me for the angle to be measured along the flow path. Nevertheless it is equally easy to code straight line angles as angles along the flow path, so an option that allows switching will be provided. The most practical way to evaluate avalanche runout is to keep track of the source point with the greatest angle to each point. Then the recursive upslope flow algebra approach will look at a grid cell and all its upslope neighbors that flow to it. Information from the upslope neighbors will be used to calculate the angle to the grid cell in question and retain it in the runout zone if the angle exceeds the alpha angle. This procedure makes the assumption that the maximum angle at a grid cell will be from the set of cells that have maximum angles to the inflowing neighbors. This will always be true of angle is calculated along a flow path, but I can conceive of cases where flow paths bend back on themselves where this would not be the case for straight line angles.

The D-infinity multiple flow direction field assigns flow from each grid cell to multiple downslope neighbors using proportions (P_{ik}) that vary between 0 and 1 and sum to 1 for all flows out of a grid cell. It may be desirable to specify a threshold T that this proportion has to exceed before a grid cell is counted as flowing to a downslope

grid cell, e.g. $P_{ik} > T$ ($=0.2$ say) to avoid dispersion to grid cells that get very little flow. T will be specified as a user input. If all upslope grid cells are to be used T may be input as 0.

Avalanche source sites are to be input as a short integer grid (name suffix **ass*, e.g. *demass*) comprised of positive values where avalanches may be triggered and 0 values elsewhere.

The following grids are output:

- *rz* — A runout zone indicator with value 0 to indicate that this grid cell is not in the runout zone and value > 0 to indicate that this grid cell is in the runout zone. Since there may be information in the angle to the associated source site, this variable will be assigned the angle to the source site (in degrees)
- *dm* — Along flow distance from the source site that has the highest angle to the point in question

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Pit Filled Elevation Grid [raster] This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

Avalanche Source Site Grid [raster] This is a grid of source areas for snow avalanches that are commonly identified manually using a mix of experience and visual interpretation of maps. Avalanche source sites are to be input as a short integer grid (name suffix **ass*, e.g. *demass*) comprised of positive values where avalanches may be triggered and 0 values elsewhere.

Proportion Threshold [number] This value is a threshold proportion that is used to limit the dispersion of flow caused by using the D-infinity multiple flow direction method for determining flow direction. The D-infinity multiple flow direction method often causes very small amounts of flow to be dispersed to some downslope cells that might overstate the affected area, so a threshold proportion can be set to avoid this excess dispersion.

Default: *0.2*

Alpha Angle Threshold [number] This value is the threshold angle, called the Alpha Angle, that is used to determine which of the cells downslope from the source cells are in the affected area. Only the cells downslope from each source area cell, up to the point where the slope from the source to the affected area is less than a threshold angle are in the affected area.

Default: *18*

Measure distance along flow path [boolean] This option selects the method used to measure the distance used to calculate the slope angle. If option is *True* then measure it along the flow path, where the *False* option causes the slope to be measure along the straight line distance from the source cell to the evaluation cell.

Default: *True*

Outputs

Runout Zone Grid [raster] This grid Identifies the avalanche’s runout zone (affected area) using a runout zone indicator with value 0 to indicate that this grid cell is not in the runout zone and value > 0 to indicate that this grid cell is in the runout zone. Since there may be information in the angle to the associated source site, this variable will be assigned the angle to the source site (in degrees).

Path Distance Grid [raster] This is a grid of the flow distance from the source site that has the highest angle to each cell.

Console usage

```
processing.runalg('taudem:dinfinityavalancherunout', -ang, -fel, -ass, -thresh, -alpha, -direct, ...)
```

See also

D-Infinity Concentration Limited Accumulation

Description

This function applies to the situation where an unlimited supply of a substance is loaded into flow at a concentration or solubility threshold C_{sol} over a region indicated by an indicator grid (dg). It a grid of the concentration of a substance at each location in the domain, where the supply of substance from a supply area is loaded into the flow at a concentration or solubility threshold. The flow is first calculated as a D-infinity weighted contributing area of an input Effective Runoff Weight Grid (notionally excess precipitation). The concentration of substance over the supply area (indicator grid) is at the concentration threshold. As the substance moves downslope with the D-infinity flow field, it is subject to first order decay in moving from cell to cell as well as dilution due to changes in flow. The decay multiplier grid gives the fractional (first order) reduction in quantity in moving from grid cell x to the next downslope cell. If the outlets shapefile is used, the tool only evaluates the part of the domain that contributes flow to the locations given by the shapefile. This is useful for a tracking a contaminant or compound from an area with unlimited supply of that compound that is loaded into a flow at a concentration or solubility threshold over a zone and flow from the zone may be subject to decay or attenuation.

The indicator grid (dg) is used to delineate the area of the substance supply using the (0, 1) indicator function $i(x)$. $A[]$ denotes the weighted accumulation operator evaluated using the D-Infinity Contributing Area function. The Effective Runoff Weight Grid gives the supply to the flow (e.g. the excess rainfall if this is overland flow) denoted as $w(x)$. The specific discharge is then given by:

$$Q(x) = A[w(x)]$$

This weighted accumulation $Q(x)$ is output as the Overland Flow Specific Discharge Grid. Over the substance supply area concentration is at the threshold (the threshold is a saturation or solubility limit). If $i(x) = 1$, then

$$C(x) = C_{sol}, \text{ and } L(x) = C_{sol} Q(x),$$

where $L(x)$ denotes the load being carried by the flow. At remaining locations, the load is determined by load accumulation and the concentration by dilution:

Here $d(x) = d(i, j)$ is a decay multiplier giving the fractional (first order) reduction in mass in moving from grid cell x to the next downslope cell. If travel (or residence) times $t(x)$ associated with flow between cells are available $d(x)$ may be evaluated as $\exp(-k t(x))$ where k is a first order decay parameter. The Concentration grid output is $C(x)$. If the outlets shapefile is used, the tool only evaluates the part of the domain that contributes flow to the locations given by the shapefile.

Useful for a tracking a contaminant released or partitioned to flow at a fixed threshold concentration.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This grid can be created by the function “D-Infinity Flow Directions”.

Disturbance Indicator Grid [raster] A grid that indicates the source zone of the area of substance supply and must be 1 inside the zone and 0 or NODATA over the rest of the domain.

Decay Multiplier Grid [raster] A grid giving the factor by which flow leaving each grid cell is multiplied before accumulation on downslope grid cells. This may be used to simulate the movement of an attenuating or decaying substance. If travel (or residence) times $t(x)$ associated with flow between cells are available $d(x)$ may be evaluated as $\exp(-k t(x))$ where k is a first order decay parameter.

Effective Runoff Weight Grid [raster] A grid giving the input quantity (notionally effective runoff or excess precipitation) to be used in the D-infinity weighted contributing area evaluation of Overland Flow Specific Discharge.

Outlets shapefile [vector: point] Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

Concentration Threshold [number] The concentration or solubility threshold. Over the substance supply area, concentration is at this threshold.

Default: *1.0*

Check for edge contamination [boolean] This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining contributing area.

Default: *True*

Outputs

Concentration Grid [raster] A grid giving the resulting concentration of the compound of interest in the flow.

Console usage

```
processing.runalg('taudem:dinfiniteconcentrationlimitedaccumulation', -ang, -dg, -dm, -q, -o, -cs
```

See also

D-Infinity Decaying Accumulation

Description

The D-Infinity Decaying Accumulation tool creates a grid of the accumulated quantity at each location in the domain where the quantity accumulates with the D-infinity flow field, but is subject to first order decay in moving from cell to cell. By default, the quantity contribution of each grid cell is the cell length to give a per unit width accumulation, but can optionally be expressed with a weight grid. The decay multiplier grid gives the fractional (first order) reduction in quantity in accumulating from grid cell x to the next downslope cell.

A decayed accumulation operator $DA[.]$ takes as input a mass loading field $m(x)$ expressed at each grid location as $m(i, j)$ that is assumed to move with the flow field but is subject to first order decay in moving from cell to cell. The output is the accumulated mass at each location $DA(x)$. The accumulation of m at each grid cell can be numerically evaluated.

Here $d(x) = d(i, j)$ is a decay multiplier giving the fractional (first order) reduction in mass in moving from grid cell x to the next downslope cell. If travel (or residence) times $t(x)$ associated with flow between cells are available $d(x)$ may be evaluated as $\exp(-k t(x))$ where k is a first order decay parameter. The weight grid is used to represent the mass loading $m(x)$. If not specified this is taken as 1. If the outlets shapefile is used the function is only evaluated on that part of the domain that contributes flow to the locations given by the shapefile.

Useful for a tracking contaminant or compound subject to decay or attenuation.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This grid can be created by the function “**D-Infinity Flow Directions**”.

Decay Multiplier Grid [raster] A grid giving the factor by which flow leaving each grid cell is multiplied before accumulation on downslope grid cells. This may be used to simulate the movement of an attenuating substance.

Weight Grid [raster] Optional.

A grid giving weights (loadings) to be used in the accumulation. If this optional grid is not specified, weights are taken as the linear grid cell size to give a per unit width accumulation.

Outlets Shapefile [vector: point] Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

Check for edge contamination [boolean] This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining contributing area.

Default: *True*

Outputs

Decayed Specific Catchment Area Grid [raster] The D-Infinity Decaying Accumulation tool creates a grid of the accumulated mass at each location in the domain where mass moves with the D-infinity flow field, but is subject to first order decay in moving from cell to cell.

Console usage

```
processing.runalg('taudem:dinfinitecachingaccumulation', -ang, -dm, -wg, -o, -nc, -dsca)
```

See also

D-Infinity Distance Down

Description

Calculates the distance downslope to a stream using the D-infinity flow model. The D-infinity flow model is a multiple flow direction model, because the outflow from each grid cell is proportioned between up to 2 downslope grid cells. As such, the distance from any grid cell to a stream is not uniquely defined. Flow that originates at a particular grid cell may enter the stream at a number of different cells. The statistical method may be selected as the longest, shortest or weighted average of the flow path distance to the stream. Also one of several ways of measuring distance may be selected: the total straight line path (Pythagoras), the horizontal component of the straight line path, the vertical component of the straight line path, or the total surface flow path.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Pit Filled Elevation Grid [raster] This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

Stream Raster Grid [raster] A grid indicating streams, by using a grid cell value of 1 on streams and 0 off streams. This is usually the output of one of the tools in the “**Stream Network Analysis**” toolset.

Weight Path Grid [raster] Optional.

A grid giving weights (loadings) to be used in the distance calculation. This might be used for example where only flow distance through a buffer is to be calculated. The weight is then 1 in the buffer and 0 outside it. Alternatively the weight may reflect some sort of cost function for travel over the surface, perhaps representing travel time or attenuation of a process. If this input file is not used, the loadings will assumed to be one for each grid cell.

Statistical Method [selection] Statistical method used to calculate the distance down to the stream. In the D-Infinity flow model, the outflow from each grid cell is proportioned between two downslope grid cells. Therefore, the distance from any grid cell to a stream is not uniquely defined. Flow that originates at a particular grid cell may enter the stream at a number of cells. The distance to the stream may be defined as the longest (maximum), shortest (minimum) or weighted average of the distance down to the stream.

Options:

- 0 — Minimum
- 1 — Maximum
- 2 — Average

Default: 2

Distance Method [selection] Distance method used to calculate the distance down to the stream. One of several ways of measuring distance may be selected: the total straight line path (Pythagoras), the horizontal component of the straight line path (horizontal), the vertical component of the straight line path (vertical), or the total surface flow path (surface).

Options:

- 0 — Pythagoras
- 1 — Horizontal
- 2 — Vertical
- 3 — Surface

Default: 1

Check for edge contamination [boolean] A flag that determines whether the tool should check for edge contamination. This is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being counted. In the context of Distance Down this occurs when part of a flow path traced downslope from a grid cell leaves the domain without reaching a stream grid cell. With edge contamination checking selected, the algorithm recognizes this and reports no data for the result. This is the desired effect and indicates that values for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge contamination checking may be overridden in cases where you know this is not an issue or want to evaluate the distance using only the fraction of flow paths that terminate at a stream.

Default: *True*

Outputs

D-Infinity Drop to Stream Grid [raster] Grid containing the distance to stream calculated using the D-infinity flow model and the statistical and path methods chosen.

Console usage

```
processing.runalg('taudem:dinfinitydistancedown', dinf_flow_dir_grid, pit_filled_grid, stream_grid)
```

See also

D-Infinity Distance Up

Description

This tool calculates the distance from each grid cell up to the ridge cells along the reverse D-infinity flow directions. Ridge cells are defined to be grid cells that have no contribution from grid cells further upslope. Given the convergence of multiple flow paths at any grid cell, any given grid cell can have multiple upslope ridge cells. There are three statistical methods that this tool can use: maximum distance, minimum distance and waited flow average over these flow paths. A variant on the above is to consider only grid cells that contribute flow with a proportion greater than a user specified threshold (t) to be considered as upslope of any given grid cell. Setting t=0.5 would result in only one flow path from any grid cell and would give the result equivalent to a D8 flow model, rather than D-infinity flow model, where flow is proportioned between two downslope grid cells. Finally there are several different optional paths that can be measured: the total straight line path (Pythagoras), the horizontal component of the straight line path, the vertical component of the straight line path, or the total surface flow path.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Pit Filled Elevation Grid [raster] This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

Slope Grid [raster] This input is a grid of slope values. This is measured as drop/distance and it is most often obtained as the output of the “**D-Infinity Flow Directions**” tool.

Statistical Method [selection] Statistical method used to calculate the distance down to the stream. In the D-Infinity flow model, the outflow from each grid cell is proportioned between two downslope grid cells. Therefore, the distance from any grid cell to a stream is not uniquely defined. Flow that originates at a particular grid cell may enter the stream at a number of cells. The distance to the stream may be defined as the longest (maximum), shortest (minimum) or weighted average of the distance down to the stream.

Options:

- 0 — Minimum
- 1 — Maximum
- 2 — Average

Default: 2

Distance Method [selection] Distance method used to calculate the distance down to the stream. One of several ways of measuring distance may be selected: the total straight line path (Pythagoras), the horizontal component of the straight line path (horizontal), the vertical component of the straight line path (vertical), or the total surface flow path (surface).

Options:

- 0 — Pythagoras

- 1 — Horizontal
- 2 — Vertical
- 3 — Surface

Default: *1*

Proportion Threshold [number] The proportion threshold parameter where only grid cells that contribute flow with a proportion greater than this user specified threshold (τ) is considered to be upslope of any given grid cell. Setting $\tau=0.5$ would result in only one flow path from any grid cell and would give the result equivalent to a D8 flow model, rather than D-Infinity flow model, where flow is proportioned between two downslope grid cells.

Default: *0.5*

Check for edge contamination [boolean] A flag that determines whether the tool should check for edge contamination. This is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being counted.

Default: *True*

Outputs

D-Infinity Distance Up [raster] Grid containing the distances up to the ridge calculated using the D-Infinity flow model and the statistical and path methods chosen.

Console usage

```
processing.runalg('taudem:dinfinitydistanceup', dinf_flow_dir_grid, pit_filled_grid, slope_grid, ...)
```

See also

D-Infinity Reverse Accumulation

Description

This works in a similar way to evaluation of weighted Contributing area, except that the accumulation is by propagating the weight loadings upslope along the reverse of the flow directions to accumulate the quantity of weight loading downslope from each grid cell. The function also reports the maximum value of the weight loading downslope from each grid cell in the Maximum Downslope grid.

This function is designed to evaluate and map the hazard due to activities that may have an effect downslope. The example is land management activities that increase runoff. Runoff is sometimes a trigger for landslides or debris flows, so the weight grid here could be taken as a terrain stability map. Then the reverse accumulation provides a measure of the amount of unstable terrain downslope from each grid cell, as an indicator of the danger of activities that may increase runoff, even though there may be no potential for any local impact.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Weight Grid [raster] A grid giving weights (loadings) to be used in the accumulation.

Outputs

Reverse Accumulation Grid [raster] The grid giving the result of the “Reverse Accumulation” function. This works in a similar way to evaluation of weighted Contributing area, except that the accumulation is by propagating the weight loadings upslope along the reverse of the flow directions to accumulate the quantity of loading downslope from each grid cell.

Maximum Downslope Grid [raster] The grid giving the maximum of the weight loading grid downslope from each grid cell.

Console usage

```
processing.runalg('taudem:dinfinityreverseaccumulation', -ang, -wg, -racc, -dmax)
```

See also

D-Infinity Transport Limited Accumulation - 2

Description

This function is designed to calculate the transport and deposition of a substance (e.g. sediment) that may be limited by both supply and the capacity of the flow field to transport it. This function accumulates substance flux (e.g. sediment transport) subject to the rule that transport out of any grid cell is the minimum between supply and transport capacity, T_{cap} . The total supply at a grid cell is calculated as the sum of the transport in from upslope grid cells, T_{in} , plus the local supply contribution, E (e.g. erosion). This function also outputs deposition, D , calculated as total supply minus actual transport.

Here E is the supply. T_{out} at each grid cell becomes T_{in} for downslope grid cells and is reported as Transport limited accumulation (t_{la}). D is deposition (t_{dep}). The function provides the option to evaluate concentration of a compound (contaminant) adhered to the transported substance. This is evaluated as follows:

Where L_{in} is the total incoming compound loading and C_{in} and T_{in} refer to the Concentration and Transport entering from each upslope grid cell.

If

else

where C_s is the concentration supplied locally and the difference in the second term on the right represents the additional supply from the local grid cell. Then,

C_{out} at each grid cell comprises is the concentration grid output from this function.

If the outlets shapefile is used the tool only evaluates that part of the domain that contributes flow to the locations given by the shapefile.

Transport limited accumulation is useful for modeling erosion and sediment delivery, including the spatial dependence of sediment delivery ratio and contaminant that adheres to sediment.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Supply Grid [raster] A grid giving the supply (loading) of material to a transport limited accumulation function. In the application to erosion, this grid would give the erosion detachment, or sediment supplied at each grid cell.

Transport Capacity Grid [raster] A grid giving the transport capacity at each grid cell for the transport limited accumulation function. In the application to erosion this grid would give the transport capacity of the carrying flow.

Input Concentration Grid [raster] A grid giving the concentration of a compound of interest in the supply to the transport limited accumulation function. In the application to erosion, this grid would give the concentration of say phosphorous adhered to the eroded sediment.

Outlets Shapefile [vector: point] Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

Check for edge contamination [boolean] This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining the result.

Default: *True*

Outputs

Transport Limited Accumulation Grid [raster] This grid is the weighted accumulation of supply accumulated respecting the limitations in transport capacity and reports the transport rate calculated by accumulating the substance flux subject to the rule that the transport out of any grid cell is the minimum of the total supply (local supply plus transport in) to that grid cell and the transport capacity.

Deposition Grid [raster] A grid giving the deposition resulting from the transport limited accumulation. This is the residual from the transport in to each grid cell minus the transport capacity out of the grid cell. The deposition grid is calculated as the transport in + the local supply - the transport out.

Output Concentration Grid [raster] If an input concentration in supply grid is given, then this grid is also output and gives the concentration of a compound (contaminant) adhered or bound to the transported substance (e.g. sediment) is calculated.

Console usage

```
processing.runalg('taudem:dinfiniteytransportlimitedaccumulation2', dinf_flow_dir_grid, supply_grid)
```

See also

D-Infinity Transport Limited Accumulation

Description

This function is designed to calculate the transport and deposition of a substance (e.g. sediment) that may be limited by both supply and the capacity of the flow field to transport it. This function accumulates substance flux (e.g. sediment transport) subject to the rule that transport out of any grid cell is the minimum between supply and transport capacity, T_{cap} . The total supply at a grid cell is calculated as the sum of the transport in from upslope grid cells, T_{in} , plus the local supply contribution, E (e.g. erosion). This function also outputs deposition, D , calculated as total supply minus actual transport.

Here E is the supply. T_{out} at each grid cell becomes T_{in} for downslope grid cells and is reported as Transport limited accumulation (t_{la}). D is deposition (t_{dep}). The function provides the option to evaluate concentration of a compound (contaminant) adhered to the transported substance. This is evaluated as follows:

Where L_{in} is the total incoming compound loading and C_{in} and T_{in} refer to the Concentration and Transport entering from each upslope grid cell.

If

else

where C_s is the concentration supplied locally and the difference in the second term on the right represents the additional supply from the local grid cell. Then,

C_{out} at each grid cell comprises is the concentration grid output from this function.

If the outlets shapefile is used the tool only evaluates that part of the domain that contributes flow to the locations given by the shapefile.

Transport limited accumulation is useful for modeling erosion and sediment delivery, including the spatial dependence of sediment delivery ratio and contaminant that adheres to sediment.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Supply Grid [raster] A grid giving the supply (loading) of material to a transport limited accumulation function. In the application to erosion, this grid would give the erosion detachment, or sediment supplied at each grid cell.

Transport Capacity Grid [raster] A grid giving the transport capacity at each grid cell for the transport limited accumulation function. In the application to erosion this grid would give the transport capacity of the carrying flow.

Outlets Shapefile [vector: point] Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

Check for edge contamination [boolean] This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining the result.

Default: *True*

Outputs

Transport Limited Accumulation Grid [raster] This grid is the weighted accumulation of supply accumulated respecting the limitations in transport capacity and reports the transport rate calculated by accumulating the substance flux subject to the rule that the transport out of any grid cell is the minimum of the total supply (local supply plus transport in) to that grid cell and the transport capacity.

Deposition Grid [raster] A grid giving the deposition resulting from the transport limited accumulation. This is the residual from the transport in to each grid cell minus the transport capacity out of the grid cell. The deposition grid is calculated as the transport in + the local supply - the transport out.

Console usage

```
processing.runalg('taudem:dinfiniteynitytransportlimitedaccumulation', dinf_flow_dir_grid, supply_grid,
```

See also

D-Infinity Upslope Dependence

Description

The D-Infinity Upslope Dependence tool quantifies the amount each grid cell in the domain contributes to a destination set of grid cells. D-Infinity flow directions proportion flow from each grid cell between multiple downslope grid cells. Following this flow field downslope the amount of flow originating at each grid cell that reaches the destination zone is defined. Upslope influence is evaluated using a downslope recursion, examining grid cells downslope from each grid cell, so that the map produced identifies the area upslope where flow through the destination zone originates, or the area it depends on, for its flow.

The figures below illustrate the amount each source point in the domain x (blue) contributes to the destination point or zone y (red). If the indicator weighted contributing area function is denoted $I(y; x)$ giving the weighted contribution using a unit value (1) from specific grid cells y to grid cells x , then the upslope dependence is: $D(x; y) = I(y; x)$.

This is useful for example to track where flow or a flow related substance or contaminant that enters a destination area may come from.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-Infinity method where the flow direction angle is determined as the direction of the steepest downward slope on the eight triangular facets formed in a 3x3 grid cell window centered on the grid cell of interest. This grid can be produced using the “**D-Infinity Flow Direction**” tool.

Destination Grid [raster] A grid that encodes the destination zone that may receive flow from upslope. This grid must be 1 inside the zone y and 0 over the rest of the domain.

Outputs

Output Upslope Dependence Grid [raster] A grid quantifying the amount each source point in the domain contributes to the zone defined by the destination grid.

Console usage

```
processing.runalg('taudem:dinfinityupslopedependence', -ang, -dg, -dep)
```

See also

Slope Average Down

Description

This tool computes slope in a D8 downslope direction averaged over a user selected distance. Distance should be specified in horizontal map units.

Parameters

D8 Flow Direction Grid [raster] This input is a grid of flow directions that are encoded using the D8 method where all flow from a cells goes to a single neighboring cell in the direction of steepest descent. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Pit Filled Elevation Grid [raster] This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

Downslope Distance [number] Input parameter of downslope distance over which to calculate the slope (in horizontal map units).

Default: 50

Outputs

Slope Average Down Grid [raster] This output is a grid of slopes calculated in the D8 downslope direction, averaged over the selected distance.

Console usage

```
processing.runalg('taudem:slopeaveragedown', -p, -fel, -dn, -slpd)
```

See also

Slope Over Area Ratio

Description

Calculates the ratio of the slope to the specific catchment area (contributing area). This is algebraically related to the more common $\ln(a/\tan \beta)$ wetness index, but contributing area is in the denominator to avoid divide by 0 errors when slope is 0.

Parameters

Slope Grid [raster] A grid of slope. This grid can be generated using either the “**D8 Flow Directions**” tool or the “**D-Infinity Flow Directions**” tool.

Specific Catchment Area Grid [raster] A grid giving the contributing area value for each cell taken as its own contribution plus the contribution from upslope neighbors that drain in to it. Contributing area is counted in terms of the number of grid cells (or summation of weights). This grid can be generated using either the “**D8 Contributing Area**” tool or the “**D-Infinity Contributing Area**” tool.

Outputs

Slope Divided By Area Ratio Grid [raster] A grid of the ratio of slope to specific catchment area (contributing area). This is algebraically related to the more common $\ln(a/\tan \beta)$ wetness index, but contributing area is in the denominator to avoid divide by 0 errors when slope is 0.

Console usage

```
processing.runalg('taudem:slopeoverarearatio', -slp, -sca, -sar)
```

See also

18.8.3 Stream Network Analysis

D8 Extreme Upslope Value

Description

Evaluates the extreme (either maximum or minimum) upslope value from an input grid based on the D8 flow model. This is intended initially for use in stream raster generation to identify a threshold of the slope times area product that results in an optimum (according to drop analysis) stream network.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D8 flow model) of them are in the domain to be evaluated.

By default, the tool checks for edge contamination. This is defined as the possibility that a result may be underestimated due to grid cells outside of the domain not being counted. This occurs when drainage is inwards from the boundaries or areas with “no data” values for elevation. The algorithm recognizes this and reports “no data” for the result for these grid cells. It is common to see streaks of “no data” values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that the result for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge

contamination checking may be turned off in cases where you know this is not an issue or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Parameters

D8 Flow Directions Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Upslope Values Grid [raster] This is the grid of values of which the maximum or minimum upslope value is selected. The values most commonly used are the slope times area product needed when generating stream rasters according to drop analysis.

Outlets Shapefile [vector: point] Optional.

A point shape file defining outlets of interest. If this input file is used, only the area upslope of these outlets will be evaluated by the tool.

Check for edge contamination [boolean] A flag that indicates whether the tool should check for edge contamination.

Default: *True*

Use max upslope value [boolean] A flag to indicate whether the maximum or minimum upslope value is to be calculated.

Default: *True*

Outputs

Extreme Upslope Values Grid [raster] A grid of the maximum/minimum upslope values.

Console usage

```
processing.runalg('taudem:d8extremeupslopevalue', -p, -sa, -o, -nc, -min, -ssa)
```

See also

Length Area Stream Source

Description

Creates an indicator grid (1, 0) that evaluates $A \geq (M) (L^y)$ based on upslope path length, D8 contributing area grid inputs, and parameters M and y. This grid indicates likely stream source grid cells. This is an experimental method with theoretical basis in Hack’s law which states that for streams $L \sim A^{0.6}$. However for hillslopes with parallel flow $L \sim A$. So a transition from hillslopes to streams may be represented by $L \sim A^{0.8}$ suggesting identifying grid cells as stream cells if $A > M (L^{(1/0.8)})$.

Parameters

Length Grid [raster] A grid of the maximum upslope length for each cell. This is calculated as the length of the flow path from the furthest cell that drains to each cell. Length is measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal. It is this length (L) that is used in the formula, $A > (M) (L^y)$, to determine which cells are considered stream cells. This grid can be obtained as an output from the “**Grid Network**” tool.

Contributing Area Grid [raster] A grid of contributing area values for each cell that were calculated using the D8 algorithm. The contributing area for a cell is the sum of its own contribution plus the contribution from all upslope neighbors that drain to it, measured as a number of cells. This grid is typically obtained as the output of the “**D8 Contributing Area**” tool. In this tool, it is the contributing area (A) that is compared in the formula $A > (M) (L^y)$ to determine the transition to a stream.

Threshold [number] The multiplier threshold (M) parameter which is used in the formula: $A > (M) (L^y)$, to identify the beginning of streams.

Default: *0.03*

Exponent [number] The exponent (y) parameter which is used in the formula: $A > (M) (L^y)$, to identify the beginning of streams. In branching systems, Hack’s law suggests that $L = 1/M A^{(1/y)}$ with $1/y = 0.6$ (or 0.56) (y about 1.7). In parallel flow systems L is proportional to A (y about 1). This method tries to identify the transition between these two paradigms by using an exponent y somewhere in between (y about 1.3).

Default: *1.3*

Outputs

Stream Source Grid [raster] An indicator grid (1,0) that evaluates $A \geq (M)(L^y)$, based on the maximum upslope path length, the D8 contributing area grid inputs, and parameters M and y . This grid indicates likely stream source grid cells.

Console usage

```
processing.runalg('taudem:lengthareastreamsource', length_grid, contrib_area_grid, threshold, exp
```

See also

Move Outlets To Streams

Description

Moves outlet points that are not aligned with a stream cell from a stream raster grid, downslope along the D8 flow direction until a stream raster cell is encountered, the “max_dist” number of grid cells are examined, or the flow path exits the domain (i.e. a “no data” value is encountered for the D8 flow direction). The output file is a new outlets shapefile where each point has been moved to coincide with the stream raster grid, if possible. A field “dist_moved” is added to the new outlets shapefile to indicate the changes made to each point. Points that are already on a stream cell are not moved and their “dist_moved” field is assigned a value 0. Points that are initially not on a stream cell are moved by sliding them downslope along the D8 flow direction until one of the following occurs: a) A stream raster grid cell is encountered before traversing the “max_dist” number of grid cells. In which case, the point is moved and the “dist_moved” field is assigned a value indicating how many grid cells the point was moved. b) More than the “max_number” of grid cells are traversed, or c) the traversal ends up going out of the domain (i.e., a “no data” D8 flow direction value is encountered). In which case, the point is not moved and the “dist_moved” field is assigned a value of -1.

Parameters

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Stream Raster Grid [raster] This output is an indicator grid (1, 0) that indicates the location of streams, with a value of 1 for each of the stream cells and 0 for the remainder of the cells. This file is produced by several different tools in the “**Stream Network Analysis**” toolset.

Outlets Shapefile [vector: point] A point shape file defining points of interest or outlets that should ideally be located on a stream, but may not be exactly on the stream due to the fact that the shapefile point locations may not have been accurately registered with respect to the stream raster grid.

Maximum Number of Grid Cells to traverse [number] This input parameter is the maximum number of grid cells that the points in the input outlet shapefile will be moved before they are saved to the output outlet shapefile.

Default: 50

Outputs

Output Outlet Shapefile [vector] A point shape file defining points of interest or outlets. This file has one point in it for each point in the input outlet shapefile. If the original point was located on a stream, then the point was not moved. If the original point was not on a stream, the point was moved downslope according to the D8 flow direction until it reached a stream or the maximum distance had been reached. This file has an additional field “dist_moved” added to it which is the number of cells that the point was moved. This field is 0 if the cell was originally on a stream, -1 if it was not moved because there was not a stream within the maximum distance, or some positive value if it was moved.

Console usage

```
processing.runalg('taudem:moveoutletstostreams', -p, -src, -o, -md, -om)
```

See also

Peuker Douglas

Description

Creates an indicator grid (1, 0) of upward curved grid cells according to the Peuker and Douglas algorithm.

With this tool, the DEM is first smoothed by a kernel with weights at the center, sides, and diagonals. The Peuker and Douglas (1975) method (also explained in Band, 1986), is then used to identify upwardly curving grid cells. This technique flags the entire grid, then examines in a single pass each quadrant of 4 grid cells, and unflags the highest. The remaining flagged cells are deemed “upwardly curved”, and when viewed, resemble a channel network. This proto-channel network generally lacks connectivity and requires thinning, issues that were discussed in detail by Band (1986).

Parameters

Elevation Grid [raster] A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed.

Center Smoothing Weight [number] The center weight parameter used by a kernel to smooth the DEM before the tool identifies upwardly curved grid cells.

Default: 0.4

Side Smoothing Weight [number] The side weight parameter used by a kernel to smooth the DEM before the tool identifies upwardly curved grid cells.

Default: 0.1

Diagonal Smoothing Weight [number] The diagonal weight parameter used by a kernel to smooth the DEM before the tool identifies upwardly curved grid cells.

Default: *0.05*

Outputs

Stream Source Grid [raster] An indicator grid (1, 0) of upward curved grid cells according to the Peuker and Douglas algorithm, and if viewed, resembles a channel network. This proto-channel network generally lacks connectivity and requires thinning, issues that were discussed in detail by Band (1986).

Console usage

```
processing.runalg('taudem:peukerdouglas', elevation_grid, center_weight, side_weight, diagonal_we
```

See also

- Band, L. E., (1986), “Topographic partition of watersheds with digital elevation models”, *Water Resources Research*, 22(1): 15-24.
- Peuker, T. K. and D. H. Douglas, (1975), “Detection of surface-specific points by local parallel processing of discrete terrain elevation data”, *Comput. Graphics Image Process.*, 4: 375-387.

Slope Area Combination

Description

Creates a grid of slope-area values = $(S_m)^m (A_n)^n$ based on slope and specific catchment area grid inputs, and parameters m and n . This tool is intended for use as part of the slope-area stream raster delineation method.

Parameters

Slope Grid [raster] This input is a grid of slope values. This grid can be obtained from the “**D-Infinity Flow Directions**” tool.

Contributing Area Grid [raster] A grid giving the specific catchment area for each cell taken as its own contribution (grid cell length or summation of weights) plus the proportional contribution from upslope neighbors that drain in to it. This grid is typically obtained from the “**D-Infinity Contributing Area**” tool.

Slope Exponent [number] The slope exponent (m) parameter which will be used in the formula: $(S_m)^m (A_n)^n$, that is used to create the slope-area grid.

Default: *2*

Area Exponent [number] The area exponent (n) parameter which will be used in the formula: $(S_m)^m (A_n)^n$, that is used to create the slope-area grid.

Default: *1*

Outputs

Slope Area Grid [raster] A grid of slope-area values = $(S_m)^m (A_n)^n$ calculated from the slope grid, specific catchment area grid, m slope exponent parameter, and n area exponent parameter.

Console usage

```
processing.runalg('taudem:slopeareacombination', slope_grid, area_grid, slope_exponent, area_exponent)
```

See also

Stream Definition By Threshold

Description

Operates on any grid and outputs an indicator (1, 0) grid identifying cells with input values \geq the threshold value. The standard use is to use an accumulated source area grid to as the input grid to generate a stream raster grid as the output. If you use the optional input mask grid, it limits the domain being evaluated to cells with mask values \geq 0. When you use a D-infinity contributing area grid (**sca*) as the mask grid, it functions as an edge contamination mask. The threshold logic is:

```
src = ((ssa >= thresh) & (mask >= s0)) ? 1:0
```

Parameters

Accumulated Stream Source Grid [raster] This grid nominally accumulates some characteristic or combination of characteristics of the watershed. The exact characteristic(s) varies depending on the stream network raster algorithm being used. This grid needs to have the property that grid cell values are monotonically increasing downslope along D8 flow directions, so that the resulting stream network is continuous. While this grid is often from an accumulation, other sources such as a maximum upslope function will also produce a suitable grid.

Threshold [number] This parameter is compared to the value in the Accumulated Stream Source grid (**ssa*) to determine if the cell should be considered a stream cell. Streams are identified as grid cells for which ssa value is \geq this threshold.

Default: *100*

Mask Grid [raster] Optional.

This optional input is a grid that is used to mask the domain of interest and output is only provided where this grid is \geq 0. A common use of this input is to use a D-Infinity contributing area grid as the mask so that the delineated stream network is constrained to areas where D-infinity contributing area is available, replicating the functionality of an edge contamination mask.

Outputs

Stream Raster Grid [raster] This is an indicator grid (1, 0) that indicates the location of streams, with a value of 1 for each of the stream cells and 0 for the remainder of the cells.

Console usage

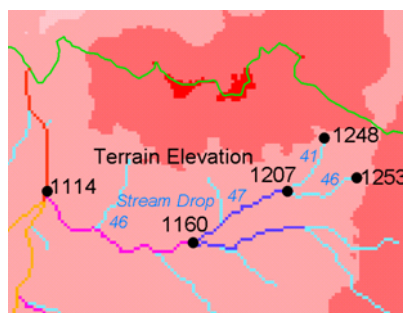
```
processing.runalg('taudem:streamdefinitionbythreshold', -ssa, -thresh, -mask, -src)
```

See also

Stream Drop Analysis

Description

Applies a series of thresholds (determined from the input parameters) to the input accumulated stream source grid (*ssa) grid and outputs the results in the *drp.txt file the stream drop statistics table. This function is designed to aid in the determination of a geomorphologically objective threshold to be used to delineate streams. Drop Analysis attempts to select the right threshold automatically by evaluating a stream network for a range of thresholds and examining the constant drop property of the resulting Strahler streams. Basically it asks the question: Is the mean stream drop for first order streams statistically different from the mean stream drop for higher order streams, using a T-test. Stream drop is the difference in elevation from the beginning to the end of a stream defined as the sequence of links of the same stream order. If the T-test shows a significant difference then the stream network does not obey this “law” so a larger threshold needs to be chosen. The smallest threshold for which the T-test does not show a significant difference gives the highest resolution stream network that obeys the constant stream drop “law” from geomorphology, and is the threshold chosen for the “objective” or automatic mapping of streams from the DEM. This function can be used in the development of stream network rasters, where the exact watershed characteristic(s) that were accumulated in the accumulated stream source grid vary based on the method being used to determine the stream network raster.



The constant stream drop “law” was identified by Broscoc (1959). For the science behind using this to determine a stream delineation threshold, see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Parameters

D8 Contributing Area Grid [raster] A grid of contributing area values for each cell that were calculated using the D8 algorithm. The contributing area for a cell is the sum of its own contribution plus the contribution from all upslope neighbors that drain to it, measured as a number of cells or the sum of weight loadings. This grid can be obtained as the output of the “**D8 Contributing Area**” tool. This grid is used in the evaluation of drainage density reported in the stream drop table.

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Pit Filled Elevation Grid [raster] A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed.

Accumulated Stream Source Grid [raster] This grid must be monotonically increasing along the downslope D8 flow directions. It is compared to a series of thresholds to determine the beginning of the streams. It is often generated by accumulating some characteristic or combination of characteristics of the watershed with the “**D8 Contributing Area**” tool, or using the maximum option of the “**D8 Flow Path Extreme**” tool. The exact method varies depending on the algorithm being used.

Outlets Shapefile [vector: point] A point shapefile defining the outlets upstream of which drop analysis is performed.

Minimum Threshold [number] This parameter is the lowest end of the range searched for possible threshold values using drop analysis. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Default: 5

Maximum Threshold [number] This parameter is the highest end of the range searched for possible threshold values using drop analysis. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Default: 500

Number of Threshold Values [number] The parameter is the number of steps to divide the search range into when looking for possible threshold values using drop analysis. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Default: 10

Spacing for Threshold Values [selection] This parameter indicates whether logarithmic or linear spacing should be used when looking for possible threshold values using drop analysis.

