QGIS 3.10 - Geo-analysis Practice

Lorenzo Amici

Oct 21, 2020

Contents:

This educational material has been developed by Eng. Lorenzo Amici, Dr. Daniele Oxoli, Prof. Maria A. Brovelli and Prof. Fabio Salice of Politecnico di Milano (Italy) with the mentorship and support of Dr. HaeKyong Kang of the Korea Research Institute for Human Settlements, within the project of collaboration between the OSGEO foundation and the UN Open GIS initiative.

Download PDF version

CHAPTER 1

General Info

The OSGeo UN Committee promotes the development and use of open source software that meets UN needs and supports the aims of the UN. Following a meeting between OSGeo Board of Directors and the UN GIS team at FOSS4G in Seoul, Korea in September 2015, the Committee has mainly worked on the UN Open GIS Initiative, a project "…to identify and develop an Open Source GIS bundle that meets the requirements of UN operations, taking full advantage of the expertise of mission partners including partner nations, technology contributing countries, international organisations, academia, NGOs, private sector. The strategic approach shall be developed with best and shared principles, standards and ownership in a prioritized manner that addresses capability gaps and needs without duplicating efforts of other Member States or entities. The UN Open GIS Initiative strategy shall collaboratively and cooperatively develop, validate, assess, migrate and implement sound technical capabilities with all the appropriate documentation and training that in the end provides a united effort to improve the effectiveness and efficiency of utilizing Open Source GIS around the world."

1.1 Purpose of this documentation

This educational material is designed as a step-by-step software learning guide for QGIS. It contains the description of some functions for environmental analysis of vector and raster data, and a hands-on example for each function described. The purpose of this quick start document is to introduce the user in the use of the main algorithms contained in QGIS and used in environmental analysis.

1.2 Target audience

The primary target audience for this documentations is intermediate QGIS users with a background of data analysis and management in a GIS environment

1.3 License

This educational material was written by Eng. Lorenzo Amici and Dr. Daniele Oxoli of Politecnico di Milano.

CHAPTER 2

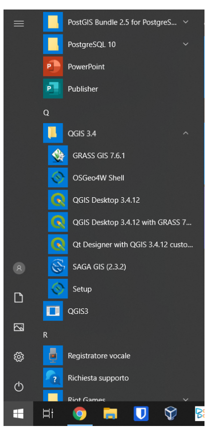
Preparation

In this section we will show all the necessary setup steps in order to follow along the QGIS exercise.

2.1 Install QGIS

Download QGIS 3.10 (Long Term Release) for Windows 64bit at https://qgis.org/downloads/QGIS-OSGeo4W-3.10.7-1-Setup-x86_64.exe. Other versions are available in the QGIS Download page.

Note: After downloading QGIS, you will have various option to launch the program, as you see in this image:



In order to complete this exercise be sure to launch the one with GRASS since we will need some functionalities that are only available using the tools that GRASS offers.

2.2 Download the data

The dataset used in the following exercises can be downloaded here; it covers an area around the city of Seoul in South Korea and contains three different type of information:

- Landsat8 images in the folder `LC08_L1TP_115034_20180721_20180731_01_T1`: it contains all the 11 available bands of the Landsat8 images in *WGS84/UTM zone 52N* coordinate system *EPSG:32652*.
- Aster Digital Elevation map (DTM at a resolution of 20 m) `Seoul_DTM.tif`: a geotif in LongLat WGS84 coordinate system *EPSG:4326*.
- OpenStreetMap vector dataset in the folder `south-korea-latest-free.shp`: these are 8 shapefiles downloaded from the OSM website. The files are in LongLat WGS84 coordinate system EPSG:4326.

Here we illustrate the `Data` folder tree structure:

```
Data
├── RASTER
│   ├── Landsat8_20160519_20170324_01_T1
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_ANG.txt
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B10.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B11.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B1.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B2.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B3.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B4.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B5.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B6.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B7.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B8.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_B9.TIF
│   │   ├── LC08_L1TP_116034_20160519_20170324_01_T1_BQA.TIF
│   │   └── LC08_L1TP_116034_20160519_20170324_01_T1_MTL.txt
│   └── Seoul_DTM.tif
└── VECTOR
    └── south-korea-latest-free.shp
        ├── buildings_a.cpg
        ├── buildings_a.dbf
        ├── buildings_a.prj
        ├── buildings_a.shp
        ├── buildings_a.shx
        ├── landuse_a.cpg
        ├── landuse_a.dbf
        ├── landuse_a.prj
        ├── landuse_a.shp
        ├── landuse_a.shx
        ├── natural.cpg
        ├── natural.dbf
        ├── natural.prj
        ├── natural.shp
        ├── natural.shx
        ├── places.cpg
        ├── places.dbf
        ├── places.prj
        ├── places.shp
        ├── places.shx
```

(continues on next page)

(continued from previous page)

```
        ├── pofw_a.cpg
        ├── pofw_a.dbf
        ├── pofw_a.prj
        ├── pofw_a.shp
        ├── pofw_a.shx
        ├── pois_a.cpg
        ├── pois_a.dbf
        ├── pois_a.prj
        ├── pois_a.shp
        ├── pois_a.shx
        ├── README
        ├── roads.cpg
        ├── roads.dbf
        ├── roads.prj
        ├── roads.shp
        ├── roads.shx
        ├── water_a.cpg
        ├── water_a.dbf
        ├── water_a.prj
        ├── water_a.shp
        └── water_a.shx
```

2.3 Create a new project

Before starting with the analysis of the data, we have to set up a new Project. To do so, run QGIS with GRASS and click on *New Project* (or press `Ctrl+N`).

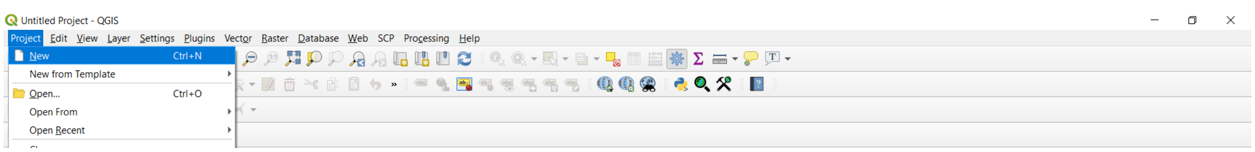


Fig. 2.3.1: Create a new project

2.3.1 Import vector data

To import vector data, go to *Layer->Add layer->Add vector layer* or use the Browser panel (usually placed above the Layer Panel on the left side of the screen; if not, you can enable it by clicking *View->Panels* and tick "Browser panel"). In the Browser, you can search the data folder and simply drag and drop the files in the map.

For this excercise, we will use the following layers:

- `buildings_a.shp`
- `landuse_a.shp`
- `natural.shp`
- `places.shp`
- `pofw_a.shp`
- `pois_a.shp`

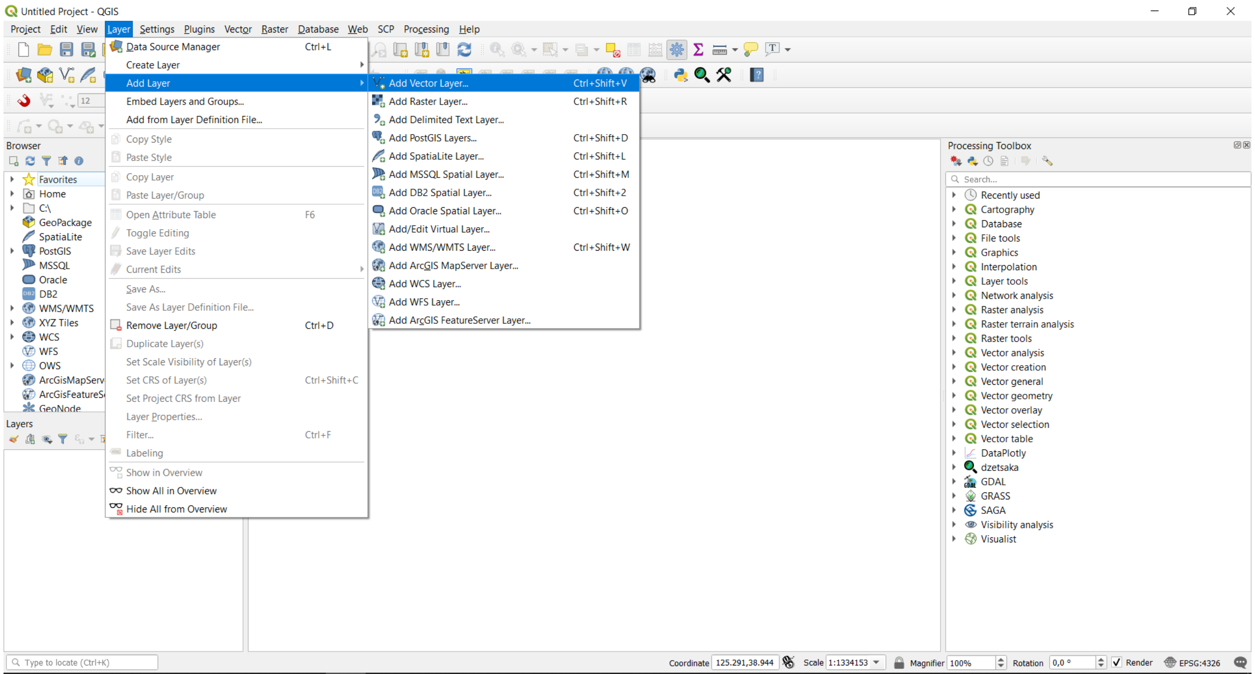


Fig. 2.3.1.1: Add a vector layer

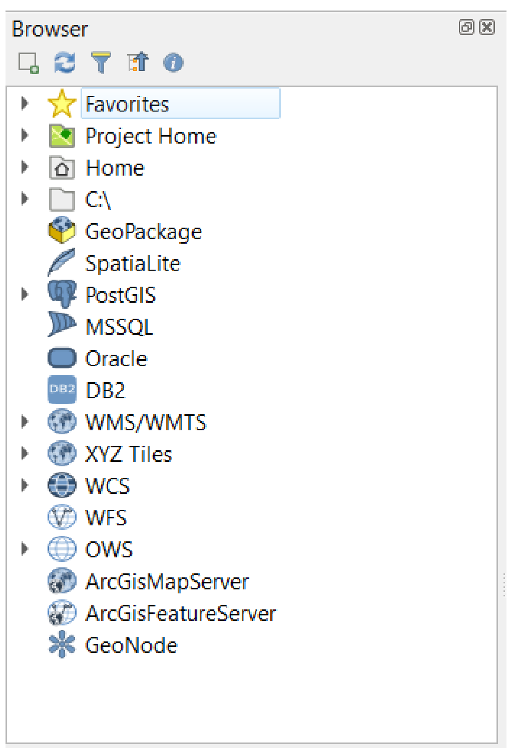


Fig. 2.3.1.2: The Browser panel

- `roads.shp`
- `water_a.shp`

2.3.2 Import raster data

To import raster data, you can go to *Layer->Add layer->Add raster layer* or drag and drop them from the Browser panel. For this exercise, you can add the following data:

- `Seoul_DTM.tif`

2.3.3 Layers panel

The Layers panel is a useful way to keep track of all the layers currently loaded in our Project. It's usually placed below the Browser panel, but if you don't have it activated you can do so by clicking on *View->Panels* and tick "Layers".

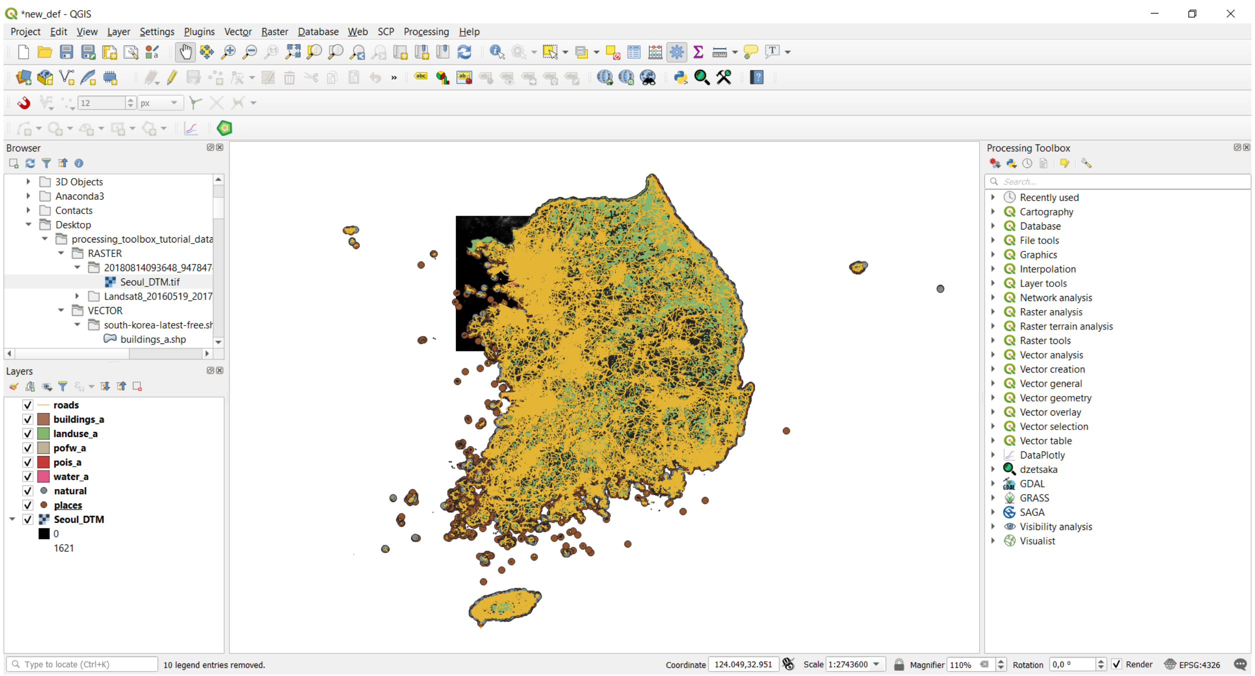
Once you add all the data it should look like this:

Fig. 2.3.3.1: The Layers panel

The Layers panel can also be used to choose which layers we want to see in our map: just by unticking a layer we hide it from our map. Also, we can choose the order of the layers, because layers can overlap and hide portions of each other, so it's important to decide which one is visualized on top of the others. To do so, you can select the layer you want to change and use the arrow symbols, or simply drag the layer in the position you want it to be.

Note: If the Project CRS (that can be seen in the bottom right corner of the window) is not specified, when importing the first layer QGIS will set the Project CRS to the CRS of that first layer. From then onwards, QGIS automatically reprojects any imported layer in the projection of the Project but only for visualization, the data remain in the original projection.

Once all the data are added, the map should look like this:



We can now save our project (*Project->Save* or `Ctrl+s`) so that you will have all the added layers and progress always available by just opening the saved project. Remember to do this from time to time during the exercise.

Note: When saving a project, QGIS creates a `.qgz` file, that represents the saved project. Note that this file does not contain directly all the added layers, but it records the path necessary to reach each one of the layers, enabling QGIS to include them when opening the saved project.

2.4 Manipulate CRS of the project and the data

The data we use in our geo-analyses often come from different sources, and therefore also have a different Coordinate Reference System (CRS). In order to homogenize the works and assure that all the tools work correctly, it is recommended to reproject all the data in the same CRS.

2.4.1 Reprojecting vector layers

This tool is available at *Vector->Data management tools->Reproject layer*. It provides a function that reprojects a vector layer, creating a new layer with the same features as the input one, but with geometries reprojected to a new CRS. The required input parameters are:

- *Input layer*: the vector layer to be reprojected (in the example :file'buildings_a')
- *Target CRS*: the target Coordinate Reference System. You can choose from the recent ones in the dropdown menu or click the icon to choose between all the available ones. Our choice will be *EPSG:32652 - WGS 84 / UTM zone 52N*

- *Reprojected*: the path and the name of the output raster layer. Note that if left empty a temporary layer will be created

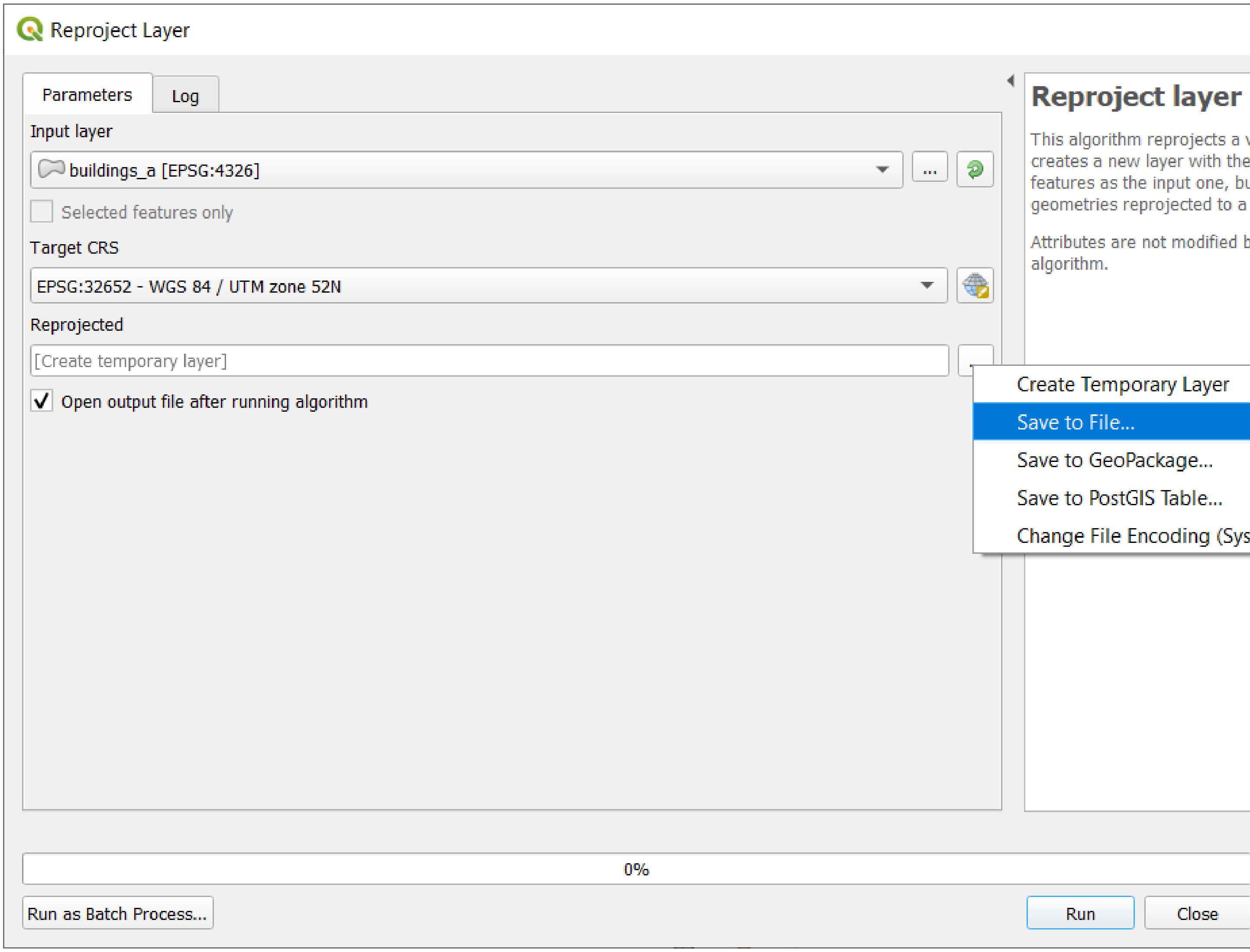


Fig. 2.4.1.1: Reproject layer function window

2.4.2 Reprojecting raster layers

Available at *Raster->Projections->Warp (reproject)*, it provides a function that reprojects a raster layer. The tool requires as input:

- *Input layer*: the raster layer to be reprojected (in the example the `Seoul_DTM`)
- *Target CRS*: the target Coordinate Reference System. You can choose from the recent ones in the dropdown menu or click the icon to choose between all the available ones. Our choice will be *EPSG:32652 - WGS 84 / UTM zone 52N*
- *Resampling method to use*: the method to be used for resampling the data, we will use the *Nearest Neighbor*

- *Reprojected*: the path and the name of the output raster layer. Note that if left empty a temporary layer will be created

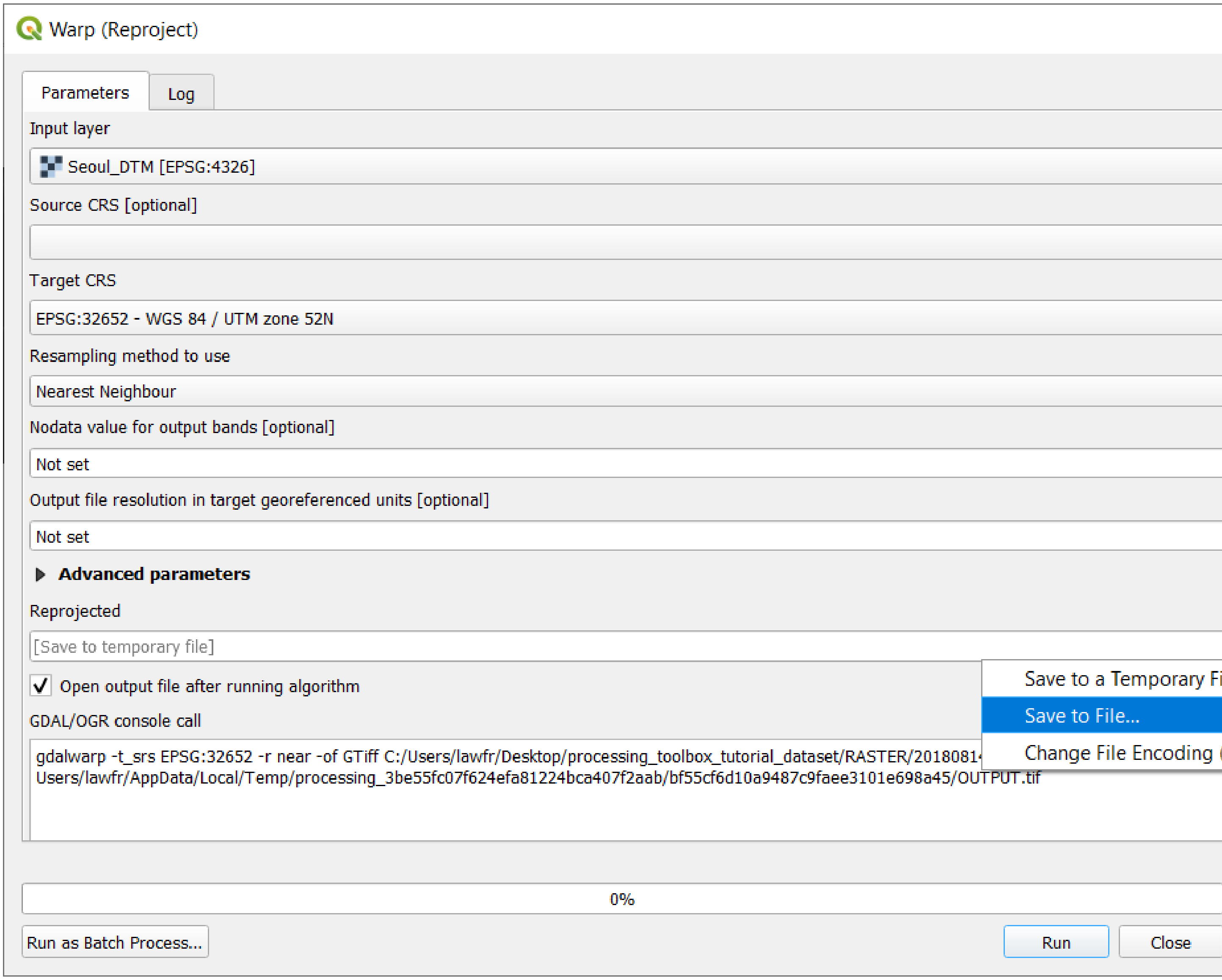


Fig. 2.4.2.1: Reproject raster function window

In order to continue with the tutorial please reproject all the imported vector and raster layers following the examples above. Once all the layers are reprojected we need to check if the CRS of the project is the same as the data: to do so, click on the CRS setting button in the bottom right of the window and select also here *EPSG:32652 - WGS 84 / UTM zone 52N*.

Note: In this exercise we will refer to the reprojected layers as `originalname_rep`. You can create your new layers with the same name in order to be consistent with the naming.

Note: After reprojecting all the layers be sure to delete the original layers. You can do this by simply selecting the layers you want to delete in the Layer panel and then click on the Remove layer/Group button.

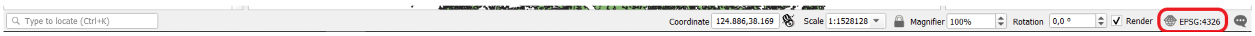
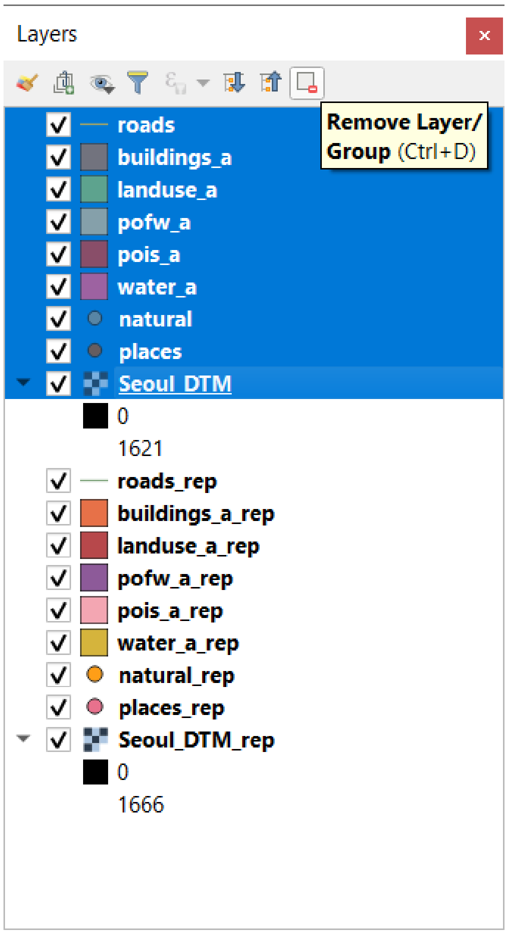


Fig. 2.4.2.2: You can see the Project CRS button highlighted in red

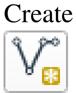


2.5 Clip all the data to the study area

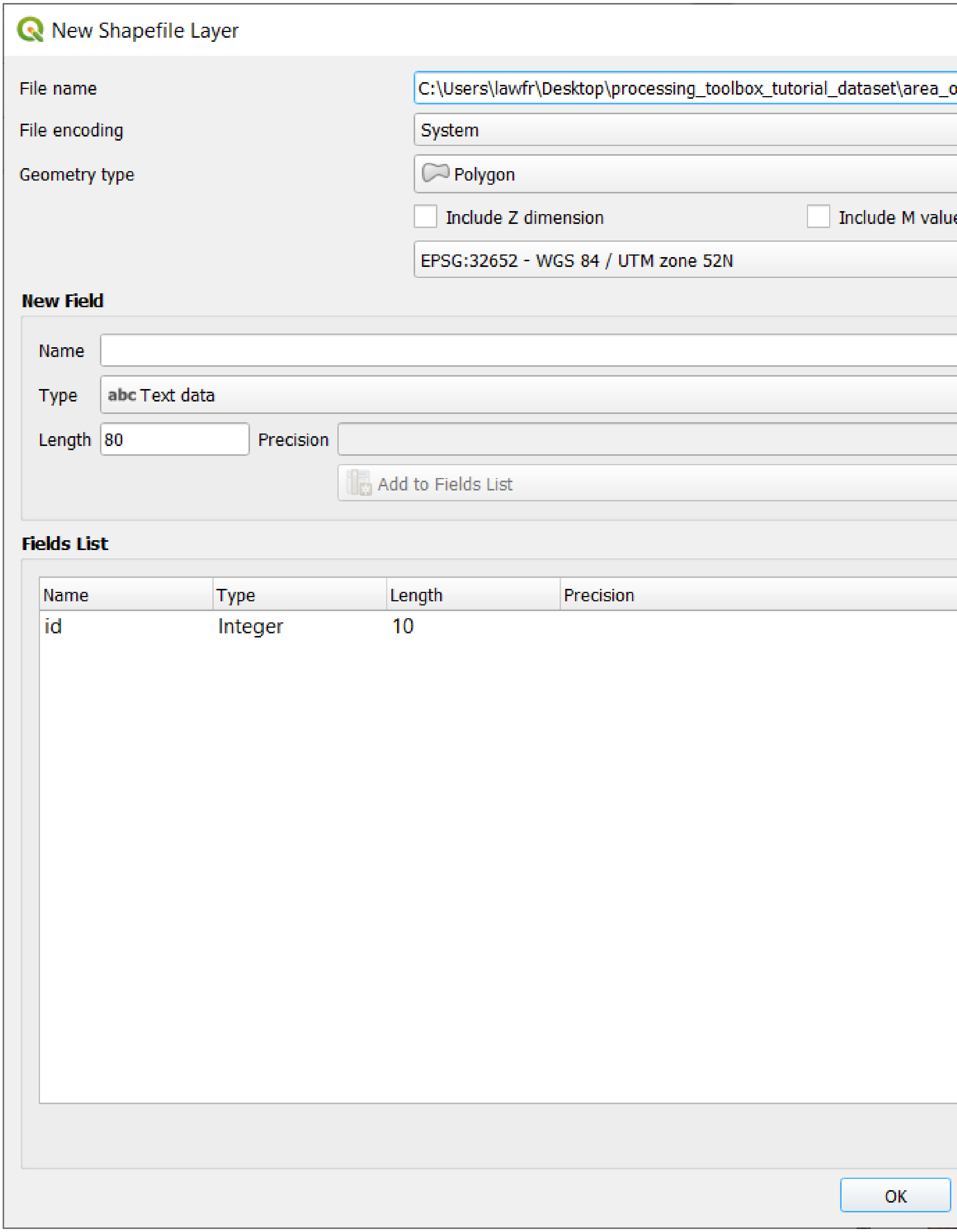
Often the data provided is in a wider area than the one needed, so we can define an area of interest and clip all the layers to that same area; in this way we shorten computational times and also provide consistency to the data.

