
hiegeo

Jan 21, 2022

Contents:

| | | |
|----------|---------------------------------|-----------|
| 1 | Purpose | 3 |
| 2 | Installation | 9 |
| 2.1 | Requirements | 9 |
| 2.2 | Install <i>hiegeo</i> | 9 |
| 3 | Usage | 11 |
| 3.1 | Input files | 11 |
| 3.2 | Example scripts | 13 |
| 4 | Module documentation | 15 |
| 5 | Notation | 17 |
| 6 | Indices and tables | 19 |
| | Python Module Index | 21 |
| | Index | 23 |

Welcome to the *hiegeo*'s documentation, a Python module to demonstrate how to model geology taking into account for hierarchy and chronological order among the geological structures.

This is the accompanying software of the manuscript “A new perspective to model subsurface stratigraphy in alluvial hydrogeological basins, introducing geological hierarchy and relative chronology”, by C.Zuffetti, A.Comunian, R.Bersezion and P.Renard, Computers and Geosciences, DOI: [10.1016/j.cageo.2020.104506](https://doi.org/10.1016/j.cageo.2020.104506).

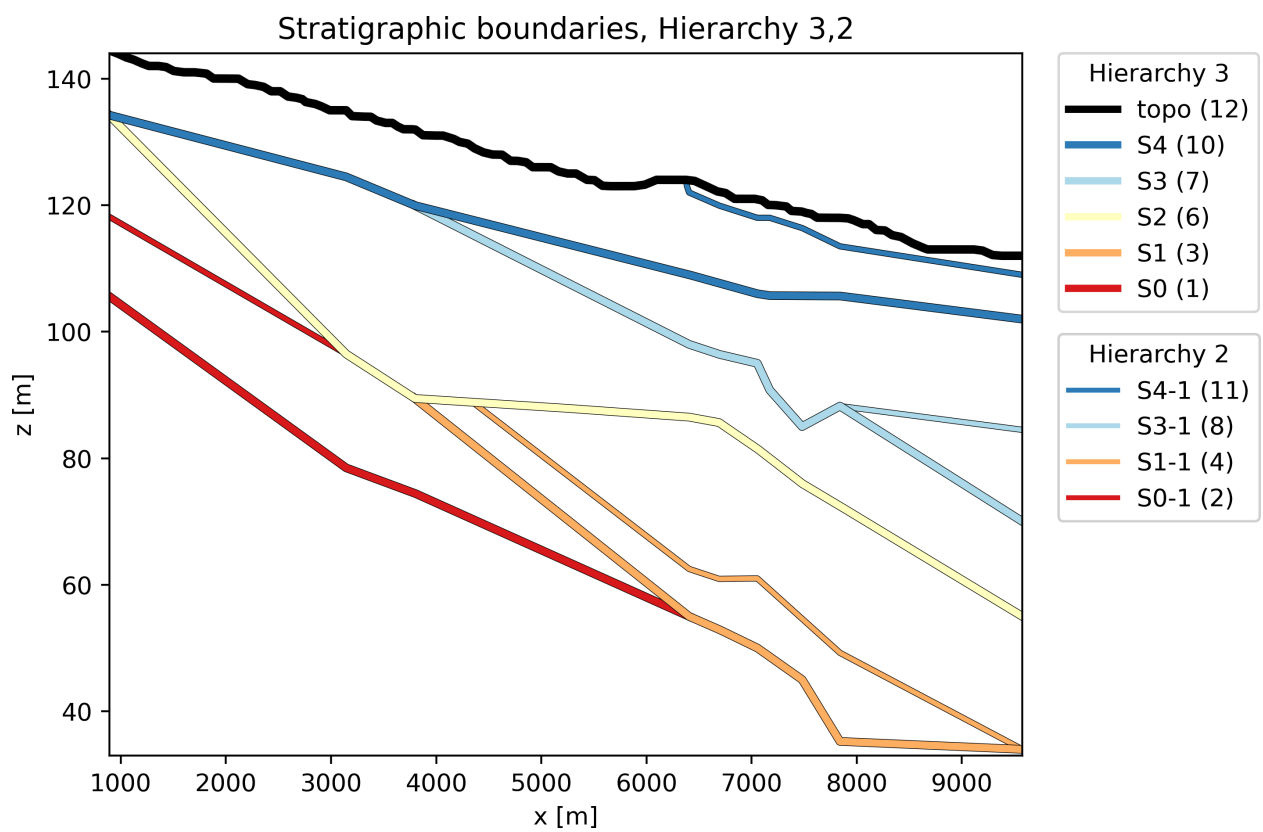
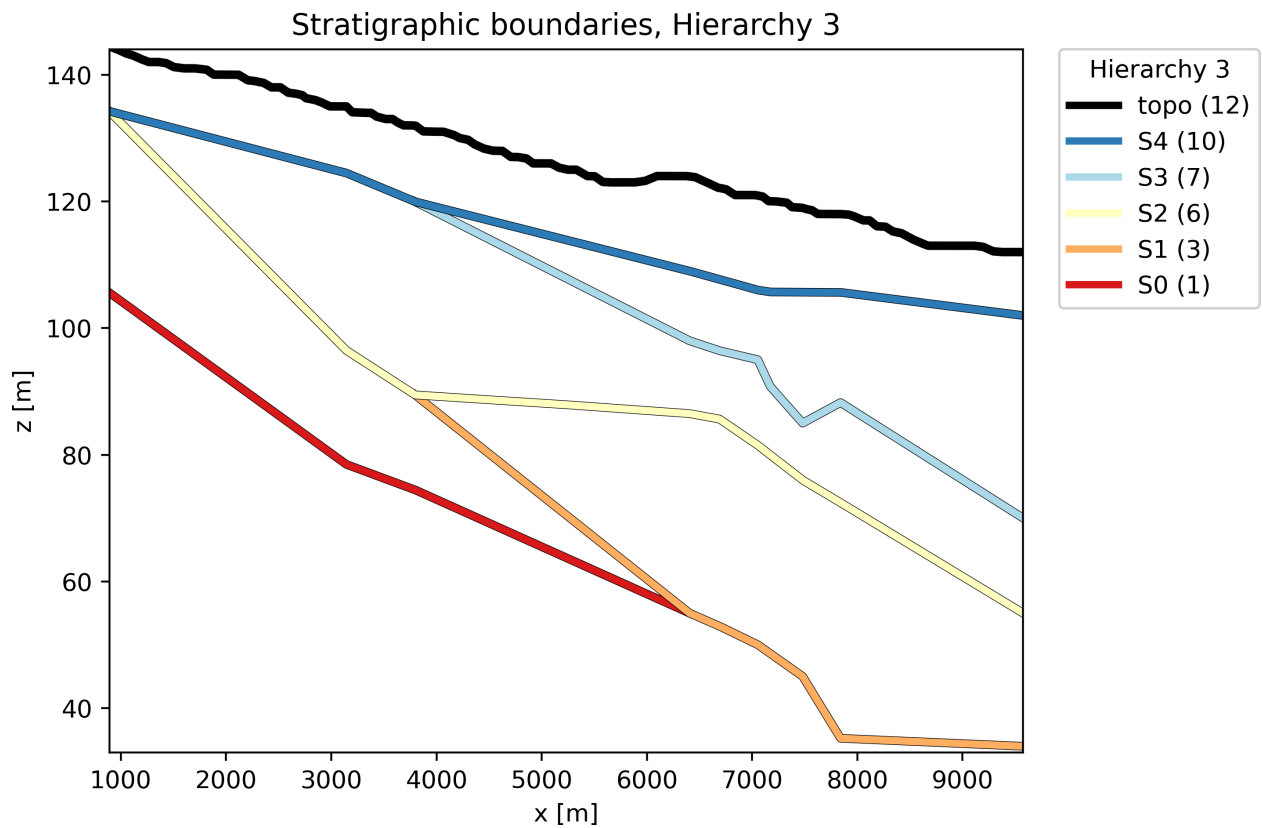
CHAPTER 1

