# Cloud segmentation cleaning + normals calculation

Method developed by Tanguy Racine, for the ERC Kast project. This tutorial segments the point cloud in chunks, clean the points from the operators or disconnected points, and calculates normals.

(Tutoriel written by Celia Trunz, 8 Septembre 2023, based on Tanguy Racine explanations.

If its the first time: you have to create a conda environment named **Cloudcompy310** with

* Cloudcompy310,
* networkx,
* Karstnet (from GitHub),
* mplsteronet,
* pygraphviz. (todo: create a ymal file with all libraries to install)

You can create the environment with the yml file created on the CHYN-2301-W PC (navigate to the D: folder with the anaconda prompt, and type **conda env create -f environment.yml** (the name is already saved in the yml file))

## Extract the sections:

* 1. In data D: (on the **CHYN-2301-W** computer) double-click **StartUp.bat** to start the couldcompare cloudcompy310 environment and start Jupyter.
  2. In Jupyter, select the notebook: **09\_SegmentCloudFromShapefile.ipynb** (under D:/CloudComPy310\_20230705/CloudComPy310/)
  3. Change the name of the cave, and extract the sections by launching the code

## Clean points file

### Look at the dense point cloud in cloudcompare

* + 1. Go to DATA **D:/ScanLeica/ --Cave Name-- /process**
    2. Double click on Cropped\_ number...
    3. Yes to all, yes to all

For better visualization click on  and Panel Properties/Colors/None

### Clean up noise caused by the operator

* + 1. Use plugin**/CANUPO/classify.** The classyfier must be trained first (Made by Tanguy on the cave Cascade de Motiers)
       1. Import the classifier. **D:/ScanLeica/ people\_bedrock\_classifier\_10082023\_strict\_on\_bedrock**
       2. Select "subsample cloud = 0.05"
    2. Separate the point cloud into two classes. **Edit/Scalar fields/Split cloud (integer values**) (Be careful to select the general point cloud and not the subsample)
    3. Select "class #2" in the DB Tree for the next step.

### Selecting Related Components

* + 1. Click on  cccc (select related components)
       1. Octree level 0.08 (8cm)
       2. Min. points per component = 10
       3. --> If operator noise:

### Get the normals:

* + 1. Select cc#0 in the left panel "DB Tree"
    2. **Edit/normals/compute**
       1. Local surface model = Plane,
       2. Octree clic on auto (often between 1 and 2 cm)
       3. Set octree radius to 0.08
       4. Orientation Use minimum Spanning Tree = knn = 12 (so that it uses more of the neighborhood)
    3. Result gives the normals.
       1. If normal are reversed (if light gray is inside): (todo-- find why it is reversed) **Edits/normals/invert**
       2. If only some part of the normal are inversed, cut in smaller chunks, then increase the radius (double, quadruple) and try to increase or decrease the Spanning tree. If the smaller chunk is all suppose to be oriented in the same direction, then reduce the spanning tree to 6.
    4. Save to **scanleica/cave.../process**. Add **\_normals\_reoriented** and save as  **.ply** binary

## Mesh reconstruction

1. Go to plugins/PoissonRecon
2. Octree depth (precision) or
3. Resolution: if you want a strong precision: put a high figure. Above 11 you don't see much difference.
4. Check density

Cloud.crop -- orthoDim 2 (looks from above) inside = True

Cloud - all points

Cc\_polylines - lines for cutting