2.5.1 Create the working area

We will create a vector polygon layer, in the shape of a rectangle, that will represent the area we are considering in this tutorial. In order to do so:

- Create a vector layer using *Layer->Create Layer->New shapefile layer* or use the shortcut symbol 
- Specify its characteristics as follows:
  - *File name*: `path/area_of_interest.shp`

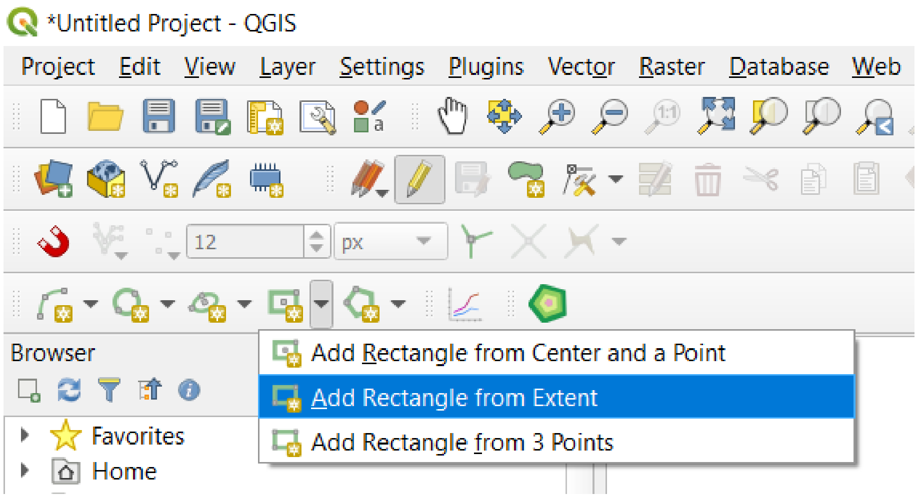
- *File encoding*: System
- *Geometry type*: Polygon
- The CRS should be by default set to the one of the maps but check it to be *EPSG:32652 - WGS 84 / UTM zone 52N*



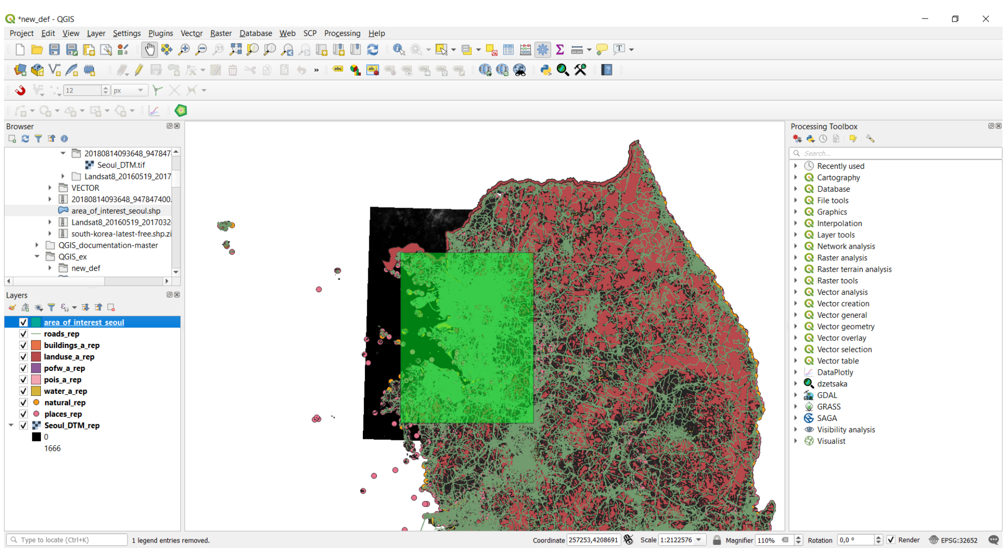
- Click "Ok"

Once you created the layer, you need to add the polygon representing the working area:

- Right click on `area_of_interest` in the Layers panel and select "Toggle editing"
- Right-click anywhere on the top toolbar (or go to *View->Toolbars* and add it from there) and enable the "Shape digitizing toolbar". This way we can add a perfect rectangle polygon to our shapefile
- Click on "Add rectangle from extent"



- Draw a rectangle in the area around Seoul by left-clicking to start drawing from an angle and then right-clicking when the size is the desired one (not too big, take the picture below as reference)



- Click on "Save layer edits" and toggle editing off

Now that you have your working area layer we can clip all our layers to it.

2.5.2 Clipping vector layers

Available at *Vector->Geoprocessing tools->Clip*. It provides an algorithm that clips a vector layer using the features of another polygon layer. Only the parts of the features in the Input layer that fall within the polygons of the Overlay layer will be added to the resulting layer. The attributes of the features are not modified, although properties such as area or length of the features will be modified by the clipping operation. The input parameters are:

- *Input layer*: the vector layer to be clipped (in the example `landuse_a_rep`). You can also choose to clip only the selected features of the vector layer if there are any
- *Overlay layer*: the `area_of_interest` layer
- *Clipped*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

2.5.3 Clipping raster layers

Available at *Raster->Extraction->Clip raster by mask layer*, it provides an algorithm that clips a raster layer using a vector layer as a mask. The input parameters are:

- *Input layer*: the raster layer to be clipped (in the example `Seoul_DTM_rep`)
- *Mask layer*: the `area_of_interest` layer
- *Clipped (extent)*: the path and the name of the output raster layer. Note that if left empty a temporary layer will be created

In order to continue with the tutorial please clip all the vector and the DTM raster layer following the examples above.

Note: In this exercise we will refer to the clipped layers as `originalname_clip`. You can create your new layers with the same name in order to be consistent with the naming

Note: After clipping all the layers you can delete the starting layers from the Layers panel.

The final configuration of the application and data should be like the one in the next image.

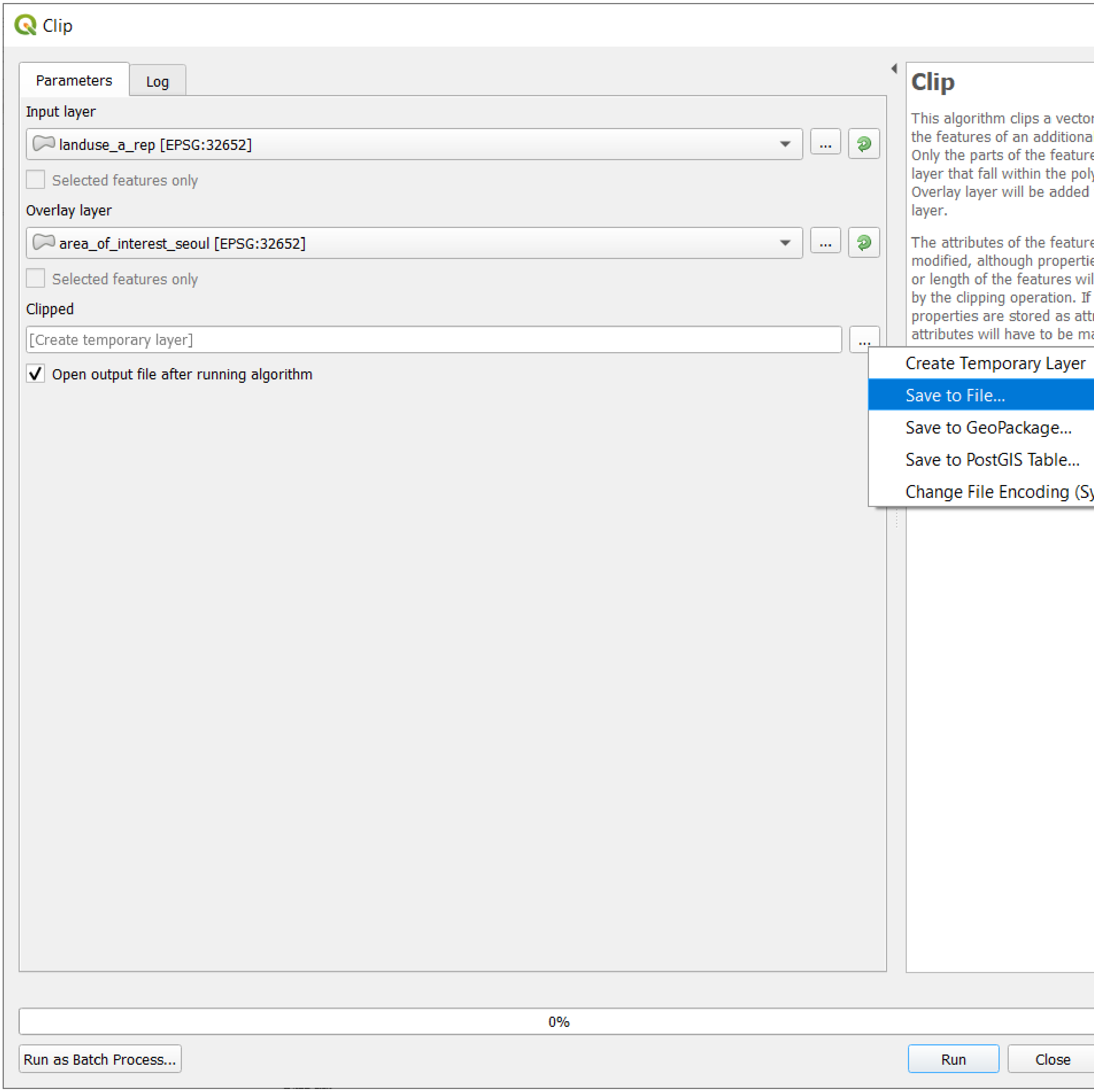


Fig. 2.5.2.1: Clip vector function layer

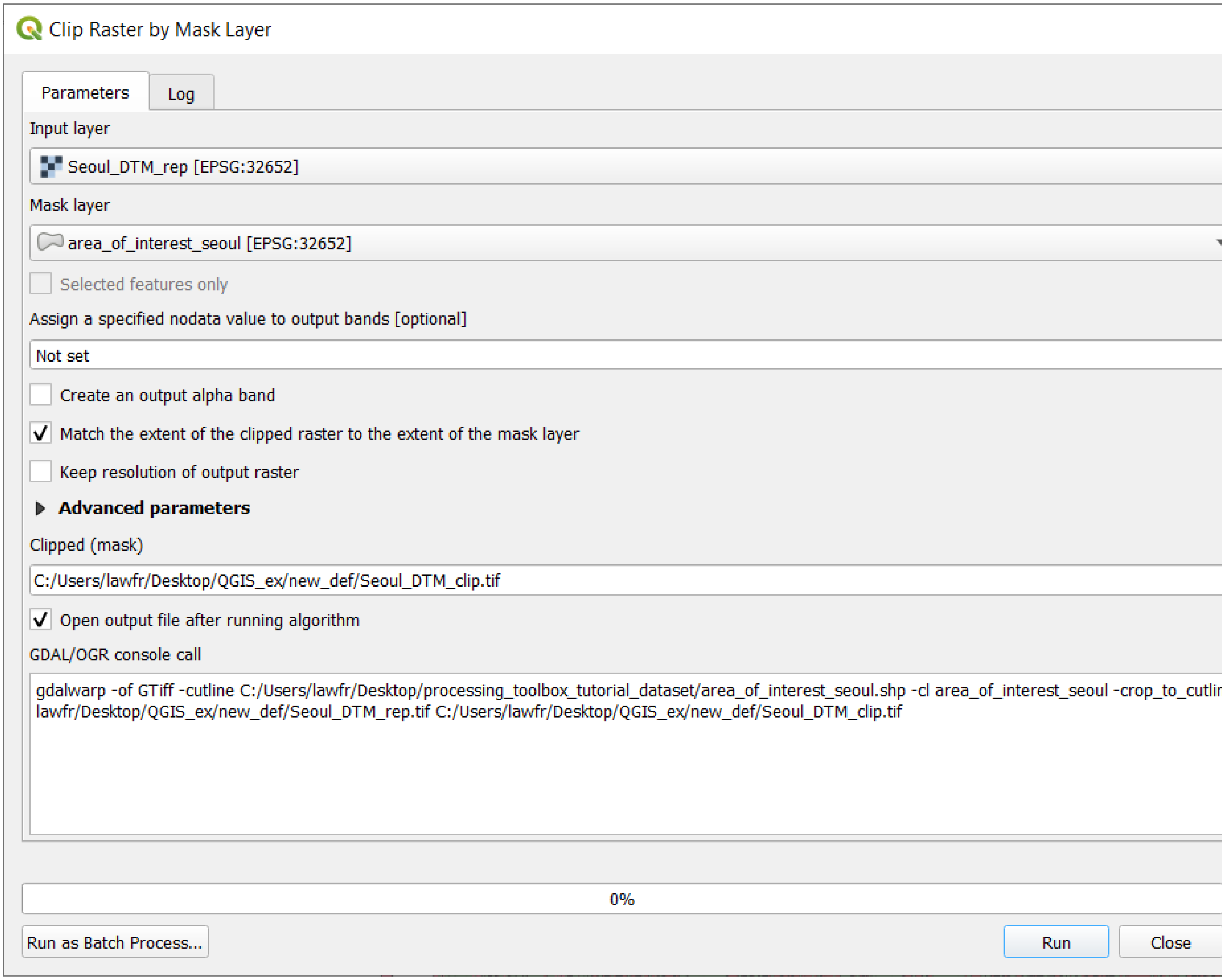
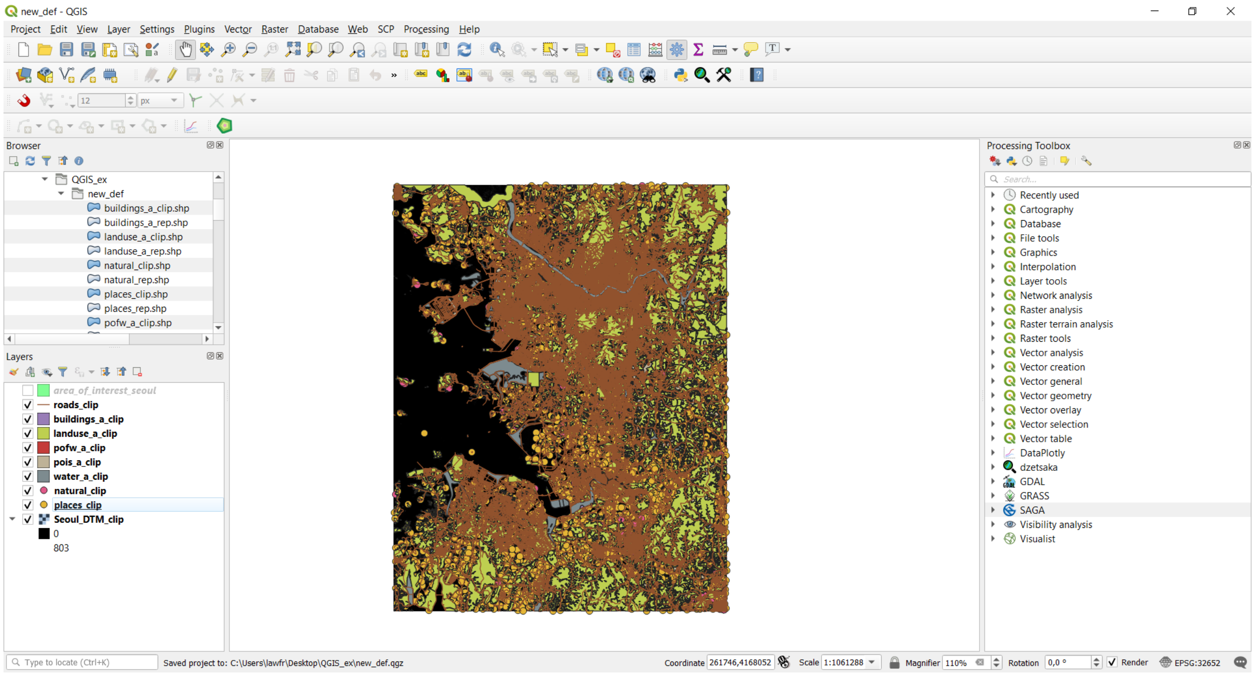


Fig. 2.5.3.1: Clip raster by mask layer function window

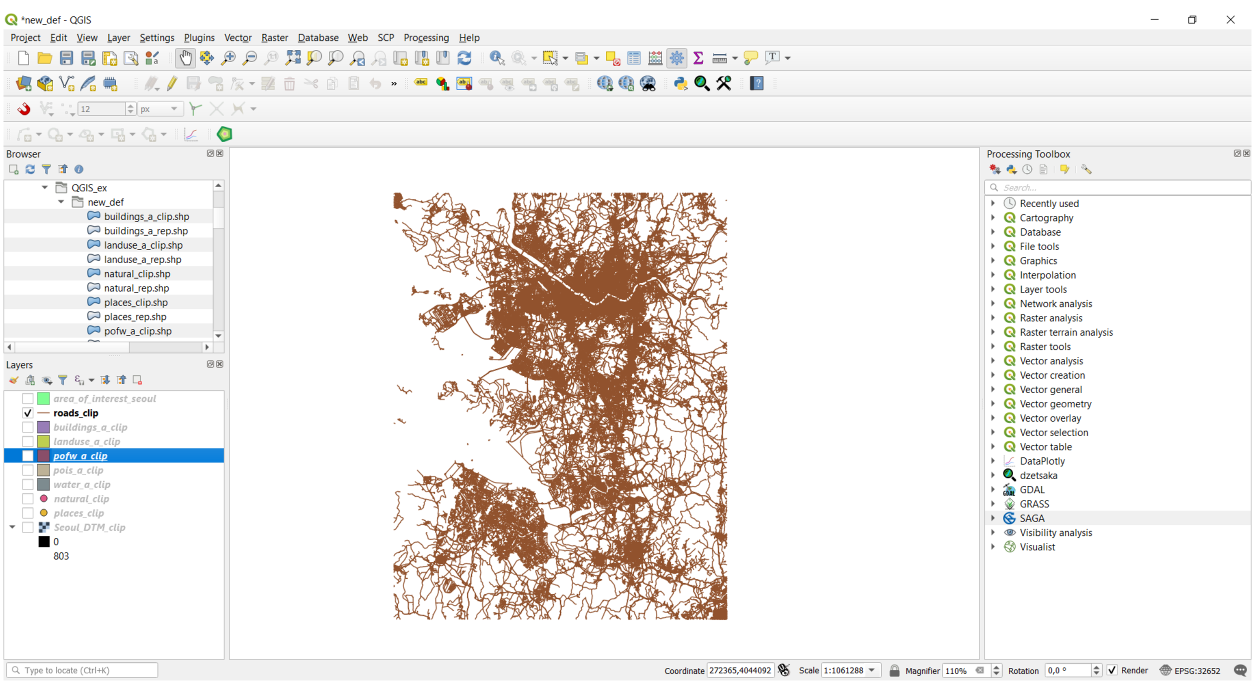

new_def - QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web SCP Processing Help
Browser
QGIS_ex
new_def
buildings_a_clip.shp
buildings_a_rep.shp
landuse_a_clip.shp
landuse_a_rep.shp
natural_clip.shp
natural_rep.shp
places_clip.shp
places_rep.shp
pofw_a_clip.shp
Layers
area_of_interest_seoul
roads_clip
buildings_a_clip
landuse_a_clip
pofw_a_clip
pois_a_clip
water_a_clip
natural_clip
places_clip
Seoul_DTM_clip
0
803
Processing Toolbox
Recently used
Cartography
Database
File tools
Graphics
Interpolation
Layer tools
Network analysis
Raster analysis
Raster terrain analysis
Raster tools
Vector analysis
Vector creation
Vector general
Vector geometry
Vector overlay
Vector selection
Vector table
DataPlotly
dzetsaka
GDAL
GRASS
SAGA
Visibility analysis
Visualist
Saved project to: C:\Users\lawfr\Desktop\QGIS_ex\new_def.qgz
Coordinate 261746,4168052
Scale 1:1061288
Magnifier 110%
Rotation 0,0 °
Render
EPSG:32652

CHAPTER 3

Vector operations

3.1 Buffer operations

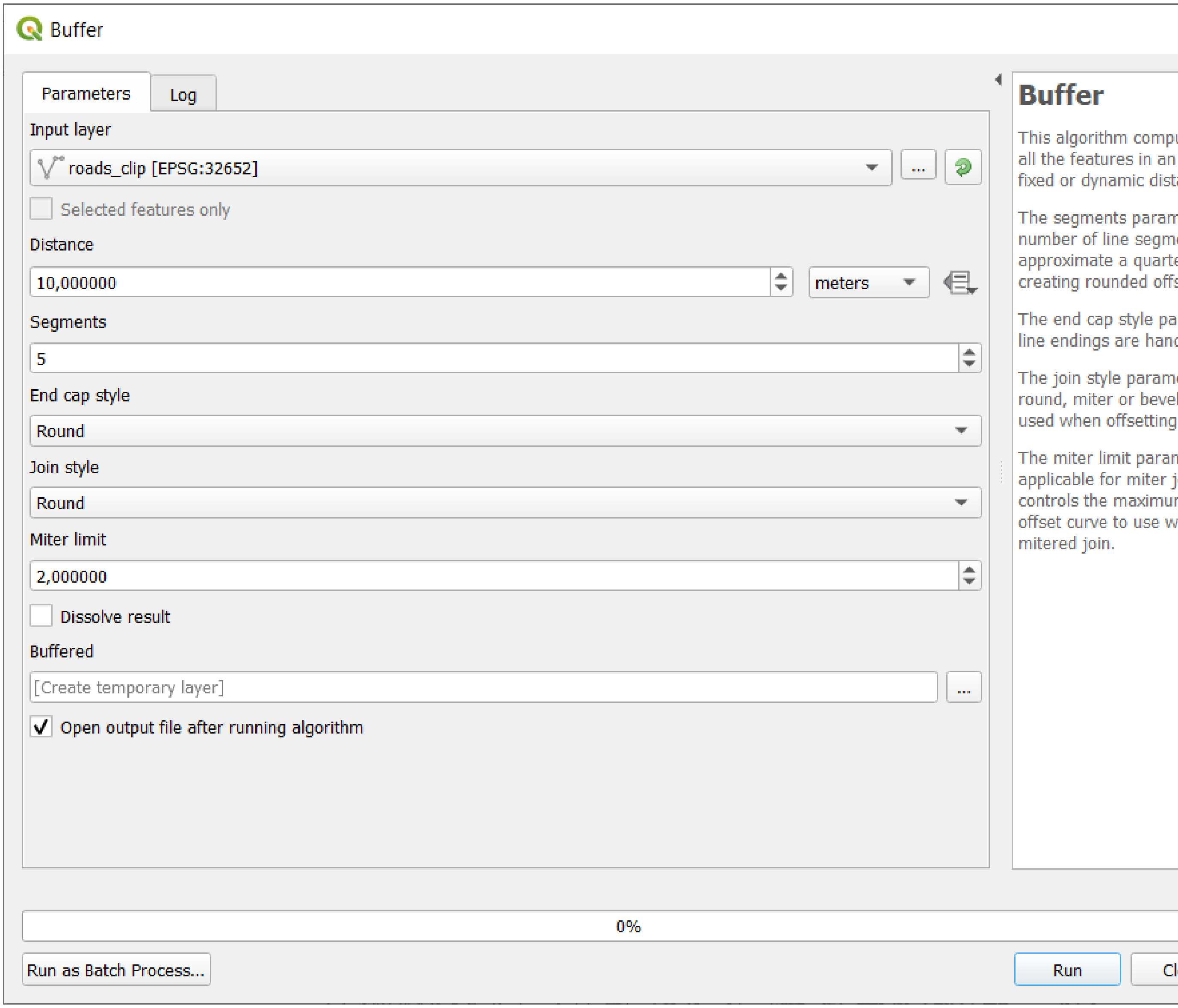
We will now focus on the analysis and manipulation of line shapefiles. To do so, let's consider only the `roads_clip` shapefile. You can turn-off all the other layers in the *Layers panel* to have a more compact visualization of the data.



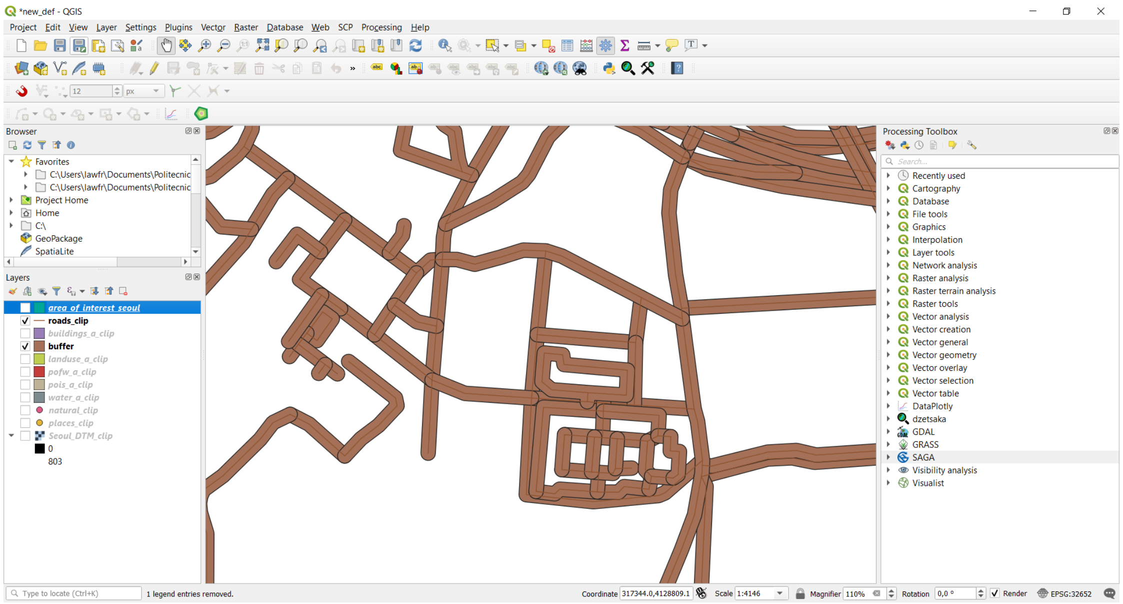
3.1.1 Single buffer

A very common operation with line shapefiles is the buffer, which allows to create an area within a specified distance from features. Note that in QGIS a buffer can be done also around point or polygon shapefiles. To create a buffer, click on *Vector->Geoprocessing Tools->Buffer* and specify the input parameters as follows:

- *Input layer*: the `roads_clip` layer
- *Distance*: the desired distance (in our case 10m). Note that you can also specify the unit of measure
- *Segments*: the number of line segments to be used to approximate a quarter circle when creating rounded offsets (in our case 5)
- *Buffered*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created



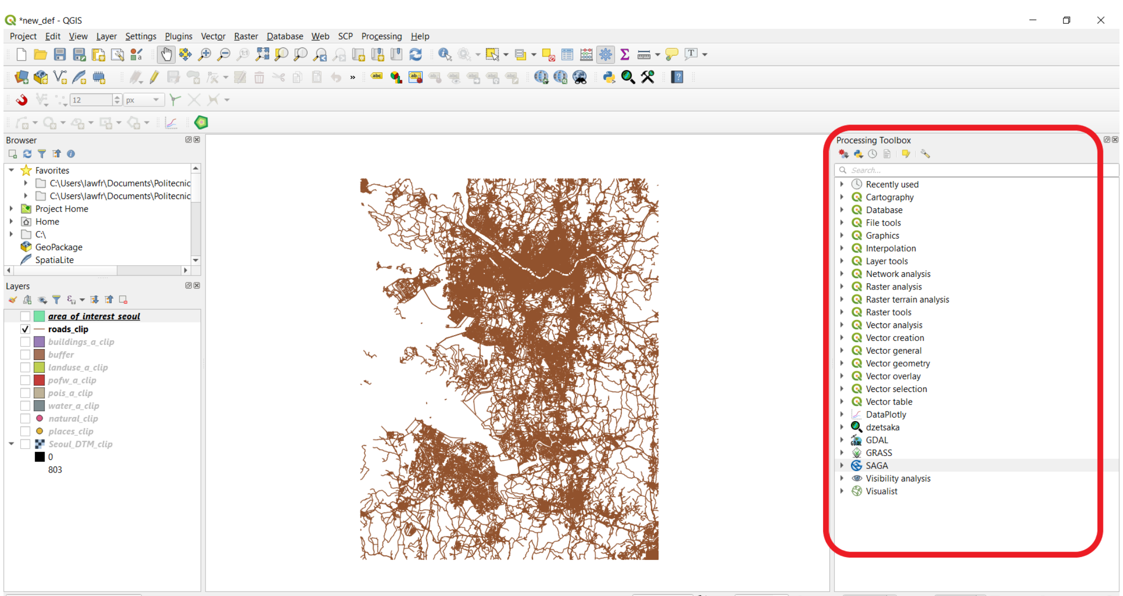
This will create a buffer area around our roads layer; if you zoom on your map, the result should look the following.