Options:

- 0 — Logarithmic
- 1 — Linear

Default: 0

Outputs

D-Infinity Drop to Stream Grid [file] This is a comma delimited text file with the following header line:

```
:: Threshold,DrainDen,NoFirstOrd,NoHighOrd,MeanDFirstOrd,MeanDHighOrd,StdDevFirstOrd,StdDevHighOrd,T
```

The file then contains one line of data for each threshold value examined, and then a summary line that indicates the optimum threshold value. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis, see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Console usage

```
processing.runalg('taudem:streamdropanalysis', d8_contrib_area_grid, d8_flow_dir_grid, pit_filled,
```

See also

- Broscoc, A. J., (1959), "Quantitative analysis of longitudinal stream profiles of small watersheds", Office of Naval Research, Project NR 389-042, Technical Report No. 18, Department of Geology, Columbia University, New York.
- Tarboton, D. G., R. L. Bras and I. Rodriguez-Iturbe, (1991), "On the Extraction of Channel Networks from Digital Elevation Data", Hydrologic Processes, 5(1): 81-100.
- Tarboton, D. G., R. L. Bras and I. Rodriguez-Iturbe, (1992), "A Physical Basis for Drainage Density", Geomorphology, 5(1/2): 59-76.
- Tarboton, D. G. and D. P. Ames, (2001), "Advances in the mapping of flow networks from digital elevation data", World Water and Environmental Resources Congress, Orlando, Florida, May 20-24, ASCE, <http://www.engineering.usu.edu/dtarb/asce2001.pdf>.

Stream Reach and Watershed

Description

This tool produces a vector network and shapefile from the stream raster grid. The flow direction grid is used to connect flow paths along the stream raster. The Strahler order of each stream segment is computed. The subwatershed draining to each stream segment (reach) is also delineated and labeled with the value identifier that corresponds to the WSNO (watershed number) attribute in the Stream Reach Shapefile.

This tool orders the stream network according to the Strahler ordering system. Streams that don't have any other streams draining in to them are order 1. When two stream reaches of different order join the order of the downstream reach is the order of the highest incoming reach. When two reaches of equal order join the downstream reach order is increased by 1. When more than two reaches join the downstream reach order is calculated as the maximum of the highest incoming reach order or the second highest incoming reach order + 1. This generalizes the common definition to cases where more than two reaches join at a point. The network topological connectivity is stored in the Stream Network Tree file, and coordinates and attributes from each grid cell along the network are stored in the Network Coordinates file.

The stream raster grid is used as the source for the stream network, and the flow direction grid is used to trace connections within the stream network. Elevations and contributing area are used to determine the elevation and contributing area attributes in the network coordinate file. Points in the outlets shapefile are used to logically split stream reaches to facilitate representing watersheds upstream and downstream of monitoring points. The program uses the attribute field "id" in the outlets shapefile as identifiers in the Network Tree file. This tool then translates the text file vector network representation in the Network Tree and Coordinates files into a shapefile. Further attributes are also evaluated. The program has an option to delineate a single watershed by representing the entire area draining to the Stream Network as a single value in the output watershed grid.

Parameters

Pit Filled Elevation Grid [raster] A grid of elevation values. This is usually the output of the "**Pit Remove**" tool, in which case it is elevations with pits removed.

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the "**D8 Flow Directions**" tool.

D8 Drainage Area [raster] A grid giving the contributing area value in terms of the number of grid cells (or the summation of weights) for each cell taken as its own contribution plus the contribution from upslope neighbors that drain in to it using the D8 algorithm. This is usually the output of the "**D8 Contributing Area**" tool and is used to determine the contributing area attribute in the Network Coordinate file.

Stream Raster Grid [raster] An indicator grid indicating streams, by using a grid cell value of 1 on streams and 0 off streams. Several of the "**Stream Network Analysis**" tools produce this type of grid. The Stream Raster Grid is used as the source for the stream network.

Outlets Shapefile as Network Nodes [vector: point] Optional.

A point shape file defining points of interest. If this file is used, the tool will only delineate the stream network upstream of these outlets. Additionally, points in the Outlets Shapefile are used to logically split stream reaches to facilitate representing watersheds upstream and downstream of monitoring points. This tool **REQUIRES THAT THERE BE** an integer attribute field "id" in the Outlets Shapefile, because the "id" values are used as identifiers in the Network Tree file.

Delineate Single Watershed [boolean] This option causes the tool to delineate a single watershed by representing the entire area draining to the Stream Network as a single value in the output watershed grid. Otherwise a separate watershed is delineated for each stream reach. Default is *False* (separate watershed).

Default: *False*

Outputs

Stream Order Grid [raster] The Stream Order Grid has cells values of streams ordered according to the Strahler order system. The Strahler ordering system defines order 1 streams as stream reaches that don't have any other reaches draining in to them. When two stream reaches of different order join the order of the downstream reach is the order of the highest incoming reach. When two reaches of equal order join the downstream reach order is increased by 1. When more than two reaches join the downstream reach order is calculated as the maximum of the highest incoming reach order or the second highest incoming reach order + 1. This generalizes the common definition to cases where more than two flow paths reaches join at a point.

Watershed Grid [raster] This output grid identified each reach watershed with a unique ID number, or in the case where the delineate single watershed option was checked, the entire area draining to the stream network is identified with a single ID.

Stream Reach Shapefile [vector] This output is a polyline shapefile giving the links in a stream network. The columns in the attribute table are:

- LINKNO — Link Number. A unique number associated with each link (segment of channel between junctions). This is arbitrary and will vary depending on number of processes used
- DSLINKNO — Link Number of the downstream link. -1 indicates that this does not exist
- USLINKNO1 — Link Number of first upstream link. (-1 indicates no link upstream, i.e. for a source link)
- USLINKNO2 — Link Number of second upstream link. (-1 indicates no second link upstream, i.e. for a source link or an internal monitoring point where the reach is logically split but the network does not bifurcate)
- DSNODEID — Node identifier for node at downstream end of stream reach. This identifier corresponds to the "id" attribute from the Outlets shapefile used to designate nodes
- Order — Strahler Stream Order
- Length — Length of the link. The units are the horizontal map units of the underlying DEM grid
- Magnitude — Shreve Magnitude of the link. This is the total number of sources upstream
- DS_Cont_Ar — Drainage area at the downstream end of the link. Generally this is one grid cell upstream of the downstream end because the drainage area at the downstream end grid cell includes the area of the stream being joined
- Drop — Drop in elevation from the start to the end of the link
- Slope — Average slope of the link (computed as drop/length)
- Straight_L — Straight line distance from the start to the end of the link
- US_Cont_Ar — Drainage area at the upstream end of the link
- WSNO — Watershed number. Cross reference to the `*w.shp` and `*w` grid files giving the identification number of the watershed draining directly to the link
- DOUT_END — Distance to the eventual outlet (i.e. the most downstream point in the stream network) from the downstream end of the link
- DOUT_START — Distance to the eventual outlet from the upstream end of the link
- DOUT_MID — Distance to the eventual outlet from the midpoint of the link

Network Connectivity Tree [file] This output is a text file that details the network topological connectivity is stored in the Stream Network Tree file. Columns are as follows:

- Link Number (Arbitrary — will vary depending on number of processes used)
- Start Point Number in Network coordinates (`*coord.dat`) file (Indexed from 0)
- End Point Number in Network coordinates (`*coord.dat`) file (Indexed from 0)

- Next (Downstream) Link Number. Points to Link Number. -1 indicates no links downstream, i.e. a terminal link
- First Previous (Upstream) Link Number. Points to Link Number. -1 indicates no upstream links
- Second Previous (Upstream) Link Numbers. Points to Link Number. -1 indicates no upstream links. Where only one previous link is -1, it indicates an internal monitoring point where the reach is logically split, but the network does not bifurcate
- Strahler Order of Link
- Monitoring point identifier at downstream end of link. -1 indicates downstream end is not a monitoring point
- Network magnitude of the link, calculated as the number of upstream sources (following Shreve)

Network Coordinates [file] This output is a text file that contains the coordinates and attributes of points along the stream network. Columns are as follows:

- X coordinate
- Y Coordinate
- Distance along channels to the downstream end of a terminal link
- Elevation
- Contributing area

Console usage

```
processing.runalg('taudem:streamreachandwatershed', -fel, -p, -ad8, -src, -o, -sw, -ord, -w, -net,
```

See also

.

Compositor de Impressão

With the Print Composer you can create nice maps and atlases that can be printed or saved as PDF-file, an image or an SVG-file. This is a powerfull way to share geographical information produced with QGIS that can be included in reports or published.

The Print Composer provides growing layout and printing capabilities. It allows you to add elements such as the QGIS map canvas, text labels, images, legends, scale bars, basic shapes, arrows, attribute tables and HTML frames. You can size, group, align and position each element and adjust the properties to create your layout. The layout can be printed or exported to image formats, PostScript, PDF or to SVG (export to SVG is not working properly with some recent Qt4 versions; you should try and check individually on your system). You can save the layout as a template and load it again in another session. Finally, generating several maps based on a template can be done through the atlas generator. See a list of tools in [table_composer_1](#):





















































Ícone	Finalidade	Ícone	Finalidade
	Guardar Projecto		Novo Compositor
	Duplicar Compositor		Gestor de Compositores
	Carregar a partir do modelo		Guardar como modelo
	Imprimir ou exportar como PostScript		Exportar como imagem
	Exportar como SVG		Exportar como PDF
	Reverter à última alteração		Restaurar a última alteração
	Zoom Total		Visualizar a 100%
	Aproximar		Afastar
	Actualizar vista		Zoom a uma área específica
	Movimentar		Mover conteúdo do item
	Seleccionar/ Mover item		Adicionar imagem
	Adicionar novo mapa do QGIS no enquadramento do mapa		Adicionar nova legenda
	Adicionar nova etiqueta		Adicionar forma básica
	Adicionar barra de escala ao compositor de impressão		Adicionar tabela de atributos
	Adicionar seta		Desagrupar itens
	Adicionar uma HTML Frame		Desbloquear todos os itens
	Agrupar itens		Abaixar itens seleccionados
	Bloquear itens seleccionados		Enviar para trás
	Elevar itens seleccionados		Alinhar à direita
	Trazer para a frente		Centraliza na vertical
	Alinhar à esquerda		Alinhar ao fundo
	Alinhar ao centro		Primeiro Elemento
	Alinhar ao topo		Próximo Elemento
	Pré-Visualizar Atlas		Imprimir Atlas
	Elemento Anterior		Atlas Configurações
	Último Elemento		
	Exportar Atlas como Imagem		


Tabela 1 do Compositor: Ferramentas do Compositor de Impressão

Todas as ferramentas do Compositor de Impressão estão disponíveis nos menus e como ícones na barra de ferramentas. A barra de ferramentas pode ser desligada ou ligada usando o botão direito do rato sobre a barra de ferramentas.

19.1 Primeiros passos

19.1.1 Abrir um novo Modelo de de Compositor de Impressão

Antes de começar a trabalhar com o compositor de impressão, necessita de carregar algumas camadas vectoriais e matriciais no enquadramento do mapa QGIS e adaptar as suas propriedades para se ajustar à sua conveniência.

Após tudo estar renderizado ou simbolizado ao seu gosto, clique no ícone  Novo Compositor de Impressão na barra de ferramentas ou escolha *Ficheiro* → *Novo Compositor de Impressão*. Será-lhe pedido para escolher um título para o novo compositor.

19.1.2 Overview of the Print Composer

Opening the Print Composer provides you with a blank canvas that represents the paper surface when using the print option. Initially you find buttons on the left beside the canvas to add map composer items; the current QGIS map canvas, text labels, images, legends, scale bars, basic shapes, arrows, attribute tables and HTML frames. In this toolbar you also find toolbar buttons to navigate, zoom in on an area and pan the view on the composer and toolbar buttons to select a map composer item and to move the contents of the map item.

Figure_composer_overview shows the initial view of the Print Composer before any elements are added.

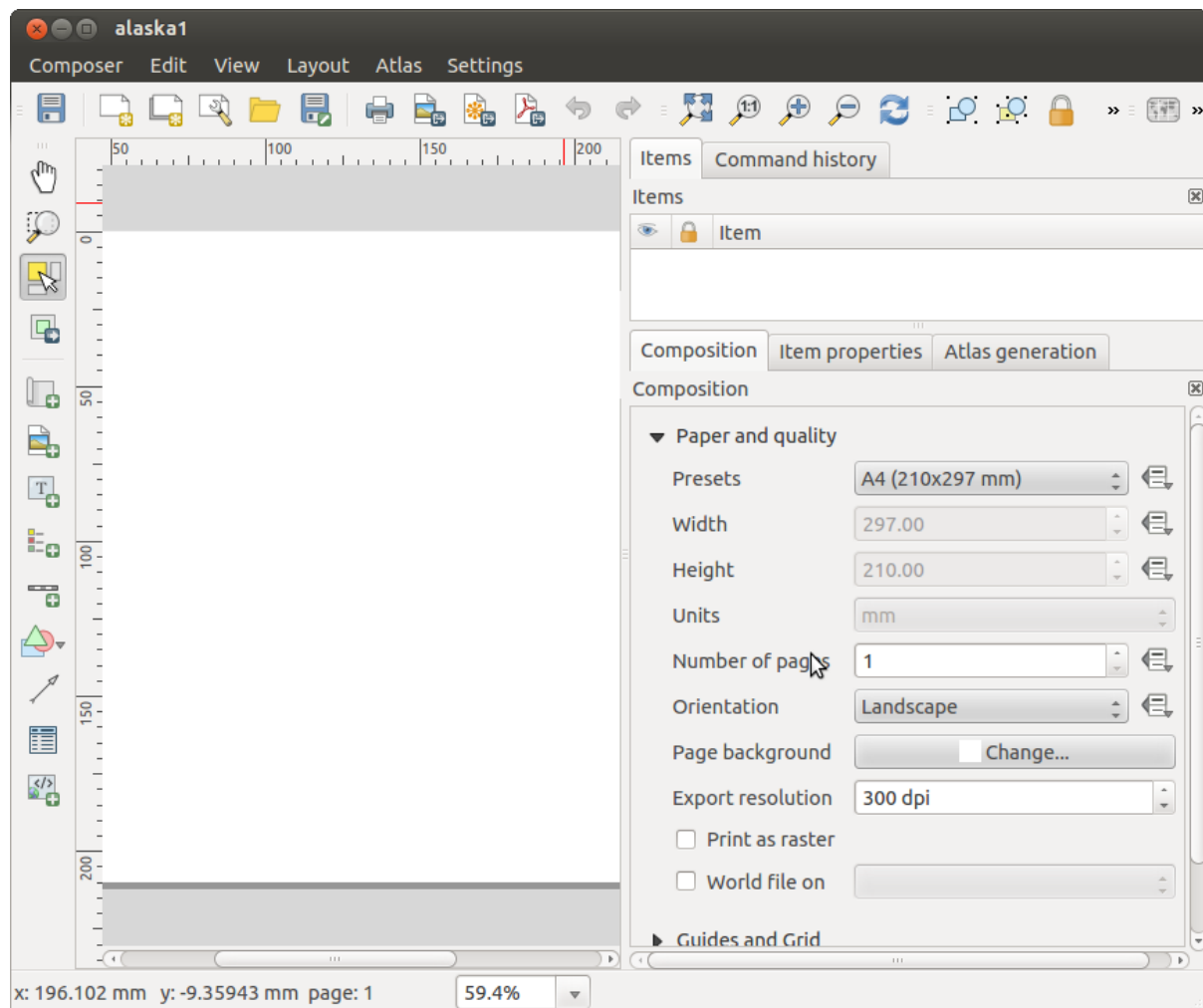




Figura 19.1: Compositor de Impressão 

On the right beside the canvas you find two panels. The upper panel holds the tabs *Items* and *Command History* and the lower panel holds the tabs *Composition*, *Item properties* and *Atlas generation*.

- The *Items* tab provides a list of all map composer items added to the canvas.
- The *Command history* tab displays a history of all changes applied to the Print Composer layout. With a mouse click, it is possible to undo and redo layout steps back and forth to a certain status.
- The *Composition* tab allows you to set paper size, orientation, the page background, number of pages and print quality for the output file in dpi. Furthermore, you can also activate the *Print as raster* checkbox.

This means all items will be converted to raster before printing or saving as PostScript or PDF. In this tab, you can also customize settings for grid and smart guides.








- The *Item Properties* tab displays the properties for the selected item. Click the  Select/Move item icon to select an item (e.g., legend, scale bar or label) on the canvas. Then click the *Item Properties* tab and customize the settings for the selected item.
- O separador *Gerador de Atlas* permite activar a criação de um atlas para o compositor actual e dá acesso aos seus parâmetros.
- Finally, you can save your print composition with the  Save Project button.

No fundo da janela do Compositor de Impressão, pode encontrar a barra de estado com a posição do rato, o número da página actual e uma combo box para definir o nível de zoom.

Pode adicionar múltiplos elementos no compositor. É também possível ter mais do que uma vista de mapa ou legenda ou escala gráfica no enquadramento do compositor de impressão, numa ou em várias páginas. Cada elemento tem as suas próprias propriedades e no caso do mapa, a sua própria extensão. Se quiser remover alguma elemento do enquadramento do compositor pode-o fazer usando a tecla *Delete* ou *Backspace*.

Ferramentas de Navegação





Para navegar no enquadramento do layout, o compositor de impressão fornece 4 ferramentas gerais:


-  Aproximar
-  Afastar
-  Zoom to full extent
-  Visualizar a 100%
-  Refresh the view (if you find the view in an inconsistent state)
-  *Mover Mapa*
-  Marquee zoom mode (zoom to a specific region of the Composer)

You can change the zoom level also using the mouse wheel or the combo box in the status bar. If you need to switch to pan mode while working in the Composer area, you can hold the *Spacebar* or the the mouse wheel. With *Ctrl+Spacebar*, you can temporarily switch to marquee zoom mode, and with *Ctrl+Shift+Spacebar*, to zoom out mode.

19.1.3 Sample Session

To demonstrate how to create a map please follow the next instructions.

1. On the left site, select the  Add new map toolbar button and draw a rectangle on the canvas holding down the left mouse button. Inside the drawn rectangle the QGIS map view to the canvas.
2. Select the  Add new scalebar toolbar button and place the map item with the left mouse button on the Print Composer canvas. A scalebar will be added to the canvas.
3. Select the  Add new legend toolbar button and draw a rectangle on the canvas holding down the left mouse button. Inside the drawn rectangle the legend will be drawn.
4. Select the  Select/Move item icon to select the map on the canvas and move it a bit.

5. While the map item is still selected you can also change the size of the map item. Click while holding down the left mouse button, in a white little rectangle in one of the corners of the map item and draw it to a new location to change it's size.
6. Click the *Item Properties* tab on the left lower panel and find the setting for the orientation. Change it the value of the setting *Map orientation* to '15.00° '. You should see the orientation of the map item change.
7. Finally, you can save your print composition with the  Save Project button.

19.1.4 Opções do Compositor de Impressão

A partir das *Configurações* → *Opções do Compositor* pode definir algumas opções que serão usadas como padrão durante o seu trabalho.

- As *Configurações padrão das Composições* deixam-mo especificar a fonte padrão a usar.
- With *Grid appearance*, you can set the grid style and its color.
- *Grid defaults* defines spacing, offset and tolerance of the grid. There are three types of grid: **Dots**, **Solid lines** and **Crosses**.
- *Guide defaults* defines the tolerance for the guides.

19.1.5 Separador de Composição — Configuração geral da composição

No separador *Composição*, pode definir as configurações globais à sua composição.

- Pode escolher uma das *Predefinições* para o seu formato de papel, ou introduza a sua *largura* e *altura* personalizada.
- A composição pode ser agora separada em várias páginas. Por exemplo, a primeira página pode mostrar o enquadramento do mapa e a segunda página irá mostrar a tabela de atributos associada à camada enquanto a terceira apenas mostra uma moldura HTML ligado ao sítio da internet da sua organização. Configure o *Número de páginas* que deseja. Pode escolher a página *Orientação* e a sua *Resolução exportada*. Quando activa a opção *imprimir como raster* indica que todos os elementos serão rasterizados antes de ser imprimidos ou guardados como PostScript ou PDF.
- *Grid* lets you customize grid settings like *spacings*, *offsets* and *tolerance* to your need.
- In *Snap to alignments*, you can change the *Tolerance*, which is the maximum distance below which an item is snapped to smart guides.

A atracção à grelha e/ou às guias inteligentes podem ser activadas a partir do menu *Vista*. Neste menu, pode também esconder ou mostrar a grelha e as guias inteligentes.

19.1.6 Composer items common options

Composer items have a set of common properties you will find on the bottom of the *Item Properties* tab: Position and size, Rotation, Frame, Background, Item ID and Rendering (See [figure_composer_common_1](#)).

- A janela *Posição e tamanho* permite que defina o tamanho e posição da moldura que contém o item. Pode também escolher que *Ponto de referência* será configurado nas coordenadas **X** e **Y** previamente definidas.
- A :guilabel:'otação' define a rotação do elemento (em graus).
- The *Frame* shows or hides the frame around the label. Click on the **[Color]** and **[Thickness]** buttons to adjust those properties.
- The *Background* enables or disables a background color. Click on the **[Color...]** button to display a dialog where you can pick a color or choose from a custom setting. Transparency can also be adjusted through the **alpha** field.

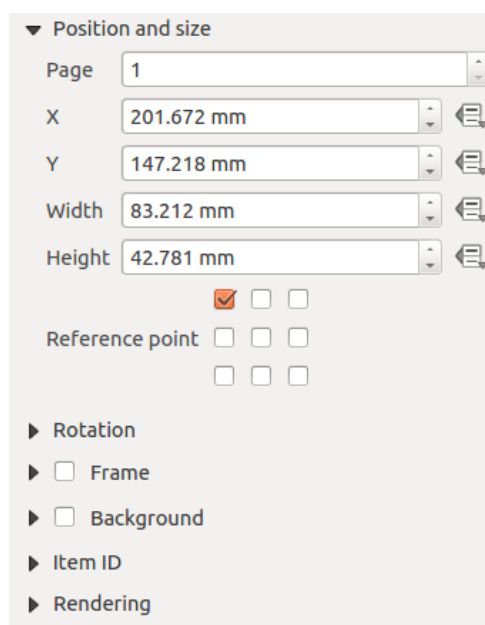





Figura 19.2: Janela comum das propriedades do item 

- Use o *Item ID* para criar uma relação para outro item do Compositor de Impressão. Isto é usado com o QGIS server e em qualquer web client potencial. Pode definir um ID num item (ex.: um mapa e um rótulo), e de seguida o web client pode enviar dados para definir a propriedade (ex.: texto do rótulo) para esse item específico. O comando `GetProjectSettings` irá listar que itens e que IDs estão disponíveis no layout.
- o modo *Renderização* pode ser seleccionado no campo opção. Veja [Rendering_Mode](#) .

Note:

- If you checked  *Use live-updating color chooser dialogs* in the QGIS general options, the color button will update as soon as you choose a new color from **Color Dialog** windows. If not, you need to close the **Color Dialog**.
- The  *Data defined override* icon next to a field means that you can associate the field with data in the map item or use expressions. These are particularly helpful with atlas generation (See [atlas_data_defined_overrides](#)).

19.2 Modo de Renderização

O QGIS permite agora renderização avançada para os itens do Compositor como para as camadas vectoriais e rasters.

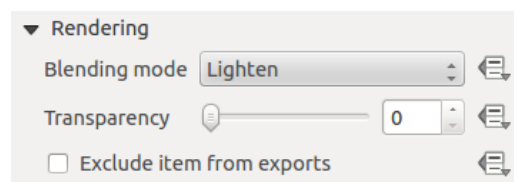






Figura 19.3: Modo de Renderização 

- *Transparency* : You can make the underlying item in the Composer visible with this tool. Use the slider to adapt the visibility of your item to your needs. You can also make a precise definition of the percentage of visibility in the the menu beside the slider.

-  *Exclude item from exports*: You can decide to make an item not visible in all exports. After activating this checkbox, the item will not be included in PDF's, prints etc..
- *Modo de mistura*: pode alcançar efeitos especiais de renderização com estas ferramentas que anteriormente só era conhecido de programas gráficos. Os pixels dos itens sobrejacente e subjacente são misturados através das configurações descritas em baixo.
 - Normal: Este é o modo padrão de renderização que usa o canal alfa do pixel superior para renderizar com o pixel abaixo dele; as cores não estão misturadas.
 - Mais claro: Selecciona o máximo de cada componente dos pixels de primeiro plano e de fundo. Tenha atenção que os resultados tendem a ser irregulares e “áspero”.
 - Ecrã: Os pixels claros da fonte são pintados sobre o destino e os pixels escuros não são. Este modo é muito útil para misturar texturas de uma camada com outra camada. (Ex.: pode usar o mapa de sombras como textura noutra camada).
 - Subexposição: A subexposição irá clarear e saturar os pixels subjacentes baseados na luminosidade do pixel superior. A maior claridade do pixel superior causa o aumento da saturação e brilho dos pixels subjacentes. Isto funciona melhor nos pixels superiores que não brilham muito, caso contrário o efeito é muito extremo.
 - Adição: Este modo de renderização simplesmente adiciona os valores dos pixels de uma camada noutra. Nos casos que os valores são acima de 1 (como no caso do RGB), o branco é exibido. Este modo é adequado para destacar elementos.
 - Escurecido: Cria um pixel resultante que retém os componentes mais pequenos dos pixels do primeiro plano e do fundo. Como o mais claro, o resultado tende a ser irregular e “áspero”.
 - Multiplicar: Aqui, os números de cada pixel superior da camada multiplica com o pixel correspondente da camada abaixo. Os resultados são imagens mais escuras.
 - Queimar: As cores escuras da camada superior torna mais escuro as camadas subjacentes. Pode ser usado para ajustar e colorizar camadas subjacentes.
 - Sobreposição: É uma combinação entre os modos de renderização de multiplicar e ecrã. Como resultado as partes claras da imagem tornam-se mais claras e as partes escuras ficam mais escuras.
 - Pouca luz: Muito semelhante à sobreposição, mas em vez de usar multiplicar/ecrã usa o queimar/subexposição. Neste modo é suposto imitar brilhar uma luz suave numa imagem.
 - Muita luz: Este modo é muito semelhante ao modo de sobreposição. É suposto simular a projecção de uma luz muito intensa numa imagem.
 - Diferença: A diferença subtrair o pixel superior com pixel de baixo e vice-versa, para obter sempre o valor positivo. A mistura com pretos não produz alteração, como valor todas as cores são zero.
 - Subtracção: Este modo de renderização simplesmente subtrair os valores do pixel de uma camada à outra. Em caso de valores negativos, o preto é exibido.

19.3 Itens do Compositor


19.3.1 The Map item




Clique no botão da barra de ferramentas  Adicionar novo mapa do compositor de impressão para adicionar ao enquadramento do mapa QGIS. Agora arraste um rectângulo para o enquadramento do compositor com o botão esquerdo do rato para adicionar o mapa. Para exibir o mapa actual, pode escolher entre três modos diferentes no separador do mapa *Propriedades do Item Properties*:

- **Rectângulo** é a configuração padrão. Apenas exhibe uma caixa vazia com a mensagem ‘O mapa será impresso aqui’.

- **Cache** renderiza um mapa para a resolução actual do ecrã. Se quiser aproximar ou afastar na janela do Compositor, o mapa não irá renderizar mas a imagem é redimensionada.
- **Desenhar** significa, que se aproximar ou afastar na janela do compositor, o mapa será novamente renderizada, mas por razões de espaço, apenas para a resolução máxima.

Cache é o modo de pré-visualização padrão para novos mapas adicionados no Compositor de Impressão.

Pode redimensionar o elemento mapa clicando no botão  Seleccionar/Mover item, seleccionado o elemento, e arrastando um dos manipuladores azuis do canto do mapa. Com o mapa seleccionado, pode agora adaptar mais propriedades no mapa no separador *Propriedades do Item*

To move layers within the map element, select the map element, click the  Move item content icon and move the layers within the map item frame with the left mouse button. After you have found the right place for an item, you can lock the item position within the Print Composer canvas. Select the map item and use the toolbar  Lock Selected Items or the *Items* tab to Lock the item. A locked item can only be selected using the *Items* tab. Once selected you can use the *Items* tab to unlock individual items. The  Unlock All Items icon will unlock all locked composer items.

Propriedades principais

The *Main properties* dialog of the map *Item Properties* tab provides the following functionalities (see [figure_composer_map_1](#)):

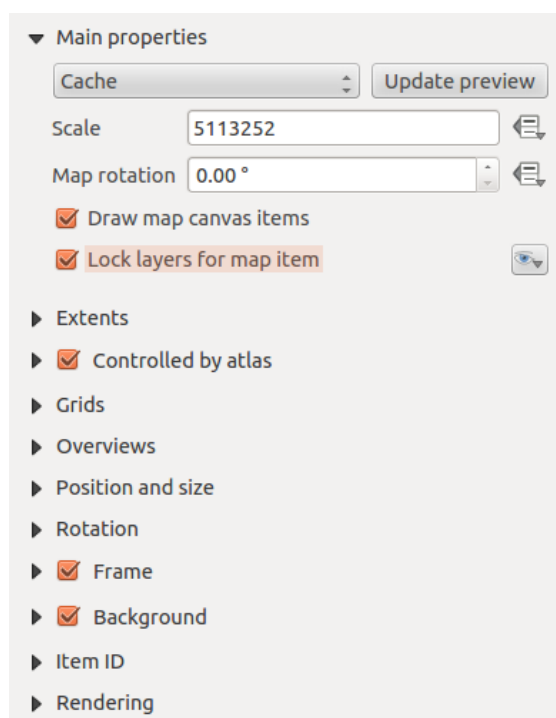






Figura 19.4: Separador das propriedades da Mapa 

- A área de **Pré-visualização** permite que defina modos de pré-visualização ‘Rectângulo’, ‘Cache’ e ‘Desenhar’, como é descrito em cima. Se mudar a vista do enquadramento do mapa QGIS ampliando ou movendo ou alterando o vector ou alterando as propriedades do raster, pode actualizar a vista do Compositor de Impressão seleccionando o elemento mapa no Compositor de Impressão e clicando no botão **[Actualizar pré-visualização]**.
- O campo *Escala* define a escala manual.

- The field *Rotation* allows you to rotate the map element content clockwise in degrees. Note that a coordinate frame can only be added with the default value 0.
- O *Desenhar itens de mapa* permite exibir anotações que podem ser posicionados no enquadramento do mapa na janela principal do QGIS.
- You can choose to lock the layers shown on a map item. Check *Lock layers for map item*. After this is checked, any layer that would be displayed or hidden in the main QGIS window will not appear or be hidden in the map item of the Composer. But style and labels of a locked layer are still refreshed according to the main QGIS interface.
- The  button allows you to add quickly all the presets views you have prepared in QGIS. Clicking on the  button you will see the list of all the preset views: just select the preset you want to display. The map canvas will automatically lock the preset layers by enabling the *Lock layers for map item*: if you want to unselect the preset, just uncheck the and press on the  button. See *Legenda do Mapa* to find out how to create presets views.

Extensões

The *Extents* dialog of the map item tab provides the following functionalities (see [figure_composer_map_2](#)):

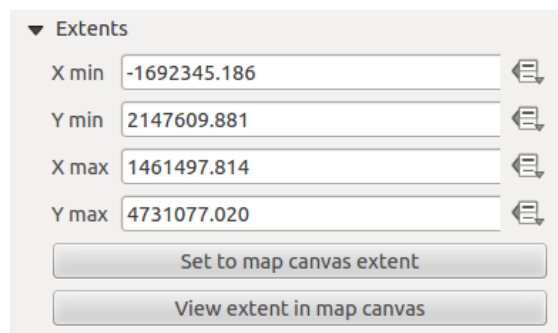


Figura 19.5: Janela de Extensões do Mapa

- The **Map extents** area allows you to specify the map extent using X and Y min/max values and by clicking the [**Set to map canvas extent**] button. This button sets the map extent of the composer map item to the extent of the current map view in the main QGIS application. The button [**View extent in map canvas**] does exactly the opposite, it updates the extent of the map view in the QGIS application to the extent of the composer map item.

If you change the view on the QGIS map canvas by changing vector or raster properties, you can update the Print Composer view by selecting the map element in the Print Composer and clicking the [**Update preview**] button in the map *Item Properties* tab (see [figure_composer_map_1](#)).

Grids

The *Grids* dialog of the map *Item Properties* tab provides the the possibility to add several grids to a map item.

- With the plus and minus button you can add or remove a selected grid.
- With the up and down button you can move a grid in the list and set the drawing priority.

When you double click on the added grid you can give it another name.

After you have added a grid, you can active the checkbox *Show grid* to overlay a grid onto the map element. Expand this option to provides a lot of configuration options, see [Figure_composer_map_4](#).

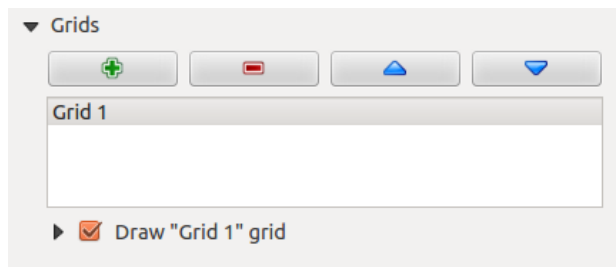



Figura 19.6: Map Grids Dialog 

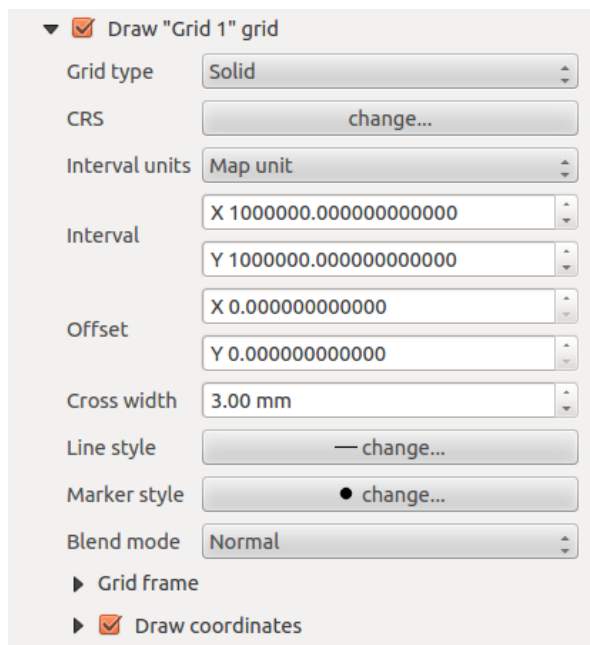


Figura 19.7: Draw Grid Dialog 

As grid type, you can specify to use a solid line or cross. Symbology of the grid can be chosen. See section [Rendering_Mode](#). Furthermore, you can define an interval in the X and Y directions, an X and Y offset, and the width used for the cross or line grid type.

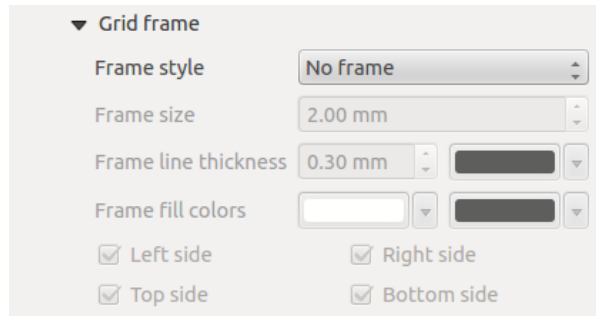


Figura 19.8: Grid Frame Dialog 

- There are different options to style the frame that holds the map. Following options are available: No Frame, Zebra, Interior ticks, Exterior ticks, Interior and Exterior ticks and Lineborder.
- Advanced rendering mode is also available for grids (see section [Rendering_mode](#)).
- The *Draw coordinates* checkbox allows you to add coordinates to the map frame. The annotation can be drawn inside or outside the map frame. The annotation direction can be defined as horizontal, vertical, horizontal and vertical, or boundary direction, for each border individually. Units can be in meters or in degrees. Finally, you can define the grid color, the annotation font, the annotation distance from the map frame and the precision of the drawn coordinates.

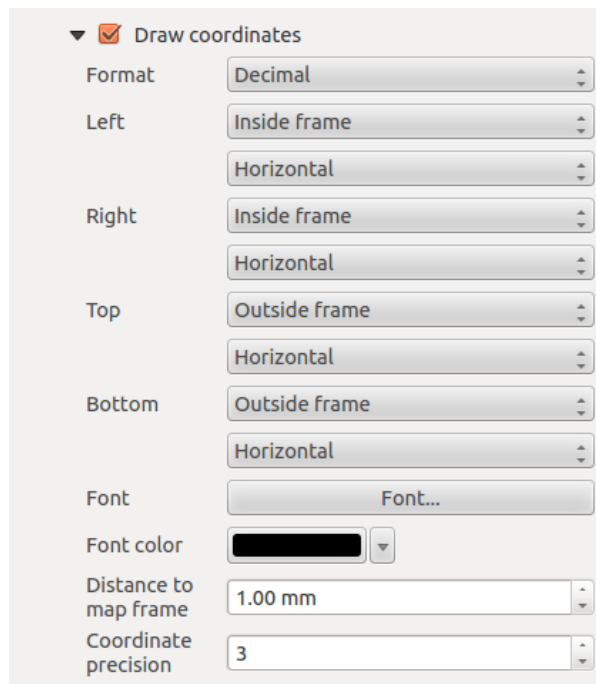


Figura 19.9: Grid Draw Coordinates dialog 

Overviews

The *Overviews* dialog of the map *Item Properties* tab provides the following functionalities:

You can choose to create an overview map, which shows the extents of the other map(s) that are available in the composer. First you need to create the map(s) you want to include in the overview map. Next you create the map

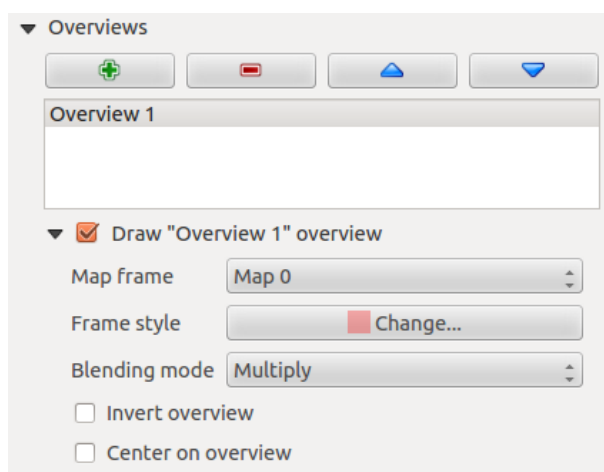


Figura 19.10: Map Overviews Dialog 

you want to use as the overview map, just like a normal map.


- With the plus and minus button you can add or remove an overview.
- With the up and down button you can move an overview in the list and set the drawing priority.

Open *Overviews* and press the green plus icon-button to add an overview. Initially this overview is named ‘Overview 1’ (see [Figure_composer_map_7](#)). You can change the name when you double-click on the overview item in the list named ‘Overview 1’ and change it to another name.

When you select the overview item in the list you can customize it.

- The *Draw “<name_overview>” overview* needs to be activated to draw the extent of selected map frame.
- The *Map frame* combo list can be used to select the map item whose extents will be drawn on the present map item.
- The *Frame Style* allows you to change the style of the overview frame.
- The *Blending mode* allows you to set different transparency blend modes. See [Rendering_Mode](#).
- The *Invert overview* creates a mask around the extents when activated: the referenced map extents are shown clearly, whereas everything else is blended with the frame color.
- The *Center on overview* puts the extent of the overview frame in the center of the overview map. You can only activate one overview item to center, when you have added several overviews.

19.3.2 The Label item

Para adicionar uma etiqueta clique no ícone  Adicionar nova etiqueta, posicione o elemento com o botão esquerdo do rato no enquadramento do compositor de impressão e personalize a aparência no separador da etiqueta *Propriedades do Item*.

The *Item Properties* tab of a label item provides the following functionality for the label item (see [Figure_composer_label](#)):

Propriedades principais

- A janela das propriedades principais é onde inserido o texto (HTML ou não) ou a expressão necessária para preencher a etiqueta adicionada ao enquadramento do compositor.