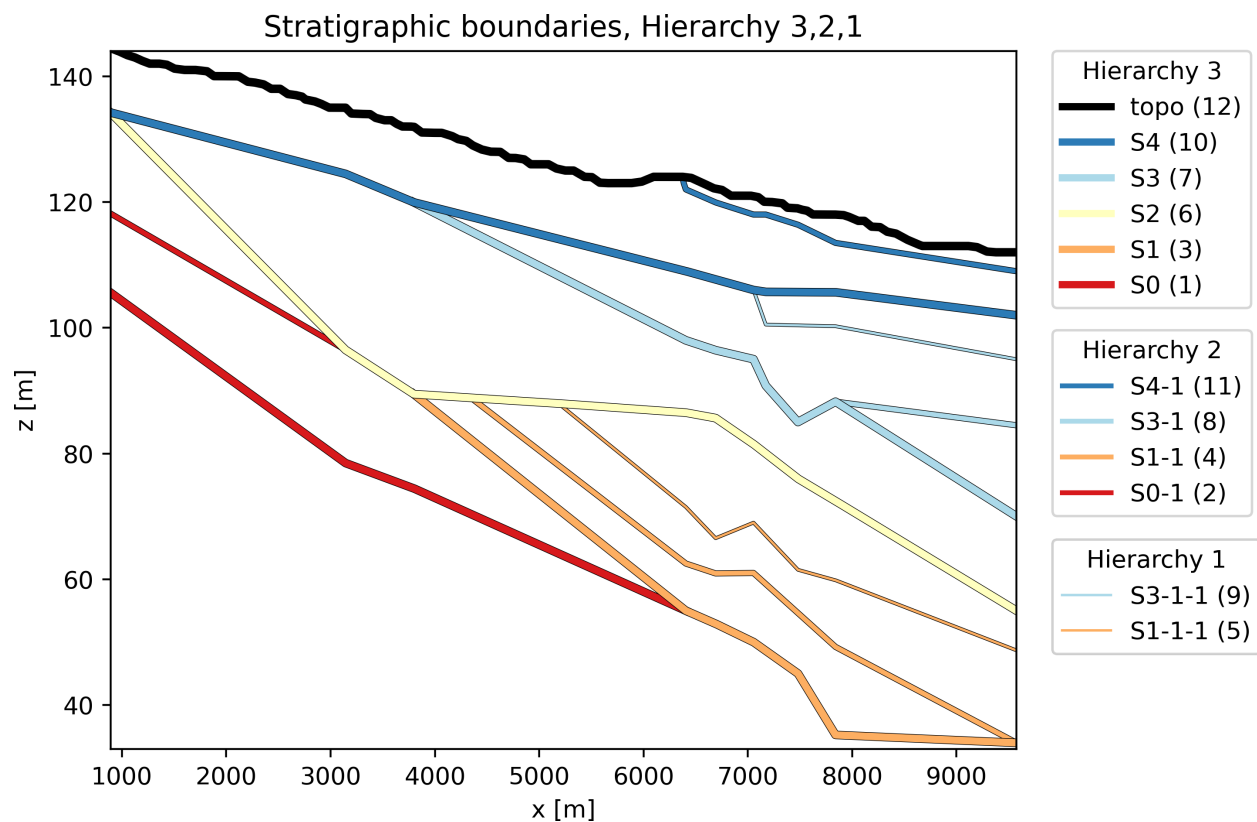
Purpose

The purpose of the *hiegeo* Python module is to illustrate the conceptualization of a novel approach to geological modeling, that takes into account for the hierarchy and the chronological order of the geological structures to be represented. It is not intended to replace any existing state of art geological modeling tool. Instead, it could be an useful tool to illustrate and demonstrate some clear modeling guidelines for hierarchical structures, which could be then implemented in more sophisticated tools.

