

---

# Mappy

**Luca Penasa**

**Aug 22, 2022**



# Contents

<b>1</b>	<b>Documentation</b>	<b>3</b>
1.1	Installing Mappy . . . . .	3
1.2	Quick start tutorial . . . . .	4
1.3	Mappy rationale . . . . .	13
1.4	Toolbox algorithms . . . . .	16
1.5	Citing Mappy . . . . .	19

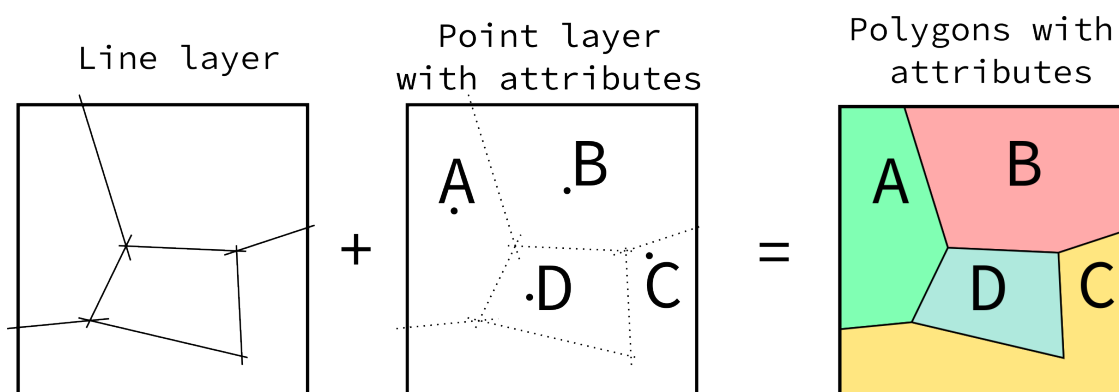


**Warning:** This project is under active development. This means it is often updated and the documentation might sometimes be out of sync with the plugin itself.

Mappy has been developed for teaching purposes and some operations might be computationally intensive on large datasets. This is especially true for the generation of the clean contacts. See also warnings in [Generating the polygonal map](#).

**Important:** This plugin was developed as the result of the PLANMAP project (N 776276) and is now maintained thanks to the EUROPLANET-GMAP infrastructure (N 871149) and supported by the JANUS camera team of the JUICE mission (ASI-INAF 2018-25-HH.0).

**Mappy** is a QGIS plugin that simplify some useful operations needed to create geological maps by the point-and-contacts principle:



Mappy features a map generation dialog and a dedicated toolbar that can be used during mapping to quickly regenerate your map. This is the suggested way to work with mappy and its use is documented in [Quick start tutorial](#).

If you want to know more of the idea behind this tool, get an overview of geological mapping with GIS have a look at the [Mappy rationale](#) section.

Mappy also features ad-hoc providers that can be reused for different projects and to generate your maps by calling the appropriate algorithms from the toolbox. Find out more in the [Toolbox algorithms](#) section.



## 1.1 Installing Mappy

### 1.1.1 Via the official QGIS plugins repository

The plugin has now been contributed to the [QGIS Python Plugins Repository](https://plugins.qgis.org/plugins/mappy/)<sup>1</sup> to make installation easier. To install it just launch QGIS and find the Plugin Manager under *Plugins ► Manage and Install*. Under *Settings* remember to enable *Show also experimental plugins* and refresh the plugin database with *Reload Repository*. Under *All* you should now be able to find *mappy*

**Warning:** If you experience problems installing Mappy and you already had a previous version of the plugin installed (e.g. during the [PLANMAP-GMAP Winter School](#)<sup>2</sup>) you might try to remove the previous version of the plugin before proceeding with the normal installation. Very early versions of the plugin might indeed cause conflicts with the currently packaged version.

### 1.1.2 From GitHub zip package (for the latest release)

If you want to try the latest version available, you can use the pre-generated zip packages that can be installed in qgis by using the plugin manager. Download it from [Releases](#)<sup>3</sup> and install the zip using the plugin manager. Releases marked as *latest* are automatically in sync with the master of this repo. Tagged releases (e.g. named v0.1.2) should be considered as stable versions, although the plugin itself is still largely experimental.

---

<sup>1</sup> <https://plugins.qgis.org/plugins/mappy/>

<sup>2</sup> <https://planetarymapping.eu>

<sup>3</sup> <https://github.com/europlanet-gmap/mappy/releases>

## 1.2 Quick start tutorial

This section should provide all the information needed to quickly start using mappy. If you are really in a hurry, jump to [Generating the polygonal map](#).

**Warning:** We assume you already loaded the basemaps you intend to use for your mapping. If you haven't, do it now! It will also simplify the selection of a meaningful CRS for your project. CRSs are extremely important when you are first setting up a mapping project. Have a look [here](#)<sup>4</sup> for an introduction.

### 1.2.1 Initial setup

---

**Important:** You can also skip this section “*Initial setup*” and use the mappy algorithm [Quick Mapping Project Setup](#) to easily set up the minimum dataset needed to quick-start your mapping project. This is still a suggested reading, as it provides some useful tips!

---

Two different layers should be created:

- A line layer (either Line or MultiLine). The associated fields are not mandatory, but you might want to consider adding a couple of fields like *certainty* and/or *type* to represent the type of contact you are tracing. Keep in mind that more information you add to your vector data, easier will be to style them later or to reuse them for other purposes.
- A point layer. Here we need at least one field that will be used as unique identifier for each geological unit that will be mapped. Choose for example *geo\_units*.

Take your time to establish the naming of your fields to be meaningful and readily understandable. The same apply for the layers. It will not influence mappy in any way, but it will help you creating a clean dataset.

The type of fields can be freely chosen, but it is highly suggested you try to define beforehand the entries that will be used when populating your map. For example a *string/char* field can be used as type for the *geo-units* field, but it might be difficult to be consistent when entering those long strings by hand (they could be easily misspelled), thus short names or a code for each unit might be preferable.

---

**Tip:** To avoid mistakes when typing in the name of the geological unit in a string field, you could define them beforehand customizing the [attribute form](#)<sup>5</sup> used to enter the field's values. An *Unique Values* or a *Value Map widget*<sup>6</sup> might be used for this task. An example will be demonstrated later.

---

The layers can be created in any format supported by QGIS (e.g. ESRI Shapefiles), but we suggest organizing your work within a [geopackage](#)<sup>7</sup> file. This open format make it possible to store within a single and portable file any number of different vector layers. To create a new geopackage use *Layer ► Create Layer ► New Geopackage Layer* or CTRL+SHIFT+N.