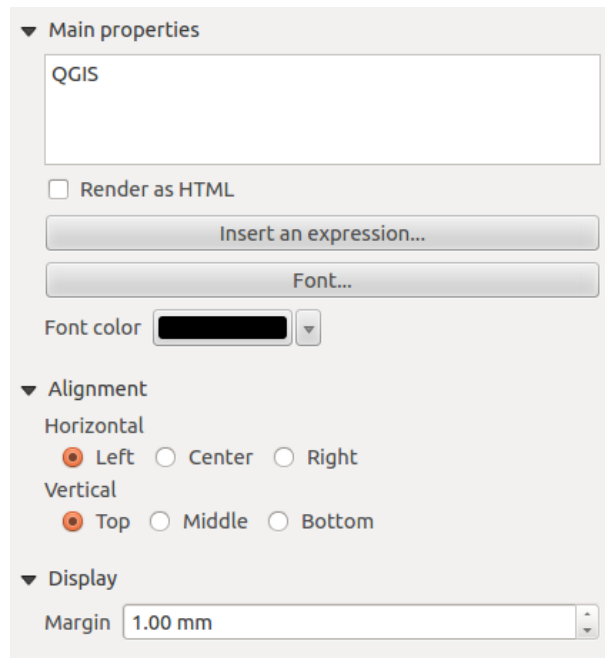





Figura 19.11: Separador das propriedades da Etiqueta 

- As etiquetas podem ser interpretadas como código HTML: activa a  *Renderizar como HTML*. Pode agora inserir um URL, uma imagem com link que por sua vez irá ligar-se a uma página de internet ou outra coisa mais complexa.
- You can also insert an expression. Click on **[Insert an expression]** to open a new dialog. Build an expression by clicking the functions available in the left side of the panel. Two special categories can be useful, particularly associated with the atlas functionality: geometry functions and records functions. At the bottom, a preview of the expression is shown.
- Define *Font* by clicking on the **[Font...]** button or a *Font color* selecting a color using the color selection tool.

Alignment and Display


- You can define the horizontal and vertical alignment in the *Alignment* zone.
- In the **Display** tag, you can define a margin in mm. This is the margin from the edge of the composer item.

19.3.3 The Image item

Para adicionar uma imagem, clique no ícone  *Adicionar imagem*, posicione o elemento com o botão esquerdo do rato no enquadramento do Compositor de Impressão e posicione e personalize a sua aparência no separador *Propriedades Item* da imagem.

The image *Item Properties* tab provides the following functionalities (see [figure_composer_image_1](#)):

You first have to select the image you want to display. There are several ways to set the *image source* in the **Main properties** area.

1. Use the browse button  of *image source* to select a file on your computer using the browse dialog. The browser will start in the SVG-libraries provided with QGIS. Besides SVG, you can also select other image formats like .png or .jpg.
2. You can enter the source directly in the *image source* text field. You can even provide a remote URL-address to an image.

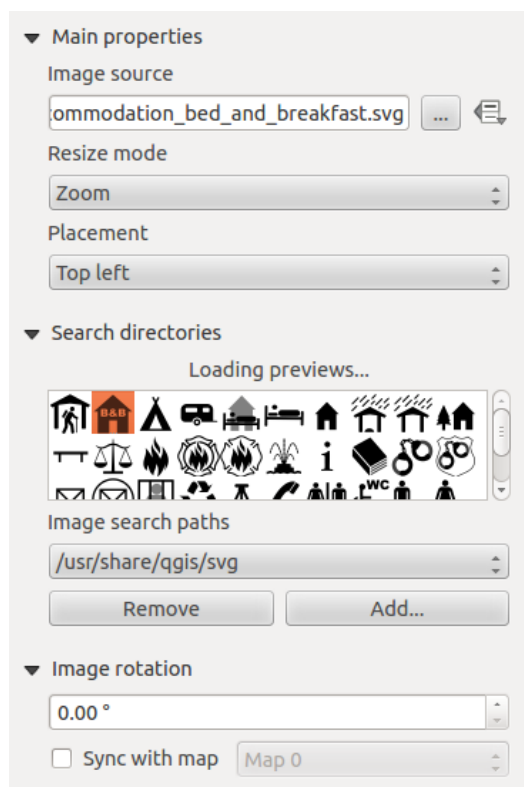



Figura 19.12: Separador Propriedades do Item da Imagem 🐧

3. From the **Search directories** area you can also select an image from *loading preview..* to set the image source.
4. Use the data defined button  to set the image source from a record or using a regular expression.

With the *Resize mode* option, you can set how the image is displayed when the frame is changed, or choose to resize the frame of the image item so it matches the original size of the image.


You can select one of the following modes:

- **Zoom:** Enlarges the image to the frame while maintaining aspect ratio of picture.
- **Stretch:** Stretches image to fit inside the frame, ignores aspect ratio.
- **Clip:** Use this mode for raster images only, it sets the size of the image to original image size without scaling and the frame is used to clip the image, so only the part of the image inside the frame is visible.
- **Zoom and resize frame:** Enlarges image to fit frame, then resizes frame to fit resultant image.
- **Resize frame to image size:** Sets size of frame to match original size of image without scaling.

Selected resize mode can disable the item options 'Placement' and 'Image rotation'. The *Image rotation* is active for the resize mode 'Zoom' and 'Clip'.

With *Placement* you can select the position of the image inside its frame. The **Search directories** area allows you to add and remove directories with images in SVG format to the picture database. A preview of the pictures found in the selected directories is shown in a pane and can be used to select and set the image source.

Images can be rotated with the *Image rotation* field. Activating the *Sync with map* checkbox synchronizes the rotation of a picture in the QGIS map canvas (i.e., a rotated north arrow) with the appropriate Print Composer image.

It is also possible to select a north arrow directly. If you first select a north arrow image from **Search directories** and then use the browse button  of the field *Image source*, you can now select one of the north arrow from the list as displayed in [figure_composer_image_2](#).

Note: Many of the north arrows do not have an ‘N’ added in the north arrow, this is done on purpose for languages that do not use an ‘N’ for North, so they can use another letter.

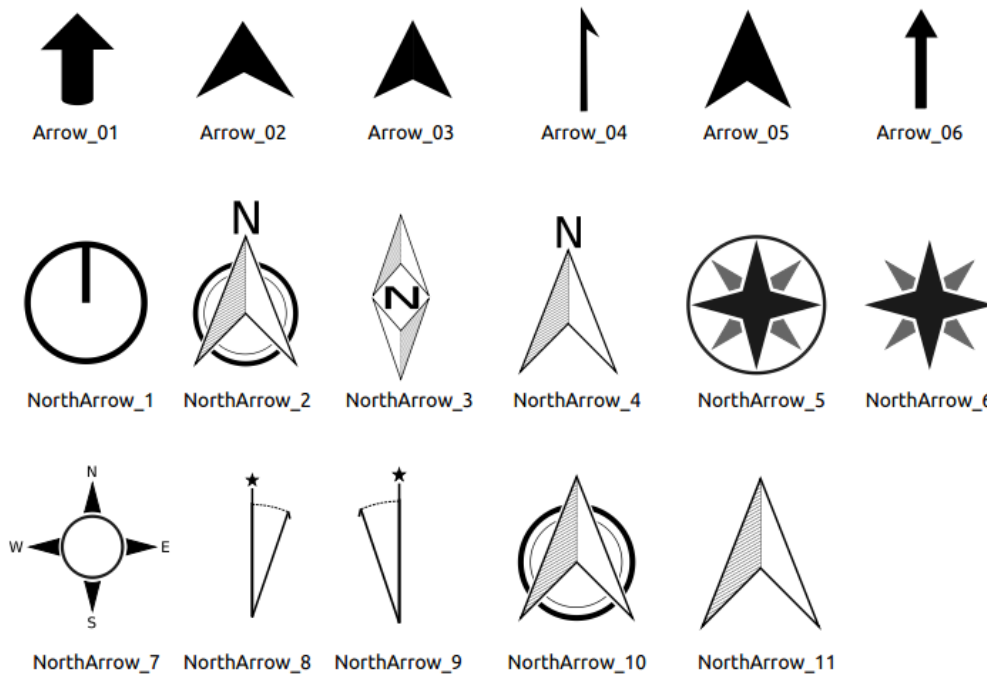



Figura 19.13: North arrows available for selection in provided SVG library

19.3.4 The Legend item

Para adicionar uma legenda de mapa, clique no ícone  Adicionar nova legenda, posicione o elemento com o botão esquerdo do rato no enquadramento do compositor de impressão e posicione e personalize a sua aparência no separador da legenda *Propriedades do Item*.

The *Item properties* of a legend item tab provides the following functionalities (see [figure_composer_legend_1](#)):

Propriedades principais

The *Main properties* dialog of the legend *Item Properties* tab provides the following functionalities (see [figure_composer_legend_2](#)):

In Main properties you can:

- Change the title of the legend.
- Set the title alignment to Left, Center or Right.
- You can choose which *Map* item the current legend will refer to in the select list.
- You can wrap the text of the legend title on a given character.

Itens Legenda

The *Legend items* dialog of the legend *Item Properties* tab provides the following functionalities (see [figure_composer_legend_3](#)):

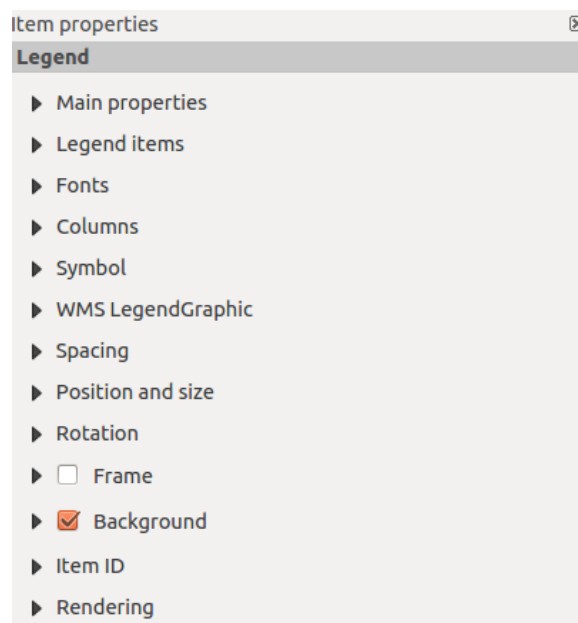


Figura 19.14: Propriedades do Separador da Legenda 🐧

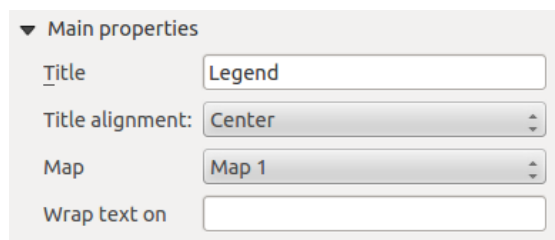


Figura 19.15: Janela das Propriedades principais da Legenda 🐧

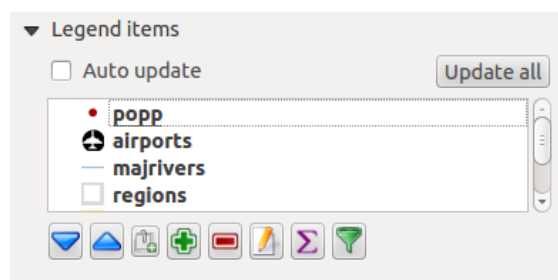


Figura 19.16: Legenda da Janela dos itens da Legenda 🐧

- The legend will be updated automatically if *Auto-update* is checked. When *Auto-update* is unchecked this will give you more control over the legend items. The icons below the legend items list will be activated.
- The legend items window lists all legend items and allows you to change item order, group layers, remove and restore items in the list, edit layer names and add a filter.
 - The item order can be changed using the **[Up]** and **[Down]** buttons or with ‘drag-and-drop’ functionality. The order can not be changed for WMS legend graphics.
 - Use the **[Add group]** button to add a legend group.
 - Use the **[plus]** and **[minus]** button to add or remove layers.
 - The **[Edit]** button is used to edit the layer-, groupname or title, first you need to select the legend item.
 - The **[Sigma]** button adds a feature count for each vector layer.
 - Use the **[filter]** button the filter the legend by map content, only the legend items visible in the map will be listed in the legend.

After changing the symbology in the QGIS main window, you can click on **[Update]** to adapt the changes in the legend element of the Print Composer.

Fonts, Columns, Symbol

The *Fonts*, *Columns* and *Symbol* dialogs of the legend *Item Properties* tab provide the following functionalities (see [figure_composer_legend_4](#)):

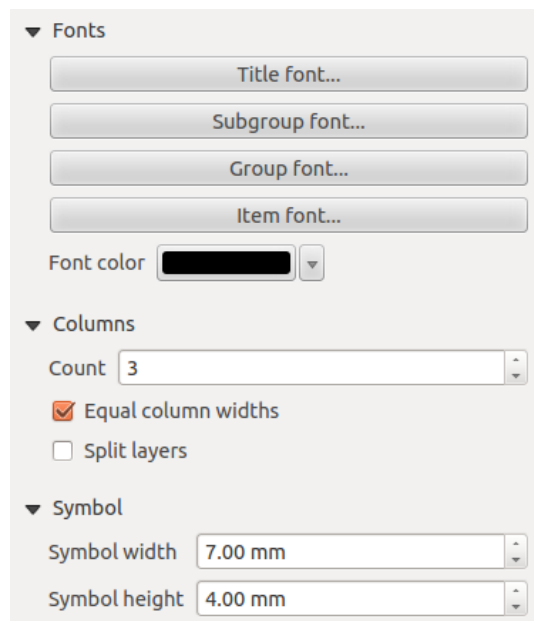



Figura 19.17: Legenda da Janela Fontes, Colunas, Símbolos e Espaçamento 

- Pode mudar a fonte do título, grupo, subgrupo, e item (camada) no item legenda. Clique no botão da categoria para abrir a janela de **Selecionar fonte**.
- You provide the labels with a **Color** using the advanced color picker, however the selected color will be given to all font items in the legen..
- Legend items can be arranged over several columns. Set the number of columns in the *Count* field.
 - A *Largura igual de colunas* define como as colunas da legenda devem ser ajustadas.
 - A opção *Dividir camadas* permite categorizar ou graduar a legenda da camada para ser dividida mediante as colunas.

- Pode alterar a largura e altura para o símbolo da legenda nesta janela.

WMS legendGraphic and Spacing

The *WMS legendGraphic* and *Spacing* dialogs of the legend *Item Properties* tab provide the following functionalities (see [figure_composer_legend_5](#)):

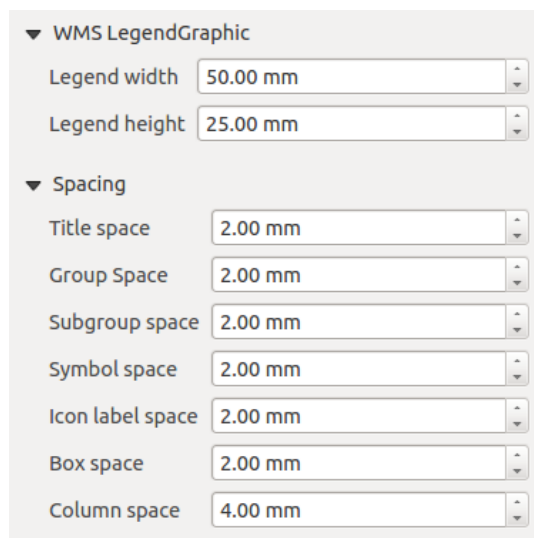



Figura 19.18: WMS legendGraphic Dialogs 

When you have added a WMS layer and you insert a legend composer item, a request will be send to the WMS server to provide a WMS legend, This Legend will only be shown if the WMS server provides the GetLegendGraphic capability. The WMS legend content will be provided as a raster image.

WMS legendGraphic is used to be able to adjust the *Legend width* and the *legend hight* of the WMS legend raster image.

Spacing around title, group, subgroup, symbol, icon label, box space or column space can be customized through this dialog.

19.3.5 The Scale Bar item

Para adicionar uma barra de escala, clique no ícone  Adicionar nova Escala Gráfica, posicione o elemento com o botão esquerdo do rato no enquadramento do Compositor de Impressão, posicione e personalize a sua aparência no separador da legenda *Propriedades do Item*.

The *Item properties* of a scale bar item tab provides the following functionalities (see [figure_composer_scalebar_1](#)):

Propriedades principais

The *Main properties* dialog of the scale bar *Item Properties* tab provides the following functionalities (see [figure_composer_scalebar_2](#)):

- Primeiro, escolha o mapa a que a barra de escala será ligada.
- De seguida, escolha o estilo da sua escala gráfica. Estão disponíveis seis estilos:
 - Os estilos **Caixa simples** e **Caixa dupla** que contêm uma ou duas linhas de caixas de cores alternadas.
 - Espessura de linhas do **Meio**, **Acima** ou **Abaixo**,

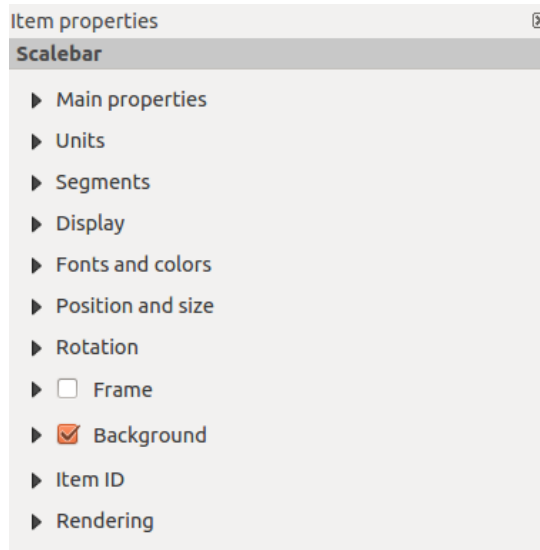


Figura 19.19: Separador das propriedades da Escala Gráfica 🐧

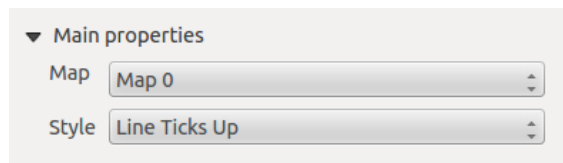


Figura 19.20: Janela de Propriedades principais da Escala Gráfica 🐧

- **Numérica** : o rácio de escala é imprimida (ex.: 1:50000).

Unidades e Segmentos

The *Units* and *Segments* dialogs of the scale bar *Item Properties* tab provide the following functionalities (see [figure_composer_scalebar_3](#)):

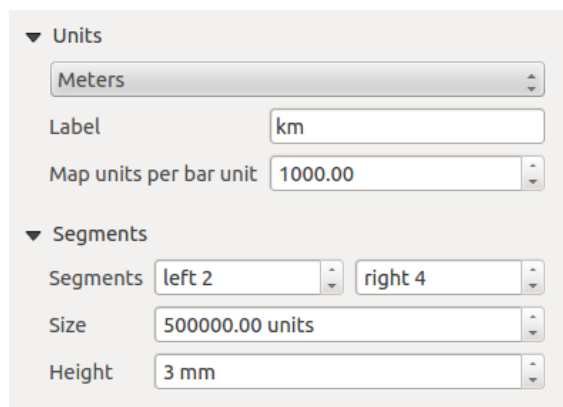


Figura 19.21: Janelas das Unidades da Escala Gráfica e Segmentos 🐧

Nestas duas janelas, pode configurar como a escala gráfica será representada.

- Select the map units used. There are four possible choices: **Map Units** is the automated unit selection; **Meters**, **Feet** or **Nautical Miles** force unit conversions.
- O campo *Etiqueta* define o texto usado para descrever a unidade da escala gráfica.

- As *Unidades de Mapa por unidades de escala* permitem fixar uma proporção entre as unidades de mapa e a sua representação na escala gráfica.
- Pode definir quantos *Segmentos* serão desenhados no lado esquerdo e direito da escala gráfica, e qual o tamanho terá cada segmento (campo *Tamanho*). A *Altura* pode também ser definida.

Display

The *Display* dialog of the scale bar *Item Properties* tab provide the following functionalities (see [figure_composer_scalebar_4](#)):

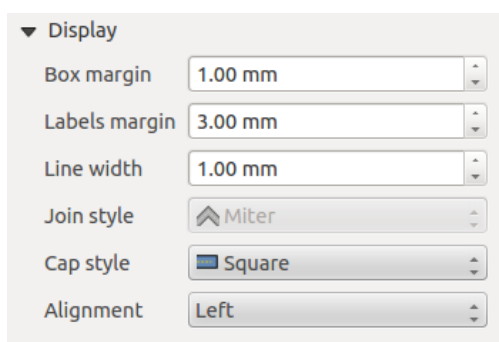


Figura 19.22: Scale Bar Display 

You can define how the scale bar will be displayed in its frame.

- *Box margin* : space between text and frame borders
- *Labels margin* : space between text and scale bar drawing
- *Line width* : line width of the scale bar drawing
- *Join style* : Corners at the end of scalebar in style Bevel, Rounded or Square (only available for Scale bar style Single Box & Double Box)
- *Cap style* : End of all lines in style Square, Round or Flat (only available for Scale bar style Line Ticks Up, Down and Middle)
- *Alignment* : Puts text on the left, middle or right side of the frame (works only for Scale bar style Numeric)

Fonts and colors

The *Fonts and colors* dialog of the scale bar *Item Properties* tab provide the following functionalities (see [figure_composer_scalebar_5](#)):

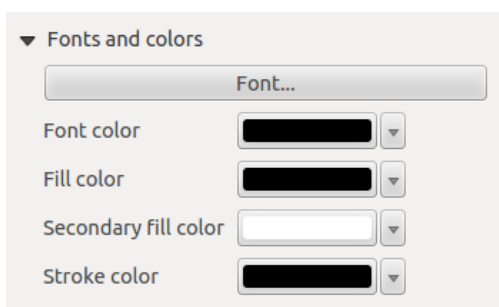




Figura 19.23: Scale Bar Fonts and colors Dialogs 

You can define the fonts and colors used for the scale bar.

- Use the **[Font]** button to set the font
- *Font color*: set the font color
- *Fill color*: set the first fill color
- *Secondary fill color*: set the second fill color
- *Stroke color*: set the color of the lines of the Scale Bare

Fill colors are only used for scale box styles Single Box and Double Box. To select a color you can use the list option using the dropdown arrow to open a simple color selection option or the more advanced color selection option, that is started when you click in the colored box in the dialog.

19.3.6 The Basic Shape Items

To add a basic shape (ellipse, rectangle, triangle), click the  Add basic shape icon or the  Add Arrow icon, place the element holding down the left mouse. Customize the appearance in the *Item Properties* tab.

When you also hold down the `Shift` key while placing the basic shape you can create a perfect square, circle or triangle.

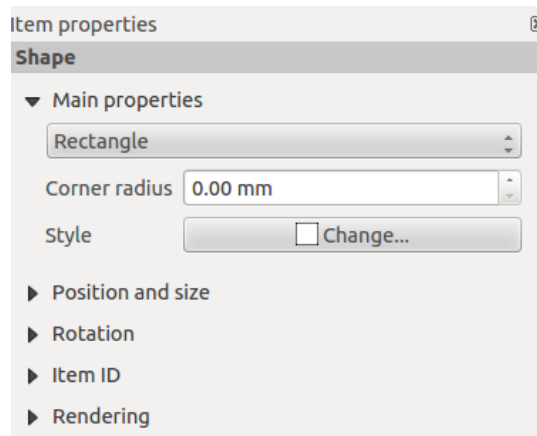



Figura 19.24: Separador de propriedades da Forma 


The *Shape* item properties tab allows you to select if you want to draw an ellipse, rectangle or triangle inside the given frame.

You can set the style of the shape using the advanced symbol style dialog with which you can define its outline and fill color, fill pattern, use markers etcetera.

For the rectangle shape, you can set the value of the corner radius to round of the corners.

Note: Unlike other items, you can not style the frame or the background color of the frame.

19.3.7 The Arrow item

To add an arrow, click the  Add Arrow icon, place the element holding down the left mouse button and drag a line to draw the arrow on the Print Composer canvas and position and customize the appearance in the scale bar *Item Properties* tab.

When you also hold down the `Shift` key while placing the arrow, it is placed in an angle of exactly 45° .

The arrow item can be used to add a line or a simple arrow that can be used, for example, to show the relation between other print composer items. To create a north arrow, the image item should be considered first. QGIS

has a set of North arrows in SVG format. Furthermore you can connect an image item with a map so it can rotate automatically with the map (see [the_image_item](#)).

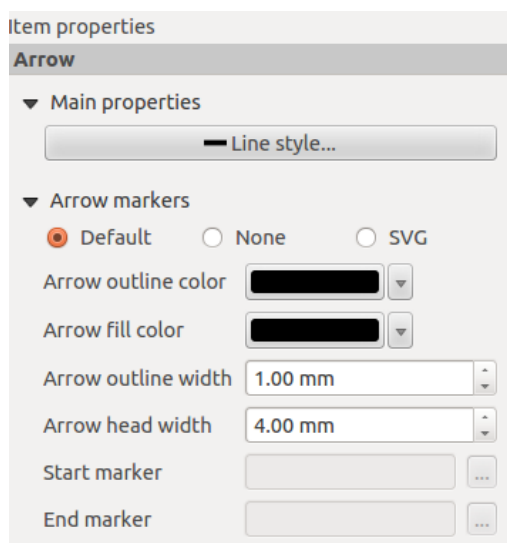



Figura 19.25: Separador das propriedades da Seta 

Item Properties

The *Arrow* item properties tab allows you to configure an arrow item.

The [**Line style ...**] button can be used to set the line style using the line style symbol editor.

In *Arrows markers* you can select one of three radio buttons.

- *Default* : To draw a regular arrow, gives you options to style the arrow head
- *None* : To draw a line without arrow head
- *SVG Marker* : To draw a line with an *SVG Start marker* and/or *End marker*

For *Default* Arrow marker you can use following options to style the arrow head.


- *Arrow outline color* : Set the outline color of the arrow head
- *Arrow fill color* : Set the fill color of the arrow head
- *Arrow outline width* : Set the outline width of the arrow head
- *Arrow head width*: Set the size of the arrow head

For *SVG Marker* you can use following options.

- *Start marker* : Choose an SVG image to draw at the beginning of the line
- *End marker* : Choose an SVG image to draw at the end of the line
- *Arrow head width*: Sets the size of Start and/or headmarker

SVG images are automatically rotated with the line. The color of the SVG image can not be changed.

19.3.8 The Attribute Table item

It is possible to add parts of a vector attribute table to the Print Composer canvas: Click the  Add attribute table icon, place the element with the left mouse button on the Print Composer canvas, and position and customize the appearance in the *Item Properties* tab.

The *Item properties* of an attribute table item tab provides the following functionalities (see [figure_composer_table_1](#)):

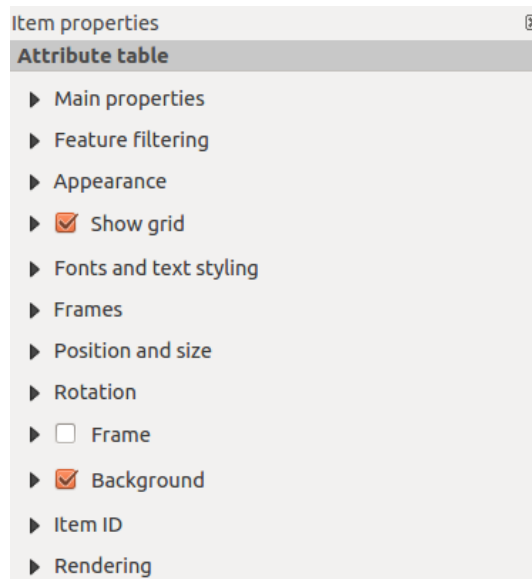



Figura 19.26: Attribute table Item properties Tab 

Propriedades principais

The *Main properties* dialogs of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_2](#)):

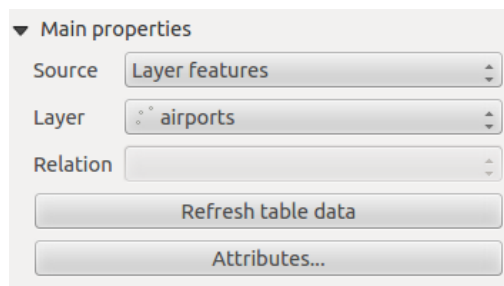



Figura 19.27: Attribute table Main properties Dialog 

- For *Source* you can normally select only 'Layer features'.
- With *Layer* you can choose from the vector layers loaded in the project.
- The button **[Refresh table data]** can be used to refresh the table when the actual contents of the table has changed.
- The button **[Attributes...]** starts the *Select attributes* menu, see [figure_composer_table_3](#), that can be used to change the visible contents of the table. After making changes use the **[OK]** button to apply changes to the table.

In the *Columns* section you can:

- Remove an attribute, just select an attribute row by clicking anywhere in a row and press the minus button to remove the selected attribute.
- Add a new attribute use the plus button. At the end a new empty row appears and you can select empty cell of the column *Attribute*. You can select a field attribute from the list or you can select to build a new attribute using a regular expression.

- Use the up and down arrows to change the order of the attributes in the table.
- Select a cel in the Headings column to change the Heading, just type a new name.
- Select a cel in the Alignment column and you can choose between Left, Center or Right alignment.
- Select a cel in the Width column and you can change it from Automatic to a width in mm, just type a number. When you want to change it back to Automatic, use the cross.
- The **[Reset]** button can allways be used to restore it to the original attribute settings.

In the *Sorting* section you can:

- Add an attribute to sort the table with. Select an attribute and set the sorting order to 'Ascending' or 'Descending' and press the plus button. A new line is added to the sort order list.
- select a row in the list and use the up and down button to change the sort priority on attribute level.
- use the minus button to remove an attribute from the sort order list.

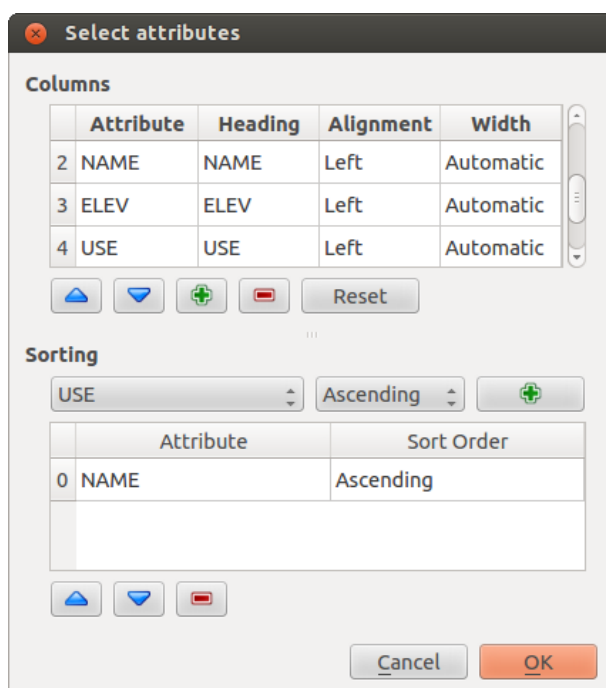



Figura 19.28: Janela Secção de atributos da tabela de atributos 

Feature filtering

The *Feature filtering* dialogs of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_4](#)):

You can:

- Define the *Maximum rows* to be displayed.
- Activate *Remove duplicate rows from table* to show unique records only.
- Activate *Show only visible features within a map* and select the corresponding *Composer map* to display the attributes of features only visible on selected map.
- Activate *Show only features intersecting Atlas feature* is only available when *Generate an atlas* is activated. When activated it will show a table with only the features shown on the map of that particular page of the atlas.

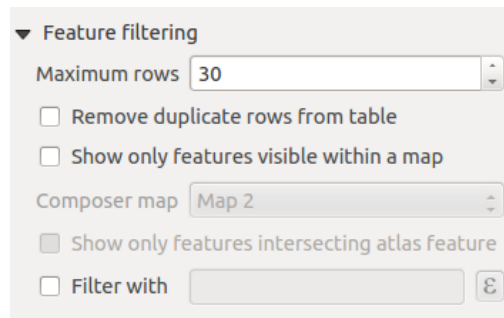



Figura 19.29: Attribute table Feature filtering Dialog 

- Activate *Filter with* and provide a filter by typing in the input line or insert a regular expressing use the given expression button. A few examples of filtering statements you can use when you have loaded the airports layer from the Sample dataset:
 - ELEV > 500
 - NAME = ' ANIAK'
 - NAME NOT LIKE ' AN%
 - regexp_match(attribute(\$currentfeature, 'USE') , '[i]')

The last regular expression will include only the arpoirts that have a letter 'i' in the attribute field 'USE'.

Appearance

The *Appearance* dialogs of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_5](#)):

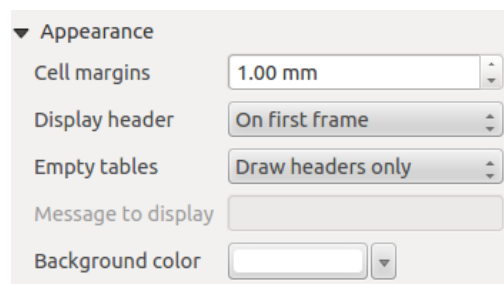



Figura 19.30: Attribute table appearance Dialog 

- With *Cell margins* you can define the margin around text in each cell of the table.
- With *Display header* you can select from a list one of 'On first frame', 'On all frames' default option, or 'No header'.
- The option *Empty table* controls what will be displayed when the result selection is empty.
 - **Draw headers only**, will only draw the header except if you have choosen 'No header' for *Display header*.
 - **Hide entire table**, will only draw the background of the table. You can activate *Don't draw background if frame is empty* in *Frames* to completely hide the table.
 - **Draw empty cells**, will fill the attribute table with empty cells, this option can also be used to provide additional empty cells when you have a result to show!
 - **Show set message**, will draw the header and adds a cell spanning all columns and display a message like 'No result' that can be provided in the option *Message to display*

- The option *Message to display* is only activated when you have selected **Show set message** for *Empty table*. The message provided will be shown in the table in the first row, when the result is an empty table.
- With *Background color* you can set the background color of the table.

Show grid

The *Show grid* dialog of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_6](#)):

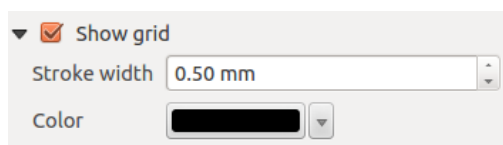



Figura 19.31: Attribute table Show grid Dialog 

- Activate *Show grid* when you want to display the grid, the outlines of the table cells.
- With *Stroke width* you can set the thickness of the lines used in the grid.
- The *Color* of the grid can be set using the color selection dialog.

Fonts and text styling

The *Fonts and text styling* dialog of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_7](#)):

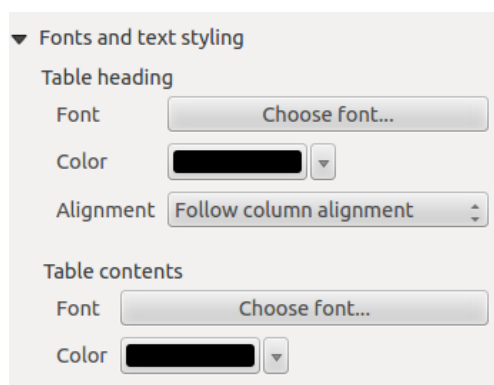


Figura 19.32: Attribute table Fonts and text styling Dialog 

- You can define *Font* and *Color* for *Table heading* and *Table contents*.
- For *Table heading* you can additionally set the *Alignment* and choose from *Follow column alignment*, *Left*, *Center* or *Right*. The column alignment is set using the *Select Attributes* dialog (see [Figure_composer_table_3](#)).

Frames

The *Frames* dialog of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_8](#)):

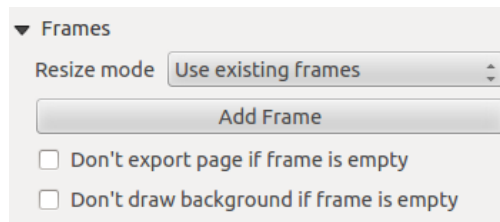




Figura 19.33: Attribute table Frames Dialog 

- With *Resize mode* you can select how to render the attribute table contents:
 - *Use existing frames* displays the result in the first frame and added frames only.
 - *Extent to next page* will create as many frames (and corresponding pages) as necessary to display the full selection of attribute table. Each frame can be moved around on the layout. If you resize a frame, the resulting table will be divided up between the other frames. The last frame will be trimmed to fit the table.
 - *Repeat until finished* will also create as many frames as the *Extend to next page* option, except all frames will have the same size.
- Use the **[Add Frame]** button to add another frame with the same size as selected frame. The result of the table that will not fit in the first frame will continue in the next frame when you use the *Resize mode Use existing frames*.
- Activate *Don't export page if frame is empty* prevents the page to be exported when the table frame has no contents. This means all other composer items, maps, scalebars, legends etc. will not be visible in the result.
- Activate *Don't draw background if frame is empty* prevents the background to be drawn when the table frame has no contents.

19.3.9 The HTML frame item


It is possible to add a frame that displays the contents of a website or even create and style your own HTML page and display it!

Click the  **Add HTML frame** icon, place the element by dragging a rectangle holding down the left mouse button on the Print Composer canvas and position and customize the appearance in the *Item Properties* tab (see [figure_composer_html_1](#)).

HTML Source

As an HTML source, you can either set a URL and activate the URL radiobutton or enter the HTML source directly in the textbox provided and activate the Source radiobutton.

The *HTML Source* dialog of the HTML frame *Item Properties* tab provides the following functionalities (see [figure_composer_html_2](#)):

- In *URL* you can enter the URL of a webpage you copied from your internet browser or select an HTML file using the browse button . There is also the option to use the Data defined override button, to provide an URL from the contents of an attribute field of a table or using a regular expression.
- In *Source* you can enter text in the textbox with some HTML tags or provide a full HTML page.
- The **[insert an expression]** button can be used to insert an expression like `[%Year($now)%]` in the Source textbox to display the current year. This button is only activated when radiobutton *Source* is selected. After inserting the expression click somewhere in the textbox before refreshing the HTML frame, otherwise you will lose the expression.

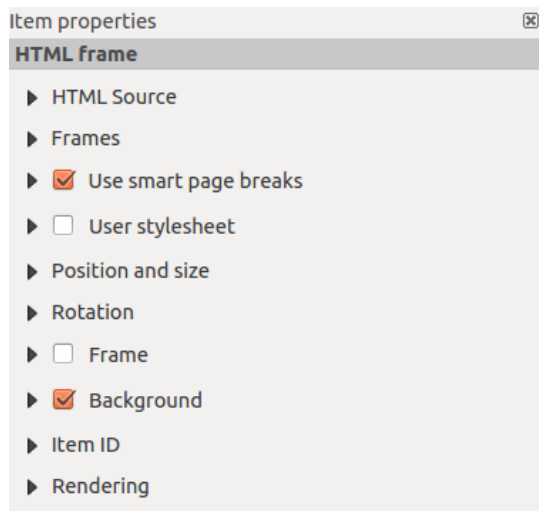



Figura 19.34: HTML frame, the item properties Tab 

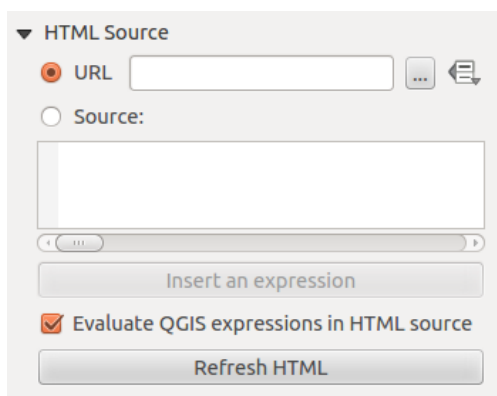



Figura 19.35: HTML frame, the HTML Source properties 

- Activate *Evaluate QGIS expressions in HTML code* to see the result of the expression you have included, otherwise you will see the expression instead.
- Use the **[Refresh HTML]** button to refresh the HTML frame(s) to see the result of changes.