3.1.2 Multi-ring buffer

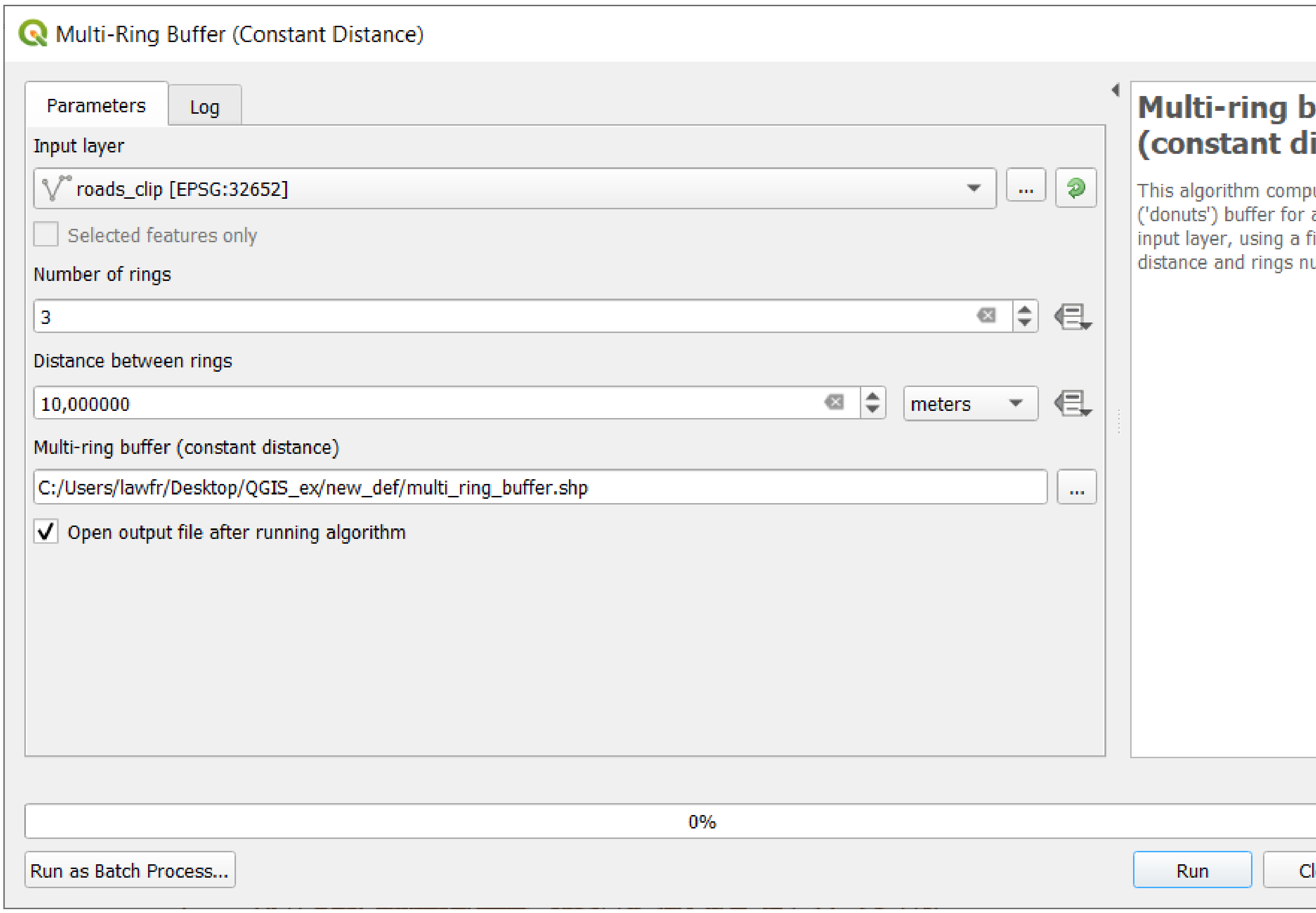
You can also create multiple buffers around the same features, within a specified distance.

To select the above function, we introduce here a very powerful instrument in QGIS: the Processing Toolbox. To add it to your window, go to *View->Panels* and tick "Processing Toolbox panel". It should appear on the right side of your screen; this panel contains all the functions of QGIS, and in particular, the search bar is very useful to find the function you need.

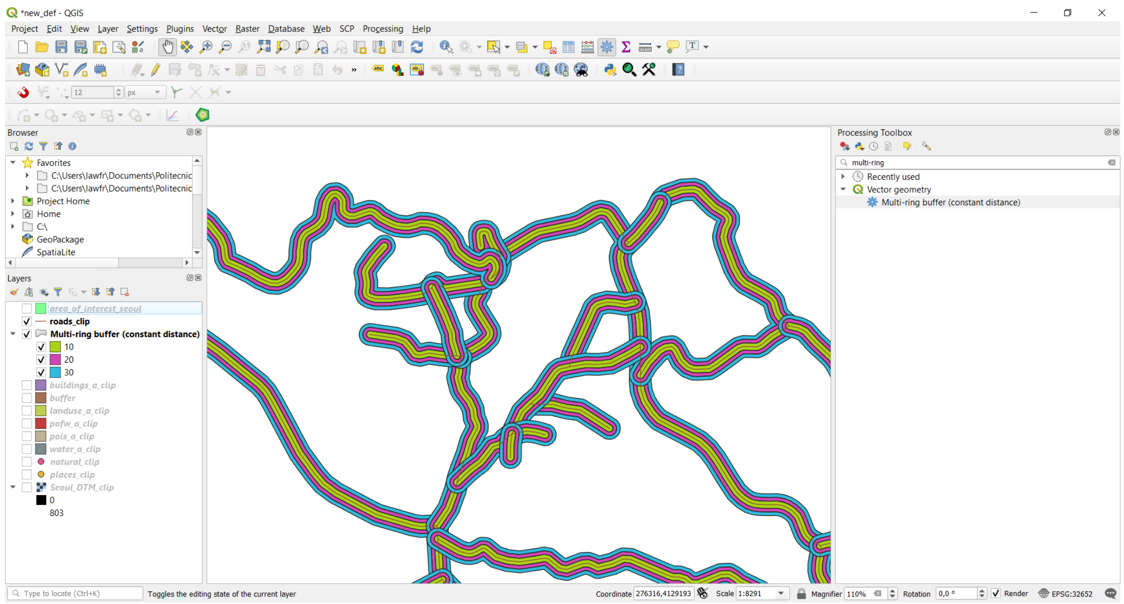


In this case, you can search for "Multi-ring" and select the function *Multi-ring buffer (constant distance)*. The input parameters are:

- *Input layer*: the `roads_clip` layer
- *Number of rings*: in our case 3
- *Distance between rings*: in our case 10m
- *Multi-ring buffer (constant distance)*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

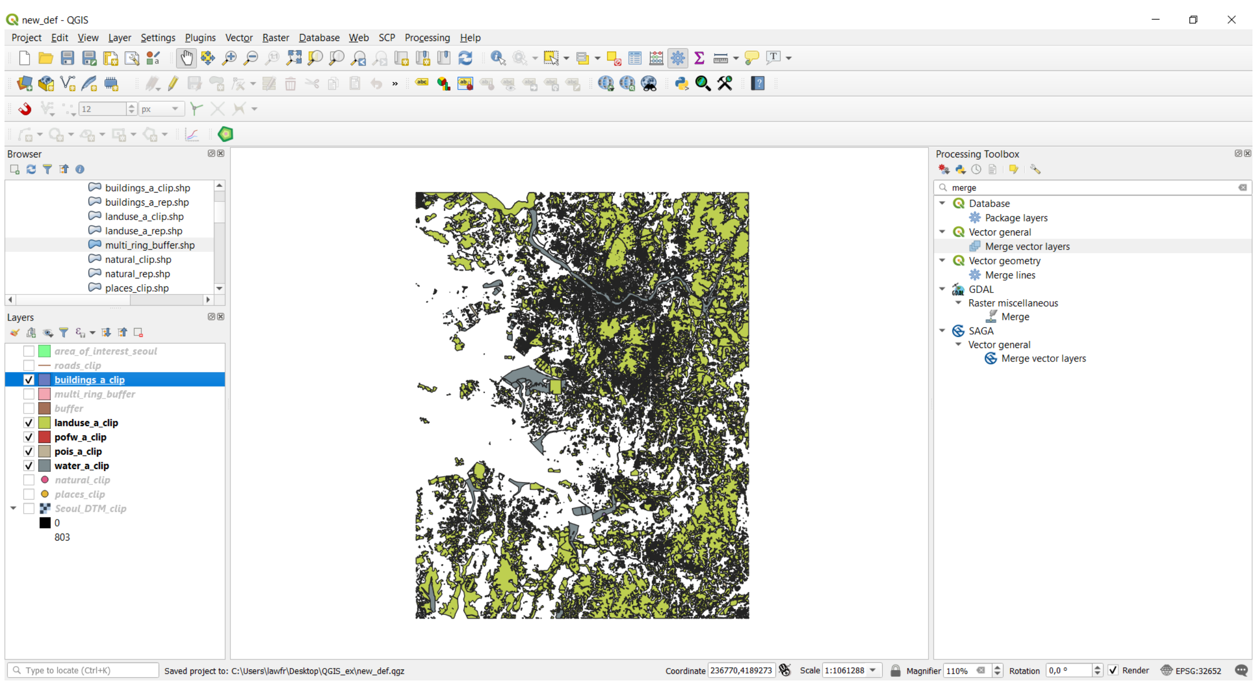


This will create three buffer areas around our roads layer; if you zoom on your map, and if properly styled, the result should look like this:



3.2 Merge vector layers

Let's now consider the polygons layers: in particular we will need the `buildings_clip`, `landuse_a_clip`, `pois_a_clip`, `pofw_a_clip` and `water_a_clip` layers, so you can hide all the others in the map.



The Merge vector layers function, available at *Processing Toolbox->Vector General->Merge vector layers*, provides an algorithm that combines multiple vector layers of the same geometry type into a single one. If the attributes tables are different, the attribute table of the resulting layer will contain the attributes from all input layers. New attributes will be added for the original layer name and source. We will use it to combine the `landuse_a_clip` and the `water_a_clip` in order to get a general layer describing both land and water features. To do so, the input parameters are:

- *Input layers*: click on the icon on the left and select `landuse_a_clip` and `water_a_clip`
- *Destination CRS*: select the Project CRS, that is *EPSG:32652 - WGS84 / UTM zone 52N*
- *Merged*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

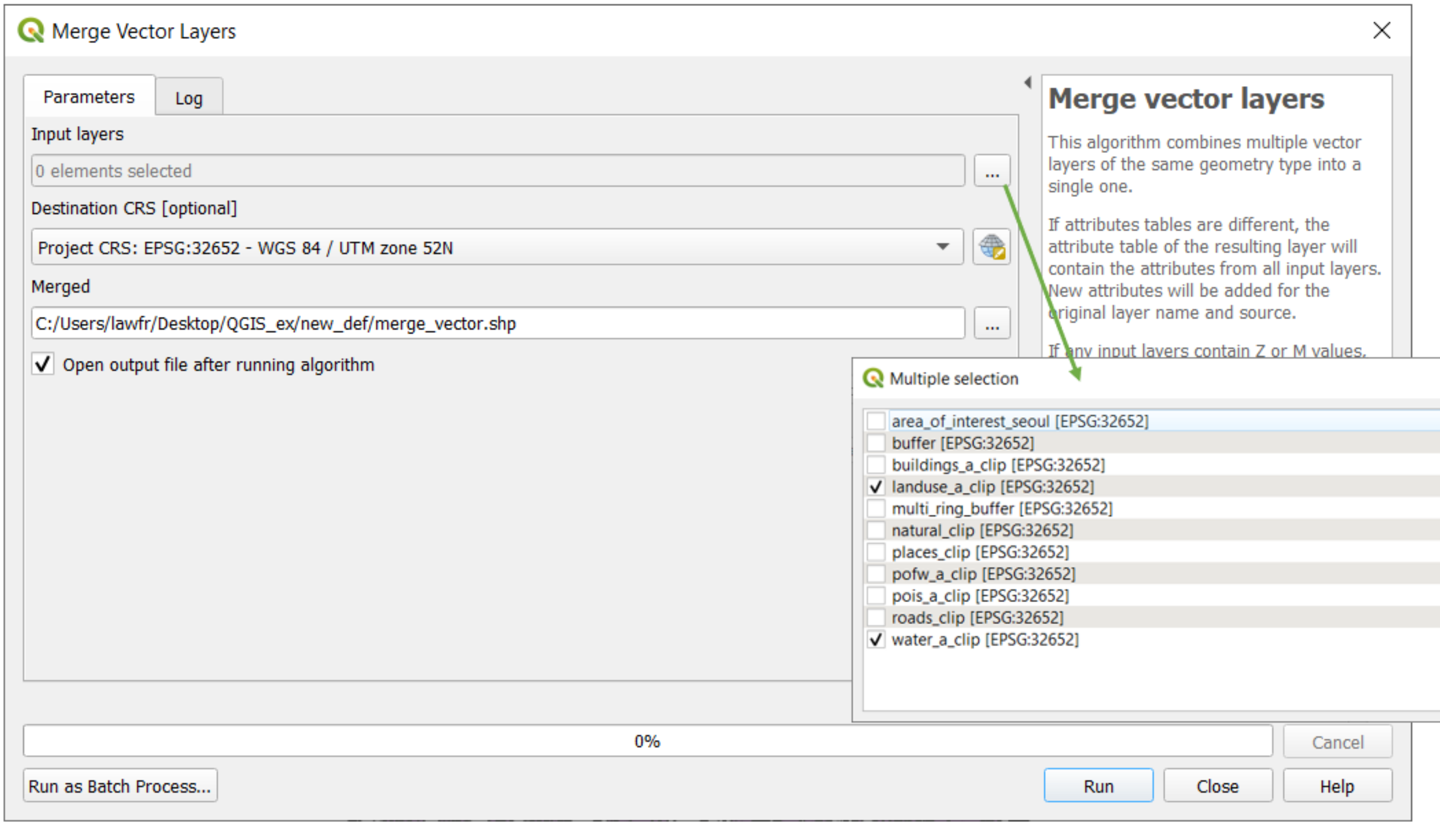
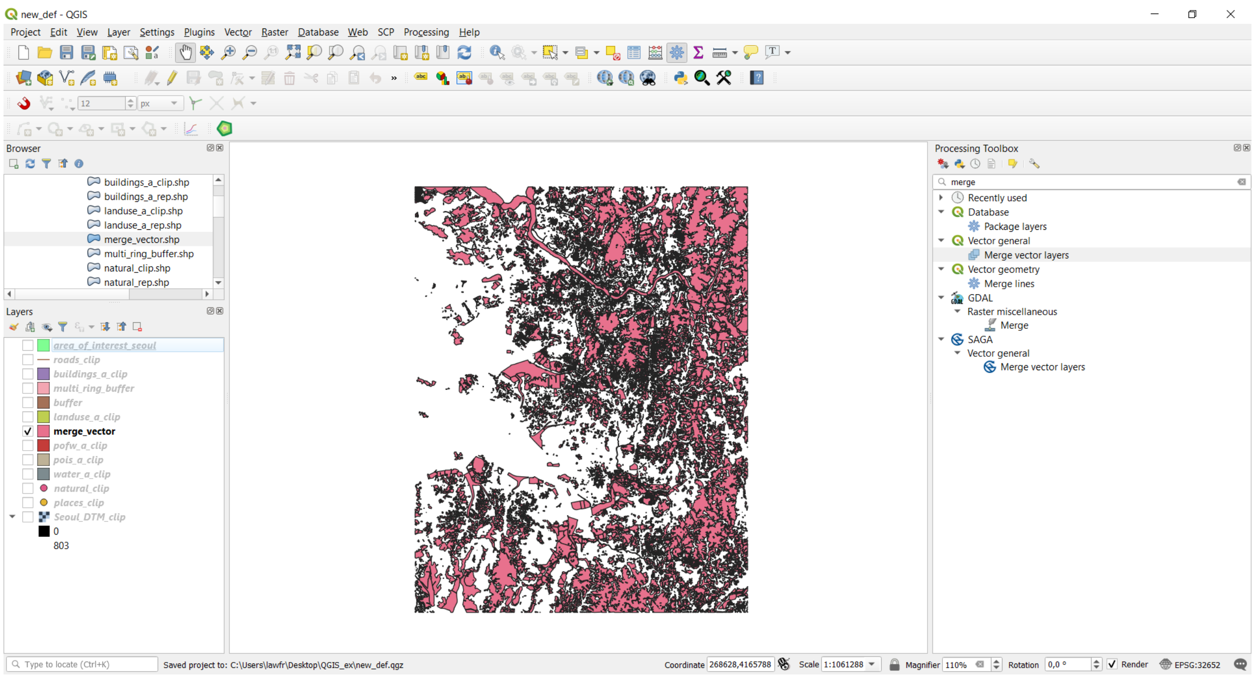


Fig. 3.2.1: Merge function window

As you can see this operation creates a new vector that contains features from both the landuse and water bodies layers:



3.3 Overlay operations

3.3.1 Union

The Union function, available at *Vector->Geoprocessing->Union*, provides an algorithm that checks overlapping features within an input layer and creates separate features for overlapping and non-overlapping parts. The area of overlap will create as many identical overlapping features as there are features that participate to that overlap.

In this exercise, we will perform an Union between the `pois_a_clip` layer (points of interest) and the `buildings_a_clip` layer, so that we will get a layer representing both residential buildings and interest/public buildings. The input parameters are:

- *Input layer*: `pois_a_clip` layer
- *Overlay layer*: `buildings_a_clip` layer
- *Union*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

The result, if zoomed, should look like the following:

3.3.2 Intersection

We can now look at the Intersection function. It is available at *Vector->Geoprocessing Tools->Intersection*, and it provides a function that extracts the overlapping portions of the features of two layers and assigns these portions the attribute of both layers. We use it to see which buildings are also religious buildings. The input parameters are:

- *Input layer*: `buildings_a_clip` layer
- *Overlay layer*: `pofw_a_clip` layer

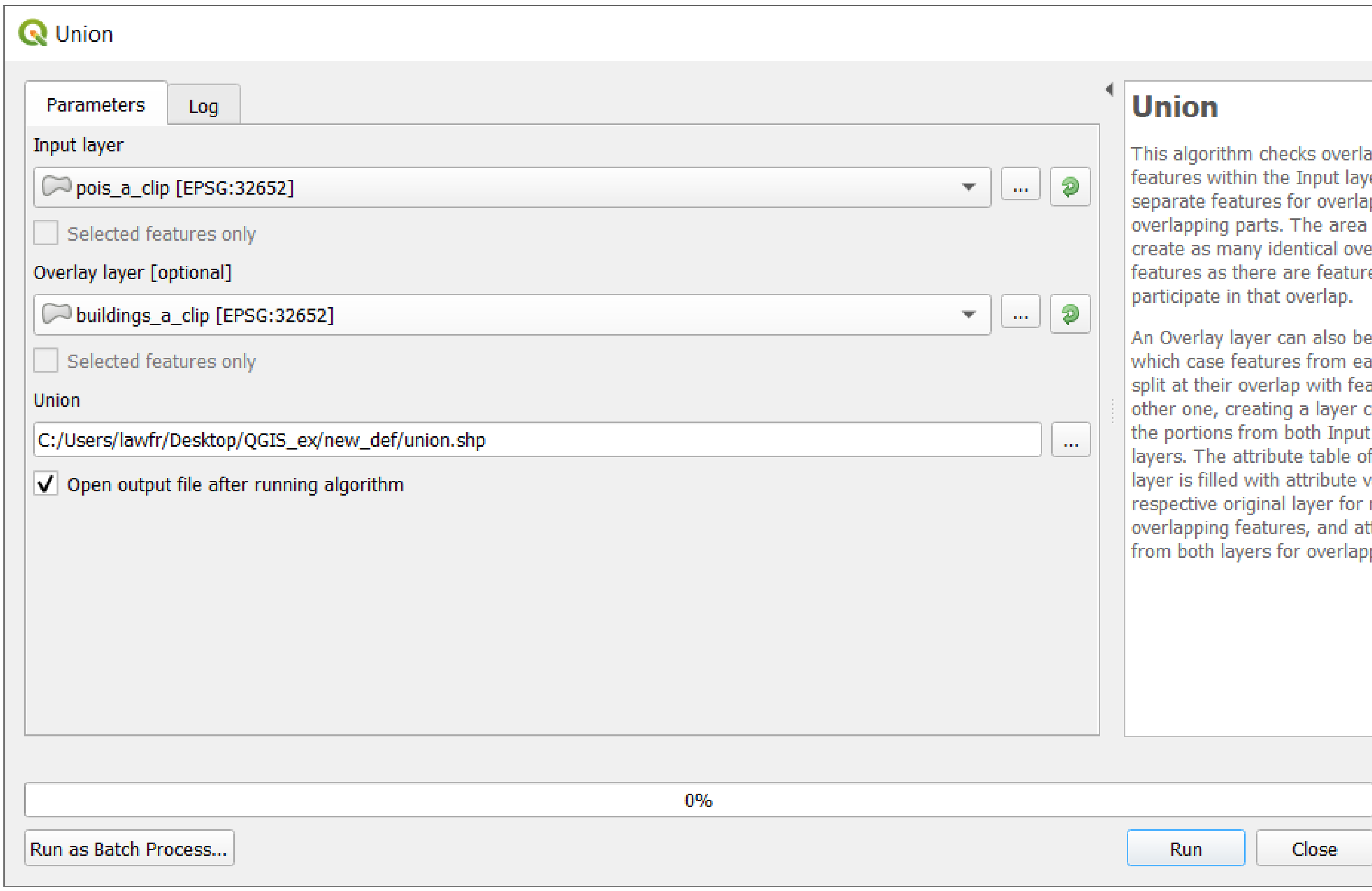


Fig. 3.3.1.1: Union function window

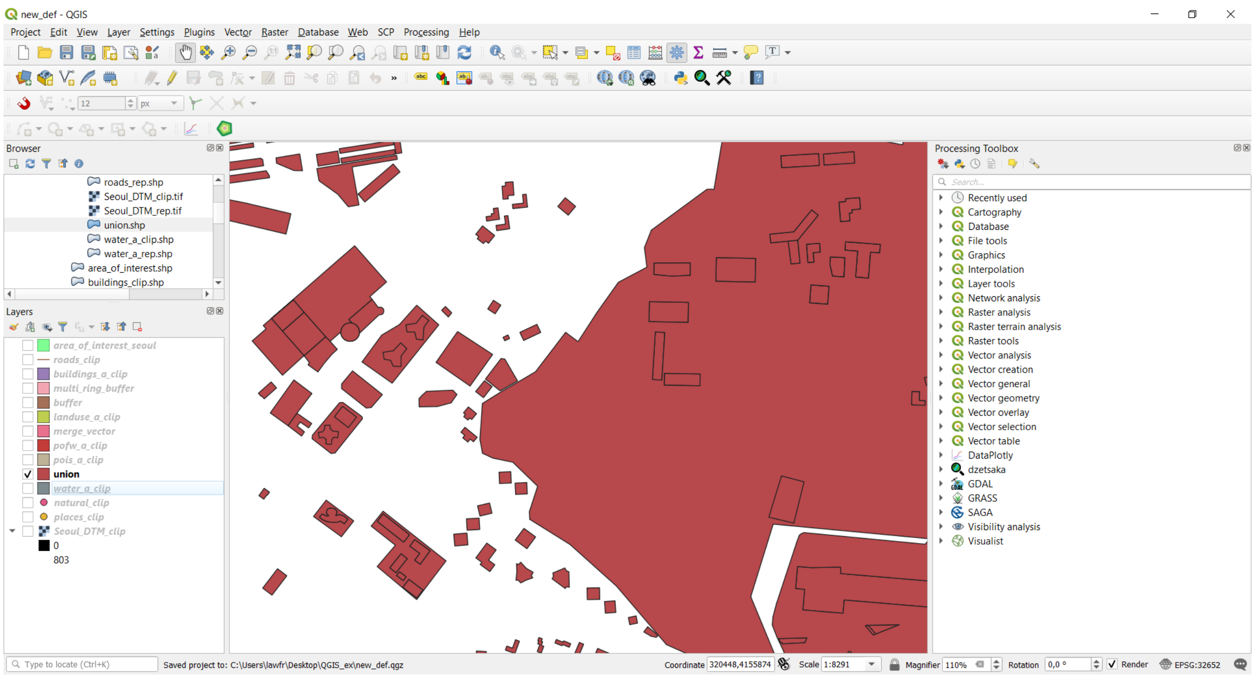


Fig. 3.3.1.2: As you can see now both residential and interest/public buildings are contained in the same vector layer

- *Intersection*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

The results should look like the following picture:

3.3.3 Dissolve

We will now use the Dissolve function applied to the `landuse_a_clip` layer. The Dissolve function, available at *Vector->Geoprocessing tools->Dissolve*, provides an algorithm that takes a vector layer and combines their features into new features. One or more attributes can be specified to dissolve features belonging to the same class (having the same value of a specific attribute); if no attribute is selected, all features will be dissolved into a single one. We will create a new landuse layer that has as many features as the different types of land use. The input parameters are:

- *Input layer*: the `landuse_a_clip` layer
- *Dissolve fields*: click the icon on the left and select the "fclass" attribute
- *Dissolved*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

If you now check the attribute table of the newly created layer it should have only a few features, each corresponding to one type of land use.

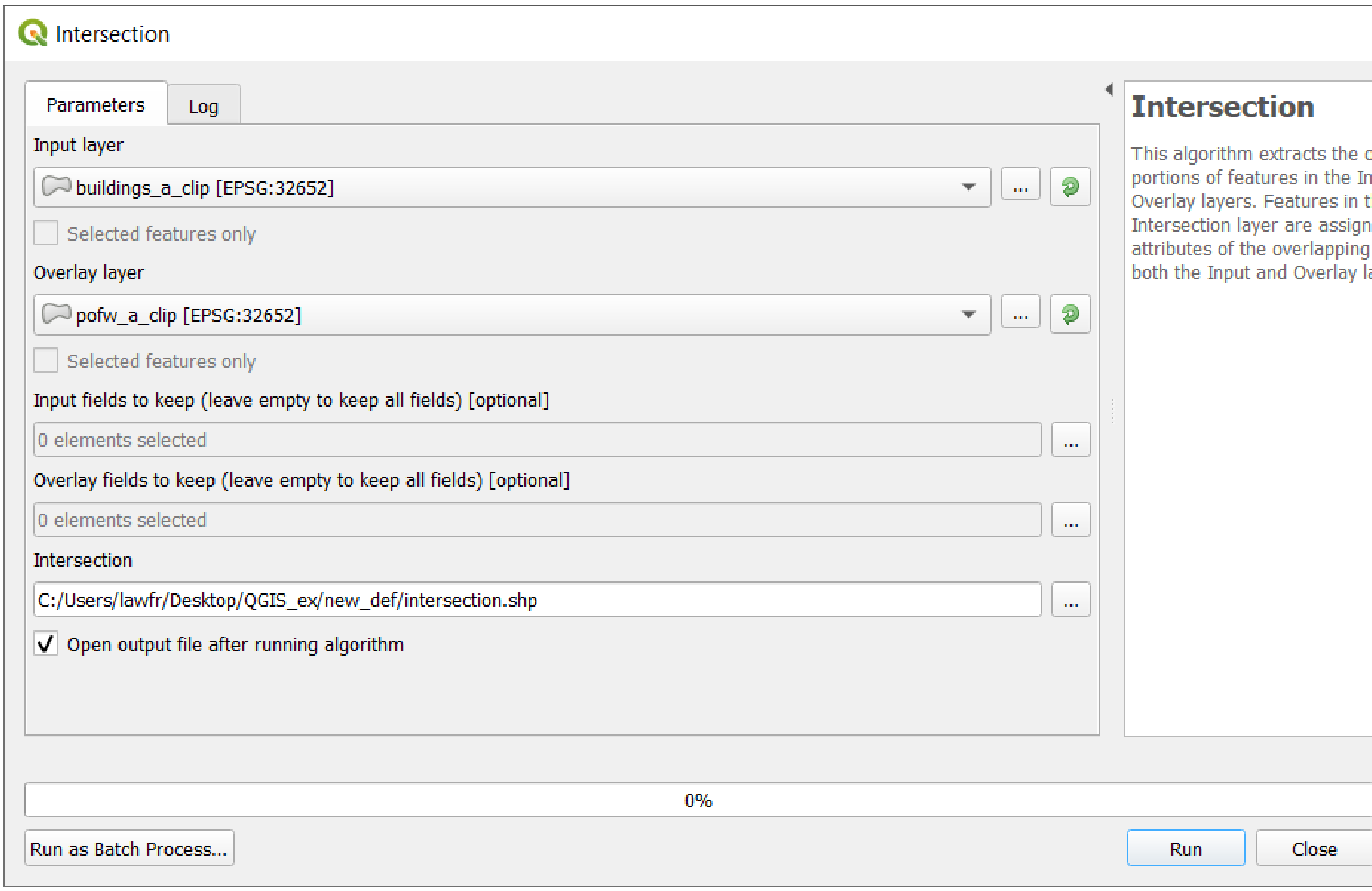


Fig. 3.3.2.1: Intersection function window

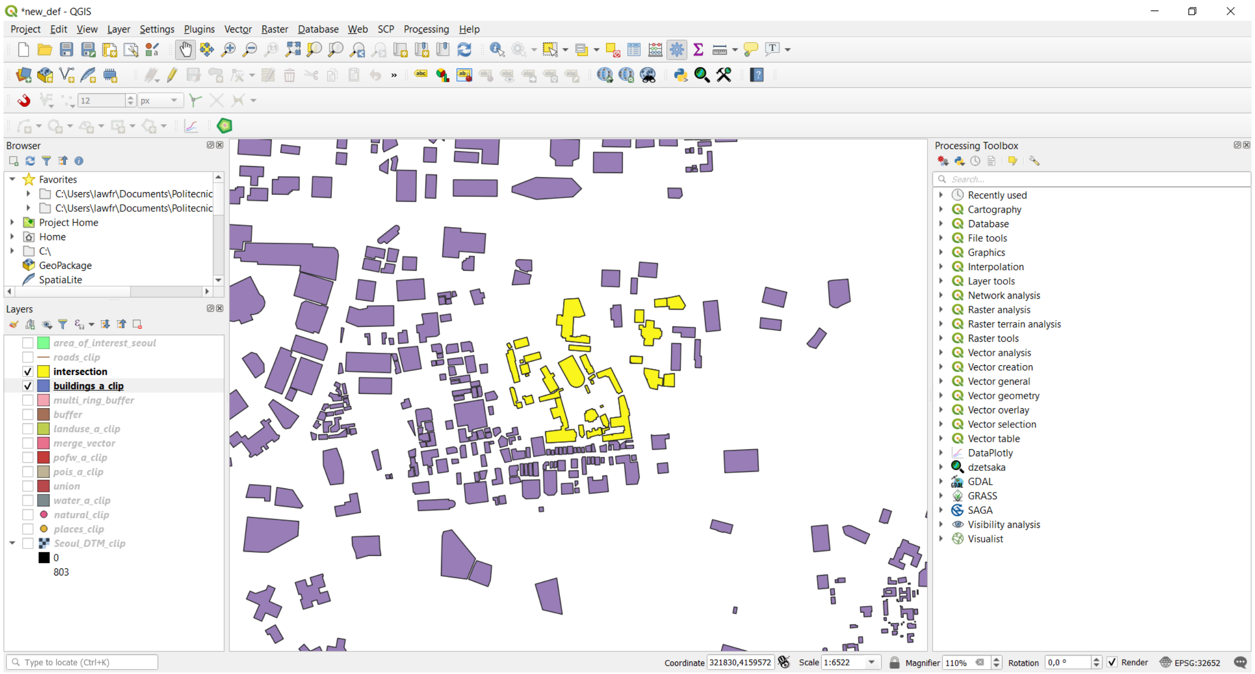


Fig. 3.3.2.2: As you can see, the intersection layer (yellow) only contains the buildings that also are religious buildings