---

<sup>4</sup> [https://docs.qgis.org/testing/en/docs/gentle\\_gis\\_introduction/coordinate\\_reference\\_systems.html](https://docs.qgis.org/testing/en/docs/gentle_gis_introduction/coordinate_reference_systems.html)

<sup>5</sup> [https://docs.qgis.org/testing/en/docs/user\\_manual/working\\_with\\_vector/vector\\_properties.html#](https://docs.qgis.org/testing/en/docs/user_manual/working_with_vector/vector_properties.html#)



**New GeoPackage Layer**

Database: /path/to/mapping\_project.gpkg

Table name: points

Geometry type: Point

☐ Include Z dimension ☐ Include M values

EPSG:4326 - WGS 84

**New Field**

Name: geo\_units

Type: Text Data

Maximum length: 10

Add to Fields List

**Fields List**

Name	Type	Length
geo_units	text	10

Remove Field

**Advanced Options**

Layer identifier: points

Layer description: Indicator points for the geological map

Feature id column: fid

Geometry column: geometry

☒ Create a spatial index

OK Cancel Help

Fig. 1.1: Creating a new geopackage layer in QGIS with one point layer named `points`, with one field `geo_unit` of type `Text Data`. Once the geopackage is created new layers can be appended to the same file by selecting the same Database geopackage. The same operation should be performed for creating the line layer that will contain the contacts.

**Warning:** Remember to set a meaningful CRS<sup>8</sup> for the layer. It might be worth taking some time to verify the CRSs that are used in the basemaps you will be using for your mapping and chose among those the most suitable. If you have loaded your basemap at the beginning as suggested you might just need to select the “Project CRS” from the options.

## 1.2.2 Drawing the contacts

Assuming you have already loaded your basemaps you can now start tracing your contacts. Just enable the editing (use *Toggle Editing* from right-clicking on the layer) for your newly created layer and add new lines. Something very important that you should keep in mind is that you should not be focusing on the outline of each unit, but rather determine the contacts between the different units.

## 1.2.3 Defining indicator points

---

**Tip:** Mappy will generate indicator points for any unlabeled unit by default. But you will still need to provide a point layer to be used for labelling the units.

---

The next step consist in labelling each region we have identified with the contacts. To do so a point layer will be used: a new point will be placed anywhere within the given unit and its fields will be populated accordingly.

We will use the following unit’s shortcuts:

Code	Legend
SCA	Outer escarpment of the volcano
EDF	Main shield volcanic edifice
SC	Summit calderas

and populate the points layer. If you want you can setup dedicated field widget to ease this work ( *Right Click on Layer ► Properties ► Attributes Forms* ).

After enabling the point layer for editing we add new points by left-clicking with the mouse in the location of interest. You must activate the adequate tool for adding new points (look for *Add Point Feature* button or use *CTRL+.*). For each unit you will need to enter the corresponding unit. If you set up a dedicated *Attribute form* the dialog will look something like this:

After adding all the needed points the results should be something like this:

Remember to save the layers by using the *Save Layer Edit* in the *Digitizing* toolbar.

---

attributes-form-properties

<sup>6</sup> [https://docs.qgis.org/testing/en/docs/user\\_manual/working\\_with\\_vector/vector\\_properties.html#edit-widgets](https://docs.qgis.org/testing/en/docs/user_manual/working_with_vector/vector_properties.html#edit-widgets)

<sup>7</sup> <https://en.wikipedia.org/wiki/GeoPackage>

<sup>8</sup> [https://docs.qgis.org/testing/en/docs/gentle\\_gis\\_introduction/coordinate\\_reference\\_systems.html](https://docs.qgis.org/testing/en/docs/gentle_gis_introduction/coordinate_reference_systems.html)

<sup>9</sup> [https://en.wikipedia.org/wiki/Olympus\\_Mons](https://en.wikipedia.org/wiki/Olympus_Mons)

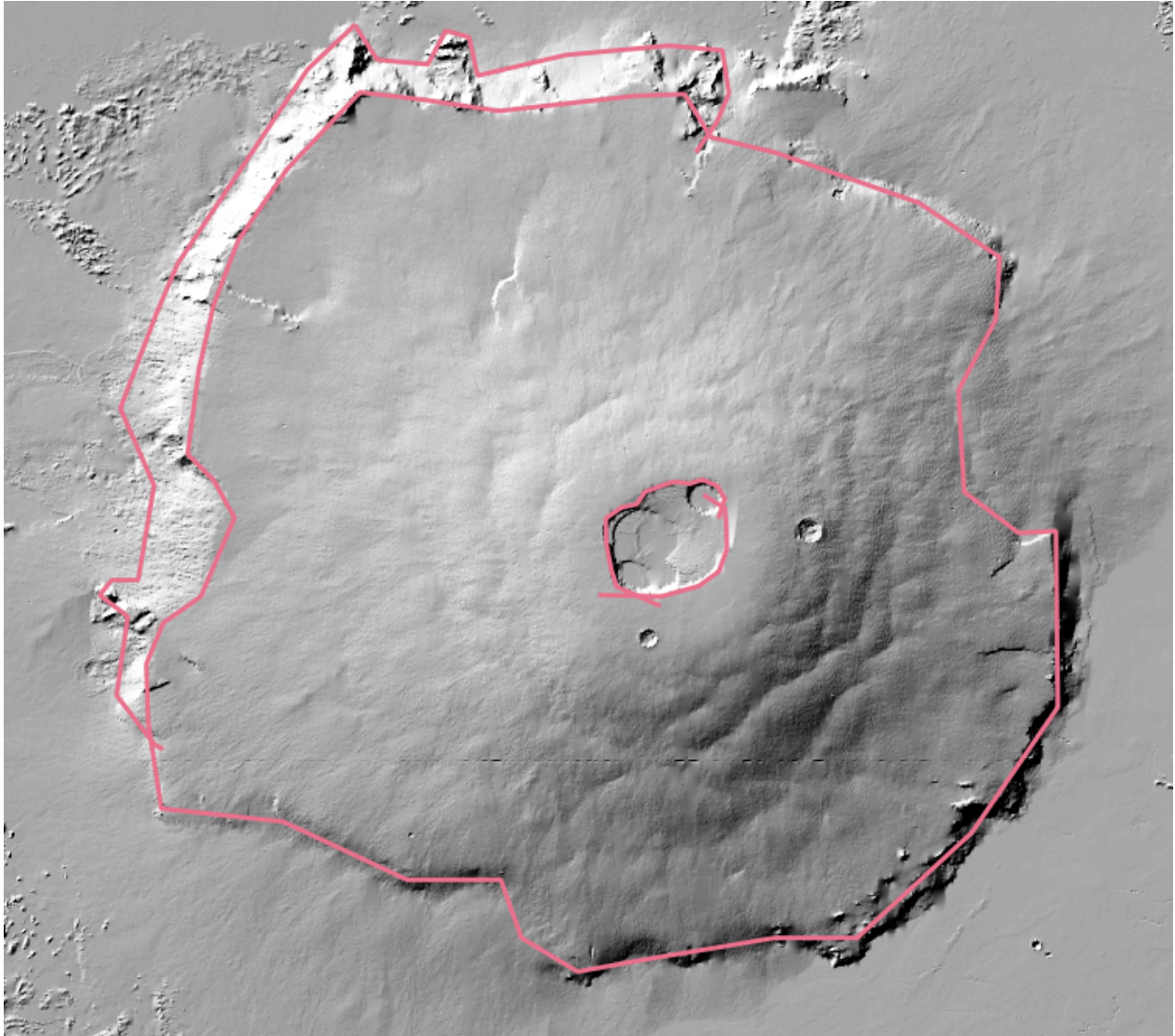


Fig. 1.2: Example of linework on the [Olympus Mons](#)<sup>9</sup> volcanic edifice on Mars, we used the lines to define the contacts between different morphological units. Notice that the linework does not need to be precise (here exaggerated), we just need the line to be intersecting to correctly define the units. Also multiple lines can be used to define one contact, provided they all intersect each other (see the calderas on the summit).

