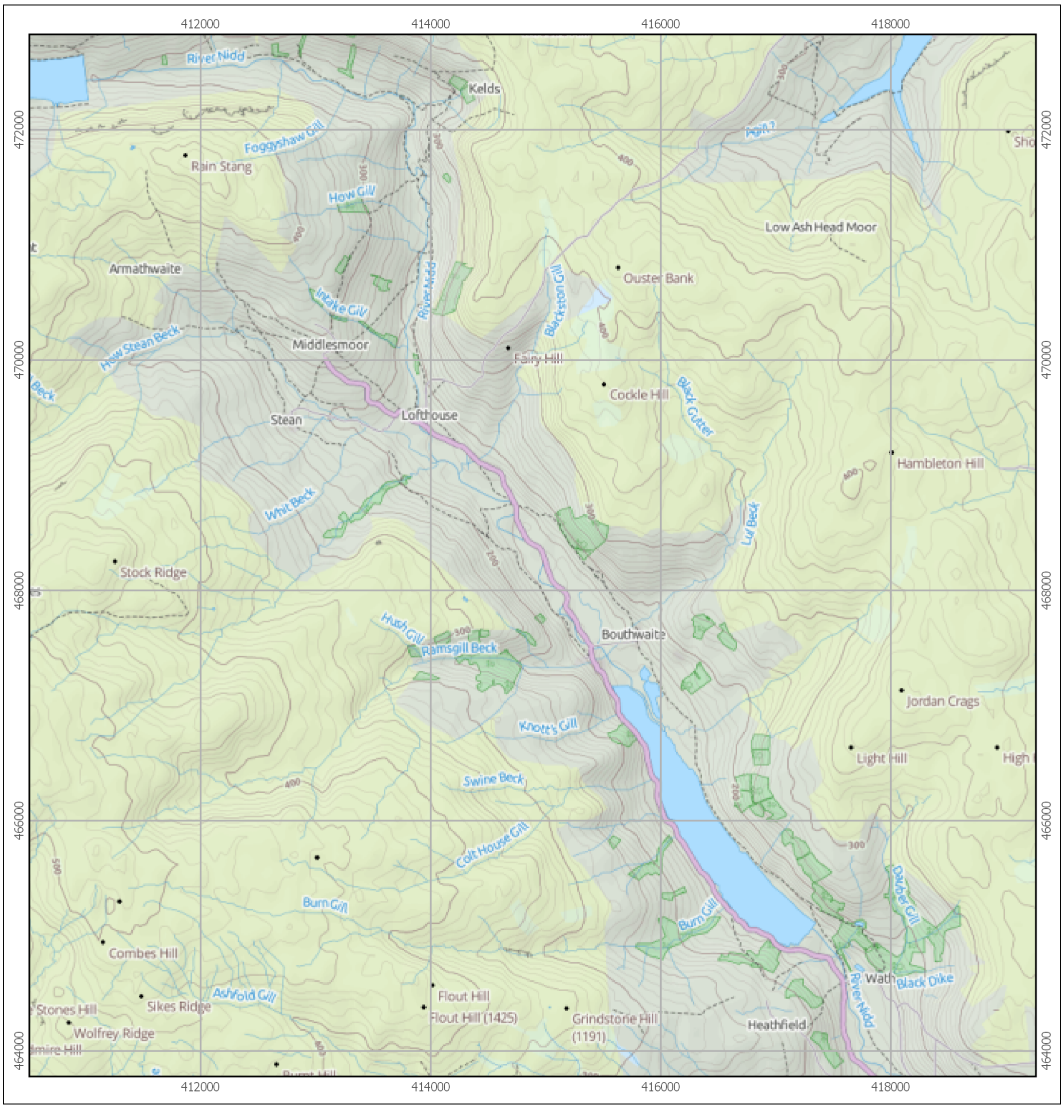
Open Source GIS

Freely available data and software



© Clare Gordon
School of Earth & Environment,
University of Leeds

18th March 2020

QGIS Development Team, 2020. QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org

Font: Clear Sans

Contents

Chapter 1

Open Source GIS tools

Contents

These are likely to stay as nothing more than rough notes but should give me somewhere to keep a record of useful processes.

1.1 OSGeo4W

A project that aims to make a wide variety of open source GIS software available. Look at

`http://www.osgeo.org/`

for a fuller list of open-source GIS tools than I can include here.

Installer provided to download and install any combination of packages (Windows only?) including QGIS and Grass (see below).

`http://trac.osgeo.org/osgeo4w/`

Useful if you want a variety of open source gis software and libraries, and makes it easier to keep all of that software up to date.

Once it is installed, as well as various programs such as QGIS you will also have access to the OSGeo4W Shell which allows you to run command line operations via libraries such as GDAL and OGR (section 1.6 on page 3)

1.2 QGIS

`http://www.qgis.org/`

Also available via OSGeo4W (see above). This is probably the open-source GIS software which is most like ArcGIS. It is being actively developed by an enthusiastic group of people and is increasing in usefulness and reliability all the time.

1.2.1 QGIS documentation

Find the documentation for QGIS at

`http://qgis.org/en/docs/index.html`

The documentation does tend to lag slightly behind development, but it is still well worth using. There is an extensive user guide, a QGIS Training Manual, and A gentle introduction to GIS. All are maintained by volunteers, and if you wish to help out there is also the Documentation Guidelines. All are available either online or as a downloadable pdf.

1.2.2 QGIS plugins

One of the strengths of QGIS are the plugins which add extra functionality. Some are installed by default, but I have also found the plugins in the following list useful. If you can't find a way to do something in QGIS it is always worth searching the plugins via Plugins > Manage and install plugins.

- Digitizing tools *Adds extra digitizing tools*
- Qgis2threejs *Turns a map with height data into a 3D view which opens in a web browser*
- QSpatiaLite *For managing SpatiaLite databases within QGIS*
- AutoTrace *Adds the equivalent of Arc's trace tool for digitizing lines and polygons*
- mmqgis *Vector layer operations plugin. Adds some extra capabilities to those provided by fTools and Processing*
- OpenLayers Plugin *Enables you to add background maps from OpenStreetMap, Google Maps, Bing Maps and MapQuest. Very useful…*
- Processing *Adds a similar interface to Arc toolbox to QGIS.*
- Profile tool *Simple profile generation from terrain data*
- qProf *An experimental plugin (as of 4/3/15) but useful because gives more advanced options for profile generation than the Profile tool*
- XyTools *Managing tabular data with x y columns. Can open and export to Excel.*

1.3 Portable GIS

GIS on a USB stick. More information and download at - `http://portablegis.xyz/` (Last accessed: 13th February 2018). Note that it does take quite a while to install and takes about 3 GB of disk space.

PortableGIS includes QGIS, but usually the long-term release version rather than the most up to date. It's still useful though, as you don't need administrative access to the computer you're using to run this.

1.4 Grass

Available for Linux, Windows and Mac.

`http://grass.osgeo.org/`

Also available via OSGeo4W (see above).

Grass tools are available via QGIS, which makes it more intuitive, and gvSIG but there is a Grass GUI front-end which can be used separately.

There is more information and a first-time user tutorial at

`http://grass.osgeo.org/documentation/first-time-users/`

1.5 gvSIG

gvSIG Community Edition homepage: `http://gvsigce.org/`

The community edition has forked from the original version. There appears to be more development in progress in this edition.

Portable installation available so can be run from a usb stick or on a computer to which you don't have admin rights.

1.6 OGR and GDAL

OGR is a suite of conversion tools for vector files. The equivalent for raster is **GDAL**. Both can be used either from the command line or via a gui, such as QGIS.

You may find that if you have installed QGIS you already have GDAL/OGR installed, and that it will be available in your file path. Otherwise try one of the following:

Note that these instructions apply to the Windows operating system. For information about installing on other platforms see `https://trac.osgeo.org/gdal/wiki/DownloadingGdalBinaries`

1.6.1 SDKShell

Based on information from: `http://bit.ly/1y9AI14`

- Download the gdal library (which includes **ogr** and **GDAL**) from `http://www.gisinternals.com/`.
  - You are looking for a binary for **Win32** or **Win64** marked **-stable** (i.e. under Stable Releases)[1]. The most recent versions appear to be at the bottom of the first table on the page.
  - On the following page look for a link marked **Compiled binaries in a single .zip package** which will probably be the first file.
  - Download the file and unzip it to a folder on your drive.
- Look for **SDKShell.bat** in the files that are unzipped, and double-click to start it.

[1] The versions change from time to time so it is difficult to provide a definite file name but in May 2017 the relevant file was `release-1800-x64-gdal-2-1-3-mapserver-7-0-4.zip`

- From the prompt that appears navigate to the folder that your data files are stored in and then type the commands as required.
  - `cd\` = to return to root, e.g. c:
  - `m:` = to change drive, e.g. move from c drive to m drive.
  - `cd gis\foldername\subfolder` = to change folder

1.6.2 OSGeo4W shell

If you have installed OSGeo4W (section 1.1 on page 1) one of the icons that appears on your desktop should be the **OSGeo4W shell**.[2]

- Double-click on the **OSGeo4W shell** shortcut.
- From the prompt that appears navigate to the folder that your data files are stored in and then type the commands as required.

1.7 SpatiaLite

SpatiaLite is a spatial database format based on SQLite[3]. Its big advantage is that all of your data can be held in one file that can easily be passed to someone else.

Databases in general have big advantages over shapefiles. Shapefiles are an historic format that is still limited to short field names, and have other limits that may not be immediately obvious, but which can cause problems when processing large amounts of data.

ArcGIS will support SpatiaLite from version 10.2 [4]. QGIS already supports it.

1.7.1 Obtaining SpatiaLite

If you are happy using command line tools then you can download a command line version of SpatiaLite but there is also a gui[5] version which a lot of people will find easier to use. Note that you will still often need to type in database commands but many tasks can be achieved with menus.

`http://www.gaia-gis.it/gaia-sins/`

If you're on a Windows pc go to the **MS Windows binaries** section towards the bottom of the page and choose either the 32 bit or 64 bit **current stable version** page. From there you'll get a list of 7zip files, download the one which starts with `spatialite_gui-`. This version is more up to date than the one listed in the main page.

Download and unzip the file of your choice using 7zip - it doesn't need to be installed, just double-click on the exe file to run it.[6]

[2]As of September 2016 `.py` options e.g. `gdal_calc`, only seem to work via OSGeo4W shell, and then without the `.py` added. Seem to get a python error each time, but it still seems to work! This may be something to do with the way that python is set up on my computer.

[3]More information about SQLite is available from `http://www.sqlite.org/`

[4]to a limited extent? 22nd March 2016.

[5]GUI = Graphic User Interface, i.e. you use a mouse and have buttons to click!

[6]There are some useful tutorials to get you up and running quickly. There is a quick start guide at `http://bit.ly/1ndXQH6`, then try the more detailed tutorial at `http://bit.ly/1lFbD8m`

1.8 Whitebox Geospatial Analysis Tools

Download from:

`http://www.uoguelph.ca/~hydrogeo/Whitebox/`

Cross-platform, open-source and transparent. Developed by a lecturer at University of Guelph in Canada for use in teaching.

More information available at the url above and on the blog -

`http://whiteboxgeospatial.wordpress.com/`

Whitebox is being actively developed and some very useful analysis tools are being added. Primarily appears to be intended for raster analysis.

Portable installation available so can be run from a usb stick or on a computer to which you don't have admin rights.

1.9 What works best?

Seem to be finding that different processes are best done in different OS Geo software, and I've seen this commented on elsewhere. Explore the tools available!

Chapter 2

QGIS techniques

Contents

Not a step-by-step guide, see the QGIS documentation for that[1]. This section includes some extra techniques.

2.1 QGIS layout tips and tricks

2.1.1 Changing page size

In version 3 onwards this isn't terribly obvious!

To change the paper size, e.g. from A4 to A3

- Right-click on the page `Page Properties`
- This opens the Item properties panel on the right which contains the **Page Size** options.

[1] `http://qgis.org/en/docs/index.html`

2.2 QGIS Processing Toolbox

The QGIS Processing Toolbox contains a wide range of tools from various providers which enable you to do all sorts of GIS tasks.

Open the toolbox by going to Processing > Toolbox

The providers that you have enabled will show in a tree, and should include headings such as **GDAL/OGR**, **QGIS geoalgorithms** and **GRASS GIS**.

At the top of the toolbox is a search facility and this is invaluable as a way to find a tool among the wide variety available.

2.2.1 To enable providers

As well as being able to enable new providers you may find that you need to tell basic ones where the scripts are on your computer. Do this as follows:

- Processing > Options then open out the tree under **Providers**.

You'll probably find several there which aren't in the list in the Toolbox itself. At the minimum you want to have the following available.

- GDAL/OGR
- QGIS geoalgorithms
- GRASS GIS 7 commands

It is unlikely that there will be an issue with how the first two are configured.

For GRASS GIS 7 to work you need to make sure that you've opened **QGIS Desktop *xxx* with GRASS GIS *xxx* to start with**. There will be version numbers where the *xxx* appear. There no longer seems to be the possibility to add paths to the processing options (QGIS 3.6).

2.3 QGIS Atlas Generation

QGIS Atlas generation is the equivalent of data-driven pages in ArcGIS. Both are ways to generate multipage documents covering an area.

2.3.1 Generating a grid for a coverage layer

Both start with an index layer, or **coverage layer**[2] in QGIS. Basically, a grid layer which shows the coverage of the pages.

You can create one manually - and this works best when your maps do not need to cover an area in a regular grid. Draw polygons around the areas to include.

Using the tool in the processing toolbox in QGIS rather than the tool in the vector menu makes it possible to save the output straight to SpatiaLite if required.

To generate a regular grid using QGIS:

[2] Note that this is nothing to do with coverages in Arc!

- Search for Vector Grid in the **Processing** toolbox
- see figure 2.1 - suggested settings are:
  - select an appropriate layer for the extent, or zoom to the required area and select **select extent on canvas** - measurements below are with some overlap and with map units = metres sizes. You may need to change these slightly depending on the size of the map area in your layout.
    * A4 landscape at 1:2 500 approximately 600 X 400 metres
    * A4 landscape at 1:5 000 approximately 1250 X 850 metres
    * A4 landscape at 1:10 000 approximately 2700 x 1800 metres
    * A4 landscape at 1:25 000 approximately 6500 X 4400 metres
    * A4 landscape at 1:50 000 approximately 12000 x 9000 metres
- Output as polygons.
- Add extra information such as page titles by editing the new layer if required.

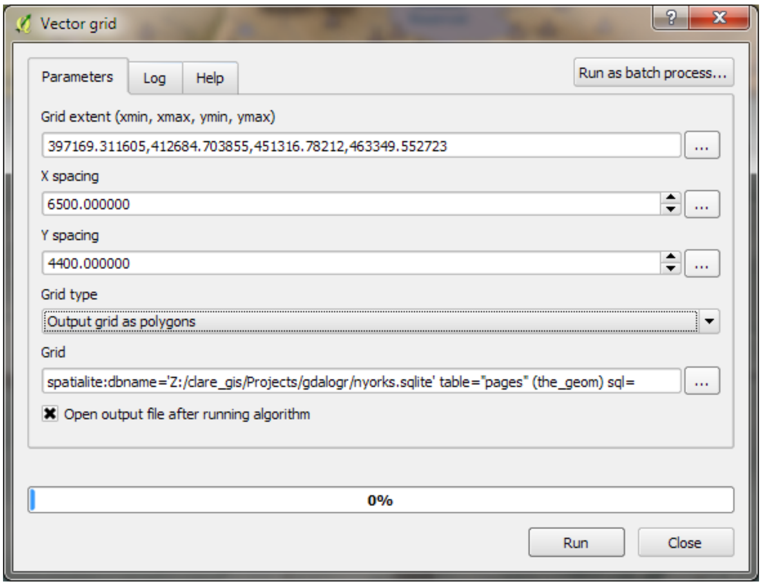


Figure 2.1: Generating a vector grid in qgis using the processing toolbox

2.3.2 Setting up a layout for the atlas

Next set up a layout in Print Composer.

- For this example the scale should be 1:10 000 (for a field slip)
- Under map item properties
  - select Controlled by atlas
  - then tick **Fixed scale**
- On the **Atlas generation** tab
  - Select the correct coverage layer (the one with the index of areas)
  - Select whether you want to hide the coverage layer or not
- On the **Atlas toolbar** at the top of the Print Composer
  - Click to **Preview Atlas**
  - when you are happy with the atlas layout use the toolbar to print or export.

Adding dynamic text

It is possible to add dynamic text, e.g. a title that changes with each page of the atlas.

- Layout > Add label
- in the properties for the label - Insert an expression
- Select from the list of possible items, e.g. if you have a **Title** field in your table - Fields and values > Title, to add a page number look under the **Variables** list and choose atlas_featurenumber

2.3.3 Showing only current atlas feature

Once an atlas has been set up variables are generated which can be used elsewhere in QGIS. To show just the current atlas feature on each page of the atlas try the following: (see figure 2.2 for details)

- In the Style panel for the atlas layer set Rule-based styles.
- Add a rule using the green plus button at the bottom of the panel and edit it so that the Rule reads `@atlas_featureid = $id`
- Style this rule as you wish.
- Add another rule and click on the radio button for **ELSE** - the Catch-all for other features
- Untick the box next to **Symbol** so that there is no style for this rule.

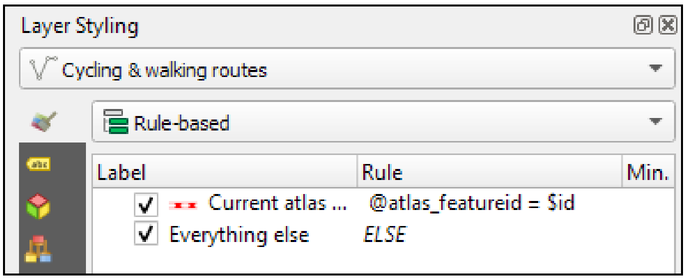


Figure 2.2: Setting styles so that only current atlas feature will display

2.3.4 Adding a different image to each atlas page

It is possible to have a different image on each page of an atlas. You need to have a folder of images alongside the QGIS project. The names of those images should match a field in the table controlling the atlas - it isn't necessary to include the file extension.

- In your QGIS Layout add a picture
- In the **Item Properties** click on the **Data Defined Override** symbol to the right of **Image source** and select Edit...
- In the **Expression String Builder** enter the following code, referencing the folder of your pictures, and the field which contains the name of the files. If it's correct the Output Preview at the bottom should reference one of your images. See figure 2.3.

```
concat('folder/',"name",'_profile.png')
```

- OK

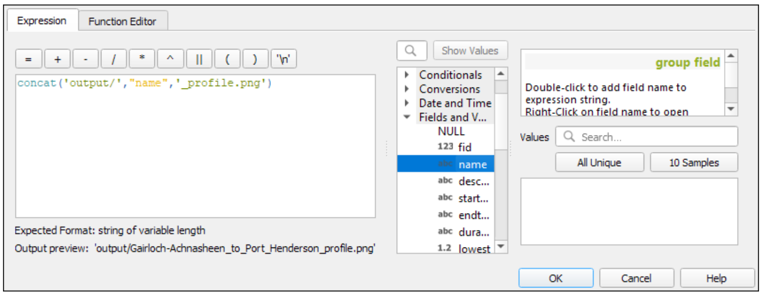


Figure 2.3: Entering the expression to link images from a folder to each atlas page

2.4 QGIS labelling - extra tips

2.4.1 Data-defined labels in QGIS

A technique which allows you to to apply different labels to different classes of features in the same feature class.

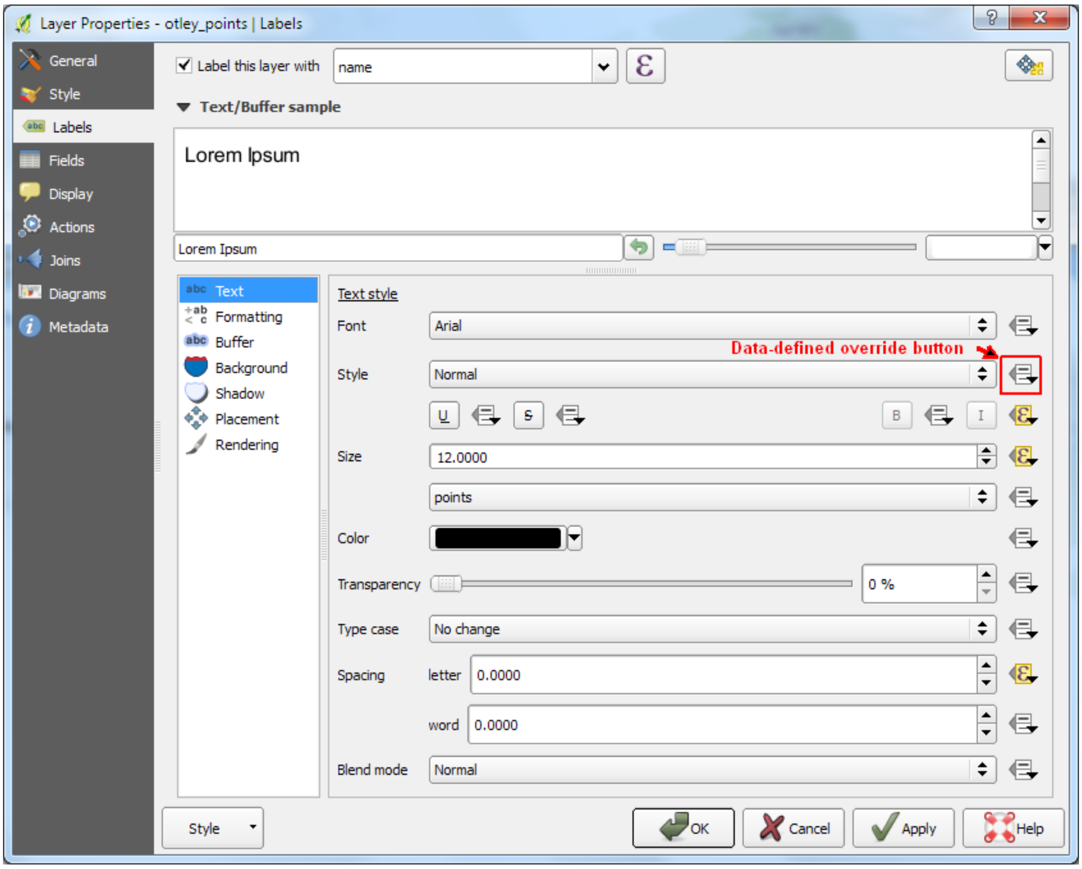


Figure 2.4: Label properties - finding data-defined labelling

- Go to the Labels tab on the Properties dialog for the layer (figure 2.4).
- Click on the Data-defined override button to the right of the field that you want to set - see figure 2.4
- Select to Edit...

- You should end up with the **Expression string builder** - figure 2.5 shows an example that has already been filled in.
- Use the expression builder to enter details of the fields and the labels that you require for each. There are examples of how to use each property at `http://bit.ly/1RJwEQt` - although this is written for an earlier version of QGIS the properties still seem to work in version 2.8.

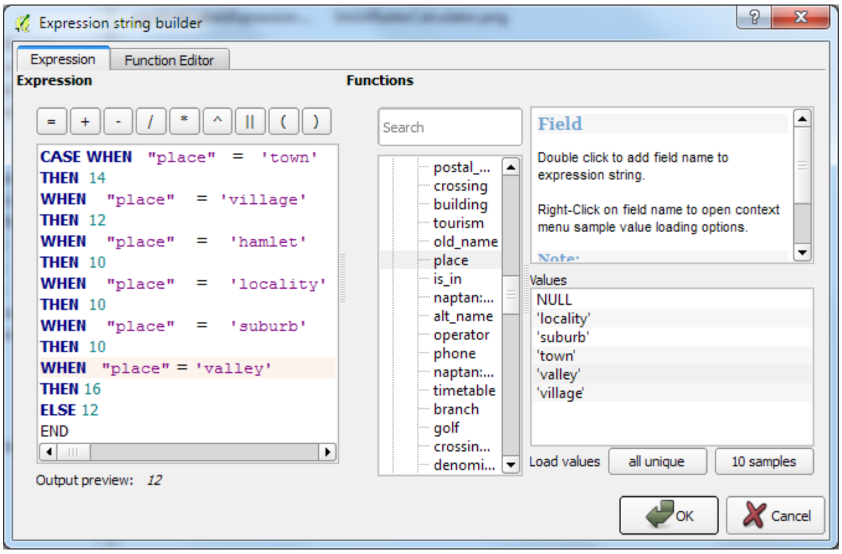


Figure 2.5: The Expression string builder for data-defined labelling

2.4.2 Using regular expressions in QGIS labelling

It is possible to use regular expressions in QGIS expression builders. For example, when using Open Street Map data as in section 6.3.2 on page 45, a lot of information is put straight into a field called `other_tags` so you might end up with fields that hold data such as:

```
"addr:housename"=>"Ledgowan Lodge","source:name"=>"OS_OpenData_StreetView"
```

If you then want to extract just one of the tags such as the house name and use that to label the polygon or point use something like the following expression in the **label with** box:

```
regexp_substr( "other_tags" , 'housename.*([\\w, ]{1,}).*')
```

Basically this means look in `other_tags` and find the first words after `housename` - including upper and lower case characters and digits, and space characters. As the housename is bounded by quotes it shouldn't go any further after that.

it's worth checking this regex more thoroughly!