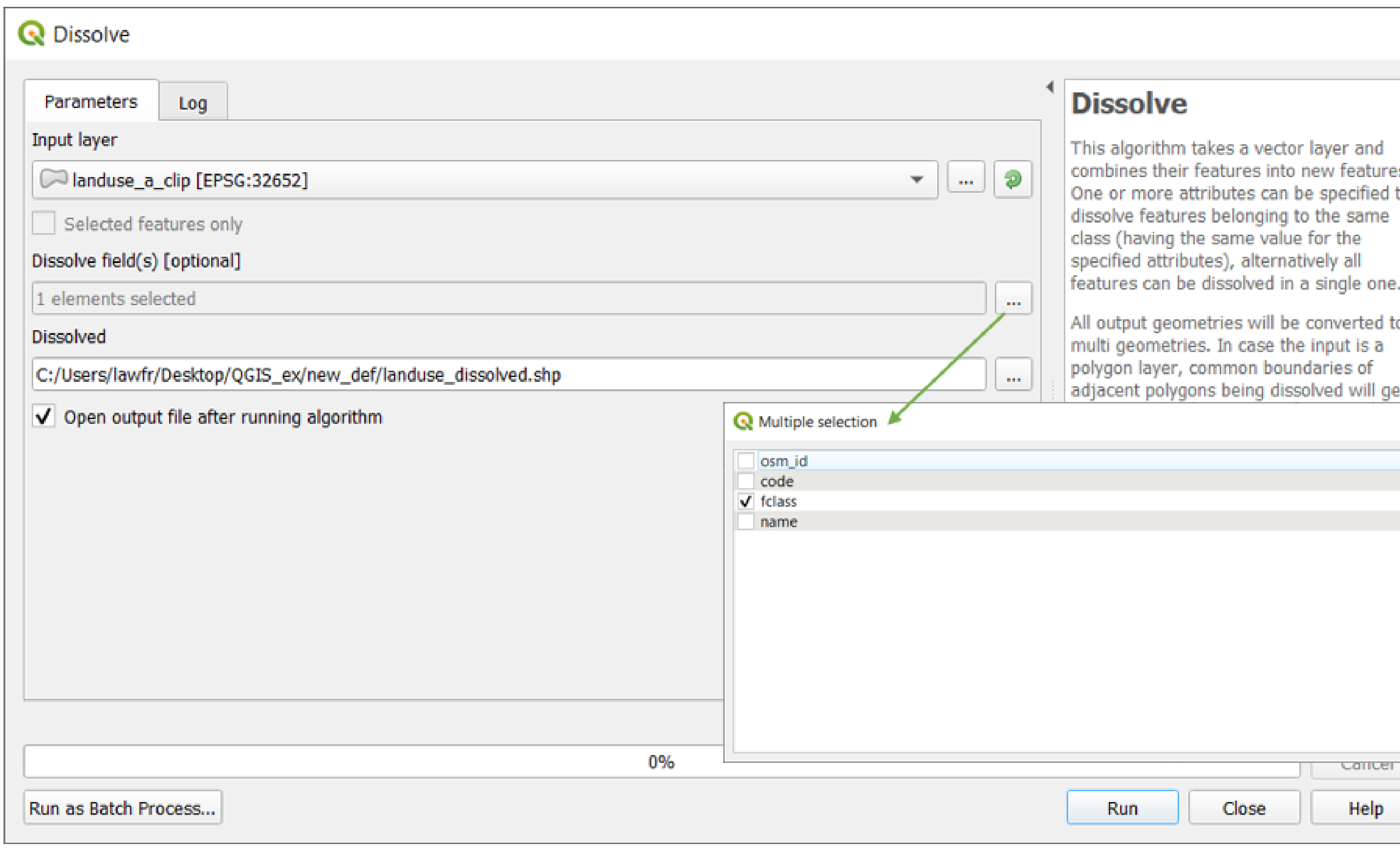
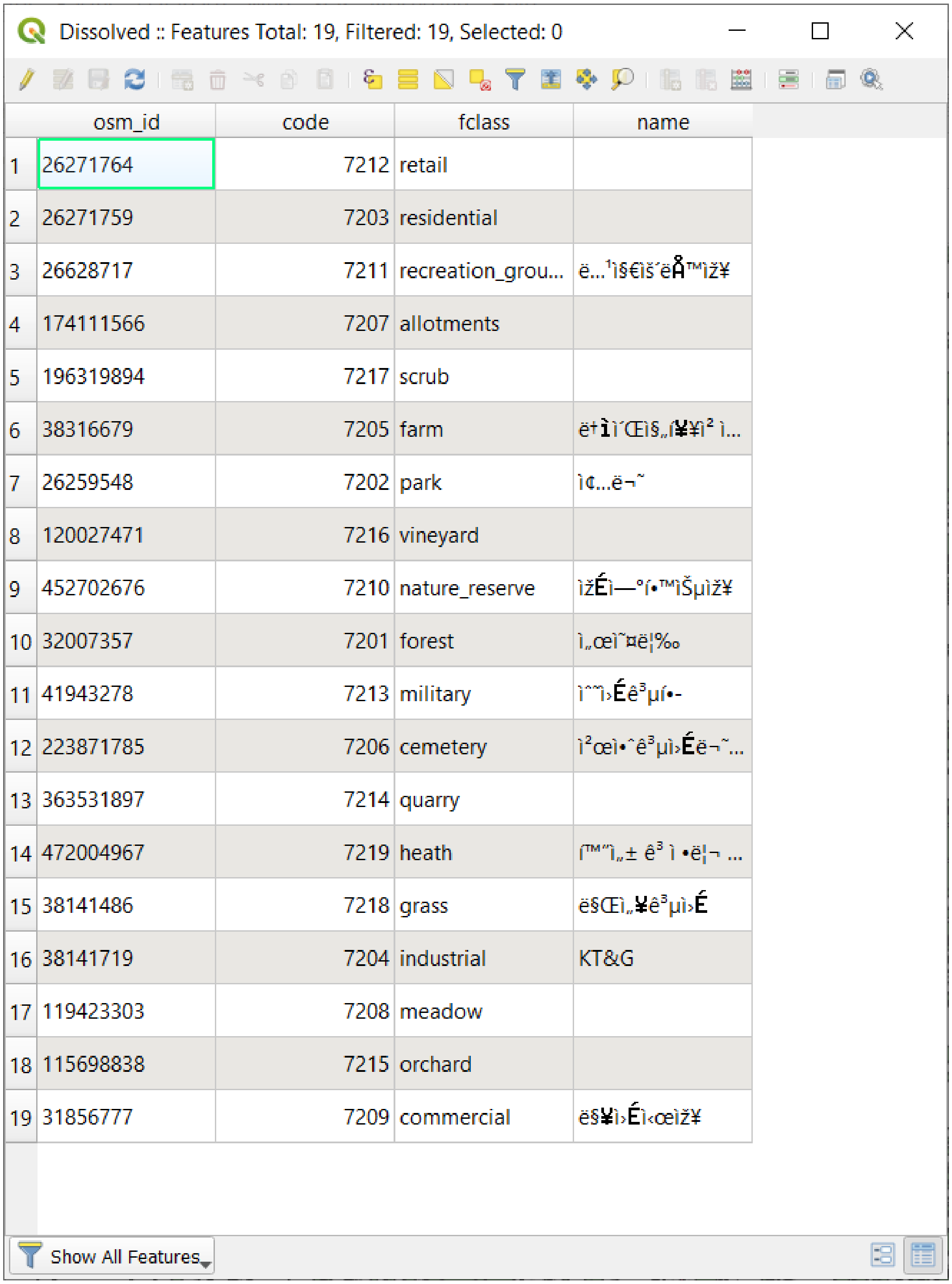


Fig. 3.3.3.1: Dissolve function window



Dissolved :: Features Total: 19, Filtered: 19, Selected: 0

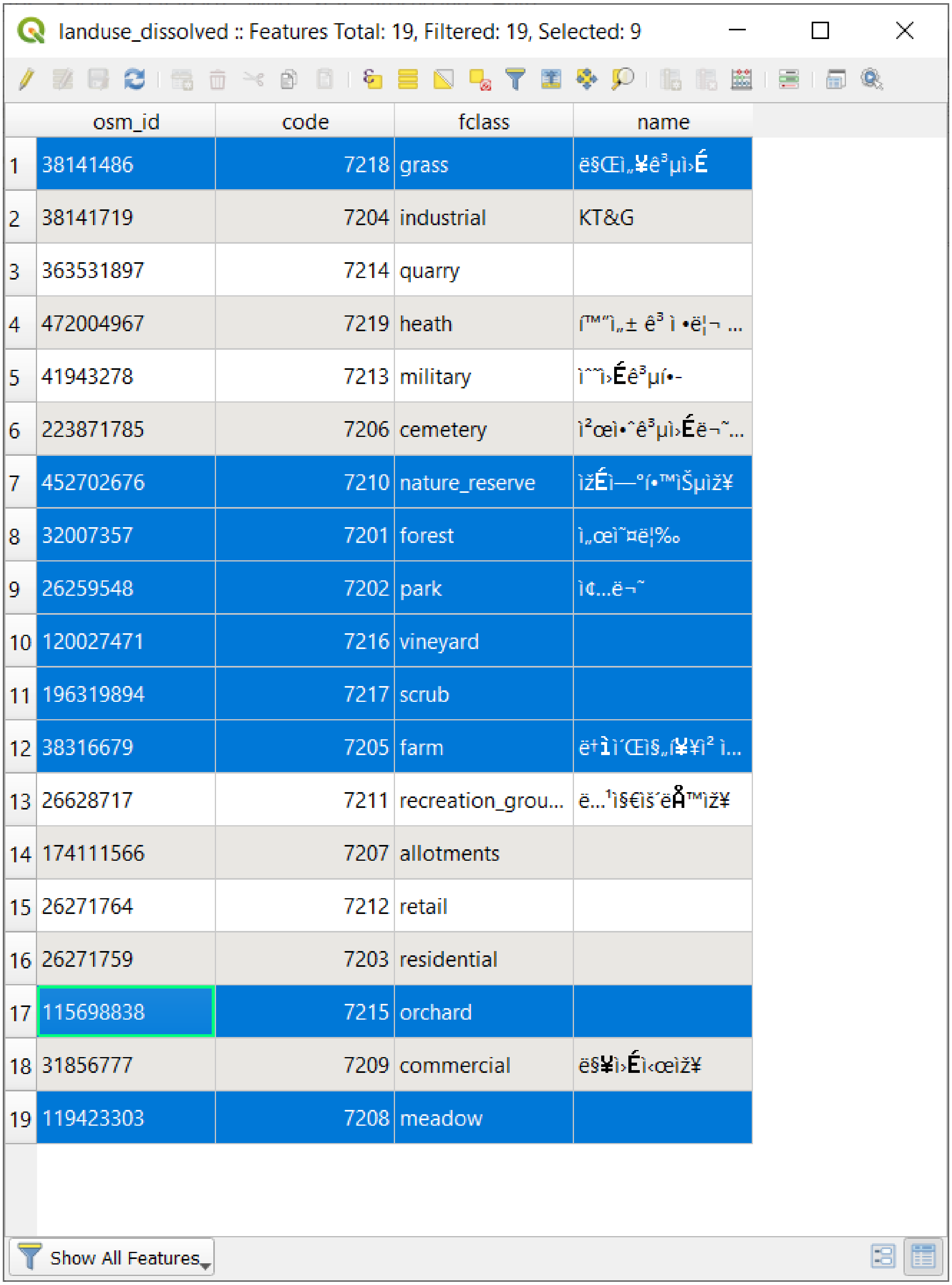
	osm_id	code	fclass	name
1	26271764	7212	retail	
2	26271759	7203	residential	
3	26628717	7211	recreation_grou...	ë…¹ì§€ì´ëÅ™ìž¥
4	174111566	7207	allotments	
5	196319894	7217	scrub	
6	38316679	7205	farm	ë†ìì´Œì§„í¥ì² ì…
7	26259548	7202	park	ì¢…ë¬˜
8	120027471	7216	vineyard	
9	452702676	7210	nature_reserve	ìžÉì—°í•™ìŠµìž¥
10	32007357	7201	forest	ì„œì˜¤ë¦‰
11	41943278	7213	military	ìˆ˜ìÉê³µí•-
12	223871785	7206	cemetery	ì²œì•ˆê³µìÉë¬˜…
13	363531897	7214	quarry	
14	472004967	7219	heath	í™”ì„± ê³ ì •ë¦¬ …
15	38141486	7218	grass	ë§Œì„¥ê³µìÉ
16	38141719	7204	industrial	KT&G
17	119423303	7208	meadow	
18	115698838	7215	orchard	
19	31856777	7209	commercial	ë§¥ìÉì‹œìž¥

Show All Features

3.3.4 Difference

We can now use the newly created `landuse_dissolved` layer to perform a Difference operation. Difference is a function available at *Vector->Geoprocessing Tools->Difference*, that extracts features of an Input layer that fall outside features in the Overlay layer (totally or partially). If a feature of the Input layer partially overlaps a feature of the Overlay layer only the portions outside the Overlay layer features are retained. We will use it to filter out all the natural land use leaving only the industrial and urbanized ones. To do so, first, we have to select the natural land use features:

- Right-click on the `landuse_dissolved` layer on the Layers panel
- Open its Attribute table
- Manually select the features having natural land use classes (like the following ones) by holding `Ctrl` and clicking on the row number on the left
  - heath
  - forest
  - scrub
  - orchard
  - natural_reserve
  - farm
  - meadow
  - vineyard
  - grass


landuse_dissolved :: Features Total: 19, Filtered: 19, Selected: 9

	osm_id	code	fclass	name
1	38141486	7218	grass	ë§Œì„¥ê³µì›É
2	38141719	7204	industrial	KT&G
3	363531897	7214	quarry	
4	472004967	7219	heath	í™”ì„± ê³ ì •ë¦¬ ...
5	41943278	7213	military	ì˜ì›Éê³µí•
6	223871785	7206	cemetery	ì²œì•ˆê³µì›Éë¬˜...
7	452702676	7210	nature_reserve	ìžÉì—°í•™ìŠµìž¥
8	32007357	7201	forest	ì„œì˜¤ë¦‰
9	26259548	7202	park	ì¢…ë¬˜
10	120027471	7216	vineyard	
11	196319894	7217	scrub	
12	38316679	7205	farm	ë†ìì´Œì§„í¥ì² ì...
13	26628717	7211	recreation_grou...	ë…¹ì§€ì´ëÅ™ìž¥
14	174111566	7207	allotments	
15	26271764	7212	retail	
16	26271759	7203	residential	
17	115698838	7215	orchard	
18	31856777	7209	commercial	ë§¥ì›Éì‹œœìž¥
19	119423303	7208	meadow	

Show All Features

Now that we have selected those features, we can go on with the Difference operation. The input parameters are:

- *Input layer*: the `landuse_a_clip` layer
- *Overlay layer*: the `landuse_dissolved` layer, but be sure to check the "Selected features only" checkbox
- *Difference*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

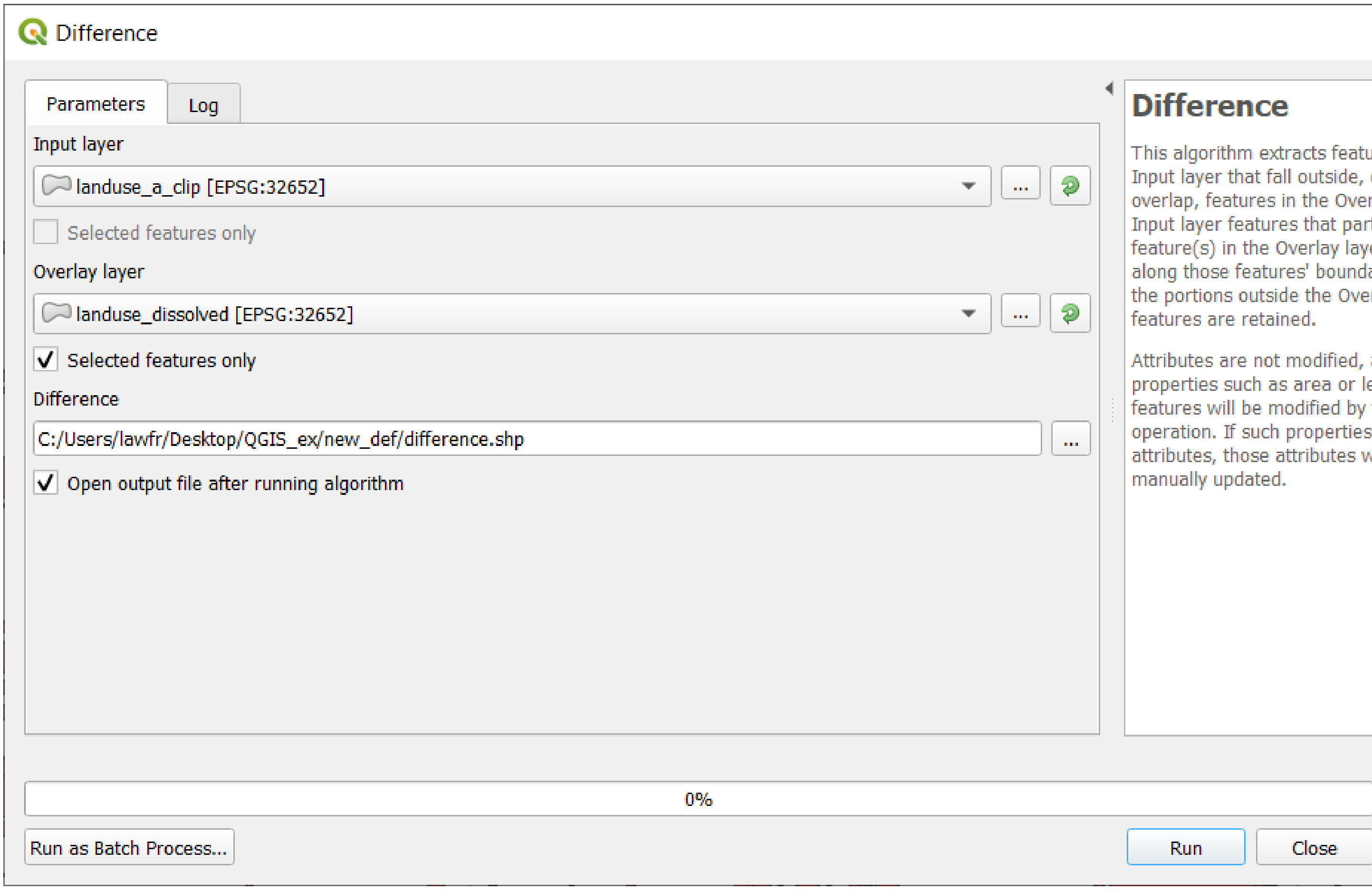


Fig. 3.3.4.1: Difference function window

The result will be a land use layer that only retains the features with industrial and urbanized land use types.

3.3.5 Symmetrical difference

The Symmetrical difference, available at *Vector->Geoprocessing Tools->Symmetrical Difference*, provides a function that extracts the portions of features from both the Input and Overlay layers that do not overlap. The attribute table of the Symmetrical Difference result layer contains original attributes from both the Input and Overlay layers.

If you look at the `landuse_a_clip` layer, you will see it has some portions that overlap with water features:

We will use the Symmetrical difference to obtain a land use layer with no feature that has parts in water bodies. To do so, the input parameters are:

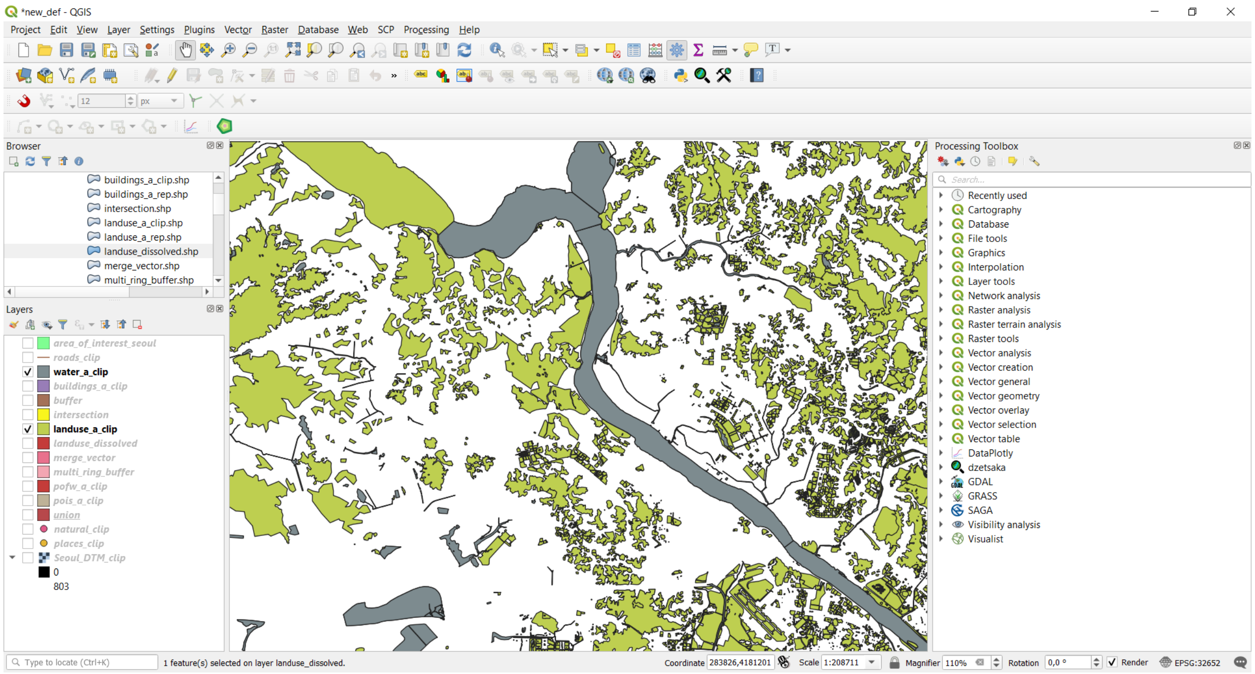


Fig. 3.3.5.1: Water bodies (gray) overlapping landuse features (green)

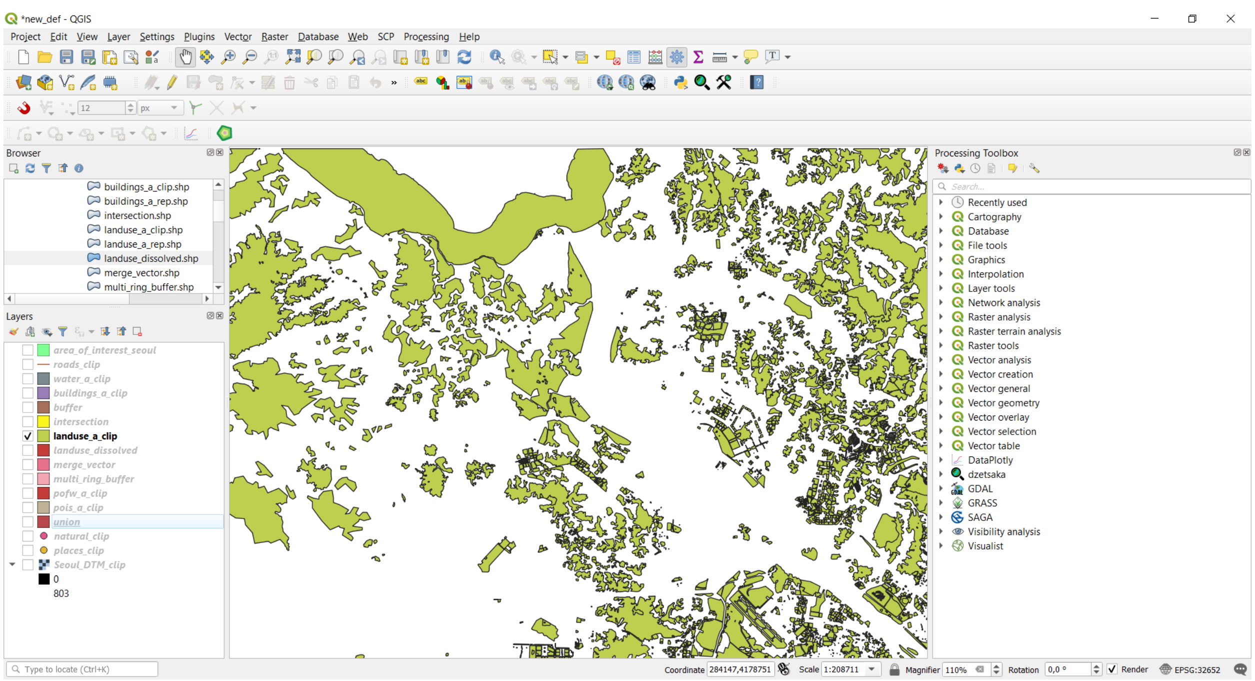


Fig. 3.3.5.2: Only landuse features (green)

- *Input layer*: `merge_vector` layer
- *Overlay layer*: `water_a_clip` layer
- *Symmetrical Difference*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

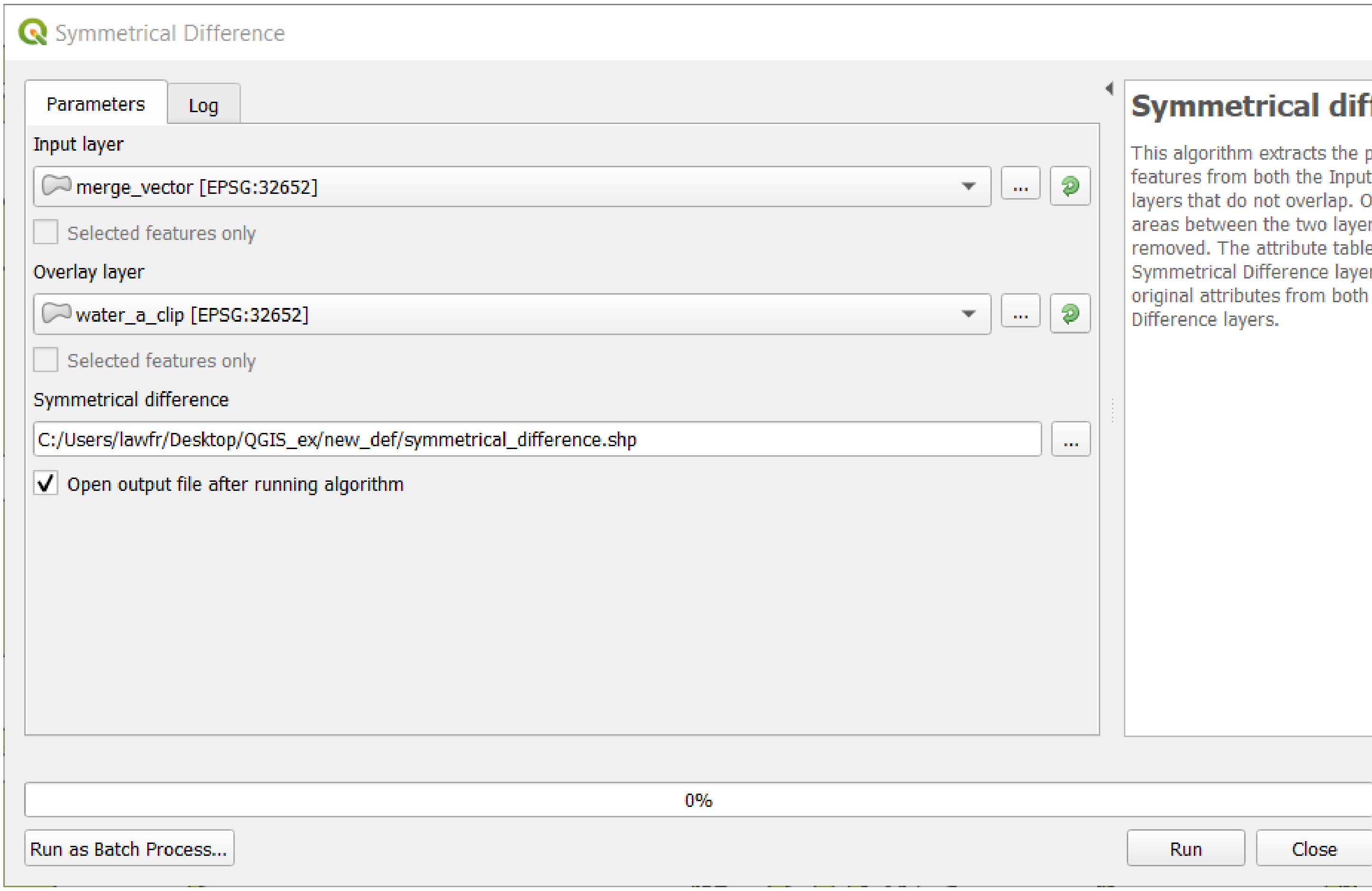
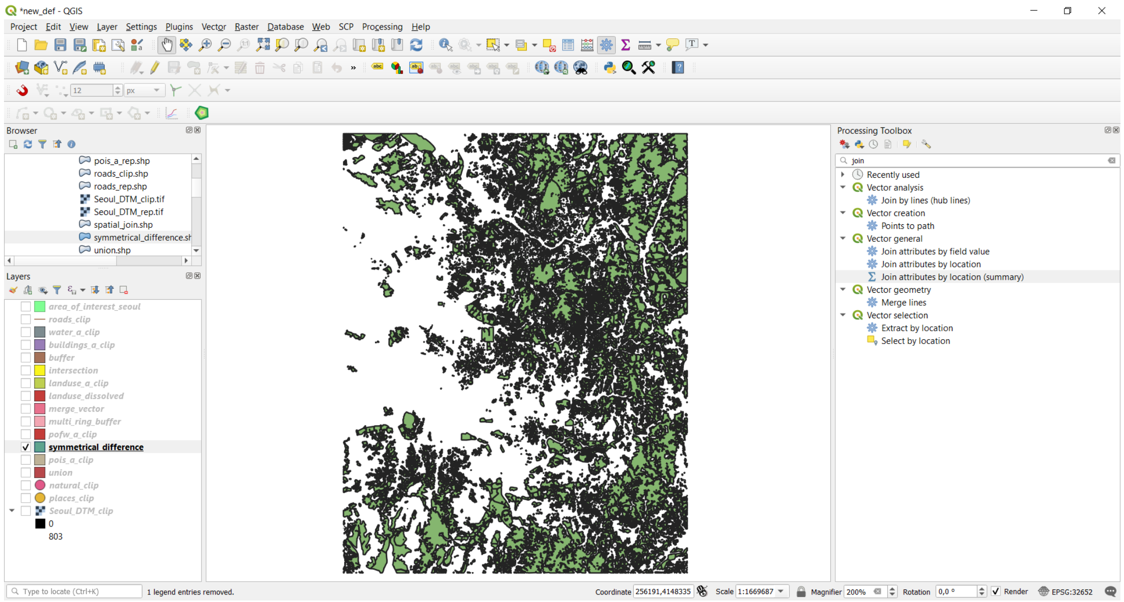


Fig. 3.3.5.3: Symmetrical difference function window

Once the execution is done, the results should look like this:



3.3.6 Spatial join

As you saw with the *Intersection* function we can find the buildings that are also religious places. We will now use the Join Attributes by Location function (generally known as Spatial Join) available at *Vector->Data Management Tools->Join Attributes by Location*, to extend the previous analysis. The Join attributes by location is a function that takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table. This additional attributes and their values are taken from a second vector layer applying spatial criteria to select the values from the second layer that are added to each feature from the first layer in the resulting one. In this way we could add the attribute that specifies the religion practised in a particular building; the input parameters are:

- *Input layer*: the `buildings_a_clip` layer
- *Join layer*: the `pofw_a_clip` layer
- *Geometric predicate*: within
- *Fields to add*: click on the icon on the left and select the "fclass" attribute
- *Join type*: take attributes of the first located feature only (one-to-one)
- Tick the "Discard record which cannot be joined" option
- *Joined layer*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

If you now look at the attribute table of the newly created layer, you will see a new attribute (fclass_2) that represents the religion practiced in that building.