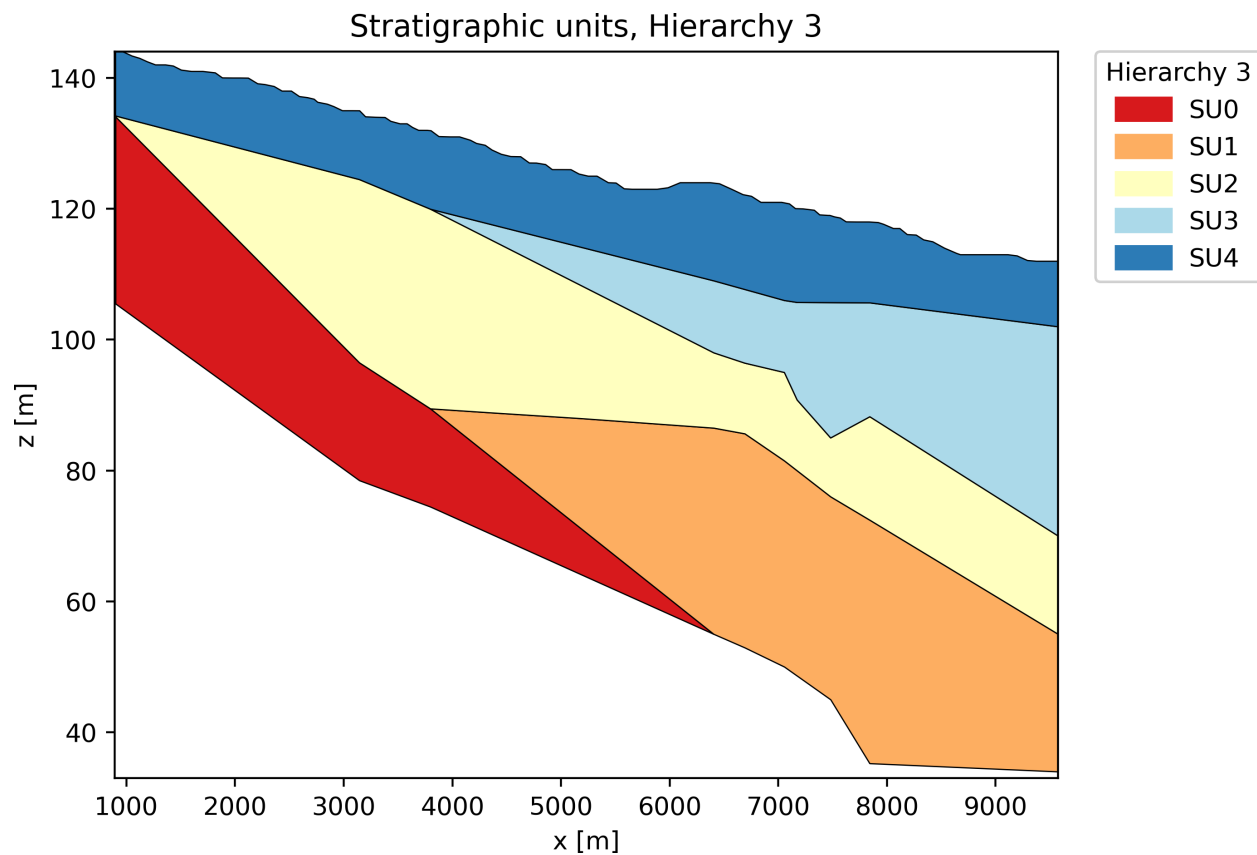
See the manuscript “A new perspective to model subsurface stratigraphy in alluvial hydrogeological basins, introducing geological hierarchy and relative chronology” (Computers and Geosciences, *accepted*) by Chiara Zuffetti, Alessandro Comunian, Riccardo Bersezio, and Philippe Renard for more details.