2.4.3 Changing font size via expressions

As the styles are written for OS VML data the font sizes are very small at 1:10 000 (the scale you probably want to print a field slip). To change font sizes:

- Open the properties for a text layer and go to the **Labels** tab

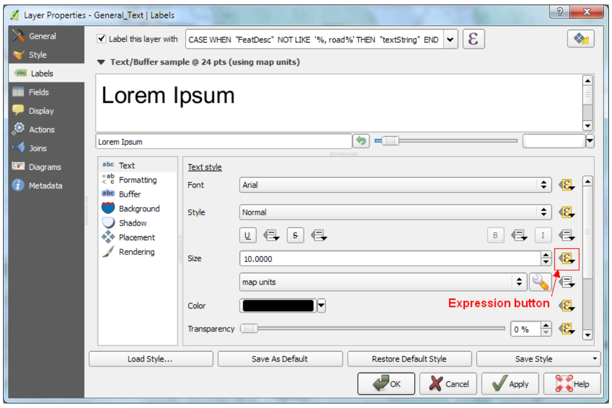


Figure 2.6: The Properties label tab and the expression button

- Click on the expression button next to the **Size** field (figure 2.6)
- and choose to Edit... the expression

The expression will say something like:

```
CASE WHEN "FeatCode" = 15701 THEN "Height" * 1.5 ELSE "Height" END
```

Change it to read something like figure 2.7:

```
CASE WHEN "FeatCode" = 15701 THEN "Height" * 2 ELSE "Height" * 1.5 END
```

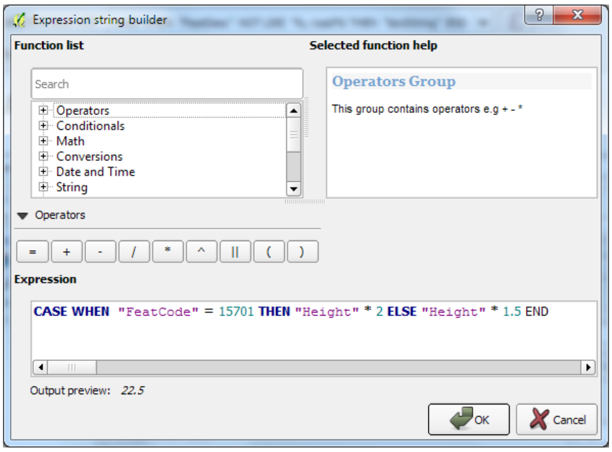


Figure 2.7: The expression editor

- OK 〉 Apply

You can play with the numbers until you are happy with the font size at your scale.

It may then be worth saving it as a new qml file.

2.5 Calculating elevation ascent and descent along a line in QGIS

This is very much note form at the moment, and I'm not entirely sure it's worth doing, but working it out was an interesting and informative exercise! Note that different ways and websites for working out descent and ascent come up with some very different answers for the same route!

I started with some gpx tracks and imported them into Spatialite as lines following the instructions in section 6.3.5 on page 6.3.5 before adding the lines to QGIS.

Your map needs to include a raster height layer.

- Use **v.to.points** from the processing toolbox (GRASS 7) to create points along the line. If you iterate over the input you can get an output file for each line.
- Files won't be in correct srs so (re)project if necessary.
- Run **v.sample** (as a batch if you have multiple lines in the original file)
- The output will be a shapefile - it doesn't seem to work if you try to save directly to csv. Convert to csv either by right-click > Save as... on each layer, or use the **Bulk Vector Export** plugin if you have lots of layers.

This should leave you with a text file with three columns, the fourth of which contains a height value, and lots of lines - one for each point in the original input.

Now you need to run the following Python script:

```
# Input is a text file containing the elevation of a point
# on each line, generated from a line
# Will then output to ascent and total descent along a line

from array import *

last = None
ascent = 0
descent = 0
with open('heights14.csv') as f:
for line in f:
values = line.split(",")
elev = float(values[1])
if last:
change = elev - last
if change > 0:
ascent += change
else:
descent += change
last = elev

print "total ascent (in map units): \%.2f" \% ascent
print "total descent (in map units): \%.2f" \% descent
```

Use a command prompt, and change the name of the input file as appropriate. The on-screen output will be the ascent and descent in map units.

2.5.1 further suggestions

- possible to make it easier to see which output height file matches which original line feature? e.g. if have name of route in original input...
- possible to automate / run python script as a batch?
- could this whole thing be turned into a QGIS processing model?

2.6 Flattening lakes in a DEM

Flattening lakes only in note form - fill out

- SAGA > Vector Polygon Tools > Convert polygon/line vertices to points
- In QGIS 2.18 add new field to resulting table for height then run v.what.rast.points - to add heights from dem to points from previous step
- In QGIS 3 and above run SAGA > Vector Raster > Add raster values to points
- SAGA > Vector Polygon Tools > Point statistics for polygons - output is a polygon shapefile with fields showing **MIN_height** etc
- GDAL > Rasterize (vector to raster) - using Min_height field from above
- Then use command line GDAL for command below

```
gdal_merge -n 0 -o output.tif origdem.tif rasterized.tif
```

May be a way of doing this with the raster calculator in QGIS but command from processing toolbox doesn't seem to work!

Suggestion: May need to create a raster mask from lakes first, then apply numbers from lakes raster just in the masked areas?

2.7 Extruding layers in Qgis2threejs

Qgis2threejs is a really simple way of displaying maps in 3D on screen. The output can be shared with non-gis users as it is simply a web page using javascript.

You need a map which includes a dem layer. You can include other layers such as imagery or hillshade to make the output "pretty".

To create a layer to extrude create a new vector polygon layer (e.g. shapefile or sqlite layer) and add it to your map. Edit the layer and add a polygon or polygons which cover the area you want to show. In the case of "flooding" I created a polygon which covered the whole area of my map.

Symbolise the polygon layer with the appropriate colour and transparency and then untick it in the table of contents so that it isn't visible.

Install and then run the **Qgis2threejs** plugin. The forms to fill in are controlled by the menu on the right.

The **World** tab (figure 2.8) allows you to change the vertical exaggeration - 1 being natural.

The DEM tab (figure 2.9) is where you set the dem layer, but also where you set the resolution of the output and what is shown on the map. In the example it is left as **Map canvas image**, which is the simplest. Play around with the settings to see what helps.

Finally select the polygon layer in the right hand menu (figure 2.10).

- The **Object type** should be **Extruded**
- The **Z coordinate** setting should be **Absolute value** and **0** so that the extrusion is above sea-level. This could be set to *Relative to DEM* so that the extrusion started from the DEM height.
- The **Height** is then set to **Fixed value** and the height above sea-level that you want the "flooding" to be.

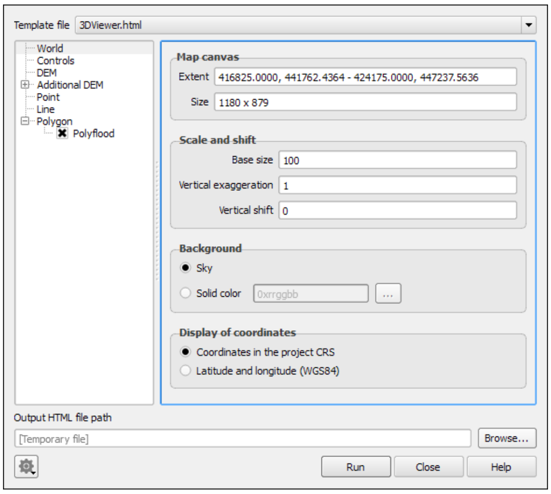


Figure 2.8: The World tab - changing vertical exaggeration

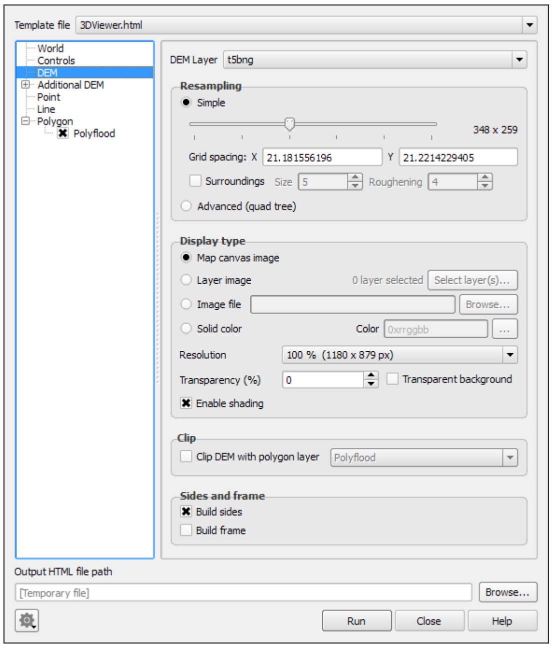


Figure 2.9: The DEM tab - setting the dem and the resolution

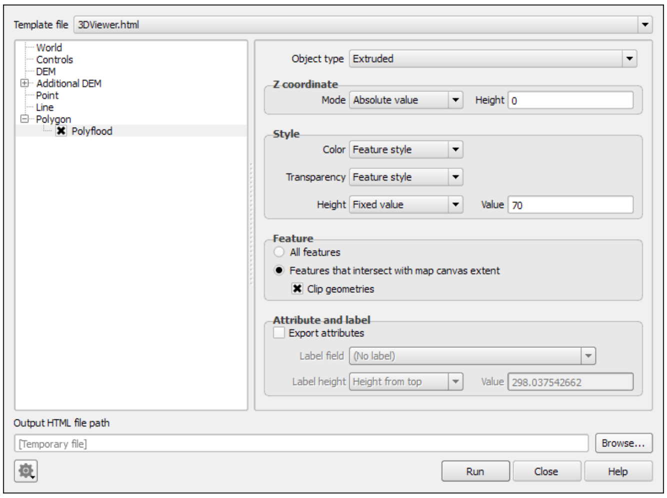


Figure 2.10: Extruding the polygon layer


Figure 2.11: The final output web page from Qgis2threejs

Once these have all been set click to Run. If you haven't entered an **Output HTML file path** the plugin will create a temporary file and open it in your default browser. This is useful as you can try out various options and then save the final run when you know you are happy with the settings. Figure 2.11 shows an example output with flooding at 70m above sea-level in the Wharfe Valley.

If you have a layer which includes heights for building polygons you could use this to extrude buildings.

Chapter 3

Raster analysis techniques

Contents

Many of these techniques involve QGIS, but also other libraries and programs referred to elsewhere in the workbook

Also see techniques under the GDAL library in section 4 on page 22.

These instructions have been followed successfully with both SRTM and OS Terrain 50 DTM data.

3.1 Creating a Raster tile index layer

It can be difficult to keep track of which tiles of raster data cover which part of your map. This simple tool in QGIS creates a vector layer which shows the outline of each tile of data.

- Go to Raster > Miscellaneous > Tile Index
- In the dialog which opens (figure 3.1) you can add the contents of any folder, even including sub-folders (click to **Recurse subdirectories**.

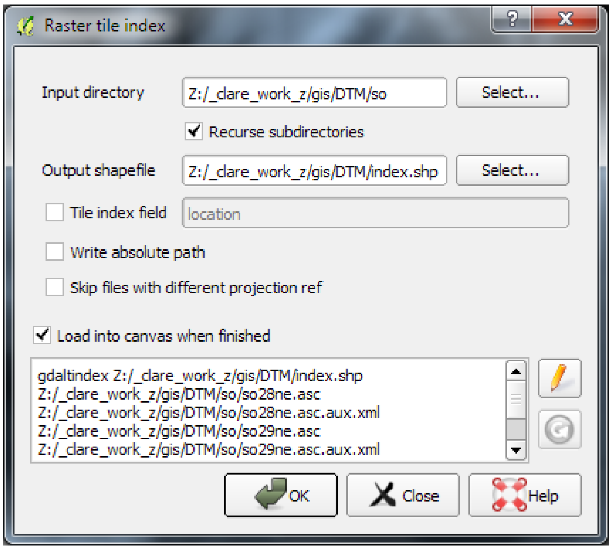


Figure 3.1: The raster tile index dialog box

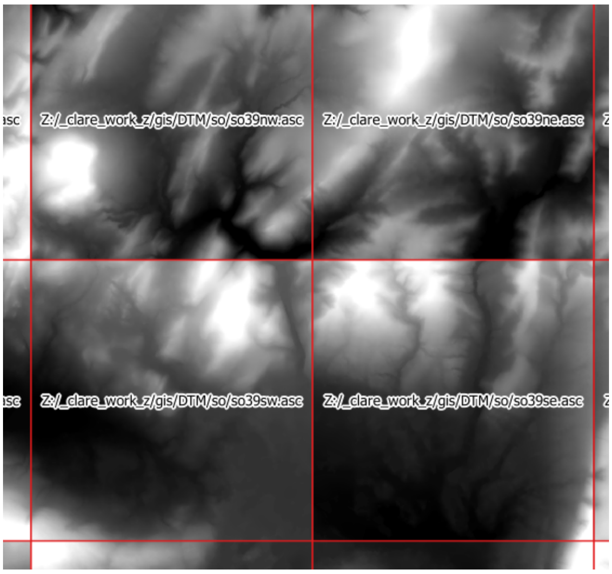


Figure 3.2: The raster tile index symbolised and labelled

Once you have run the tool you should have a shapefile which you can symbolise and label to give you outlined tiles containing the location and name of each tile of data - figure 3.2. Simple but very useful!

It is also worth noting that if you choose the same output location each time, QGIS will add the new tiles to the existing ones. This would be useful if you have multiple datasets in separate file locations and want to keep track of your total holding.

3.2 Merge rasters

3.2.1 Finding file properties

When downloaded SRTM data is unzipped the properties are: 16 bit signed; 1 band; GCS = WGS84.

To find the properties of a file use the Browser (View > Panels > Browser) and right-click on a file to see the properties.

There are various ways to merge raster files.

3.2.2 GDAL merge via Processing Toolbox

To open the processing toolbox: Processing > Toolbox [1]

Then find: GDAL/OGR > GDAL Miscellaneous > Merge raster layers

Select the input files, then select the correct properties, e.g. **16bit** for SRTM files.

Alternatively use:

Raster > Miscellaneous > Merge - this doesn't have as many options, but does seem to work.

3.2.3 Merge rasters with GDAL

Another alternative is to merge rasters using GDAL by following the instructions in section 4.4 on page 24

3.2.4 Merge rasters with GRASS

This is a bit more fiddly but works well. To do it in the Grass GUI use the instructions in section 7.1 on page 61, but Grass tools are also available in the QGIS Processing toolbox.

Processing Toolbox > GRASS commands > Raster > r.patch

It is then possible to create a hillshade layer by doing:

Processing Toolbox > GRASS commands > Raster > r.shaded.relief

3.3 Reproject (warp) raster

Use Raster > Projections > Warp (Reproject)

To keep nodata values as they are select **No Data Values** as **0**.

Alternatively (and maybe more reliably) right-click on the raster layer in the table of contents - Save as… and under **CRS** select the projection that you wish to change the file to. See figure 3.3.

It is also possible to use this form to change some values to NoData. This can be useful for example, to change anything from -10 to 0 to NoData, thereby getting rid of the sea. Though this isn't always necessary - you may actually want them all to be 0 - as section 3.4 below.

3.4 Setting raster values to 0

To make all values below 0 = 0: Raster > Raster Calculator and enter the following statement [2] (merged@1 = name of merged SRTM):

[1] For this the **Processing** plugin needs to be installed and activated.

[2] found on web at `http://www.qgis.org/en/docs/user_manual/working_with_raster/raster_calculator.html` (24th April 2014)

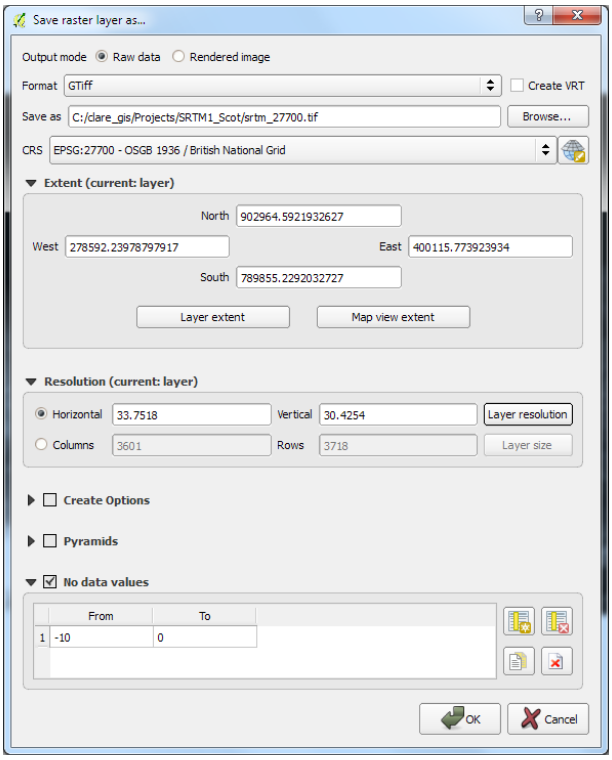


Figure 3.3:
Saving a raster layer as a new file - also changing projection and setting NoData values

```
("merged@1" >= 0) * "merged@1"
```

Alternatively try:

```
("merged@1" <= 0)=0
```

Can make **noData = 0** by using Raster > Conversion > Translate to save as new file and set noData to 0.

3.5 Setting 0 to nodata (null)

Alternatively, if you want to set values to nodata then:

Raster > Conversion > Translate and again set nodata to 0 - it seems to work both ways!

3.6 Filling nodata in a raster with data from another raster

This is covered in the GDAL chapter in section 4.7 on page 27

3.7 Creating raster DEM from vector contours

Based on information at

`http://linfiniti.com/2010/12/3d-visualisation-and-dem-creation-in-qgis-with-the-grass-plugin/`

- Set up a Grass map set and add the vector contour file. This can be a shapefile. Make sure that the layer is visible in the QGIS map.
- Under the **Module** list in the Grass tools dialog find and run these functions in this order
  - **v.to.rast.attr** - converts the vector to a raster based on an attribute. Use the height attribute from the contours. Open your result once it is run.
  - **r.surf.contour** - creates a surface from the rasterized contours. Can take a **very** long time to run.

3.8 "flooding" a raster

3.8.1 Flooding a raster with SAGA raster calculator

In QGIS use the Processing Toolbox (Processing > Toolbox) to go to

SAGA > Grid - Calculus > Raster calculator

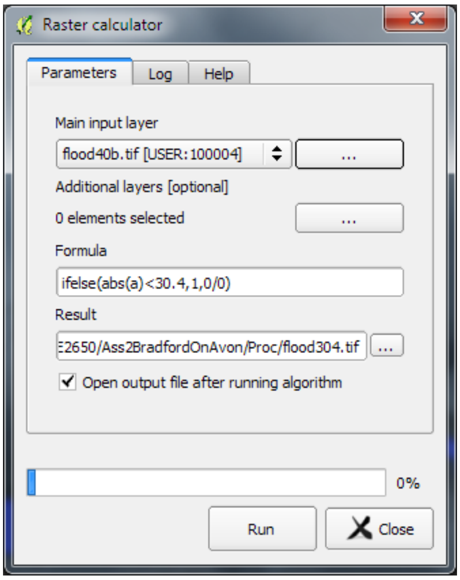


Figure 3.4: "Flooding" a dem with the SAGA raster calculator

Fill in the appropriate sections, choosing the correct raster layer for the input (figure 3.4). In the statement (see below) the `(a)` references the first file in the list of input. If you needed more than one input file you would reference them as `b, c` etc.

```
ifelse(abs(a)<30,1,0/0)
```

Once you have run the tool you can then convert the resulting raster to a polygon by using the tool from the menu in QGIS - Raster > Conversion > Polygonize

3.8.2 Flooding a raster with QGIS raster calculator

Alternatively use the raster calculator in QGIS which is a little more user-friendly!

- Raster > Raster calculator
- Fill in the dialog as shown in figure 3.5 where the name of your raster height data is in quote marks.

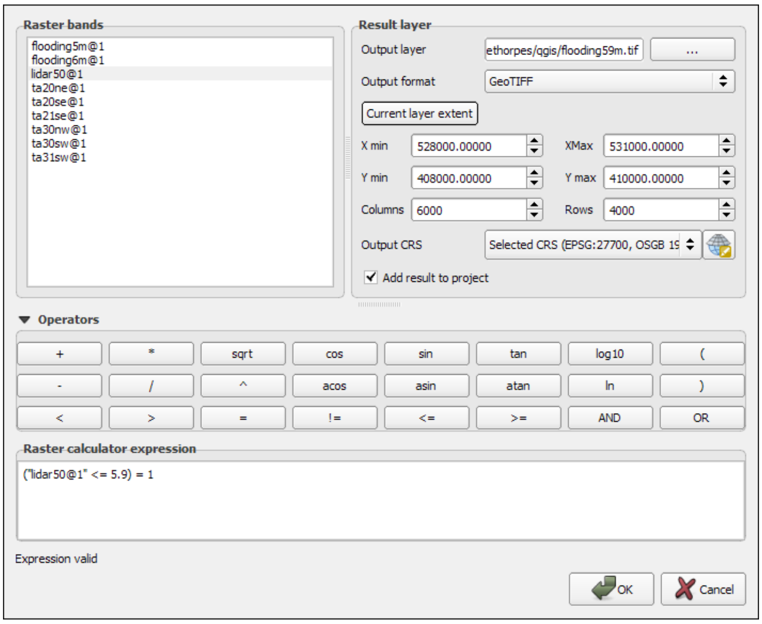


Figure 3.5: “Flooding” a dem with the QGIS raster calculator

```
("dem@1" <= 5.9) = 1
```

The open the properties dialog for the new layer:

- On the **Transparency** tab set the **Additional no data value** to `0`
- On the **Style** tab load max/min values - which should give `1-0` with only 1 showing on your map when you click Apply. You may need to change the color gradient so that it is **White to black**

3.8.3 Flooding a raster with GDAL

This is covered in section 4.7.1 on page 28 in the GDAL chapter.

Chapter 4

The GDAL library

Contents

When working with large datasets, either a single large file, or multiple smaller files, being able to run the processing on the command line is a major benefit. GUI[1] programs are likely to crash, but command line applications such as GDAL and ogr often just keep going quietly until they

[1] Graphical User Interface, e.g. ArcGIS, QGIS, Word, Excel etc

are done. Open Source GIS programes such as QGIS use the GDAL and OGR libraries for processing, but the instructions below show how to run the commands without a gui. In general GDAL works with raster files, and OGR with vector.

See section 1.6 on page 3 for details of how to obtain and run both the ogr and GDAL libraries.

(These instructions assume some knowledge of how to use a command window on Windows. e.g. how to navigate between directories, create new directories etc.)

The GDAL manual pages are an absolutely essential reference! `http://www.gdal.org/` - and particularly the pages at `http://www.gdal.org/gdal_utilities.html`

There is a useful quick reference sheet at `https://github.com/dwtkns/gdal-cheat-sheet`

4.1 Finding out information about rasters with gdalinfo

To find out information about raster files, e.g. coordinate system; type; bands; nodata value etc use `gdalinfo`

Navigate to the folder holding your files then:

```
gdalinfo raster.tif
```

4.2 Batch commands for converting raster files

Example shows how to convert raster files in sub-directories, with resulting output staying in input directory. Note that all batch commands in this document work on Windows and have not been tried on any other operating system.

Enter the following in a text file and save it with a **.bat** extension in the folder that contains either the files to be converted, or sub-folders containing files to be converted.