Frames

The *Frames* dialog of the HTML frame *Item Properties* tab provides the following functionalities (see [figure_composer_html_3](#)):

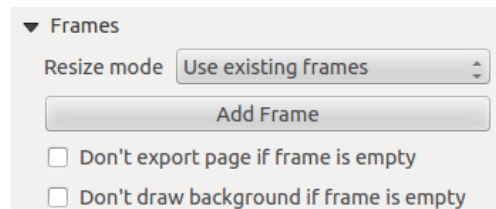



Figura 19.36: HTML frame, the Frames properties 

- With *Resize mode* you can select how to render the HTML contents:
 - *Use existing frames* displays the result in the first frame and added frames only.
 - *Extend to next page* will create as many frames (and corresponding pages) as necessary to render the height of the web page. Each frame can be moved around on the layout. If you resize a frame, the webpage will be divided up between the other frames. The last frame will be trimmed to fit the web page.
 - *Repeat on every page* will repeat the upper left of the web page on every page in frames of the same size.
 - *Repeat until finished* will also create as many frames as the *Extend to next page* option, except all frames will have the same size.
- Use the **[Add Frame]** button to add another frame with the same size as selected frame. If the HTML page that will not fit in the first frame it will continue in the next frame when you use *Resize mode* or *Use existing frames*.
- Activate *Don't export page if frame is empty* prevents the map layout from being exported when the frame has no HTML contents. This means all other composer items, maps, scalebars, legends etc. will not be visible in the result.
- Activate *Don't draw background if frame is empty* prevents the HTML frame being drawn if the frame is empty.

Use smart page breaks and User style sheet

The *Use smart page breaks* dialog and *Use style sheet* dialog of the HTML frame *Item Properties* tab provides the following functionalities (see [figure_composer_html_4](#)):

- Activate *Use smart page breaks* to prevent the html frame contents from breaking mid-way a line of text so it continues nice and smooth in the next frame.
- Set the *Maximum distance* allowed when calculating where to place page breaks in the html. This distance is the maximum amount of empty space allowed at the bottom of a frame after calculating the optimum break location. Setting a larger value will result in better choice of page break location, but more wasted space at the bottom of frames. This is only used when *Use smart page breaks* is activated.

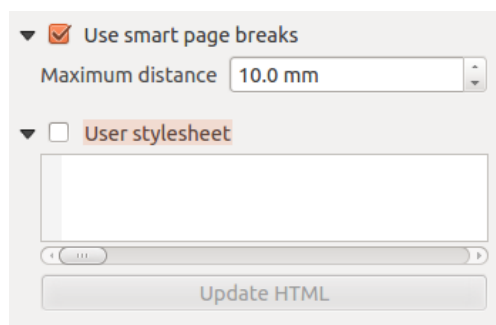



Figura 19.37: HTML frame, Use smart page breaks and User stylesheet properties 


- Activate *User stylesheet* to apply HTML styles that often is provided in cascading style sheets. An example of style code is provide below to set the color of <h1> header tag to green and set the font and fontsize of text included in paragraph tags <p>.

```
h1 {color: #00ff00;
}
p {font-family: "Times New Roman", Times, serif;
font-size: 20px;
}
```

- Use the [Update HTML] button to see the result of the stylesheet settings.


19.4 Manage items

19.4.1 Size and position

Each item inside the Composer can be moved/resized to create a perfect layout. For both operations the first step is to activate the  *Select/Move item* tool and to click on the item; you can then move it using the mouse while holding the left button. If you need to constrain the movements to the horizontal or the vertical axis, just hold the *Shift* while moving the mouse. If you need a better precision, you can move a selected item using the *Arrow* keys on the keyboard; if the movement is too slow, you can speed up it by holding *Shift*.

A selected item will show squares on its boundaries; moving one of them with the mouse, will resize the item in the corresponding direction. While resizing, holding *Shift* will maintain the aspect ratio. Holding *Alt* will resize from the item center.

The correct position for an item can be obtained using snapping to grid or smart guides. Guides are set by clicking and dragging in the rulers. Guide are moved by clicking in the ruler, level with the guide and dragging to a new place. To delete a guide move it off the canvas. If you need to disable the snap on the fly just hold *Ctrl* while moving the mouse.

You can choose multiple items with the  *Select/Move item* button. Just hold the *Shift* button and click on all the items you need. You can then resize/move this group just like a single item.


Once you have found the correct position for an item, you can lock it by using the items on the toolbar or ticking the box next to the item in the *Items* tab. Locked items are **not** selectable on the canvas.

Locked items can be unlocked by selecting the item in the *Items* tab and unchecking the tickbox or you can use the icons on the toolbar.

To unselect an item, just click on it holding the *Shift* button.

Inside the *Edit* menu, you can find actions to select all the items, to clear all selections or to invert the current selection.

19.4.2 Alignment

Raising or lowering functionalities for elements are inside the  Raise selected items pull-down menu. Choose an element on the Print Composer canvas and select the matching functionality to raise or lower the selected element compared to the other elements (see [table_composer_1](#)). This order is shown in the *Items* tab. You can also raise or lower objects in the *Items* tab by clicking and dragging an object's label in this list.

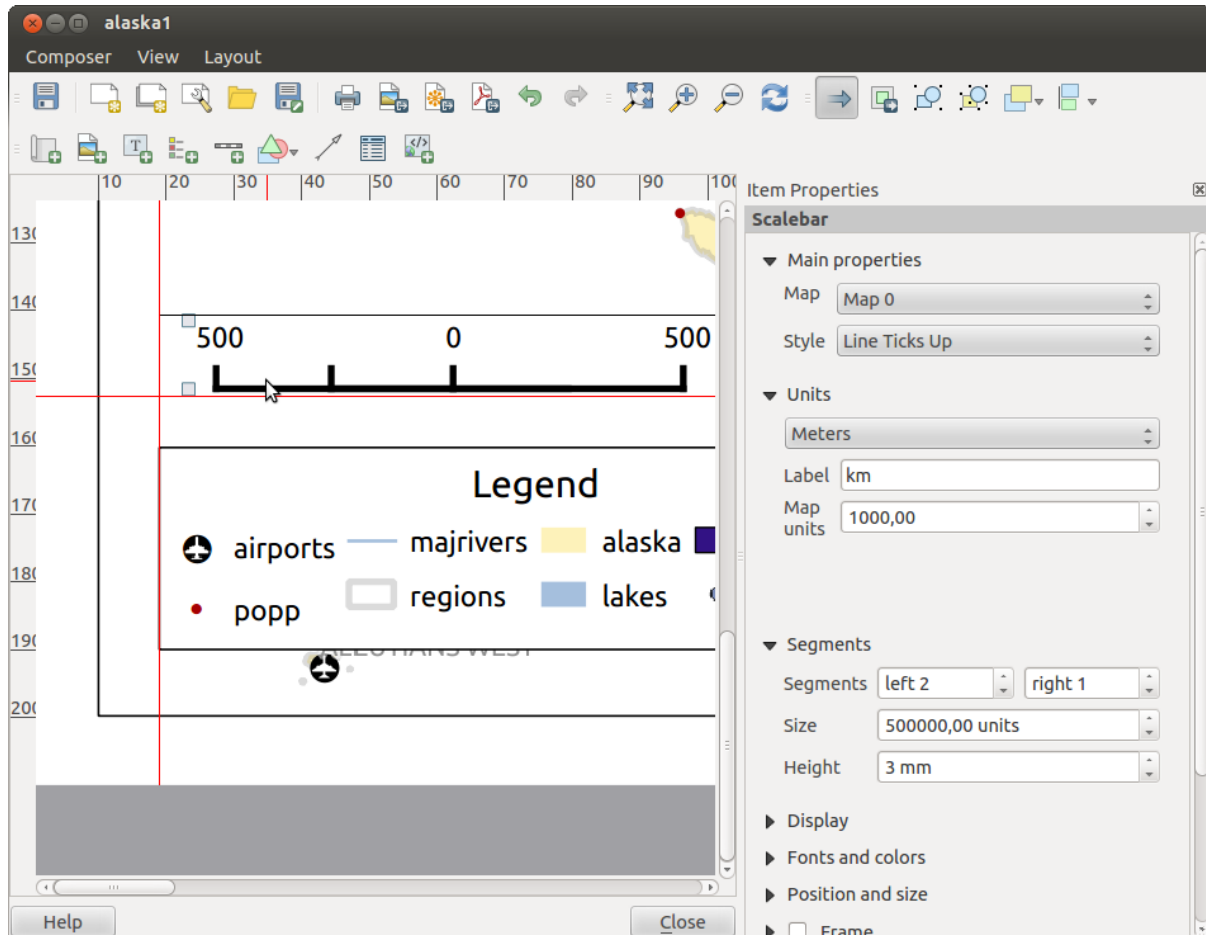




Figura 19.38: Linhas Guia de alinhamento no Compositor de Impressão 

There are several alignment functionalities available within the  Align selected items pull-down menu (see [table_composer_1](#)). To use an alignment functionality, you first select some elements and then click on the matching alignment icon. All selected elements will then be aligned within to their common bounding box. When moving items on the Composer canvas, alignment helper lines appear when borders, centers or corners are aligned.



19.4.3 Copy/Cut and Paste items

The print composer includes actions to use the common Copy/Cut/Paste functionality for the items in the layout. As usual first you need to select the items using one of the options seen above; at this point the actions can be found in the *Edit* menu. When using the Paste action, the elements will be pasted according to the current mouse position.

Note: HTML items can not be copied in this way. As a workaround, use the [Add Frame] button in the *Item Properties* tab.

19.5 Ferramentas de Reverter e Restaurar

During the layout process, it is possible to revert and restore changes. This can be done with the revert and restore tools:

-  Revert last changes
-  Restore last changes

This can also be done by mouse click within the *Command history* tab (see [figure_composer_29](#)).



Figura 19.39: Histórico de comandos no Compositor de Impressão 

19.6 Geração de Atlas

The Print Composer includes generation functions that allow you to create map books in an automated way. The concept is to use a coverage layer, which contains geometries and fields. For each geometry in the coverage layer, a new output will be generated where the content of some canvas maps will be moved to highlight the current geometry. Fields associated with this geometry can be used within text labels.

Every page will be generated with each feature. To enable the generation of an atlas and access generation parameters, refer to the *Atlas generation* tab. This tab contains the following widgets (see [Figure_composer_atlas](#)):

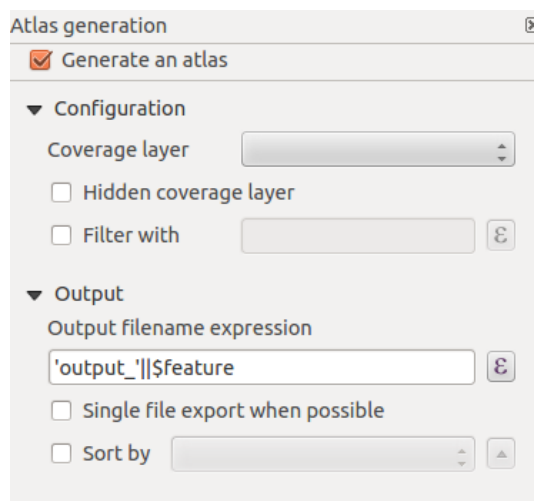





Figura 19.40: Separador de Geração de Atlas 

-  *Generate an atlas*, which enables or disables the atlas generation.
- A *Coverage layer*  combo box that allows you to choose the (vector) layer containing the geometries on which to iterate over.

- An optional *Hidden coverage layer* that, if checked, will hide the coverage layer (but not the other ones) during the generation.
- An optional *Filter with* text area that allows you to specify an expression for filtering features from the coverage layer. If the expression is not empty, only features that evaluate to `True` will be selected. The button on the right allows you to display the expression builder.
- A caixa de texto *Expressão do nome do ficheiro de saída* que é usada para criar o nome do ficheiro para cada geometria se necessária. É baseado em expressões. Este campo é significativo apenas para a renderização de múltiplos ficheiros.
- A *Single file export when possible* that allows you to force the generation of a single file if this is possible with the chosen output format (PDF, for instance). If this field is checked, the value of the *Output filename expression* field is meaningless.
- An optional *Sort by* that, if checked, allows you to sort features of the coverage layer. The associated combo box allows you to choose which column will be used as the sorting key. Sort order (either ascending or descending) is set by a two-state button that displays an up or a down arrow.

You can use multiple map items with the atlas generation; each map will be rendered according to the coverage features. To enable atlas generation for a specific map item, you need to check *Controlled by Atlas* under the item properties of the map item. Once checked, you can set:

- An input box *Margin around feature* that allows you to select the amount of space added around each geometry within the allocated map. Its value is meaningful only when using the auto-scaling mode.
- A *Fixed scale* that allows you to toggle between auto-scale and fixed-scale mode. In fixed-scale mode, the map will only be translated for each geometry to be centered. In auto-scale mode, the map's extents are computed in such a way that each geometry will appear in its entirety.

19.6.1 Labels


In order to adapt labels to the feature the atlas plugin iterates over, you can include expressions. For example, for a city layer with fields `CITY_NAME` and `ZIPCODE`, you could insert this:

```
The area of [% upper(CITY_NAME) || ', ' || ZIPCODE || ' is ' format_number($area/1000000,2) %] km2
```

The information `[% upper(CITY_NAME) || ', ' || ZIPCODE || ' is ' format_number($area/1000000,2) %]` is an expression used inside the label. That would result in the generated atlas as:


The area of PARIS,75001 is 1.94 km2

19.6.2 Data Defined Override Buttons


There are several places where you can use a  Data Defined Override button to override the selected setting. These options are particularly usefull with Atlas Generation.

For the following examples the *Regions* layer of the QGIS sample dataset is used and selected for Atlas Generation. We also assume the paper format *A4 (210X297)* is selected in the *Composite* tab for field *Presets*.


With a *Data Defined Override* button you can dynamically set the paper orientation. When the height (north-south) of the extents of a region is greater than it's width (east-west), you rather want to use *portrait* instead of *landscape* orientation to optimize the use of paper.

In the *Composition* you can set the field *Orientation* and select *Landscape* or *Portrait*. We want to set the orientation dynamically using an expression depending on the region geometry. press the  button of field *Orientation*, select *Edit ...* so the *Expression string builder* dialog opens. Give following expression:


```
CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 'Landscape' ELSE 'Portrait'
```

Now the paper orients itself automatically for each Region you need to reposition the location of the composer item as well. For the map item you can use the  button of field *Width* to set it dynamically using following expression:

```
(CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 297 ELSE 210 END) -
```

Use the  button of field *Height* to provide following expression:

```
(CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 210 ELSE 297 END) -
```

When you want to give a title above map in the center of the page, insert a label item above the map. First use the item properties of the label item to set the horizontal alignment to  *Center*. Next activate from *Reference point* the upper middle checkbox. You can provide following expression for field *X* :

```
(CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 297 ELSE 210 END) /
```

For all other composer items you can set the position in a similar way so they are correctly positioned when page is automatically rotated in portrait or landscape.


Information provided is derived from the excellent blog (in english and portugese) on the Data Defined Override options [Multiple_format_map_series_using_QGIS_2.6](#) .

This is just one example of how you can use Data Defined Overrides.

19.6.3 Preview

Once the atlas settings have been configured and map items selected, you can create a preview of all the pages by clicking on *Atlas* → *Preview Atlas* and using the arrows, in the same menu, to navigate through all the features.





19.6.4 Criação

The atlas generation can be done in different ways. For example, with *Atlas* → *Print Atlas*, you can directly print it. You can also create a PDF using *Atlas* → *Export Atlas as PDF*: The user will be asked for a directory for saving all the generated PDF files (except if the  *Single file export when possible* has been selected). If you need to print just a page of the atlas, simply start the preview function, select the page you need and click on *Composer* → *Print* (or create a PDF).

19.7 Criando um ficheiro de Saída

[Figure_composer_output](#) shows the Print Composer with an example print layout, including each type of map item described in the sections above.

The Print Composer allows you to create several output formats, and it is possible to define the resolution (print quality) and paper size:

- The  **Print** icon allows you to print the layout to a connected printer or a PostScript file, depending on installed printer drivers.
- The  **Export as image** icon exports the Composer canvas in several image formats, such as PNG, BPM, TIF, JPG,...
- The  **Export as PDF** saves the defined Print Composer canvas directly as a PDF.
- The  **Export as SVG** icon saves the Print Composer canvas as an SVG (Scalable Vector Graphic).

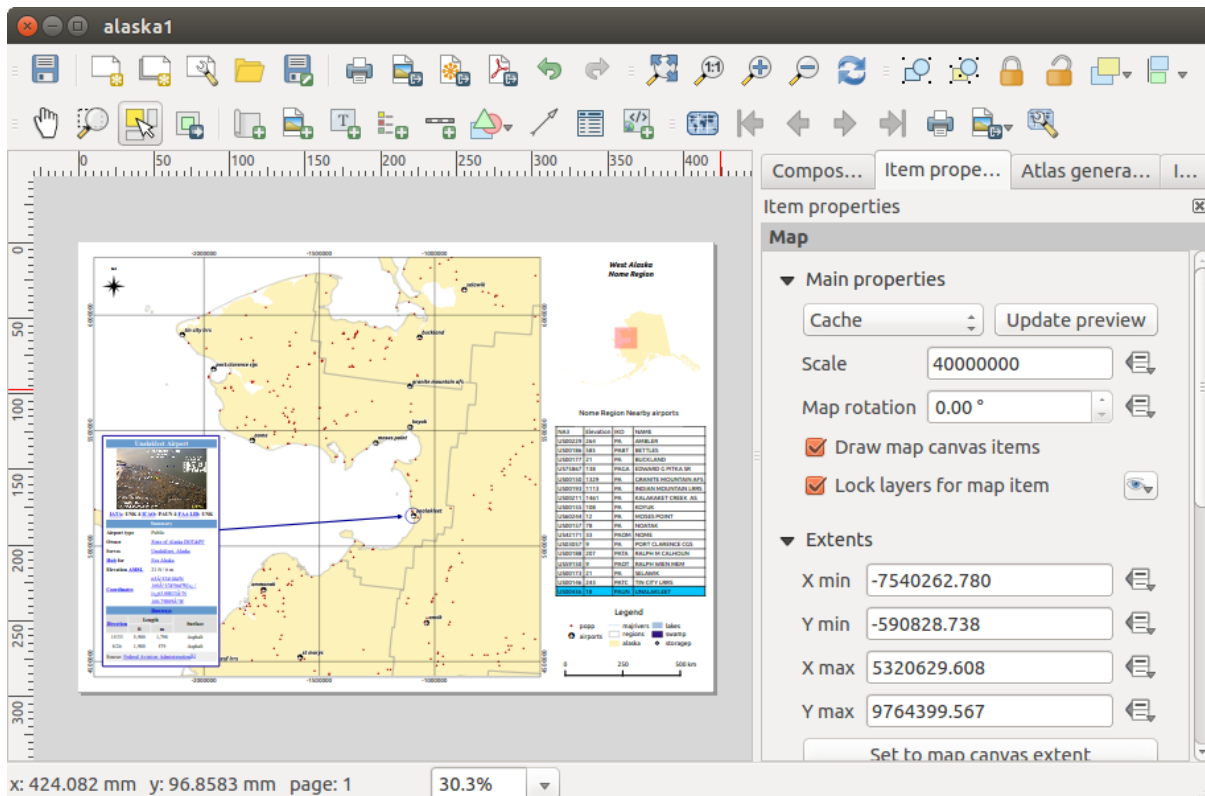





Figura 19.41: Print Composer with map view, legend, image, scale bar, coordinates, text and HTML frame added 


If you need to export your layout as a **georeferenced image** (i.e., to load back inside QGIS), you need to enable this feature under the Composition tab. Check *World file on* and choose the map item to use. With this option, the 'Export as image' action will also create a world file.

Note:



- Currently, the SVG output is very basic. This is not a QGIS problem, but a problem with the underlying Qt library. This will hopefully be sorted out in future versions.
- Exporting big rasters can sometimes fail, even if there seems to be enough memory. This is also a problem with the underlying Qt management of rasters.

19.8 Gerir o Compositor

With the  Save as template and  Add items from template icons, you can save the current state of a Print Composer session as a .qpt template and load the template again in another session.

The  Composer Manager button in the QGIS toolbar and in *Composer* → *Composer Manager* allows you to add a new Composer template, create a new composition based on a previously saved template or to manage already existing templates.

By default, the Composer manager searches for user templates in `~/qgis2/composer_template`.

The  New Composer and  Duplicate Composer buttons in the QGIS toolbar and in *Composer* → *New Composer* and *Composer* → *Duplicate Composer* allow you to open a new Composer dialog, or to duplicate an existing composition from a previously created one.

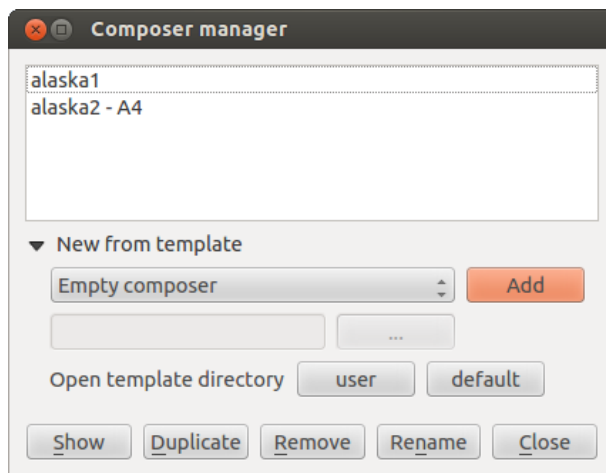


Figura 19.42: O gestor do Compositor de Impressão 

Finally, you can save your print composition with the  Save Project button. This is the same feature as in the QGIS main window. All changes will be saved in a QGIS project file.

20.1 Módulos QGIS

O QGIS tem uma arquitectura de módulos. Isso permite várias novas características/ funções que podem facilmente ser adicionadas na aplicação. Muitas das características no QGIS estão actualmente implementadas como módulos.

You can manage your plugins in the plugin dialog which can be opened with *Plugins > Manage and install plugins*

When a plugin needs to be updated, and if plugins settings have been set up accordingly, QGIS main interface could display a blue link in the status bar to tell you that there are some plugins updating waiting to be applied.

20.1.1 The Plugins Dialog

The menus in the Plugins dialog allow the user to install, uninstall and upgrade plugins in different ways. Each plugin have some metadata displayed in the right panel:

- information if the plugin is experimental
- description
- rating vote(s) (you can vote for your preferred plugin!)
- tags
- some useful links as the home page, tracker and code repository
- author(s)
- version available

You can use the filter to find a specific plugin.



Tudo

Aqui, todos os módulos disponíveis são listados, incluindo os módulos base e externos. Use o **[Actualizar tudo]** para procurar por novas versões dos módulos. Além disso, pode usar **[Instalar módulo]**, se o módulo está listado mas não instalado, e **[Desinstalar módulo]** assim como **[Reinstalar módulo]**, se o módulo está instalado. Se o módulo está instalado, pode ser activado/desactivado usando a caixa de verificação.



Instalado

Neste menu, só encontra apenas os módulos instalados. Os módulos externos podem ser desinstalados e re-instalados usando os botões **[Desinstalar módulo]** and **[Reinstalar módulo]**. Pode também **[Actualizar Tudo]**.

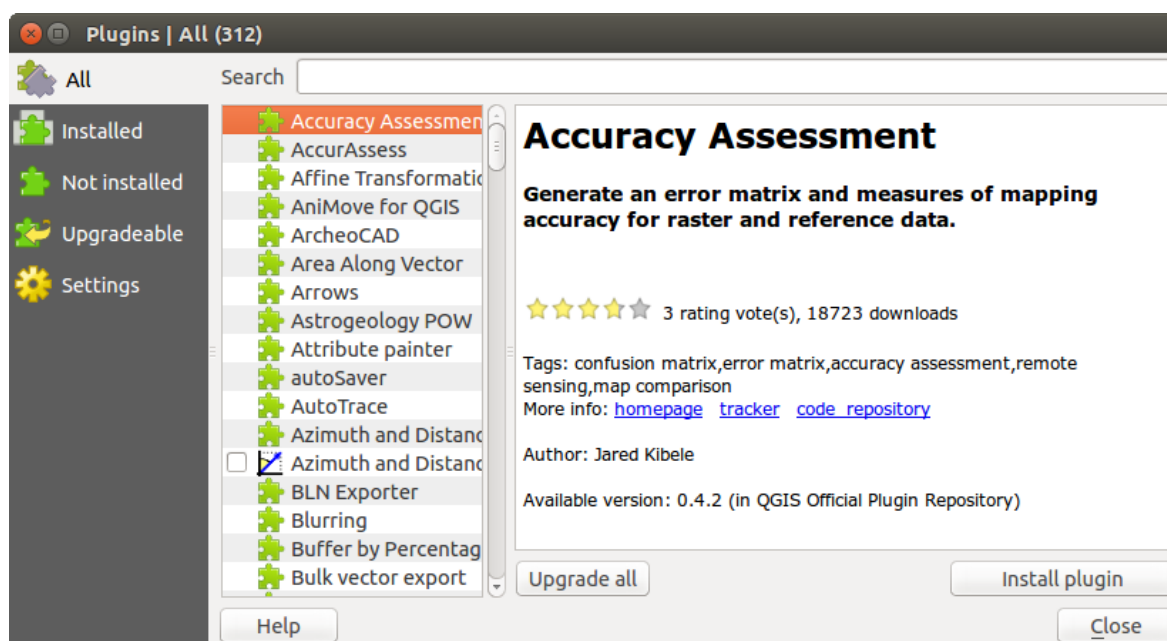




Figura 20.1: O menu  Tudo 

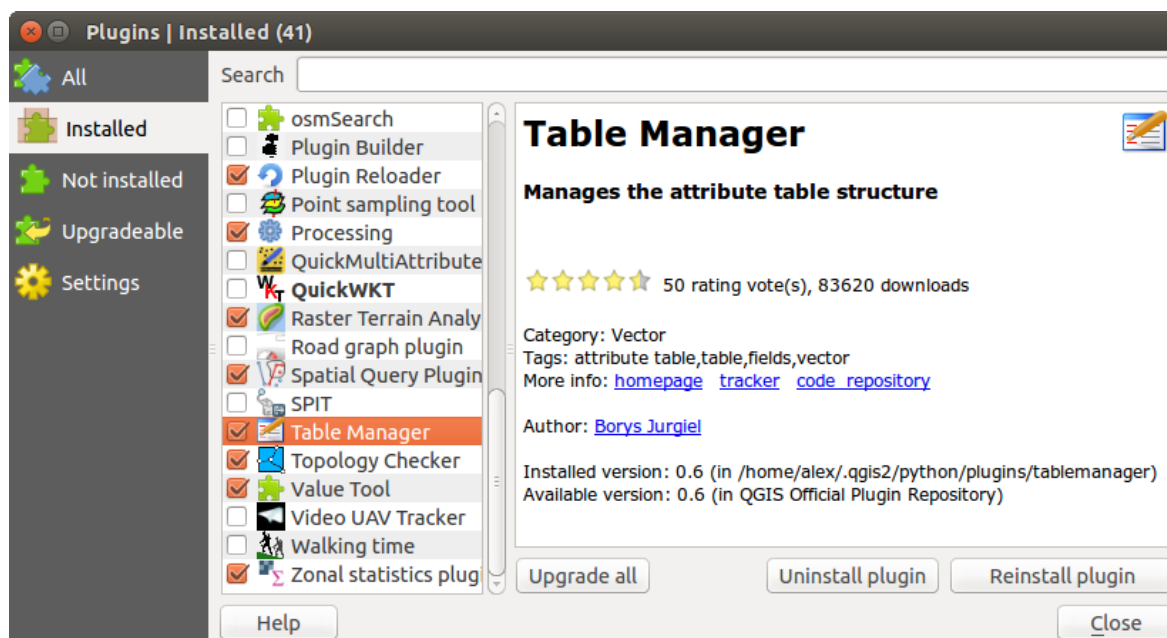





Figura 20.2: O menu  Instalado 

 Não instalado


Este menu lista todas os módulos disponíveis que não estão instalados. Pode usar o botão **[Instalar módulo]** para implementar o módulo no QGIS.



Figura 20.3: O menu  Não instalado 

 Actualizar módulo


Se activar o *Mostrar módulos experimentais* no menu  *Configurações*, pode usar este menu para ver versões mais recentes do módulo. Isto pode ser feito com o botão **[Actualizar módulo]** ou com o **[Actualizar tudo]**

 Configurações

Neste menu, pode usar as seguintes opções:

- *Verificar actualizações ao iniciar*. Quando existe um novo módulo ou está disponível uma actualização de um módulo, o QGIS irá informá-lo ‘todas as vezes que o QGIS iniciar’, ‘uma vez por dia’, ‘a cada 3 dias’, ‘a cada semana’, ‘a cada 2 semanas’ ou ‘a cada mês’.
- *Mostrar módulos experimentais*. O QGIS irá exibir os módulos numa face precoce de desenvolvimento, que é instável para produção de dados.
- *Mostrar módulos obsoletos*. Estes módulos estão obsoletos e geralmente são instáveis para produção de dados.

Para adicionar um repositório externo de um autor, clique **[Adicionar]** na secção *Repositórios de módulos*. Se não quiser um ou mais repositórios adicionados, estes podem ser desactivados através do botão **[Editar...]**, ou remover completamente com o botão **[Apagar]**.

A função *Procurar* está disponível na maior parte do menu (excepto  *Configurações*). Aqui, pode procurar por módulos específicos.

Tip: Módulos base e externos

Os módulos QGIS estão implementados como **Módulos Base** ou **Módulos Externos**. Os **Módulos Base** são mantidos pela Equipa de Desenvolvimento QGIS e vêm automaticamente com qualquer distribuição QGIS. Estes módulos são escritos numa das duas linguagens de programação: C++ ou Python. Os **Módulos Externos** são

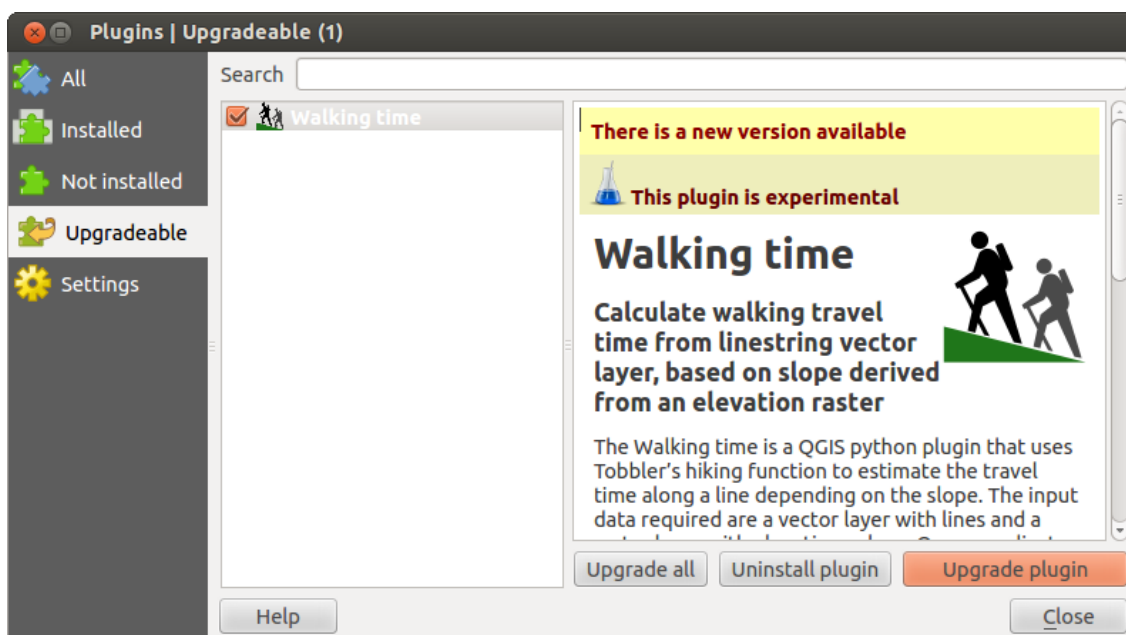


Figura 20.4: O menu  *Atualizar* 

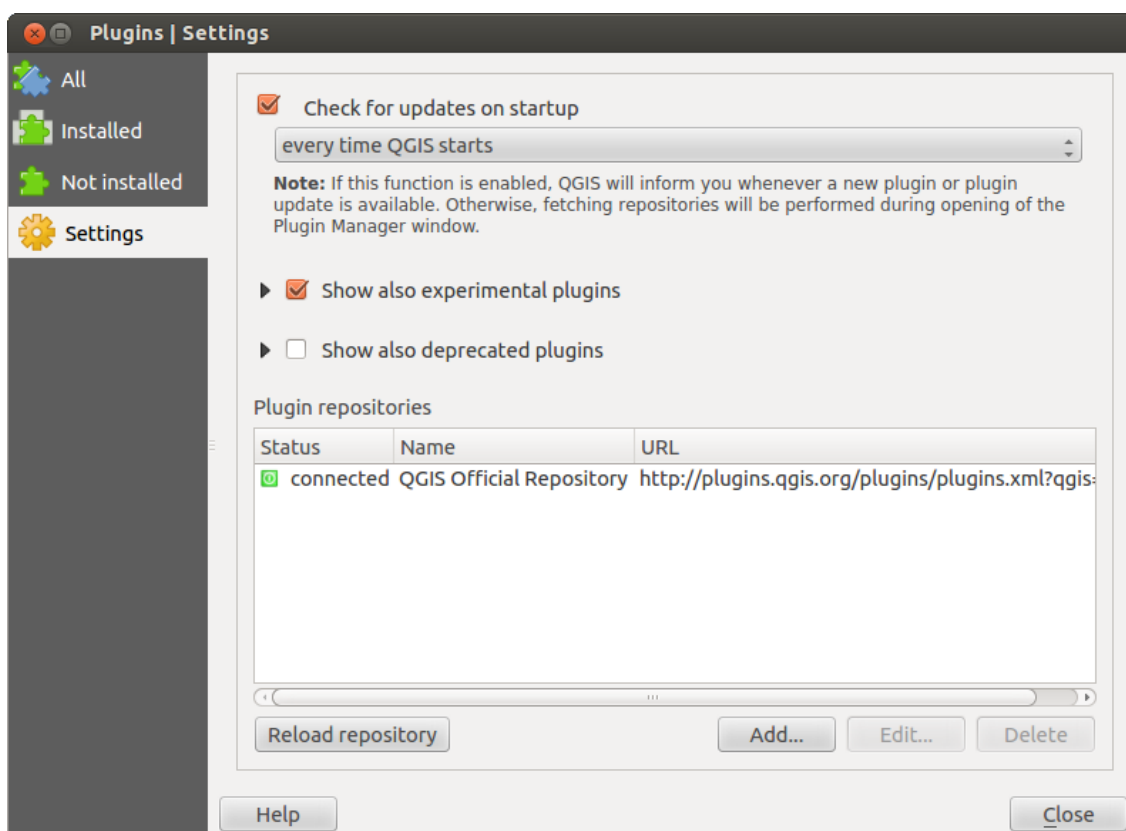


Figura 20.5: O menu  *Configurações* 

todos que actualmente estão escritos em Python. Estes módulos são armazenados em repositórios externos e são mantidos por autores individuais.

Documentação detalhada acerca do seu uso, versão QGIS mínima, Sítio na Internet, autores, e outras informações importantes são fornecidas no Repositório 'Oficial' do QGIS em <http://plugins.qgis.org/plugins/>. Para repositórios externos, pode estar disponível documentação com os módulos externos. Geralmente não incluímos neste manual.

20.2 Usando os Módulos QGIS Core

Ícone	Módulo	Descrição	Referência do Manual
	Captura de Coordenadas	Capture coordenadas com o rato em diferentes SRC	<i>Módulo de Captura de Coordenadas</i>
	Gestor BD	Faça a gestão das suas bases de dados dentro do QGIS	<i>Módulo Gestor BD</i>
	Conversor DXF2Shape	Converte do DXF para o formato de ficheiro SHP	<i>Módulo de Conversão Dxf2Shp</i>
	eVis	Ferramenta de Visualização Event	<i>Módulo eVis</i>
	fTools	Um conjunto de ferramentas vectoriais	<i>Módulo fTools</i>
	Ferramentas GPS	Ferramentas para carregamento e importação de dados GPS	<i>Módulo GPS</i>
	GRASS	Funcionalidade GRASS	<i>Integração GRASS SIG</i>
	Ferramentas GDAL	Funcionalidade matricial GDAL	<i>Módulo de Ferramentas GDAL</i>
	Georeferenciador GDAL	Georeferenciar Rasters com o GDAL	<i>georeferenciar</i>
	Mapa de Densidade	Criar um mapa de densidade a partir da entrada de pontos vectoriais	<i>Módulo de Mapa de Densidade</i>
	Módulo de Interpolação	Interpolação baseada nos vértices da camada vectorial	<i>Módulo de Interpolação</i>
	Edição Offline	Edição Offline e sincronização com a base de dados	<i>Módulo Edição Offline</i>
	Oracle Spatial GeoRaster	Acesso ao Oracle Spatial GeoRasters	<i>Módulo Oracle Spatial GeoRaster</i>
	Gestor de módulos	Gerir módulos core e externos	<i>The Plugins Dialog</i>
	Análise do Terreno Matricial	Computar características geomorfológicas a partir de MDE	<i>Módulo de Análise do Terreno Matricial</i>
	Módulo de Análise de Grafos	Análise do Caminho mais curto	<i>Módulo de Cálculo de Rotas</i>
	Módulo SQL Anywhere	Acesso a BD SQL anywhere	<i>Módulo SQL Anywhere</i>
	Interrogação Espacial	Interrogação espacial nos vectores	<i>Módulo de Consulta Espacial</i>
	SPIT	Ferramenta de Importação Shapefile para PostgreSQL/PostGIS	<i>Módulo SPIT</i>
	Estatísticas Locais	Calcular estatísticas matriciais a partir de polígonos vectoriais	<i>Módulo de Estatística Zonal</i>
	MetaSearch	Interage com os serviços de catálogo de metadados (CSW)	<i>MetaSearch Catalogue Client</i>

20.3 Módulo de Captura de Coordenadas

O módulo de captura de coordenadas é de uso fácil e fornece a habilidade de exibir as coordenadas no enquadramento do mapa em dois Sistema de Referência de Coordenadas (SRC) seleccionados.