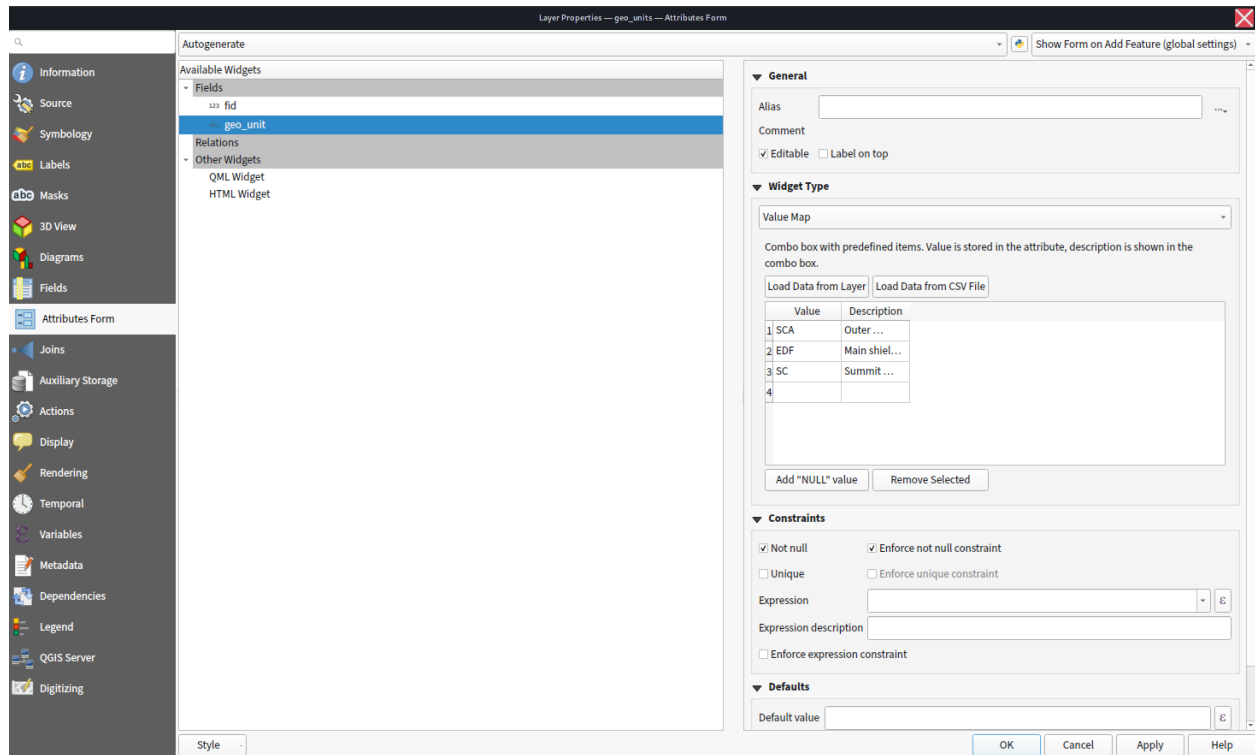


Fig. 1.3: Example of attribute form set up with the predefined unit's names (notice you can also create them as a CSV file to load).

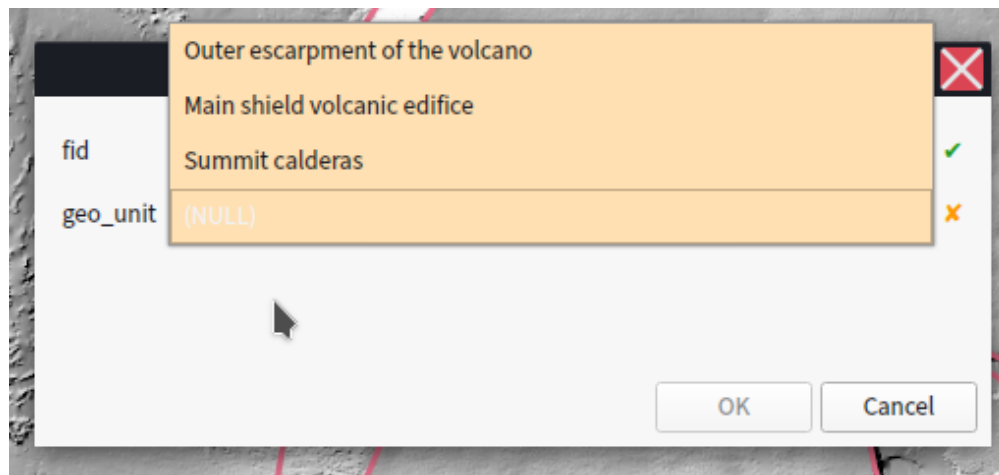


Fig. 1.4: Example of Feature Attribute Dialog, customized with a *Value Map* field widget. It ensures no wrong codes can be inserted into the database.