The command below is to convert asc dtm files to geotiff, but there are a lot of potential format conversions listed at `http://www.gdal.org/formats_list.html`

The page at `http://www.gdal.org/gdal_translate.html` gives information about the options. The command shown also defines the output as British National Grid, but this option can be omitted, or changed to another coordinate reference system.

Note that line breaks can cause issues, so if something isn't working this could be why.

```
for /r %%g
 in (*.asc) do gdal_translate -a_srs epsg:27700 -of GTiff "%%g" "%%~dpng.tif"
```

add `@echo` before `gdal_translate` to see output without actually translating files, i.e. as a check first.

Open SDKShell or OSGeo4W Shell and navigate to the folder containing the batch file. Type in name of batch file and press return. Hopefully it will run, and convert all the appropriate files!

4.3 Converting tiff to GPKG with pyramids

Recommended particularly if using data on mobile devices (e.g. QField). Convert a tiff to a geopackage as follows:

- Note: won't handle float64 files.
- Choice of tile format for Float32 is AUTO, PNG or TIFF
- Change crs as required...
- code should be on a single line

```
gdal_translate --config OGR_SQLITE_SYNCHRONOUS OFF -co  APPEND_SUBDATASET
    =YES -co TILE_FORMAT=TIFF -a_srs EPSG:27700 -of GPKG raster.tif raster.gpkg
```

Then add pyramids

- again, command should be on a single line
- tailor levels to size of raster - but will get a warning, e.g. `Too big overview factor : 128.  Would result in a 9x7 overview`

```
gdaladdo
     --config OGR_SQLITE_SYNCHRONOUS OFF -r AVERAGE raster.gpkg 2 4 8 16 32 64 128 256
```

4.4 Merge rasters with GDAL

There are various ways that gdal can be used to merge rasters[2] but I found that the most reliable on Windows is to merge them as **virtual rasters**.

4.4.1 Creating virtual Rasters

The web page at `http://bit.ly/1SvFIte` suggests using virtual rasters to carry out as much of the processing as possible, only converting to a permanent format such as tif, when the intermediate steps have been carried out. I found that this enabled me to use GDAL to merge a large number of O.S. terrain 50 dtm files together successfully which were causing problems in QGIS and ArcGIS.

- If using Ordnance Survey OpenData Terrain 50 DTM data start by unzipping the main zip file containing data for the whole of Great Britain, then unzip the required zip files in the individual directories.
- For other datasets unzip all files into a folder.
- using SDKShell or OSGeo4W Shell navigate to the folder above the folders containing the `.asc` or `.tif` files[3] and run the following command:

```
dir *.asc /s /b > filenames.txt
```

This should generate a text file containing a list of all of the ascii files within the subfolders.

If you're in the same folder as the files leave out the `/s` to list the asc files in the folder that you're already in.

Now type the following in SDKShell or OSGeo4W Shell:

[2] gdal_merge would be the obvious one but doesn't seem to work on Windows as of March 2016, gdalwarp can also work, but has some issues.

[3] this does mean that you need to make sure that there are no files of the same format in any other subfolders first

```
gdalbuildvrt -input_file_list filenames.txt output.vrt
```

The output should be a `.vrt` file which looks surprisingly small. It links to the original files - so don't delete those or the vrt file will no longer be valid, but this will now load in QGIS if you want to check that it looks OK, and it will allow you to run the following commands.

4.4.2 gdalwarp

Setting coordinate system

To set the coordinate system correctly to epsg:27700 (OS raster files often seem to have a custum version which doesn't appear as epsg:27700 or are `undefined` so setting epsg:27700 explicitly makes things easier later) or to transform it if necessary:

```
gdalwarp -s_srs epsg:27700 -t_srs epsg:27700 -r bilinear -of VRT t50.vrt
t50_bng.vrt
```

`-s_srs` is optional, but if you are transforming the projection does appear to make it a bit more reliable.

Limiting extent of output

It is also possible to limit the extent of the output area at this stage to save processing time and space later. Section 4.6.2 on page 27 suggests ways to retrieve the bounding box for the area you require.

```
gdalwarp -te -4.027863 53.104537 -3.962975 53.13153 -te_srs epsg:4326 -s_srs
epsg:27700 -t_srs epsg:27700 -r cubic -of VRT input.vrt T50bng.vrt
```

4.4.3 Convert virtual rasters to final format

Once you've done all of the necessary processing in GDAL you can convert the vrt files that you want to use in GIS into geotiffs.

```
gdal_translate -of GTiff input.vrt output.tif
```

gdal_translate will allow you to set the datatype if necessary, e.g. Int16, UInt16, Float32 - the full list is at `http://www.gdal.org/gdal_translate.html` - but this isn't usually necessary.

If you want to produce a hillshade you can use the vrt as input, but will have to give tiff as the output.

```
gdaldem hillshade input.vrt hillshade.tif
```

4.5 Creating a gridded raster from a shapefile

Starting with a shapefile which contains points with height data. The output will be an interpolated raster file.

4.5.1 Checking contents of shapefile

Start by checking the contents of the shapefile:

```
ogrinfo input.shp -sql "select * from input where fid IN (1,3)"
```

- Make a note of the extent, which will be shown as `(xmin, ymin) - (xmax, ymax)`
- Also check the coordinate system - `Layer SRS WKT: ...`

4.5.2 Assigning the correct coordinate system

If the coordinate system is not correctly set for the data, e.g. if it is listed as **(unknown)** then you need to set it before you go any further. This is not the point at which you set a new coordinate system if you want your final map to be something different - that will come later. This is just ensuring that the data can be read correctly by the GDAL commands.

```
ogr2ogr -a_srs epsg:5383 output_5383.shp input.shp
```

4.5.3 Running gdal_grid

Full information on this command is at `http://www.gdal.org/gdal_grid.html`

```
gdal_grid -a invdist:power=2.0:smoothing=10.0:nodata=-9999 -ot Float64 -of
GTiff -a_srs epsg:5383 -txe 277000 289000 -tye 5943260 5955260 -outsize 120
120 -l input input.shp output.tif
```

-outsize should be `((xmax-xmin)/output_cellsize) ((ymax-ymin)/output_cellsize)` - where above the output_cellsize is 100x100 m.

smoothing seemed to need to be set fairly high as otherwise the output was chequered. This may be because the input point features were essentially in a grid.

Note that gdal_grid apparently creates an image whose origin is the bottom-left, rather than the usual top-left[4]. As a consequence running further gdal tools on the output may cause errors. If this is the case use the following command to reorientate the output:

```
gdalwarp input.tif output.tif
```

output.tif will now be orientated correctly.

4.6 Clipping a raster

Clipping a raster can save disk space and processing time by limiting the area which contains data.

[4] Found at `http://osgeo-org.1560.x6.nabble.com/gdal-dev-Gdal-Grid-lidar-td5209490.html` - October 2017.

4.6.1 Clipping a raster to a vector polygon

Clipping a raster to a polygon allows the raster to only have values in a particular area. This can be an irregular area, any other area within the bounding box will have a value of **noData**.

```
gdalwarp -cutline polygon.shp -crop_to_cutline input.tif outputclipped.tif
```

4.6.2 Clipping a raster to a bounding box

It is possible to clip a raster to a particular bounding box when using the **gdalwarp** command to reproject (see section 4.4.2 on page 25), but if all you want is to limit the size of the output then this will work too.

If you don't already know the coordinates of the area you require use the Bounding Box tool at `http://boundingbox.klokantech.com/` to retrieve CSV format coordinates for the area you need. These will be in WGS84 (epsg:4326) but can still be used with data in other coordinate systems - in the command below `-projwin` shows the bounding box coordinates and `-projwin_srs epsg:4326` shows the coordinate system, in this case in WGS84. This assumes that the output is in a projected coordinate system such as British National Grid (epsg:27700) rather than WGS84.

Alternatively use the BoundingBox plugin in QGIS[5] to find the coordinates in the coordinate system of a particular project.

For either source the bounding box coordinates need to be changed from `xmin ymin xmax ymax` (which works for gdalwarp) to `xmin ymax xmax ymin`

Does this vary depending on hemisphere?

```
gdal_translate -projwin_srs epsg:4326 -projwin -4.68 57.53 -3.89 57.26
        input.tif -of GTiff output.tif
```

4.7 Filling nodata in a raster with data from another raster

Useful when you have a dem such as lidar which covers a small area and you want to fill in the larger area with heights from another dataset, such as OS Terrain 5.

Based on instructions at `http://bit.ly/1szHmBv`

This assumes that if any tiles of data need to be merged that has already been done.

Start by ensuring that **nodata** is set to the same value in both rasters. Environment Agency lidar is already set to `-9999`

Use `gdalinfo` to check the details for each file. It's worth making a note of the nodata value, the pixel size, and the lower left and upper right coordinates of the file which is the correct size for your final result.

At this stage you can use `gdalwarp` to change the resolution and size of the Terrain 5 tiles at the same time as changing the nodata value.

```
gdalwarp -t_srs epsg:27700 -of GTiff -ot Float32 -te 226000 374000 227000
    375000 -tr 1 1 -r cubic -dstnodata -9999 terrain5.asc t5proj.tif
```

[5] https://github.com/Maaka2890/BoundingBox - June 2017

- `-te` sets the extent of the file - type in the coordinates for the **lower left** and **upper right** corners of the area that you want to output eventually.
- `-tr` sets the pixel size or resolution of the raster to 1 by 1
- `-dstnodata -9999` sets nodata to -9999

To be on the safe side run the same command on both the terrain 5 and the lidar files.

If it is necessary to change the nodata for one of the files without changing anything else it can be done using `gdal_translate`

```
gdal_translate -of GTiff -a_nodata -9999 input.tif output.tif
```

In the following

- rasterA = lidar - the smaller area with gaps
- rasterB = terrain 5 - a larger area with complete coverage

First create a nodata mask equal in size to rasterB - the nodata value in this case is `-9999`

```
gdal_calc.py -A rasterB.tif --outfile=mask.tif --calc="-9999"
```

The resulting layer should be entirely nodata.

Next merge rasterA over this mask.

```
gdal_merge.py -o rasterA_extended.tif mask.tif rasterA.tif
```

The output should be a layer the same size as rasterB but with data from rasterA surrounded by nodata.

Finally use gdal_calc to update the output from rasterA only where there is nodata (below should all be on one line)

```
gdal_calc.py -A rasterA_extended.tif -B rasterB.tif --outfile=result.tif
--calc="A*(A>-9999) + B*(A==-9999)"
```

The final output (result.tif) should be a complete dem.

4.7.1 "Flooding" a raster with gdal_calc

It is also possible to use **gdal_calc** to set areas of a raster to null based on the calculation. For example, to “flood” all areas of a raster below 20 m -

```
gdal_calc -A dem.tif --outfile=under20.tif --calc="A<20" --NoDataValue=0
```

If you then want to convert the result to a vector polygon you can use **gdal_polygonize** -

```
gdal_polygonize under20.tif -f "GPKG" data.gpkg flood20
```

`-f` takes any of the usual ogr formats (`https://www.gdal.org/ogr_formats.html`)

The following calculator statement will give a similar raster but with the depth of flooding

Check that this is what it does really show!

```
gdal_calc -A dem.tif --outfile=under20.tif --calc="A*(A<20)" --NoDataValue=0
```

4.8 Aligning one raster with another

Based on instructions at `http://www.digital-geography.com/topography-bathymetry-gdal-rescue/`

`gdalwarp` can be used to align two rasters from different data sources so that the data from each can be used together. For example, using bathymetry data from ETOPO1 and onshore elevation data from SRTM to produce a single elevation file. ETOPO1 has a lower resolution than SRTM.

Start by using **gdalinfo** to find information about the best resolution file:

```
gdalinfo srtm.tif
```

The result should be something like figure 4.1

```
[redacted]\Bathymetry>gdalinfo joined_dem.tif
Driver: GTiff/GeoTIFF
Files: joined_dem.tif
       joined_dem.tif.aux.xml
Size is 9001, 14401
Coordinate System is:
GEOGCS["WGS 84",
    DATUM["WGS_1984",
        SPHEROID["WGS 84",6378137,298.257223563,
            AUTHORITY["EPSG","7030"]],
        AUTHORITY["EPSG","6326"]],
    PRIMEM["Greenwich",0],
    UNIT["degree",0.0174532925199433],
    AUTHORITY["EPSG","4326"]]
Origin = (-9.000277799999999,59.000138900000003)
Pixel Size = (0.000555555555555556,-0.000277777777777778)
Metadata:
  AREA_OR_POINT=Area
Image Structure Metadata:
  INTERLEAVE=BAND
Corner Coordinates:
Upper Left  (  -9.0002778,  59.0001389) (  9d 0' 1.00"W, 59d 0' 0.50"N)
Lower Left  (  -9.0002778,  54.9998611) (  9d 0' 1.00"W, 54d59'59.50"N)
Upper Right (  -3.9997222,  59.0001389) (  3d59'59.00"W, 59d 0' 0.50"N)
Lower Right (  -3.9997222,  54.9998611) (  3d59'59.00"W, 54d59'59.50"N)
Center      (  -6.5000000,  57.0000000) (  6d30' 0.00"W, 57d 0' 0.00"N)
Band 1 Block=9001x1 Type=Int16, ColorInterp=Gray
  Min=-588.000 Max=1229.000
  Minimum=-588.000, Maximum=1229.000, Mean=53.270, StdDev=197.218
  NoData Value=-32767
  Metadata:
    STATISTICS_MAXIMUM=1229
    STATISTICS_MEAN=53.270323861226
    STATISTICS_MINIMUM=-588
    STATISTICS_STDDEV=197.21843095241
```

Figure 4.1: Output of gdalinfo

Now use the information from that command to put together the following command to change the ETOPO1 data to the same extent and resolution as the SRTM data:

```
gdalwarp -te -9.0002778 54.9998611 -3.9997222
 59.0001389 -tr 0.000555555555556 0.000277777777778 -r cubic etopo1.tif bathymetry.tif
```

- `-te` = specifies the window - or **Corner coordinates** for **Lower Left** and **Upper Right**
- `-tr` = specifies the resolution or **Pixel size**
- `-r` specifies the resampling method, e.g. cubic, bilinear etc.

The output should be a file using the data from ETOPO1 but the extent and pixel size from the SRTM.

It will now be possible to, for example, use the instructions in section 4.7 on page 27 to use the data from ETOPO1 for the bathymetry and the data from SRTM for the land elevation:

```
gdal_calc.py -A bathymetry
 .tif -B topography.tif -\--outfile="joinedDEM.tif" -\--calc="A*(A<=0)+B*(B>0)"
```

4.9 GDAL hillshade and slope

It is also possible to use GDAL to generate a hillshade layer:

```
gdaldem hillshade merge.tif hillshade.tif
```

And similarly generate a slope:

```
gdaldem slope merge.tif slope.tif
```

4.10 GDAL generating a colour relief image

It is possible to use gdal to generate a colour relief image which can be displayed in GIS.

The command uses a colour ramp file to generate the colours and you can either create your own or download one. To create your own see the information at `http://www.gdal.org/gdaldem.html`. Figure 4.3 shows an example for Scotland and surrounding sea.

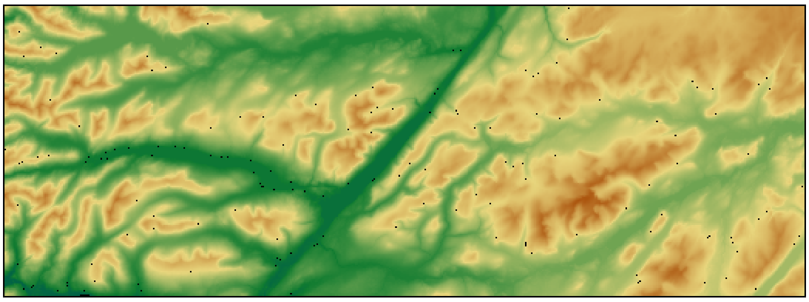


Figure 4.2: A colour relief image generated with gdaldem

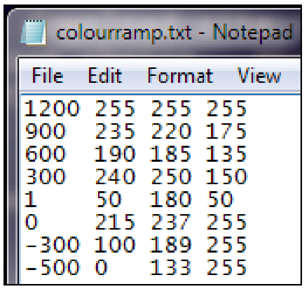


Figure 4.3: Example of a colour ramp text file for use in gdaldem

One possible download site is `http://soliton.vm.bytemark.co.uk/pub/cpt-city/` where there is a vast choice. Most colour ramps have a choice of download formats, try using **.cpt**. The results of some of these are rather odd and you may need to edit the ramp to make it work well for your purposes.

Once you have a colour ramp use the following command:

```
gdaldem color-relief inputdem.tif color-ramp.txt outputdem.tif
```

Once you have a colour relief image and a hillshade image it is possible to merge them into a single output by using a python script which can be downloaded from `http://fwarmerdam.blogspot.com/2010/01/hsvmergepy.html` though apparently greens in the result can be rather bright! I also found that blues could become rather grey (before - figure 4.4; after - figure 4.5).

```
hsv_merge.py col-relief.tif hillshade.tif blended.tif
```

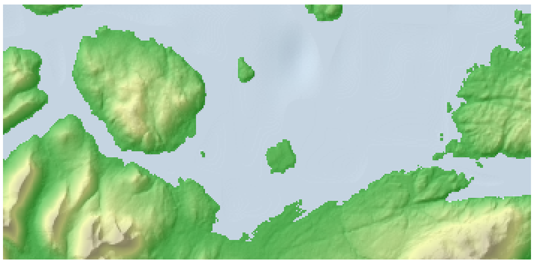


Figure 4.4: Original colour relief and hill-shade

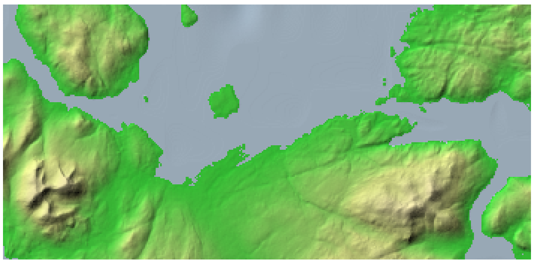


Figure 4.5: Blended colour relief and hill-shade

4.11 Warping (reprojecting) raster files

Example shows a geotiff file, but works with other formats too.

Setting output type to same as original input may be necessary for good quality output - e.g. `-ot UInt16` (Unsigned 16 bit int)

Also might be worth selecting the resampling method as default is nearest neighbour which is probably worst quality. e.g. `-r bilinear`

```
gdalwarp -t_srs EPSG:23029 -ot UInt16 -r bilinear inputraster.tif
outputraster.tif
```

4.12 Smoothing a raster and creating contours

When creating contours it can be useful to "smooth" the DEM first. For example, using Environment Agency lidar DSM trees and buildings create lots of extra contours. If you use gdalwarp to change the resolution first the contours are also smoothed. So this first step is optional, but may help.

```
gdalwarp -tr 10 10 -r cubicspline input.tif output-10.tif
```

`-tr` shows the output resolution. In this case the original resolution (pixel size) was **1,1**, the output was **10,10**.

Now run the command again, this time in reverse, so changing the pixel size back to it's original. Data will have been lost so it will look smoother - see figure 4.6.

```
gdalwarp -tr 1 1 -r cubicspline input-10.tif output-10-1.tif
```

It's also possible to **resize a raster** using a similar command:

```
gdalwarp -ts 1600 0 -r cubic input.tif output.tif
```

`-ts` sets the output size. If either the x or y figure is `0` that will be guessed from the other measurement, so hopefully resizing proportionally.

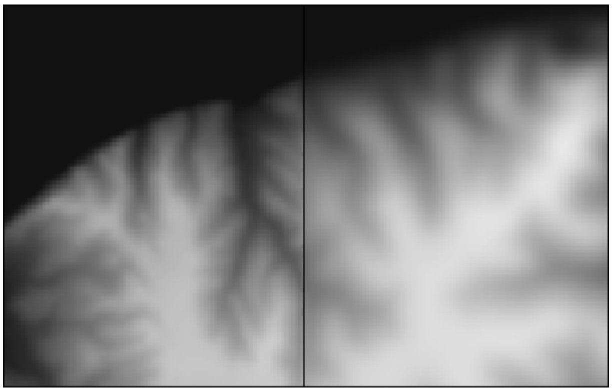


Figure 4.6: A dem before (left) and after (right) smoothing

4.12.1 Generating contours

```
gdal_contour -a height input.tif output_contour25.shp -i 25.0
```

Change `-i` parameter to change contour interval.

This creates a shapefile which can then be imported to spatialite by using the command below. It doesn't appear to be possible to add the output straight to a sqlite database.

```
ogr2ogr -f "SQLite" -append
```

Figure 4.7 shows the difference between contours generated using a smoothed DEM and an un-smoothed DEM.

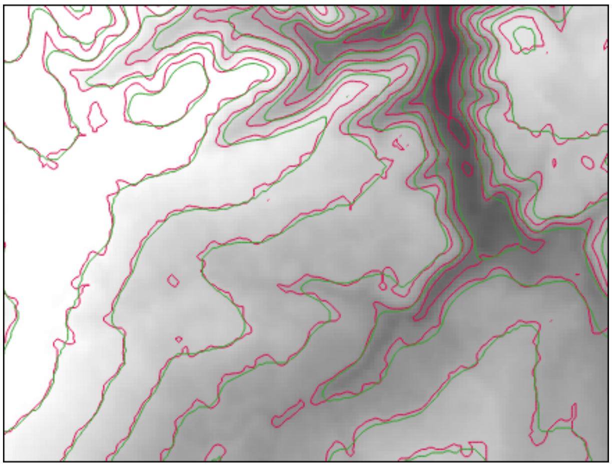


Figure 4.7: Generated contours showing the difference between using the original DEM (red/pink) and the smoothed DEM (green)

4.13 Rasterizing a vector feature class

To turn vector points, lines or polygons into a raster format.

In the example below:

-a requires a numerical class to "burn" into the raster. Oddly this still seems to work if you just pick a random class.

How do you tell gdal_rasterize which field to use to classify raster?

-l gives the layer name within the geopackage, still need this if using shapefiles, but the layer name is most likely the same as the shapefile name.

For extent can use **-te** as `xmin, ymin, xmax, ymax`. If leave blank the full extent of the input will be used.

Then either use

-tr xres yres for resolution or

-ts width height for output size in pixels and lines.

```
gdal_rasterize -a class -l LandCover Data.gpkg Landcover.tif -te 393400 467000
433700 502400 -tr 10 10
```

Chapter 5

The ogr library

Contents

When working with large datasets, either a single large file, or multiple smaller files, being able to run the processing on the command line is a major benefit. GUI[1] programs are likely to crash, but command line applications such as GDAL and ogr often just keep going quietly until they are done. Open Source GIS programes such as QGIS use the GDAL and OGR libraries for processing, but the instructions below show how to run the commands without a gui. In general GDAL works with raster files, and OGR with vector.

See section 1.6 on page 3 for details of how to obtain and run both the ogr and GDAL libraries.

(These instructions assume some knowledge of how to use a command window on Windows. e.g. how to navigate between directories, create new directories etc.)

The GDAL manual pages are an absolutely essential reference! `http://www.gdal.org/` - and particularly the pages at `http://www.gdal.org/ogr_utilities.html`

There is a useful quick reference sheet at `https://github.com/dwtkns/gdal-cheat-sheet`

A lot of the techniques for using ogr with SpatiaLite are covered in chapter 6 on page 42. There is also some information about Geopackages at section 5.7 on page 41.

[1] Graphical User Interface, e.g. ArcGIS, QGIS, Word, Excel etc

5.1 Finding information about vector files

At it's simplest:

```
ogrinfo input.shp
```

For a shapefile this just gives very basic information, e.g. that it holds polygons! Try the following suggestions for more information [2]. This output includes the details of each record, but includes the feature count and projection.

```
ogrinfo input.shp -sql "SELECT * FROM input"
```

Alternatively (and even more simply) try

```
ogrinfo input.shp -al
```

If you have a lot of data the output is rather alarming, though, as it gives you the contents of all records. You could try:

```
ogrinfo input.shp -sql "SELECT * FROM input WHERE fid IN(1,3)"
```

That will return the first and third features and also works with the `-al` switch. Or try the following:

```
ogrinfo input.shp -sql "SELECT * FROM input" -fid 1
```

If you actually want the full information try directing the output to file, otherwise it will probably be unreadable.

```
ogrinfo input.gpx tracks > output.txt
```

If looking for information about a format which includes multiple layers, e.g. Spatialite, gpx, gml, try the following (example is a gpx file which includes a layer called **tracks** - discovered by using `ogrinfo file.gpx`

```
ogrinfo -so input.gpx tracks
```

`-so` means **summary only** and prevents ogrinfo outputting the full contents of the layer but still gives the basic information. Try the following if you just want a summary:

```
ogrinfo -so input.shp -al
```

Using the `-q` flag means **quiet** and allows you to get the contents of the layer without the information about feature numbers, projection etc.

```
ogrinfo -q input.shp -al
```

[2] Suggestions found at `http://www.sarasafavi.com/intro-to-ogr-part-i-exploring-data.html` - 5th May 2016

5.2 Converting vector files

The main “work horse” for vector file conversions and processing is the ogr2ogr command.