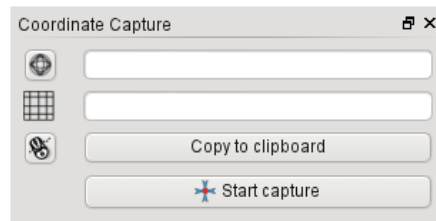










Figura 20.6: Módulo de Captura de Coordenadas 

1. Inicie o QGIS, seleccione  *Propriedades da Camada* a partir do menu *Configurações* (KDE, Windows) ou *Ficheiro* (Gnome, OSX) e clique no separador *Projecção*. Como alternativa pode também clicar no ícone  Estado do SRC status no canto inferior direito da barra de estado.
2. Clique na caixa de verificação  *Activar projecção “on the fly”* e seleccione o sistema de coordenadas projectado à sua escolha (veja também *Trabalhando com Projeções*).
3. Activate the coordinate capture plugin in the Plugin Manager (see *The Plugins Dialog*) and ensure that the dialog is visible by going to *View → Panels* and ensuring that  *Coordinate Capture* is enabled. The coordinate capture dialog appears as shown in Figure [figure_coordinate_capture_1](#). Alternatively, you can also go to *Vector → Coordinate Capture* and see if  *Coordinate Capture* is enabled.
4. Clique no ícone  Clique para seleccionar o SRC para a exibição de coordenadas e seleccione um SRC diferente a partir do que seleccionou acima.
5. Para iniciar a capturar coordenadas, clique em **[Iniciar captura]**. Pode clicar agora em qualquer sítio do enquadramento do mapa e o módulo irá mostrar as coordenadas em ambos os SRC seleccionados.
6. Para activar o rastreo das coordenadas do rato clique no ícone  rastreo do rato.
7. Pode também copiar as coordenadas seleccionadas para a área de transferência.

20.4 Módulo Gestor BD

The DB Manager Plugin is officially part of the QGIS core and is intended to replace the SPIT Plugin and, additionally, to integrate all other database formats supported by QGIS in one user interface. The  DB Manager Plugin provides several features. You can drag layers from the QGIS Browser into the DB Manager, and it will import your layer into your spatial database. You can drag and drop tables between spatial databases and they will get imported. You can also use the DB Manager to execute SQL queries against your spatial database and then view the spatial output for queries by adding the results to QGIS as a query layer.

O menu *Base de Dados* permite ligar a uma base de dados existente, para começar a janela de SQL e sair do módulo Gestor BD. Quando estiver ligado a uma base de dados existente os menus *Esquema* e *Tabela* apareceram adicionalmente.

O menu *Esquema* inclui ferramentas para criar e apagar esquemas (vazias) e, se estiver topologia disponível (ex.: PostgreSQL 2), para iniciar o *TopoViewer*.

O menu *Tabela* permite criar e editar tabelas e apagar as tabelas e as vistas. É também possível esvaziar tabelas e mover tabelas de um esquema para outro. Como mais uma funcionalidade pode executar o comando VÁCUO

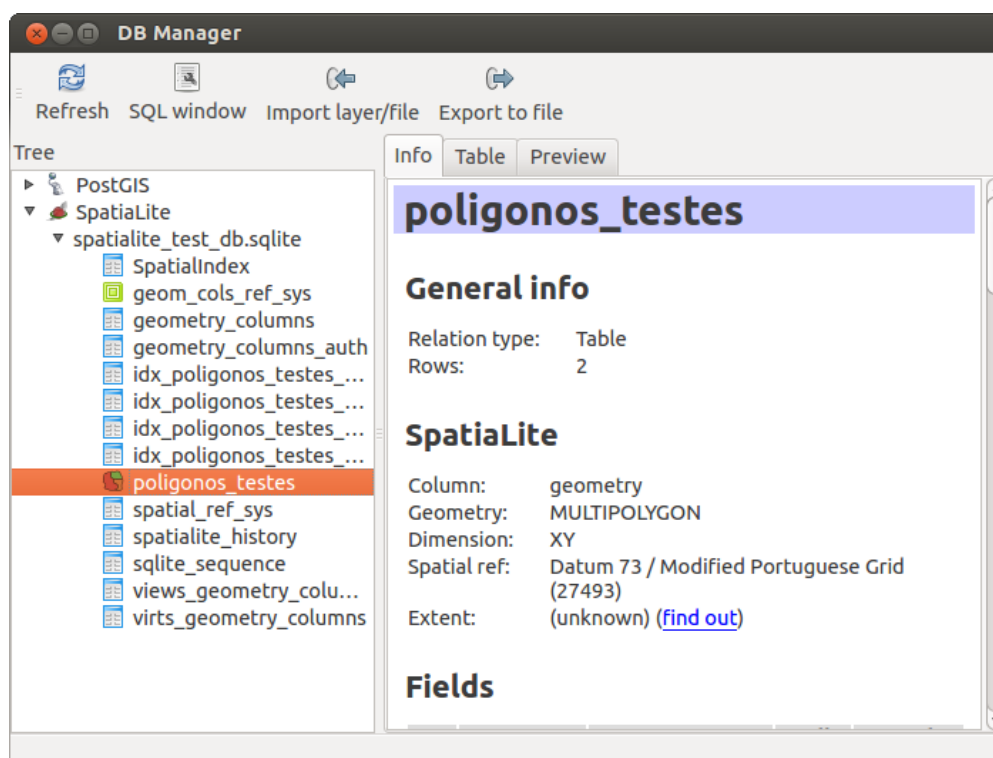



Figura 20.7: Janela do Gestor DB 

e depois o comando ANÁLISE para cada tabela seleccionada. O VÁCUO recupera simplesmente o espaço e torna-o disponível para re-utilizar e ANÁLISE de actualização de estatísticas para determinar o uma maneira mais eficiente de executar uma interrogação. Finalmente, pode importar camadas / ficheiros, se eles forem carregados no QGIS ou existente no sistema de ficheiros. E pode exportar as suas tabelas das base de dados para Shape com a característica de Exportação de Ficheiro.

A janela *Árvore* lista todas as bases de dados existentes suportadas pelo QGIS. Com o duplo clique pode ligar à base de dado. Com o botão direito do rato pode renomear e apagar esquemas e tabelas existentes. As tabelas também podem ser adicionadas ao enquadramento do QGIS com o menu de contexto.

Se estiver ligado à base de dados, a janela **principal** do Gestor DB oferece três separadores. O separador *In-formação* fornecem informação sobre a tabela e a sua geometria assim como os campos existentes, restrições e índices. Permite também correr a Análise de Vácuo e criar um índice espacial na tabela seleccionada, se não estiver actualmente feita. O separador *Tabela* mostra todos os atributos e o separador *Pré-visualização* renderiza as geometrias como pré-visualização.

20.5 Módulo de Conversão Dxf2Shp

The dxf2shape converter plugin can be used to convert vector data from DXF to shapefile format. It requires the following parameters to be specified before running:

- **Input DXF file:** Enter the path to the DXF file to be converted.
- **Output Shp file:** Enter desired name of the shapefile to be created.
- **Output file type:** Specify the geometry type of the output shapefile. Currently supported types are polyline, polygon, and point.
- **Export text labels:** When this checkbox is enabled, an additional shapefile point layer will be created, and the associated DBF table will contain information about the “TEXT” fields found in the DXF file, and the text strings themselves.

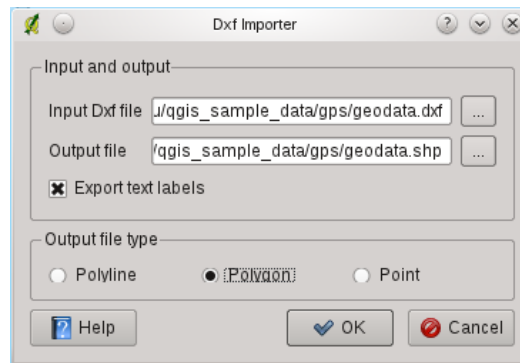



Figura 20.8: Módulo de Conversão Dxf2Shp

20.5.1 Usando o módulo

1. Start QGIS, load the Dxf2Shape plugin in the Plugin Manager (see *The Plugins Dialog*) and click on the  icon, which appears in the QGIS toolbar menu. The Dxf2Shape plugin dialog appears, as shown in *Figure_dxf2shape_1*.
2. Enter the input DXF file, a name for the output shapefile and the shapefile type.
3. Active a caixa de verificação *Exportar rótulos de texto* se quer criar uma camada extra de pontos com os rótulos.
4. Clique [OK].

20.6 Módulo eVis

(This section is derived from Horning, N., K. Koy, P. Ersts. 2009. eVis (v1.1.0) User's Guide. American Museum of Natural History, Center for Biodiversity and Conservation. Available from <http://biodiversityinformatics.amnh.org/>, and released under the GNU FDL.)

The Biodiversity Informatics Facility at the American Museum of Natural History's (AMNH) Center for Biodiversity and Conservation (CBC) has developed the Event Visualization Tool (eVis), another software tool to add to the suite of conservation monitoring and decision support tools for guiding protected area and landscape planning. This plugin enables users to easily link geocoded (i.e., referenced with latitude and longitude or X and Y coordinates) photographs, and other supporting documents, to vector data in QGIS.

eVis is now automatically installed and enabled in new versions of QGIS, and as with all plugins, it can be disabled and enabled using the Plugin Manager (see *The Plugins Dialog*).

O módulo eVis é constituído por três módulos: a 'ferramenta de Ligação à Base de Dados', ferramenta do ID de Evento', e a 'Pesquisa de Eventos'. Tudo isto junto, permitirá ver fotografias geocodificadas e outros documentos que estão ligados aos elementos armazenados nos ficheiros vectoriais, base de dados ou folhas de cálculo.

20.6.1 Pesquisa de Eventos

The Event Browser module provides the functionality to display geocoded photographs that are linked to vector features displayed in the QGIS map window. Point data, for example, can be from a vector file that can be input using QGIS or it can be from the result of a database query. The vector feature must have attribute information associated with it to describe the location and name of the file containing the photograph and, optionally, the compass direction the camera was pointed when the image was acquired. Your vector layer must be loaded into QGIS before running the Event Browser.

Iniciar o módulo de Pesquisa de Eventos

To launch the Event Browser module, click on *Database* → *eVis* → *eVis Event Browser*. This will open the *Generic Event Browser* window.

The *Event Browser* window has three tabs displayed at the top of the window. The *Display* tab is used to view the photograph and its associated attribute data. The *Options* tab provides a number of settings that can be adjusted to control the behavior of the eVis plugin. Lastly, the *Configure External Applications* tab is used to maintain a table of file extensions and their associated application to allow eVis to display documents other than images.

Percebendo a janela de Exibição

To see the *Display* window, click on the *Display* tab in the *Event Browser* window. The *Display* window is used to view geocoded photographs and their associated attribute data.

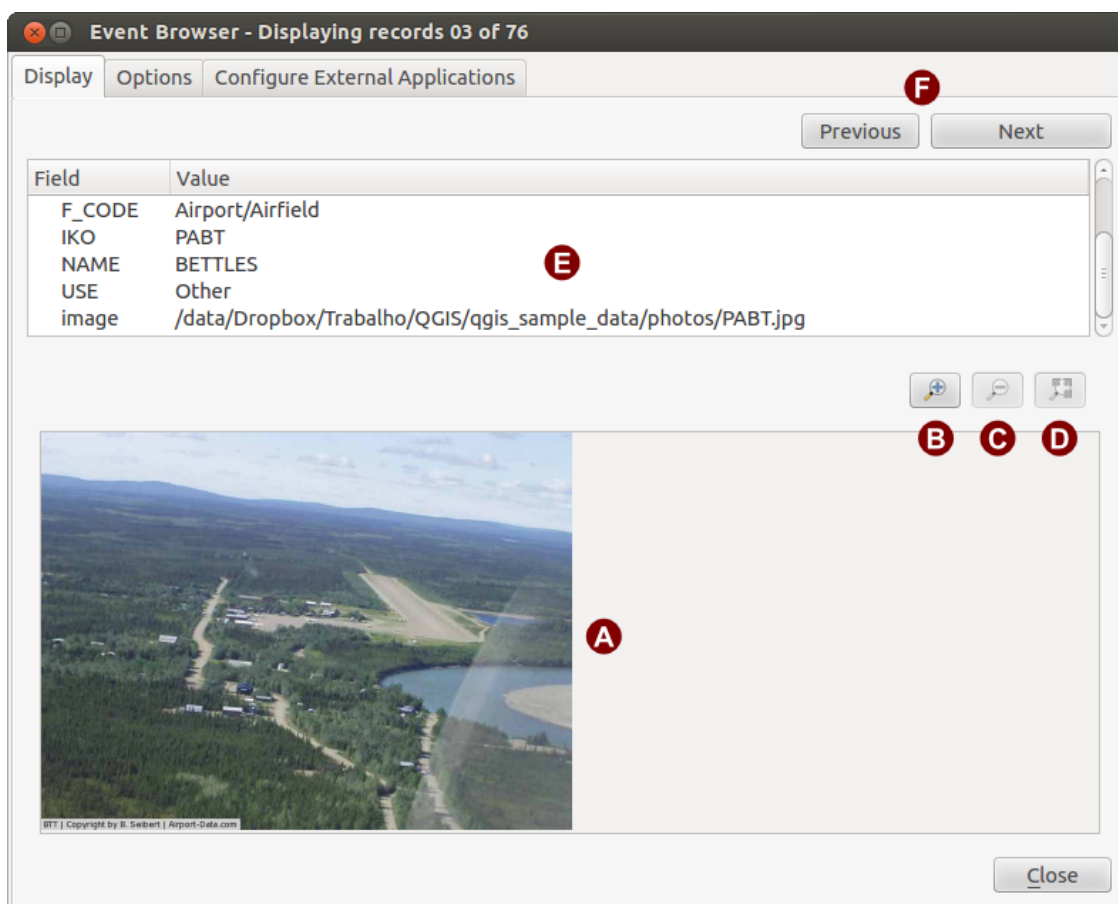


Figura 20.9: A janela de exibição eVis

1. ****Janela de exibição***: A janela onde a fotografia irá aparecer.
2. **Botão Aproximar**: Aproxima para ver com mais detalhe. Se a imagem inteira não for possível de exibir na janela de exibição, uma barra de deslocamento aparecerá no lado esquerdo e no fundo da janela para que possa mover à volta da imagem.
3. **Botão Afastar**: Afastar para visualizar mais área.
4. **Zoom to full extent** button: Displays the full extent of the photograph.
5. **Attribute information window**: All of the attribute information for the point associated with the photograph being viewed is displayed here. If the file type being referenced in the displayed record is not an image but is of a file type defined in the *Configure External Applications* tab, then when you double-click

on the value of the field containing the path to the file, the application to open the file will be launched to view or hear the contents of the file. If the file extension is recognized, the attribute data will be displayed in green.

6. **Botões de Navegação:** Use o botão Anterior e Seguinte para carregar o elemento anterior e seguinte quando mais que um elemento é seleccionado.

Entendendo a janela de Opções

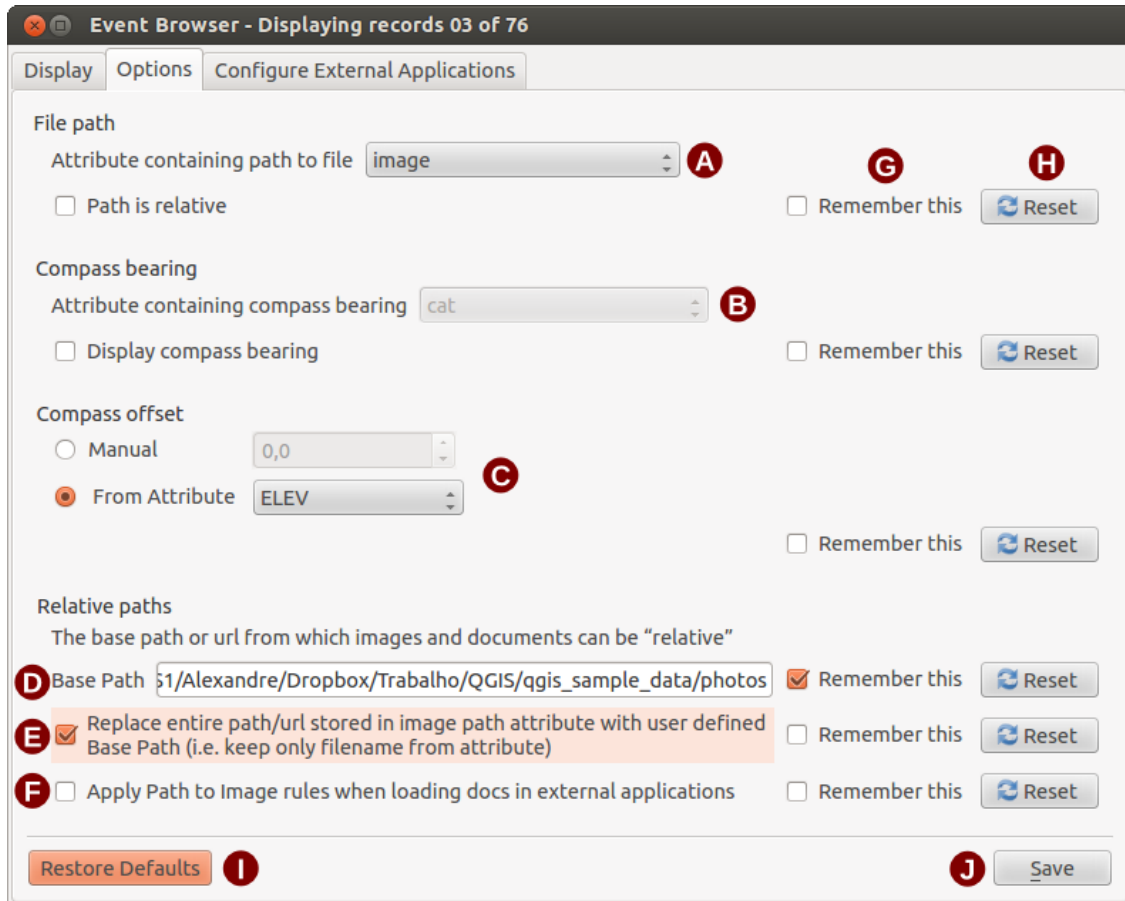


Figura 20.10: A janela de Opções eVis

1. **File path:** A drop-down list to specify the attribute field that contains the directory path or URL for the photographs or other documents being displayed. If the location is a relative path, then the checkbox must be clicked. The base path for a relative path can be entered in the *Base Path* text box below. Information about the different options for specifying the file location are noted in the section *Especificando a localização e nome da fotografia* below.
2. **Compass bearing:** A drop-down list to specify the attribute field that contains the compass bearing associated with the photograph being displayed. If compass bearing information is available, it is necessary to click the checkbox below the drop-down menu title.
3. **Compass offset:** Compass offsets can be used to compensate for declination (to adjust bearings collected using magnetic bearings to true north bearings). Click the *Manual* radio button to enter the offset in the text box or click the *From Attribute* radio button to select the attribute field containing the offsets. For both of these options, east declinations should be entered using positive values, and west declinations should use negative values.
4. **Directório do caminho base:** O caminho base para o caminho relativo definido na [Figure_eVis_2](#) (A) será anexado.

5. **Replace path:** If this checkbox is checked, only the file name from A will be appended to the base path.
6. **Apply rule to all documents:** If checked, the same path rules that are defined for photographs will be used for non-image documents such as movies, text documents, and sound files. If not checked, the path rules will only apply to photographs, and other documents will ignore the base path parameter.
7. **Remember settings:** If the checkbox is checked, the values for the associated parameters will be saved for the next session when the window is closed or when the **[Save]** button below is pressed.
8. **Restaurar valores:** Restaurar os valores nesta linha para a configuração padrão.
9. **Restore defaults:** This will reset all of the fields to their default settings. It has the same effect as clicking all of the **[Reset]** buttons.
10. **Guardar:** Isto guardará as configurações sem fechar o painel das *Opções*.

Entendendo a janela de Configuração de Aplicações Externas

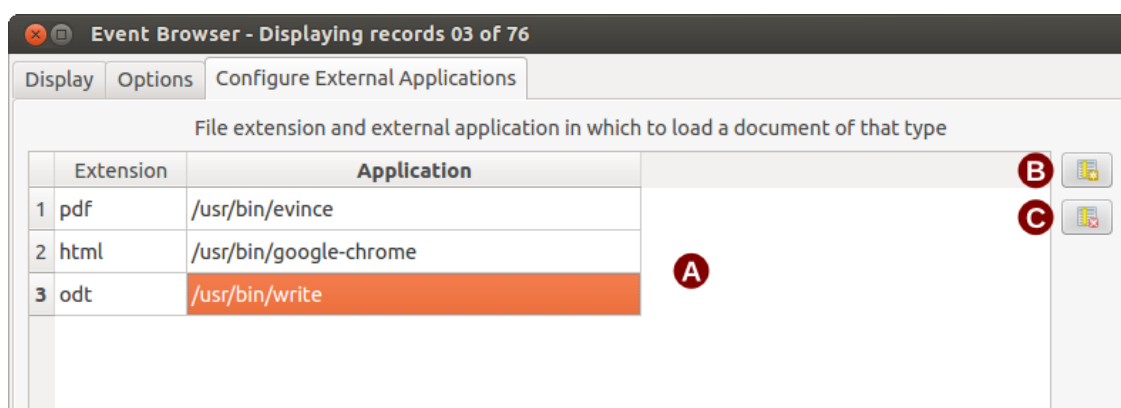


Figura 20.11: A janela de aplicações Externas *eVis*

1. **File reference table:** A table containing file types that can be opened using *eVis*. Each file type needs a file extension and the path to an application that can open that type of file. This provides the capability of opening a broad range of files such as movies, sound recordings, and text documents instead of only images.
2. **Adicionar novo tipo de ficheiro:** Adicionar um novo tipo de ficheiro com uma extensão única e o caminho para a aplicação que pode abrir o ficheiro.
3. **Apagar a linha actual:** Apagar o tipo de ficheiro destacado na tabela e definido por uma extensão de ficheiro e o caminho para aplicação associada.

20.6.2 Especificando a localização e nome da fotografia

The location and name of the photograph can be stored using an absolute or relative path, or a URL if the photograph is available on a web server. Examples of the different approaches are listed in [Table *evis_examples*](#).

X	Y	FILE	BEARING
780596	1784017	C:\Workshop\eVis_Data\groundphotos\DSC_0168.JPG	275
780596	1784017	/groundphotos/DSC_0169.JPG	80
780819	1784015	http://biodiversityinformatics.amnh.org/\ evis_testdata/DSC_0170.JPG	10
780596	1784017	pdf:http://www.testsite.com/attachments.php?\ attachment_id-12	76

20.6.3 Especificando a localização e nome de outros documentos suportados

Supporting documents such as text documents, videos, and sound clips can also be displayed or played by eVis. To do this, it is necessary to add an entry in the file reference table that can be accessed from the *Configure External Applications* window in the *Generic Event Browser* that matches the file extension to an application that can be used to open the file. It is also necessary to have the path or URL to the file in the attribute table for the vector layer. One additional rule that can be used for URLs that don't contain a file extension for the document you want to open is to specify the file extension before the URL. The format is — `file extension:URL`. The URL is preceded by the file extension and a colon; this is particularly useful for accessing documents from wikis and other web sites that use a database to manage the web pages (see Table [evis_examples](#)).

20.6.4 Usando a Pesquisa de Eventos

When the *Event Browser* window opens, a photograph will appear in the display window if the document referenced in the vector file attribute table is an image and if the file location information in the *Options* window is properly set. If a photograph is expected and it does not appear, it will be necessary to adjust the parameters in the *Options* window.

If a supporting document (or an image that does not have a file extension recognized by eVis) is referenced in the attribute table, the field containing the file path will be highlighted in green in the attribute information window if that file extension is defined in the file reference table located in the *Configure External Applications* window. To open the document, double-click on the green-highlighted line in the attribute information window. If a supporting document is referenced in the attribute information window and the file path is not highlighted in green, then it will be necessary to add an entry for the file's filename extension in the *Configure External Applications* window. If the file path is highlighted in green but does not open when double-clicked, it will be necessary to adjust the parameters in the *Options* window so the file can be located by eVis.

If no compass bearing is provided in the *Options* window, a red asterisk will be displayed on top of the vector feature that is associated with the photograph being displayed. If a compass bearing is provided, then an arrow will appear pointing in the direction indicated by the value in the compass bearing display field in the *Event Browser* window. The arrow will be centered over the point that is associated with the photograph or other document.

To close the *Event Browser* window, click on the **[Close]** button from the *Display* window.

20.6.5 Ferramenta de Eventos ID

The 'Event ID' module allows you to display a photograph by clicking on a feature displayed in the QGIS map window. The vector feature must have attribute information associated with it to describe the location and name of the file containing the photograph and, optionally, the compass direction the camera was pointed when the image was acquired. This layer must be loaded into QGIS before running the 'Event ID' tool.

Iniciar o módulo de Eventos ID

To launch the 'Event ID' module, either click on the  Event ID icon or click on *Database* → *eVis* → *Event ID Tool*. This will cause the cursor to change to an arrow with an 'i' on top of it signifying that the ID tool is active.


To view the photographs linked to vector features in the active vector layer displayed in the QGIS map window, move the Event ID cursor over the feature and then click the mouse. After clicking on the feature, the *Event Browser* window is opened and the photographs on or near the clicked locality are available for display in the browser. If more than one photograph is available, you can cycle through the different features using the **[Previous]** and **[Next]** buttons. The other controls are described in the [ref:evis_browser](#) section of this guide.

20.6.6 Ligação da Base de Dados


The 'Database Connection' module provides tools to connect to and query a database or other ODBC resource, such as a spreadsheet.

eVis can directly connect to the following types of databases: PostgreSQL, MySQL, and SQLite; it can also read from ODBC connections (e.g., MS Access). When reading from an ODBC database (such as an Excel spreadsheet), it is necessary to configure your ODBC driver for the operating system you are using.

Iniciar o módulo da Ligação à Base de Dados

To launch the 'Database Connection' module, either click on the appropriate icon  or click on *Database* → *eVis* → *Database Connection*. This will launch the *Database Connection* window. The window has three tabs: *Predefined Queries*, *Database Connection*, and *SQL Query*. The *Output Console* window at the bottom of the window displays the status of actions initiated by the different sections of this module.

Ligar à base de dados

Click on the *Database Connection* tab to open the database connection interface. Next, use the *Database Type*  combo box to select the type of database that you want to connect to. If a password or username is required, that information can be entered in the *Username* and *Password* textboxes.

Enter the database host in the *Database Host* textbox. This option is not available if you selected 'MS Access' as the database type. If the database resides on your desktop, you should enter "localhost".

Enter the name of the database in the *Database Name* textbox. If you selected 'ODBC' as the database type, you need to enter the data source name.

When all of the parameters are filled in, click on the [**Connect**] button. If the connection is successful, a message will be written in the *Output Console* window stating that the connection was established. If a connection was not established, you will need to check that the correct parameters were entered above.

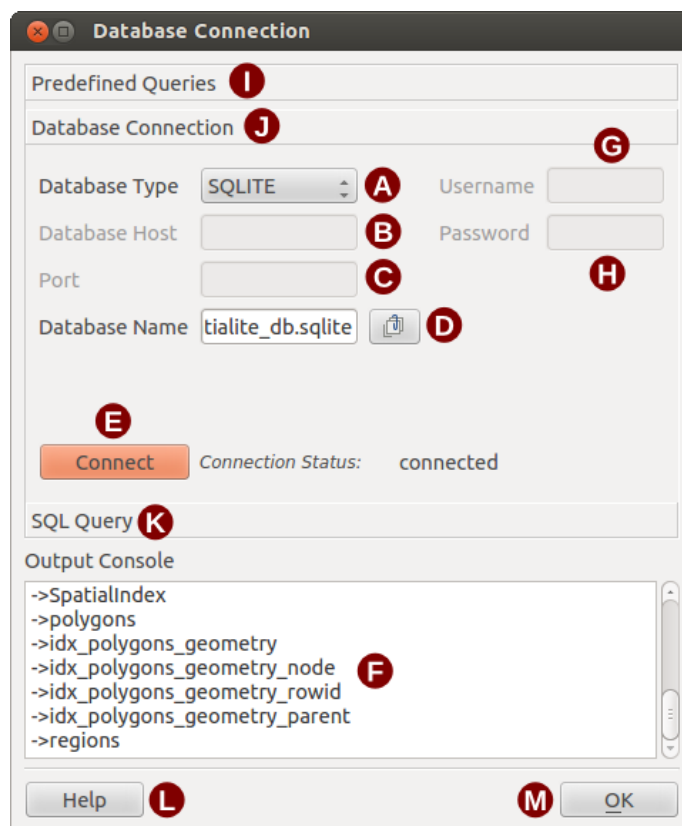


Figura 20.12: A janela de ligação da Base de Dados eVis

1. **Database Type:** A drop-down list to specify the type of database that will be used.

2. **Servidor da Base de Dados:** O nome do servidor da base de dados.
3. **Port:** The port number if a MySQL or PostgreSQL database type is selected.
4. **Database Name:** The name of the database.
5. **Connect:** A button to connect to the database using the parameters defined above.
6. **Output Console:** The console window where messages related to processing are displayed.
7. **Utilizador:** Utilizador para usar quando a base de dados é protegida por uma palavra-passe.
8. **Palavra-passe:** Palavra-passe para usar quando a base de dados é protegida por uma palavra-passe.
9. **Consultas Pré-Definidas:** Separador para abrir a janela “Consultas Pré-definidas”.
10. **Ligação da Base de Dados:** Separador para abrir a janela “Ligação à Base de Dados”.
11. **Consulta SQL:** Separador para abrir a janela da “Consulta SQL”.
12. **Help:** Displays the online help.
13. **OK:** Fecha a janela principal da “Ligação à Base de Dados”.



Correndo consultas SQL

SQL queries are used to extract information from a database or ODBC resource. In eVis, the output from these queries is a vector layer added to the QGIS map window. Click on the *SQL Query* tab to display the SQL query interface. SQL commands can be entered in this text window. A helpful tutorial on SQL commands is available at <http://www.w3schools.com/sql>. For example, to extract all of the data from a worksheet in an Excel file, `select * from [sheet1$]` where `sheet1` is the name of the worksheet.

Click on the **[Run Query]** button to execute the command. If the query is successful, a *Database File Selection* window will be displayed. If the query is not successful, an error message will appear in the *Output Console* window.

In the *Database File Selection* window, enter the name of the layer that will be created from the results of the query in the *Name of New Layer* textbox.

1. **Janela de texto da Consulta SQL:** Um ecrã para introduzir as consultas SQL.
2. **Correr Consulta:** Botão para executar a consulta introduzida na *Janela de Consulta SQL*.
3. **Janela da Consola:** A janela da consola onde as mensagens relacionadas com o processamento são exibidas.
4. **Help:** Displays the online help.
5. **OK:** Fecha a janela principal *Ligação à base de dados*.

Use the *X Coordinate*  and *Y Coordinate*  combo boxes to select the fields from the database that stores the X (or longitude) and Y (or latitude) coordinates. Clicking on the **[OK]** button causes the vector layer created from the SQL query to be displayed in the QGIS map window.

To save this vector file for future use, you can use the QGIS ‘Save as...’ command that is accessed by right-clicking on the layer name in the QGIS map legend and then selecting ‘Save as...’

Tip: Criando uma camada vectorial a partir de uma folha de cálculo Microsoft Excel

When creating a vector layer from a Microsoft Excel Worksheet, you might see that unwanted zeros (“0”) have been inserted in the attribute table rows beneath valid data. This can be caused by deleting the values for these cells in Excel using the *Backspace* key. To correct this problem, you need to open the Excel file (you’ll need to close QGIS if you are connected to the file, to allow you to edit the file) and then use *Edit* → *Delete* to remove the blank rows from the file. To avoid this problem, you can simply delete several rows in the Excel Worksheet using *Edit* → *Delete* before saving the file.

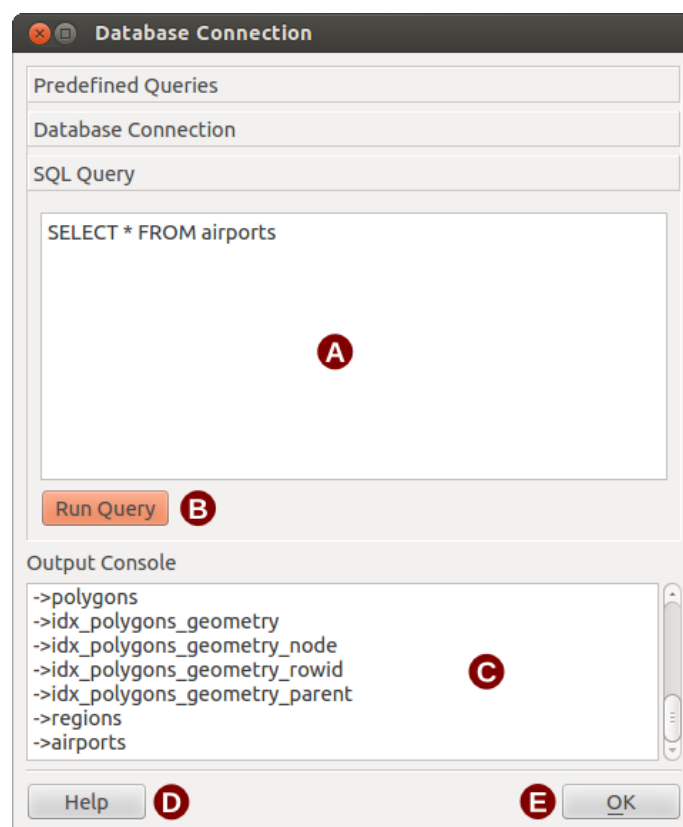




Figura 20.13: O separador de Consulta SQL eVis

Correndo consultas pré-definidas

With predefined queries, you can select previously written queries stored in XML format in a file. This is particularly helpful if you are not familiar with SQL commands. Click on the *Predefined Queries* tab to display the predefined query interface.

To load a set of predefined queries, click on the  Open File icon. This opens the *Open File* window, which is used to locate the file containing the SQL queries. When the queries are loaded, their titles as defined in the XML file will appear in the drop-down menu located just below the  Open File icon. The full description of the query is displayed in the text window under the drop-down menu.

Select the query you want to run from the drop-down menu and then click on the *SQL Query* tab to see that the query has been loaded into the query window. If it is the first time you are running a predefined query or are switching databases, you need to be sure to connect to the database.

Click on the **[Run Query]** button in the *SQL Query* tab to execute the command. If the query is successful, a *Database File Selection* window will be displayed. If the query is not successful, an error message will appear in the *Output Console* window.

1. **Open File:** Launches the “Open File” file browser to search for the XML file holding the predefined queries.
2. **Predefined Queries:** A drop-down list with all of the queries defined by the predefined queries XML file.
3. **Query description:** A short description of the query. This description is from the predefined queries XML file.
4. **Janela da Consola:** A janela da consola onde as mensagens relacionadas com o processamento são exibidas.
5. **Help:** Displays the online help.
6. **OK:** Fecha a janela principal da “Ligação à Base de Dados”.

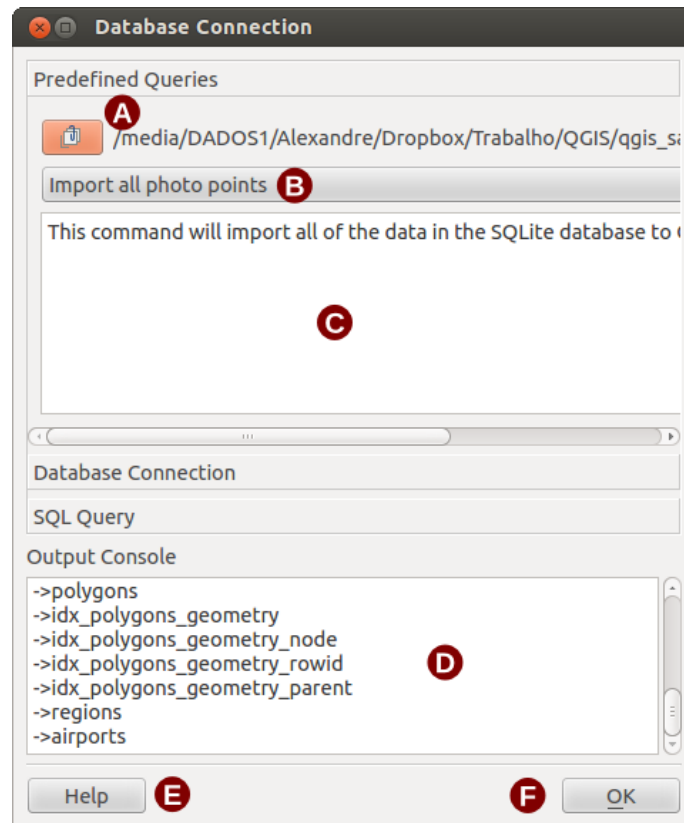


Figura 20.14: The eVis Predefined Queries tab

XML format for eVis predefined queries

The XML tags read by eVis

Etiqueta	Descrição
consulta	Define o início e o fim da instrução da consulta.
descrição curta	A short description of the query that appears in the eVis drop-down menu.
descrição detalhada	Uma descrição detalhada da consulta exibida na janela de texto da Consulta Pré-Definida.
database-type	The database type, defined in the Database Type drop-down menu in the Database Connection tab.
database-port	The port as defined in the Port text box in the Database Connection tab.
database-name	The database name as defined in the Database Name text box in the Database Connection tab.
databaseusername	The database username as defined in the Username text box in the Database Connection tab.
databasepassword	The database password as defined in the Password text box in the Database Connection tab.
sqlstatement	O comando SQL.
ligação automática	A flag (“true” or “false”) to specify if the above tags should be used to automatically connect to the database without running the database connection routine in the Database Connection tab.

A complete sample XML file with three queries is displayed below:

```
<?xml version="1.0"?>
<doc>
  <query>
```

```

<shortdescription>Import all photograph points</shortdescription>
<description>This command will import all of the data in the SQLite database to QGIS
  </description>
<databasetype>SQLITE</databasetype>
<databasehost />
<databaseport />
<databasename>C:\textbackslash Workshop\textbackslash
eVis\_Data\textbackslash PhotoPoints.db</databasename>
<databaseusername />
<databasepassword />
<sqlstatement>SELECT Attributes.*, Points.x, Points.y FROM Attributes LEFT JOIN
  Points ON Points.rec_id=Attributes.point_ID</sqlstatement>
<autoconnect>>false</autoconnect>
</query>
<query>
<shortdescription>Import photograph points "looking across Valley"</shortdescription>
<description>This command will import only points that have photographs "looking across
  a valley" to QGIS</description>
<databasetype>SQLITE</databasetype>
<databasehost />
<databaseport />
<databasename>C:\Workshop\eVis_Data\PhotoPoints.db</databasename>
<databaseusername />
<databasepassword />
<sqlstatement>SELECT Attributes.*, Points.x, Points.y FROM Attributes LEFT JOIN
  Points ON Points.rec_id=Attributes.point_ID where COMMENTS='Looking across
  valley'</sqlstatement>
<autoconnect>>false</autoconnect>
</query>
<query>
<shortdescription>Import photograph points that mention "limestone"</shortdescription>
<description>This command will import only points that have photographs that mention
  "limestone" to QGIS</description>
<databasetype>SQLITE</databasetype>
<databasehost />
<databaseport />
<databasename>C:\Workshop\eVis_Data\PhotoPoints.db</databasename>
<databaseusername />
<databasepassword />
<sqlstatement>SELECT Attributes.*, Points.x, Points.y FROM Attributes LEFT JOIN
  Points ON Points.rec_id=Attributes.point_ID where COMMENTS like '%limestone%'
</sqlstatement>
<autoconnect>>false</autoconnect>
</query>
</doc>

```

20.7 Módulo fTools

O objectivo do módulo python fTools é fornecer um recurso único para várias tarefas SIG comuns aos dados vectoriais, sem a necessidade de software adicional, bibliotecas, ou trabalho complexo. Fornece um conjunto em crescimento de formas de gestão de dados espaciais e funções de análise que são rápidas e funcionais.

O fTools está automaticamente instalado e activado nas novas versões do QGIS, juntamente com todos os módulos, e pode ser desactivado e activado através do Gestor de Módulos (Veja Secção *The Plugins Dialog*). Quando activado, o módulo fTools adiciona o menu *Vector* ao QGIS, fornecendo funções, desde Ferramentas de Análise e Investigação a Ferramentas de Geometria e Geoprocessamento, assim como várias ferramentas úteis de Gestão de Dados.