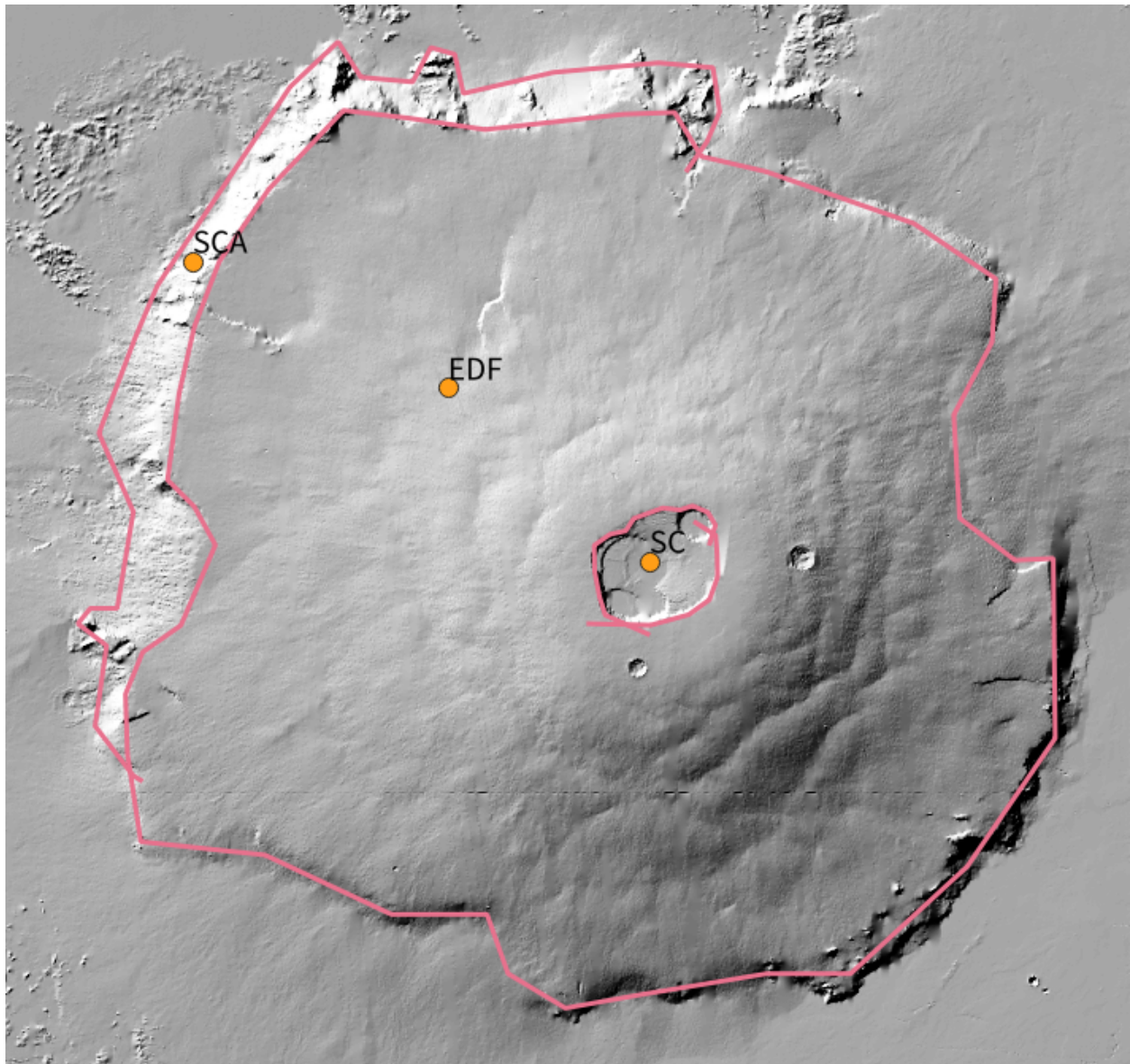


Fig. 1.5: Example of points used to define the names of the units. Short-codes were used to uniquely identify the unit. To show the labels follow *Right-Click on layer* ► *Properties* ► *Labels* and configure accordingly.

### 1.2.4 Generating the polygonal map

Mappy can be accessed through a dedicated toolbar:



The gear button will toggle the configuration dialog:

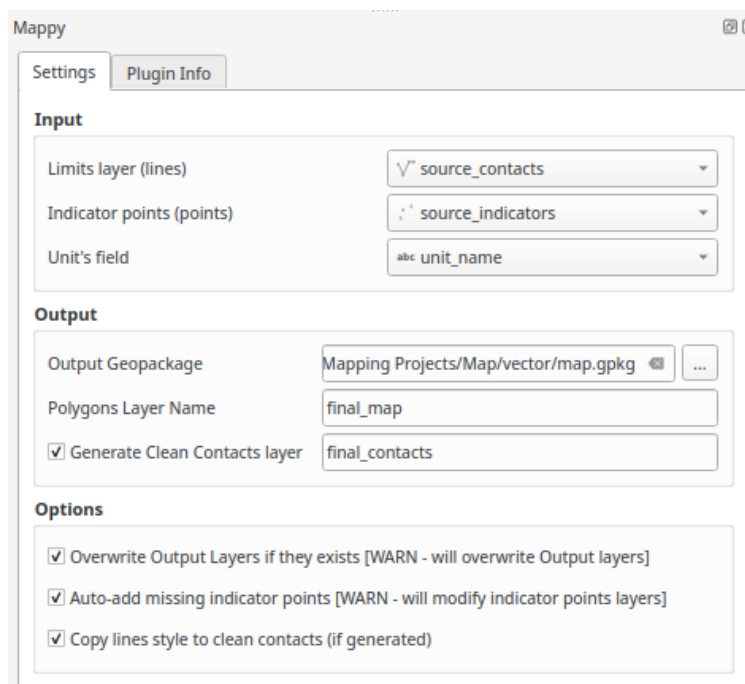


Fig. 1.6: Mappy settings dialog

The second button (Recompute Map) is used to quickly recompute the map on the basis of the configuration provided in the main dialog. It is meant to be the only button that you are going to use every time you need to refresh the polygons during the mapping.

The config dialog requires you to set:

#### Input

- **Line-geometry layer:** it will be used as source of your contacts, any line layer will work (you can organize the fields of your contacts freely).
- **Point-geometry layer:** it will be used to label the newly generated polygons, and it must have at least one field that will be used as unit's name.
- **Unit's field** is the field of the points layer that is used by mappy. Although mappy transfer all the existing fields to the polygons the field is still needed so that mappy knows what field to use for generating the legend/colors.



## Output

- **Output Geopackage:** The geopackage in which the output layers will be written
- **Polygons layer:** the name that will be used in the output geopackage to name the polygonal layer
- **Clean Contacts layer name:** (if enabled) the name for the clean contacts layer in the output geopackage

If clean contacts generation is enabled mappy will generate a copy of your input contacts after cleaning them from dangling ends. This layer is intended to be used as overlay for your final layout

**Warning:** The operation of dangling-ends removal that is used to generate the *Clean Contacts* layer is computationally intensive in the present implementation. If you experience qgis unresponsiveness during the map computation try to disable this feature and use it only at the end for finalizing the map.

## Options

- **Overwrite Output Layer:** it will take the freedom to overwrite the output polygonal layer if it is already present
- **Auto-add Missing Indicator Points:** it will automatically compute and add any missing indicator point. You will then still need to populate the fields to label your units.
- **Copy Line Style:** it will copy the styling of your contacts to your *Clean Contacts* layer (if generated). In this way you can work with correctly styled layers and have the style copied to final version of your contacts.

---

After setting and double-checking your config you can now press the *Recompute Map* toolbar button and have your polygons computed.

### 1.2.5 Styling the map

**Warning:** Mappy will take care of styling the polygons for you, it will also copy the style of you contacts layer to the Clean Contacts layer. Anyway if you need to style them by hand here are some instructions.

The map can then be styled with *Right-click on Layer ► Properties ► Symbolology*. A *Categorized* symbology might be used, like so (you can use the *Classify* button to automatically add all your unit's definitions, also remember to modify the *Legend* items if you want a full description of the map in the printing layout):

---

**Tip:** Notice you might want to set the *Stroke Style* to *No Pen* for the polygonal layer, so that only the line contacts will be visible in the final map.