5.2.1 Converting gpx to shapefiles

The file format downloaded from a gps unit, or from a route planner on the web, is usually gpx. Gpx files will display in GIS applications, but usually need to be converted to a GIS specific format to allow further manipulation or analysis.

gpx files contain multiple types of data (tracks, routes, waypoints) which can be extracted individually.

Use `ogrinfo` (section 5.1 on page 35) to check the names of the layers. Not all will have content but they will be shown anyway.

The example below shows track information being extracted to a single shapefile.

```
ogr2ogr -f "ESRI Shapefile" output.shp input.gpx tracks
```

5.2.2 Converting csv to shapefiles

This assumes that you have a csv file which contains **x** and **y** data in two columns, e.g. the example below where the coordinate system is British National Grid.

```
point,lat,long,poi,onmap
1,226320,375186,Coastguard Hut,yes
2,226050,375300,Lost hammer,yes
3,226290,375270,Picnic bench,yes
4,226340,375320,Pillar,yes
5,226380,375130,Bird’s nest,yes
6,226480,375080,Lost mobile phone,yes
7,226485,375190,Cairn,no
8,225950,375436,St Gwenfaen’s Well,yes
```

Start by creating a **VRT** file with the same basic name as the csv file, e.g. if the file above is `poi.csv` the vrt file should be `poi.vrt`. Edit the vrt file in a text editor and add the following text, but altered to suit the input csv.

```
<OGRVRTDataSource>
<OGRVRTLayer name="poi">
   <SrcDataSource>poi.csv</SrcDataSource>
   <GeometryType>wkbPoint</GeometryType>
   <LayerSRS>EPSG:27700</LayerSRS>
   <GeometryField encoding="PointFromColumns" x="lat" y="long"/>
</OGRVRTLayer>
</OGRVRTDataSource>
```

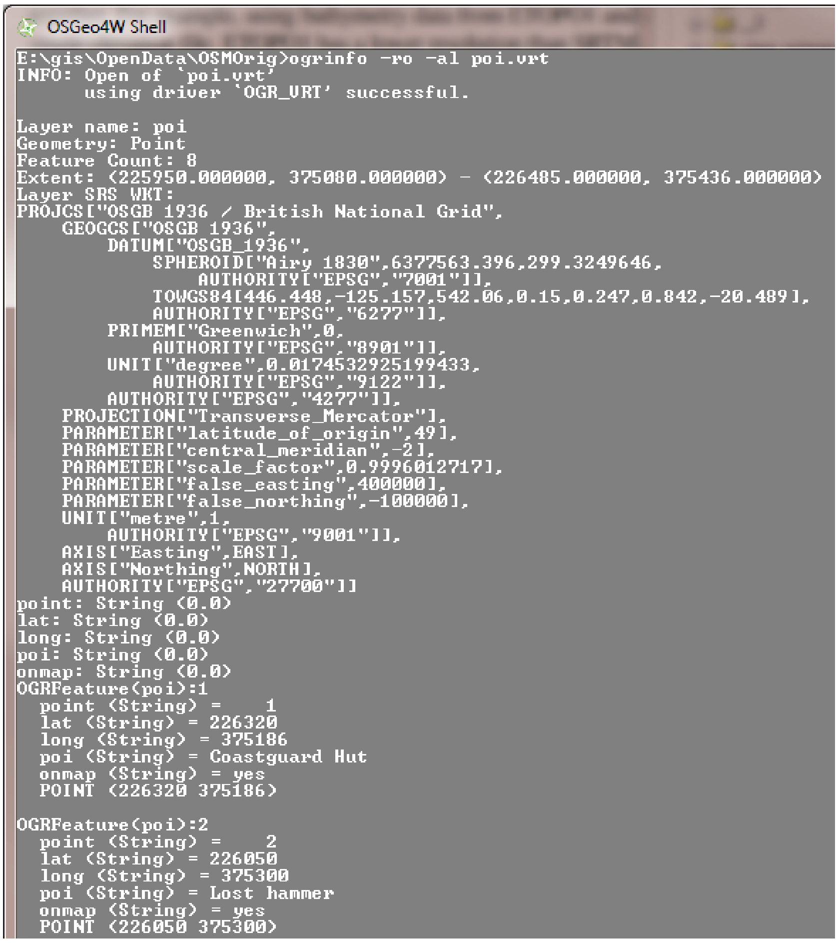
Run **ogrinfo** to check that the formats are correct:

```
ogrinfo -ro -al poi.vrt
```

The output should be a list of the contents of the csv file with a geometry type field for each, similar to that shown in figure 5.1 which shows a field called **POINT** and a location for each feature.

If this looks satisfactory run the following command to put the contents of the csv file into a new output file:

```
ogr2ogr -f "ESRI Shapefile" -t_srs epsg:27700 output.shp poi.vrt
```



```
OSGeo4W Shell
E:\gis\OpenData\OSMOrig>ogrinfo -ro -al poi.vrt
INFO: Open of `poi.vrt'
      using driver `OGR_VRT' successful.

Layer name: poi
Geometry: Point
Feature Count: 8
Extent: (225950.000000, 375080.000000) - (226485.000000, 375436.000000)
Layer SRS WKT:
PROJCS["OSGB 1936 / British National Grid",
    GEOGCS["OSGB 1936",
        DATUM["OSGB_1936",
            SPHEROID["Airy 1830",6377563.396,299.3249646,
                AUTHORITY["EPSG","7001"]],
            TOWGS84[446.448,-125.157,542.06,0.15,0.247,0.842,-20.489],
            AUTHORITY["EPSG","6277"]],
        PRIMEM["Greenwich",0,
            AUTHORITY["EPSG","8901"]],
        UNIT["degree",0.0174532925199433,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","4277"]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["latitude_of_origin",49],
    PARAMETER["central_meridian",-2],
    PARAMETER["scale_factor",0.9996012717],
    PARAMETER["false_easting",400000],
    PARAMETER["false_northing",-100000],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    AXIS["Easting",EAST],
    AXIS["Northing",NORTH],
    AUTHORITY["EPSG","27700"]]
point: String (0.0)
lat: String (0.0)
long: String (0.0)
poi: String (0.0)
onmap: String (0.0)
OGRFeature(poi):1
  point (String) =    1
  lat (String) = 226320
  long (String) = 375186
  poi (String) = Coastguard Hut
  onmap (String) = yes
  POINT (226320 375186)

OGRFeature(poi):2
  point (String) =    2
  lat (String) = 226050
  long (String) = 375300
  poi (String) = Lost hammer
  onmap (String) = yes
  POINT (226050 375300)
```

Figure 5.1: The output from ogrinfo for the example vrt file above

5.2.3 Converting gml to shapefiles

This batch method works for converting OS gml files to shapefiles, but the output will be a list of folders with the same name as the shapefiles. If you then want to merge the shapefiles you first have to rename the folders to remove the point. Try the **Bulk Rename Utility** from `http://www.bulkrenameutility.co.uk/`. It is possible to download a version which doesn't need installing and will run off a usb device.

To convert individual layers from an Ordnance Survey gml into shapefiles use the following commands:

To find out the names of layers in the gml file

```
ogrinfo input.gml
```

Then convert the layer that you want:

```
ogr2ogr -f "ESRI Shapefile" output.shp input.gml LayerName
```

5.2.4 Handling geometry types

Sometimes it appears that geometry types don't always convert from one file format to another as expected and you end up with an error message something like:

```
ERROR 1:  Cannot insert feature with geometry of type POLYGON in column GEOMETRY.
Type MULTIPOLYGON expected
```

If this happens try adding `-nlt POLYGON` to the ogr2ogr command, e.g.

```
ogr2ogr -f "SQLite" -dsco SPATIALITE=YES output.sqlite input.gdb inputvector
                              -nlt POLYGON
```

The list of possible geometry types is available on the ogr2ogr manual page at http://www.gdal.org/ogr2ogr.html

5.3 Merging vector files and folders

5.3.1 Batch commands for merging vector files - shapefiles

These instructions assume that you are using the Windows operating system.

If each shapefile has slightly different name, such as the OS VectorMap Local shapefile download from Digimap, use the following and change file names appropriately.

It is possible to keep or convert the coordinate system by adding `-t_srs EPSG:27700` - the EPSG code can be different from the original, though I'm not sure how reliable this is.[3]

Method found in comments at

http://darrencope.com/2010/05/18/merge-subdirectories-of-shapefiles-using-ogr/

Uses ogr2ogr (via SDKShell - url in section 1.6 on page 3 or OSGeo4W shell). Works if subfolders each contain shapefiles which have same name in each shapefile. e.g. Spanish BTN vector files downloaded from

http://centrodedescargas.cnig.es/CentroDescargas/index.jsp

- Start SDKShell or OSGeo4W Shell and navigate to folder above location of files
- `mkdir merged`
- set up 'bat' file containing the following set of statements for each category of shapefiles that you have unzipped, e.g. Building, Building_Line, Building_Text etc.

(doubled percentage symbols to escape)

```
for /R %%f in (*Building.shp) do (
if not exist merged\Building.shp (
ogr2ogr -f "esri
 shapefile" -s_srs EPSG:27700 -t_srs EPSG:27700 merged\Building.shp %%f) else
(
ogr2ogr -f "esri shapefile" -s_srs EPSG:27700
  -t_srs EPSG:27700 -update -append merged\Building.shp %%f -nln Building )
)
```

[3] Find the code you need at http://epsg.io/ if necessary. 27700 is British National Grid.

5.3.2 Merging subfolders with ogr2ogr

Uses OGR2OGR (via SDKShell[4] or OSGeo4W Shell[5]). Works if subfolders each contain shapefiles which have same name in each shapefile. e.g. Spanish BTN vector files downloaded from

`http://centrodedescargas.cnig.es/CentroDescargas/index.jsp`

BUT depends on choosing a subfolder which has example of every shapefile included in other folders. Not always possible...

- Start SDKShell or OSGeo4W Shell and Navigate to folder above location of files
- `mkdir merged`
- set up 'bat' file containing the following

(doubled percentage symbols to escape)

```
for %%I in (C:\gis\BTN25_1181\*.shp) do (
for /R %%f in (%%~nxI) do (
if not exist C:\gis\merged\%%~nxI (
ogr2ogr -f "esri shapefile" C:\gis\merged\%%~nxI %%f
) else (
ogr2ogr -f "esri shapefile" -update -append C:\gis\merged\%%~nxI %%f
)
)
)
```

Method found at

`http://darrencope.com/2010/05/18/merge-subdirectories-of-shapefiles-using-ogr/`

- change folder path as appropriate.
- File folder on first line should be a folder which contains examples of every individual naming of shapefile.
- run in ogr2ogr command prompt

Merging OS Terrain 5 contour files

On the other hand I found the following worked for OS Terrain 5 contour shapefiles downloaded from Digimap, and for VectorMap District shapefiles downloaded from OS OpenData when I couldn't get the previous code to work. (The new files don't necessarily have to go into a new folder...)

```
mkdir merged
for /R %%f in (*line*.shp) do (
if not exist merged\contours.shp (
ogr2ogr -f "esri shapefile" merged\contours.shp %%f) else (
ogr2ogr
  -f "esri shapefile" -update -append merged\contours.shp %%f -nln contours )
)
```

[4] url in section 1.6 on page 3
[5] url in section 1.1 on page 1

5.4 Clipping the output extent with ogr2ogr

It is possible to use ogr2ogr to restrict the extent of the data that you export or convert. This can save disk space and processing time and is well worth doing if you have a dataset which is larger than you need.

5.4.1 Clipping vector data to a polygon

Clipping to an existing polygon will allow you to save irregular areas. The clipping and the input files need to be the same coordinate system.

```
ogr2ogr -clipsrc clipping_polygon.shp output.shp input.shp
```

5.4.2 Clipping vector data to a bounding box

If you don't already know the coordinates of the area you require use the Bounding Box tool at `http://boundingbox.klokantech.com/` to retrieve CSV format coordinates for the area you need. These will be in WGS84 (epsg:4326).

Alternatively use the BoundingBox plugin in QGIS[6] to find the coordinates in the coordinate system of a particular project..

```
ogr2ogr -clipsrc xmin ymin xmax ymax output.shp input.shp
```

`-clipsrc` shows that the x and y coordinates are in the same coordinate system as the source data. It is also possible to use `-clipdst` to show that the coordinates are the same as the destination data if you are using the same ogr2ogr command to reproject.

5.5 Extracting specific data from vector files

Assume that you have a shapefile containing a range of data, such as woodland, grassland, built-up and mud. If you want to extract only the features which apply to grassland and create a new shapefile, it is possible to do that using ogr2ogr.

```
ogr2ogr -where "type = 'grassland'" grassland.shp input.shp
```

If you are used to using sql queries then you can run more complex operations such as

```
ogr2ogr -where "type = 'grassland' and name like '%meadow%'" grassland.shp
input.shp
```

[6]https://github.com/Maaka2890/BoundingBox - June 2017

5.6 Reprojecting vector files with ogr

`ogr2ogr` can also be used to reproject vector files as well as convert them. Note that output file name comes before input file name.

```
ogr2ogr -t_srs EPSG:27700 output.shp input.shp
```

The source srs should usually be found automatically, but you can also set that:

```
ogr2ogr -s_srs EPSG:4326 -t_srs EPSG:27700 output.shp input.shp
```

Alternatively the following will *assign* the resulting srs to the output without reprojecting the data.

```
ogr2ogr -a_srs EPSG:27700 output.shp input.shp
```

5.7 Geopackages

For full information on the specification see `http://www.geopackage.org/`
Ogr2ogr commands are similar to those used for SpatiaLite but with slight differences.
To add data to a new geopackage

```
ogr2ogr -f "gpkg" output.gpkg input.osm
```

As with sqlite it is also possible to transform the output coordinate system in the same command

```
ogr2ogr -f "gpkg" -t_srs epsg:27700 output.gpkg input.osm
```

To add to an existing geopackage

```
ogr2ogr -f "gpkg" -update output.gpkg input.osm
```

To add a directory of shapefiles

```
ogr2ogr -f "gpkg" output.gpkg ./path/to/dir
```

5.7.1 Using spatialite with geopackages

To check the version of spatialite available in ogr try running the following. Not all spatialite commands are available in geopackages, but if no version is returned, then presumably none of them are.

```
ogrinfo geopackage.gpkg sqlite -sql "SELECT sqlite_version() AS version"
```

To include sql commands and add to a new named layer where the source is different from the destination

```
ogr2ogr -f "gpkg" output.gpkg input.osm -sql "select * from lines where
waterway IN ('riverbank')" -nln riverbank"
```

or to include sql commands and add to a new named layer where the source and destination database are the same. This successfully keeps the geometry of the original layer.

```
ogr2ogr -f "gpkg" Data.gpkg Data.gpkg -append -update -sql "select * from
LandCover where bhab = 'Woodland'" -nln Woodland
```

Chapter 6

SpatiaLite

Contents

Note: November 2017. It seems possible that SpatiaLite is going to be overtaken by Geopackages, which are also based on sqlite. The geopackage spec seems to be in much more active development now. See `http://www.geopackage.org/`. Some of the information below will also work with geopackages. For more information see section 5.7 on page 41.

SpatiaLite is a spatial database format based on SQLite[1]. It's big advantage is that all of your data can be held in one file that can easily be passed to someone else.

Databases in general have big advantages over shapefiles. Shapefiles are an historic format that is still limited to short field names, and have other limits that may not be immediately obvious, but which can cause problems when processing large amounts of data.

ArcGIS will support SpatiaLite from version 10.2 (see section 6.13 on page 58 for instructions on how to create a database in Arc and import feature classes from a file or personal geodatabase). QGIS already supports it.

6.1 Useful references

The SpatiaLite pages are not always easy to navigate, but the home page is at

`http://www.gaia-gis.it/gaia-sins/`

This seems to be the most detailed tutorial -

`https://www.gaia-gis.it/spatialite-2.3.0/spatialite-tutorial-2.3.0.html`

SpatiaLite cookbook:

`http://www.gaia-gis.it/gaia-sins/spatialite-cookbook/index.html#toc`

[1] More information about SQLite is available from `http://www.sqlite.org/`

6.2 Adding data to SpatiaLite with QGIS

By far the simplest way to add vector data to a SpatiaLite database is to add it via QGIS.

- Load any external data into QGIS in the usual way
- Go to Database > DB Manager > DB Manager
- Connect to the appropriate database, or create a new one, and then select it.
- To add layers from QGIS - Table > Import layer/file
- select the layer and fill in the dialog box (figure 6.1)
- click on OK and then click on the **Refresh** button to actually see it.

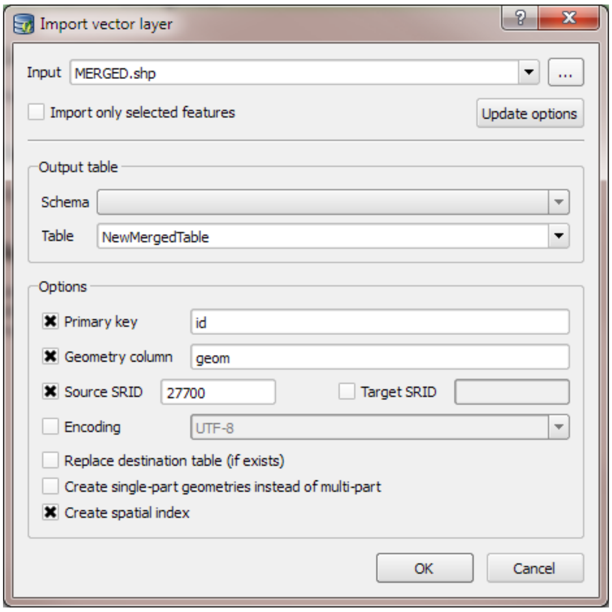


Figure 6.1: Importing layers to a SpatiaLite database from QGIS

It is also possible to add the output from processing tools in QGIS in a similar way. Some of the native tools will only output shapefiles, but if you want to avoid creating these, check the processing toolbox for alternatives, e.g. under GDAL or SAGA. A lot of these tools will allow you to output temporary files and display them in QGIS. Once the output is visible in the table of contents you can follow the steps above to import it to SpatiaLite.

6.3 Importing data to SpatiaLite with ogr2ogr

Most of the methods of importing data listed here use the ogr library. Instructions for obtaining this are in section 1.6 on page 3.

6.3.1 Importing shapefiles to SpatiaLite

It is possible to batch import and merge shapefiles to a SpatiaLite database with ogr2ogr and this basically uses the same code as section 5.3.1 on page 38. This allows you to import shapefiles from multiple subfolders.

```
for /R \%\%f in (*bedrock.shp) do (
   if not exist Geology.sqlite (
   ogr2ogr -f "SQLite" -dsco SPATIALITE=yes -s_srs
     EPSG:27700 -t_srs EPSG:27700 Geology.sqlite \%\%f -nln bedrock ) else (
   ogr2ogr -f "SQLite" -s_srs EPSG
    :27700 -t_srs EPSG:27700 -update -append Geology.sqlite \%\%f -nln bedrock )
   )
```

6.3.2 SpatiaLite, QGIS and Open Street Map osm files

This method works well when you have a `.osm.pbf` file downloaded from Geofabrik - either covering whole countries or regions, or a `.osm` file covering a custom area exported from OpenStreetMap using the Overpass API or using the OSMDownloader plugin in QGIS.

If you follow this method with a large file you can trim the data to the area you need once you have it in a SpatiaLite database - instructions are in section 6.10.8 on page 55.

Based on information from: `http://bit.ly/1y9AI14`

- Download a .osm.pbf file from `http://download.geofabrik.de/`
- **OR** download an area from `http://www.openstreetmap.org`
  - click on Export at the top of the screen
  - choose an area - either the whole screen area or click to **Manually select a different area** and change the size of the bounding box
  - then click the link for **Overpass API**.
  - The downloaded file will just be called **map**. Give it an extension of `.osm`
- **OR** use the **OSMDownloader** plugin within QGIS
- Once you have the data start SDKShell or the OSGeo4W shell (see section 1.6 on page 3).
- From the prompt that appears navigate to the folder that your osm file is stored in and then type in the following command.

```
ogr2ogr -f "SQLite" -dsco SPATIALITE=YES output.sqlite input.osm.pbf
```

You can merge multiple pbf or .osm files by **appending** additional files to an existing sqlite database as follows:

```
ogr2ogr -f "SQLite" -append output.sqlite input.osm.pbf
```

It is also possible to do a reprojection as part of this command (*though do check the reliability of this*):

```
ogr2ogr -f "SQLite" -dsco SPATIALITE=YES output.sqlite input.osm.pbf -t_srs
"epsg:27700"
```

Be patient, as it can take a while to run and may not look as if anything is happening whilst it does. If you look at the target folder you should (after a while) see that the `output.sqlite` file has been created and that there is also an `output.sqlite-journal` file. The journal file is basically a temporary file and will disappear once the full conversion has taken place.

Once the script has finished running the journal file should disappear and you should find that you have an sqlite database which you can open in either SpatiaLite gui or QGIS.

Making river polygons

Both rivers and coastlines are currently part of the lines table. If you wish to add filled in rivers, or show the sea as blue, try processing the layers as follows.

Note that this often crashes for a large dataset. If that is the case try the second method which is slightly more fiddly.

First method

- Open the Processing tools and search for **Polygonize**
- Make sure that you double-click on the QGIS version - NOT the gdal version!
- If you're putting data into a spatialite database you can select that for the output then edit the output file name so that it reads something like the statement below:

```
spatialite:dbname='C:/gis/Shropshire/shropshire.sqlite' table="rivers"
(the_geom) sql=select * from lines where waterway IN ('riverbank')
```

The layer which is added to QGIS might not look quite right. If not, then just remove it and add it again from the spatialite database and it should be OK.

Second method This works if you have your data in a spatialite database already. It can still be very slow on a large dataset, but seems less likely to crash (if you're patient and don't keep clicking on things!)

- Open the DB Manager - Database > DB Manager > DB Manager and select your database
- Open the SQL window by pressing F2
- Type in the following query:

```
CREATE TABLE riverbank AS SELECT * FROM lines WHERE waterway IN ('riverbank')
```

- Run the query - a new table should be added to your database
- Recover geometry -

```
SELECT RecoverGeometryColumn('riverbanks','geometry',27700, 'LINESTRING',2);
```

- Add the new layer to QGIS
- Now, in QGIS, use the **polygonize** tool as shown in the first method, but this time change the output table name, but don't add the sql query to the end of the statement. If you are likely to want to label the output tick to **Keep table structure of line layer**

Once the polygonize tool has run you can remove the riverbanks table from QGIS.

Sea or land areas

To generate sea areas you will need to “complete” the polygon first. So select the coastline lines and save as a new file. Then edit and (using snapping) draw lines out sea-wards and back in to the other end of the coast. You should then be able to use Vector > Geometry Tools > Lines to Polygons to create a polygon for the sea. Or reverse this and create a polygon for the land, if required.

Alternatively (and probably more reliably!) download the coast shapefiles from `http://openstreetmapdata.com/data` . These cover the whole Earth, so you will want to select the area you need.

6.3.3 Importing shapefiles to SpatiaLite with ogr2ogr

Shapefiles each contain a single layer, so the new layer name in SpatiaLite will be the same as the shapefile name.

To import to an existing database:

```
ogr2ogr -f "SQLite" -append -t_srs "epsg:27700" database.sqlite shapefile.shp
```

To create a new database replace `-append` with `-dsco SPATIALITE=YES`

6.3.4 Importing gml files to SpatiaLite with ogr2ogr

Examples were worked using OS VectorMap District gml format data, but this also works for VectorMap Local and OpenMap Local gml files.

The main advantage of gml over shapefile is that field names can be more than 8 characters, so not ending up with some strange truncations.