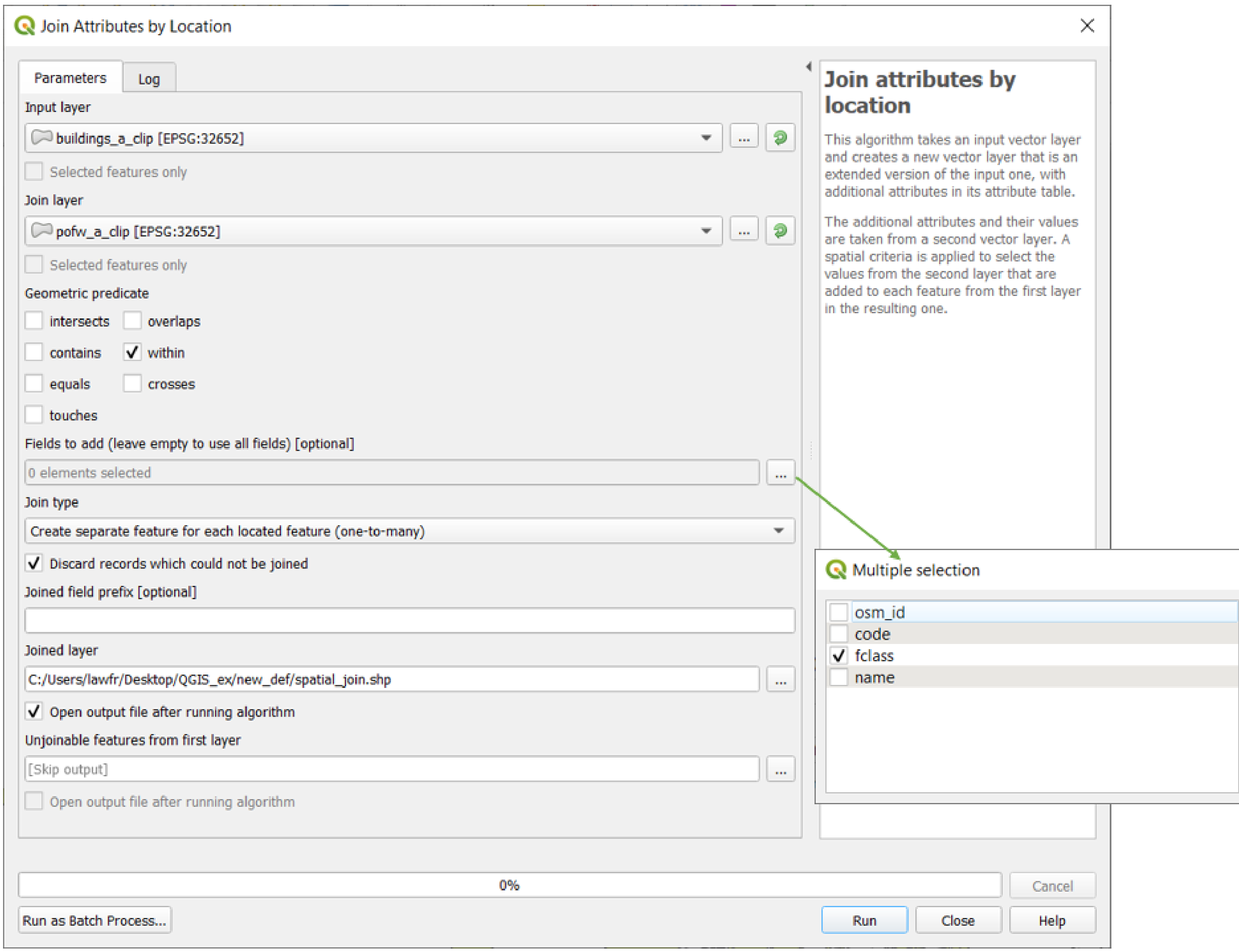
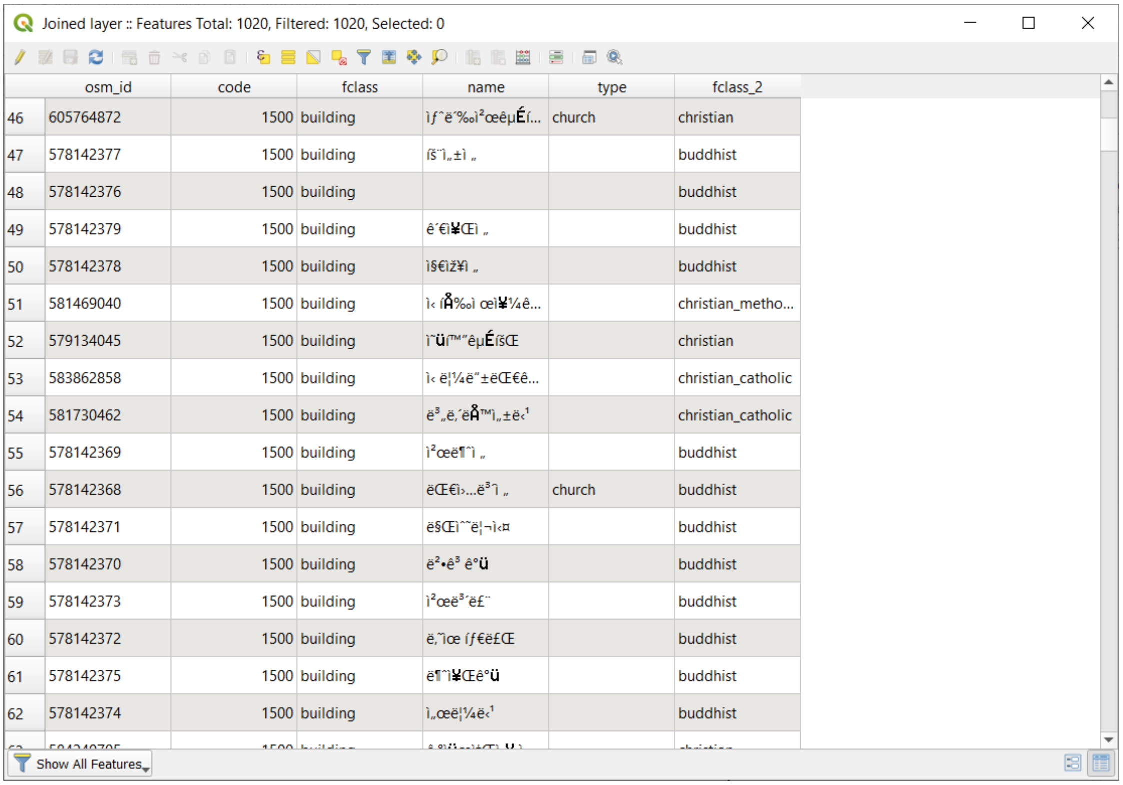


Fig. 3.3.6.1: Join Attributes by Location function window



Note: The Join Attributes by Location function, specifying the "intersects" predicate, also translates the "Identity" function of ArcGIS

3.4 Proximity analysis

We will now focus on operations involving point layers. Please note that QGIS, when clipping a Point layer, also converts its geometry type to MultiPoint. This geometry type is not suitable for some of the functions we will use in the next step and therefore we will first see how to convert the geometry type back to Point.

3.4.1 Convert geometry type

Available at *Processing Toolbox->Vector geometry->Convert geometry type*, it provides an algorithm that allows to convert the MultiPoint features to single Point features. To do so, the input parameters are:

- *Input layer*: the point layer whose geometry we want to convert. In this example we use the `places_clip` layer
- *New geometry type*: select Centroids
- *Converted*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

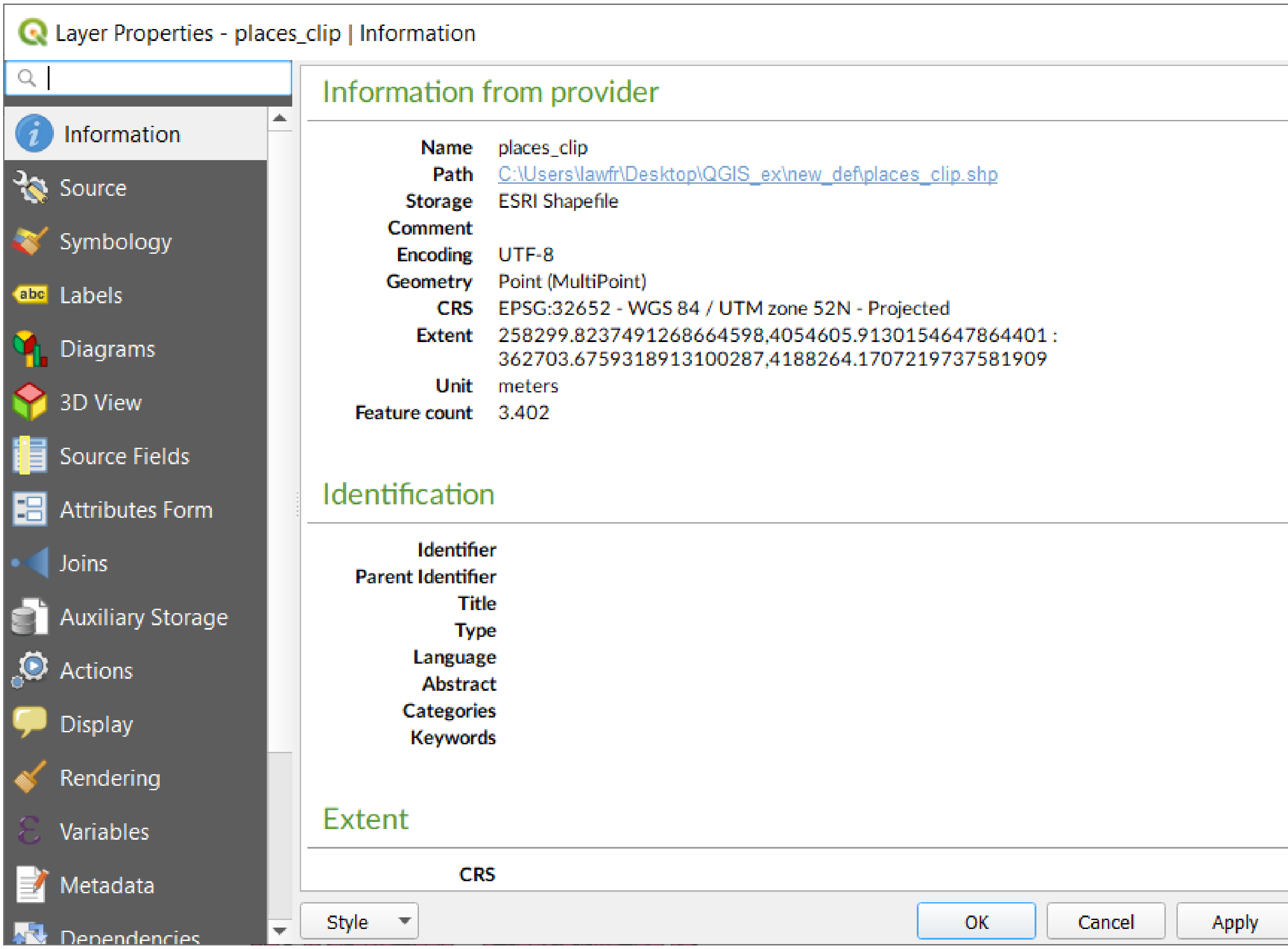


Fig. 3.4.1: In the information of the `places_clip` layer we can see that its geometry is MultiPoint

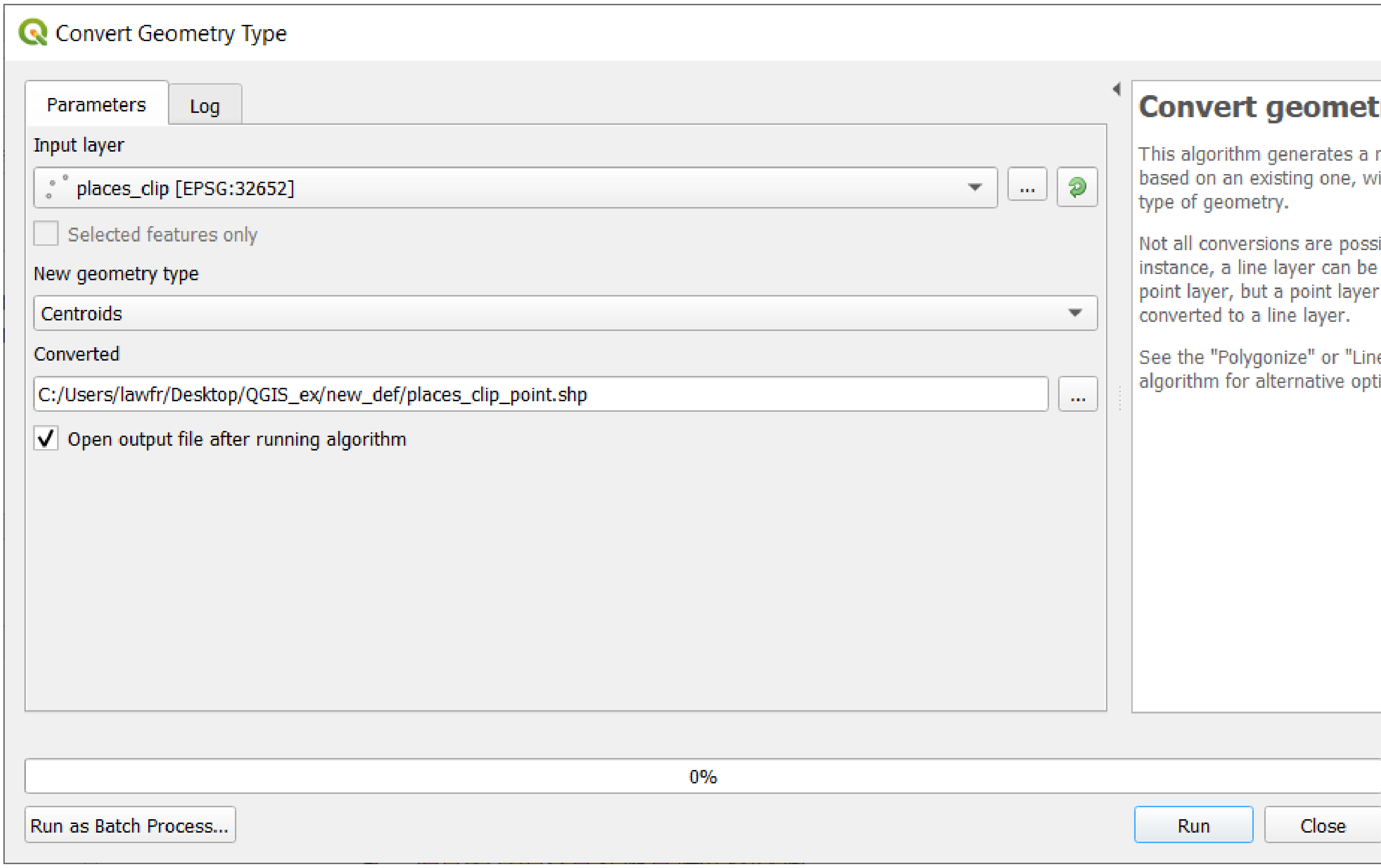


Fig. 3.4.1.1: Convert geometry function window

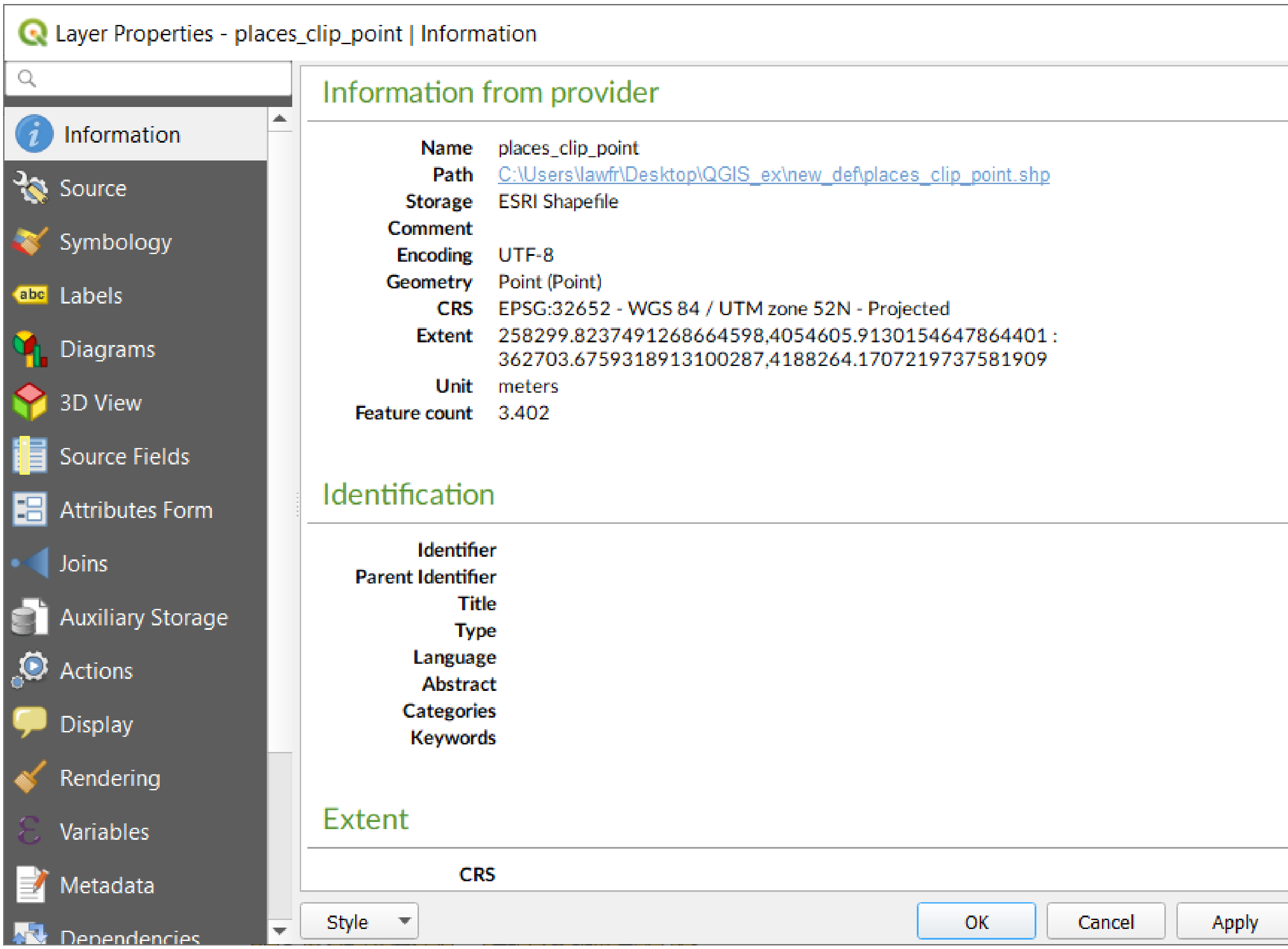


Fig. 3.4.1.2: Now the `places_clip` layer's geometry is MultiPoint

In order to continue with the following functions, please convert the geometry type also for the `natural_clip` point layer.

Note: After you are done with the conversion, you can remove the previous point layers and include in the project only the new ones.

3.4.2 Average Nearest Neighbor

Available at *Processing Toolbox->Vector Analysis->Nearest Neighbour Analysis*, it provides a function that performs nearest neighbor analysis for a point layer. The output is generated as an HTML file with the computed statistics. We perform the Nearest Neighbor Analysis with the `natural_clip_point` point layer; the input parameters are:

- *Input layer*: the `natural_clip_point` layer
- *Nearest Neighbor*: the path and the name of the output HTML file. Note that if left empty a temporary file will be created

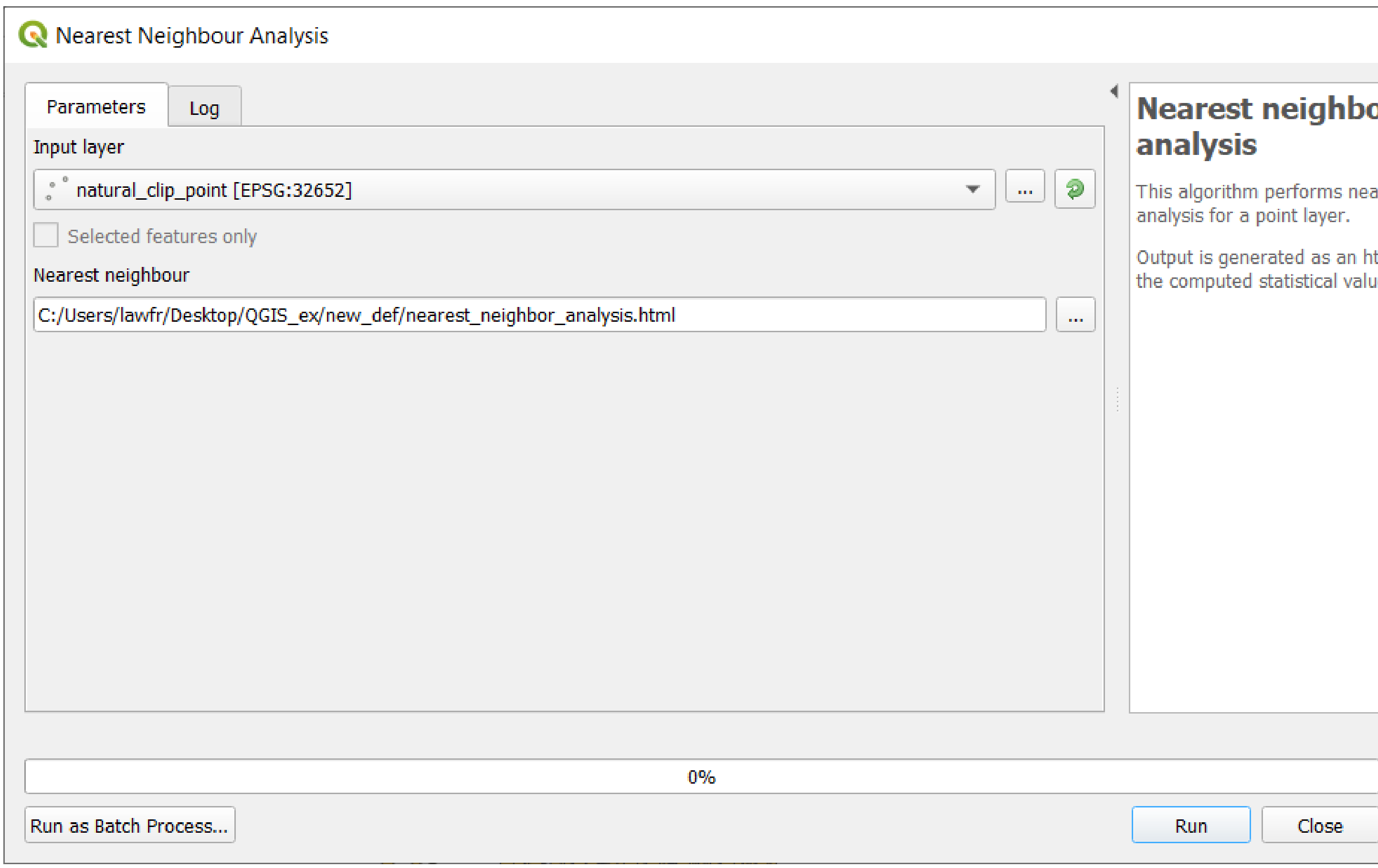


Fig. 3.4.2.1: Nearest Neighbor Analysis function window

Once the operation is done, you can open the HTML file containing the results and you will see information about the Observed mean distance, Expected mean distance, Nearest neighbour index, Number of points, and Z-Score.

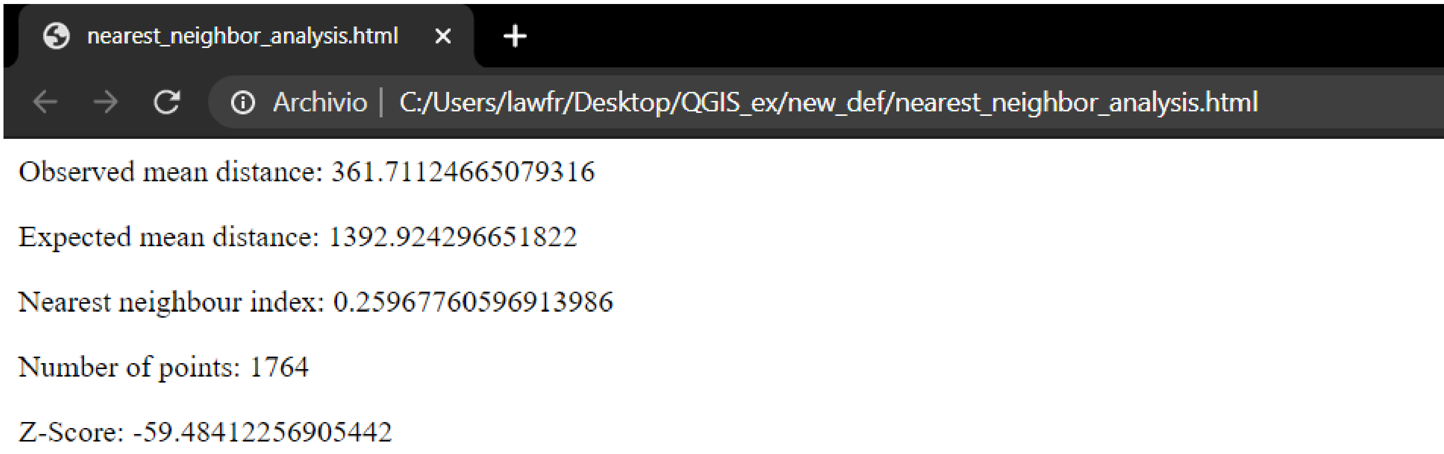


Fig. 3.4.2.2: The HTML result when opened in the browser

We now see how to calculate the nearest feature to a given point or set of points in QGIS. We distinguish between distances from point to point, and from point to a line or polygon layer.