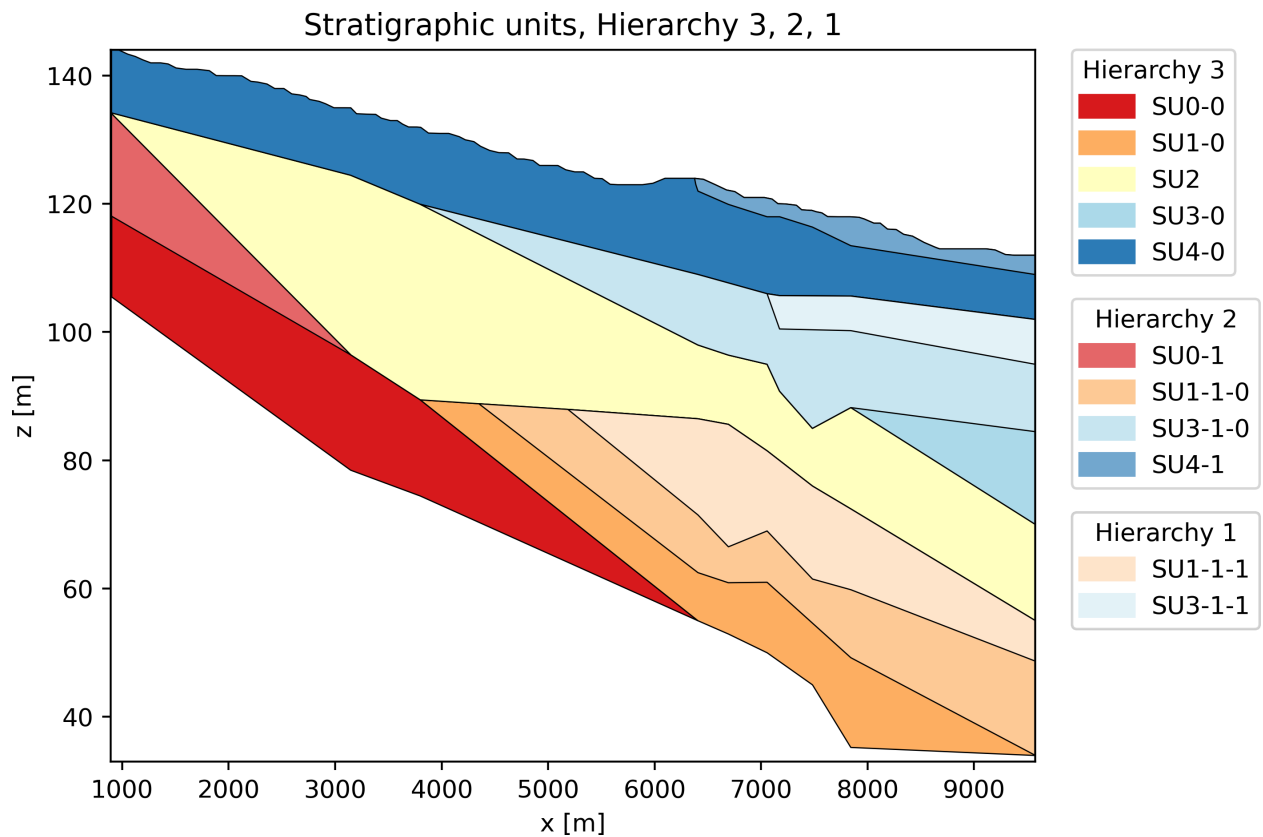
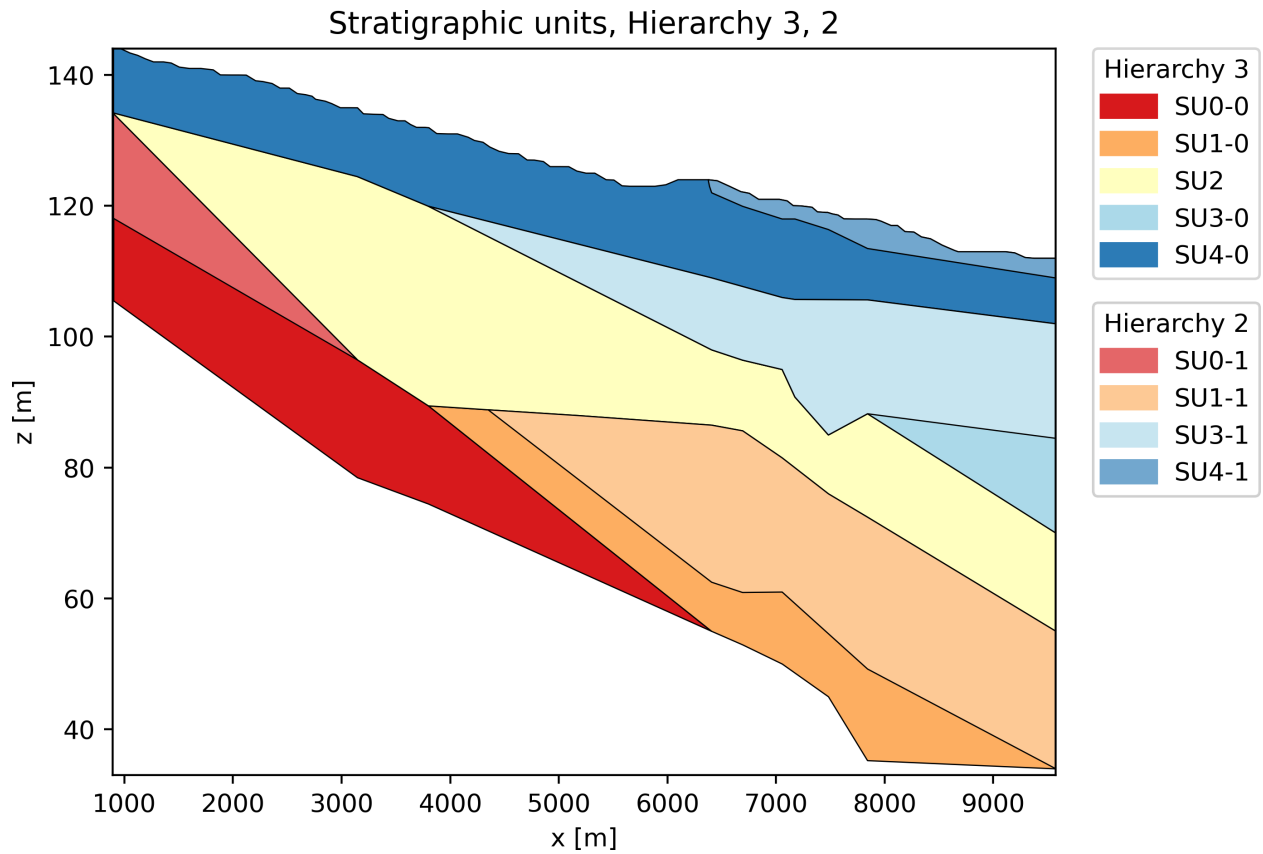
Hereinafter you can see one of the possible output provided by *hiegeo*, for example a representation of the geology in terms of stratigraphic boundaries (SBs) including only the 3rd level of hierarchy, the 3rd and the 2nd, and 3rd, 2nd and 1st:





The same representation at different hierarchical levels can be obtained in terms of Stratigraphic Units (SUs):





In addition, you can also have a representation of the geological hierarchy as a tree structure:

topo

S4

└ S4-1

S3

└ S3-1

└ S3-1-1

S2

S1

└ S1-1

└ S1-1-1

S0

└ S0-1

Installing and using *hiegeo* should be relatively easy, and thanks to Python's flexibility, that should be feasible on many operative systems (OSs), including MS Windows, Mac OS X and Linux.

2.1 Requirements

To use the *hiegeo* module you need a standard Python3.X installation, together with the main mathematical and plotting libraries (`numpy`, `pandas`, `matplotlib` etc) and the module `anytree`, which is required to handle the hierarchical structures. If `pip` is used for the installation, all there requirement should be automatically satisfied.

2.1.1 Python

Probably, unless you are working with Linux and you can use some package manager like *Synaptic*, the easy way to have Python up and running on your system is to use *Anaconda* (<https://www.anaconda.com/>). Therefore, download and install the Python (version 3.X) of Anaconda which is suitable for your OS by following the instruction provided on the Anaconda web site.

2.2 Install *hiegeo*

The suggested way is to use `pip` (which should be also already available with *Anaconda*).

hiegeo is available at the [Python Package Index repository](#). Therefore, in can be easily installed (together with its dependencies) with the command:

```
pip install hiegeo
```

Alternatively, if you prefer to download the sources from <https://bitbucket.org/alecomunian/hiegeo>, you can:

- 1) Clone or download this repository on your hard drive.
- 2) If required, unpack it and `cd hiegeo`.

- 3) Inside the project directory, from the command line:

```
pip install -e .
```

- 4) To check if it worked, open a Python terminal and try:

```
import hiegeo
```

This section describe an example usage of *hiegeo*.

3.1 Input files

For properly run *hiegeo*, you need two files:

- 1) A *.JSON* file, that contains info about the discretization grid and the rendering of the plots.
- 2) A data file (*.CSV* format) that include the actual data set, with information about the contact points, chronology and hierarchy.

3.1.1 JSON file

An example *JSON* file is provided in the file `examples/hiegeo_test.json`. It contains three main sections:

colors This section contains the definition (HEX format) of the color that should be given to each SB. The color should be provided for the highest hierarchy SBs only.

data_file The name of the *CSV* data file containing the data-set itself.

grid Info about the grid used to discretize the domain. The notation should be straightforward.

3.1.2 CSV file

This file contains all the required information to create and plot SBs and SUs.

Its header should look like:

```
,gis_id,chronology,hierarchy,sb_name,su_name,x,z  
0,1,10,3,S4,LCN,897.1953,134.202585
```

```

1,2,10,3,S4,LCN,3142.7824,124.47562
2,3,10,3,S4,LCN,3797.4044,119.94562
...

```

Apart from the very first column, that contains some index that are used by the Python library `pandas` (but here they could be set to whatever), the other columns content should be:

gis_id This is simply a legacy column that contains the ID of the points extracted from the GIS software. You can set whatever value here as these are not used by *hiegeo*.

chronology This is a very important column since it contains information about the chronology of the corresponding SB. The value should be an integer, that grows from the oldest to the youngest SB.

hierarchy Another very important column, contains the hierarchy (sometimes called “rank”) of the SB. High level of hierarchy means high importance in sedimentary terms. In the provided example we have integer values ranging from 1 to 3.