---

your map might look like this:

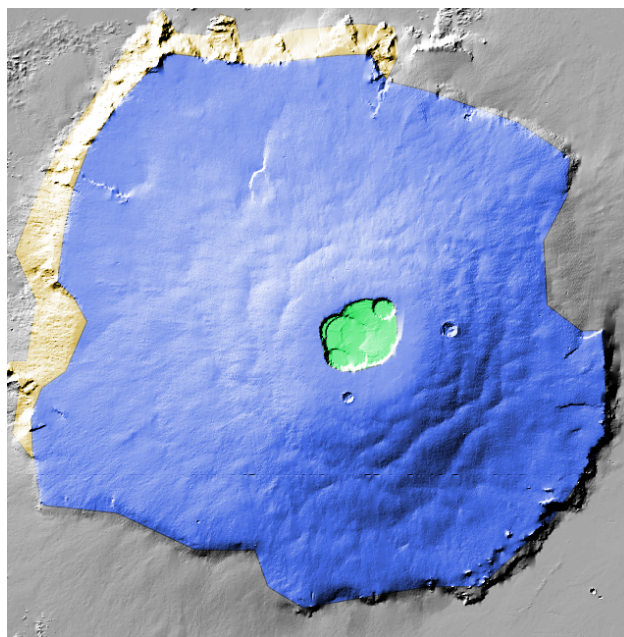
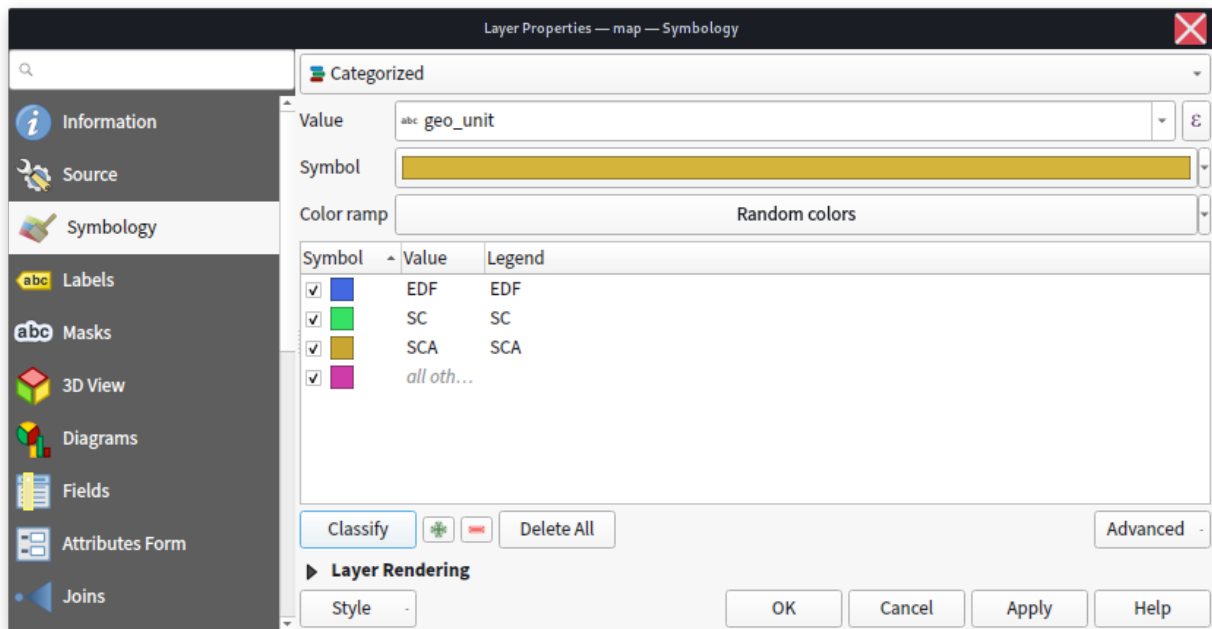


Fig. 1.7: Polygonal map with styling.



**Tip:** To enable this kind of transparency find the *Blending Mode* option under the layer's *Symbology* ► *Layer Rendering* dialog, and set it to *Overlay*

## 1.3 Mappy rationale

**If you are not interested in the theory behind mappy you can skip to the [mappy tutorial](#)**

This document delineates some useful guidelines and highlights some common problems encountered by mappers when approaching a mapping project, with a special attention to planetary sciences. Although the discipline is well established the tools used to generate new maps have been progressively digitized thanks to Geographic information Systems (GIS) software.

There are many GIS software available both commercially and open-source. Open source tools are now extremely mature and can compete with commercial products under many point of views. This guide is based on QGIS but in large part the same concepts can be ported to your preferred GIS environment.

**Tip:** Some common open-source GISs are QGIS, GRASS, OpenJump, uDig, SAGA, GVSIG

### 1.3.1 Geological mapping

Depending on the scope of the geological survey the cartography can depict different types of content: whether lithostratigraphy, chronostratigraphy, morphology, or other criteria are considered, the building blocks of a geological map is the geological unit, as the base element used to subdivide a region into geologically homogeneous patches of terrain (Fig. 1.8) .

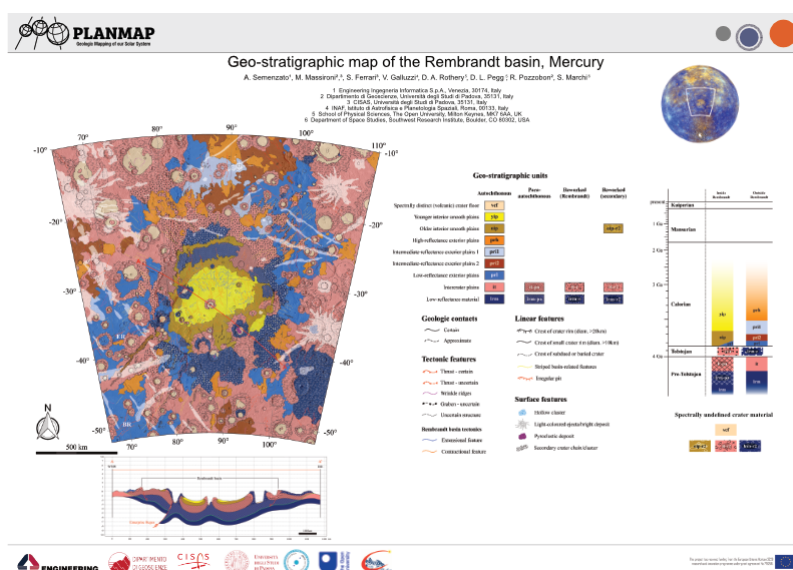


Fig. 1.8: An example of a geological map realized on Mercury

The basic building blocks of a map are the **units**, which are represented with different colors on the map. Each unit is separated from the surrounding ones by lines (contacts). Some key points to keep in mind when realizing a geological map are:

- **Completeness:** no area within the region subject to the mapping can be left unassigned.
- **Limited:** the map should be limited to a specific region, which is defined by an exterior boundary.
- Any number of additional properties can be added on top of the geological map, in the form of linear elements or polygonal shapes (which can exploit either transparency or pattern infilling)

---

**Tip:** There are quite a number of predefined infilling patterns and symbology available for geological mapping. The main reference is the [SGS FGDC Standard<sup>10</sup>](#).

---

### 1.3.2 Polygons-based mapping

Digitized maps are normally represented within GIS software as polygonal layers. Of course polygonal layers are indeed needed to visualize the mapping, making it possible to assign different colors to different units. For this reason many mappers might start a new map by creating a new polygonal layer that she/he will then populate with the different polygons representing the units.