3.4.3 Distance from point to point

Available at *Processing toolbox->Vector analysis->Distance to nearest hub (points)*, it provides an algorithm that computes the distance between point features taken as the origin and their closest point destination. In this case, we will calculate the distance from the `places_clip_point` layer to the `natural_clip_point` layer. The input parameters are:

- *Source points layer*: the `places_clip_point` shapefile
- *Destination hubs layer*: the `natural_clip_point` shapefile
- *Hub layer name attribute*: osm_id
- *Measurement units*: meters
- *Hub distance*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

The result is a copy of the places layer, but each point feature has two additional attributes: the id of the nearest natural point feature (HubName) and the distance from it (HubDist), as you can see from its attribute table:

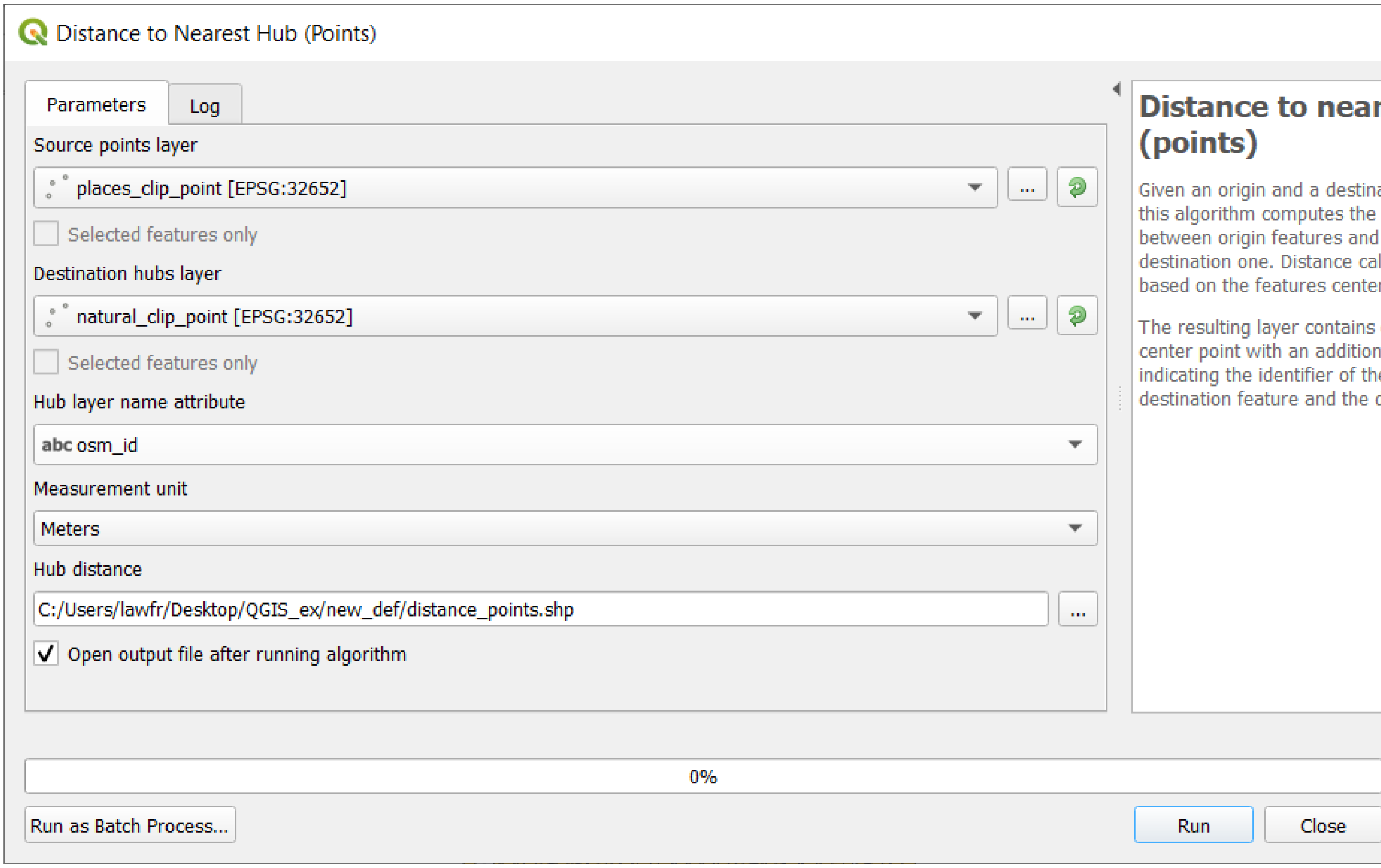
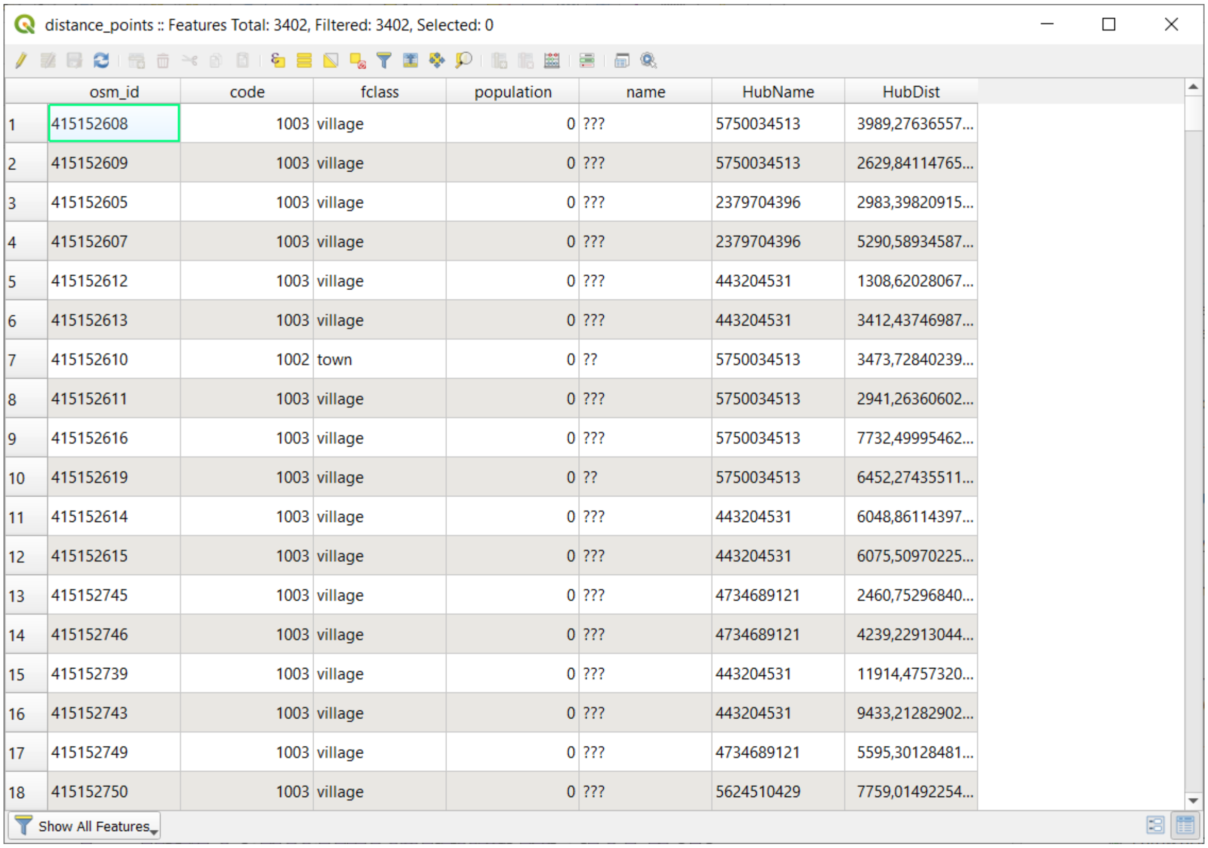


Fig. 3.4.3.1: Distance from point to point function window


distance_points :: Features Total: 3402, Filtered: 3402, Selected: 0

	osm_id	code	fclass	population	name	HubName	HubDist
1	415152608	1003	village	0	???	5750034513	3989,27636557...
2	415152609	1003	village	0	???	5750034513	2629,84114765...
3	415152605	1003	village	0	???	2379704396	2983,39820915...
4	415152607	1003	village	0	???	2379704396	5290,58934587...
5	415152612	1003	village	0	???	443204531	1308,62028067...
6	415152613	1003	village	0	???	443204531	3412,43746987...
7	415152610	1002	town	0	??	5750034513	3473,72840239...
8	415152611	1003	village	0	???	5750034513	2941,26360602...
9	415152616	1003	village	0	???	5750034513	7732,49995462...
10	415152619	1003	village	0	??	5750034513	6452,27435511...
11	415152614	1003	village	0	???	443204531	6048,86114397...
12	415152615	1003	village	0	???	443204531	6075,50970225...
13	415152745	1003	village	0	???	4734689121	2460,75296840...
14	415152746	1003	village	0	???	4734689121	4239,22913044...
15	415152739	1003	village	0	???	443204531	11914,4757320...
16	415152743	1003	village	0	???	443204531	9433,21282902...
17	415152749	1003	village	0	???	4734689121	5595,30128481...
18	415152750	1003	village	0	???	5624510429	7759,01492254...

Show All Features

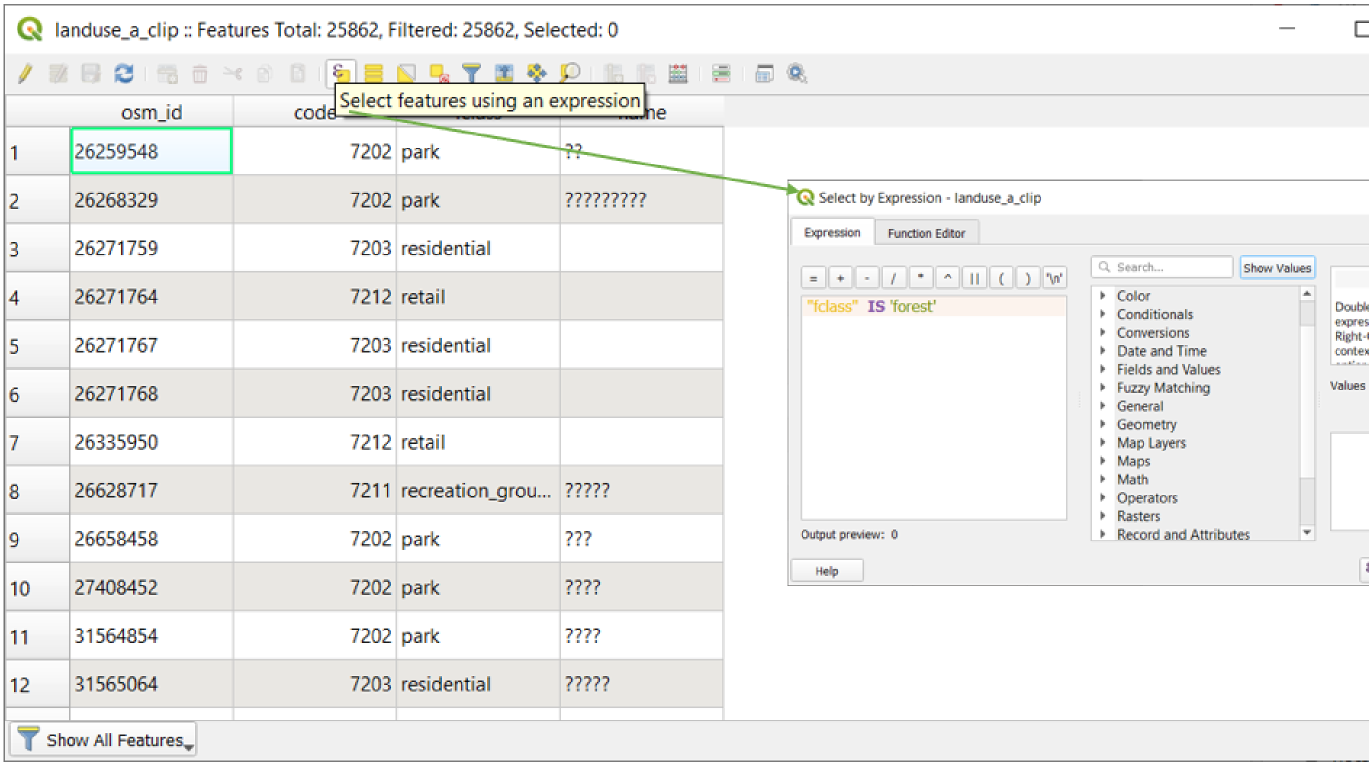
3.4.4 Distance from point to layer

Available at *Processing Toolbox->Vector analysis->Distance to nearest hub (line to hub)*, it provides an algorithm that computes the distance between point features taken as origin and their closest destination line or polygon feature.

Note: Distance calculations are based on the centroid of the line or polygon features.

In this case, we calculate the closest forest to each place. To do so, we select all the forests from the `landuse_a_clip` layer:

- Right-click on the `landuse_a_clip` layer in the Layers panel and click on Open attribute table
- Click on the Select features using an expression button
- In the window, write the following expression: `"fclass" is 'forest'`, and then click "Select features"



Once we have selected all the forests from the `landuse_clip` layer, we can run the Distance to nearest hub (line to hub) function. The input parameters are:

- *Source points layer*: the `places_clip_point` layer
- *Destination hubs layer*: the `landuse_a_clip` layer, considering only selected features
- *Hub layer name attribute*: osm_id
- *Measurement units*: meters
- *Hub distance*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

The result is a line layer representing all the distances from each places point feature to the nearest forest feature.

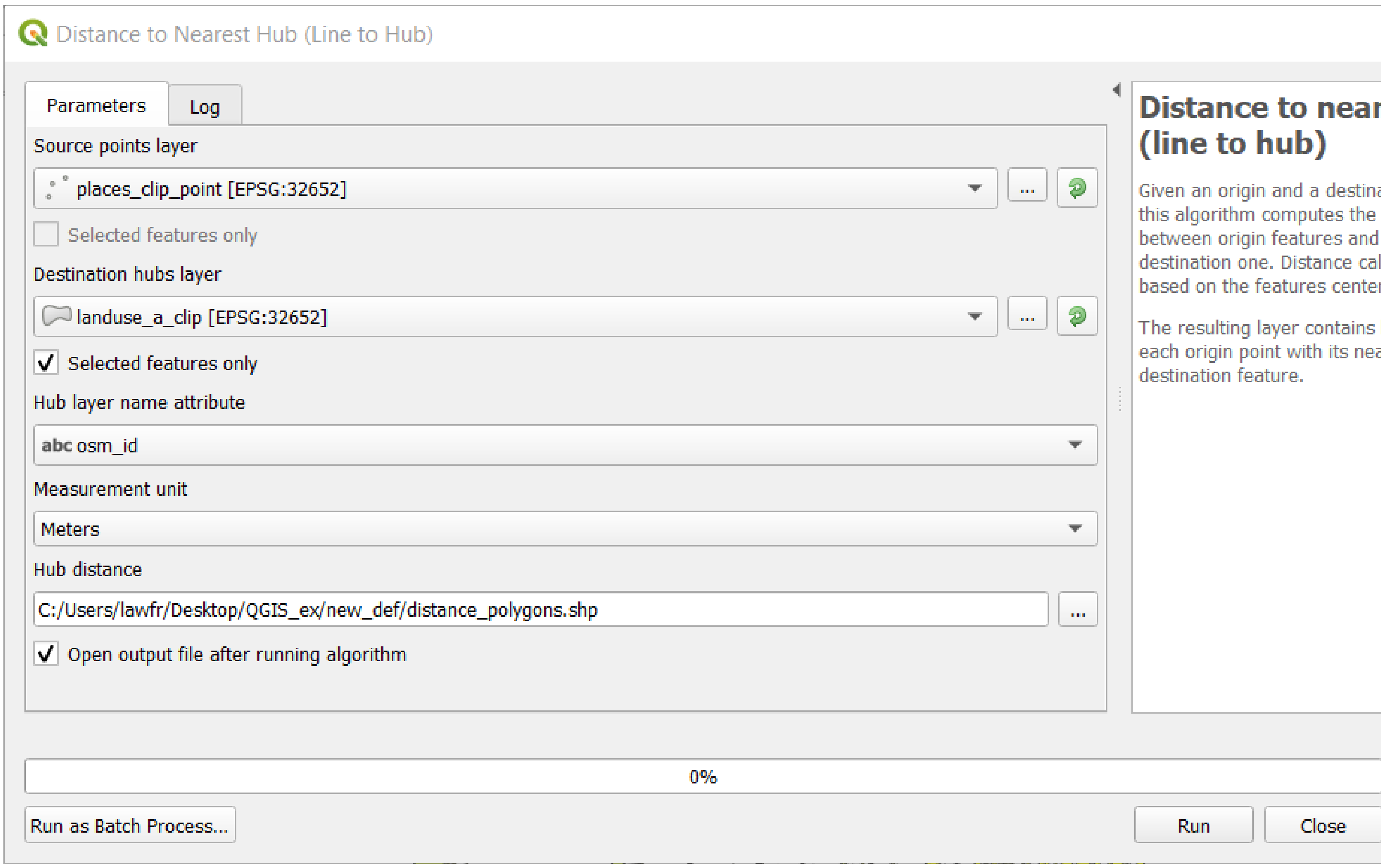
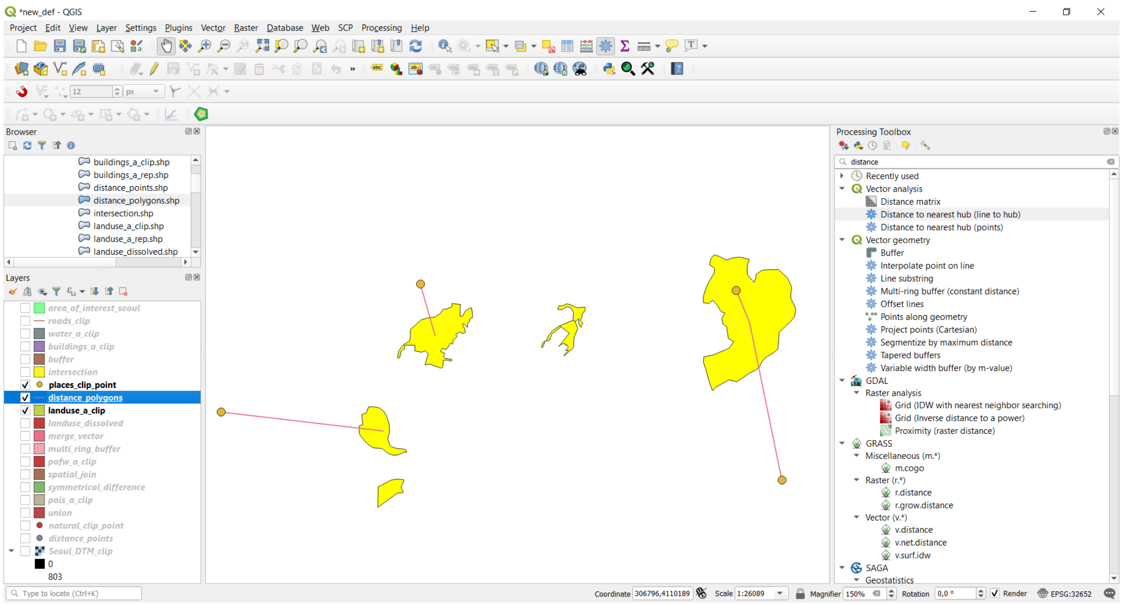


Fig. 3.4.4.1: Distance from point to layer function window



3.5 Network analysis

3.5.1 Shortest path

In order to find the shortest or fastest path in QGIS, we can use the Shortest path function. Having a layer representing a network, we can calculate the shortest path between two chosen points on the map (point to point), from a point layer to a chosen end point (layer to point) or from a chosen start point to a point layer (point to layer).

We illustrate here only the point to point option for the sake of computation, but the others are easily deduced from the following example. The function is available at *Processing Toolbox->Network analysis->Shortest path (point to point)*, and the input parameters are:

- *Vector network layer*: the `roads_clip` layer
- *Path type to calculate*: "shortest" (you can also calculate the fastest path given a network layer with maximum velocity information)
- *Start point*: click on the icon on the right, then choose a starting point from the map
- *End point*: click on the icon on the right, then choose an ending point from the map
- *Shortest path*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

As you can see this algorithm creates a new line layer that represents the shortest path in the provided network to go from the starting point to the end point:

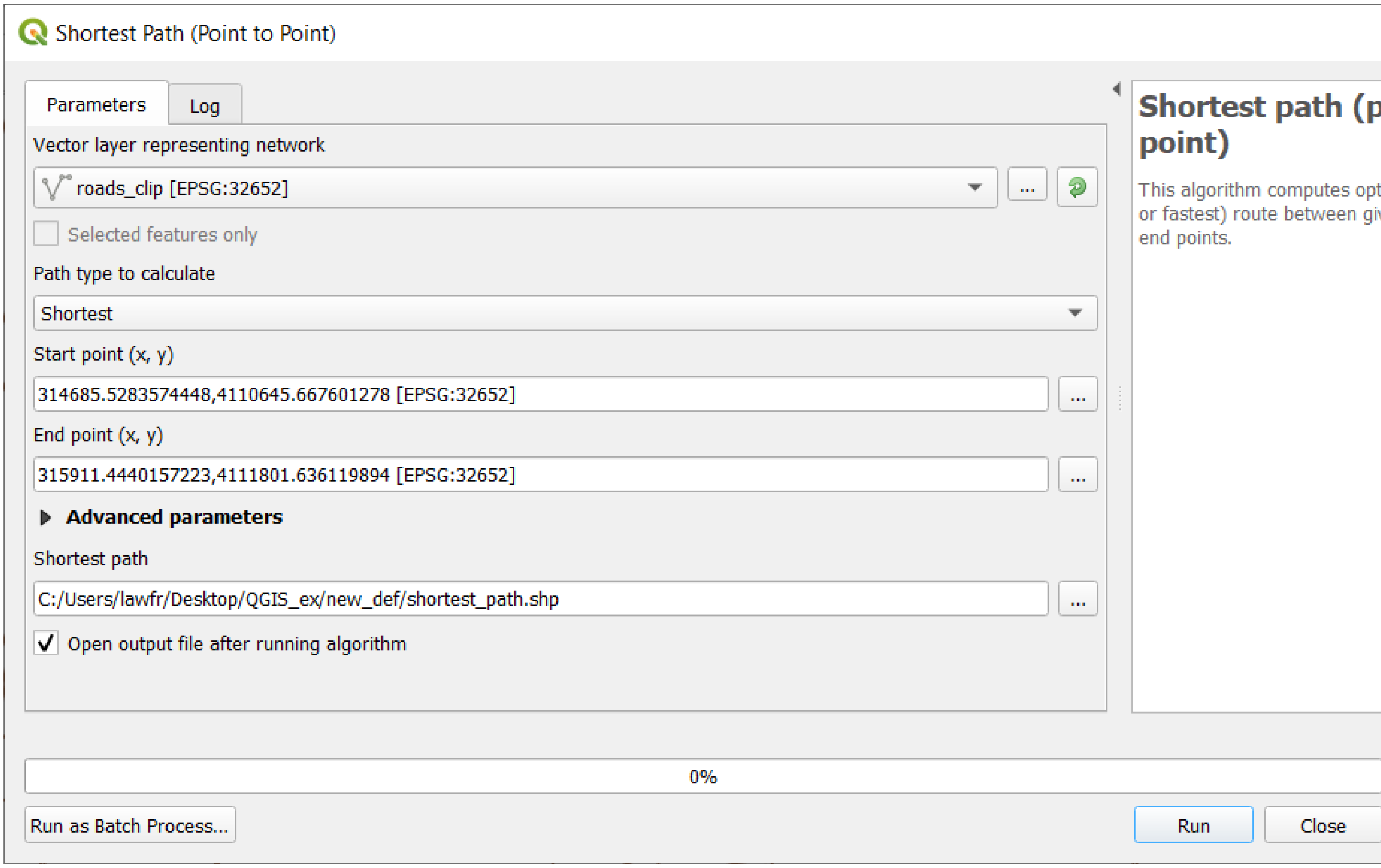
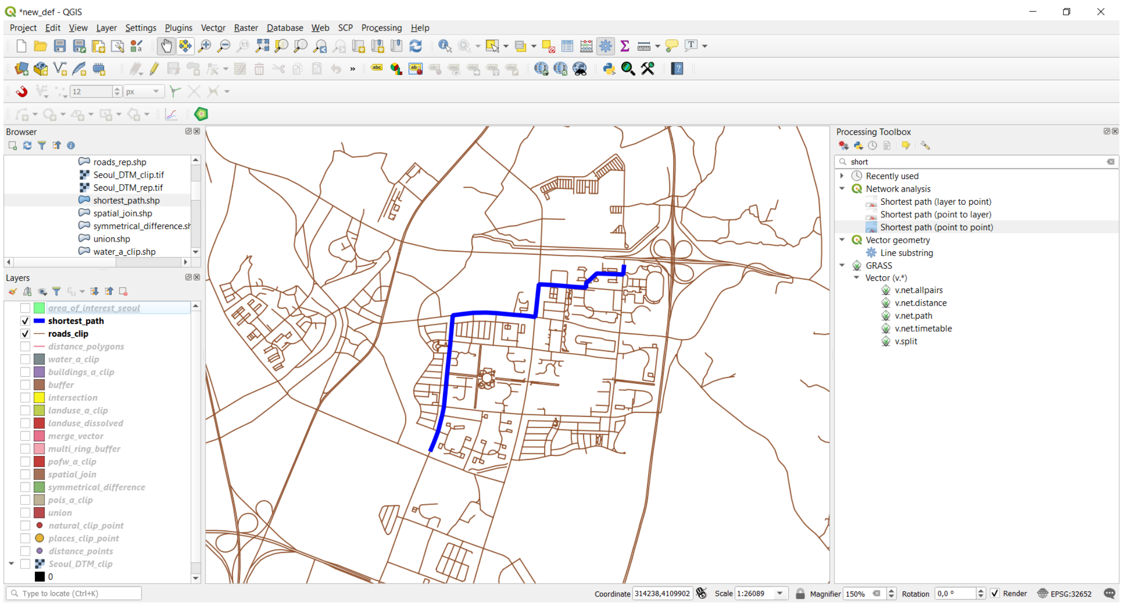


Fig. 3.5.1.1: Shortest path (point to point) function window



3.5.2 Generate service area

In QGIS, we can also generate a service area using the Service Area function. To perform such, you can search for Service area in the Processing toolbox searchbar and select *Service area (from point)*. This function allows creating a vector with all the parts of a network layer that can be reached within a distance or a time, starting from a point chosen on the map. The same can be done starting from a point layer using *Service area (from layer)*. We will use the last one to calculate the service area for all the places points with a maximum travel distance of 100 meters. The input parameters are:

- *Vector network layer*: the `roads_clip` network
- *Vector layer with start points*: the `places_clip_point` layer
- *Path type to calculate*: "shortest" (you can also calculate the fastest path given a network layer with maximum velocity information)
- *Travel cost*: 100 (so that the maximum travel distance is 100m)
- *Service area*: the path and the name of the output vector layer. Note that if left empty a temporary layer will be created

The result will highlight the service area for each point of the places layer:

