Cloudbuster

Requirements

Python3

Download under: https://www.python.org/downloads/release/python-380 and follow installation instructions. Do NOT use the windows warehouse installer! This installation lacks important modules.

On Windows 10 It might be necessary to install:

microsoft visual c++ 2015 redistributable
microsoft visual c++ 2013 redistributable
microsoft visual c++ 2012 redistributable

When using pip for the installation of all Python modules/packages **pip3** should be used for Linux.

When using pip Windows requires the installation with:

```
python -m pip install -U <package name> or
py -m pip install -U <package name>
```

skimage library:

Follow the installation instructions under: https://scikit-image.org/docs/dev/install.html

```
pip3 install scikit-image
```

e.g. for command line installation in Linux.

Open3D library:

Follow the installation instructions under https://pypi.org/project/open3d-python/

```
pip3 install open3d-python
```

With Ubuntu 22.04, Python 3.10 installation should be done via:

```
sudo apt-get install python3-open3d
```

numpy:

Follow the installation instructions under https://numpy.org/install/

```
pip3 install numpy
```

Matplotlib:

Follow the installation instructions under https://matplotlib.org/stable/users/installing.html

```
pip3 install -U matplotlib
```

PCA:

Follow the installation instructions under https://pypi.org/project/pca/

```
pip3 install pca
```

Pandas:
Follow the installation instructions under https://pypi.org/project/pandas/

```
pip3 install pandas
```

Scikit-learn:
Follow the installation instructions under https://scikit-learn.org/stable/install.html

```
pip3 install scikit-learn
```

wxpython:

Follow the installation instructions under https://wxpython.org/pages/downloads/index.html

```
pip3 install wxPython
```

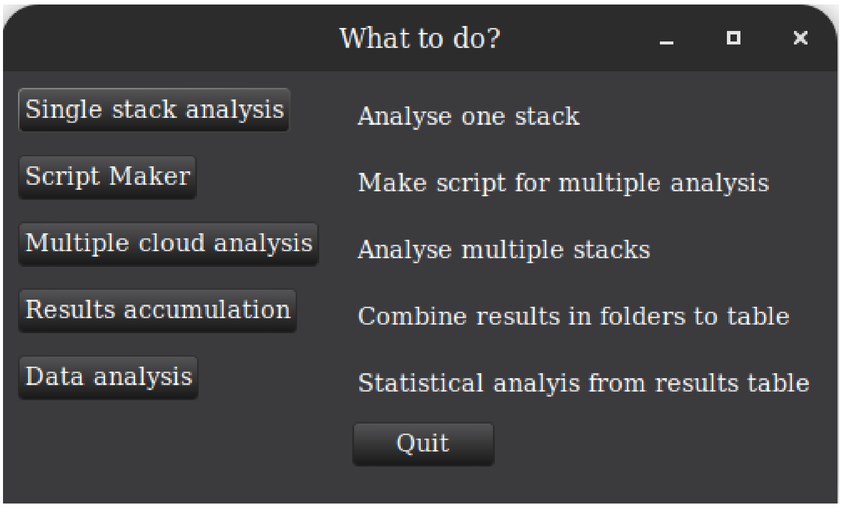
wxpython additionally might require the installation of the gtk3+ library: https://www.gtk.org/docs/installations/ .

Running the script:

Safe the script in your local folder, in a terminal window enter the folder and type: `python3 Auswahl.py` to start the selection script. Depending on your python installation under windows also the command `py Auswahl.py` might be correct. It might be necessary to add the full path to `Auswahl.py` on Mac.