20.7.1 Ferramentas de Análise









Ícone	Ferramenta	Finalidade
	Matriz de Distância	Mede a distância entre duas camadas de pontos, e fornece resultados como a) matriz de distância quadrado, b) matriz de distância linear, ou c) Somatório de distâncias. Pode limitar distâncias usando k elementos próximos.
	Soma de comprimentos de linha	Calcula o somatório total de comprimentos de linha para cada polígono de uma camada vectorial do tipo polígono.
	Pontos no polígono	Conta o número de pontos que ocorrem em cada polígono numa camada vectorial do tipo polígono.
	Lista de valores únicos	Origina uma lista de valores únicos num campo de uma camada vectorial.
	Estatísticas básicas	Computa estatísticas básicas (média, desvio-padrão, N elementos, soma, CV) de um campo de entrada.
	Análise de Vizinhança	Calcula as estatísticas de vizinho mais próximo avaliando o nível de agrupamento numa camada de pontos vectorial
	Coordenada(s) média	Computa o centro médio normal ou com pesos de toda a camada vectorial, ou elementos múltiplos baseados num campo ID único.
	Intersecções de Linha	Localiza intersecções entre linhas, e resulta numa shapefile de pontos. É útil para localizar intersecções em estradas ou linhas de água, ignora intersecções de linha com um comprimento > 0.

Tabela Ftools 1: Ferramentas de Análise fTools

20.7.2 Ferramentas de investigação








Ícone	Ferramenta	Finalidade
	Seleção aleatória	Selecciona aleatoriamente n números de elementos, ou n percentagem de elementos.
	Seleção aleatória dentro de subconjuntos	Selecciona aleatoriamente elementos dentro de subconjuntos baseando-se num campo ID único.
	Pontos aleatórios	Gera pontos pseudo-aleatórios sobre uma camada vectorial.
	Pontos regulares	Gera uma grelha de pontos regulares sobre uma região específica ou enquadramento e exporta-os para uma shapefile de pontos.
	Grelha vectorial	Gera uma grelha de linhas ou polígonos baseada num espaçamento específico dado pelo utilizador.
	Seleccionar por localização	Selecione elementos baseados na sua localização relativa a outra camada para formar uma nova selecção, ou adicionar ou subtrair da selecção actual.
	Polígono a partir da extensão da camada	Cria uma camada do tipo polígono com um rectângulo único a partir da extensão de uma camada matricial ou vectorial.

Tabela Ftools 2: Ferramentas de investigação fTools

20.7.3 Ferramentas de geoprocessamento

Ícone	Ferramenta	Finalidade
	Formas convexas	Cria forma convexa(s) mínimas para uma camada, ou baseada num campo ID.
	Buffer(s)	Cria buffer(s) à volta dos elementos baseando-se na distância ou num campo de distância.
	Cruzar	Sobrepõe camadas e obtém como resultado áreas onde ambas as camadas intersectam.
	Unir	Sobrepõe camadas e obtém como resultado áreas com e sem intersecção.
	Diferença simétrica	Sobrepõe camadas e obtém como resultado áreas de diferenças de camadas que não intersectam.
	Cortar	Sobrepõe camadas e obtém como resultado áreas que intersectam a camada de corte.
	Diferença	Sobrepõe camadas e obtém como resultado áreas que não intersectam a camada de corte.
	Dissolver	Junta elementos baseados num campo de entrada. Todos os elementos com valores idênticos são combinados de forma a obter um elemento único.
	Eliminar fragmento de polígonos	Une elementos seleccionados com os polígonos vizinhos com a maior área ou o limite comum maior.

Tabela Ftools 3: Ferramentas de geoprocessamento fTools

20.7.4 Ferramentas de geometria

Ícone	Ferramenta	Finalidade
	Verificar a validade da geometria	Check polygons for intersections, closed holes, and fix node ordering.
	Exportar/Adicionar geometrias de colunas	Adiciona a informação da camada vectorial para camadas de pontos (XCOORD, YCOORD), linha (LENGTH), ou polígono (AREA, PERIMETER).
	Centróides de polígonos	Calcula os verdadeiros centróides para cada polígono numa camada do tipo polígono.
	Triangulação Delaunay	Calcula e produz uma saída baseada na triangulação de delaunay (como polígonos) de uma camada de pontos vectorial.
	Polígonos de Voronoi	Calcula polígonos de voronoi de uma camada de pontos vectoriais.
	Simplificar geometrias	Generaliza linhas ou polígonos com o algoritmo modificado Douglas-Peucker.
	Adensar geometrias	Adensa linhas ou polígonos através de adição de vértices.
	Multipartes para partes simples	Converte elementos multiparte para múltiplos elementos de +artes simples. Cria polígonos e linhas simples.
	Partes simples para multipartes	Junta elementos múltiplos para um único elemento multiparte baseado no campo ID único.
	Polígonos para linhas	Converte polígonos para linhas, polígonos multiparte para múltiplos partes simples de linhas.
	Linhas para polígonos	Converte linhas para polígonos, linhas multiparte para múltiplas partes simples de polígonos.
	Extrair Nós	Extrai nós de camadas de linhas e polígonos e resulta numa camada de pontos.

Tabela Ftools 4: Ferramentas de Geometria fTools

Note: A ferramenta *Simplificar geometrias* pode ser usada para remover nós duplicados em geometrias de linhas e polígonos, faça este truque definindo o parâmetro *Tolerância de generalização* para 0.

20.7.5 Ferramentas de Gerenciamento de dados

Ícone	Ferramenta	Finalidade
	Definir a projecção actual	Especifica o SRC para shapefiles que não tenham SRC definidos.
	Juntar atributos por localização	Junta atributos adicionais à camada vectorial baseando-se na relação espacial. Os atributos de uma camada vectorial são acrescentados à tabela de atributos de outra camada e é exportado como shapefile.
	Separar uma camada vectorial	Divide uma camada em múltiplas camadas separadas através de um campo de entrada.
	Unir shapefiles para num só Criar índice espacial	Une várias shapefiles dentro de uma pasta num novo shapefile tendo como base o tipo de camada (ponto, linha, área). Cria um índice espacial para os formatos OGR suportados.

Tabela Ftools 5: Ferramentas de Gerenciamento de Dados fTools

20.8 Módulo de Ferramentas GDAL

20.8.1 O que são as Ferramentas GDAL?

O módulo Ferramentas GDAL oferece uma colecção GUI de ferramentas da Biblioteca de Abstracção de Dados Geoespaciais, <http://gdal.osgeo.org>. Estes são ferramentas de gestão raster para consultar, reprojectar, torcer e unir um conjunto variado de formatos raster. Inclui também ferramentas para criar camadas de contornos (vector), ou relevos sombreados a partir de MDT matriciais, e para fazer um vrt (Virtual Raster Tile em formato XML) a partir de uma colecção de um ou mais ficheiros raster. Estas ferramentas estão disponíveis quando o módulo está instalado e activado.

Biblioteca GDAL

A biblioteca GDAL consiste num conjunto de programas da linha de comandos, cada um com uma lista cheia de opções. Os utilizadores que sabem usar a linha de comandos podem preferir a execução dos comandos no terminal, com acesso a todo o conjunto de opções. O módulo Ferramentas GDAL oferece uma interface fácil para as ferramentas, expondo apenas as opções mais populares.

20.8.2 Lista das ferramentas GDAL

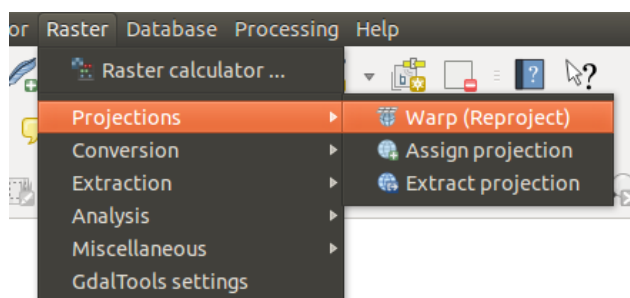










Figura 20.15: Lista do menu *Ferramentas GDAL*



Projecções

 <p><i>Torcer</i> (<i>Reprojectar</i>)</p>	<p>Este utilitário serve para os mosaicos de imagem, reprojecção e torções. O programa pode reprojectar para qualquer projecção suportada, e pode também ser aplicado a PC armazenados na imagem se a imagem tiver dados em “bruto” com controlo da informação. Para mais informação pode ler no sítio na internet do GDAL http://www.gdal.org/gdalwarp.html.</p>
 <p><i>Atribuir projecção</i></p>	<p>Esta ferramenta permite definir uma projecção para os rasters que estão georeferenciados mas não têm a informação da projecção. Com isto também ajuda na possibilidade de alterar a definição de uma projecção actual. Tanto o modo ficheiro único como o modo batch são suportados. Para mais informações, visite a página do utilitário no sítio na internet do GDAL, http://www.gdal.org/gdalwarp.html.</p>
 <p><i>Extrair projecção</i></p>	<p>Este utilitário ajuda-o a extrair a informação da projecção de um ficheiro de entrada. Se quiser extrair a projecção do directório pode usar o modo Batch. Irá criar os ficheiros <code>.prj</code> e <code>.wld</code>.</p>







Conversão

 <i>Digitalizar</i>	<p>Este programa torna geometrias vectoriais (pontos, linhas e polígonos) em banda(s) de uma imagem raster. Os vectores são lidos a partir de formatos OGR suportados. Repare que os dados do vector devem ser do mesmo sistema de coordenadas que os dados rasters; a projecção “on-the-fly” não é fornecida. Para mais informação veja http://www.gdal.org/gdal_rasterize.html.</p>
 <i>Vectorizar</i>	<p>Este utilitário cria polígonos vectoriais para todas as regiões de pixeis ligadas num matricial que partilha um valor comum de pixel. Cada polígono é criado com um atributo que indica o valor do pixel desse polígono. O utilitário irá criar um vector de saída se não existir, por defeito em formato ESRI shapefile. Veja também http://www.gdal.org/gdal_polygonize.html.</p>
 <i>Traduzir</i>	<p>Este utilitário pode ser usado para converter matriciais em diferente formatos, potencialmente executa algumas operações como subconfigurações, amostragem, rescalonamento de pixeis no processo. Para mais informações pode ler em http://www.gdal.org/gdal_translate.html</p>
 <i>RGB para PCT</i>	<p>Este utilitário irá computar uma tabela pseudo-cor otimizada para uma dada imagem RBG usando um algoritmo da mediana cortado num histograma RGB de resolução reduzida. De seguida converte a imagem numa imagem de pseudo-cor usando tabelas de cor. Esta conversão utiliza o pontilhado de Floyd-Steinberg (erro de difusão) para maximizar a qualidade visual da imagem de saída. Este utilitário é também descrito em http://www.gdal.org/rgb2pct.html</p>
 <i>PCT para RGB</i>	<p>Este utilitário irá converter uma banda de pseudo-cor num ficheiro de entrada para um ficheiro RGB de saída do formato desejado. Para mais informação veja http://www.gdal.org/pct2rgb.html</p>

Extracção

 <i>Con-torno</i>	<p>Este programa gera um ficheiro de contornos vectoriais a partir de um modelo digital do terreno (MDT) matricial. Pode encontrar mais informação em http://www.gdal.org/gdal_contour.html .</p>
 <i>Corta-dor</i>	<p>Este utilitário permite o corte (extracção de um subconjunto) usando um enquadramento seleccionado ou baseado no limite de um vector. Mais informação pode ser encontrado em http://www.gdal.org/gdal_translate.html.</p>

Análise

 <p><i>Crivo</i></p>	<p>Este utilitário remove polígonos rasters mais pequenos que o tamanho de limiar (em pixels) fornecido e substitui-os com o valor do pixel mais alto do vizinho mais próximo. O resultado pode ser escrito na banda raster existente, ou copiado para um novo ficheiro. Para mais informação veja see http://www.gdal.org/gdal_sieve.html .</p>
 <p><i>Próximo ao Preto</i></p>	<p>Este utilitário irá digitalizar a imagem e tentar definir todos os pixels que existem perto do preto (ou perto do branco) à volta do limite para exactamente preto (ou branco). Isto é usado usualmente para “corrigir” perdas em fotos áreas comprimidas para que esses pixels de cor possam ser tratadas como transparentes nas operações de mosaico. Veja também http://www.gdal.org/nearblack.html.</p>
 <p><i>Preencher sem dados</i></p>	<p>Este utilitário preenche as regiões raster seleccionadas (usualmente conhecidas com áreas sem valor) por interpolação de de pixels validados à volta das bordas da área. Pode encontrar mais informação em http://www.gdal.org/gdal_fillnodata.html .</p>
 <p><i>Proximidade</i></p>	<p>Este utilitário gera um mapa de proximidade raster indicando a distância desde o centro de cada pixel para o centro do pixel mais próximo identificado como pixel alvo. Os pixels alvo são os que estão presentes no raster inicial em que cada valor do pixel é definido como valores pixels alvo. Para mais informação veja http://www.gdal.org/gdal_proximity.html .</p>
 <p><i>Grelha (Interpolação)</i></p>	<p>Este utilitário cria um grelha regular (raster) a partir da leitura de dados dispersos de um fonte de dados OGR. Os dados de entrada serão interpolados para preencher os nós da grelha com valores, pode escolher vários métodos de interpolação. O utilitário também é descrito no sítio na internet do GDAL http://www.gdal.org/gdal_grid.html .</p>
 <p><i>MDE (Modelos de Elevação)</i></p>	<p>Ferramentas para analisar e visualizar MDT. Podem ser criados, relevos sombreados, declives, exposições, relevo colorido, índice de rugosidade do terreno, índice de posição topográfica e mapas de irregularidades a partir de matricias GDAL suportados. Para mais informação poderá ler em http://www.gdal.org/gdaldem.html</p>

Diversos

 <p><i>Construir Matricial Virtual (Catálogo)</i></p>	<p>Este programa constrói um VRT (Conjunto de Dados Virtual) que é um mosaico da lista dos conjuntos de dados do GDAL. Veja também http://www.gdal.org/gdalbuildvrt.html .</p>
 <p><i>Juntar</i></p>	<p>Este utilitário irá criar mosaicos de imagem a partir de um conjunto. Todas as imagens devem ter o mesmo sistema de coordenadas e ter o mesmo número de bandas correspondentes, mas podem ser sobrepostas, e em diferentes resoluções. Nas áreas de sobreposição, a última imagem será copiada sobre as mais recentes. O utilitário é também descrito em http://www.gdal.org/gdal_merge.html .</p>
 <p><i>Informação</i></p>	<p>Este utilitário cria uma lista com várias informações sobre o conjunto de dados raster GDAL suportado. Pode encontrar mais informação em http://www.gdal.org/gdalinfo.html .</p>
 <p><i>Construir Reduções</i></p>	<p>O utilitário gdaladdo pode ser usado para construir ou reconstruir imagem de visualização para a maioria dos formatos suportados com um dos algoritmos de redução de escala. Para mais informação veja http://www.gdal.org/gdaladdo.html .</p>
 <p><i>Índice de Quadriculas</i></p>	<p>Este utilitário constroi uma shapefile com o registo de cada ficheiro raster de entrada, um atributo contendo um nome do ficheiro, e a geometria do polígono do limite do raster. Veja também http://www.gdal.org/gdaltindex.html .</p>

configurações das Ferramentas GDAL

Use este diálogo para embeber variáveis GDAL.

20.9 Módulo Georeferenciador

O módulo Georeferenciador é uma ferramenta para gerar world files para rasters. Permite a possibilidade de referenciar rasters para sistemas de coordenadas projectadas ou geográficas através da criação de um novo GeoTiff ou adicionando um world file à imagem existente. A georeferenciação do raster passa por uma abordagem simples de localização de pontos no raster para que possa com precisão determinar as suas coordenadas.

Características

Ícone	Finalidade	Ícone	Finalidade
	Abrir matricial		Iniciar Georeferenciamento
	Gerar Script GDAL		Carregar pontos GCP
	Salvar pontos GCP como		Configurações da transformação
	Adicionar Ponto		Apagar Ponto
	Mover ponto GCP		Movimentar
	Aproximar		Afastar
	Aproximar à Camada		Última Visualização
	Próxima Visualização		Ligar Georeferenciador ao QGIS
	Ligar QGIS ao Georeferenciador		Esticar totalmente o histograma
	Esticar localmente o histograma		

Tabela 1 Georeferenciador: Ferramentas do Georeferenciador

20.9.1 Procedimento comum

Dois procedimentos alternativos podem ser usados, como coordenadas X e Y (DMS (dd mm ss.ss), DD (dd.dd)) ou coordenadas projectadas (mmmm.mm) que correspondem ao ponto seleccionado na imagem:

- O raster por si próprio por vezes fornece cruces com coordenadas “escritas” na imagem. Neste caso, pode introduzir as coordenadas manualmente.
- Usando camadas georeferenciadas. Estes podem conter informação vectorial ou raster que contenham os mesmos objectos/elementos que esteja na imagem que queira georeferenciar e projectar e a projecção que quer que a imagem tenha. Neste caso, pode introduzir as coordenadas clicando no conjunto de dados referenciado carregado no enquadramento do mapa do QGIS.

O procedimento normal do georeferenciamento de uma imagem envolve múltiplos pontos seleccionados no matricial, especificando as suas coordenadas, e escolhendo o tipo de transformação mais relevante. O módulo, baseado nos parâmetros de entrada e da informação irá computar os parâmetros do world file. Quanto mais coordenadas fornecer, melhor o resultado que irá obter.

O primeiro passo é iniciar o QGIS, carregar o Módulo Georeferenciador (veja *The Plugins Dialog*) e clique no ícone **Imselecionl:Raster** → *Georeferenciador* que aparece no menu da barra de ferramentas do QGIS. A janela do módulo do Georeferenciador aparece como demonstra a [Figura_de_georeferenciador_1](#).

Para este exemplo, estamos a usar uma carta militar do Sul de Dakota do SDGS. Pode ser visualizada mais tarde juntamente com a informação proveniente da localização do GRASS `spearfish60`. Pode transferir a carta militar aqui: http://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz.

Introduzindo pontos de controlo (GCPs)

1. Para iniciar o georeferenciamento de um raster sem georeferenciação, necessitamos de carregá-lo usando o botão . O raster será mostrado na janela principal da área de trabalho. Uma vez carregado o raster, podemos começar a introduzir pontos de referência.
2. O botão Adicionar Ponto é usado para adicionar pontos na área de trabalho principal e introduzir as suas coordenadas (veja [Figura figura_do_georeferenciador_2](#)). Para este procedimento tem três opções:
 - Clique num ponto na imagem matricial e introduza as coordenadas X e Y manualmente.

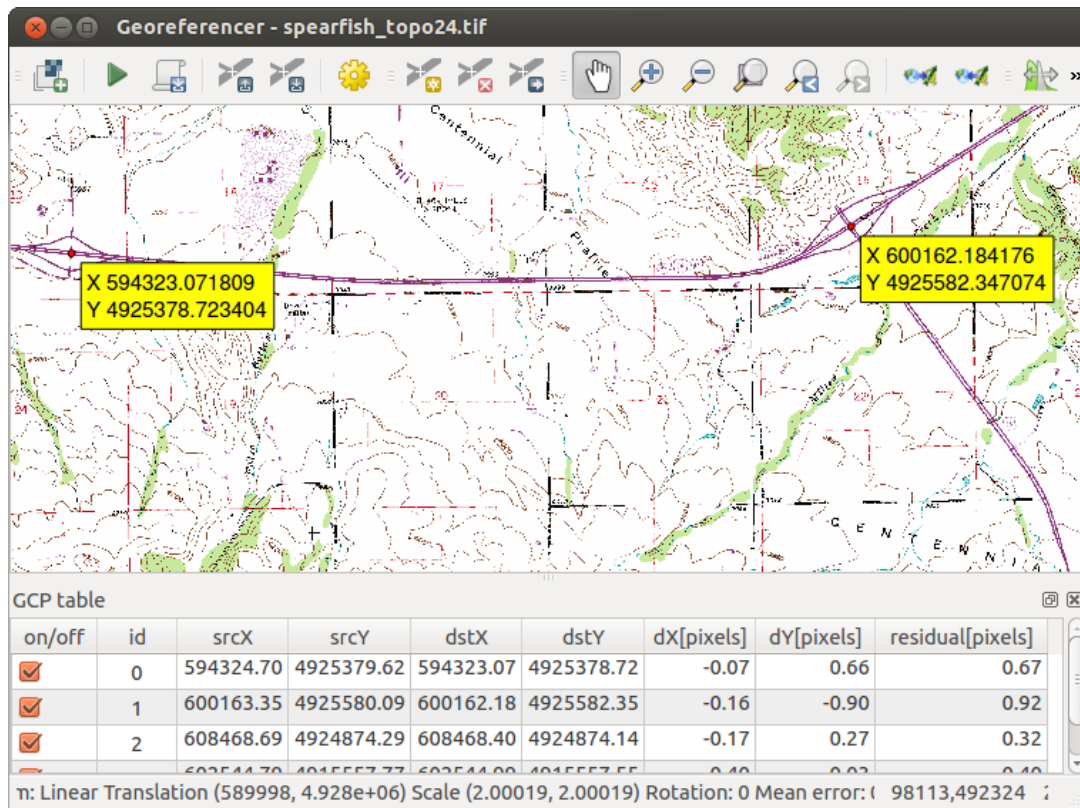




Figura 20.16: Janela do Módulo Georeferenciador

- Clique no ponto da imagem matricial e escolha o botão  proveniente do enquadramento do map para adicionar as coordenadas X e Y com a ajuda do mapa georeferenciado que já se encontra carregado no enquadramento do mapa do QGIS.
 - Com o botão , pode mover os GCP em ambas as janelas, se estiverem no sítio errado.
3. Continue a introduzir pontos. Deve ter no mínimo quatro pontos, e quanto mais coordenadas fornecer, melhor o resultado será. Existe ferramentas adicionais na janela do módulo para aproximar e movimentar a área de trabalho de forma a localizar o conjunto relevante de pontos GCP.

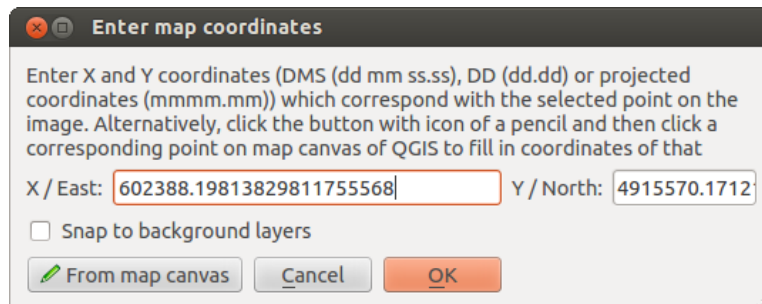




Figura 20.17: Adicionar pontos à imagem matricial

Os pontos que adiciona ao mapa serão guardados num ficheiro de texto deparado ([filename].points) normalmente junto com a imagem matricial. Isto permite que possamos reabrir o módulo do Georeferenciador mais tarde e adicionar novos pontos ou apagar existentes para otimizar o resultado. O ficheiro de pontos contem valores na forma de: mapX, mapY, pixelX, pixelY. Pode usar o  Carregar pontos GCP e o  Guardar pontos GCP como para gerir os ficheiros.

Definindo as configurações de transformação

Depois de adicionar os GCP à imagem matricial, necessita de definir as configurações de transformação para o processo de georeferenciamento.

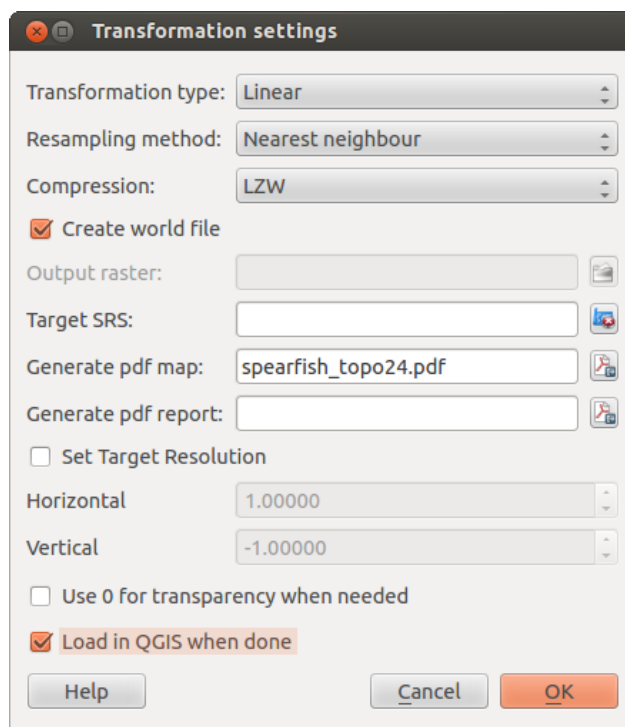


Figura 20.18: Definindo as configurações de transformação do georeferenciador 🐧

Algoritmos de Transformação disponíveis

Dependendo da quantidade de pontos de controlo que registre, pode querer usar diferentes algoritmos de transformação. A escolha do algoritmo de transformação também depende do tipo e qualidade dos dados de entrada e a quantidade de distorção geométrica que está disposto a introduzir no resultado final.

Actualmente, os seguintes *Tipos de transformação* estão disponíveis:

- O algoritmo **Linear** é usado para criar o world-file, e é diferente dos outros algoritmos, e não transforma verdadeiramente o raster. Este algoritmo provavelmente não será suficiente se estiver a lidar com material digitalizado.
- A transformação de **Helmert** executa um simples escalonamento e transformações de rotação.
- O algoritmo **Polinomial** 1-3 estão entre os algoritmos mais utilizados introduzidos para coincidir com a origem e o destino dos pontos de controlo. O algoritmo polinomial mais amplamente utilizado é a transformação polinomial de segunda ordem, o que permite alguma curvatura. Primeira transformação polinomial ordem (afim) preserva a colinearidade e permite apenas o escalonamento, translação e rotação.
- O algoritmo **Suavizador de Placas Finas (TPS)** é o método mais moderno de georeferenciação, que permite deformações locais nos dados. Este algoritmo é útil quando originais de baixa qualidade estão a ser georeferenciados.
- A transformação **Projectiva** é uma rotação linear e de translação de coordenadas.

Definir o método de Reamostragem

O tipo de amostragem que escolhe irá depender dos seus dados de entrada e do objectivo do exercício. Se não quiser mudar as estatísticas da imagem, deverá querer escolher ‘Vizinho mais próximo’, visto que a ‘Amostragem cúbica’ irá fornecer um resultado mais suavizado.

É possível escolher entre cinco diferentes métodos de amostragem.

1. Vizinho mais próximo
2. Linear
3. Cúbica
4. Cúbico Suavizado
5. Lanczos

Definindo as configurações de transformação

Existe várias opções que necessitam ser definidos para a saída do matricial georeferenciado.

- A caixa de verificação *Criar world file* ‘está apenas disponível se decidir usar o tipo de transformação linear, pois significa que a sua imagem raster actualmente não será transformada. Nesse caso, o campo :guilabel: ‘Raster de Saída não está activo, porque apenas será criado um novo world-file.
- Para outro tipo de transformação necessita de definir um *Raster de Saída*. Por defeito um novo ficheiro ([filename]_modified) será criado na mesma pasta junto da imagem raster original.
- Como próximo passo, necessita de definir *SRC de Destino* (Sistema de Referência Espacial) para o matricial georeferenciado (veja *Trabalhando com Projecções*).
- Se quiser, pode **gerar um mapa em pdf** e também um **relatório em pdf**. O relatório inclui informação acerca dos parametros de transformação usados. Uma imagem com os resíduos e a lista com todos os pontos de controlo e os seus EMQ.
- Além disso, pode activar a caixa de verificação :guilabel: ‘Definir a Resolução do Alvo’ e definir a resolução do pixel do raster de saída. A resolução horizontal e vertical por defeito é de 1.
- :guilabel: ‘Usar 0 para a transparência quando necessário’ pode ser activado, se os pixeis com o valor 0 será visualizado como transparente. No nosso exemplo, todas as áreas brancas da carta militar seriam transparentes.
- Finalmente, :guilabel: ‘Carregar no QGIS quando concluído’ carrega o raster de saída automaticamente para o enquadramento do mapa do QGIS depois de ser feita a transformação.


Mostrar e adaptar as propriedades do matricial

Clicando na janela das *Propriedades do matricial* no menu *Configurações* abre as propriedades do matricial da camada que quer georeferenciar.

Configurar o georeferenciador


- Pode definir se deseja mostrar as coordenadas GCP e/ou IDs.
- Assim como as unidades de resíduos pixel e as unidades de mapa podem ser escolhidos.
- Para o relatório PDF a margem esquerda e direita pode ser definida e pode também configurar o tamanho da folha para o mapa PDF.
- Finalmente pode activar o *Mostrar a janela do georeferenciador ancorada*.

Correndo a transformação

Depois de recolher todos os GCPS e as configurações de transformação definidas, pressione o botão  :sup: ‘Iniciar georeferenciamento’ para criar o novo raster georeferenciado.

20.10 Módulo de Interpolação

The Interpolation plugin can be used to generate a TIN or IDW interpolation of a point vector layer. It is very simple to handle and provides an intuitive graphical user interface for creating interpolated raster layers (see [Figure_interpolation_1](#)). The plugin requires the following parameters to be specified before running:

- **Input Vector layers:** Specify the input point vector layer(s) from a list of loaded point layers. If several layers are specified, then data from all layers is used for interpolation. Note: It is possible to insert lines or polygons as constraints for the triangulation, by specifying either “points”, “structure lines” or “break lines” in the *Type*  combo box.
- **Interpolation attribute:** Select the attribute column to be used for interpolation or enable the *Use Z-Coordinate* checkbox to use the layer’s stored Z values.
- **Interpolation Method:** Select the interpolation method. This can be either ‘Triangulated Irregular Network (TIN)’ or ‘Inverse Distance Weighted (IDW)’.
- **Number of columns/rows:** Specify the number of rows and columns for the output raster file.
- **Ficheiro de Saída:** Especifique o nome para o ficheiro raster de saída.
- *Adicionar resultado ao projecto* para carregar o resultado para o enquadramento do mapa.

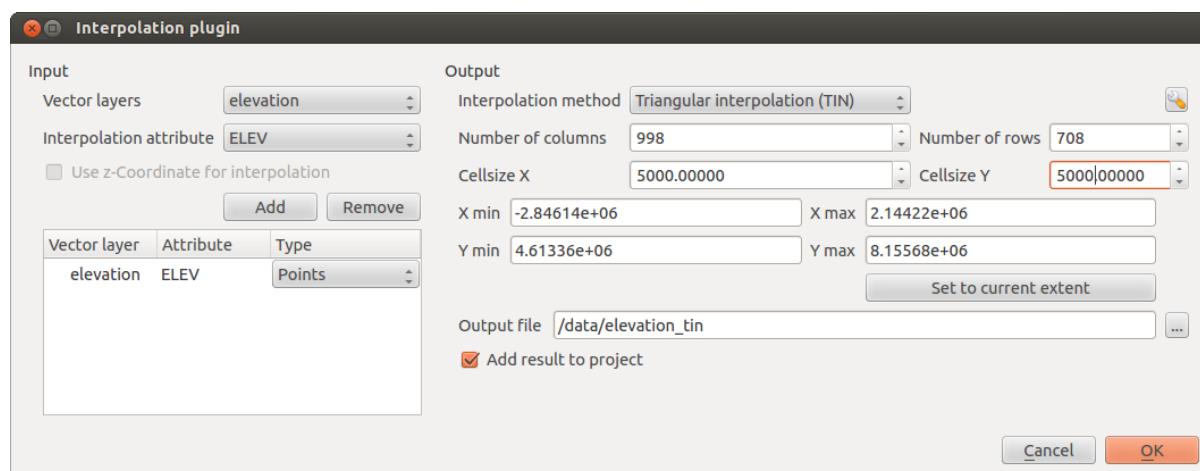





Figura 20.19: Módulo de Interpolação 


20.10.1 Usando o módulo

1. Inicie o QGIS e carregue uma camada de pontos vectorial (ex.: `elevp.csv`).
2. Load the Interpolation plugin in the Plugin Manager (see [The Plugins Dialog](#)) and click on the *Raster* → *Interpolation* →  *Interpolation*, which appears in the QGIS menu bar. The Interpolation plugin dialog appears as shown in [Figure_interpolation_1](#).
3. Seleccione uma camada de entrada (ex.: `elevp` ) e a coluna (ex.: `ELEV`) para interpolação.



4. Select an interpolation method (e.g., 'Triangulated Irregular Network (TIN)'), and specify a cell size of 5000 as well as the raster output filename (e.g., `elevation_tin`).
5. Clique [OK].

20.11 Módulo Edição Offiline


For data collection, it is a common situation to work with a laptop or a cell phone offline in the field. Upon returning to the network, the changes need to be synchronized with the master datasource (e.g., a PostGIS database). If several persons are working simultaneously on the same datasets, it is difficult to merge the edits by hand, even if people don't change the same features.

O módulo  Edição Offline automatiza a sincronização pela cópia do conteúdo da fonte de dados (usualmente PostGIS ou WFS-T) para uma base de dados SpatialLite e armazena as edições the offline em tabelas dedicadas. Depois de estarem ligadas outra vez à rede, é possível aplicar edições offline no conjunto de dados principal.

20.11.1 Usando o módulo

- Open some vector layers (e.g., from a PostGIS or WFS-T datasource).
- Save it as a project.
- Go to *Database* → *Offline Editing* →  *Convert to offline project* and select the layers to save. The content of the layers is saved to SpatialLite tables.
- Edite as camadas offline.
- After being connected again, upload the changes using *Database* → *Offline Editing* →  *Synchronize*.


20.12 Módulo Oracle Spatial GeoRaster

Nas bases de dados Oracle, os dados raster podem ser armazenados em objectos SDO_GEOCASTER disponíveis com a extensão do Oracle Spatial. No QGIS, o  :sup: 'Módulo Oracle GeoRaster' é suportado pelo GDAL, e depende de um produto Oracle de base de dados instalado e a trabalhar na sua máquina. Enquanto software proprietário a Oracle fornece o seu software gratuitamente para finalidades de desenvolvimento e testes. Aqui está um exemplo de como as imagens raster são carregadas para o GeoRaster:

```
$ gdal_translate -of georaster input_file.tif geor:scott/tiger@orcl
```

Isto irá carregar o raster para a tabela padrão GDAL_IMPORT table, como coluna designada de RASTER.

20.12.1 Gerindo ligações

Primeiramente, o Módulo Oracle GeoRaster deve estar activado usando o Gestor de Módulos (veja Secção *load_core_plugin*). A primeira vez que carrega um GeoRaster no QGIS, deve criar a ligação que à base de dados Oracle que contém os dados. Para fazer isso, comece por clicar no botão da barra de ferramentas  Seleccionar GeoRaster, irá abrir a janela de diálogo *Seleccionar Oracle Spatial GeoRaster*. Clique em [Novo] para abrir a janela de diálogo, e especifique os parâmetros de ligação (Veja [Figure_oracle_raster_1](#)):

- **Nome:** Introduza um nome para a ligação de base de dados.
- **Instância da Base de Dados:** Introduza o nome das bases de dados que se irá ligar.

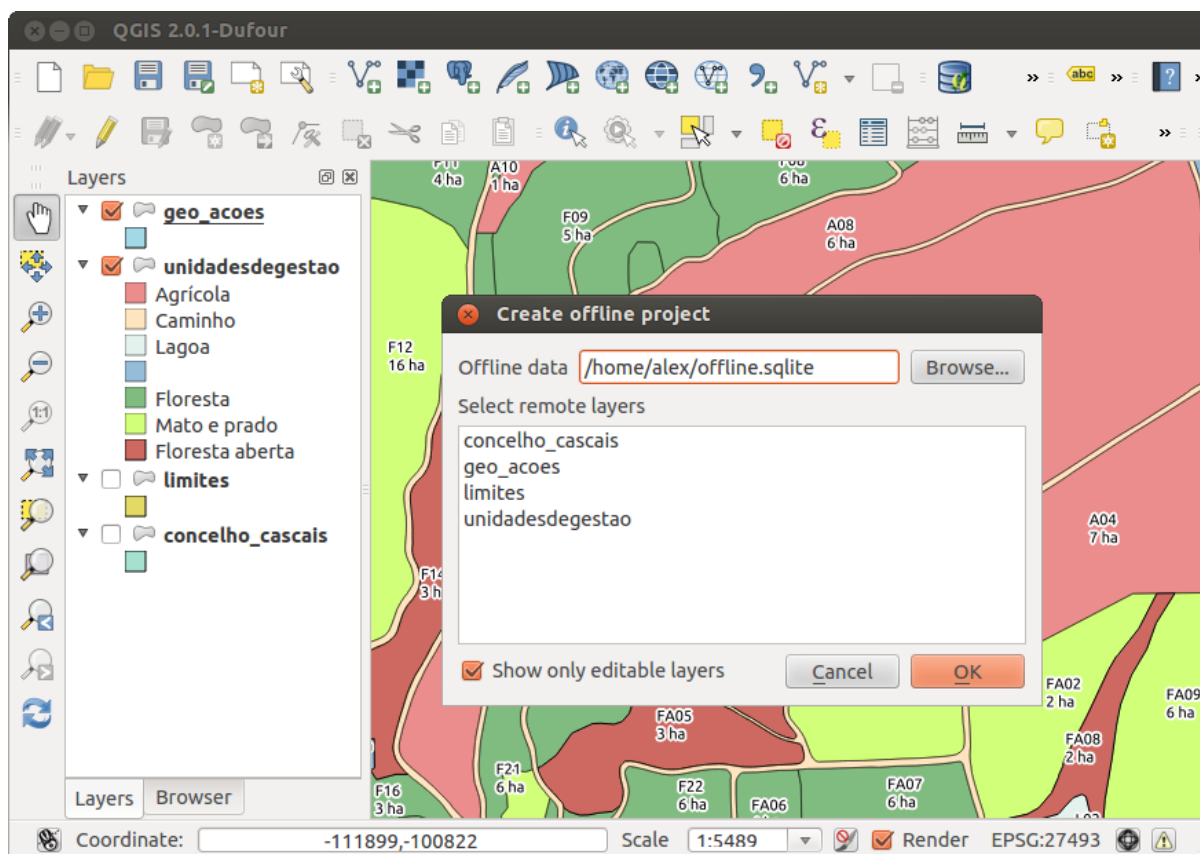


Figura 20.20: Criação de um projecto offline de camadas PostGIS ou WFS

- **Nome de Utilizador:** Especifique o seu nome de utilizador que irá usar para aceder à base de dados.
- **Palavra-chave:** A palavra-chave associada ao seu nome de utilizador que é requerido ao acesso à base de dados.

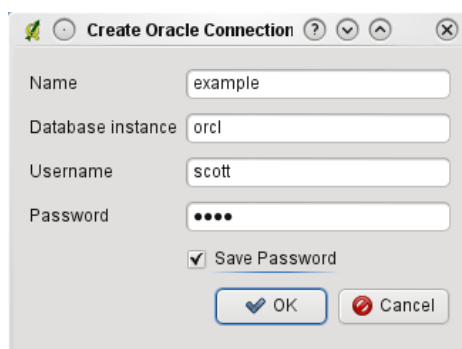


Figura 20.21: Janela de Criação de Ligação Oracle

Agora, volte à janela principal *Oracle Spatial GeoRaster* (veja [Figure_oracle_raster_2](#)), use a lista drop-down para escolher uma ligação, e use o botão **[Ligar]** para estabelecer a ligação. Pode também **[Editar]** a ligação, abrindo a janela anterior e efectuar alterações na informação da ligação, ou usar o botão **[Apagar]** para remover a ligação da lista drop-down.