Although this approach seems to perfectly make sense (indeed the final product should be a polygonal layer) it inherently has several drawbacks:

- the mapper ends up tracing each contacts two times (one for each confining polygon)
- whenever it comes to editing the map the same edits must be exactly replicated two times
- the boundary of the polygons, which represent the contacts, should be stylable (e.g. to represent how much certain we are about a contact). This is not possible when a polygonal representation. The only solution would be to duplicate (again!) the contacts as a new line layer that now must be also edited whenever the map is updated.
- the mapper moves much of her/his attention in the polygons generation, rather than in the identification of the contacts. Furthermore, the mapper is forced to define the boundary of each polygon at once, which requires to define the contacts between that unit and (potentially) many others surrounding units.
- polygonal layers render impossible the application of any automated algorithm for map simplification (e.g. to smooth the contacts or even the node distances)

---

**Note:** Polygons-based mapping has a long tradition, and many GIS software offer dedicated editing tools that can alleviate many of these problems, but their correct use is demanded to the user, and not enforced by the data types.

---

---

<sup>10</sup> <https://pubs.usgs.gov/tm/2006/11A02/>

### 1.3.3 The Contacts and points method

The solution to these issues might appear rather counterintuitive and require to adopt a different perspective: although the polygonal-layer map is the final product that must be generated it might not be the best format to work on. Indeed, the best solution is to avoid completely the data duplication that is inherent in any polygonal representation.

To this aim we can treat each geological element for what they really are:

- Contacts are lines and not the (duplicated) boundary between two polygons
- These lines must intersect each other to define valid regions
- The polygons are just the infilling in between the contacts

Starting from these observations it is possible to establish a different methodology for creating a consistent map that does not suffer from all the issues detailed above:

1. Contacts are traced as lines. This also has the benefit to allowing the user to append additional attributes to the lines themselves (i.e. if a contact is more or less certain). To uniquely define a region in the plane the lines just need to intersect each-other. There is no need for precision snapping of the start and end points of the lines.
2. These contacts can be readily transformed into a polygonal layer by **polygonizing** them. All GISs provide adequate tools for this operation.
3. Attributes for the polygons can then be assigned either manually or using a point layer that labels each defined region. The latter approach allows for a complete and predictable reproducibility of the final product.
4. Any number of different segments can be used to define a single contact ( provided they intersect somehow), making it possible to assign different attributes to different portions of the same contact (i.e. if the contact is certain in some parts and uncertain in others). This is especially important for styling the contacts in the final layout.

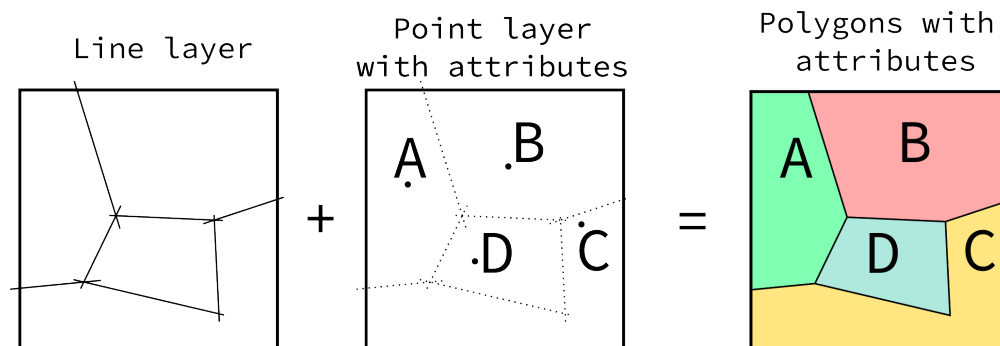


Fig. 1.9: Creating maps by using lines and points rather than polygons

This approach is also more similar to the geological reasoning that is performed when mapping, especially when remotely sensed imagery is used (as in planetary mapping): the operator tries to identify the boundaries between different terrains rather than directly defining the area covered by the units themselves. This approach has several advantages:

- No need to employ additional tools to grant the geometric consistency, which is automatically enforced by the data types. Indeed, errors (e.g. not intersecting lines) are readily visible because a polygon would be either missing or aggregated.
- Easy editing. Any time a contact is edited the changes will automatically propagate to all affected polygons.

- Reusable contacts. Contacts can be reused in map layout and styled to represent different types of contacts.
- The mapper can solve the contacts working on just two units at a time, rather than having to completely define the boundary of a polygon (which might be in contact with many units).
- The mapping can be performed at progressively finer level of details with no significant efforts: the mapper identify the most relevant units first and then proceed to refine the interpretation just by adding new intersecting lines.

---

**Note:** This approach leads to defining the contacts and label-points layer as the **sources** of a geological map, which can then be transformed to different types (the polygons) for display purposes. Such transformation is predictable and reproducible. This idea suggests that a good and reusable map should indeed also provide these layers to the public rather than just the polygonal ones. Making the map truly upgradable by third parties.

---

These ideas can be easily implemented in any GIS of choice by performing a polygonization of the contacts and a spatial join of the points to assign the attributes. If the contacts layer shall be used in the final layout imperfect intersections would leave dangling segments at the end of the lines, which require additional editing.

## 1.4 Toolbox algorithms

---

**Important:** The suggested way of using Mappy for your daily work is through the mappy dedicated configuration dialog and the mappy toolbar, as explained in the [Quick start tutorial](#) document.

---

Mappy provides several algorithms that can be accessed through the QGIS *Processing Toolbox*:

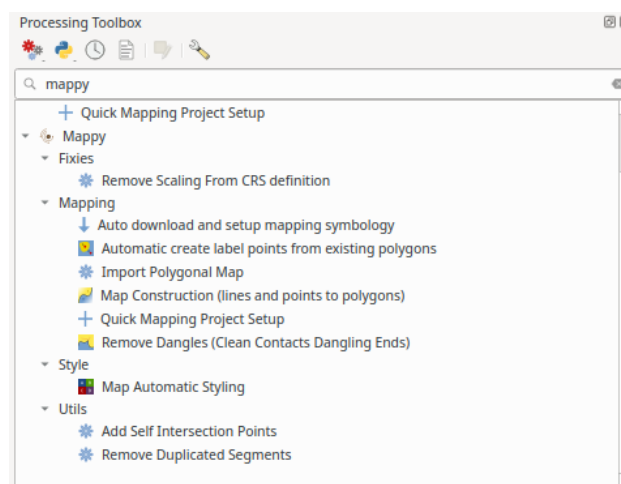


Fig. 1.10: The QGIS processing toolbox and the algorithms provided by mappy.

The algorithms are internally used by the mappy toolbar to regenerate the map while you

are working, but can be easily used to generate custom workflows by using the [QGIS model designer](#)<sup>11</sup>.