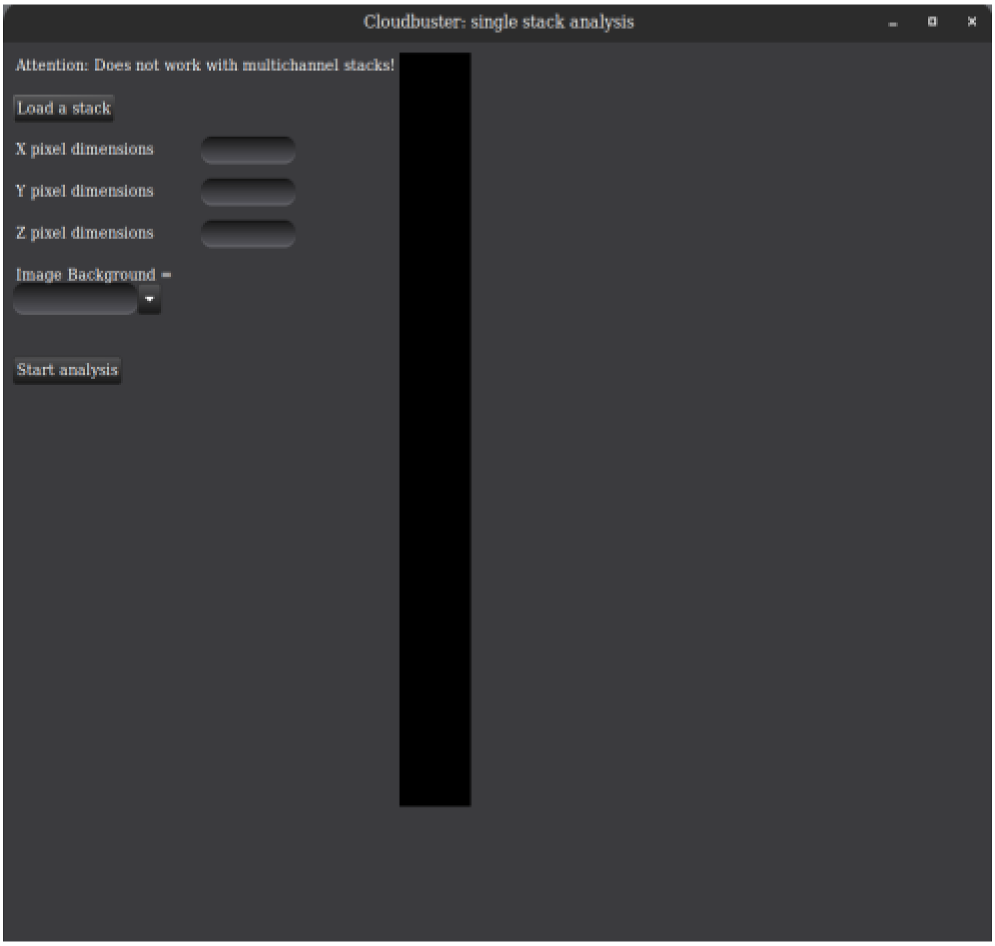
How to use:

Five scripts are bundled with a single selection script : Auswahl.py.

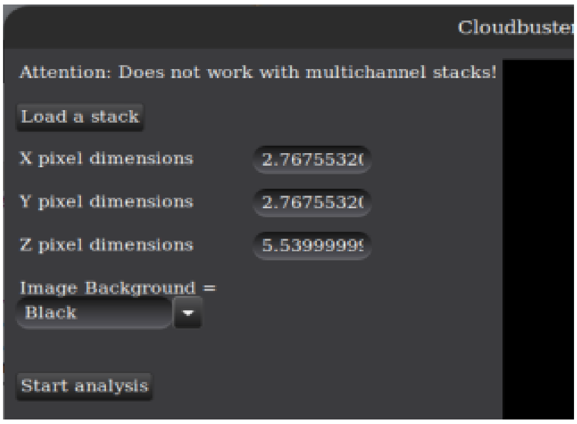


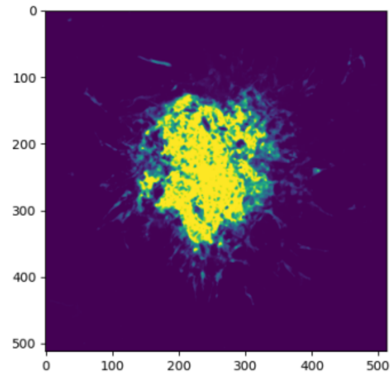
This script allows to select between the single spheroid analysis script: cloudbuster.py, the bulk analysis script: masscloud.py, the results table collecting script: accumulate.py and the data analysis script: statistics.py

Cloudbuster.py (Single cloud analysis):

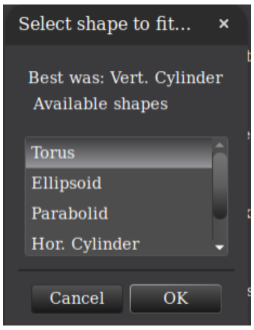


The script assumes grey scale tif image stacks. It does not work on multichannel stacks. After calling the script a user interface will appear (appearance depends on os), where an image stack can be loaded using the „Load a stack“ button. If metadata exist for the image stack, pixel resolution for X, Y, Z, dimensions will show in th X (Y,Z) pixel dimensions. At the same time a maximum projection of the stack will be displayed in a separate window. The dimensions values can be changed manually, if necessary or no metadata could be detected.