Importing all contents of gml file

To import all contents of a gml file to a new spatiaLite database:

```
ogr2ogr -f "SQLite" -dsco SPATIALITE=YES -t_srs "espg:27700" vmd.sqlite
SE04.gml
```

If the target database already exists you can import further gml files and append them to the existing layers:

```
ogr2ogr -f "SQLite" -append -t_srs "espg:27700" vmd.sqlite SE06.gml
```

Importing single layers from gml files

First check layer names using `ogrinfo input.gml`

If the target database doesn’t already exist:

```
ogr2ogr -f "SQLite" -dsco SPATIALITE=YES -t_srs "epsg:27700" -nln
SE_RailwayTrack vmd.sqlite SE04.gml RailwayTrack
```

Once the target database has been created:

```
ogr2ogr -f "SQLite" -append -t_srs "epsg:27700" -nln SE_RailwayStation
vmd.sqlite SE04.gml RailwayStation
```

Importing and combining multiple gml files

To add layers from all gml files in a folder into merged classes in a new SpatiaLite database put the following code into a batch file (Windows) and run it using the OSGeo4W shell. All files need to be in the same folder.

```
rem set filepath, database name and required
rem output projection before running batch file

set filepath=E:\gis\vmd\
set database=vmd.sqlite
set coordinates=27700

for %%I in (%filepath%*.gml) do (
for /R %%f in (%%~nxI) do (
   if not exist %filepath%%database% (
   ogr2ogr -f "SQLite" -dsco SPATIALITE=YES
 -s_srs EPSG:%coordinates% -t_srs EPSG:%coordinates% %filepath%%database% %%f
   ) else (
ogr2ogr -f "SQLite" -s_srs EPSG
 :%coordinates% -t_srs EPSG:%coordinates% -append %filepath%%database% %%f
   )
)
)
```

6.3.5 Importing gpx files to SpatiaLite with ogr2ogr

Some of the information in this section was found at `http://bit.ly/25ayhRm` where there are also suggestions for some queries/functions that can be run once the data is in SpatiaLite.

Importing single gpx files

To add the tracks from a single gpx file to a *new* SpatiaLite database and rename the input layer (`-nln`). It's also a good idea to "confirm" the srs:

```
ogr2ogr -f "SQLite" -dsco SPATIALITE=YES -t_srs "EPSG:4326" -nln Layer1
                    output.sqlite input.gpx tracks
```

To add the waypoints from a single gpx file to an *existing* SpatiaLite database and rename the input layer you no longer need the `-dsco` switch, but do need `-append`:

```
ogr2ogr -f "SQLite" -t_srs "EPSG:4326" -nln Layer2 output.sqlite input.gpx
                        -append waypoints
```

Importing multiple gpx files

To add tracks from all gpx files in a folder into a single class in an existing SpatiaLite database put the following code into a batch file (Windows) and run it. The gpx, spatialite database and bat files all need to be in the same folder.

```
for /R %%f in (*.gpx) do (
echo %%~nf
ogr2ogr -append -f "SQLITE" -dsco SPATIALITE=yes
  -t_srs epsg:4326 Yorkshire.sqlite %%~nf.gpx tracks -nln gpxtracks
)
```

6.3.6 Importing csv text file to SpatiaLite

This assumes that you have a csv file which contains **x** and **y** data in two columns, e.g. the example below where the coordinate system is British National Grid.

```
point,lat,long,poi,onmap
1,226320,375186,Coastguard Hut,yes
2,226050,375300,Lost hammer,yes
3,226290,375270,Picnic bench,yes
4,226340,375320,Pillar,yes
5,226380,375130,Bird's nest,yes
6,226480,375080,Lost mobile phone,yes
7,226485,375190,Cairn,no
8,225950,375436,St Gwenfaen's Well,yes
```

Start by creating a **VRT** file with the same basic name as the csv file, e.g. if the file above is `poi.csv` the vrt file should be `poi.vrt`. Edit the vrt file in a text editor and add the following text, but altered to suit the input csv.

```
<OGRVRTDataSource>
<OGRVRTLayer name="poi">
   <SrcDataSource>poi.csv</SrcDataSource>
   <GeometryType>wkbPoint</GeometryType>
   <LayerSRS>EPSG:27700</LayerSRS>
   <GeometryField encoding="PointFromColumns" x="lat" y="long"/>
</OGRVRTLayer>
</OGRVRTDataSource>
```

Run **ogrinfo** to check that the formats are correct:

```
ogrinfo -ro -al poi.vrt
```

The output should be a list of the contents of the csv file with a geometry type field for each. If this looks satisfactory run the following command to put the contents of the csv file into a new spatialite database as a layer:

```
ogr2ogr -f "SQLite" -dsco SPATIALITE=YES -t_srs epsg:27700 output.sqlite
poi.vrt
```

6.4 Using scripts in SpatiaLite

Example SQL script using shapefile downloaded from Geofabrik and imported into SpatiaLite gui. The sql script below has been typed into a text editor and saved as a .sql file. When run it will select data, save selection as a new table, then reproject table. The example used is the Geofabrik download for Greece. The roads shapefile has been imported into SpatiaLite. An **Outline.shp** polygon shapefile was created in QGIS (Srid is the same as OSM - **4326**), an outline of an area was roughly digitised. The shapefile is also imported into SpatiaLite and is used to select data from the full roads table which is used to create a new table before the original one is **dropped**. The resulting, smaller, table is then reprojected to ETRS 1989 UTM Zone 34N.

```
CREATE TABLE LoutrakiRoads AS
SELECT Outline.name AS areaname, roads.*
FROM Outline, roads
WHERE Outline.name = 'Loutraki'
AND Intersects(Outline.Geometry,roads.Geometry);
```

```
select
  RECovergeometrycolumn('LoutrakiRoads', 'Geometry', 4326, 'LINESTRING', 2);

DROP TABLE roads;

VACUUM;

alter table LoutrakiRoads add column geomproj BLOB;

update LoutrakiRoads set geomproj = transform(Geometry,25834);

select
  RECovergeometrycolumn('LoutrakiRoads', 'geomproj', 25834, 'LINESTRING', 2);
```

6.5 Recover Geometry Column

This is an operation that you'll often need to carry out after completing other tasks in SpatiaLite.

6.5.1 Updating Geometry with spatialite-gui

- Open out the list of fields in the table in the tree on the left of the window
- Right-click on the geometry column and Check Geometries - this will let you see what the settings should be (hopefully)
- Right-click on the geometry column again and select Recover geometry column then use the output from the **Check Geometries** command to fill in the dialog box with SRID; Dims (i.e. CoordDimension); and Geometry Type
- Click OK and the geometry should be recovered.

6.5.2 Updating Geometry with the RecoverGeometryColumn command

The geometry type should be set to the same as the original - i.e. POLYGON; POLYLINE; LINESTRING; MULTIPOINT etc. It will fail if it isn't!

To check the original geometry run:

```
SELECT Count(*), GeometryType
 ("geometry"), Srid("geometry"), CoordDimension("geometry")
FROM "origTable"
GROUP BY 2, 3, 4
```

Then use the information from there to populate the

```
SELECT RecoverGeometryColumn('newTable', 'geometry',27700,'MULTIPOLYGON');
```

If the process doesn't seem to be working check that the srid that you require is actually available - see next section...

6.5.3 Search spatial reference systems

A full list of spatial reference systems (Coordinate systems) is available from http://epsg.io/. The srid appears at the top of the information page, e.g. **EPSG:27700**.

To check that the srid is available in SpatiaLite search as follows:

```
SELECT * FROM spatial_ref_sys WHERE Srid = 27700;
```

Another useful field is `ref_sys_name`, e.g.

```
SELECT * FROM spatial_ref_sys WHERE ref_sys_name LIKE '%ETRS89%'
```

If the srid that you require is not available then you can add a record to the table for yourself using the data from http://epsg.io/

6.6 Reprojecting data in SpatiaLite

To reproject the data in the new database tables use the following sql commands: (This example reprojects the data to British National Grid - srid: 27700.) [2]

The first command checks the geometry of the table (note: all field and table names in the command below may be lowercase - check!):

```
SELECT * FROM geom_cols_ref_sys
WHERE f_table_name = 'tableName'
AND f_geometry_column = 'Geometry';
```

Next add a column to hold the new geometry:

```
alter table tableName add column geombng BLOB;
```

Now transform the geometry of the new column:

```
update tableName set geombng = transform(Geometry, 27700);
```

Lastly, set the geometry column so that it is read as such in the database using the instructions in section 6.5.2 on page 50.

6.7 Adding a primary key to a table that doesn't have one

To deal with this: Create new table

If using spatialite-gui can right-click on a table and find CREATE statement

Copy this and paste into sql window

New table MUST have a primary key defined - maybe use CREATE statement from original table that hasn't been intersected.

```
CREATE TABLE "Places_tr2" (
"id" INTEGER PRIMARY KEY,
"geom" POINT,
"gml_id" TEXT,
"distinctiveName" TEXT,
"classification" TEXT,
"fontHeight" TEXT,
"featureCode" INTEGER);
```

[2]Based on information from: http://bit.ly/1ki0fkL

Then copy data from the intersected table to the new one

```
INSERT INTO
  Places_tr(geom, gml_id, distinctiveName, classification, fontHeight, featureCode)
SELECT geom,
  gml_id, distinctiveName, classification, fontHeight, featureCode FROM Places_backup;
```

Then recover geometry field as in section 6.5.2 on page 50.

6.8 Merging two tables without overwriting primary key

If you're not careful it is possible to end up with duplicate primary keys in the merged spatialite tables so it's worth doing it this way

Make a note of the maximum primary key (OGC_FID) from the table you want to merge into as follows.

```
select max(OGC_FID) from table1
```

Then update the table to be merged from so that the primary keys are all above that number

```
update table2 set OGC_FID = OGC_FID + maxkey1 + 1
```

Finally, copy all the features from table 2 into table 1

```
insert into table1 select * from table2
```

6.9 Copying a table from one SpatiaLite database to another

To copy a table from one SpatiaLite database to another it is possible to use the following method as long as both databases have the same encoding.

Using SpatiaLite-gui[3]

- Start with the database you want to move the table **to** connected
- Files > Attach DataBase and select the database which contains the table that you want to copy

The database should appear as a new link in the tree on the left-hand side. Note the letter which appears at the beginning, e.g. **a:**

- Assuming that there isn't already a table that you want to insert the data into in the target database, right-click on the table in the source db that you want to copy and Show CREATE statement
- Copy this statement and paste into the SQL query window at the top - add a semi-colon at the end. It is possible to give it a different name from the original if you want to.
- Then enter the following statement, replacing the targetname, sourcename and updating the field list appropriately (the list should be the same in both tables)

```
INSERT INTO targettable(Id, Value, Content) SELECT * FROM a.sourcetable;
```

Spatialite-gui should create the new table and add the contents of the source table to the target table. You may need to refresh the database to see the new table. SpatiaLite may well not update the geometry (recover the geometry column) for the new table, in which case you'll need to do that manually. See section 6.5.1 on page 50 for a relatively painless way to do that.

[3]details for download in section 1.7.1 on page 4

6.10 Examples of queries

Various examples. Many of these are useful for creating a view containing part of the data from the original table, or for creating a new table.

6.10.1 General sql queries

Count number of records in table which have same content, e.g. this example will collate contents of the place field, and give a list of the number of occurences of each.

```
select place, count(*) as count from points group by place order by place
```

Retrieve records from a table where field NOT an item from a list (i.e. in example col1 will not contain either x or y or z)

```
SELECT * FROM tblname WHERE NOT col1 IN ('x','y','z')
```

6.10.2 Adding x,y columns to a table

It can be useful to be able to show x and y coordinates for features in a table - maybe to export them as csv?

```
select pkuid as id,
name,
cast(X(geometry) as integer) AS easting,
cast(Y(geometry) as integer) AS northing,
geometry as geometry
from pointdata
```

Casting the coordinates as integer removes decimal points from the output.

It is also possible to use the query to transform the coordinates.

Check that this transformation works!

So if you have data in WGS84 it should be possible to output x and y coordinates in British National Grid. So the query above would become:

```
select pkuid as id,
name,
cast(X(transform(geometry, 27700)) as integer) AS easting,
cast(Y(transform(geometry, 27700)) as integer) AS northing,
geometry as geometry
from pointdata
```

6.10.3 Adding new fields and calculating data

To add data to a new field, first create the field:

```
ALTER TABLE pointtable ADD COLUMN easting integer;
ALTER TABLE pointtable ADD COLUMN northing integer;
```

Then calculate the output that you want the field to contain and add it to the field(s):

For example adding *x* and *y* coordinates

```
UPDATE pointtable
SET easting = (CAST(x(geometry) as int)),
SET northing = (CAST(y(geometry) as int));
```

The next example shows how to calculate the length of linestrings in meters (check using cartesian coordinate system) and converting the result to miles in one query:

```
ALTER TABLE linetable ADD COLUMN Length double;
UPDATE linetable
SET Length = CvtToMi(GLength(geometry));
```

6.10.4 Rounding number of decimal places in float fields

The example shows length columns (double format) being rounded from 8 decimal places to 2.

```
UPDATE gpxtracks_round
SET length_km = round(length_km, 2),
length_miles = round(length_miles,2)
```

6.10.5 Extracting part of a text field

Given a field with the following content I want to extract just the numbers for the ascent and descent:

```
Ascent:  378ft; Descent:  400ft.  Some other notes here.
```

The `instr` function allows a substring to be extracted from a string without knowing the length:

```
select name, notes,
substr(notes,
  instr(notes, 'Ascent: ') + 8,
  instr(notes, 'ft;') - 8 - instr(notes, 'Ascent: ')) ascent_ft,
substr(notes,
  instr(notes, 'Descent: ') + 9,
  instr(notes, 'ft.') - 9 - instr(notes, 'Descent: ')) descent_ft
from gpxtracks order by length_km desc
```

6.10.6 Maths within query

The example given here converts a field containing a height in feet into metres and rounds it to two decimal places.

```
select name, ascent_ft, round(ascent_ft / 3.28, 2) as ascent_m from gpxtracks;
```

6.10.7 Comparing two tables in spatialite

To compare two tables and find which rows appear in one but not the other:

```
SELECT DISTINCT field1
FROM table1
WHERE field1 NOT IN
(SELECT DISTINCT field1 FROM table2);
```

Example use: Finding out which new features have been added to an Open Street Map download since your previous download.

Alternatively use the following to see which existing rows have had changes:

```
select a.name, b.name, a.other, b.other
from table1 as a
join table2 as b on
(a.id = b.id)
where a.name != b.name
or a.other != b.other
```

6.10.8 Select features which intersect another layer and create new table

If run in SpatiaLite-gui has the advantage of keeping the new feature class within the existing database. If you only need to work with a small area of a larger dataset it is worth running this before you do anything else - it could save you a lot of time!

```
CREATE TABLE TorrPlaces AS
SELECT AreaOfInterest.name as areaname, places.*
from AreaOfInterest, places
where AreaOfInterest.name = 'Torridon'
AND Intersects(AreaOfInterest.geombng,places.geombng);
```

Then need to rebuild geometry column for new table as in section 6.5.2 on page 50.

Unfortunately this produces a data table which doesn't have a primary key, and this appears to have an odd effect on selections and the identify tool in QGIS. Adding a primary key isn't entirely straightforward, though. See instructions in section 6.7 on page 51.

6.10.9 Vacuuming a database

If you have been moving a lot of data around within a database, or have been deleting/dropping feature classes, spatialite may not have cleared the space and the size of the database may be becoming rather large. To shrink the size back down to the minimum you can **Vacuum** it.

To vacuum a database from within OGR try running the following command:

```
ogrinfo Data.gpkg -sql "vacuum"
```

6.11 SpatiaLite views

A view is a way of selecting data from a table and then being able to add it to a GIS. Data in views is not editable, but if you edit the original table the view will update accordingly.

6.11.1 Creating a view

Build a select query that you are happy with first, and then create a view by using the `CREATE VIEW` command:

```
CREATE VIEW "viewOfTable" AS
SELECT pkuid AS ROWID,
name,
geometry AS geometry
FROM pointtable;
```

Using the `CREATE TABLE` command instead will, as it says, create a new table rather than a view. This would then be editable, but would not update if the original table was changed. You'd probably also need to recover the geometry column too.

OR

In Spatialite gui use the **Query / View composer** to build the view and save it as a **Spatial View**.

6.11.2 Turning a view into a spatial view

If you use the **Query/View composer** in Spatialite gui to compile and create the new view then you can set it to **Create Spatial View** and the geometry columns will be correctly set so that this next step is unnecessary.

To turn a view into a spatial view which can then be displayed in a GIS, you need to register it in the `views_geometry_columns` table as follows:

```
INSERT INTO views_geometry_columns
(view_name, view_geometry, view_rowid, f_table_name, f_geometry_column, read_only)
VALUES
('viewOfTable', 'geometry', 'ROWID', 'pointtable', 'geometry',1);
```

Don't be tempted to set `read_only` as `0` - that can cause problems!

6.12 Adding data to QGIS from a SpatiaLite database

- Layer > Add SpatiaLite layer or use the "feather" icon on the **Manage Layers** toolbar
- You'll need to select the correct database and then click to Connect before you are given a list of layers that you can add to your map.

6.12.1 Applying styles with qml files

Qml files make it possible to set up rules and apply styles and labels to data, including Open Street Map data, in QGIS. There are various examples of styles available on-line to download, but which one works for your data does depend on the original format, e.g. data that was downloaded as shapefiles will have different fields to data downloaded as osm and which has been converted to spatiaLite without ever being a shapefile.

Qml files for Geofabrik data downloaded as shapefiles are available from:

`https://github.com/charleyglynn/OSM-Shapefile-QGIS-stylesheets`

or from:

https://github.com/3liz/osm-in-qgis

or for osm files converted to SpatiaLite:

https://github.com/anitagraser/QGIS-resources

Download the files and save to your local disk.

To use the points styles you will need to have downloaded the svg icons from the url below[4]. Unzip them to the svg folder of your qgis installation which will probably be something like `C:` ▸ `OsGeo4W64` ▸ `apps` ▸ `qgis` ▸ `svg` or `...` ▸ `apps` ▸ `qgis` ▸ `svg`. You will then be able to select these as svg markers in the symbol manager as well as correctly use the points style file.

http://www.mediafire.com/download/344x43313q3xx8n/osm_icons4qgis_v1.zip

To style a layer:

- Open the layer properties and go to the **Style** tab.
- Click on Load Style... and select the appropriate qml file then click to Open and then Apply

The styles and labelling from the qml should be applied to your layer.

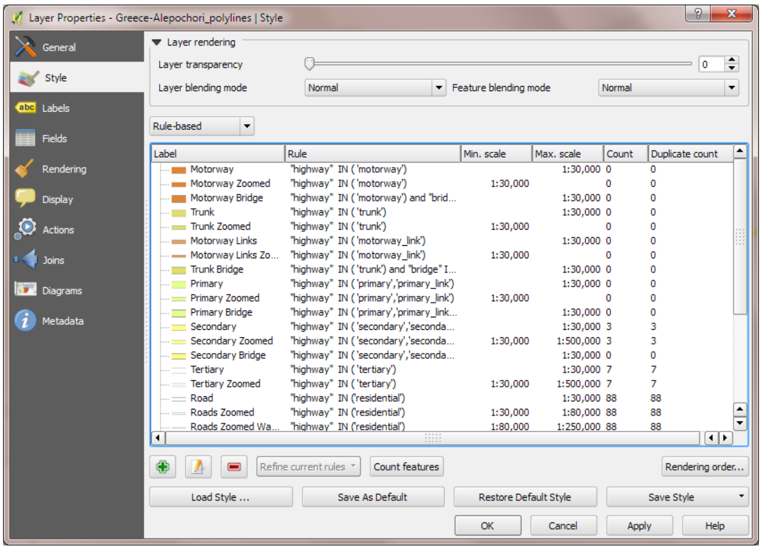


Figure 6.2: Rule-based styles from a qml file

It is worth having a look at the resulting styles - figure 6.2. The qml files do not include all possible styles and you may find that you need to create some more rules of your own to style the data that you have. The existing styles provide some examples that you can experiment with. Use the attribute table of the layer to see how tags have been used in the particular area that you have downloaded.

Making VML contours visible

To change the VML contour line styles so that the contours show:

- Go to the **style** tab in the properties for the line layer
- Find the contours - which are Values **15400** and **15401** and style them as you wish.

[4]Last visited: 8th May 2014

Manually editing qml files

It is sometimes necessary to manually edit qml files when it isn't possible to make the changes in the QGIS expression editor. For example when applying OS styles to VectorMap Local data the contour labels have been made transparent.

- Open **Text.qml** in a text editor
- Search for the **FeatCode** for the contour labels which is **15403**
- You are looking for a property which defines **FontTransp** - i.e. the transparency of the font! There will be a statement which says in part `15403 THEN 100 ELSE 0`. (There is also BufferTransp which you can change in the same way if you wish.)
- change the above snippet so that it reads `15403 THEN 20 ELSE 0` - or another number between 0 (opaque) and 100 (fully transparent).
- Save the file with a new name and then apply it to the text layer in your map.

6.13 SpatiaLite in ArcGIS

From version 10.2 ArcGIS supports SpatiaLite.

6.13.1 Create SpatiaLite database in Arc

Instructions here based at those on North River Geographic Systems blog at `http://bit.ly/1RSUOX2`

- Open the ArcPy terminal in ArcMap - Geoprocessing > Python
- type in the code below, one line at a time. The first line sets the location and name of the new database - the extension must be `.sqlite`

```
>>> sqlite_database = 'C:/clare_gis/project/test.sqlite'
>>> arcpy.gp.CreateSQLiteDatabase (sqlite_database,"SPATIALITE")
```

The new SpatiaLite database should now be in the location that you specified.

6.13.2 Importing feature classes from a geodatabase

To add data from a personal or file geodatabase to a SpatiaLite database do the following in the Catalog window.

- Right-click on the SpatiaLite database in Catalog and select Import > Feature Class (Single) OR Import > Feature Class (Multiple)
- Use the dialog box to select the feature class(es) that you wish to add
- OK

I have found (ArcGIS version 10.3) that I couldn't always then add these feature classes to a map in Arc, but they will work successfully in QGIS.

6.13.3 Exporting from a SpatiaLite to an ESRI File Geodatabase

To export data from an existing SpatiaLite database into a new or existing File Geodatabase do the following in ArcMap:

- In the Catalog right-click on the sqlite database
- Export > To Geodatabase (multiple)...
- Create a new gdb if necessary
- and then export.

Also possible to do for single layers - may be more reliable on large databases.

6.14 Routing with spatialite

This example uses Open Street Map data - download an `.osm.pbf` dataset from `http://download.geofabrik.de/`

Create a new empty database in spatialite-gui, then close the gui.

Download spatialite tools from `http://www.gaia-gis.it/gaia-sins/` - clicking to download **MS Windows binaries**. Specifically look for `spatialite_osm_net` (`spatialite_osm_map` for non network layers?) and unzip to your data file.

In a command window run:-

```
spatialite_osm_net -o england-latest.osm.pbf -d england.sqlite -T roads -m
```

Larger databases *may* work better without the **-m** switch at the end.

Now reopen the spatialite-gui

Files > Advanced > Build Network then fill the resulting dialog in as shown in figure 6.3.

Now open qgis and add the layers `roads` and `roads_nodes` to a map.

Open the database in Qspatialite and run the following command to query the shortest path between node 1 and node 1000.

```
SELECT * FROM roads_net
WHERE nodefrom = 1
AND nodeto = 1000
```

The output will have a number of rows. The first will be the only one with geometry information and will be a summary of the route. The following rows will be a step-by-step route guide from node to node.

To display in QGIS, first run the following command:

```
CREATE TABLE route1 AS
SELECT * FROM roads_net
WHERE nodefrom = 1
AND nodeto = 1000
AND geometry IS NOT NULL
```

Then recover geometry:

```
SELECT RecoverGeometryColumn('route1','Geometry',4326,'LINESTRING','XY')
```

replacing **4326** if appropriate.