20.12.2 Seleccionando um GeoRaster

Uma vez a ligação estar estabelecida, a janela exibirá os nomes dos sub-conjunto de dados de todas as tabelas que contém as colunas GeoRaster nessa base de dados no formato de um nome do subconjunto de dados GDAL.

Clique num dos subconjuntos de dados listados e de seguida clique em **[Seleccionar]** para escolher o nome da tabela. Agora, outra lista de subconjunto de dados irá mostrar os nomes das colunas GeoRaster nessa tabela. Geralmente costuma ser uma lista pequena, uma vez que a maioria dos utilizadores não irá ter mais de um ou duas colunas GeoRaster na mesma tabela.

Clique numa dos subconjuntos de dados listados e depois clique em **[Seleccionar]** para escolher uma das combinações tabela/coluna. A janela irá agora mostrar todas as linhas que contêm os objectos GeoRaster. Repare que essa lista de subconjuntos de dados irá agora aparecer em pares a Tabela de dados e Id's do Raster.

A qualquer altura a entrada de Selecção pode ser editada de forma a ir directamente a um GeoRaster conhecido ou voltar para o princípio e seleccionar outro nome da tabela.

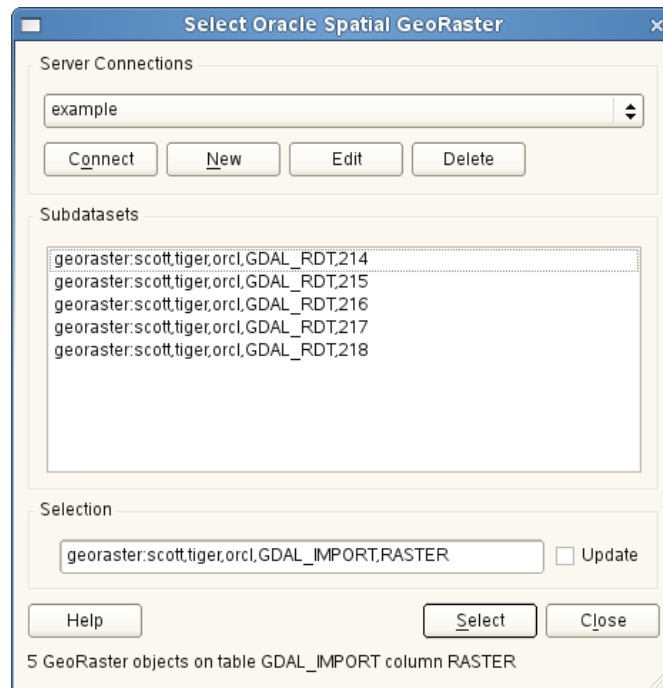


Figura 20.22: Janela de Selecção Oracle GeoRaster

A entrada de Selecção de dados pode ser também usada para introduzir cláusulas WHERE no fim da string de identificação (ex.: `geor:scott/tiger@orcl,gdal_import,raster,geoid=`). Veja http://www.gdal.org/frmt_georaster.html para mais informação.

20.12.3 Exibindo o GeoRaster

Finalmente, ao seleccionar um GeoRaster a partir da lista da tabela de dados matriciais e id's, a imagem matricial será carregada no QGIS.

A janela *Seleccionar Oracle Spatial GeoRaster* pode ser agora fechada e na próxima vez que abrir irá manter a mesma ligação, e irá mostrar previamente a mesma lista do subconjunto de dados tornando-se assim mais fácil abrir outra imagem do mesmo contexto.

Note: GeoRasters que contêm pirâmides serão exibidas muito mais rapidamente mas as pirâmides necessitam de ser gerados fora do QGIS usando Oracle PL/SQL ou gdaladdo.

O seguinte é um exemplo usando gdaladdo:

```
gdaladdo georaster:scott/tiger@orcl,georaster\_table,georaster,geoid=6 -r
nearest 2 4 6 8 16 32
```

Este é um exemplo usando PL/SQL:

```
$ sqlplus scott/tiger
SQL> DECLARE
  gr sdo_georaster;
BEGIN
  SELECT image INTO gr FROM cities WHERE id = 1 FOR UPDATE;
  sdo_geor.generatePyramid(gr, 'rLevel=5, resampling=NN');
  UPDATE cities SET image = gr WHERE id = 1;
  COMMIT;
END;
```

20.13 Módulo de Análise do Terreno Matricial



O módulo de Análises de Terreno Raster pode ser usado para calcular o declive, exposição, ensobrimento, índice de rugosidade e relevo para modelos digitais do terreno (MDT). É muito simples de lidar e fornece uma interface gráfica intuitiva para o utilizador para criar novas camadas raster (veja [Figure_raster_terrain_1](#)).

Descrição das análises:

- **Declive:** Calcula o ângulo de declive para cada célula em graus (baseado na primeira derivada de estimação).
- **Exposições:** Exposição (começando com o 0 para direcção norte, em graus contra-relógio).
- **Hillshade:** Creates a shaded map using light and shadow to provide a more three-dimensional appearance for a shaded relief map.
- **Índice de Rugosidade:** Medição quantitativa da heterogeneidade do terreno como é descrito por Riley et al. (1999). É calculado para cada localização, pelo resumo da alteração da elevação dentro de uma grelha de 3x3 pixels.
- **Relief:** Creates a shaded relief map from digital elevation data. Implemented is a method to choose the elevation colors by analysing the frequency distribution.

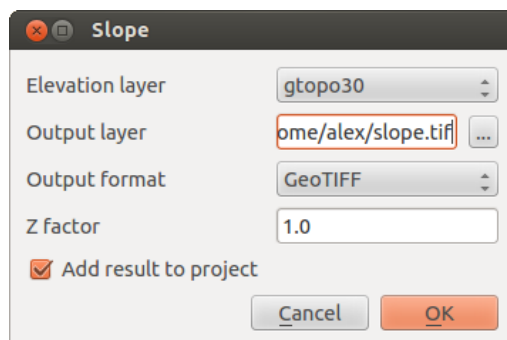


Figura 20.23: Módulo Modelação Matricial do Terreno (cálculo do declive)

20.13.1 Usando o módulo


1. Inicie o QGIS e carregue a camada matricial `gtopo30` da localização amostra do GRASS.
2. Carregue o módulo de Análise do Terreno Raster no Gestor de Módulos (veja *The Plugins Dialog*).
3. Selecione um método de análise do menu (ex.: *Raster* → *Análises do Terreno* → *Declive*). A janela *Declive* aparece como é mostrado na [Figure_raster_terrain_1](#).
4. Especifique um caminho de saída do ficheiro, e o tipo de ficheiro.
5. Clique [OK].

20.14 Módulo de Mapa de Densidade

O módulo *Mapa de Densidade* usa a Estimativa de Densidade Kernel para criar um matricial de densidade (mapa de temperatura) a partir de uma camada de pontos vectorial. A densidade calculada é baseada no número de pontos numa localização, com um conjunto largo de pontos agrupados resultando em valores altos. Os mapas de temperatura permitem facilmente identificar “pontos quentes” e agrupar pontos.

20.14.1 Activar o módulo de Mapa de Densidade

Primeiramente, este módulo base necessita de ser activado usando o Gestor de Módulos (veja *load_core_plugin*).


Depois da activação do Mapa de Densidade  o ícone pode ser encontrado na barra de ferramentas do Raster, e no menu *Raster* → *Mapas de Densidade*.


Selecione o menu *Ver* → *Barra de Ferramentas* → *Matricial* para exibir a Barra de Ferramentas Matricial se não estiver visível.

20.14.2 Usando o módulo de Mapa de Densidade

Ao clicar no botão da ferramenta **lhetmapl** *Mapas de Densidade* abre a janela do módulo Mapas de Densidade (veja *figure_heatmap_2*).

A janela tem as seguintes opções:

- **Camada de entrada de pontos:** Apresenta todas as camadas de pontos vectoriais no projecto actual e é usado para seleccionar a camada que vai ser analisada.
- **Ficheiro raster de saída:** Usa o botão  para seleccionar a pasta e o nome do ficheiro para o raster de saída que o módulo de mapas de densidade vai gerar. A extensão do ficheiro não é requerida.
- **Formato de Saída:** selecciona o formato de saída. Embora todos os formatos suportados pelo o GDAL poderem ser escolhidos, na maioria dos casos o GeoTIFF é o melhor formato de escolha.
- **Raio:** usado para especificar o raio de pesquisa (ou largura do kernel) do mapa de densidade em metros ou em unidades de mapa. O raio especifica a distância em torno de um ponto no qual se fará sentir a influência do ponto. Os valores altos resultam em maior suavização, mas valores pequenos podem mostrar detalhes finos e a variação da densidade de pontos.

Quando a caixa de verificação  *Avançada* é activada, opções adicionais serão disponibilizadas:

- **Linhas e Colunas:** usada para mudar as dimensões do raster de saída. Esses valores estão também ligados aos valores **Tamanho de célula X** e **Tamanho de célula Y**. Aumentando o número de linhas ou colunas irá diminuir o tamanho de célula e aumenta o tamanho do ficheiro de saída. Os valores nas Linhas e Colunas também estão ligados, portanto duplicando o número de linhas irá automaticamente duplicar o número de colunas e o tamanho da célula irá passar para metade. A área geográfica do raster de saída irá ser o mesmo!
- **Tamanho da célula X e Tamanho da célula Y:** controle o tamanho geográfico de cada pixel para o raster de saída. Alterando estes valores irá também mudar o número de Linhas e Colunas do raster de saída.
- **Forma do kernel:** A forma do kernel controla o rácio que influencia o ponto decrescente como a distância proveniente do ponto de crescente. Diferentes kerneis enfraquecem em diferentes rácios, portanto um kernel triweight dá elementos de maior peso para distâncias perto do ponto que o kernel de Epanechnikov dá. Consequentemente, o resultado triweight resulta em pontos quentes “nítidos”, e Epanechnikov resulta em pontos quentes “suavizados”. Um número de funções kernel padrão estão disponíveis no QGIS, que são descritos e ilustrados no [Wikipedia](#).
- **Rácio de declínio:** pode ser usado com kerneis triangulares para maior controlo de como a densidade a partir dos elementos decrescem com a distância a partir do elemento.

- O valor de 0 (=mínimo) indica que a densidade irá ser concentrada no centro do raio dado e será extinto no borda.
- O valor de 0.5 inca que esses pixeis da borda do raio serão atribuídos metade da densidade dos pixeis que estão no raio do centro de pesquisa.
- O valor de 1 refere que a densidade espalha-se por todo o raio do circulo de pesquisa. (Isto é equivalente ao kernel ‘Uniforme’).
- Um valor maior que 1 indica que a densidade é mais alta para a borda do raio de pesquisa, e em seguida, no centro.

A camada de pontos de entrada pode ter também campos de atributos que podem afectar como influencia o mapa de densidade:

- **Usar o raio a partir do campo:** define o raio de pesquisa para cada elemento da camada de entrada.
- **Usar o peso a partir do campo:** permite a introdução de elementos para ser pesado por um campo de atributo. Isto pode ser usado para aumentar a influência de certos elementos existente no resultado de mapa de densidade.

Quando o nome do ficheiro matricial de saída é especificado, o botão [OK] pode ser usado para criar o mapa de densidade.

20.14.3 Tutorial: Criando um Mapa de Densidade

Para o seguinte exemplo nós iremos usar a camada de pontos vectorial `airports` do conjunto de dados amostra do QGIS (veja *Amostra de Dados*). Outro excelente tutorial QGIS de produzir mapas de densidade podem ser encontrados em <http://qgis.spatialthoughts.com>.

Na `Figure_Heatmap_1` estão demonstrados os aeroportos do Alaska.

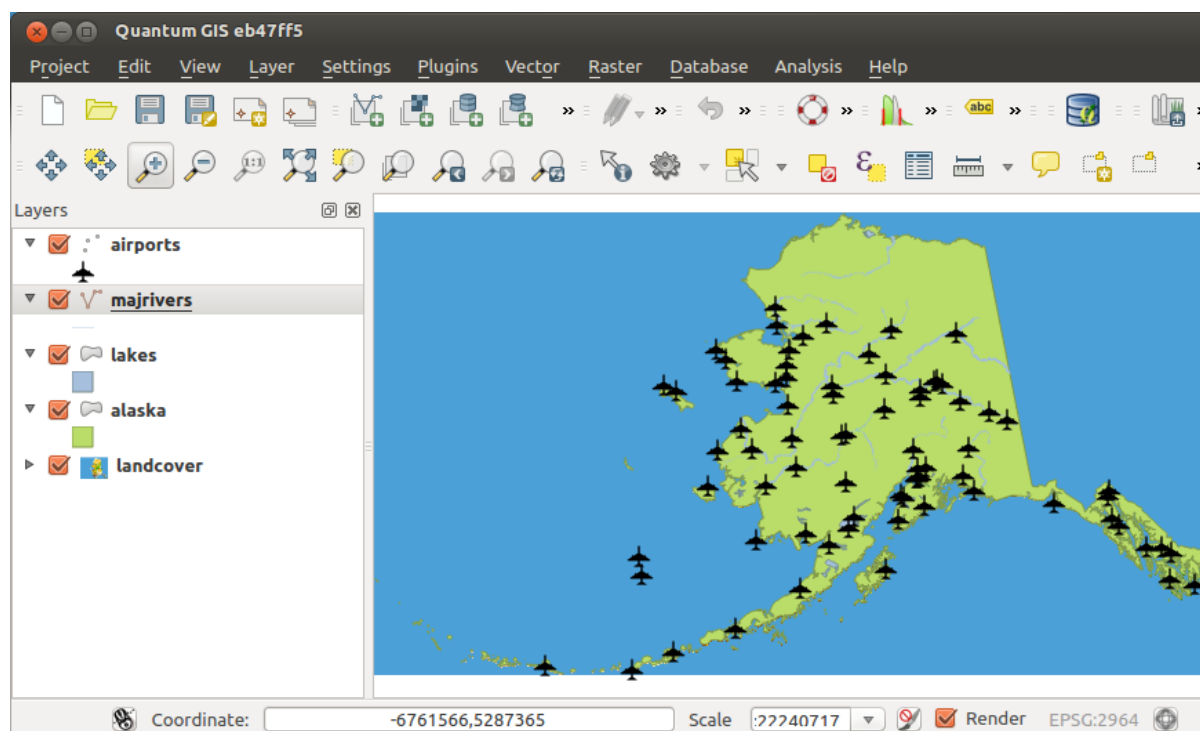




Figura 20.24: Aeroportos do Alaska 

1. Seleccione o botão da ferramenta **!heatmap!** *Mapa de Densidade* para abrir a janela do módulo Mapas de Densidades (veja `Figure_heatmap_2`).

2. No campo *Introduzir camada de pontos*  seleccione aeroportos da lista de camadas de pontos carregadas no projecto actual.
3. Especifique um ficheiro de saída clicando o botão  perto do campo *Raster de saída* . Introduza o nome do ficheiro `heatmap_airports` (não é necessário extensão).
4. Deixe o *Formato de Saída* o formato GeoTIFF como padrão.
5. Altere o *Raio* para 1000000 metros.
6. Clique em **[OK]** e carregue o mapa de temperatura dos aeroportos (veja [Figure_Heatmap_3](#)).

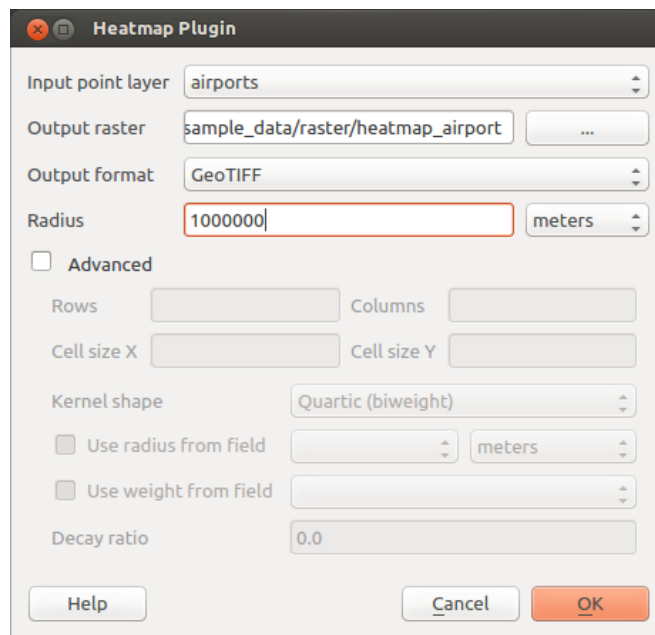




Figura 20.25: Janela do Mapa de Densidade 

O QGIS irá gerar o mapa de densidade e adicionar os resultados à sua janela do mapa. Por defeito, o mapa de densidade é sombreado a cinzento, com as áreas mais brilhantes concentradas nos aeroportos. O mapa de densidade pode ser personalizado no QGIS para melhorar a aparência.

1. Abra a janela de propriedades da camada do `heatmap_airports` (seleccione a camada `heatmap_airports`, abra o menu de contexto com o botão direito do rato e seleccione *Propriedades*).
2. Seleccione o separador *Estilo*.
3. Alterar o *Tipo de Renderização*  para 'Banda única pseudocor'.
4. Seleccione um suitable *mapa de cor* adequado , por exemplo `YlOrRed`.
5. Clique no botão **[Carregar]** para recolher os valores mínimos e máximos para cada matricial, de depois clique **no botão [Classificar]**.
6. Pressione **[OK]** para actualizar a camada.

O resultado final é demonstrado na [Figure_Heatmap_4](#).

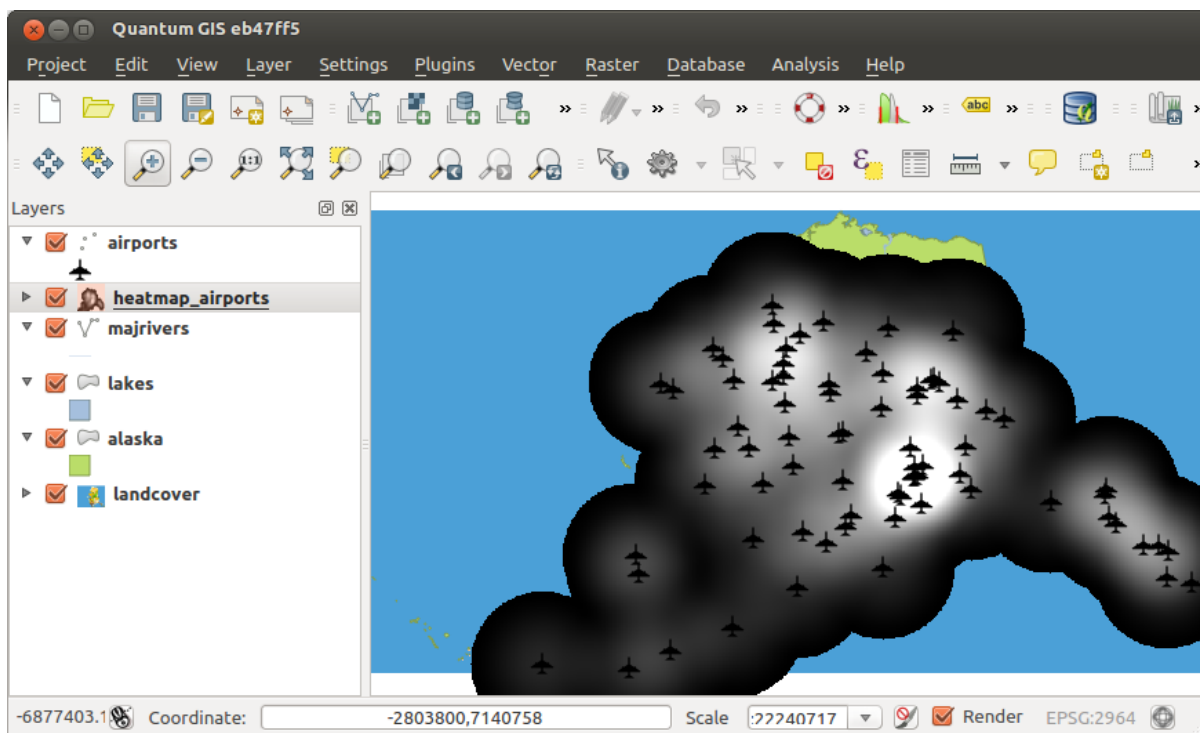


Figura 20.26: O mapa de densidade após ser carregado fica com uma superfície cinzenta 🐧

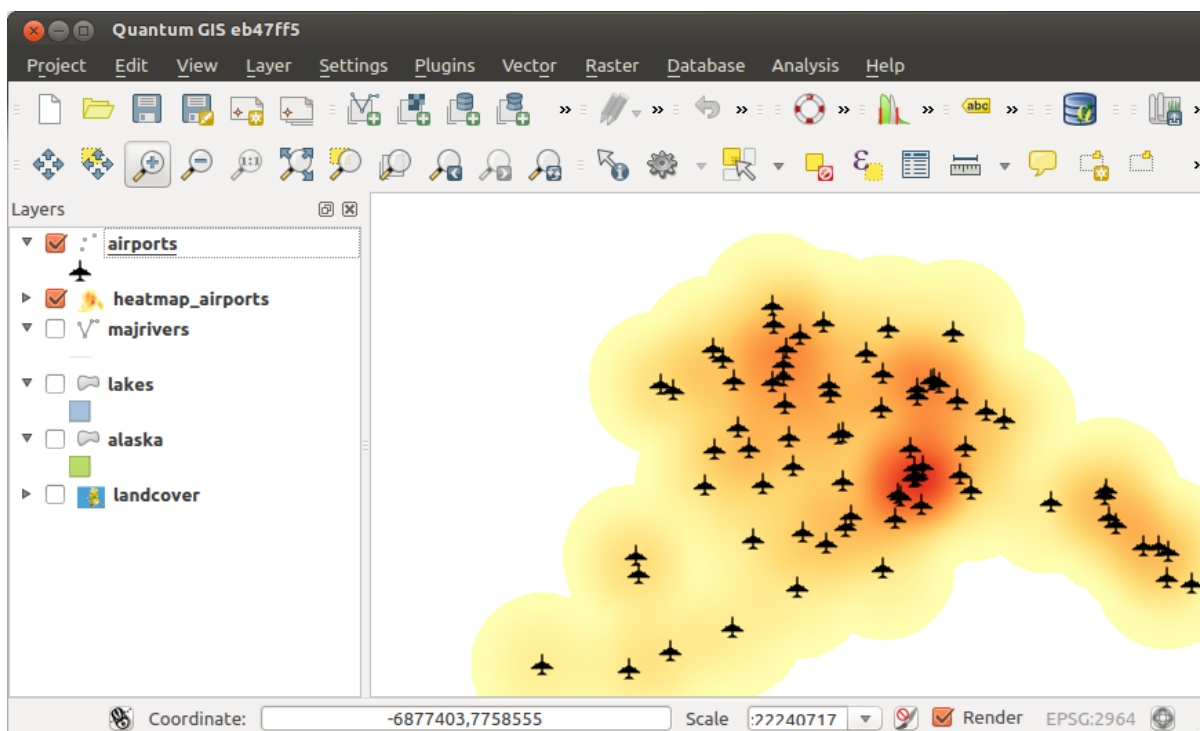
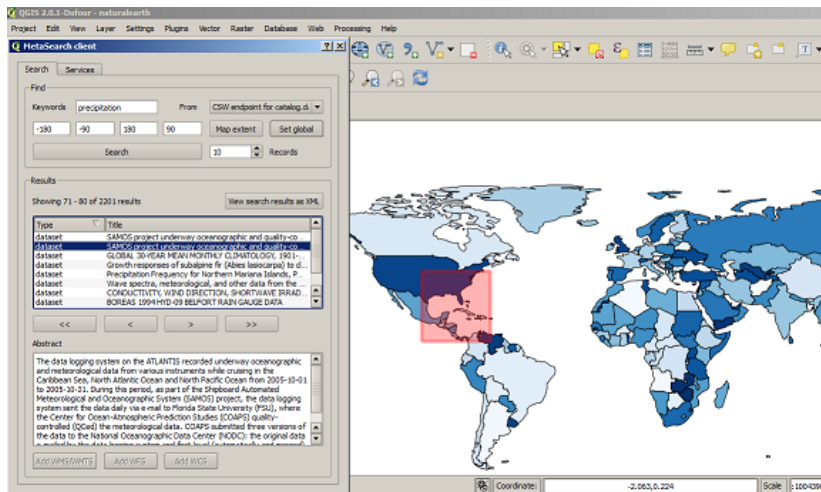


Figura 20.27: Mapa de Densidade decorado dos aeroportos do Alaska 🐧

20.15 MetaSearch Catalogue Client



20.15.1 Introduction

MetaSearch is a QGIS plugin to interact with metadata catalogue services, supporting the OGC Catalogue Service for the Web (CSW) standard.

MetaSearch provides an easy and intuitive approach and user-friendly interface to searching metadata catalogues within QGIS.

20.15.2 Installation

MetaSearch is included by default with QGIS 2.0 and higher. All dependencies are included within MetaSearch.

Install MetaSearch from the QGIS plugin manager, or manually from <http://plugins.qgis.org/plugins/MetaSearch>.

20.15.3 Working with Metadata Catalogues in QGIS

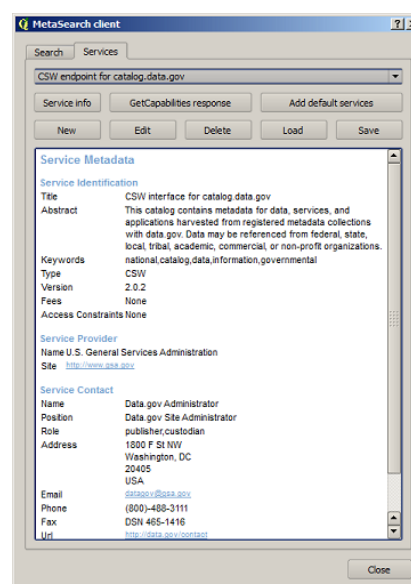
CSW (Catalogue Service for the Web)

CSW (Catalogue Service for the Web) is an OGC (Open Geospatial Consortium) specification, that defines common interfaces to discover, browse, and query metadata about data, services, and other potential resources.

Startup

To start MetaSearch, click the MetaSearch icon or select Web / MetaSearch / MetaSearch via the QGIS main menu. The MetaSearch dialog will appear. The main GUI consists of two tabs: 'Services' and 'Search'.

Managing Catalogue Services



The 'Services' tab allows the user to manage all available catalogue services. MetaSearch provides a default list of Catalogue Services, which can be added by pressing 'Add default services' button.

To all listed Catalogue Service entries, click the dropdown select box.

To add a Catalogue Service entry, click the 'New' button, and enter a Name for the service, as well as the URL/endpoint. Note that only the base URL is required (not a full GetCapabilities URL). Clicking ok will add the service to the list of entries.

To edit an existing Catalogue Service entry, select the entry you would like to edit and click the 'Edit' button, and modify the Name or URL values, then click ok.

To delete a Catalogue Service entry, select the entry you would like to delete and click the 'Delete' button. You will be asked to confirm deleting the entry.

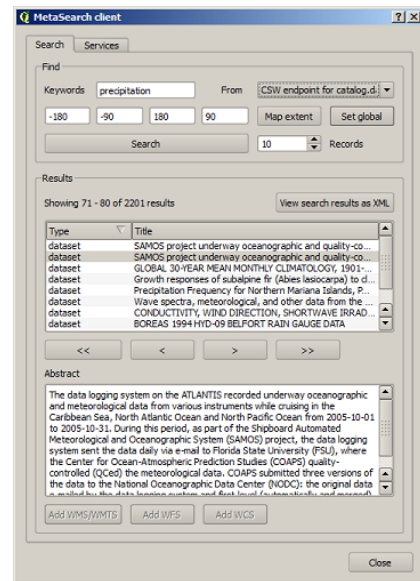
MetaSearch allows for loading and saving connections to an XML file. This is useful when you need to share settings between applications. Below is an example of the XML file format.

```
<?xml version="1.0" encoding="UTF-8"?>
<qgsCSWConnections version="1.0">
  <csw name="Data.gov CSW" url="http://catalog.data.gov/csw-all"/>
  <csw name="Geonorge - National CSW service for Norway" url="http://www.geonorge.no/geonetwork/eng/GeonorgeNationalCSWServiceForNorway">
  <csw name="Geoportale Nazionale - Servizio di ricerca Italiano" url="http://www.pcn.minambiente.it/geonetwork/eng/GeoportaleNazionaleServizioRicercaItaliano">
  <csw name="LINZ Data Service" url="http://data.linz.govt.nz/feeds/csw"/>
  <csw name="Nationaal Georegister (Nederland)" url="http://www.nationaalgeoregister.nl/geonetwork/eng/NationaalGeoregister">
  <csw name="RNDT - Repertorio Nazionale dei Dati Territoriali - Servizio di ricerca" url="http://www.rndt.it/geonetwork/eng/RepertorioNazionaledeiDatiTerritorialiServizioRicerca">
  <csw name="UK Location Catalogue Publishing Service" url="http://csw.data.gov.uk/geonetwork/eng/UKLocationCataloguePublishingService">
  <csw name="UNEP/GRID-Geneva Metadata Catalog" url="http://metadata.grid.unep.ch:8080/geonetwork/eng/UNEPGRIDGenevaMetadataCatalog">
</qgsCSWConnections>
```

To load a list of entries, click the 'Load' button. A new window will appear; click the 'Browse' button and navigate to the XML file of entries you wish to load and click 'Open'. The list of entries will be displayed. Select the entries you wish to add from the list and click 'Load'.

The 'Service info' button displays information about the selected Catalogue Service metadata such as service identification, service provider and contact information. If you would like to view the raw XML response, click the 'GetCapabilities response' button. A separate window will open displaying Capabilities XML.

Searching Catalogue Services



The ‘Search’ tab allows the user to query Catalogue Services for data and services, set various search parameters and view results.

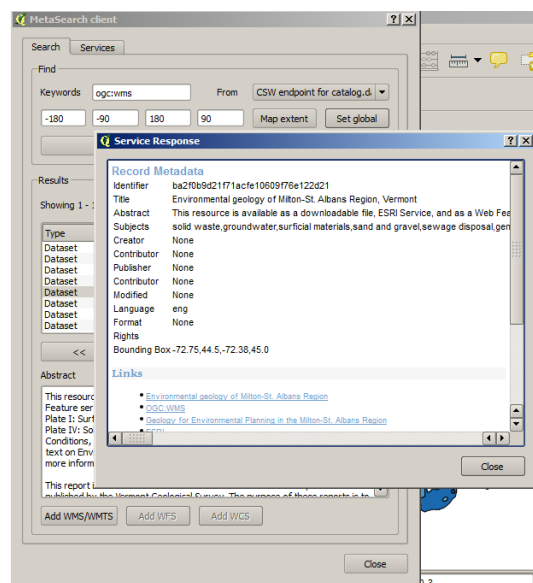
The following search parameters are available:

- **Keywords:** free text search keywords
- **From:** the Catalogue Service to perform the query against
- **Bounding box:** the spatial area of interest to filter on. The default bounding box is the map view / canvas. Click ‘Set global’ to do a global search, or enter custom values as desired
- **Records:** the number of records to return when searching. Default is 10 records

Clicking the ‘Search’ button will search the selected Metadata Catalogue. Search results are displayed in a list and are sortable by clicking on the column title. You can navigate through search results with the directional buttons below the search results. Clicking the ‘View search results as XML’ button opens a window with the service response in raw XML format.

Clicking a result will show the record’s abstract in the ‘Abstract’ window and provides the following options:

- if the metadata record has an associated bounding box, a footprint of the bounding box will be displayed on the map
- double-clicking the record displays the record metadata with any associated access links. Clicking the links opens the link in the user’s web browser
- if the record is an OGC web service (WMS/WMTS, WFS, WCS), the appropriate ‘Add to WMS/WMTS/WFS/WCS’ buttons will be enabled for the user to add to QGIS. When clicking this button, MetaSearch will verify if this is a valid OWS. The OWS will then be added to the appropriate QGIS connection list, and the appropriate WMS/WMTS/WFS/WCS connection dialogue will then appear



Settings

You can fine tune MetaSearch with the following settings:

- **Results paging:** when searching metadata catalogues, the number of results to show per page
- **Timeout:** when searching metadata catalogues, the number of seconds for blocking connection attempt. Default value is 10

20.16 Módulo de Cálculo de Rotas

O módulo de Análise de Redes é um módulo C++ para o QGIS, que calcula o caminho mais curto entre dois pontos em qualquer camada poli-linha e exhibe esse caminho sobre a rede viária.

Características principais:

- Calcula o caminho, o seu comprimento e o tempo de viagem.
- Optimiza por comprimento ou por tempo de viagem.
- Exporta o caminho para uma camada vectorial.
- Destaca as direcções na estrada (isto é lento e usado principalmente para fins de depuração e para testes de configuração).

Como camada vectorial de rede viária pode ser usada em qualquer formato de camada vectorial poli-linha suportado pelo QGIS. Duas linhas ligadas por um ponto em comum são considerados ligados. Por favor repare, que é necessário usar o SRC da camada como SRC do projecto enquanto edita as camadas de rede viária. Isto é devido ao facto da necessidade de recalculas as coordenadas entre diferentes SRC introduzindo alguns erros que podem resultar descontinuidades, mesmo quando o 'snapping' é usado.

Na tabela de atributos da camada os seguintes campos podem ser usados:

- Velocidade na secção da rede viária (campo numérico).
- direcção (qualquer tipo, que possa ser usada para string). Direcções com avanço e recuo corresponde a uma estrada de sentido único, para ambas as direcções corresponde a estradas de dois sentidos.

Se alguns campos não tiverem qualquer valor ou não existam serão usados valores por defeito. Pode mudar os valores por defeito e algumas configurações na janela de configurações do módulo.

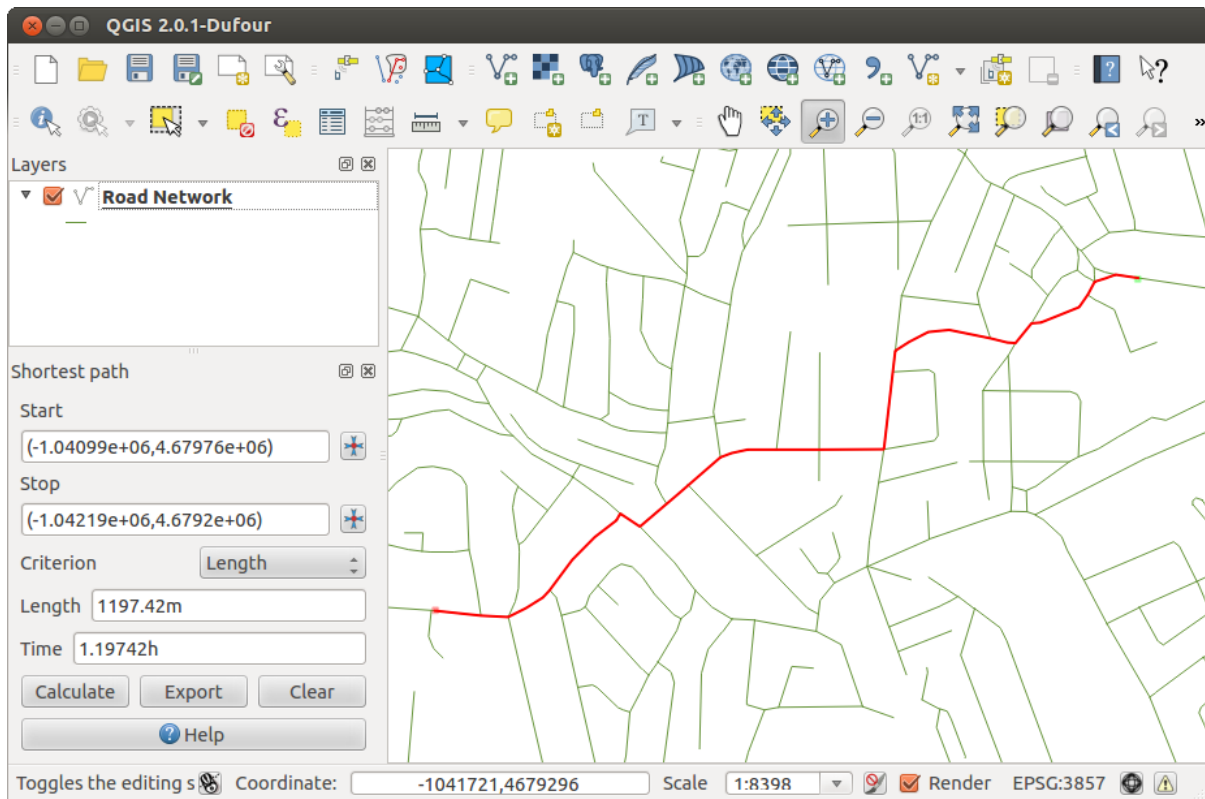


Figura 20.28: Módulo de Cálculo de Rotas


20.16.1 Usando o módulo

Depois da activação do módulo, irá ver um painel adicional no lado esquerdo da janela principal do QGIS. Agora introduza alguns parâmetros na janela *Configuração do Caminho mais curto* no menu *Vector* → *Caminho mais curto* (veja [figure_road_graph_2](#)).

Após configurar a *Unidade de tempo*, *Unidade de distância* e *Tolerância da topologia* pode escolher a camada vectorial no separador *Camada de Transportação*. Aqui pode também escolher *Campo de direcção* e o *Campo da velocidade*. No separador *Configurações padrão* pode definir *:guilabel: 'Direcção* para o cálculo.

Finalmente no painel *:guilabel: 'Caminho mais curto'* seleccione um ponto de Iniciar e Parar na camada de rede viária e clique em [**Calcular**].

20.17 Módulo de Consulta Espacial

O módulo  Módulo de Consulta Espacial permite que faça consultas espaciais (ex.: seleccionar elementos) numa camada alvo com referência a outra camada. A funcionalidade é baseada na biblioteca GEOS e depende de uma camada fonte de elementos seleccionada.

Os operadores possíveis são:

- Contém
- Igual
- Sobrepõe-se
- Cruza
- Intersecta

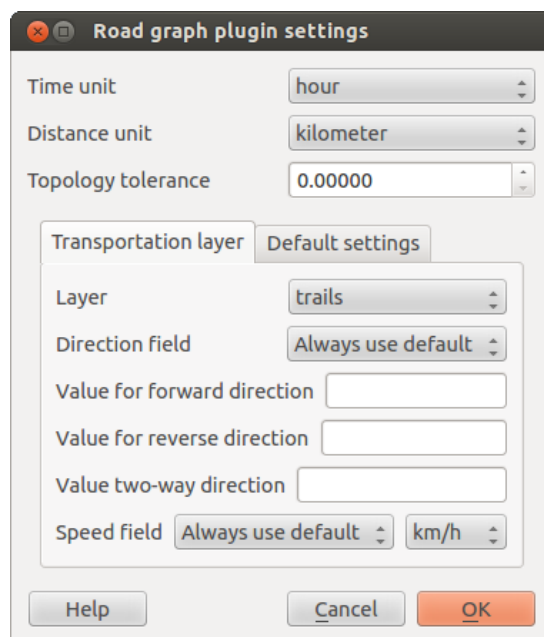




Figura 20.29: Configurações do módulo de Cálculo de Rotas 




- É separado
- Toca
- Dentro de

20.17.1 Usando o módulo

Como exemplo, nós queremos encontrar as regiões no conjunto de dados do Alaska que contém os aeroportos. Os seguintes passos são necessários:

1. Inicie o QGIS e carregue uma camada vectorial `regions.shp` e `airports.shp`.
2. Carregue o módulo Consulta Espacial no Gestor de Módulos (see *The Plugins Dialog*) e clique no ícone  Consulta Espacial que aparece no menu de ferramentas do QGIS. A janela do módulo aparece.
3. Selecciona a camada `regions` como camada fonte e `airports` como camada de elementos referência.
4. Selecciona 'Contém' como operador e clique [Aplicar].