A selector field requires the information for the background of the image: black or white. Upon pressing the „Start analysis“- button the script starts to analyse and transform the data toward the final point cloud, as indicated by the symbols in the centre of the window. „Cloudbuster“ tries to find an optimal fit out of 5 geometric bodies. When this point has been reached during the analysis process, the user will be asked for his choice. Above the choice list, „Cloudbuster“ will propose a fitting shape. If cancel is pressed the analyses continues with the proposed shape.



The results and a number of intermediate steps are saved without normals in new created sub-folders (3D_files, results) to the original image stacks for potential later usage. The 3D_files folder contains:

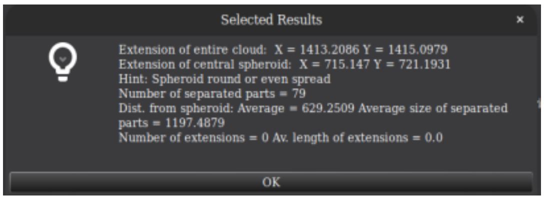
..._color.ply = a colorized point cloud of all identified elements (cells, particles, spheroids).

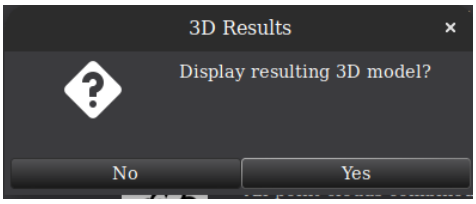
..._fit.ply = the fitted ellipsoid for the largest spheroid.

..._extensions.ply = the isolated extensions from the largest spheroid.

..._largest.ply = point cloud of the largest spheroid.

Part of the quantification results are shown in in a new message box, when the image stack has been processed completely. Along with the results information a further request box opens asking if a 3D display of the resulting point cloud is wanted:





All quantification data are stored in the results folder in CSV files that includes the original filename to which identifiers have been added:

..._final_results.csv = results file with spheroid size, separeted parts (cells) quantification and extensions quantification.

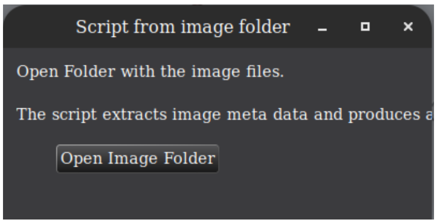
..._ind_parts_results.csv = list of the separated cells and their quantification.

..._ind_exten_results.csv = list of the identified extensions from the central spheroid and their quantification.

One addtional „.ply“ file without normals in the same folder and with the same name like the image stack represents a raw point cloud adaptation of the original image stack. The „.ply“ files can be loaded into several cross platform 3D software, including Meshlab and Blender.

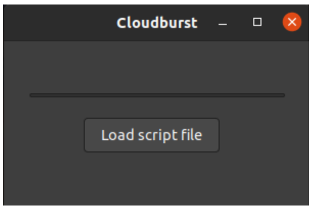
Script Maker:

This part is intended to ease the generation of a script file to be use by the „Multiple cloud analysis“ option (see below).



Upon pressing the „Open Image Folder“ a folder selector pops up and the folder containing all .tif stacks to be analysed can be selected. Pressing OK results in a selection box with 5 geometric body, the data should be fitted to. When this is done a script file (script.txt) will be saved in the same folder, containing all necessary meta data required for the next option:

Multiple cloud analysis:



This rather unspectacular box is all what requires some input by the user.

Intended to analyse several spheroid tif stacks in a sequence. Upon pressing the „Load script file“ – button a file dialog opens. It requires a script file (plain text format) that includes the name of the tif file, the X,Y,Z pixel (voxel) resolution in µm (or equivalent), and a value for the backgound color : 0.0 = black background, 1.0 = white background. Values must be separated by a comma. Each line must be finished with a return, resulting in 1 line more than actual entries.