You should then be able to right-click on the **route1** table in Qspatialite and Load in QGIS

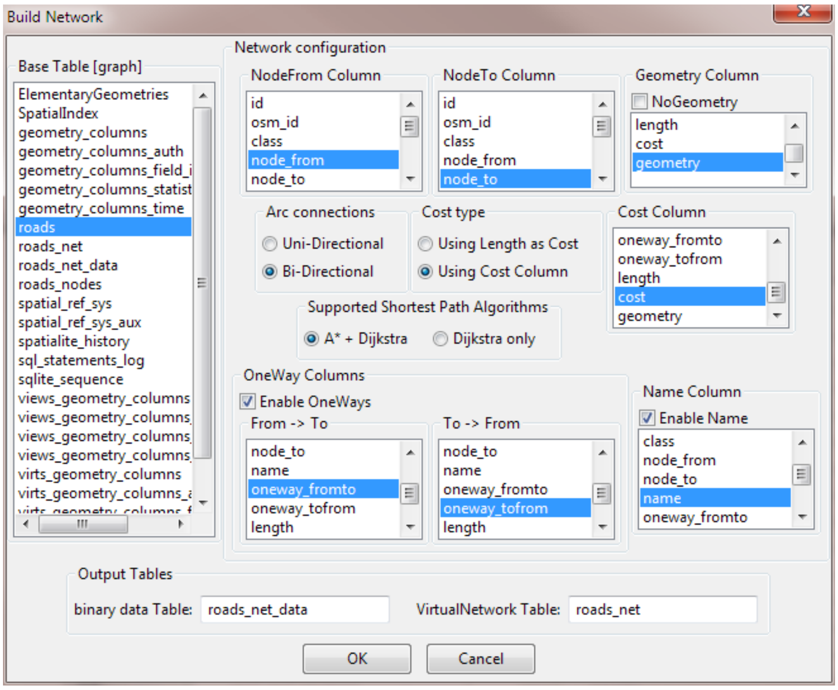


Figure 6.3: Building a spatialite network

6.15 Rasterlite2

Download from `https://www.gaia-gis.it/fossil/librasterlite2/home`

In particular unzip the following to the folder containing your spatialite database, and your raster files:

- `rl2tool.exe`
- `spatialite_gui.exe` [5]

To run commands open a command window in that folder.

Tutorial at: `https://www.gaia-gis.it/fossil/librasterlite2/wiki?name=rasterlite2-doc` which includes setting up a raster coverage and adding raster files as tiles. Also shows how to open in QGIS via a local WMS. It does appear that rasters in RasterLite are only available as a local WMS.

Is it possible to run analysis on Rasterlite layers?

[5] As of February 2016 this is a release candidate version which now supports rasterlite2.

Chapter 7

Grass GUI

Contents

These instructions apply to the Grass 7 Gui.

7.1 Merging rasters in Grass

Start by setting up a map set and region with the required coordinate system and resolution - Settings > Region > Set Region

Then import all of the raster data tiles:

- File > Import Raster Data > Common Formats Import
- Fill in dialog as in figure 7.1

Now you need to "collect" a list of the rasters

- File > Manage Maps > List
- select **raster map(s)** on **Required** tab
- on **Print** tab select **Print fully-qualified map names (including mapsets)** and change **Field separator** to a comma
- Click Run

You should end up with output that looks something like figure 7.2

In the output window select the line of file names - in figure 7.2 it's the line which starts **SJ20NE@earcej...** and copy it (either Ctrl + C or right-click and Copy

Once you have this copied, it can also be used to set the region to a smaller area by going to Settings > Region > Set Region and pasting the list into the space for **Set region to match raster map(s)**. This will save processing time and space!

It is also possible to give a **Name for output file** on the **Print** tab and the output will be directed there.

Now, finally, you can merge the rasters.

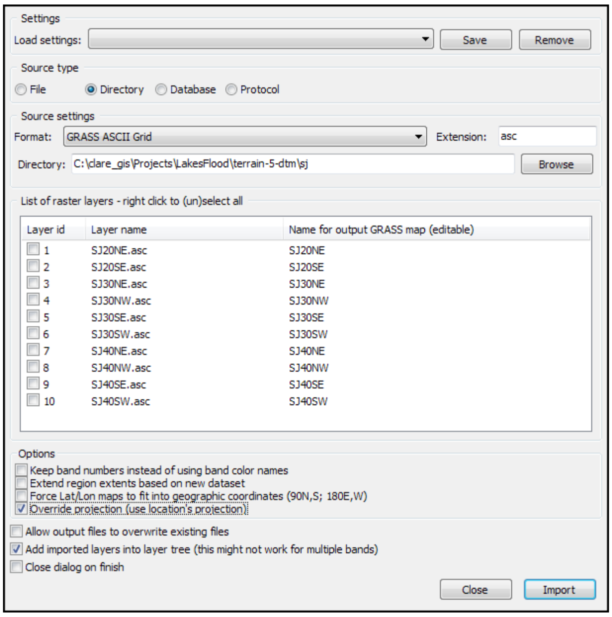


Figure 7.1: Importing multiple rasters in Grass GUI

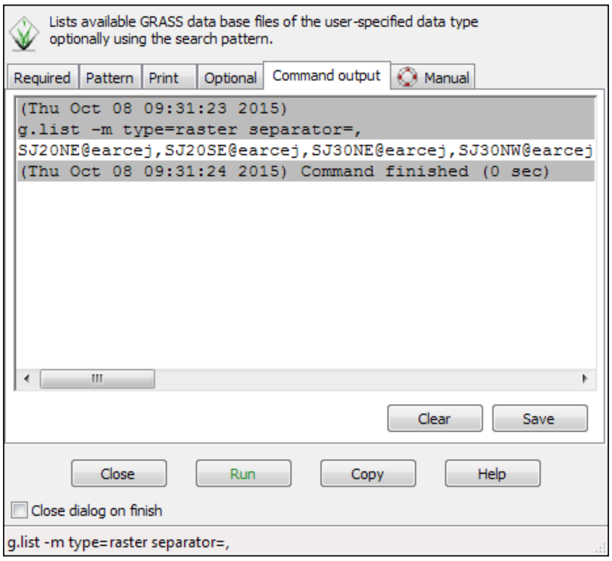


Figure 7.2: Output of file listing in Grass

- Raster > Overlay Rasters > Patch Raster Maps
- As you have copied the list of rasters you can paste that straight into the **Name of raster maps to be patched together** space (Ctrl + V or right-click and **paste**).
- Check that you've selected to **Add created map(s) into layer tree**
- Click Run - and hopefully it will (slowly!) merge your files.

Once the merge has finished the new merged raster should appear on the map display.

Finally it is possible to export the merged raster so that it can be used in other programs such as QGIS or ArcGIS.

- Either right-click on the merged raster and select Export

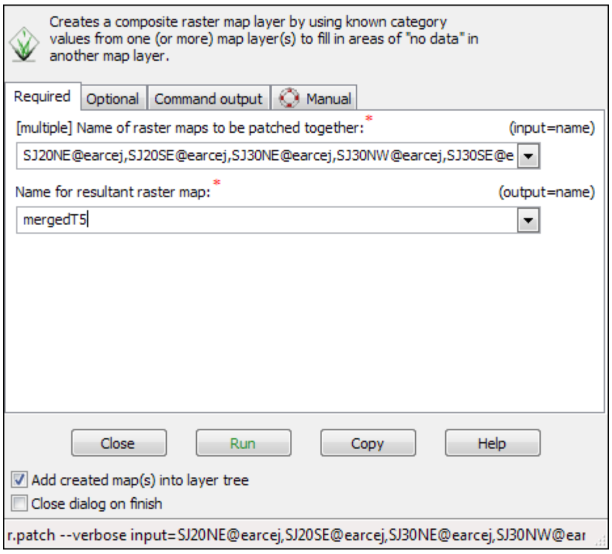


Figure 7.3: Dialog to patch or merge rasters

- or File 〉 Export Raster Map 〉 Common Export Formats
- Fill in the resulting dialog as shown in figure 7.4
- Click on Run to export the file

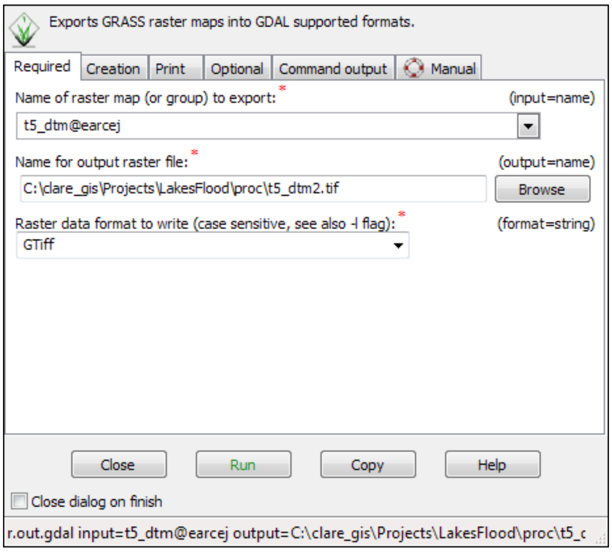


Figure 7.4: Exporting a raster from Grass as a tif file

7.2 Damming lakes in Grass GUI

Very brief notes for something that I probably don't need to do very often!

- Prepare raster dem and import to Grass
- in Arc or QGIS create a polygon feature class, digitise a dam across a valley, add height attribute which is substantially higher than ground underneath

- either convert to raster in Arc or Qgis, or import to Grass and run **v.to.rast.attr** to produce a raster file.
- use **r.mapcalculator** to run **dam - dem**
- run **r.null** on the output dam minus dem raster and set `nodata` to `0`.
- back in **r.mapcalculator** run **dem + dam minus dem no null** = `demwithdam`
- then use **r.lake** and click on a "seed" just behind the dam to flood a lake.

7.3 Converting ungridded xyz file to gridded raster

Based on data from Bill McCaffrey which was ungridded xyz files. For example:

```
179406.64 7616212.49 -711.52
179407.29 7616211.33 -711.51
179407.91 7616210.24 -711.52
179406.31 7616214.66 -711.59
179407.00 7616213.43 -711.58
179407.69 7616212.19 -711.53
etc
```

So the columns are xyz and the y column doesn't have groups of the same coordinate, which it would have if it was gridded. The required output is gridded xyz, but also useful to have an image format such as GTiff.

In this example the columns are separated by a space, but it could be a comma, or tab etc.

This workflow requires Grass 7 gui - don't have as many options in QGIS Grass plugin

Doesn't appear to interpolate to fill in no data areas.

- open Grass 7 gui and set up a Mapset with the required coordinate system (for Laverda = epsg:28350), but don't worry about extent yet.
- find and run **r.in.xyz** module
- Fill in details on **required**, **Statistic** and **Input** (Field separator is space) tabs then on **Optional** tab click to **Scan data file for extent then exit**
- When it finishes running make a note of the x and y min and max
- In gui - Settings > Region > Set region - go to **bounds** tab and enter values from previous tool (x min = w; x max = e; y min = s; y max = n), in **Resolution** tab enter a figure (in metres) for **2D grid resolution**, e.g. `10` for 10 m.
- Now run **r.in.xyz** again, this time not setting **Scan data file for extent...**
- If you've selected **Add created map(s) into layer tree** the map should appear in the Map Display, if it doesn't, right-click on it in the Map layers tree and select File > Map Display > Add raster and select the layer

Remember that operations only work on current region

Can combine layers by using **r.patch** or **r.series**. `r.series` gives a choice of method to deal with areas where both layers have values, i.e. overlap, so this is probably the best one to use.

Use **r.fillnulls** to fill null data areas. Works well on rectangular areas, but may need to use a mask if want to fill an irregular area. I haven't tried using a mask yet.

How do I work out whether output from this process (ungridded xyz to grass) is correct/what's necessary?

Can export layers using File > Export raster map > Common Export Formats - export as xyz and float64.

Export as xyz appears to give the correct format **but remember that this also depends on the region resolution and extent!** and can also export GTiff this way (this is the default)

OR

use **r.out.xyz** which allows a choice of field separator.

output format = gridded (regular) xyz, e.g.

```
ni=5 nj=total points/5
x  y z
1  1 87878
2  1 7897
3  1 879
4  1 78907
5  1 7890
1  2 7890
2  2 7690
3  2 7890
```

etc

Chapter 8

Python

Contents

8.1 Basics

To run a python script in the QGIS Python Console:

```
exec(open('e:/path/to/file/code.py'.encode('utf-8')).read())
```

8.2 Trouble shooting

To view the Log panel in QGIS go to

- View > Panels > Log panel

To find the python version number at the console:

```
>>> import sys
>>> sys.version
```

8.2.1 pyqgis - "No module named _gdal_array"

Problem that I had when trying to run gdal processing tools via a python script within the Python console in QGIS 3.4.3. Output not generated and when I looked in the log I had the following error message:

```
Import Error:  No module named _gdal_array
```

Solution[1] was to open the OSGeo4W shell and type the following command:

```
» python -m pip install -U numpy
```

Updating in python 3 instead of 2

By default the OSGeo4W prompt runs python 2. If you want python 3 type the following:

```
py3_env
```

8.2.2 To find the home directory in python

```
from os.path import expanduser
home = expanduser("~")
print(home)
```

The result should be something like

```
C:\Users\earcej
```

[1] Found at https://trac.osgeo.org/osgeo4w/ticket/588 Viewed on 14th January 2019

8.3 Rasterio package for spatial rasters

8.3.1 Open raster data with Rasterio and view metadata

Read the raster into a numpy array as follows.

`lidar_dtm =` means that the data will then be available outside of the "with". Otherwise the file is closed at the end of the loop.

```
import rasterio as rio

# Open DTM
with rio.open(filenamedtm) as lidar_dtm:
    print("Bounds =", lidar_dtm.bounds)
    print("Image structure =", lidar_dtm.tags(ns='IMAGE_STRUCTURE'))
    lidar_dtm_mask = lidar_dtm.dataset_mask()
```

```
print(lidar_dtm.meta)
```

will give something like

```
{'driver': 'GTiff', 'dtype': 'float32', 'nodata': -3.4028234663852886e+38,
   'width': 4000, 'height': 2000, 'count': 1, 'crs': CRS.from_epsg(32613),
   'transform': Affine(1.0, 0.0, 472000.0, 0.0, -1.0, 4436000.0)}
```

Pull out one metadata property at a time

Entering:	Gives you:
`print(lidar.driver)`	`GTiff`
`print(lidar.bounds)`	`BoundingBox(left=472000.0, bottom=4434000.0, right=476000.0, top=4436000.0)`
`print(lidar.nodata)`	`-3.4028234663852886e+38`
`print(lidar.crs)`	`EPSG:27700`
`print("Height of raster tile = ", lidar.height)`	`Height of raster tile = 2000`
`print("Width of raster tile = ", lidar.width)`	`Width of raster tile = 4000`

8.4 Pandas dataframes

8.4.1 Read csv into a dataframe

8.5 Geopandas package for vector data

8.5.1 Find names of layers in multi-layer format vector

To find out the layers present in a multi-layer format, e.g. osm, gpkg, sqlite, try the following

```
import fiona
print(fiona.listlayers('map.osm'))
```

8.5.2 Reading a layer into a geopandas geodataframe

To read a layer into a geopandas geodataframe:

```
import geopandas as gpd
gdf = gpd.read_file('data.gpkg', driver='GPKG', layer='lines')
```

8.5.3 Finding out metadata

Once the layer is in a geodataframe find information as follows:

Find the CRS (Coordinate reference system):

```
print(gdf.crs)
```

output will be an epsg code in a similar format to the following:

```
'init': 'epsg:4326'
```

To reproject the geodataframe try the following:

```
gdfbng = gdf.to_crs({'init': 'epsg:27700'})
```

To find out the geometry of the features (which may not all be the same...) try:

```
print(gdf.geometry)
```

Output options will include LINESTRING; MULTIPOLYGON; POINT etc.

Number of columns and rows:

```
print(gdf.shape)
```

Column names:

```
print(gdf.columns)
```

Print the first 5 rows of the geodataframe (use `gdf.tail()` for the last 5 rows):

```
print(gdf.head())
```

A quick overview of the contents of one record (0 = the first record):

```
print(gdf.loc[0])
```

A quick overview of the contents of one column by name:

```
print(gdf['name'])
```

Search for text in a field:

```
street = gdf.loc[gdf['name'] == "North Road"]
print(street)
```

It is possible to combine filters, e.g.

```
water = gdf[gdf['natural'].str.contains("water") & gdf['natural'].notnull()]
```

To list the unique values in a field use the following:

```
print(gdf['natural'].unique())
```

8.6 Plotting with geopandas and matplotlib

To print a quick plot of a geopandas geodataframe use the following:

```
import geopandas as gpd
import matplotlib.pyplot as plt

gdf = gpd.read_file('data.gpkg', driver='GPKG', layer='lines')

gdf.plot()
plt.show()
```

8.6.1 Matplotlib

Geopandas uses the matplotlib package for plotting so options come from there.

- The **figure** is the overall plot
- the **ax** is the object which holds the axes, i.e. the individual **subplots**
- each **ax** then has **x-axis** and **y-axis**

For a plot with more control try the following.

```
import geopandas as gpd
import matplotlib.pyplot as plt

# reading in data
gdf = gpd.read_file('data.gpkg', driver='GPKG', layer='lines')

f, ax = plt.subplots(1, figsize=(10,10))
ax = gdf.plot(ax=ax,
column="colname",
cmap='tab20c')
f.suptitle('Title here')
lims = plt.axis('equal')
plt.show()
```

To plot two datasets onto one subplot

```
fig, ax = plt.subplots(1,figsize=(5,5))

Becks.plot(ax = ax, linewidth=0.6)
UKOutline.plot(ax = ax, linewidth=0.4)

plt.show()
```

To plot an image file (3 colour bands)

```
# Plot data to check it.  Can't do it directly need to reopen as an image
import rasterio
import numpy as np
import matplotlib.pyplot as plt

source = rasterio.open(outputColourRelief)
red = source.read(1)
green = source.read(2)
blue = source.read(3)
pix = np.dstack((red,green,blue))
bounds = (source.bounds.left, source.bounds.right,
source.bounds.bottom, source.bounds.top)

# Plotting
f = plt.figure(figsize=(6,6))
ax = plt.imshow(pix, extent=bounds)

plt.show()

# Save figure to disk if required
# plt.savefig(outputpath + 'output.png')
```

8.6.2 Creating multiple subplots

In the `add_subplot` command `111` is equal to `1,1,1` which means the following:

- the number of rows
- the number of columns
- the plot number

So to create a single subplot:

```
fig = plt.figure()

# set up axes
ax = fig.add_subplot(111)
ax = gdf.plot(ax=ax,
column="colname",
cmap='tab20c')

# show the plot
plt.show
```

To create two subplots:

```
# Initialise the plot
fig = plt.figure(figsize=(10,4))

# Plot the data
ax1 = fig.add_subplot(121)
ax2 = fig.add_subplot(122)

# Plot the data
ax1.bar([1,2,3],[3,4,5])
ax2.barh([0.5,1,2.5],[0,1,2])

# show the plot
plt.show()
```

8.6.3 Formatting plots

Colours

Various ways of referencing colours, but the simplest is probably the named colours. See figure 8.1

To change colours of the data:

- **facecolor** and **edgecolor** allow fill and line to be coloured separately
- **facecolor = 'none'** turns of fill completely

for example:

```
Becks.plot(ax = ax, color='darkblue', linewidth=0.6)
UKOutline.plot(ax = ax, facecolor='none', edgecolor='cornflowerblue', linewidth=0.4)
```

To change the colour of the plot background:

```
ax.set_facecolor('dimgrey')
```

Figure 8.1: Python matplotlib named colours

Axis visibility

To turn axis figures on or off:

```
    ax.get_xaxis().set_visible(False)
ax.get_yaxis().set_visible(False)
```

Adding titles

- **fig.suptitle()** adds a title to the whole figure
- **ax.set_title()** adds a title to the subplot

For example:

```
fig.suptitle('Names of watercourses in the United Kingdom', fontsize=18)
ax.set_title('Becks', fontsize=14)
```

8.7 Useful snippets

8.7.1 Get today's date

```
from datetime import datetime
currdate = datetime.today().strftime('%Y-%m-%d')
```

Output should be: `2020-01-29`

8.7.2 Creating a test pandas dataframe

```
import pandas as pd

d = {'date': ['2020-01-28','2020-01-29'], 'col2': [3,4]}
df = pd.DataFrame(data=d)
```

8.7.3 Converting time field in a pandas dataframe to a datetime datatype

```
import pandas as pd
df['datedatatype'] = pd.to_datetime(df['date'])
```

Test data type of columns with

```
print(df.dtypes)
```

Output in datedatatype field will be `datetime64` rather than `Object` (string)

8.8 Sources of help

Basically a list of bookmarks!

Python GDAL/OGR Cookbook - https://pcjericks.github.io/py-gdalogr-cookbook/

Unofficial Windows Binaries for Python Extension Packages - https://www.lfd.uci.edu/~gohlke/pythonlibs/

Chapter 9

Extra techniques

Contents

9.1 OpenStreetMap with osmconvert and osmfilter

This is useful if you only want to export a particular subset of OSM data, e.g. place names. If you convert the whole file you end up with the same output as the osm files that you can download directly.

The general instructions were found at `https://help.openstreetmap.org/questions/19514/export-of-placenames`

Download **osmconvert** from:

`http://wiki.openstreetmap.org/wiki/Osmconvert`

Download **osmfilter** from:

`http://wiki.openstreetmap.org/wiki/Osmfilter`

Both are `.exe` files which you can run from the command line[1].

To use:

- Download a `.pbf` file from `http://download.geofabrik.de/`
- Place the pbf file in the same folder as the exe files

[1]These instructions assume you know how to use the windows command line, if you don't go to `https://www.computerhope.com/issues/chusedos.htm` for some basic instructions (page last accessed: 23rd April 2018)

- Open a command window and navigate to the folder then type the commands below using the names that match your pbf file

Convert the file using:

```
osmconvert great-britain-latest.osm.pbf -o=great-britain.o5m
```

To filter the output (or it will be **very** large) use `osmfilter`. There are full details at the link above, but for example, to output all railway stations in GB.

NOTE: the hyphen before `keep` should be double.

```
osmfilter great-britain.o5m --keep="railway=station" -o=stations.osm
```

or, saving just nodes (if you don't want nodes use `-drop-nodes`

```
osmfilter great-britain.o5m --keep="place=city or place=town or place=village"
--drop-ways --drop-relations -o=places.osm
```

A list of tags in OSM data can be found at:

`http://wiki.openstreetmap.org/wiki/Map_Features`

In addition **osmfilter** can be used to find all keys (or tags) in the data that is being used:

```
osmfilter great-britain.o5m --out-key=place --out-count
```

Once you have an osm file, it's then possible to use **ogr2ogr** to convert the output `osm` file to another format for use in gis. See section 5 on page 34 for full instructions. For example, within a gdal command window (not necessarily the same as a Windows command window) type the following to convert OSM to a shapefile to contain point features:

```
ogr2ogr -f "ESRI shapefile" -t_srs epsg:27700 places.shp places.osm
```

If you need to specify the geometry type of data, e.g. POINT, LINESTRING, POLYGON, MULTIPOLYGON etc try the following. You'll probably end up with a folder with the name of the shapefile which contains multiple shapefiles, one for each geometry type.

```
ogr2ogr -f "ESRI shapefile" -t_srs epsg:27700 places.shp places.osm -nlt POINT
```

9.2 Working with geotagged images

9.2.1 Adding geotag information to photographs

Modern digital cameras, or smart phones and tablets, will often add locational information to photographs. If your devices make that possible then that is the simplest method.

For older photographs or equipment that doesn't add tags, use one of the two methods below to add them depending on whether you have a gps track available or not.

Adding geotags to photos from a gps track

If you have a track from a gps recorded while you were taking the photographs then you can add geotags to the photos using the track.

You do need either to know the time difference between the photos and the gps track, or to be able to accurately locate at least one of the photographs.

Download GPSPrune from `http://bit.ly/23HbyZU`. You don't need to unzip the file, but you do need Java to run it. If it doesn't run when you double-click on the jar file, try right-clicking and Open with > Java Platform SE binary.

You will need to download exiftool if you don't already have it, and tell gpsprune where it is located.

- Download exiftool from `http://www.sno.phy.queensu.ca/~phil/exiftool/`
- unzip to a folder and make a note of the full path to the folder (e.g. `E:\gis\Software\exiftool\`)
- In gpsprune - Settings > Set program paths - in the **Path to exiftool** enter the path and add the name of the exiftool exe at the end so that the above example will read `E:\gis\Software\exiftool\exift` `k))`
- click on **check** to make sure that gpsprune can find the file and if Path found changes to **Yes** click on OK

Start by opening a gpx file downloaded from your gpx unit

- File > Open file and select your gpx file.

Any tracks should appear in the main window. To see the track against a map background View > Show map - then check your track is where you expect it to be! (Figure 9.1.)