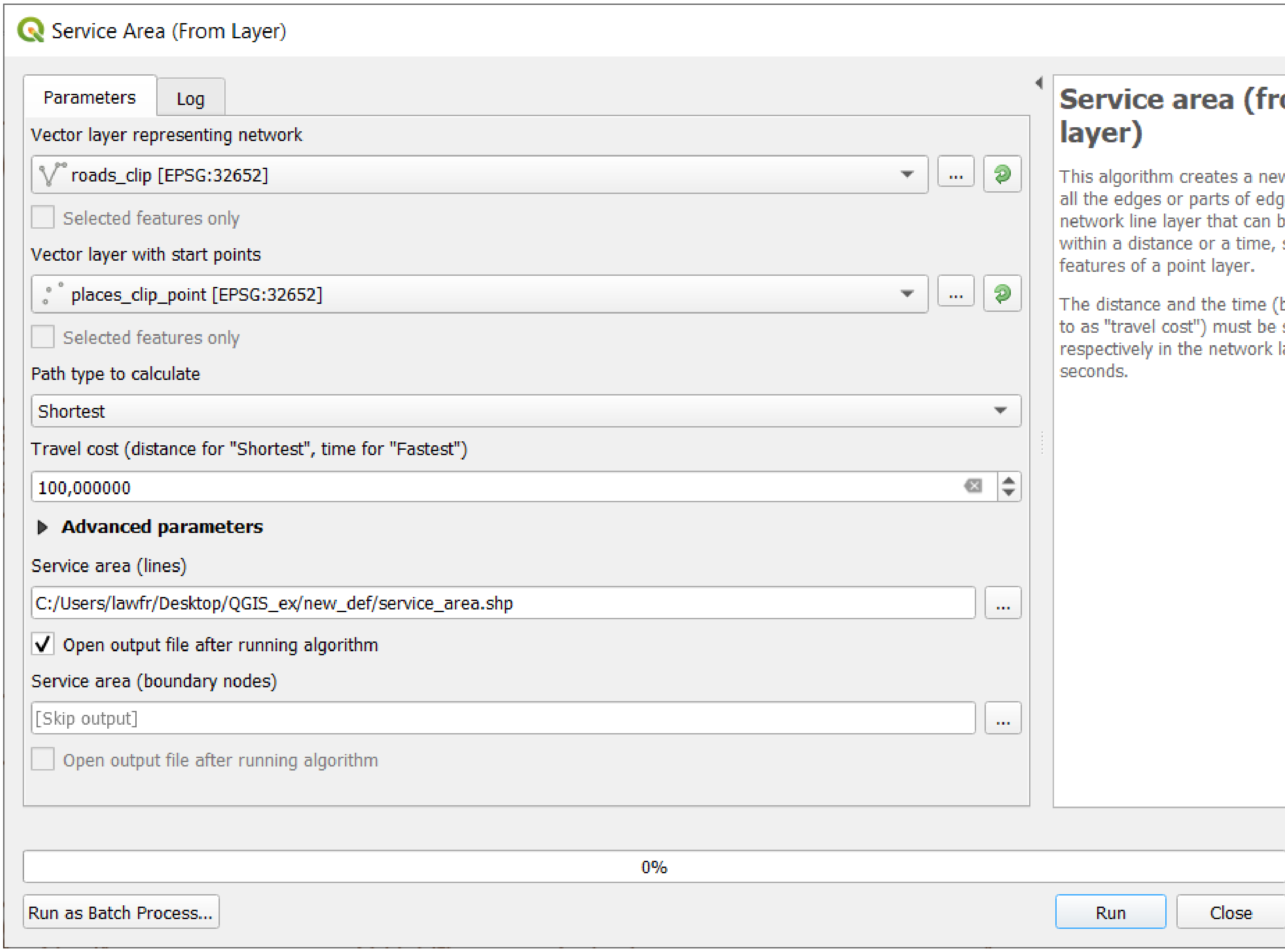
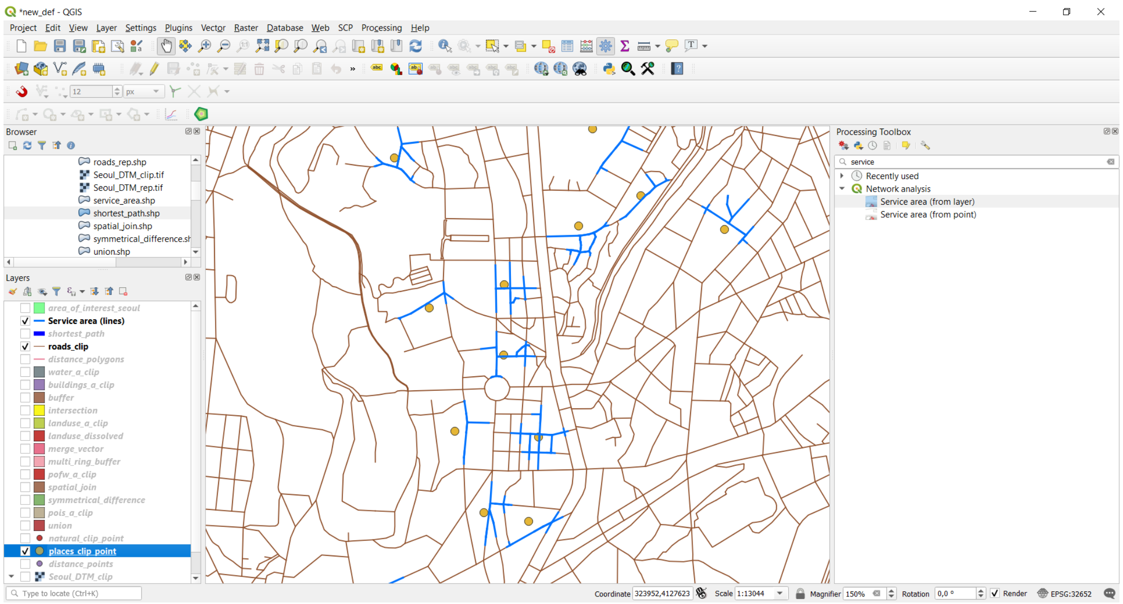


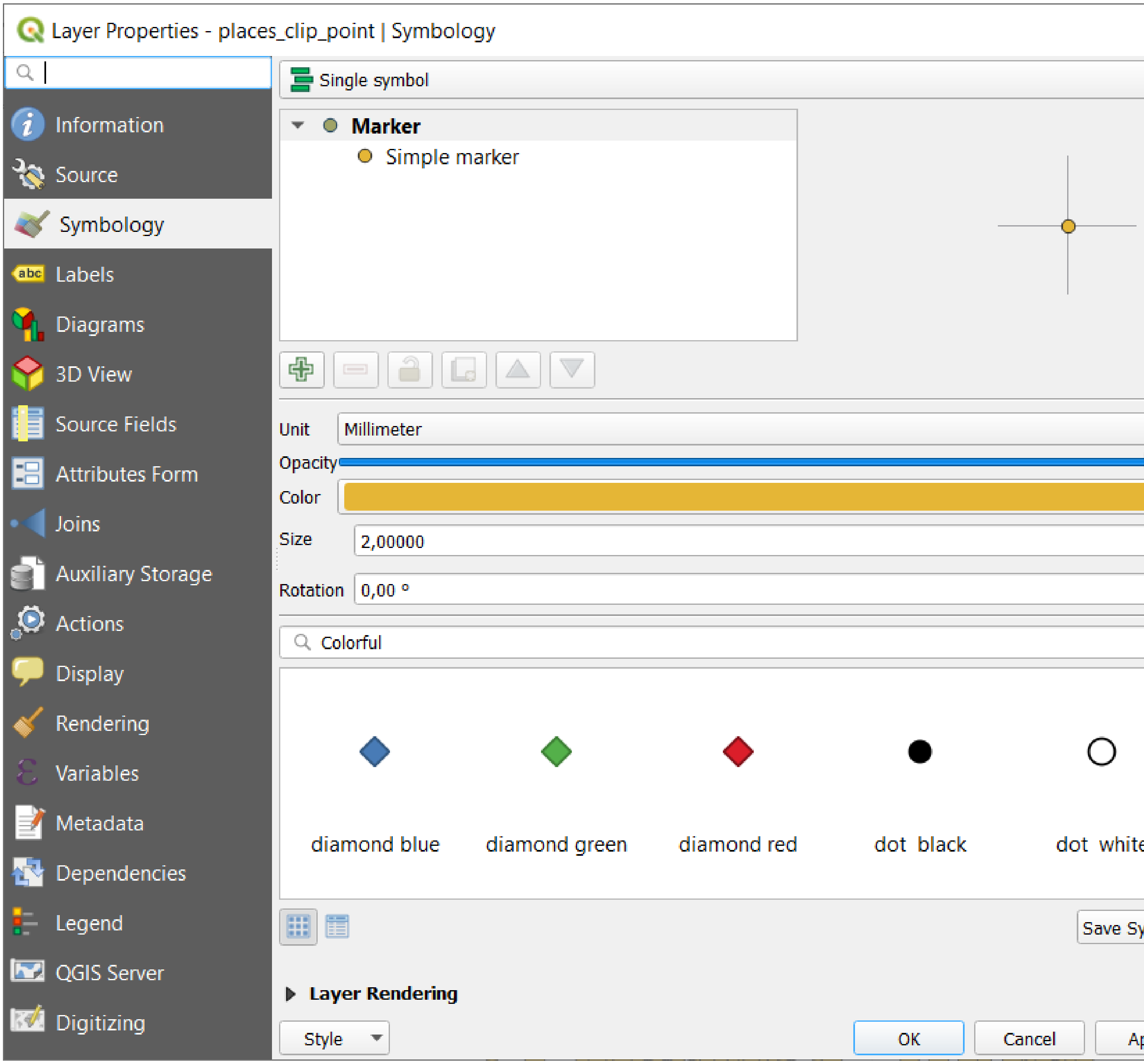
Fig. 3.5.2.1: Generate service area function window



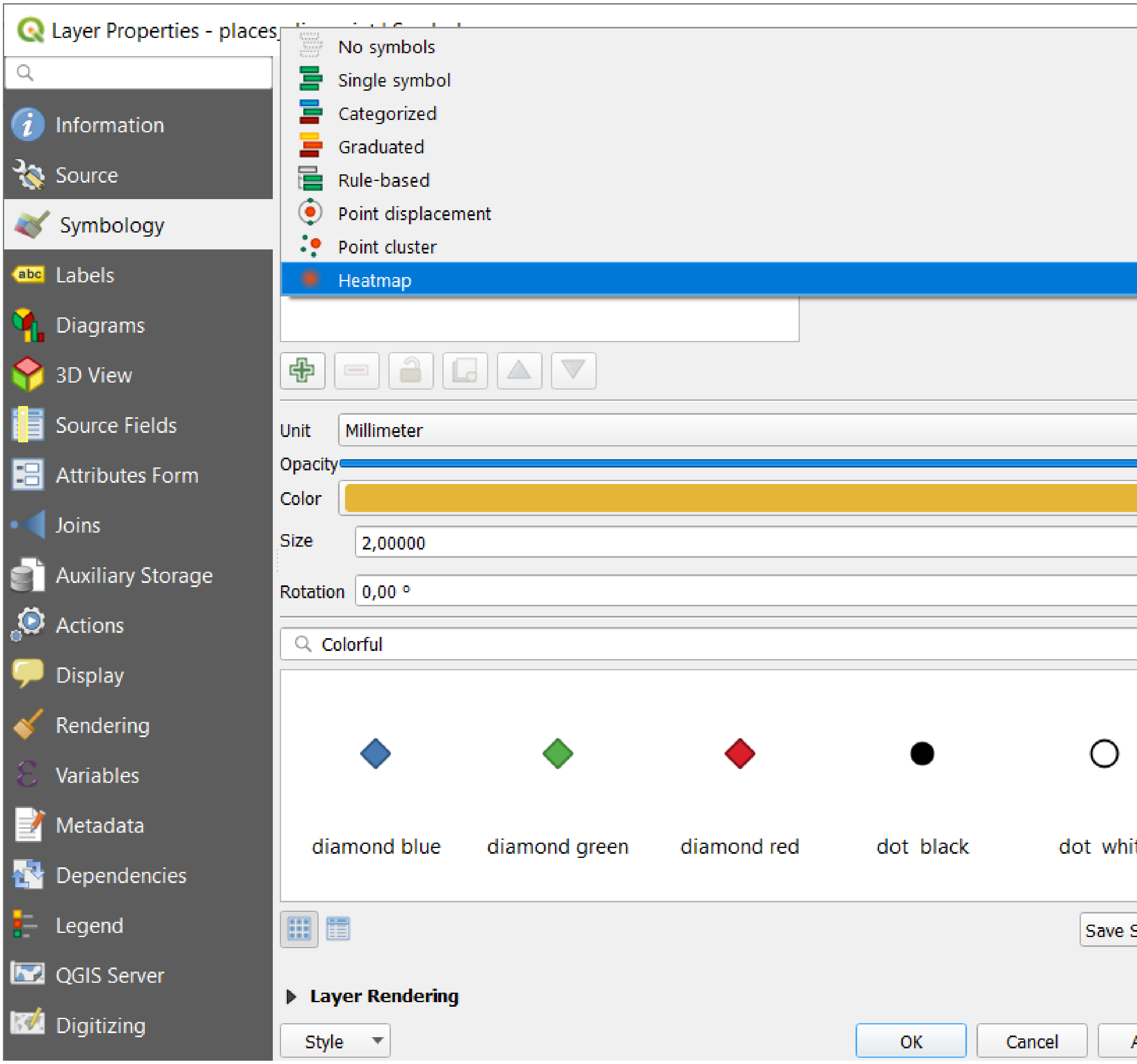
3.6 Kernel density

A useful tool to visualize the density of a point layer it is the Heatmap. In QGIS, we can directly use a styling option of the layer.

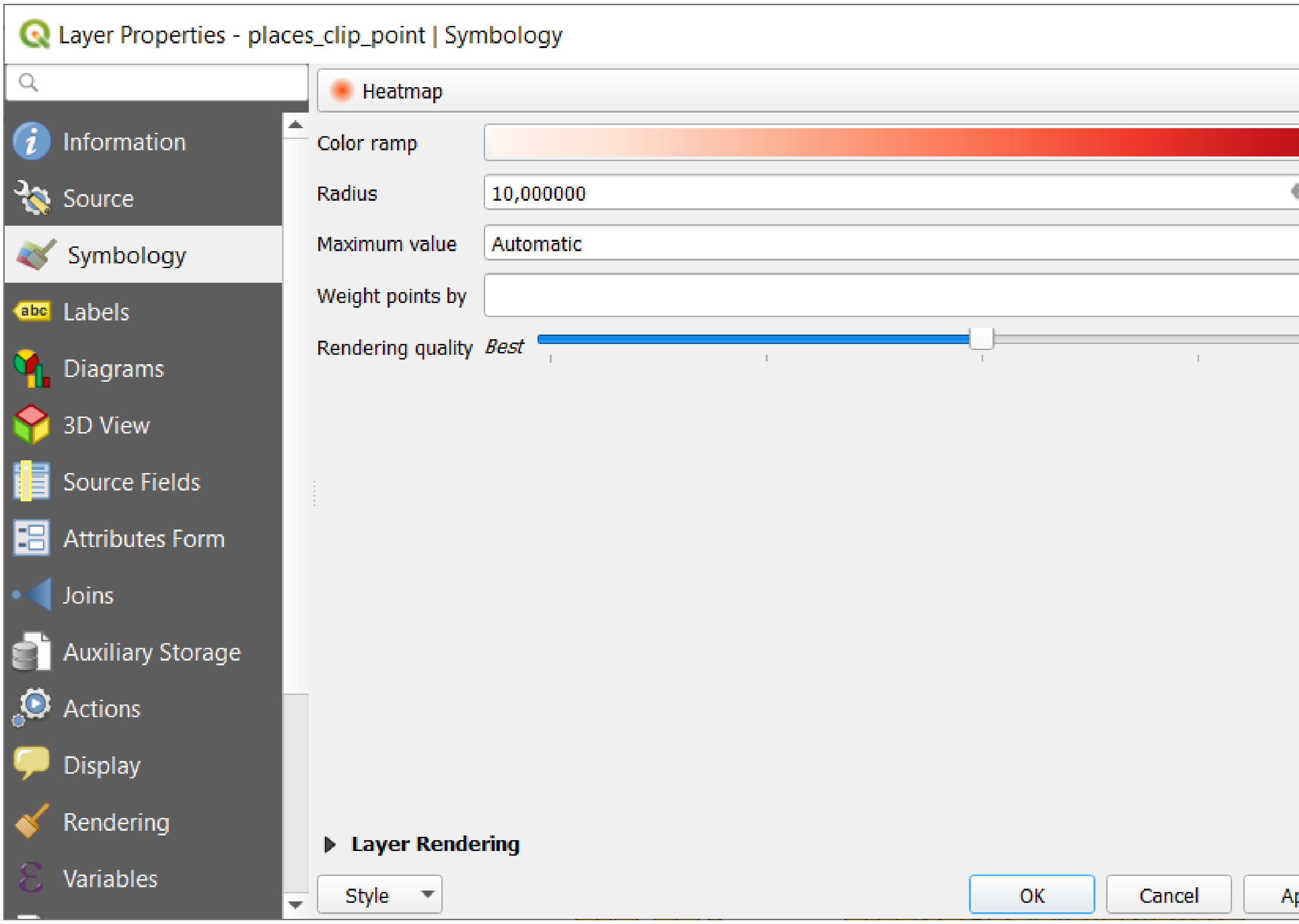
- Right-click the `places_clip` layer on the Layers panel
- Select Properties
- In the menu on the left, select "Symbology"



- In the dropdown menu on top, select Heatmap



- Now you only have to choose the parameters:
  - Color ramp: reds
  - Radius and radius unit of measure: 10 millimetres
  - Maximum value (leave Automatic)



- Then click OK button on the bottom

The result is the visualization of the places layer as a heatmap:

3.7 Inverse Distance Weighting

We can also interpolate a point layer to create a raster layer out of it. One method to do so is the Inverse Distance Weighting (IDW) interpolation technique. To perform IDW interpolation in QGIS we have to rely on the external functionalities provided by GRASS GIS that are directly accessible form the Processing Toolbox of QGIS. The function is called *v.surf.idw*, and it's available at *Processing Toolbox->GRASS->Vector(.v*)*. We can compute a raster by interpolating the `places_clip_point` layer, using as the interpolation variable the population. But first, we need to select all the cities and towns that have a valid population attribute (i.e. > 0): to do so, follow these steps:

- Right click on the `places_clip_point` layer and open its Attribute table
- Click on "Select features using an expression"
- In the window, write the following expression: `"population" > 0`, and then click "Select features"

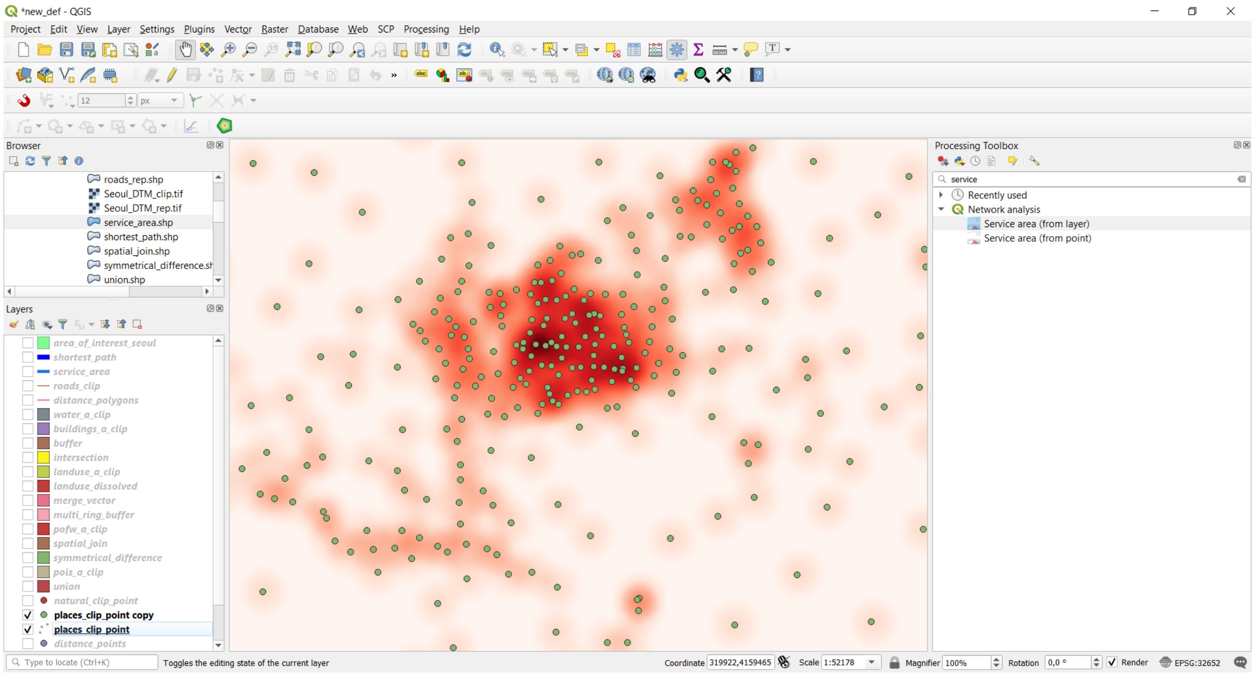
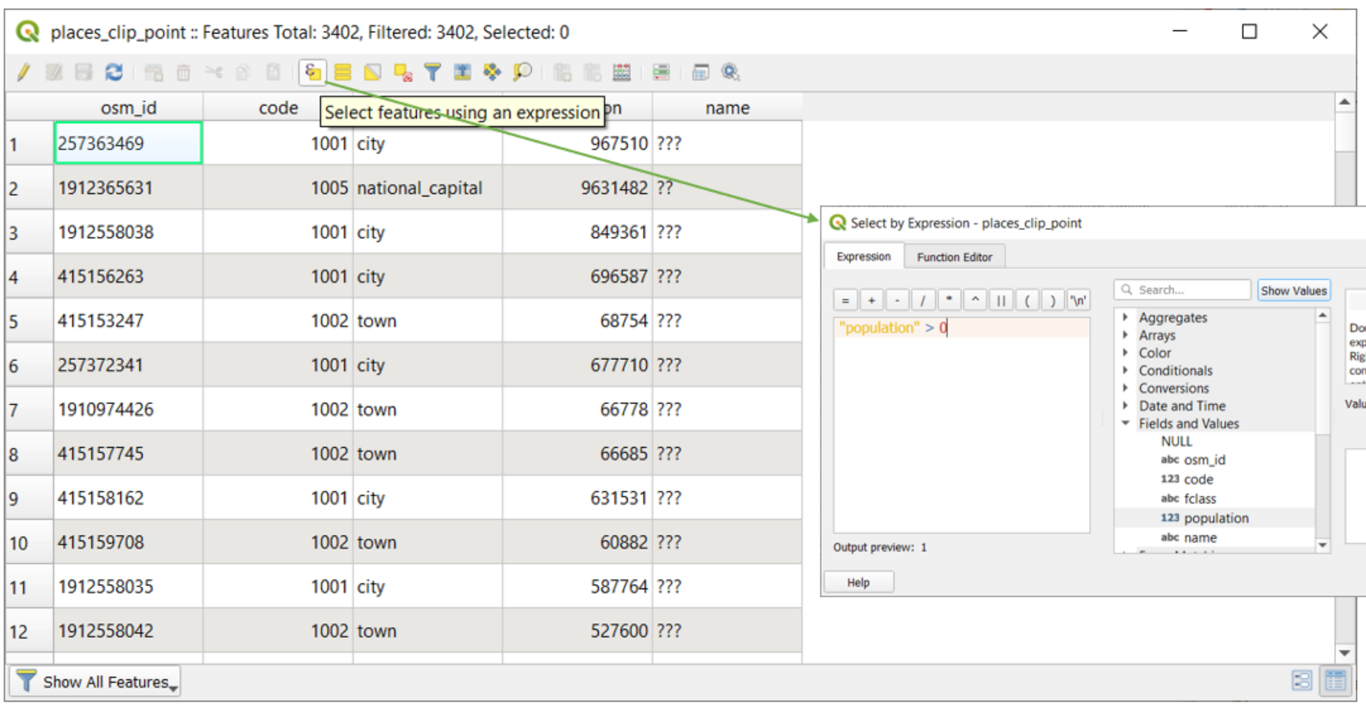


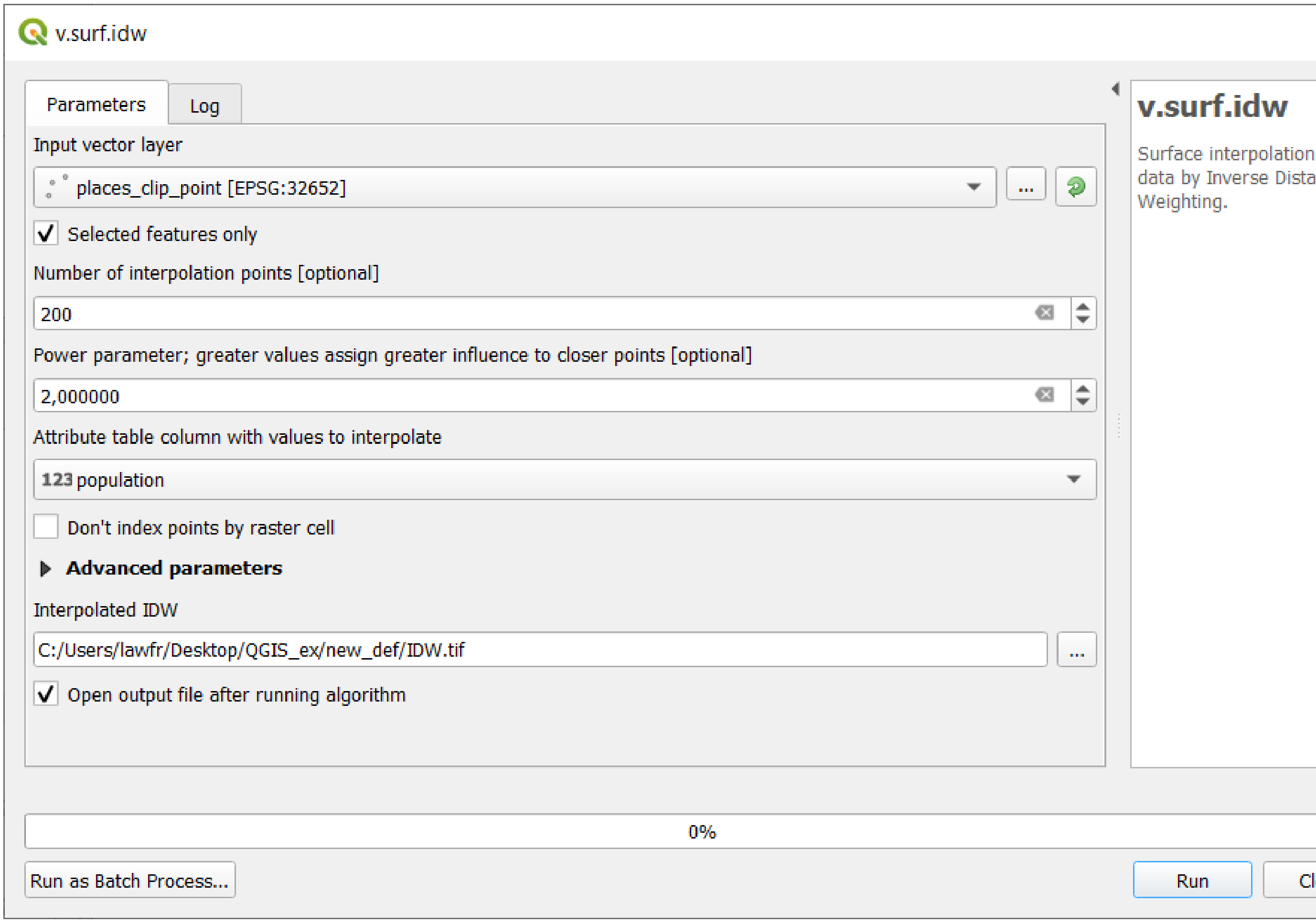
Fig. 3.6.1: Comparison between the heatmap and a copy of the places_clip layer with the point symbology



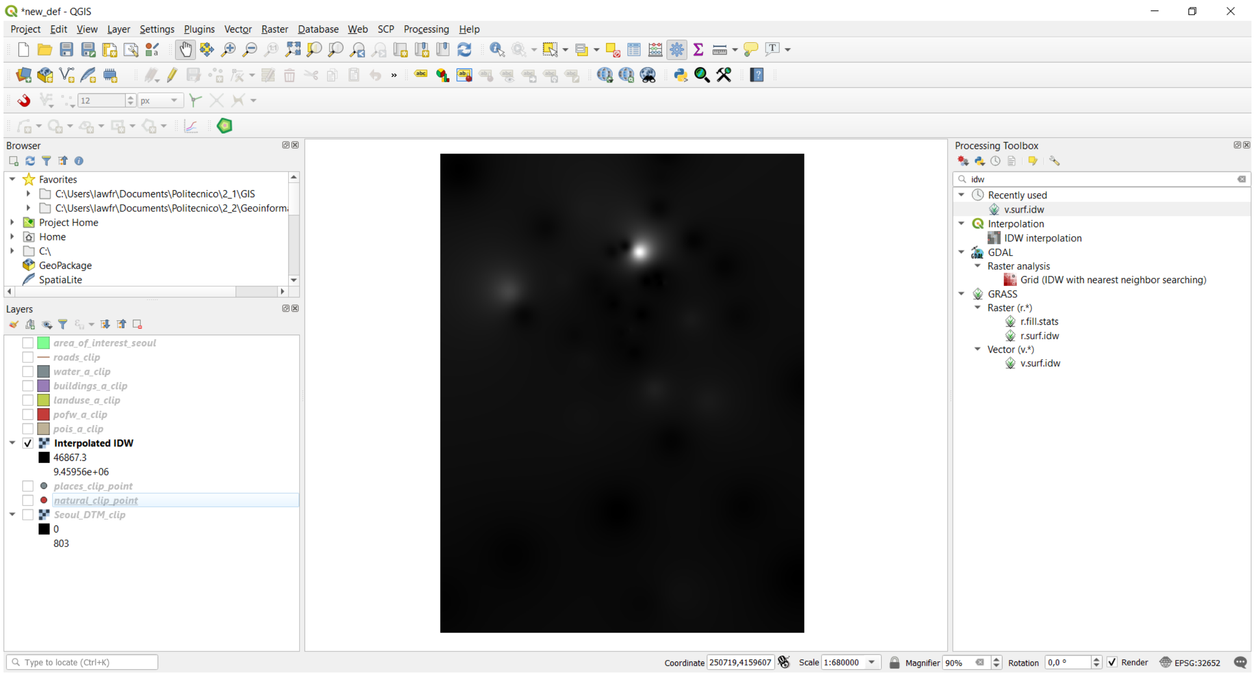
Now that we have selected all the places with population greater than 0, we can proceed with the IDW interpolation. Click on the v.surf.idw function and input the following parameters:

- *Input vector layer*: the `places_clip_point` layer, be sure to tick the "Selected features only" option
- *Number of interpolation points*: 200

- *Attribute table column with values to interpolate*: select "population"
- *Interpolated IDW*: the path and the name of the output raster layer. Note that if left empty a temporary layer will be created