Agora obtém uma lista de elementos ID da consulta e tem várias opções como é mostrado na `figure_spatial_query_1`.

- Clique no  Criar camada com a lista de itens
- Selecciona um ID da lista e clique em  Criar uma camada com os seleccionados.
- Selecciona 'Remover da selecção actual' no campo *E uso resultado para* .
- Additionally, you can *Zoom to item* or display *Log messages*.

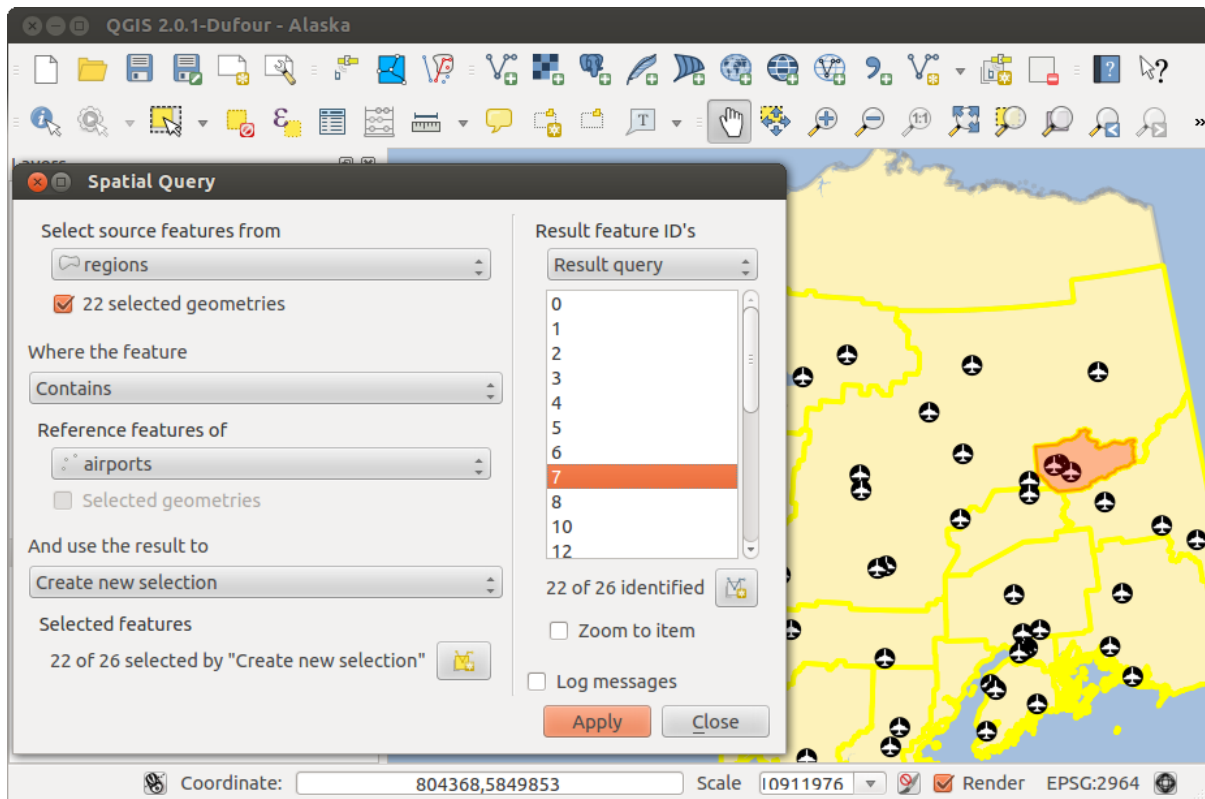




Figura 20.30: Análise da Consulta Espacial - regiões que contêm os aeroportos 🐧


20.18 Módulo SPIT

O QGIS vem com um módulo chamado SPIT (Ferramenta de Importação de Shapefile para PostGIS). O SPIT pode ser usado para carregar múltiplas shapefiles de uma só vez e inclui o suporte para esquemas. Para usar o SPIT, abra o Gestor de Módulos do menu *Módulos*, no menu  *Instalado* e marque a caixa perto do  :guilabel:'SPIT' e clique [OK].

Para importar uma shapefile, use *Database* → *Spit* → *Importar Shapefiles para PostgreSQL* da barra de menu para abrir a janela *SPIT - Ferramenta de Importação de Shapefile para PostGIS*. Seleccione a base de dados PostGIS que quer ligar e clique em [Ligar]. Se quiser, pode definir ou alterar opções de importação. Agora pode adicionar um ou mais ficheiros para a fila clicando no botão [Adicionar]. Para processar os ficheiros, clique no botão [OK]. O progresso de importação assim como algum tipo de erros/avisos serão exibidos em cada shapefile processada.

20.19 Módulo SQL Anywhere

SQL Anywhere is a proprietary relational database management system (RDBMS) from Sybase. SQL Anywhere provides spatial support, including OGC, shapefiles and built-in functions to export to KML, GML and SVG formats.

 SQL Anywhere allows you to connect to spatially enabled SQL Anywhere databases. The *Add SQL Anywhere layer* dialog is similar in functionality to the dialogs for PostGIS and SpatialLite.

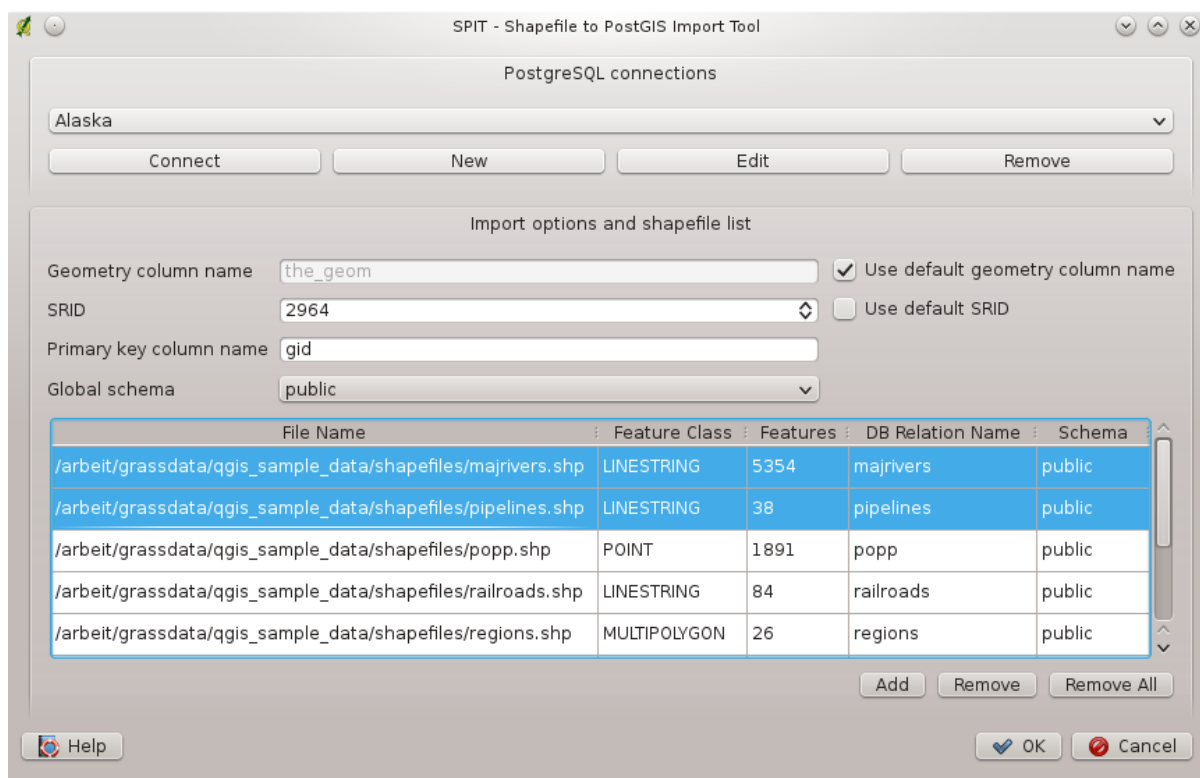


Figura 20.31: Usando o Módulo SPIT para importar Shapefiles para PostGIS 🐧

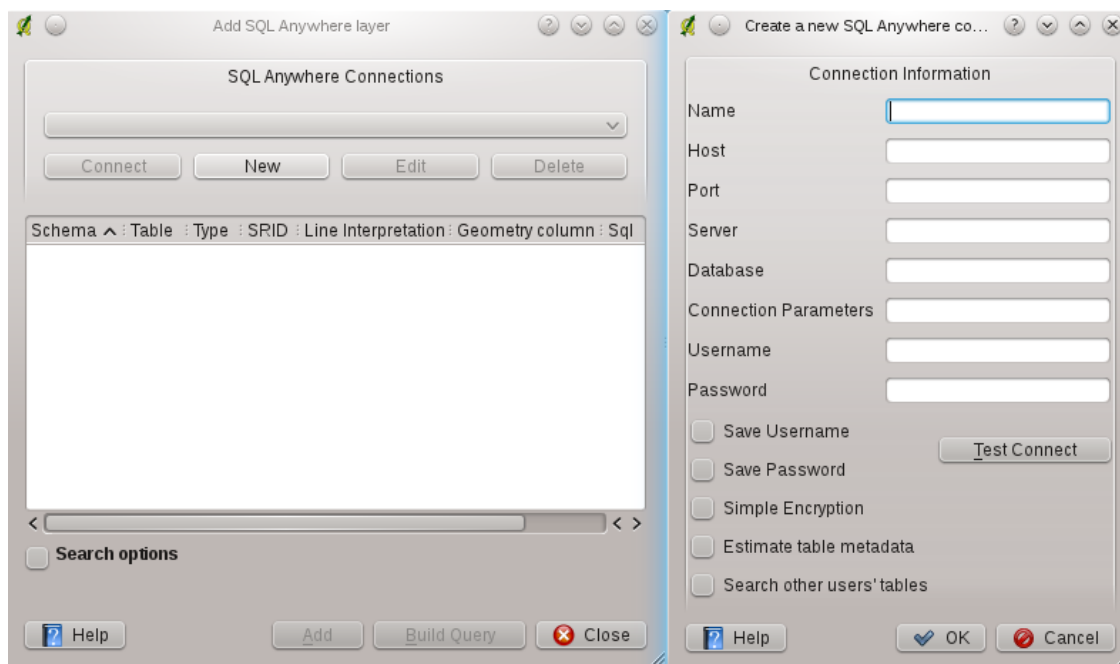


Figura 20.32: Janela SQL Anywhere (KDE) 🐧

20.20 Módulo Verificador de Topologia

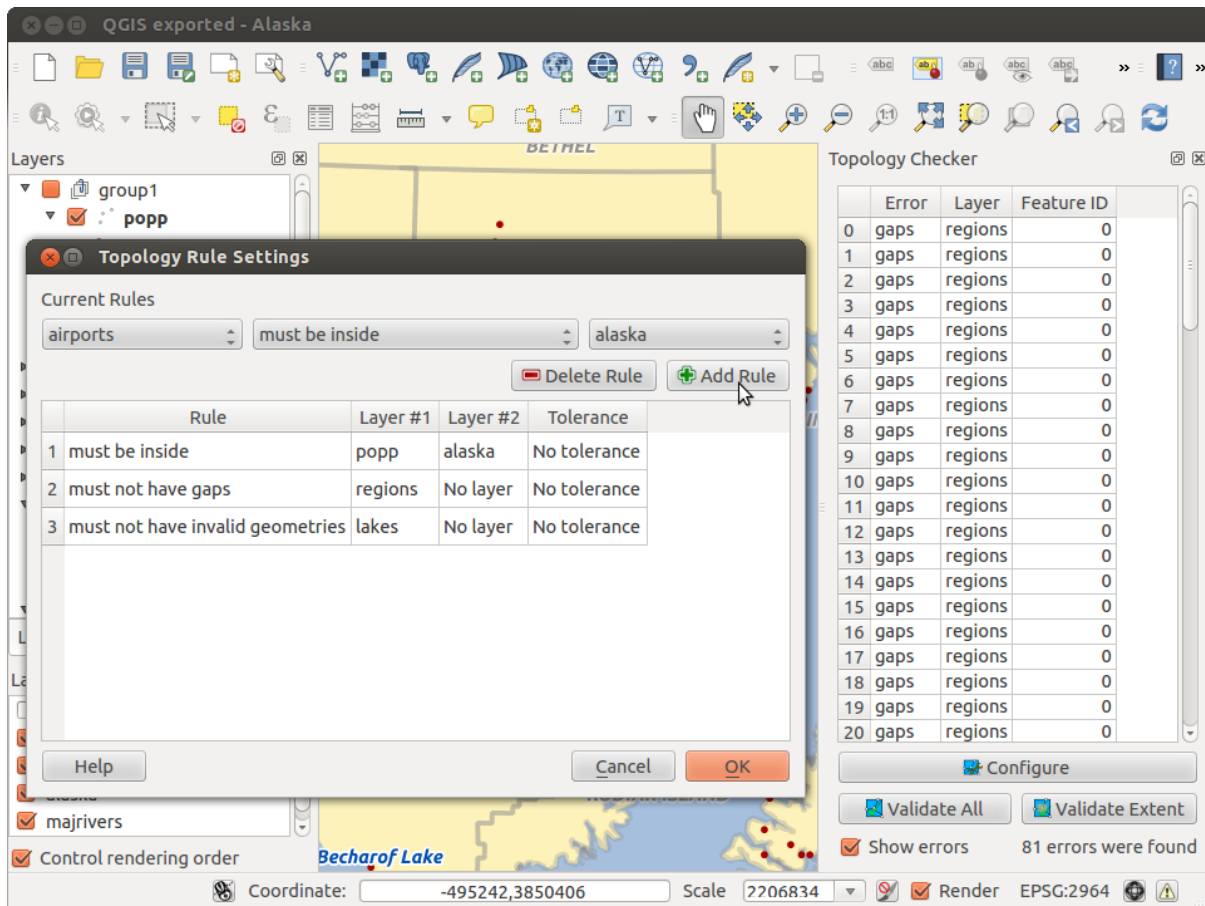


Figura 20.33: O Módulo Verificador de Topologia

Topology describes the relationships between points, lines and polygons that represent the features of a geographic region. With the Topology Checker plugin, you can look over your vector files and check the topology with several topology rules. These rules check with spatial relations whether your features ‘Equal’, ‘Contain’, ‘Cover’, are ‘CoveredBy’, ‘Cross’, are ‘Disjoint’, ‘Intersect’, ‘Overlap’, ‘Touch’ or are ‘Within’ each other. It depends on your individual questions which topology rules you apply to your vector data (e.g., normally you won’t accept overshoots in line layers, but if they depict dead-end streets you won’t remove them from your vector layer).

QGIS has a built-in topological editing feature, which is great for creating new features without errors. But existing data errors and user-induced errors are hard to find. This plugin helps you find such errors through a list of rules.

É muito simples para criar regras topológicas com o módulo Verificador de Topologia

Nas **camadas do tipo vector** as seguintes regras estão disponíveis:

- **Must be covered by:** Here you can choose a vector layer from your project. Points that aren’t covered by the given vector layer occur in the ‘Error’ field.
- **Must be covered by endpoints of:** Here you can choose a line layer from your project.
- **Must be inside:** Here you can choose a polygon layer from your project. The points must be inside a polygon. Otherwise, QGIS writes an ‘Error’ for the point.
- **Must not have duplicates:** Whenever a point is represented twice or more, it will occur in the ‘Error’ field.
- **Must not have invalid geometries:** Checks whether the geometries are valid.
- **Must not have multi-part-geometries:** All multi-part points are written into the ‘Error’ field.


On **line layers**, the following rules are available:

- **End points must be covered by:** Here you can select a point layer from your project.
- **Must not have dangles:** This will show the overshoots in the line layer.
- **Must not have duplicates:** Whenever a line feature is represented twice or more, it will occur in the 'Error' field.
- **Must not have invalid geometries:** Checks whether the geometries are valid.
- **Must not have multi-part geometries:** Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. All multi-part lines are written into the 'Error' field.
- **Must not have pseudos:** A line geometry's endpoint should be connected to the endpoints of two other geometries. If the endpoint is connected to only one other geometry's endpoint, the endpoint is called a pseudo node.

On **polygon layers**, the following rules are available:

- **Must contain:** Polygon layer must contain at least one point geometry from the second layer.
- **Must not have duplicates:** Polygons from the same layer must not have identical geometries. Whenever a polygon feature is represented twice or more it will occur in the 'Error' field.
- **Must not have gaps:** Adjacent polygons should not form gaps between them. Administrative boundaries could be mentioned as an example (US state polygons do not have any gaps between them...).
- **Must not have invalid geometries:** Checks whether the geometries are valid. Some of the rules that define a valid geometry are:
 - Os anéis do polígono devem estar fechados.
 - Anéis que definem buracos devem estar dentro de anéis que definem os limites exteriores.
 - Os anéis não se podem intersectar a si mesmo (nem mesmo tocar ou cruzar um no outro).
 - Os anéis não podem tocar outros anéis, à excepção de um ponto.
- **Must not have multi-part geometries:** Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. For example, a country consisting of multiple islands can be represented as a multi-polygon.
- **Must not overlap:** Adjacent polygons should not share common area.
- **Must not overlap with:** Adjacent polygons from one layer should not share common area with polygons from another layer.

20.21 Módulo de Estatística Zonal

Com o  Módulo de Estatística Zonal pode analisar os resultados de uma classificação temática. Permite calcular vários valores de pixels de um raster com ajuda de uma camada poligonal (veja [figure_zonal_statistics](#)). Pode calcular a soma, valor médio e contagem total dos pixels que estão dentro do polígono. Este módulo gera uma colunas de saída na camada vectorial com um prefixo definido pelo utilizador.

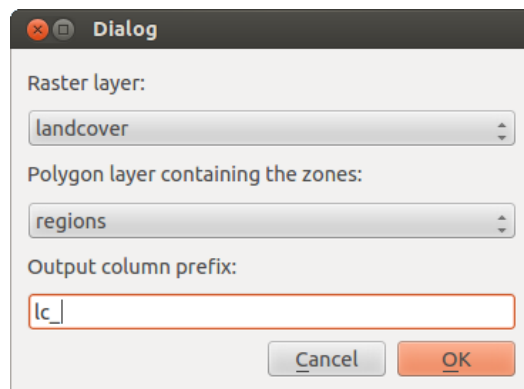


Figura 20.34: Janela de Estatística Zonal (KDE) 

Ajuda e Suporte

21.1 Listas de Discussão

O QGIS está activo no seu desenvolvimento e como tal não irá estar sempre a trabalhar como deseja. A forma mais desejável para obter ajuda será juntar-se na lista de discussão qgis-users. As suas questões irão chegar a uma audiência mais ampla e as respostas irão beneficiar outros.

21.1.1 qgis-users

Esta lista de discussão é usada para uma discussão geral do QGIS, assim como as questões específicas relacionadas com a instalação e o uso. Pode subscrever na lista de discussão do qgis-users visitando o seguinte URL: <http://lists.osgeo.org/mailman/listinfo/qgis-user>

21.1.2 fossgis-talk-liste

Para os utilizadores de língua alemã o FOSSGIS e.V. Alemão fornece uma lista de discussão fossgis-talk-liste. Esta lista de discussão é usada para a discussão dos SIG de código aberto incluindo o QGIS. Pode subscrever na lista de discussão fossgis-talk-liste visitando o seguinte URL: <https://lists.fossgis.de/mailman/listinfo/fossgis-talk-liste>

21.1.3 qgis-developer

Se não for um programador que está a enfrentar problemas de natureza técnica, vai desejar juntar-se À lista de discussão qgis-developer aqui: <http://lists.osgeo.org/mailman/listinfo/qgis-developer>

21.1.4 qgis-commit

Cada vez que um commit é feito no repositório de código QGIS um email é publicado nesta lista. Se pretende estar ocorrente de todas as alterações ao código base actual, pode subscrever-se nesta lista em: <http://lists.osgeo.org/mailman/listinfo/qgis-commit>

21.1.5 qgis-trac

A lista fornece notificações de email relacionadas com a gestão do projecto, incluindo relatórios de erros, tarefas, pedidos de novas características. Pode subscrever esta lista em: <http://lists.osgeo.org/mailman/listinfo/qgis-trac>

21.1.6 qgis-community-team

Esta lista lida com tópicos como a documentação, conteúdos de ajuda, manual de utilizador, experiência online incluindo sítios na internet, blogues, listas de discussão, fóruns, e apoio nas traduções. Se também desejar trabalhar no guia de utilizador, esta lista é boa para começar a fazer perguntas para as suas questões. Pode subscrever na lista em: <http://lists.osgeo.org/mailman/listinfo/qgis-community-team>

21.1.7 qgis-release-team

Esta lista lida com os tópicos relacionados com os processos de lançamento, empacotamento de binários para os vários SO e anúncio de novos lançamentos para o mundo. Pode subscrever esta lista em: <http://lists.osgeo.org/mailman/listinfo/qgis-release-team>

21.1.8 qgis-tr

Esta lista lida com os esforços feitos nas traduções. Se quiser trabalhar na tradução dos manuais ou na interface gráfica de utilizador (GUI), esta lista é boa para começar a fazer perguntas. Pode subscrever nesta lista em: <http://lists.osgeo.org/mailman/listinfo/qgis-tr>

21.1.9 qgis-edu

Esta lista lida com os esforços na área de educação do QGIS. Se quiser trabalhar nos materiais de educação, esta lista é boa para começar a fazer perguntas. Pode subscrever nesta lista em: <http://lists.osgeo.org/mailman/listinfo/qgis-edu>

21.1.10 qgis-psc

A lista é usado para discutir assuntos do Comité de Direcção relacionados com a boa gestão e direcção do QGIS. Pode subscrever nesta lista em: <http://lists.osgeo.org/mailman/listinfo/qgis-psc>

É bem-vindo para subscrever em qualquer lista. Por favor lembre-se que a contribuição na lista é feita pela resposta de questões e partilhar as suas experiências. Note que o qgis-commit e qgis-trac estão desenhados para notificar e não para publicações de utilizadores.

21.2 IRC

Nós também mantemos um IRC - visite-nos ao juntar-se no canal #qgis no irc.freenode.net. Por favor espere por uma resposta para a sua questão porque as outras pessoas do canal podem estar a fazer outras coisas que podem demorar um pouco a repararem na sua questão. Se perder a discussão no IRC, não é problema! Nós registamos toda a discussão, portanto é fácil estar actualizado. Vá a <http://qgis.org/irclogs> and read the IRC-logs.

Suporte comercial para o QGIS está também disponível. Verifique o sítio na internet <http://qgis.org/en/commercial-support.html> for more information.

21.3 BugTracker

Enquanto que a lista de discussão do qgis-users é útil para as típicas questões ‘como eu faço o XYZ no QGIS’, pode notificar-nos sobre erros existentes no QGIS. Pode submeter erros usando o rastreio de erros QGIS em <http://hub.qgis.org/projects/quantum-gis/issues>. Quando cria uma nova notificação para erro. por favor forneça um endereço email onde poderemos pedir informação adicional.

Por favor tenha em atenção que o seu erro pode nem sempre ter a prioridade que deseja (vai depender da severidade). Alguns erros podem requer esforços significativos de programadores para remediar e a mão-de-obra nem sempre está disponível para isso.

Os pedidos de novas características também podem ser submetidos usando o mesmo sistema de senhas que é usado para os erros. Por favor certifique-se que selecciona o tipo `Feature`.

Se encontrou um erro e corrigiu-o pode submeter esse patch. Mais uma vez, o adorável sistema de avisos redmine em <http://hub.qgis.org/wiki/quantum-gis/issues> tem também disto. Active a caixa de verificação `Patch fornecido` e anexe o seu patch antes de submeter o erro. Alguém dos programadores irá revê-lo e aplicá-lo no QGIS. Por favor não fique alarmado se o seu patch não for logo aplicado — os programadores podem estar ocupados com outros compromissos.

21.4 Blogue

A comunidade QGIS também possui um blogue em <http://planet.qgis.org/planet/> que tem alguns artigos interessantes para utilizadores e programadores assim como outros blogs da comunidade. Está convidado para contribuir com o seu blogue QGIS!

21.5 Módulos

A página web <http://plugins.qgis.org> fornece o portal oficial de módulos QGIS. Aqui pode encontrar uma lista de todos os módulos estáveis e experimentais do QGIS através do 'Repositório Oficial de Módulos QGIS'.

21.6 Wiki

Finalmente, nós mantemos uma página WIKI em <http://hub.qgis.org/projects/quantum-gis/wiki> onde pode encontrar uma variedade de informação útil relacionada com o desenvolvimentos so QGIS, planos de lançamento, hiperligações para sítios de transferência, mensagens de dicas de tradução entre outros. Dê uma vista de olhos, tem lá boas explicações!

.

22.1 Licença Geral Pública GNU

Versão 2, Junho de 1991

Direitos Reservados (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

É permitido a todos a cópia e distribuição de cópias escritas deste documento de licença, mas não é permitido alterá-lo.

Preâmbulo

As licenças da maior parte do software são criadas para limitar a sua liberdade de o partilhar e alterar. Pelo contrário, a GNU General Public License tem o propósito de garantir a sua liberdade de partilhar e modificar software livre—para certificar que o software é livre para todos os utilizador. Esta Licença Pública Geral aplica-se à maior parte dos software da Free Software Foundation e para outros programas cujos autores pretendam utilizá-lo. (em vez disso, alguns softwares da Free Software Foundation são protegidos pela GNU Library General Public License). Também pode aplica-lo aos seus programas.

Quando falamos de software livre, nós estamos a referir à liberdade, não preço. As nossas Licenças Gerais Públicas estão feitas para assegurar que pode ter a liberdade de distribuir cópias de software livre (e cobrar por esse serviço se desejar), de onde recebe código fonte ou pode obtê-lo se quiser, e pode alterar o software ou usar peças dele em novos programas; e fica a saber que pode fazer essas coisas.

Para proteger os seus direitos, nós necessitamos de fazer algumas restrições que proíbem qualquer pessoa de lhe negar os direitos ou de lhe perguntar se abdica dos direitos. Estas restrições traduzem-se em certas responsabilidades para si se deseja distribuir cópias do software, ou se quer modificá-lo.

Por exemplo, se distribuir cópias de um programa, seja gratuito ou por um valor, deve dar aos destinatários todos os direitos que tem. Deve ter a certeza que eles, também recebem ou podem obter o código fonte. E deve mostrar-lhes estes termos para que possam conhecer os seus direitos.

Nós protegemos os seus direitos de duas formas: (1) direitos reservados do software, e (2) oferta desta licença que dá-lhe permissão legal para copiar, distribuir e/ou modificar o software.

Além disso, para nossa proteção e de casa um dos autores, queremos-nos certificar de que todos percebem que não há garantia para este software livre. Se o software é modificado por outra pessoa e passado, pretendemos que os detentários saibam que essa não é a versão original. Assim, quaisquer problemas introduzidos por outras pessoas não irão interferir com a reputação dos autores originais.

Por fim, qualquer programa livre é ameaçado em permanência por patentes de software. Pretendemos evitar o perigo de que os redistribuidores de um programa livre irão obter licenças de patentes individualmente, tornando-se proprietários do programa. Para evitar isto, tornamos claro que qualquer patente deverá ser licenciada para o uso livre de qualquer um, ou não licenciada.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. A Licença aplica-se a qualquer programa ou outro trabalho que contenha uma nota colocada pelo detentor dos direitos a referir que pode ser distribuído segundo os termos desta General Public License. O “Programa” abaixo, refere-se a qualquer propracama ou trabalho, e a “trabalho baseado no Programa” refere-se tanto ao programa ou qualquer trabalho derivado sob o direitos de autor: o que significa, um trabalho com o Programa ou parte dele, tanto copiado como com modificações e/ou traduzido para outra língua. (Desta forma, a tradução é incluída sem o termo “modificação”.) Cada licença é endereçada como “tu”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Deste modo, não é intenção desta secção reclamar direitos ou contestar os seus direitos do trabalho escrito inteiramente por você; serve mais para exercer o direito de controlo da distribuição de derivados ou trabalhos colectivos baseados no Programa.

Em adição, a mera agregação de outro trabalho que não está baseado no Programa dentro do Programa (ou com o trabalho baseado no Programa) ou num volume de armazenamento ou numa média distribuição não traz o outro trabalho ao alcance desta Licença.

3. Pode copiar e distribuir o Programa (ou basear o seu trabalho nele, segundo a Secção 2) no código do objecto ou na forma executável nos termos da Secção 1 e 2 acima fornecidos desde que faça um dos seguintes casos:
 - (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

- (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Exceção do QGIS Qt para GPL

In addition, as a special exception, the QGIS Development Team gives permission to link the code of this program with the Qt library, including but not limited to the following versions (both free and commercial): Qt/Non-commercial Windows, Qt/Windows, Qt/X11, Qt/Mac, and Qt/Embedded (or with modified versions of Qt that use the same license as Qt), and distribute linked combinations including the two. You must obey the GNU General Public License in all respects for all of the code used other than Qt. If you modify this file, you may extend this exception to your version of the file, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

22.2 Licença de Documentação Livre GNU

Versão 1.3, Novembro de 2008

Copyright 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preâmbulo

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APLICABILIDADE E DEFINIÇÕES

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **Document**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called **Opaque**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.)

To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. CÓPIA INTEGRAL

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. CÓPIAS EM QUANTIDADE

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICAÇÕES

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

1. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
2. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
3. State on the Title page the name of the publisher of the Modified Version, as the publisher.
4. Preserve all the copyright notices of the Document.
5. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
6. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

7. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
8. Incluir uma cópia inalterada desta Licença.
9. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
10. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
11. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
12. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
13. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
14. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
15. Preserve qualquer Cláusula de Garantias.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINAÇÃO DE DOCUMENTOS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLEÇÕES DE DOCUMENTOS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGREGAÇÃO COM TRABALHADORES INDEPENDENTES

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRADUÇÃO

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINAÇÕES

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURAS REVISÕES DESTA LICENÇA

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a

proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RE-LICENCIAMENTO

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADENDA: Como usar esta Licença nos seus documentos

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Literatura e Referências Web

GDAL-SOFTWARE-SUITE. Biblioteca de Abstracção de Dados Geoespaciais. <http://www.gdal.org>, 2013.

GRASS-PROJECT. Sistema de suporte de recursos de análise geográfica. <http://grass.osgeo.org> , 2013.

NETELER, M., AND MITASOVA, H. Open source gis: A grass gis approach, 2008.

OGR-SOFTWARE-SUITE. Biblioteca de Abstracção de Dados Geoespaciais. <http://www.gdal.org/ogr> , 2013.

OPEN-GEOSPATIAL-CONSORTIUM. Especificação de implementação de serviços do mapa Web (1.1.1) . <http://portal.opengeospatial.org>, 2002.

OPEN-GEOSPATIAL-CONSORTIUM. Especificação de implementação de serviços do mapa Web (1.3.0) . <http://portal.opengeospatial.org>, 2004.

POSTGIS-PROJECT. Suporte espacial para postgresql. <http://postgis.refrations.net/> , 2013.

-
- %%, 102
 - ódulos, 665

 - Ações, 102
 - Ajuda de contexto, 33
 - ampliação com a roda do rato, 31
 - anotação, 40
 - apache, 157
 - apache2, 157
 - Arc/Info_ASCII_Grid, 135
 - Arc/Info_Binary_Grid, 135
 - ArcInfo_Binary_Coverage, 68
 - Atalhos do teclado, 33
 - Atlas_Generation, 660
 - Atracção, 114
 - attribute table, 126
 - Attribute_Actions, 102
 - Attribute_Table, 650
 - Attribute_Table_Selection, 126
 - atualizar renderização durante o desenho, 35
 - Avoid_Intersections_Of_Polygons, 116

 - Banda_Raster_Simples, 137
 - Barra de Escala
 - Map_Scalebar, 646
 - barra de ferramentas, 28
 - barra de ferramentas do layout, 29
 - Browse_Maps, 63

 - cálculo da escala, 32
 - caixa de ferramentas GRASS, 179
 - personalizar, 186
 - Pesquisar, 186
 - Calculator_Field, 132
 - CAT, 147
 - Categorized_Renderer, 82
 - CGI, 156
 - Colliding_labels, 90
 - color_Ramp, 79
 - colorBrewer, 79
 - Comma Separated Values, 68
 - Common_Gateway_Interface, 156
 - Compose_Maps, 629
 - Composer_Manager, 663
 - Composer_Template, 630

 - Coordinate_Reference_System, 151
 - Cor do mapa, 140
 - Cor_Interpolação, 140
 - Create_Maps, 629
 - Create_New_Layers, 123
 - crossing the 180 degrees longitude line, 73
 - CRS, 151
 - CSV, 68, 118
 - Current_Edits, 117
 - Custom_color_Ramp, 79

 - Datum_transformação, 61
 - DB_Manager, 75
 - Debian_Squeeze, 157
 - define an action, 102
 - Derived_Fields, 132
 - Digitalização, 116
 - Discreto, 140
 - Displacement_plugin, 85
 - documentação, 7
 - documento de licença, 719

 - editing, 114
 - Elements_Alignment, 658
 - Enquadramento do Mapa, 45
 - EPSG, 57
 - Equal_Interval, 83
 - Erdas Imagine, 135
 - Escala, 34
 - ESRI, 65
 - European_Petroleum_Search_Group, 57
 - exemplo de acções, 102
 - Export_as_image, 662
 - Export_as_PDF, 662
 - Export_as_SVG, 662
 - Expressions, 107

 - FastCGI, 156
 - Ferramenta_Consulta, 130
 - Ferramentas de Análise, 682
 - Ferramentas de investigação, 683
 - Ferramentas do Georeferenciador, 689
 - ficheiro de saída guardado como imagem, 20
 - Field_Calculator, 132
 - Field_Calculator_Functions, 108
-

- GDAL, 135
- GeoTIFF, 135
- GeoTiff, 135
- GiST (Generalized Search Tree) index, 73
- GML, 147
- GNU General Public License, 719
- Gradient_color_Ramp, 79
- Graduated_Renderer, 83
- GRASS, 171, *ver* Criando novos vetores;editando;criando uma nova camafa
 - armazenamento de atributos, 176
 - caixa de ferramentas, 183
 - configurações da categoria, 177
 - configurações de simbologia, 178
 - edição da região, 179
 - edição de tabela, 179
 - exibir região, 179
 - exibir resultados, 180, 183
 - ferramentas de digitalização, 176
 - ligação de atributos, 176
 - região, 179
 - tolerância de atracção, 178
- Grelha
 - Grids
 - Map_Grid, 637
- Histograma, 142
- HTML_Frame, 655
- Identificar elementos, 37
- IGNF, 57
- Import_Maps, 63
- impressão rápida do compositor de impressão, 20
- Imprimindo
 - Export_Map, 662
- Institut_Geographique_National_de_France, 57
- InteProxy, 155
- Inverted_Polygon_Renderer, 85
- janela principal, 21
- join, 104
- join layer, 104
- Layout_Maps, 629
- legenda, 29
- load a shapefile, 66
- loading_raster, 135
- Map_Legend, 643
- Map_Navigation, 115
- Map_Template, 630
- MapInfo, 68
- marcadores, 41
- Marcadores espaciais
 - ver* marcadores, 41
- medir, 35
 - áreas, 35
 - ângulos, 35
 - comprimento da linha, 35
 - Melhorar_Contraste, 138
 - menus, 22
 - merge attributes of features, 122
 - Merge_Attributes_of_Selected_Features, 122
 - Merge_Selected_Features, 122
 - Metadados, 142
 - modelo de dados vectorial GRASS, 175
 - Movimentar, 115
 - MSSQL Spatial, 75
 - Multi_Banda_Raster, 137
 - multipolygon, 121
 - Nós, 118
 - Natural_Breaks_(Jenks), 83
 - nesting projects, 43
 - New_GPX_Layer, 123, 124
 - New_Shapefile_Layer, 123
 - New_SpatialLite_Layer, 123
 - New_Spatialite_Layer, 123
 - Node_Tool, 117
 - Non_Spatial_Attribute_Tables, 128
 - OGC, 147
 - OGR, 65
 - OGR Simple Feature Library, 65
 - ogr2ogr, 72
 - opções da linha de comandos, 17
 - Open_Geospatial_Consortium, 147
 - OpenStreetMap, 70
 - Oracle Spatial, 75
 - OSM, 70
 - pgsql2shp, 72
 - Picture_database, 641
 - Pirâmides, 142
 - Point_Displacement_Renderer, 85
 - PostGIS, 70
 - PostGIS spatial index, 73
 - PostgreSQL, 70
 - Pretty_Breaks, 83
 - print_composer
 - ferramentas, 629
 - Proj.4, 60
 - Proj4, 59
 - Proj4_texto, 59
 - Projecções, 57
 - Proxy, 149
 - proxy-server, 149
 - QGIS_mapserver, 155
 - QGIS_Server, 156
 - QSpacialite, 75
 - qualidade de renderização, 35
 - Quantis, 83
 - Raster, 135
 - Raster_Calculator, 144
 - Relations, 128
 - Renderer_Categorized, 82

Renderer_Graduated, 83
 Renderer_Point_Displacement, 85
 Renderer_Single_Symbol, 82
 Rendering halting, 34
 Rendering_Mode, 634
 Rendering_Rule-based, 85
 Renderização, 33
 Renderização dependente da escala, 34
 Revert_Layout_Actions, 659
 ring polygons, 121
 Rotate_Point_symbols, 122
 Rotated_North_Arrow, 641
 Rule-based_Rendering, 85

 SC, 57
 SC_Prédefinido, 60
 SC_prédefinido, 57
 Search_Radius, 114
 Secured_OGC_Authentication, 155
 Seleccionar_usando_Consulta, 132
 Serviço de Cobertura Web, 155
 SFS, 147
 Shapefile, 65
 Shapefile_to_Postgis_Import_Tool, 708
 Shared_Polygon_Boundaries, 115
 shp2pgsql, 72
 Simbologia, 89, 137
 Single_Symbol_Renderer, 82
 Sistema_de_Coordenadas, 57
 SLD, 156
 SLD/SE, 156
 Snapping_On_Intersections, 116
 Snapping_Tolerance, 114
 Spatialite, 74
 Spatialite_Manager, 75
 SPIT, 708
 Split_Features, 122
 SQLite, 74
 SRC, 151
 ST_Shift_Longitude, 73

 teclas de direcções para mover, 32
 Tiger_Format, 68
 Toggle Editing, 116
 Topological_Editing, 115
 Três_Bandas_Cor_Raster, 137
 Transparência, 141

 UK_National_Transfer_Format, 68
 US_Census_Bureau, 68

 Vértice, 118
 Vértices, 118
 visibilidade da camada, 29

 WCS, 147, 155
 WFS, 147, 155
 WFS-T, 155
 WFS_Transactional, 155

 WKT, 57, 118
 WMS, 147
 WMS-C, 152
 WMS_1.3.0, 155
 WMS_client, 147
 WMS_identify, 153
 WMS_layer_transparency, 151
 WMS_metadata, 153
 WMS_properties, 153
 WMS_tiles, 152
 WMTS, 152
 WMTS_client, 147
 Work_with_Attribute_Table, 124

 Zoom_In Zoom_Out, 115