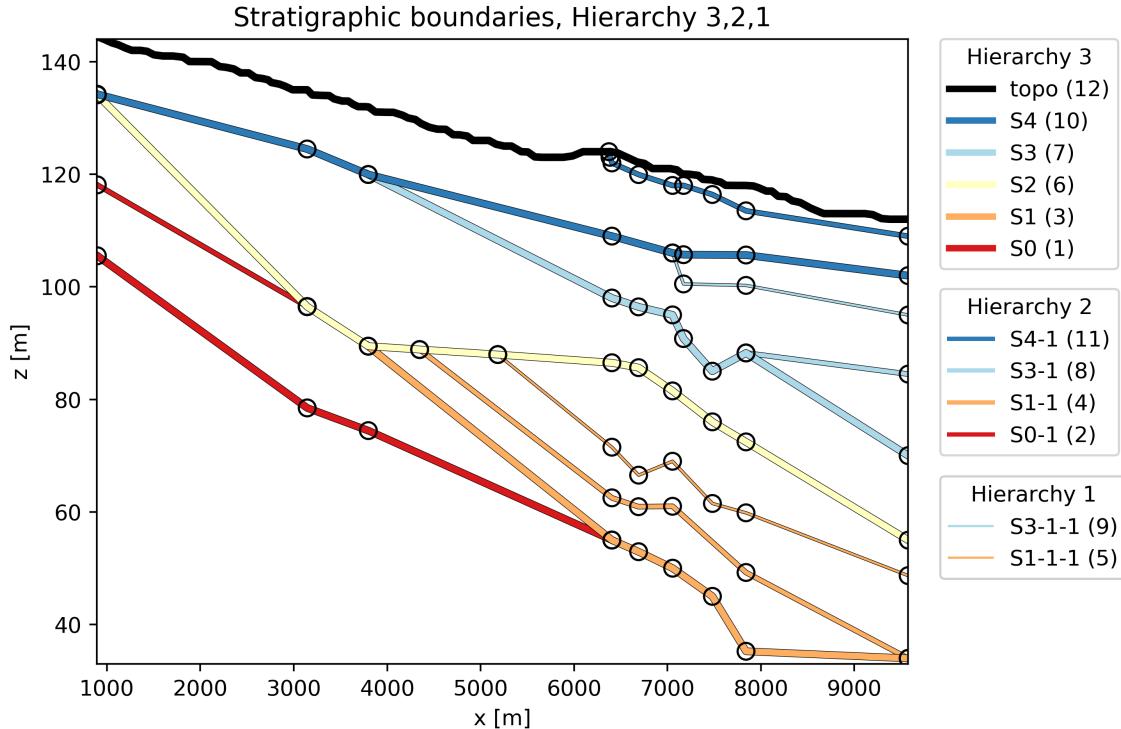
sb_name This is the name to be given to the SB. It should be a string

su_name This is the name of the SU. Actually, this column is not implemented and the SU is named according to and internally consistent nomenclature.

x The coordinate along the x axis of the (contact) points

y The coordinate along the y axis of the (contact) points

The following picture provides an illustration of the (subsurface) points contained in the data-set used for demonstrating the working principle of *hiegeo*.



3.2 Example scripts

Here two example scripts are provided in the *examples* folder. One file (*hiegeo_test-simple.py*) contains a full working example to read data, plot with a basic layout SBs and SUs, and provide a hierarchical representation of the geology with a tree structure. The other file, instead (*hiegeo_test-full.py*) provides a more complete example where plots are made for three different levels of hierarchical representation, with advanced plot legend and includes the creation of a GSLIB output file.

Both script should be sufficiently documented to allow running them without additional information.

You can move to the folder `examples` and there, from the command line, run one of the two provided demonstration scripts. For example:

```
./hiegeo_test-simple.py
```

and hit <Enter>. On Linux you will probably need to give execution rights to the file, like this:

```
chmod +x hiegeo_test-simple.py
```


CHAPTER 4

Module documentation

CHAPTER 5

Notation

Hereinafter some acronyms that will be used throughout the document:

SB Stratigraphic boundaries.

SU Stratigraphic units.

CHAPTER 6

Indices and tables

- `genindex`

h

hiegeo, [15](#)

H

hiegeo (*module*), [15](#)