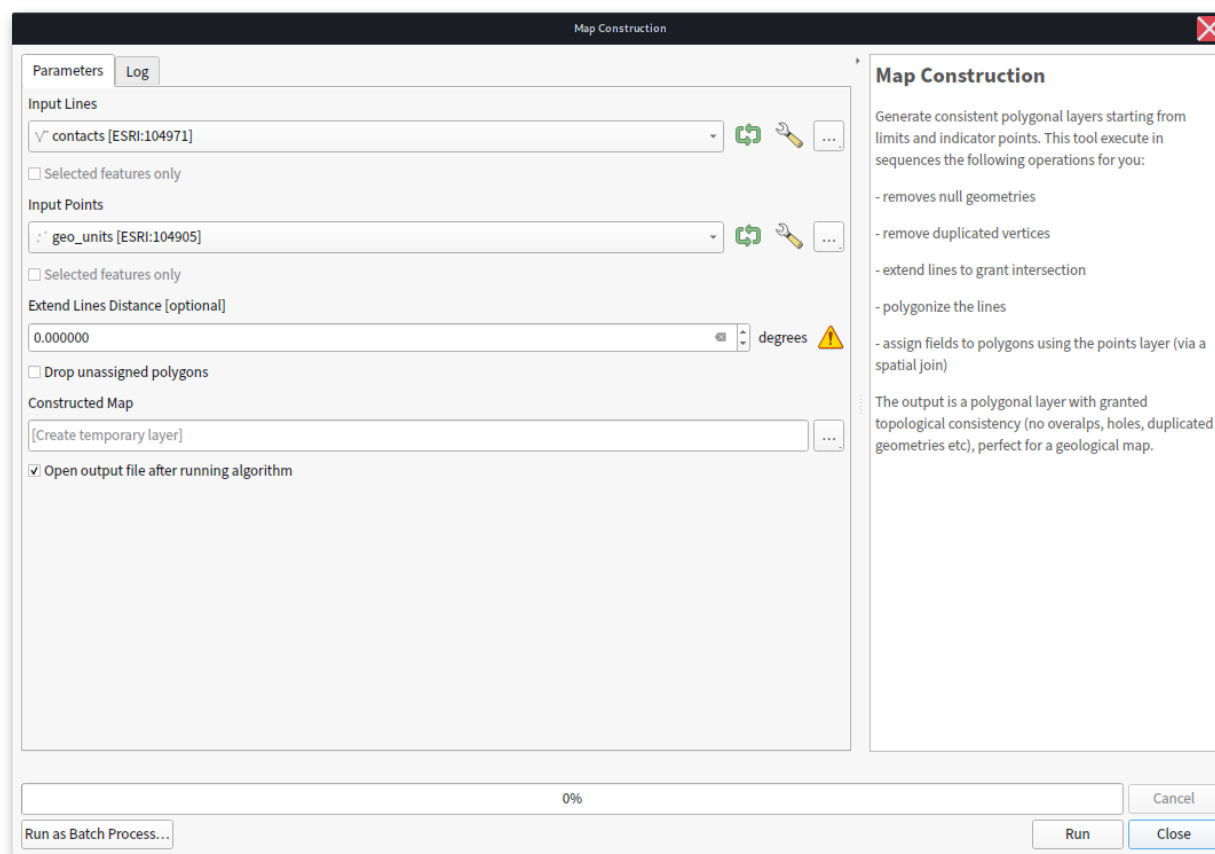
This page gives an introduction on how to use (some of) the algorithms provided by Mappy, which can be activated from the toolbox.

### 1.4.1 Quick Mapping Project Setup

This algorithm will generate the initial dataset (point layer and line layers) to jump-start your mapping project in a consistent and simple way.

### 1.4.2 Map generation

It will generate a new polygonal layer that can be used for printing the map. The mappy tools can be found within the *Processing Toolbox* dialog, under *Mappy*. We are especially interested into the *Map Construction* processing tool where we need to select the layers that will be used as contacts and the one that represent the points.



There are not many options here but at the present implementation we have:

- **Extend Line Distance.** Is the distance in map-units that will be used to extend each end segment of the lines. This might be useful if some of your lines do not intersect by

<sup>11</sup> [https://docs.qgis.org/testing/en/docs/user\\_manual/processing/modeler.html](https://docs.qgis.org/testing/en/docs/user_manual/processing/modeler.html)

a minimum distance. Of course is better to fix the problem by modifying the contacts layer itself.

- Drop unassigned polygons. If selected, mappy will not produce polygons that were not assigned with the point layer.
- Constructed Map. Is where the output will be placed.

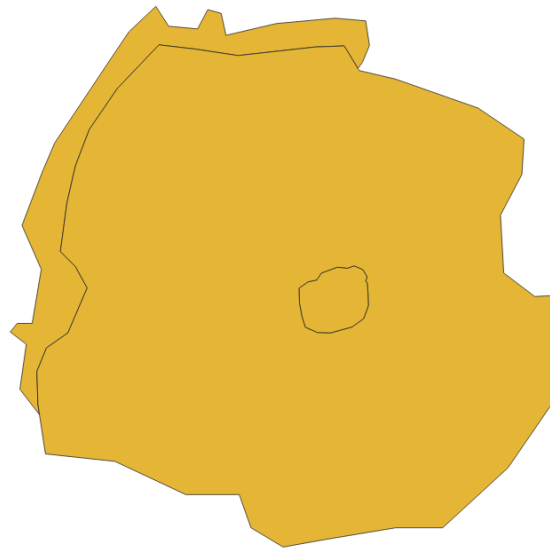
---

**Tip:** The default value for the *Constructed Map* option is to create a temporary layer every time the algorithm is run. During the mapping you might want to run the algorithm many times to have a peek at how your final map will look like. To avoid having to restyle everytime the newly generated map you could just set this option to overwrite a file on disk (e.g. a layer within your geopackage). In this way mappy can be run and QGIS will automatically find out that the layer was changed, and it will reload it (you can force the reloading by pressing *F5* if needed).

Alternatively you can use the tool *Map Automatic Styling*, provided by mappy to quick-style your newly created map.

---

After the map is generated it will look like this:

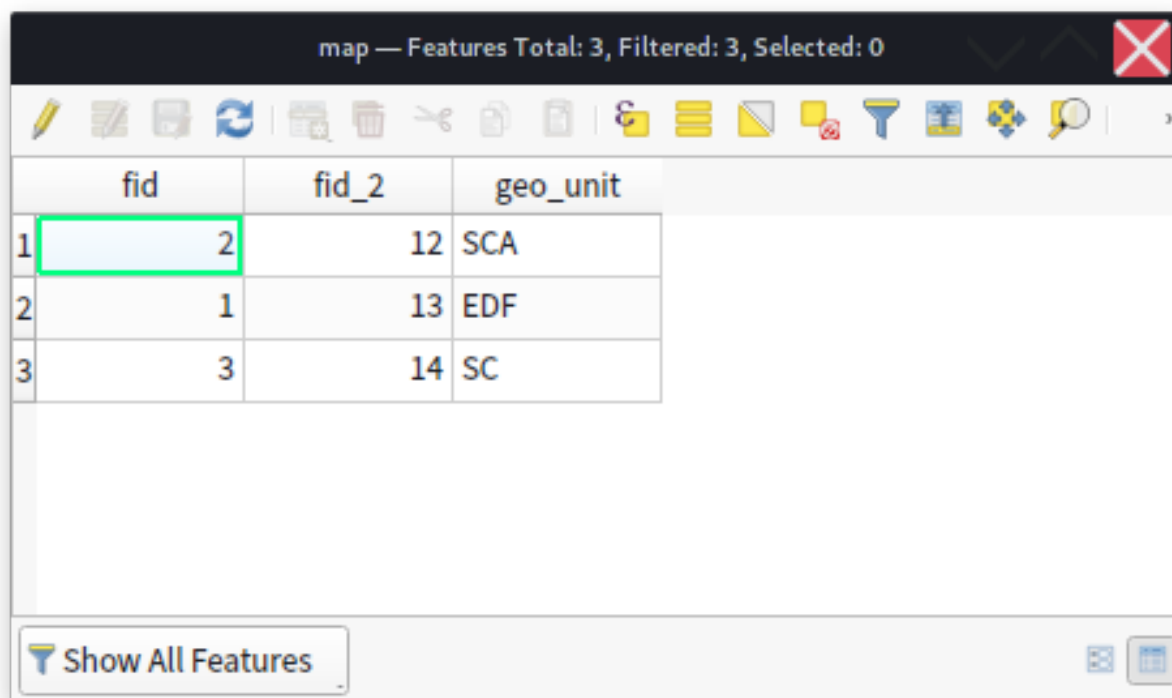


Mappy generated the polygonal layer for you and added the properties defined within the point layer. You can verify this by opening the attribute table for the polygonal layer:

### 1.4.3 Remove Dangles

Sometimes it might be useful to also style the contacts depending on their meaning or other qualities. For this task mappy provides the *Remove Dangles* tool which can remove any dangling line end or start to produce a new contacts layers, free of any imprecision that the operator used during mapping. The original attributes are retained and can be used to style the contacts:

In [Fig. 1.11](#) we cleaned the original contacts with *Remove Dangles* and used a different style depending on the type attribute of the contacts. This feature make it possible to re-use the contacts for layout purposes.



## 1.5 Citing Mappy

While a dedicated paper comes out you can cite this abstract if you found mappy useful:

Penasa, L.; Pozzobon, R.; Galluzzi, V.; Brandt, C. H.; Frigeri, A.; Lucchetti, A.; Toffoli, B. D.; Naß, A.; Rossi, A. P.; Massironi, M. Mappy: A Python Plugin to Ease Geological Mapping with QGIS; SGI Trieste, 2022.

or by using the [Zenodo](https://zenodo.org/doi/10.5281/zenodo.5233646)<sup>12</sup> doi [10.5281/zenodo.5233646](https://doi.org/10.5281/zenodo.5233646)<sup>13</sup>.

Mappy has been developed for the [PLANMAP-GMAP Winter School](#)<sup>14</sup>. If you found the school material and this tool useful you might consider adding an acknowledgment to your work as:

We acknowledge support from the EU's H2020 research and innovation program under grant agreement N° 871149 (GMAP).

Feel free to modify this statement to best suit your needs, but please remember to report the grant number N° 871149.

<sup>12</sup> <https://zenodo.org>

<sup>13</sup> <https://doi.org/10.5281/zenodo.5233646>

<sup>14</sup> <https://www.planetarymapping.eu/>



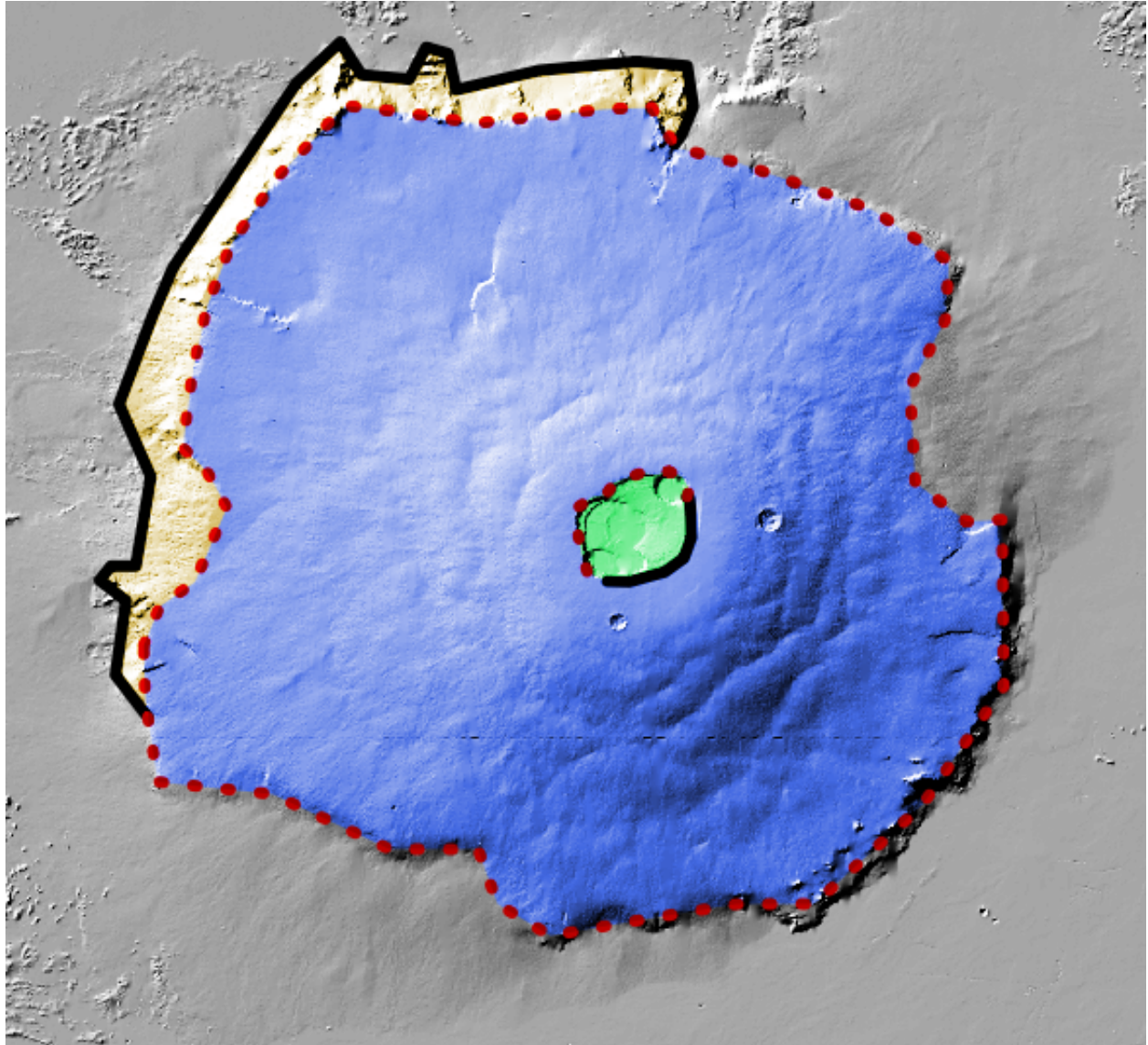


Fig. 1.11: Example of cleaned contacts that are then styled and used to finalize the map