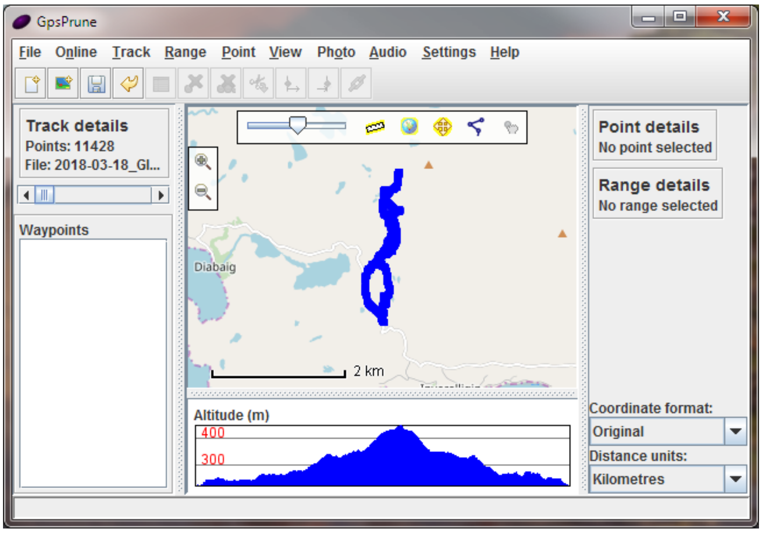


Figure 9.1: Viewing a track in gpsPrune

Next add the photos you want to geotag

- Photo > Add Photos and either select a folder or individual photos

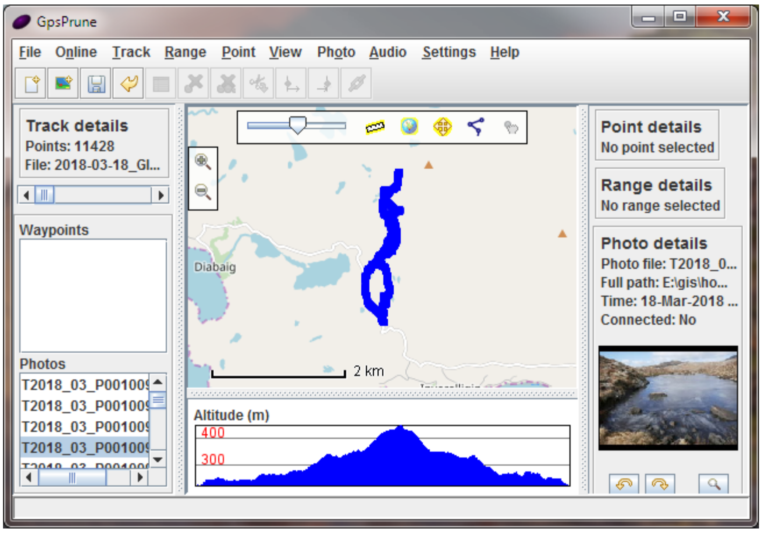


Figure 9.2: Previewing a photograph in gpsPrune. The photo isn't yet geotagged.

A list of the photos will appear, but as the photos are not yet geotagged there will be nothing extra visible on the map. You can click on a photograph to see more details and a thumbnail preview (figure 9.2).

If you can accurately locate one of the photographs on a map setting the time offset between the photos and the gpx track manually seems to be the most reliable method.

- Select the photograph in the list on the left
- On the map zoom in to the location of the photo and click on the track to select a **point**
- Photo > Connect to Point - a yellow dot should appear at the same location as the selected point
- Photo > Correlate photos - the photo that you connected to a point should be selected and the time difference shown, so just press Next
- Now you should see a list of photos ready to correlate. The **Time offset** will be set automatically and the **Time difference** for each photo listed - figure 9.3.
- If you are happy with the list click OK

If you can't accurately locate any of the photos, or would rather set the time offset during correlation you can go straight to Photo > Correlate photos and add the **Time offset** at the top yourself.

Once you have correlated the photos, check the locations carefully first[2], then save them to the exif information of the photographs by going to Photo > Save to exif.

Adding geotag information to images without a gps track

The **Geosetter** program makes it possible to add locational information using a Google map. Download the program from:

`http://www.geosetter.de/en`

It hasn't been updated since 2011 but still works, and uses Exiftool, which is still being updated (see section 9.2.2 on page 79 for more information about Exiftool). If you download the zip version of geosetter you can run it from anywhere without needing administrator rights to install it.

[2] You can do this either by checking against the map within GPSPrune, or, if the map isn't showing up, you can export a kmz file and open that in Google Earth to check it. If you **Export image thumbnails to kmz** then clicking on the markers will show the photograph.

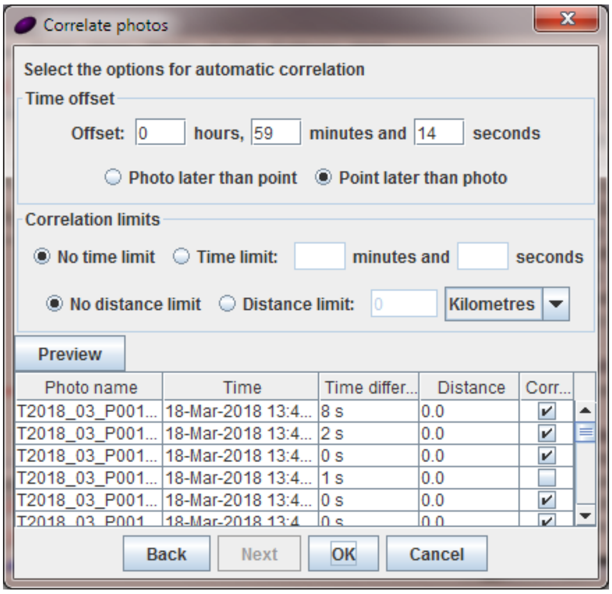


Figure 9.3: Setting the time correlation between the photographs and the track

9.2.2 Extracting location data from geotagged images

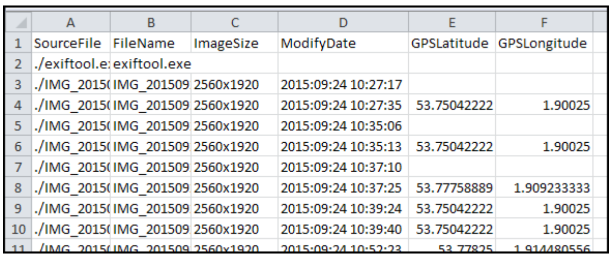
Extracting location data from geotagged images - command line tool

To extract latitude and longitude from geotagged jpg images to a csv file try the following:

- Download exiftool from `http://www.sno.phy.queensu.ca/~phil/exiftool/`
- unzip to the folder where your images are stored, and rename `exiftool(-k).exe` to `exiftool.exe`
- open a command window in the same folder then enter the command below (all on one line)

```
exiftool -csv -filename -imagesize -ModifyDate -gps:GPSLatitude -n
    -gps:GPSLongitude -n ./> long2.csv
```

The output from the command above should look something like figure 9.4 when opened in Excel.



	A	B	C	D	E	F
1	SourceFile	FileName	ImageSize	ModifyDate	GPSLatitude	GPSLongitude
2	./exiftool.e:	exiftool.exe				
3	./IMG_2015(	IMG_201509	2560x1920	2015:09:24 10:27:17		
4	./IMG_2015(	IMG_201509	2560x1920	2015:09:24 10:27:35	53.75042222	1.90025
5	./IMG_2015(	IMG_201509	2560x1920	2015:09:24 10:35:06		
6	./IMG_2015(	IMG_201509	2560x1920	2015:09:24 10:35:13	53.75042222	1.90025
7	./IMG_2015(	IMG_201509	2560x1920	2015:09:24 10:37:10		
8	./IMG_2015(	IMG_201509	2560x1920	2015:09:24 10:37:25	53.77758889	1.909233333
9	./IMG_2015(	IMG_201509	2560x1920	2015:09:24 10:39:24	53.75042222	1.90025
10	./IMG_2015(	IMG_201509	2560x1920	2015:09:24 10:39:40	53.75042222	1.90025
11	./IMG_2015(	IMG_201509	2560x1920	2015:09:24 10:52:23	53.77825	1.914480556

Figure 9.4: csv output from exiftool

Notice, though, that the Longitude decimal degrees should be negative for areas west of the Greenwich Meridian - such as in the example. You will need to correct these in Excel[3].

[3] I used the function `=E2-(E2*2)`. There is probably something more obvious!

This file can then be imported into a GIS to show locations of photographs on a background map. You may need to edit the csv file first to clean up column names and remove lines for files which are not images.

To find out which tags you can use in the command above try the following:

```
exiftool -a -G1 -s ./> allinfo.csv
```

Extracting location data from geotagged images with a graphical user interface

Geotag is a graphical front end to the exiftools which means that you don't have to type in commands. It does require that your computer has a Java SE 7 runtime. The page at `http://geotag.sourceforge.net/Requirements/` gives more details, and also gives links to other tools that may be useful.

- In addition to downloading **exiftools** (instructions above) download Geotag from `http://geotag.sourceforge.net/Download/` [4]
- Unzip the downloaded file and put both exiftools and Geotag in the same folder.

When you run Geotag (which you may need to do by right-clicking on the jar file, and Open with... > Java Platform SE Binary) you'll probably need to start by telling it where to find exiftool (the renamed version) by editing the settings.

Once Geotag is up and running File > Add images from directory... to load your images. You should end up with a list of images showing gps and related information, and a preview window below (figure 9.5). It's now possible to carry out various tasks with the images by right-clicking on the list. This includes exporting all image locations to a kml file which can then be opened in Google Earth or GIS.

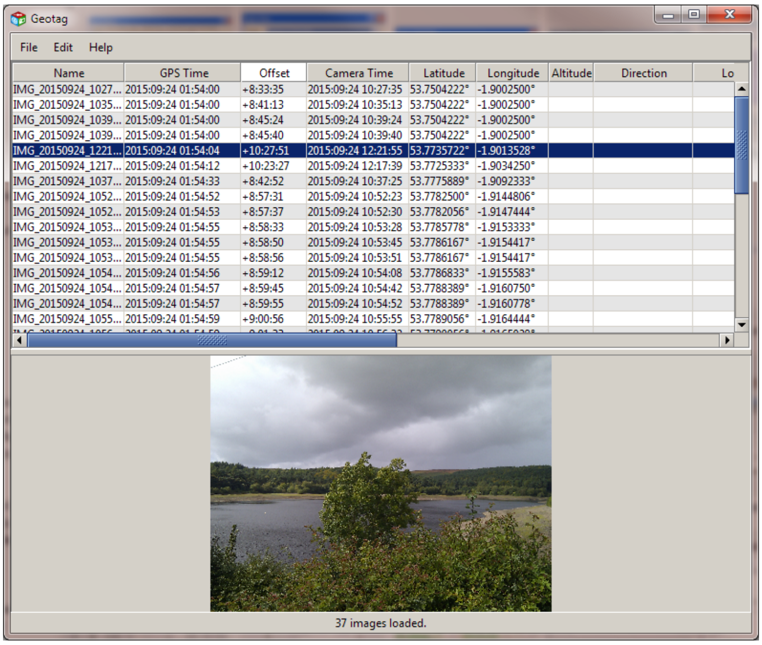


Figure 9.5: Geotag with images loaded

[4] Last visited: 2nd February 2016

9.2.3 Displaying geotagged photos

Displaying geotagged photos in Google Earth

Download GPSPrune from `http://bit.ly/23HbyZU`. You don't need to unzip the file, but you do need Java to run it. If it doesn't run when you double-click on the jar file, try right-clicking and Open with > Java Platform SE binary.

Once open, add photos by going to Photo > Add photos

Add a background map by clicking on the globe symbol on the toolbar at the top.

Select a photo in the list on the left to see more details. See figure 9.6.

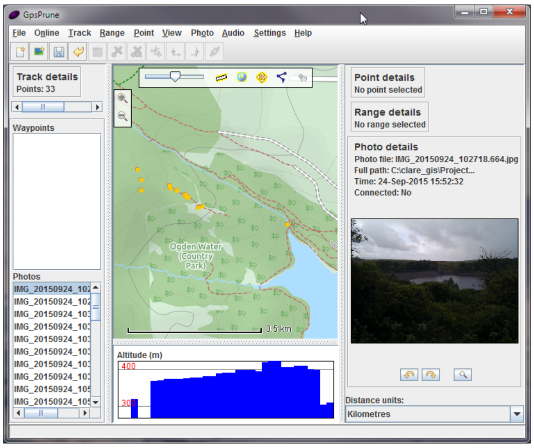


Figure 9.6: GpsPrune with images loaded

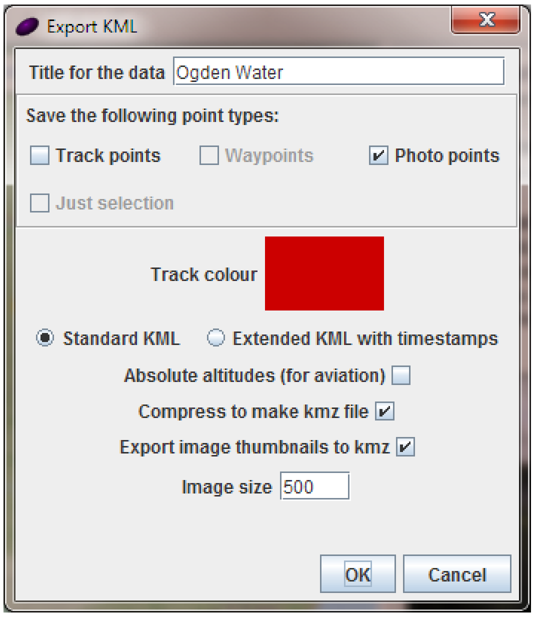


Figure 9.7: Exporting KMZ from GpsPrune

If your photographs do not already have geotags they won't appear on the map, in this case use the instructions in section 9.2.1 on page 77 to add the geotags before you go any further.

To export the images to Google Earth: File > Export KML.

To be able to see the images as pop up thumbnails in Google Earth fill in the dialog box as shown in figure 9.7

Displaying geotagged images in QGIS

Start by using gpsprune to create a kml file from a collection of geotagged photos - see section 9.2.3 on page 81 for how to do this.

- Add the kml or kmz file to a map in QGIS[5].
- Check that the eVis plugin is activated (Plugins > Manage and install plugins... then tick next to eVis.

[5]Check that **On the Fly (OTF)** is set for the coordinates if they are not lat/long. The kml file will be WGS1984 and won't show in the correct place on the map unless OTF is set - Project > Project Properties > CRS tab and check the box to **Enable 'on the fly' CRS transformation (OTF)**

- In the QGIS table of contents select the kml with details of the photos
- click on the eVis **Event browser** button
- On the **Options** tab set **Attribute containing path to file** to **Name**
- Check that **Path is relative** is selected
- Reset **Base path** to the folder containing the photos
- Finally, use the **Display** tab to browse photos - as you move between the photos in the viewer the location on the map is marked by a red asterisk.

Chapter 10

Workflows for using specific datasets in GIS

Contents

10.1 Workflow for 1:25 000 UK map using open source data in QGIS

These instructions work for both OS VectorMap District and Local vector layers downloaded from Digimap. Note the different ways of displaying contours, though.

10.1.1 Download data

From OS OpenData - `http://bit.ly/1lmb3Q6` Download:

- VectorMap District or OS Open Map - Local vector files in **gml** format
- Terrain 50 contour files in gml format (one download for whole country!) for use with VectorMap District data.

10.1.2 VectorMap District vector files

This section uses gml files from the OS. Section 6.3.4 on page 47 shows how to use gml files in SpatiaLite, using VectorMap District as the example. The advantage of using this method to convert the files to SpatiaLite is that field names can be longer than is possible in shapefiles, and all of the data ends up being stored in one file!

Download the appropriate Ordnance Survey style files from `http://bit.ly/1KuvuDA`

Note that documentation is included in the style downloads - use this! And also go to the following urls for more information.

`https://www.ordnancesurvey.co.uk/docs/user-guides/os-opendata-getting-started-guides.pdf`

`https://www.ordnancesurvey.co.uk/resources/carto-design/cartographic-stylesheets.html`

`https://www.ordnancesurvey.co.uk/docs/user-guides/cartography-stylesheet-user-guide.pdf`

- If using the instructions at section 6.3.4 on page 47 you can merge the VectorMap District files as you convert them to SpatiaLite if you need to.
- If area is too big, select required area and export features to a new feature class (saves rendering time!) see section 6.10.8 on page 55
- add the layers to your map as shown in the documentation for the VMD style sheets
- Symbolise using the downloaded qml files

10.1.3 Contours and spot heights from Terrain 50

- Select required grid squares of data and add to a SpatiaLite database using the instructions in section 6.3.4 on page 47.
- add the contour data to QGIS
- Symbolise using the downloaded qml files.

Labelling Terrain 50 contours

Terrain 50 contour style sheets from `http://bit.ly/1KuvuDA` don't include labels so use the properties to set those.

Do they still not include labels? May 2018

If you want to include the unit of measure as well as the height try `"propertyValue" || "propertyValue_uom"` in the **Label this layer with** box (|| is the concatenation symbol!).

10.1.4 Changing styles from qml files

The files may need a little extra work, for example, the VectorMap Local styles as you download them don't include contours or contour labels (The VectorMap District data doesn't include contours anyway - you'll need to download Terrain 50 contours instead).

Use the instructions in section 6.12.1 on page 57 to make VML contours visible.

To display **labels** for the contours you need to edit the qml file in a text editor. They are in the file, but have been made transparent.

Use the instructions in section 6.12.1 on page 58 to show the contour labels.

10.1.5 OpenStreetMap data

Particularly for maps of rural areas it can be useful to include extra information that isn't available in the open OS data, e.g. footpaths and walls and fences. While these aren't always complete in OpenStreetMap they can still add useful information to a map.

Download and process data from OpenStreetMap following the instructions in section 6.3.2 on page 45.

It will be necessary to reproject the OSM files into British National Grid if you are going to use them with OS data - see section 6.6 on page 51.

The most useful additions to VectorMap District are:

- from the lines feature class:
  - paths and tracks
  - walls and fences
- from the points feature class:
  - hills

Symbolise using qml files - some suggestions in section 6.12.1 on page 56.

10.1.6 Copyright notices

For Ordnance Survey OpenData:

```
Contains OS data © Crown Copyright [and database right] (20yy)
```

Find out more about their data licensing at
http://bit.ly/2cjSS01
For Open Street Map data

```
© OpenStreetMap contributors
```

There is more information about OSM copyright at
http://www.openstreetmap.org/copyright

10.2 Workflow for using open source tools to add Environment Agency lidar to a map

To use open source tools to add Environment Agency lidar to a map:

- Download data from: http://environment.data.gov.uk/ds/survey#/
- Unzip asc files
- Convert asc files to geotiff using GDAL (section 4.2 on page 23)
- Merge files using GDAL (section 4.4 on page 24)
- If necessary reproject the merged files (section 4.11 on page 31)
- Open QGIS and add merged file(s)
- Set raster values to 0 using the raster calculator - section 3.4 on page 18 - first alternative.
- then set 0 to nodata (null) using section 3.5 on page 19

Then eventually you have a raster dem with "holes" where there is no data! If you want to fill in the holes using data from another dem dataset, e.g. OS Terrain 5 or 50, try the instructions in section 4.7 on page 27.

10.2.1 Copyright statement for Environment Agency lidar data

© Environment Agency copyright and/or database right 20*yy*. All rights reserved.

Also a good idea to add the url of the download site.

10.3 Workflow for creating basemaps for Spanish mapping areas

- Download BTN/BCN 25 vector data from http://bit.ly/1WsHmyh (see separate sheet...) and unzip into folders
- bulk rename layers and move them so that they are all in same folder but with a distinguishing name for each tile.[1]
- Merge layers following instructions below

10.3.1 Merging Spanish vector data tiles with ogr2ogr

Create a .bat file containing the following code - with sections added for each layer required.

```
mkdir merged

for /R %%f in (Vector25\*BCN0801P_TOP_SIN_GEO.shp) do (
if not exist merged\BCN0801P_TOP_SIN_GEO.shp (
ogr2ogr -f "esri shapefile
 " -s_srs EPSG:25831 -t_srs EPSG:25831 merged\BCN0801P_TOP_SIN_GEO.shp %%f) else (
ogr2ogr -f "esri shapefile" -s_srs EPSG:25831 -t_srs EPSG:25831
  -update -append merged\BCN0801P_TOP_SIN_GEO.shp %%f -nln BCN0801P_TOP_SIN_GEO )
)

for /R %%f in (Vector25\*BCN0201L_CUR_NIV.shp) do (
if not exist merged\BCN0201L_CUR_NIV.shp (
ogr2ogr -f "esri shapefile
 " -s_srs EPSG:25831 -t_srs EPSG:25831 merged\BCN0201L_CUR_NIV.shp %%f) else (
ogr2ogr -f "esri shapefile" -s_srs EPSG:25831 -t_srs
  EPSG:25831 -update -append merged\BCN0201L_CUR_NIV.shp %%f -nln BCN0201L_CUR_NIV )
)
```

Ensure that you are using the correct EPSG code[2] and it's not worth trying to reproject using ogr2ogr...

Use SDKShell to run the bat file - the folder that new files are saved to, needs to be different to the folder that the existing shapefiles are in!

10.3.2 Reprojecting etc Spanish vector tiles

- In Arc add the resulting layers and batch reproject to the EPSG code that you used in the bat script! ogr2ogr seems to create a custom projection which causes problems later on

[1] Bulk Rename Utility from http://www.bulkrenameutility.co.uk/Main_Intro.php does a good job of this and can be used without having to install it on a computer, so avoiding the need for admin access.

[2] Check at http://epsg.io/ if required

Layers used in Spain base maps
• BCN0801P - Geographic names • BCN0802P - Geographic names - natural features? • BCN0803L - Geographic names - mountain ridges • BCN0204P - spot heights • BCN0201L - contours • BCN0301L - streams • BCN0310L - dry river? • BCN0325L - LIN_EMB (? something watery) • BCN0328S - water tanks • BCN0316S - lakes • BCN0604L - main roads • BCN0610L - minor roads • BCN0620L - another type of road • BCN0623L - tracks • BCN626L - paths • BCN635L - some sort of path? • BCN0501S - towns • BCN0507S - buildings Label with the **ETIQUETA** field. Also possible to download OSM water polygon layers for whole world to fill in the sea, from `http://openstreetmapdata.com/data/water-polygons` Select just the polygons that you need, export to a new shapefile, then reproject to your current projection.

Table 10.1: Layers used in Spain base maps

10.3.3 Changing Spanish vector map tiles to ED50

To change to ED50 from ETRS1989 (e.g. 25831), so that maps can be used with GPS

- In Arc - define projection for any rasters with <undefined>
- project all raster tiles to the new projection, e.g. 23031
- *Individually* project shapefiles to new coordinates. Batch projection is difficult to set transformation - which is required to make all of these files fit together.

10.4 Creating a basemap with open source data in ArcMap

These instructions show how to produce a base map for geological mapping, though can also produce a respectable background map for anything. The idea is that it uses entirely open source data and covers areas for which nothing else that is satisfactory is available.

- Follow the instructions in section 1.6.1 on page 3 to obtain the SDKShell which will enable you to run the ogr and gdal libraries.

10.4.1 OpenStreetMap background map

- Follow the instructions in section 6.3.2 on page 45 to export the OpenStreetMap layers as a spatialite database.

Saving the data as shapefile, which would be the obvious option, has the disadvantage of shortening field names, and causing problems with non-standard characters in the data. Exporting to spatialite doesn't have these disadvantages and later versions of ArcMap, e.g. version 10.4, are able to open spatialite databases. Exporting as a FileGDB is theoretically possible, but involves compiling your own version of GDAL!

10.4.2 Contours or height data

These will be generated from the SRTM 1-Arc second dataset and you can put them into the spatialite database created in the previous section.

- Download data for the appropriate area from `https://earthexplorer.usgs.gov/`. Registration is required, but it is free.
- If your area is covered by more than one srtm tile use the instructions in section 4.4.1 on page 24 to merge the srtm tiles into a single output.
- Use the instructions in section 4.12 on page 31 to smooth the output and generate contours. Contours will be generated in shapefile format and can either be used as is, or saved to a spatialite geodatabase by using the command in that section.

Warning: Be aware that the horizontal resolution of the SRTM data is only 30 m so for a detailed map, such as a 1:10 000 map, the contours will not cover every dip and rise in the land. In particular, be extra cautious in areas where there are sharp cliffs etc.

10.4.3 Creating the map

Open ArcMap and go to the sqlite database in ArcCatalog. Your final map will work more smoothly if you bring your data into a **File Geodatabase**, so create a new one then drag each layer from the sqlite database to the file geodatabase. The coordinate system will remain the same and the data will still have the correct encoding.