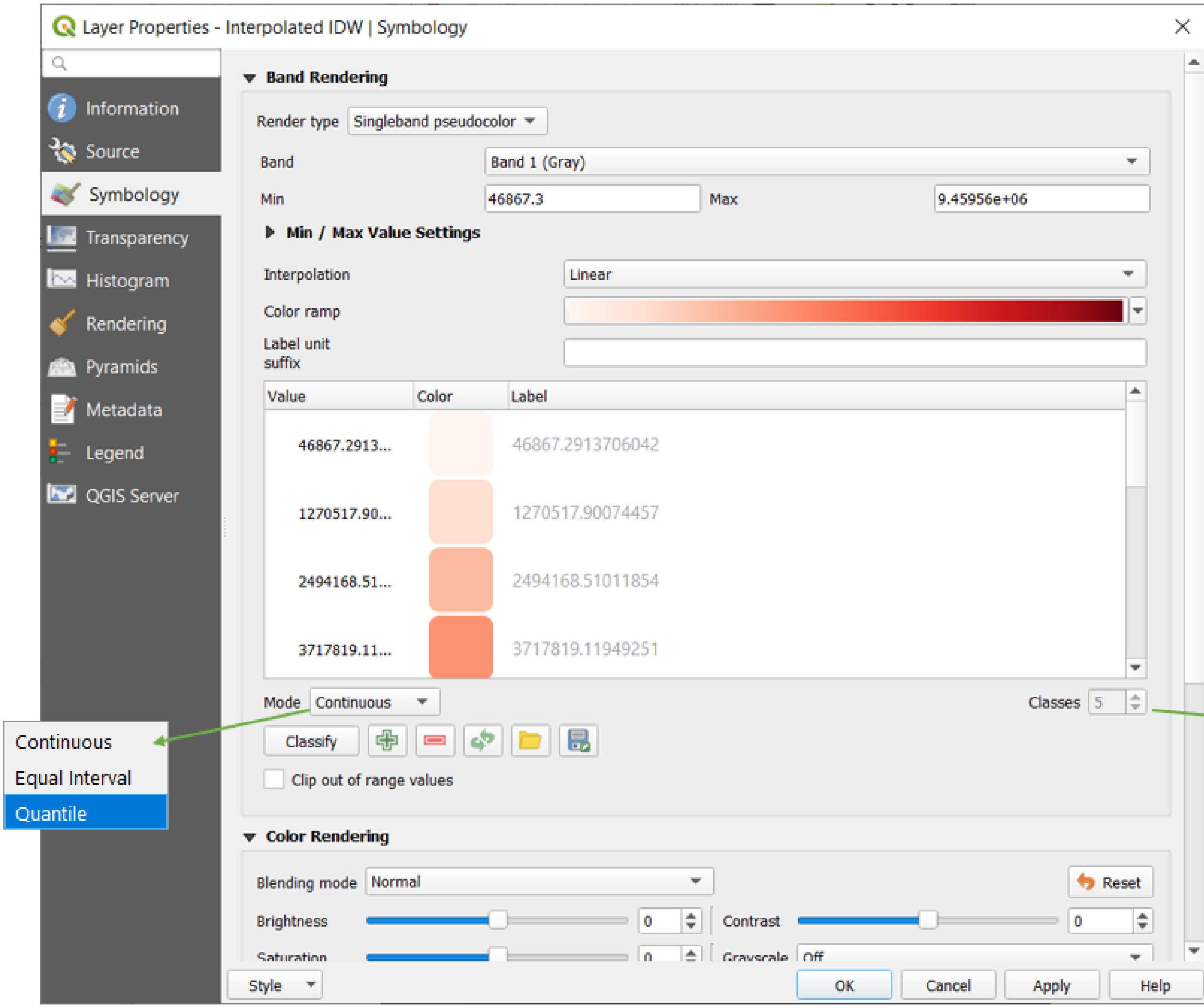


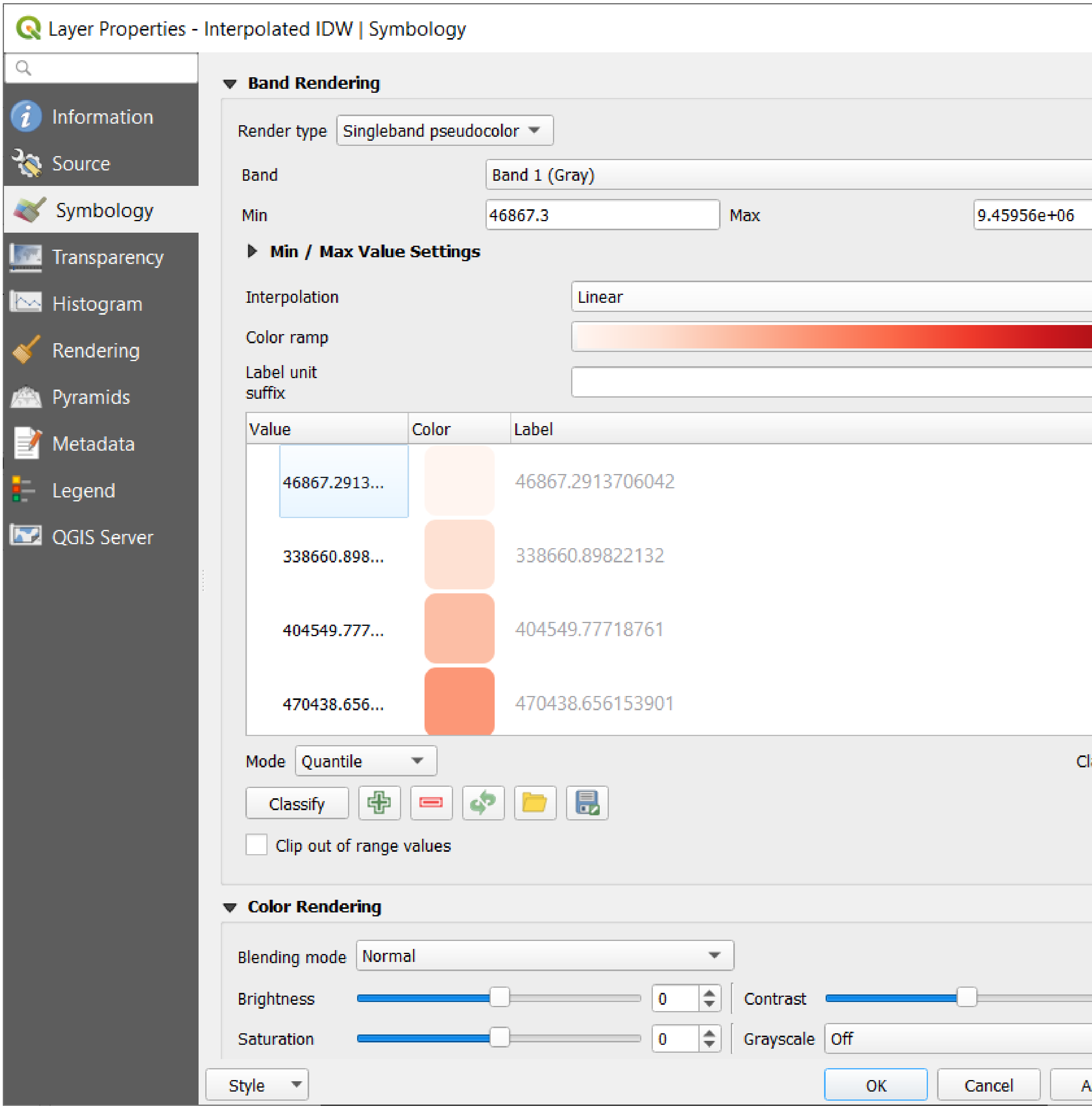
The result should initally look like this:



Since this result is not really optimal for an analysis, we can try to style it in a better way. To do so:

- Right-click on the newly created raster and click on "Properties"
- In the window, select "Symbology" from the menu on the left
- Change the first option ("Render type") to "Singleband pseudocolor"
- Now change the classification mode to "Quantile"
- Change the number of classes to 9
- Click "Ok"


Layer Properties - Interpolated IDW | Symbology
Information
Source
Symbology
Transparency
Histogram
Rendering
Pyramids
Metadata
Legend
QGIS Server
Band Rendering
Render type
Singleband pseudocolor
Band
Band 1 (Gray)
Min
46867.3
Max
9.45956e+06
Min / Max Value Settings
Interpolation
Linear
Color ramp
Label unit suffix
Value
Color
Label
46867.2913...
46867.2913706042
1270517.90...
1270517.90074457
2494168.51...
2494168.51011854
3717819.11...
3717819.11949251
Mode
Continuous
Classes
5
Continuous
Equal Interval
Quantile
Classify
Clip out of range values
Color Rendering
Blending mode
Normal
Reset
Brightness
0
Contrast
0
Saturation
Grayscale
Off
Style
OK
Cancel
Apply
Help



Now the result should be subdivided in 9 classes that contain an equal number of pixels each; the raster should look like this:

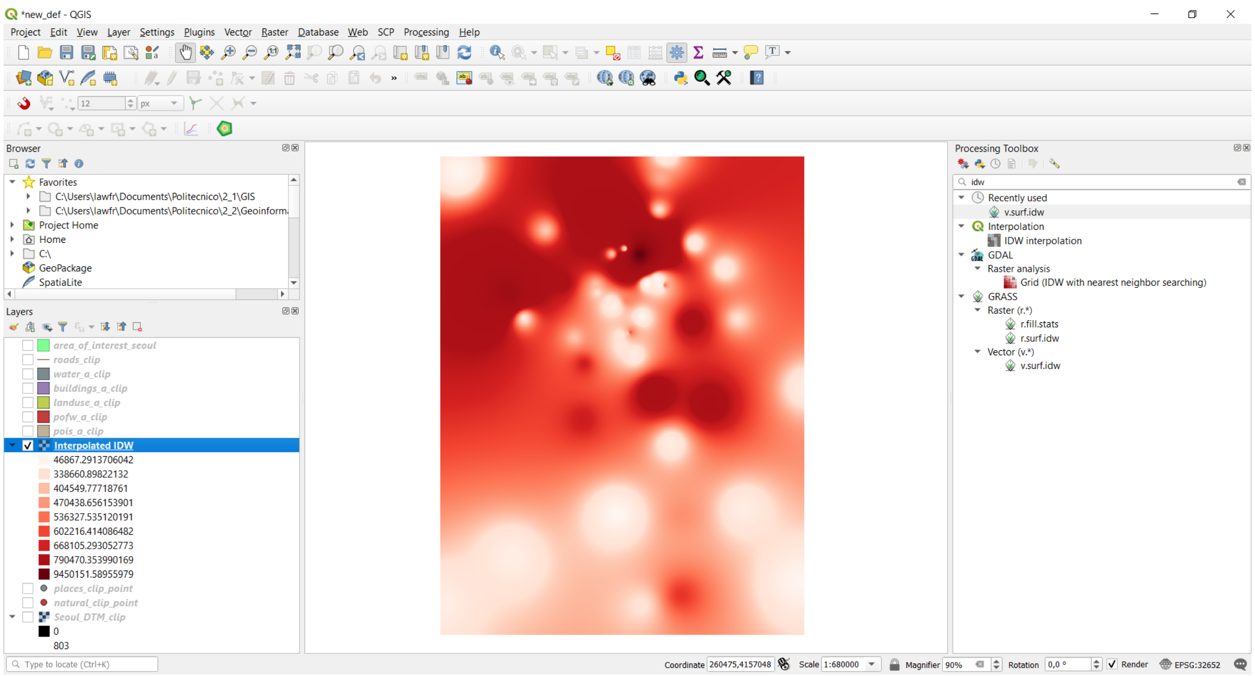


Fig. 3.7.1: Now it's easier to distinguish the variation in population: note for example the dark red area in the top part of the image that represents the area where the capital Seoul is.

CHAPTER 4

Raster operations

4.1 Merge raster

We see now how to Merge multiple raster layers. We do it by combining the Landsat8 imagery bands with the aim of creating a multi-band raster. To do so, we have to first add the raster bands into our QGIS project. From the the Browser panel, navigate to the raster data folder and into the folder containing the Landsat data (`Landsat8_20160519_20170324_01_T1`). Then add the raster ending with B2, B3 and B4 (in Landsat images they represent respectively the blue, green and red channels). These images are already in our project CRS (*EPSG:32652 - WGS 84 / UTM zone 52N*) so once you added the raster data, go directly to *Raster->Miscellaneous->Merge*, and put this input parameters:

- *Input layers*: click on the left symbol and select the Landsat_B2, Landsat_B3 and Landsat_B4 images (`LC08_L1TP_116034_20160519_20170324_01_T1_B2.TIF`, `LC08_L1TP_116034_20160519_20170324_01_T1_B3.TIF` and `LC08_L1TP_116034_20160519_20170324_01_T1_B4.TIF`)
- Tick the "Place each input file into a separate band" checkbox: when not activated, this option is used to merge raster layers covering different area and not overlapping between each other. In our case we want to merge images that totally overlap between one another
- *Output data type*: change it to UInt16 to be consistent with the original rasters
- *Merged*: the path and the name of the output raster layer. Note that if left empty a temporary layer will be created

The obtained multiband raster at first will look like this:

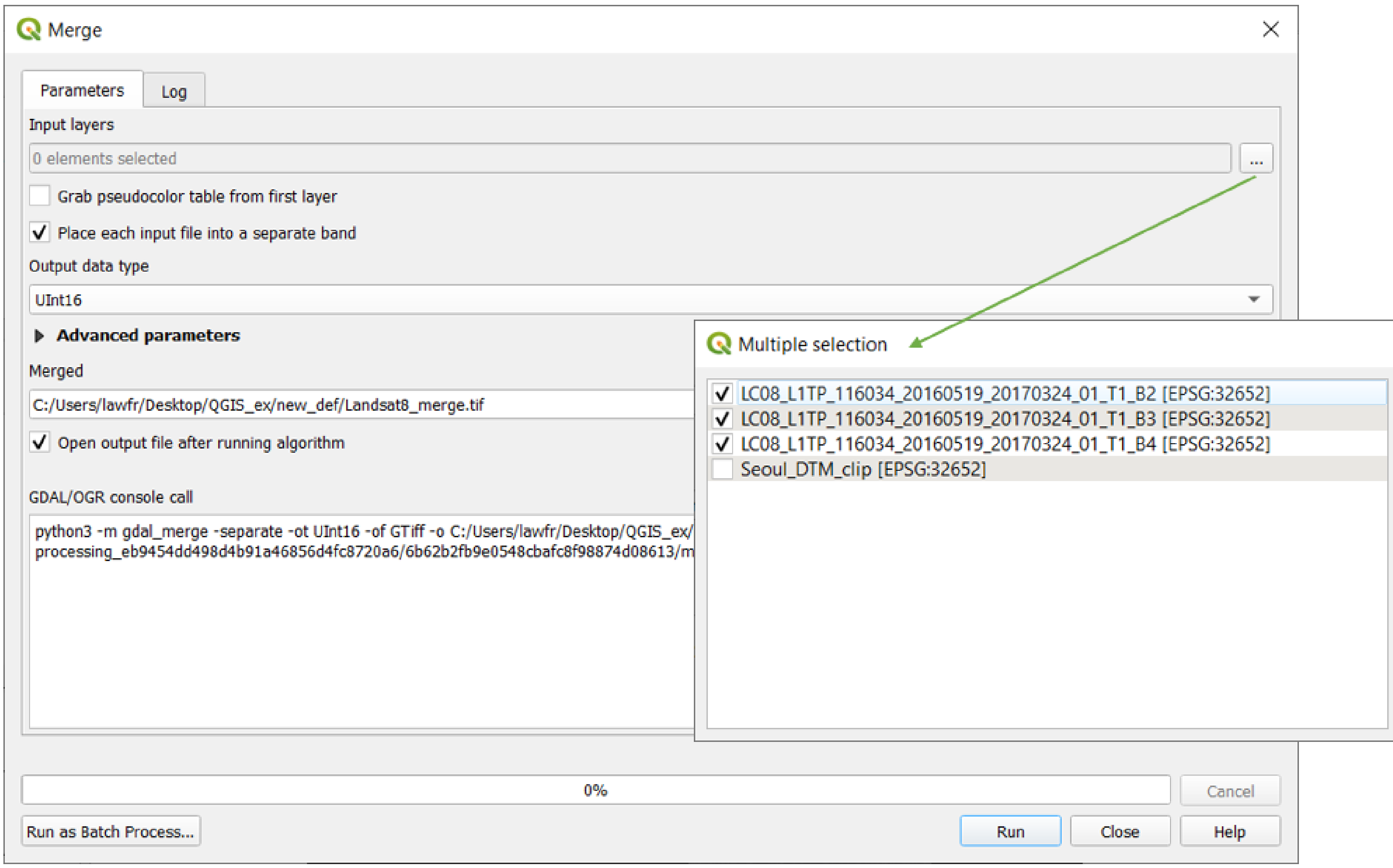
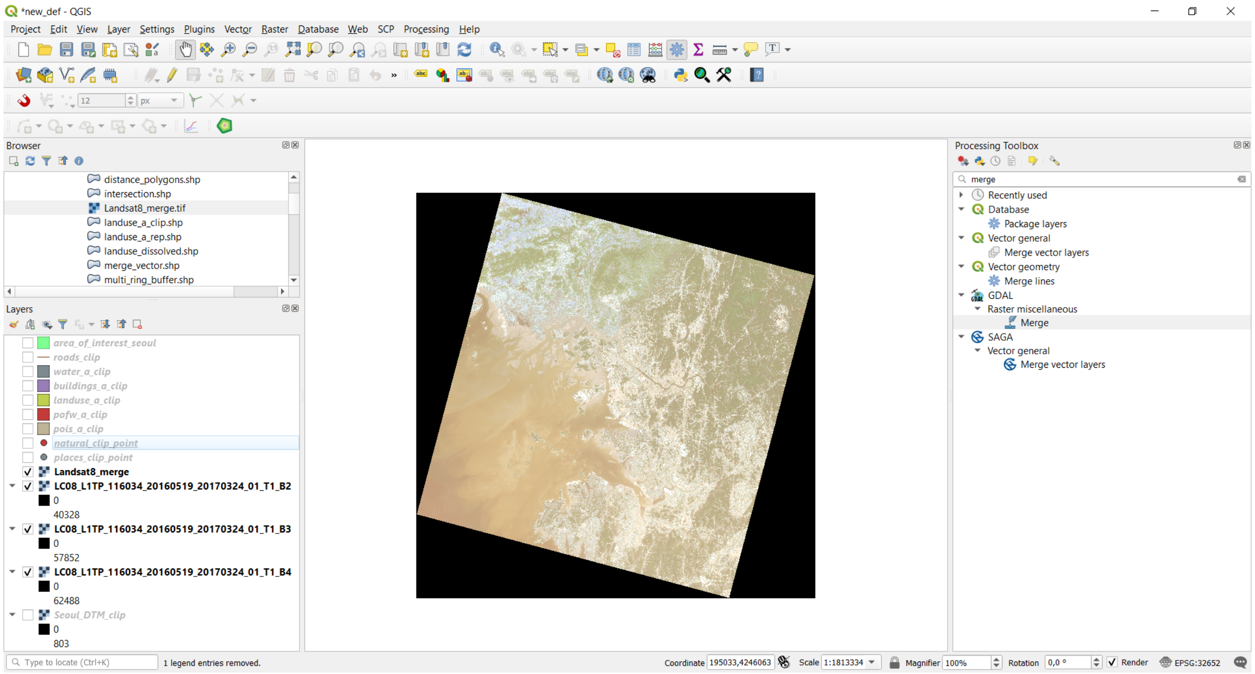
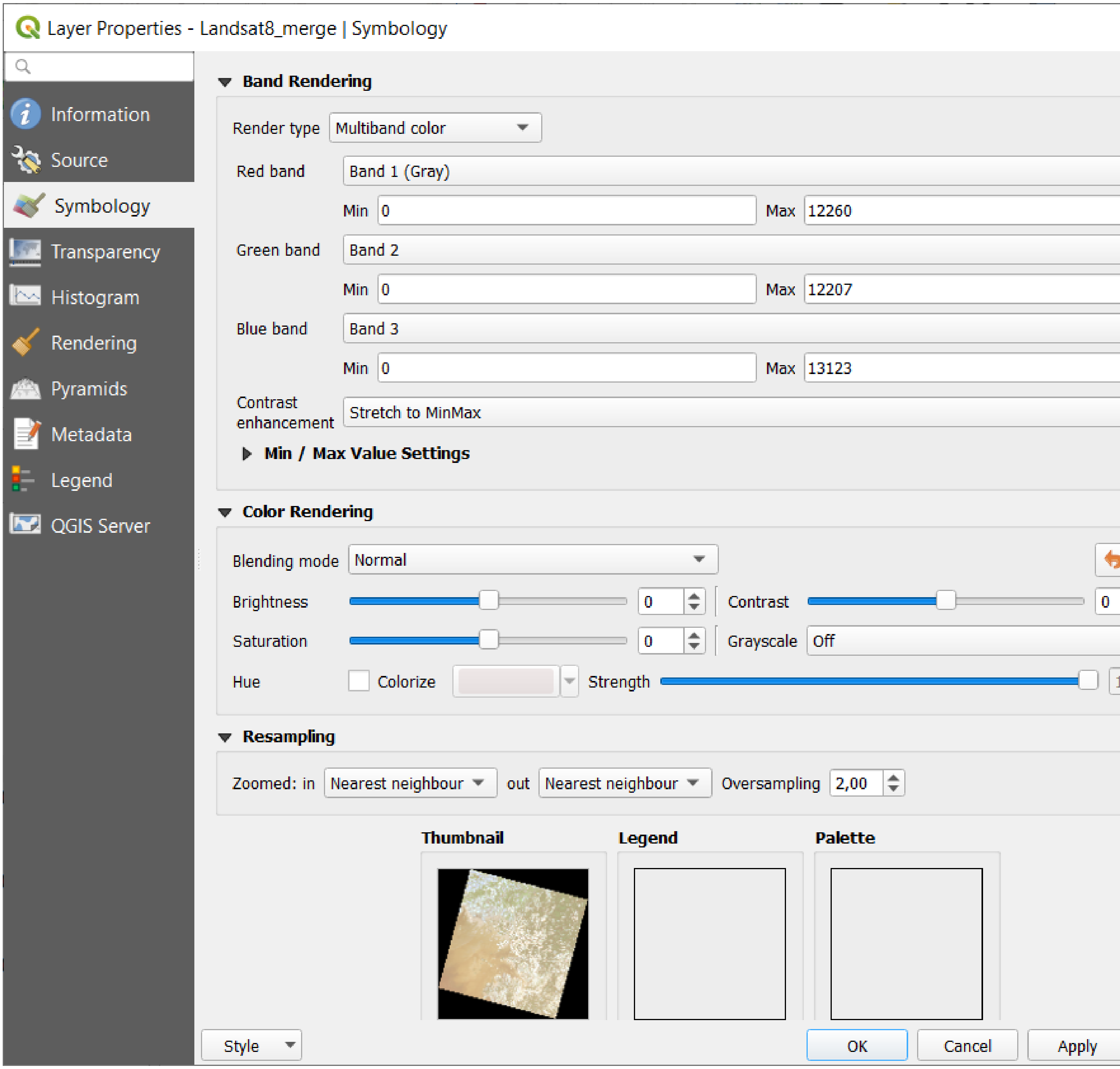


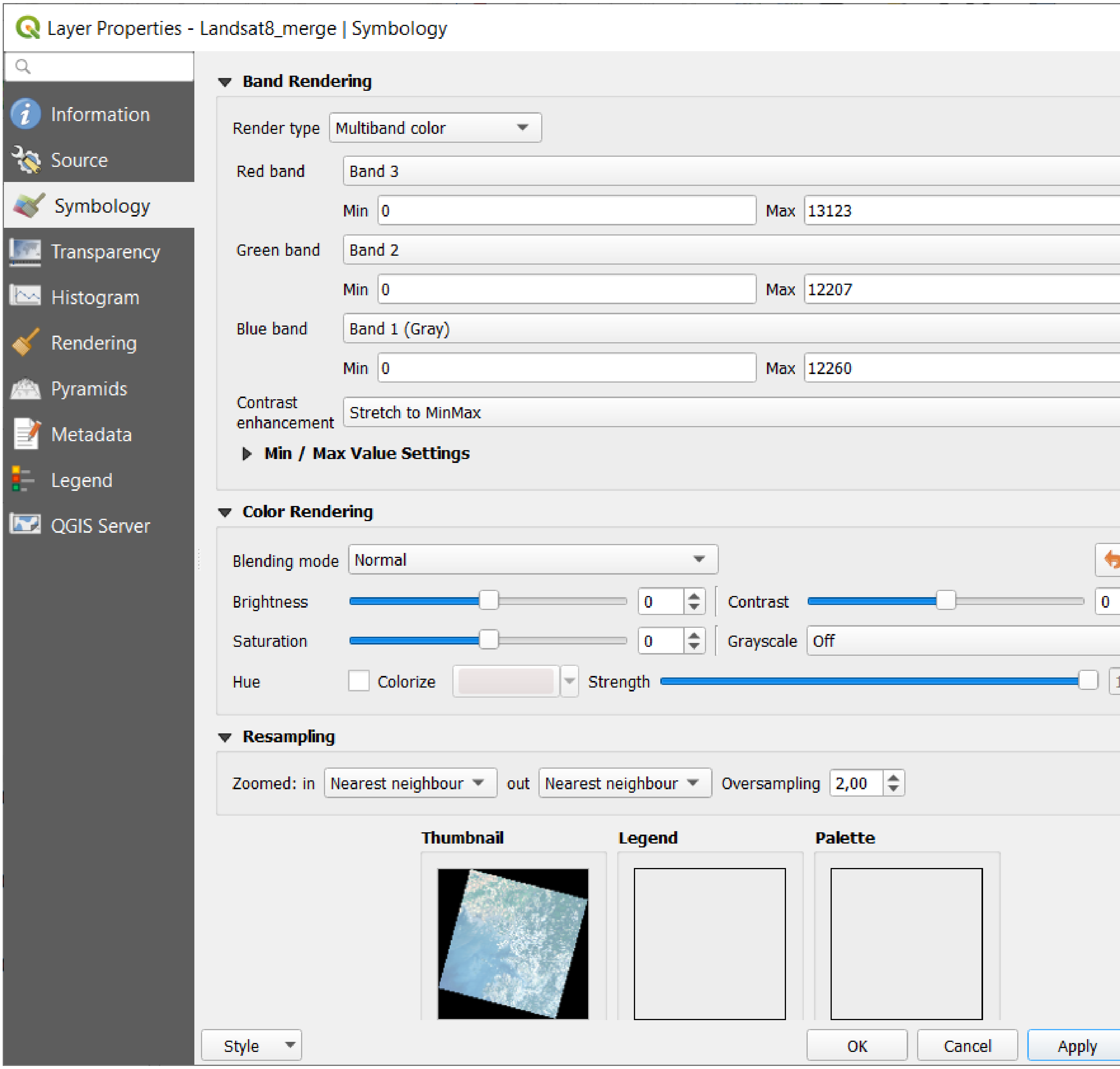
Fig. 4.1.1: The merge raster function window



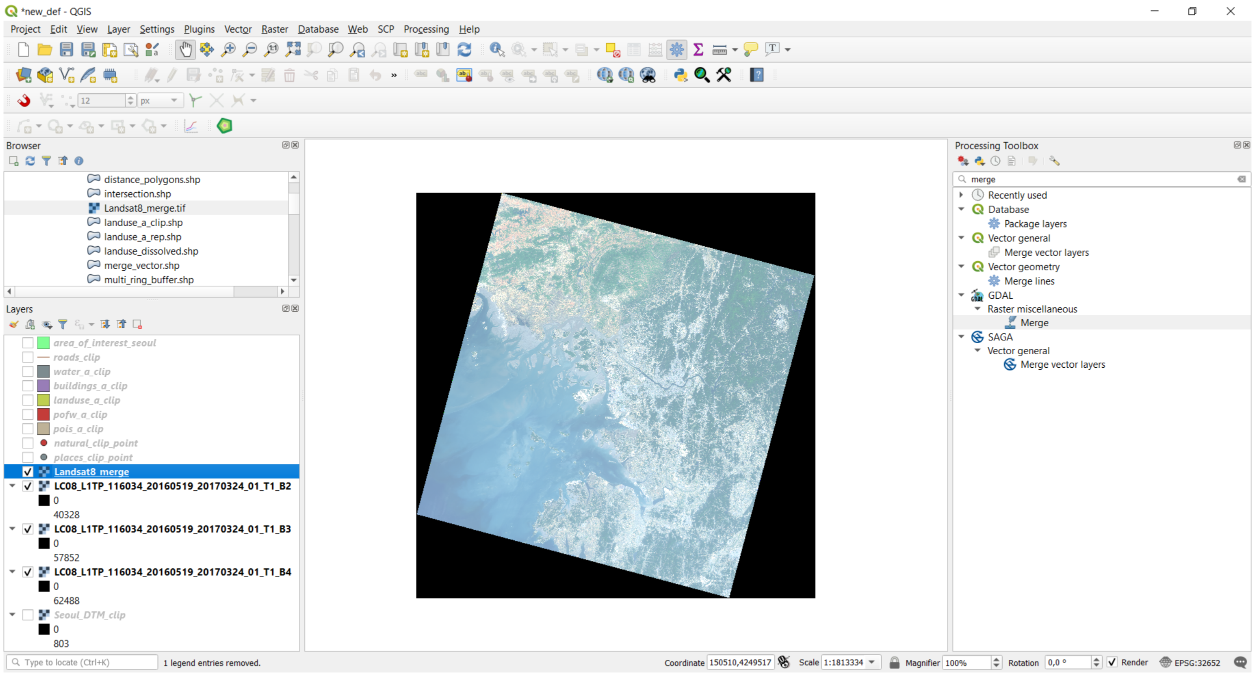
To obtain what is called a “true color image”, so a multiband raster whose color resembles the real ones, you have to change the band assigned to each color. To do so, right-click on the `Landsat-merge` raster and select “Properties”. Go in the “Symbology” section, and you will see a window like the following:



Then, assign to the Red band the Band 3, to the Green band the Band 2 and to the Blue band the Band 1.



Click "Ok" and the true color map obtained should look like this:



Note: This is not an accurate true color map because there was no atmospheric correction applied to the Landsat images, so you'll find that its colors are brighter than the real ones.

CHAPTER 5

Index

- genindex

Index