- Open the **OSMTemplate.mxd** file which can be downloaded from and save it with a new name
- Change the coordinate system if necessary (Data frame properties)
- Switch on a basemap layer and navigate to the approximate location of your data
- Each layer is most likely to have a red exclamation mark next to it - indicating that it can't find the data. Right-click on a layer then Data > Repair Data Source. Browse to your file geodatabase and select the appropriate layer, e.g. the OSMplaces layer will need to be linked to the **points** layer.
- If you're lucky, when you do one layer the rest will follow, but if not repeat for each layer.

Upload OSMTemplate.mxd to my website

For coastal areas download water polygons from `http://openstreetmapdata.com/data/water-polygons` and add those to your map. This is symbolised by the **sea** layer in the map.

10.5 Using USGS topo GeoPDF files in GIS

USGS GeoPDF topo maps (latest series) can be downloaded for free from:

`http://viewer.nationalmap.gov/basic/`

Information on how to convert layers from these to raster layers so that they can be used in GIS is available for download from:

`https://www2.usgs.gov/faq/categories/9797/3704`

But essentially:

First check for layer names:

```
gdalinfo input.pdf -mdd LAYERS
```

The output should include a long list of layers. The information you need is the name after the equals sign, for example from:

```
LAYER_15_NAME=Map_Frame.Terrain.Contours
LAYER_16_NAME=Map_Frame.Terrain.Shaded_Relief
```

you will need to use `Map_Frame.Terrain.Contours` in the following command

```
gdal_translate input.pdf output.tif -config GDAL_PDF_LAYERS
"Map_Frame.Terrain.Contours" -config GDAL_PDF_BANDS 3 -config GDAL_PDF_DPI 400
```

Multiple layers can be exported by listing them in the command:

```
gdal_translate input.pdf output.tif -config GDAL_PDF_LAYERS
"Map_Frame.Hydrography,Map_Frame.Terrain.Contours" -config GDAL_PDF_BANDS 3
-config GDAL_PDF_DPI 400
```

There are more options in the document linked above.

Note that gdal needs to have poppler available to be able to read GeoPDF, but the process seems to run fine using GDAL via the OSGeo4W Shell (section 1.6.2 on page 4).

Chapter 11

Using QGIS to create geological maps

Suggestions for using QGIS to create geological maps instead of ArcGIS. To be used in conjunction with the booklet “Introduction to GIS with ArcGIS: Drawing geological maps”. These are not complete step-by-step instructions but include links to other sources which will give you more information.

Note that this section is a work in progress. Updates will be posted in Minerva as part of updated full workbooks.

11.1 Obtaining QGIS

QGIS is an open source program and can be downloaded for free from:

`http://www.qgis.org`

It runs on Windows, Mac and Linux.

11.2 Finding and installing QGIS plugins

One of the strengths of QGIS is the wide variety of plugins that you can choose from in order to add extra functionality. The list of equivalent techniques in section 11.3 suggests that you install specific plugins in order to carry out some of the tasks that you will need to do.

To find and install plugins:

- Plugins > Manage and install plugins

The plugin manager should open with a view of all the available plugins - figure 11.1.

Not all will install successfully, but the ones that I have suggested below should be straightforward.

You can see which are already installed. Click on a plugin to see more information about what it does. Tick the box next to it and then click on Install plugin to add it to QGIS.

You can easily uninstall plugins in the same way if you don’t find them useful.

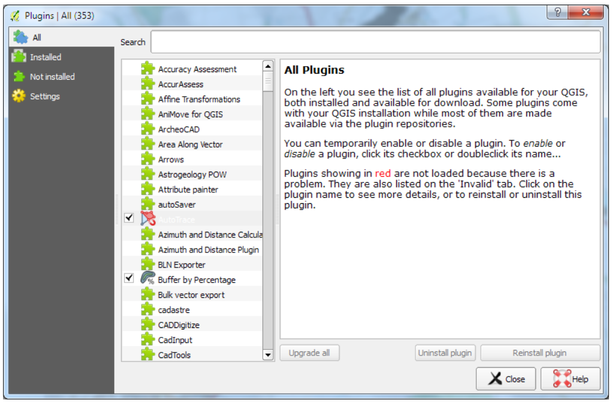


Figure 11.1: The QGIS plugin manager

Open Source GIS

Items in this section give information about alternatives to ArcGIS and other ESRI products. For example, QGIS.

Many of the examples in section 11.3 recommend that you consult the QGIS documentation. This is available from the page at

`http://www.qgis.org/en/docs/index.html`

From the list there select the **QGIS User Guide** which matches your version of QGIS[1].

The **QGIS Training Manual** is a work in progress[2] but already includes a lot of really useful instructions for carrying out various tasks in QGIS. It's enormous! So check out the table of contents.

There is also a **Gentle Introduction in GIS** which gives useful information about GIS principles, such as vector and raster data; attributes; collecting data; coordinate reference systems; and spatial analysis.

All three are available from the link above.

Video information

There are loads of helpful videos for QGIS on YouTube. The following presenters should get you started:

Hans van der Kwast: `https://www.youtube.com/channel/UCHGe8ccP2z2wHQa6B04m4Lg`

Klas Karlsson: `https://www.youtube.com/user/klakar70`

Other documentation

The **QGIS Tutorials and Tips** page at `http://www.qgistutorials.com/en/` has some really useful and clear tutorials.

Edina Digimap provide a brief tutorial in their resources. Go to `http://bit.ly/1N1kYoe` for the information on how to view data; apply styles; and define coordinate systems in QGIS.

[1] I haven't provided direct links, but given suggestions of search terms, so that you can find the correct information for any version of QGIS

[2] As of 17th March 2020.

Books

The books below are available from the Kennedy Library - see Clare in room 10.140b. They can be signed out for short-term loan (i.e. a few hours).

Bruy, A., Svidzinska, D. (2015) *QGIS by example* Birmingham: Packt Publishing.

  As it says! Real-world examples to introduce you to QGIS techniques.

Menke, K., Smith, R., Pirelli, L., Van Hoesen, J. (2015) *Mastering QGIS* Birmingham: Packt Publishing.

  The next step on from the book above with more in-depth information.

Marquez, A. (2015) *PostGIS Essentials* Birmingham: Packt Publishing.

  PostGIS is a database format - a useful alternative to ESRIs geodatabases!

11.3 Suggested equivalents

The table below shows the techniques that are part of creating a geological map in ArcGIS and their equivalents in QGIS. Where possible I have included links to tutorials and videos. Don't forget about the QGIS documentation either.

For an overview it is really worth watching the **QGIS 3 for Absolute Beginners** video by Klas Karlsson. It is basically an overview of what you need to do to create your map, though it doesn't include absolutely everything.

`https://youtu.be/kCnNWyl9qSE`

Table 11.1: Using QGIS instead of ArcGIS

ArcGIS technique	**QGIS equivalent**
The Catalog panel	QGIS has an integral browser (View > Panels > Browser) to view GIS files. To find out metadata for GIS layers use the layer properties - right-click on a layer then Properties. Look for the **Metadata** tab on the left, and the **Properties** box. This tells you, for example, the file type, and then more information depending on whether the file is raster (pixel (cell) size; Data type, e.g. 32 bit float, etc) or vector. Both types will include the layer's spatial reference system.
Georeferencing with the toolbar	Use plugin manager to check that **Georeferencer GDAL** is installed and there is a tick in the box next to it. If not install it now and make sure it is activated. • Raster > Georeferencer > Georeferencer Instructions are in QGIS documentation - search for **Georeference** There are also clear instructions at `https://www.qgistutorials.com/en/docs/georeferencing_basics.html` Suggested video (YouTube) - `https://youtu.be/t1X1U8255Q0`

Continued on next page

Table 11.1 – *Continued from previous page*

ArcGIS technique	QGIS equivalent
Storage - shapefiles and geodatabases	Shapefiles - can be used in most GIS software, though have limitations (e.g. length of field names!) QGIS will open ESRI file geodatabases, so you can still use data you've created previously, but the native geodatabase format in QGIS is **Geopackage**, **.gpkg**. Advantages include being lightweight and portable (all data in one file). Also possible to save styles in database - useful for passing data and styles to other people/projects. Can also be opened in Arc. See instructions for using a geopackage at: `https://freegistutorial.com/how-to-create-a-geopackage-on-qgis-3-2/`
Adding xy data from Excel or csv	If you have a csv file, or save an Excel spreadsheet as csv, you can add xy data as follows: • Layer > Add Layer > Add delimited text layer... • Set the coordinate system under **Geometry Definition** in the dialog and check that the X and Y fields are correct • Check your data is going to be imported correctly with the preview under Sample data
ESRI Basemap layers	The equivalent in QGIS is the **QuickMapServices** plugin. Once you've installed it the menu item appears under Web > QuickMapServices. While the selection of data isn't as large, it is still useful when you want a quick background to display, or to check that your data is in the right location! Layers include OpenStreetMap (which can provide a street map) and Landsat. More layers can be added via Web > QuickMapServices > Settings > Add/Edit/Remove.
Creating new features by editing	In QGIS select in the contents the layer to which you want to add new features • Right-click and **Toggle editing** • Then click on the **Add ...feature** button on the toolbar You should then be able to create a new feature. Right-click to finish creating and remember to save your edits, though QGIS will remind you if you turn off editing. Do this by right-clicking on the layer again, and **Toggle editing**

Continued on next page

Table 11.1 – *Continued from previous page*

ArcGIS technique	**QGIS equivalent**
Trace tool	The trace tool is available from the Snapping toolbar. To open this go to View > Toolbars and tick next to the **Snapping toolbar**. You can then select the magnet symbol on the toolbar to enable snapping, then click on the right-hand button to **Enable tracing**.
Clipping and merging while editing	Turn on the **Advanced digitizing** and **Digitizing tools** toolbars by going to View > Toolbars and selecting them. To merge features - while editing select the multiple features that you want to merge then click the **Merge selected features** button on the Advanced digitizing toolbar. To clip features - while editing select the feature(s) to clip. From the Digitizing tools toolbar click on **Cut with polygon from another layer** and select the layer with which to clip.
Snapping	Go to Settings > Snapping Options.... The **Snap to** dropdown will let you switch snapping off completely, or select whether to snap to vertices or segments or both. Use the layer selection to say which layers you want to select to (**Advanced** can be useful if you want to avoid snapping to contours, for example), and the **Tolerance** sets how close to a feature you need to be before you snap to it.
Labelling	Labelling in QGIS is developing rapidly at the moment. There are plenty of options, and possibilities include **Data defined settings** and **expressions**. Search the documentation for **Label Tool**. or look at the labelling information in the video (YouTube) at `https://youtu.be/kCnNWyl9qSE?t=910`
Labelling - moving labels manually (Annotation)	Setting up **Data-defined override** fields then allows you to use buttons on the Labelling toolbar to move, rotate and change labels manually as you may often need to for structural symbols. As in Arc do as much as you can by using the automatic settings first. For details go to the QGIS User Manual and look for the section on **Using data-defined override for labelling**. You will need to add 3 fields called `x`, `y` and `rotation`, all with a data type of **decimal number(real)** and then set these up as data-defined override fields in the label dialog. Start editing the layer and the labelling toolbar will allow you to move the labels around. Also look at the video at `https://youtu.be/1X2NLHhEUfQ` This technique doesn't allow you to move labels around in the layers tree - they will still appear on top of other layers.

Continued on next page

Table 11.1 – *Continued from previous page*

ArcGIS technique	**QGIS equivalent**
Rotation	QGIS will rotate symbols in a similar way. Search the documentation for **rotate point symbols** and **rotate features**
Adding extra symbols	If anything this is easier in QGIS than Arc. In the documentation look for **The Symbol Library**. Try creating the symbols that you need in Inkscape and save as .svg then add to QGIS as marker symbols (for points). It is also possible to create svg fills for polygons. It's also worth searching on-line for svg symbols that you can use, though many of the geological symbols will use USA conventions rather than UK.
Print layout	Print composer - search for **Print composer** in documentation. A lot of improvements to this in latest versions of QGIS (v.3 onwards)
Print size and reference scale	Not reference scale, but choose between mm, pixels and map units for symbology, points and map units for labels. Setting labels to **points** (the default) means that the text will appear the same size on screen at any zoom level. Setting labels to **map units** means that if you zoom out the text will appear smaller, if you zoom in it will appear larger. This makes it possible to control the print size at a particular scale. Setting feature sizes to **mm** (the default) or **pixels** means that the feature symbol will appear the same size on screen at any zoom level. Setting feature sizes to **map units** means that if you zoom out they will appear smaller, if you zoom in they will appear larger. Again, this makes it possible to control the print size at a particular scale.
Measured grid to show coordinates	Grid options for each map in a print composer layout. Possible to add grids for multiple coordinate systems - could be used to add e.g. British National Grid and WGS84 lat/long to the same map. More information in documentation for Print Composer.
Additional data frames and/or extent indicator	Use Overviews. More information in documentation for Print Composer. Need to add layers to existing map, then turn off layers you don't want to show. Create another map area on the layout to show small scale map, then **Lock layers for map item** on the map properties. You can then turn these layers off on the main map, and turn the geology etc layers back on and it won't change what shows on the overview. Use options under **Overviews** to set up extent.

Continued on next page

Table 11.1 – *Continued from previous page*

ArcGIS technique	**QGIS equivalent**
Viewing data in 3D with ArcScene	The QGIS plugin **Qgis2threejs** allows you to export any map which includes height data so that it can be viewed in any web browser. There are some fantastic examples online of projects that people have done with this plugin. Try the video on YouTube at `http://bit.ly/1BRt8cw`

Chapter 12

Ideas for teaching with Open Source GIS

Contents

Last Modified: 05 Dec 18, 11:12:51

Notes on how open source GIS could be used to replace Arc in teaching my modules

In general these instructions/notes work in version 2.18 of QGIS - the LTR as of May 2018. Some techniques do not work in version 3, mainly because plugins haven't been updated.

12.1 General questions and considerations

How would students find command line, if given full instructions? e.g. would be good to use GDAL/OGR, particularly for merging rasters!

Is OSGeo4W Shell available in clusters? or an alternative for running GDAL

Look at QGIS in clusters How is installing plugins handled?

Does GRASS toolbox work on clusters? May have to give instructions for setting up processing toolbox

At some point all ES students will be studying Python - can I use this somehow?

12.2 Geological Sciences

Using QGIS as a replacement for ArcDesktop for drawing geological maps is covered in section 11 on page 90.

12.2.1 Paso Dunes

EPSG: 32721

Processing > Toolbox > GDAL/OGR > GDAL Conversion > Rasterize (Vector to raster)

Fill in:

- Input layer = **XYpaso_dunes**
- Field to use for a burn-in value = `z`
- Output raster size units = **Pixels**
- Width/Horizontal resolution = **500** to start with
- Height/Vertical resolution = **500** to start with
- Output extent - Click ... > Use layer/canvas extent and choose input layer.
- Rasterized = ... > Save to file - choose output file location and name.

Set NoData value as transparent - Layer styling panel > Opacity tab - Fill in **Additional no data value** as `0`

Symbolise with Singleband pseudocolor - linear interpolation.

Terrain profile plugin - Save result as svg and alter proportions in Inkscape (and add title etc), then export to png. Can be added to a layout.

Map swipe tool - plugin

Quick Map Services - plugin. Satellite views from Bing, ESRI and Google. Would need instructions on how to include more than basic list? Go to Settings > More Services > Get Contributed Pack > Save then more services will appear. Try Web > Quick Map Services > ESRI > ESRI Satellite or Google > Google Satellite

Qgis2threejs plugin - to save view once in browser - Shift + S

12.2.2 Mt St Helens

Need to provide tif files, which can be downloaded from `http://bit.ly/2grwsLe`

Sort out coordinate system! Downloads from this site appear to be epsg:26710, but GDAL will only find epsg:26910, which is the one the previously downloaded files used. But, if I put this in, it does work, so stick with 26910?

For the students change the tif files to epsg:26910 and give them those to prevent confusion!

Hillshade: Processing > Toolbox > GDAL/OGR > GDAL Analysis > Hillshade

Can't find a way to create vector TINs but this isn't essential.

To work out 2D and 3D area try **r.surf.area** from GRASS toolbox. Output is html and/or text!

Presumably **Current region plan area** is 2D area and **Estimated region Surface Area** is 3D area?

To show difference in volume - Processing Toolbox > SAGA > Raster Calculus > Raster Difference

After running look at layer properties > Metadata > Properties and make a note of the **STATISTICS_MAXIMUM** and **STATISTICS_MINIMUM**.

Put the max and min into the layer styling max and min and the difference between the layers becomes visible.

Work out how to symbolise this sensibly as some rogue pixels make it difficult! May just need to experiment. Then is it possible to use this to produce an actual change of volume? Try the following:

- Processing toolbox > QGIS geoalgorithms > Vector general tools > Polygon from layer extent
- **Zonal Statistics plugin** - and set a prefix for the output
- Statistics will appear in attribute table of extent shapefile. The units given are pixels - so the cellsize of your raster data.

OR

- Processing toolbox > SAGA > Raster calculus > Raster Volum
- Oddly, doesn't seem to produce output unless you run it as a batch process, then volume is output in Log.

Profile is not an issue - use Profile plugin. Poss best if create a line shapefile and put line in that - not least keeps it on map once plugin is closed. Also poss to add two raster layers so that can show both profiles at once.

Save profile as svg or pdf and open in Inkscape to add title, labels, key etc. Then export to png to add to layout.

For layout - two dataframes?

Viewing in 3D - QGIS2threejs - no problem

12.2.3 Cononish

Layer › Add Layer › Add delimited text layer... - input needs to be csv so **Save as...** csv in Excel.

Then have to set layer CRS as doesn't allow CRS input, and comes out undefined. So right-click, Save As... and set CRS for new file - epsg:27700 for this exercise.

Interpolation plugin - only IDW seems to work? Is that just this data? Could be, have been able to use TIN with gravity data, and will also output a vector TIN if you click on the unmarked button at upper right of the dialog!

Interpolation plugin doesn't seem to work terribly well - even more inclined to crash than grass tool?

Processing › Toolbox › SAGA › Shapes - Tools › Shapes buffer (fixed distance)

BGS styles - Poss to download BGS 50k geology qml files from Digimap - `http://digimap.edina.ac.uk/webhelp/geology/using_data_with_gis/legend_files.htm` then apply in Properties › Styles . Styles cover all UK rock types so need to be thinned down for legends.

Finish instructions for applying qml files

Merging DTMs with QGIS gui? Output is usually better from command line GDAL - more options. But I can't imagine that most students will appreciate this.

12.2.4 Coastal erosion

Mainly uses websites etc. Adding rasters, then digitising - creating new shapefiles? or spatialite? Editing is easy.

Then presentation - including an Excel spreadsheet? Can add an attribute table to a Print Composer. Not very fancy!

12.3 GIS for GP and Meteorology (SOEE1160/2240)

Most will be similar to GS, but main thing is interpolation and creation of contours

12.3.1 Interpolation

Interpolation plugin - but rather inclined to crash.

If GRASS doesn't work try going to Processing › Options... › Providers › GRASS GIS 7 commands then double-click next to GRASS7 folder. If set up via OSGeo4W64 seems to be located at `C:` › `OSGeo4W64` › `apps` › `grass` › `grass-7.2.1` or whatever the number of the current version is.

Check what Grass folder is on cluster computers

GRASS **v.surf.idw** seems more reliable.

Processing toolbox › GRASS › Vector › v.surf.idw

Also try v.surf.bspline, v.surf.rst etc?

12.3.2 Creating contours from raster

Processing toolbox › GDAL/OGR › GDAL Extraction › Contour

12.4 GIS for Geoscientists (SOEE2650)

12.4.1 Projections and coordinate systems

Various issues with this Probably because QGIS does too good a job at dealing with on-the-fly reprojection!

projections - check how processing tasks work if undefined isn't obvious. e.g. try running assessment 2

Need to use rivers layer that isn't from ArcGIS Online, e.g. Natural Earth rivers vector data which has crs of epsg:4326

How does QGIS handle <undefined>? Displays as current **project** crs, but also shows that as crs in layer properties. So not possible to show undefined...

Also handles adding layers differently, so when I start with Monoscale, then add rivers, then river catchments, all line up perfectly. **which isn't what I want!** Other 2 combinations quite happily don't work...

Measuring distances - length is fine. Is there a way to measure area of polygon by clicking on it?

Use **Identify** which derives area. BUT doesn't work with just changing project crs as it does in Arc. Have to use **Reproject layer** tool and create a new layer. Then doesn't look any different on map (because OTF), but area is different.

12.4.2 GPS data

Can add gpx straight to QGIS, but then can't edit. Save as... to make editable. Can reproject at same time.

Somehow need to make it clear that if something shows on a map in GIS it isn't necessarily in the right format to use.

12.4.3 Elevation models

Instead of *Mosaic to new raster* etc

Use GDAL via Shell? Or stick to Processing Toolbox version of GDAL?

Toolbox GDAL -

- Build Virtual Raster - OK to add multiple tiles? Have to add to QGIS first. Untick Layer stack or it doesn't work as required.
- Warp (reproject) - seem to have to save as tif output
- Translate (convert format) - not necessary if have already saved as tif. Or can do this instead as it also allows reprojection
- Slope and aspect
- Hillshade
- Color relief - can't be used in further analysis, is just for display.
- possible to smooth dtm?

Or via menu -

- Raster > Miscellaneous > Build Virtual Raster (Catalog) - can set **Target SRS** as part of this
- Then Conversion > Translate (Convert format) to convert to GTiff
- Hillshade - uses **Raster Terrain Analysis plugin** then Raster > Terrain Analysis > Hillshade. Take up to top of table of contents and make transparent.

Use vector mask to limit processing? **Clip raster by mask layer** so only have heights for Lake District national park.

For Slope and aspect can change styles to only show one class.

Download csv version of DoBIH

Line of sight analysis

Select by location = Spatial Query plugin

Find = MMQGIS Search which selects feature when clicked on in result list.

Line of sight - Install Viewshed analysis plugin and use intervisibility. But need target points as well as observation points so isn't as simple as in Arc.

Viewshed Analysis plugin does do viewshed rasters and makes it possible to add offsets within the tool - which is simpler than Arc.

12.4.4 Layout and presentation

Covered under Geological Sciences?

12.4.5 Model building

If students are doing a compulsory python module could we cover python in GIS instead of model building?

12.4.6 Raster analysis

Creating single DEM covered above...

Raster properties

Right-click on layer in table of contents. Properties > Metadata tab.

Columns and Rows = Dimensions X and Y. Also Bands is there too.

cell size = pixel size

Data type, e.g. Thirty two bit floating point.

Pyramids not created automatically. To generate go to **Pyramids tab** in properties, select required Resolutions, then click **Build Pyramids** then shown as alternatives under Dimensions

Raster attribute tables? Not available. So need to find another way to find area of flooding etc.

try **GRASS - r.report** tool from Processing toolbox. Outputs a text report, so can't select particular categories in the attribute table, but does show the number of pixels of a particular value.

Raster statistics

Available in the properties - Metadata tab as shown above.

Limiting processing extent, cellsize etc

May need to be done on a tool-by-tool basis, e.g. possible to limit area in GDAL tools.

Interpolating a surface

Covered under Meteorology/Geophysics?

Watershed delineation

General method - suggests use GRASS tools

- Fill voids in DEM - **r.fill.dir** does both this and a flow direction raster
- run **r.watershed** -
  - the **Elevation** input being the filled DEM, don't select any other input
  - set **Minimum size of exterior watershed basin** to `5000`
  - save the outputs for stream segments and drainage direction to file - can save any of other outputs too, but these are the important ones.
- If required, then run **r.water.outlet** to produce a watershed area for a single stream/river
  - the **input raster map** is the drainage direction from above
  - for **Coordinates of outlet point (x, y)** click on the ... button at the end of the selection box and then click on the outlet of the required stream on the stream segments layer from above - be accurate about this!
  - then choose a file name to save the basin to

The output is a tif raster file with values of `nan` and `1` of which `1` is the watershed for that particular output area.

Raster to vector conversion

If a vector outline of a watershed is required run **Polygonize (raster to vector)** from the GDAL/OGR section of the Processing Toolbox.

Census data

joining table to shapefile/attribute table

Masks

Reclassifying rasters

- Raster calculator
- Reclassify tool

Proximity - raster

Replacing values in a raster with values from another raster

Zonal statistics

Flooding a raster - equivalent to SetNull in raster calculator?

Select by location - already covered?

Vector attribute table statistics?

12.4.7 Environmental decision making

12.4.8 3D

Covered under Geological Sciences?

Index

Todo list