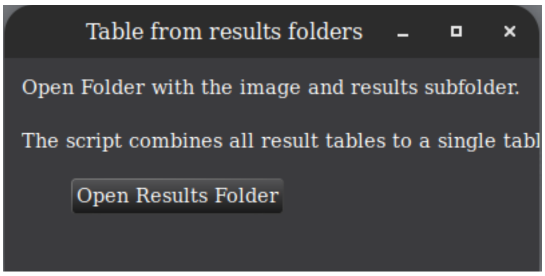
Example:

```
A_Stack.tif,2.0757,2.0757,4.1600,0.0↲
B_Stack.tif,2.7676,2.7676,5.5400,0.0↲
C_Stack.tif,2.7676,2.7676,5.5400,0.0↲
↲
```

All those data can be added manually, but it is easier to use the „Script Maker“ option (see above), which accumulates all the required data above on its own.

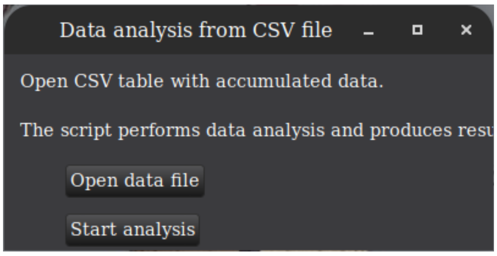
For each image a folder will be created in which the results and intermediate results are stored. Results will be stored in subfolders, equivalent to the single spheroid analysis. When all files are analysed a message box will open saying: „Process finished“.

Accumulate.py (Results accumulation):



Intended to collect all results files in a folder, including subfolders. Upon pressing the „File Folder“ button a file selection dialog will open. Starting with the selected folder and continuing in all sub-folders, all csv files containing „Final_Results“ will be collected to single csv files: „acc_results.csv“. These file contains in addtion to the results data the file names in the first column. When the process is finished a message box will open showing the file names.

Data analysis:



This script is intended to be used after a larger set of data has been collected from several measurements and results accumulation has been used to bundle the results. After pressing the „Open data file“ button a file selection dialog will open and you will need to open the „acc_results.csv“ file here. Upon pressing the „Start analysis“ button the data analysis will start. The analysis comprises of four major machine learning data analysis components:

- PCA = identification of the 5 most relevant parameters from all performed measurements.
- Correlation = correlation analysis of the 5 most relevant parameters.
- Hierarchical Clustering = Sorting and clustering of the samples according to variances.
- SBScan Clustering = Cluster analysis of the data using SBScan, including identification of the ideal number of clusters.

For each analysis a graph will pop up and the results are stored in the „acc_results.csv“ folder. The results consist of:

„PCA_top.csv“ = Top 5 relevant parameters of all measurements (dimensions contributing most to the data variance).

„PCA_top_raws.csv“ = Raw data of the PCA.

„Correlation.csv“ = Correlation table of the „Top 5“ parameters.

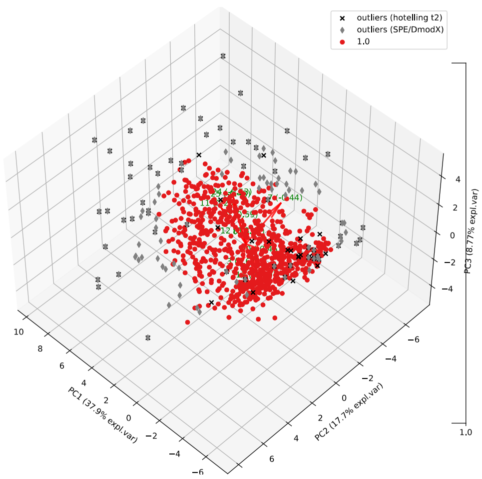
„Hier_Clu.csv“ = Sample names and associated ranking according to hierarchical clustering.

„SBScan_clu.csv“ = Sample names and associated cluster numbers according to k-means clustering.

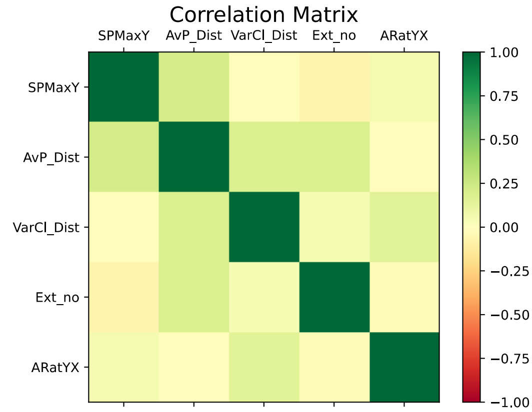
„SBScan_raw.csv“ = Raw data from k-means cluster analysis.

Addtional to the tables above graphs for the individual analysis steps are stored:

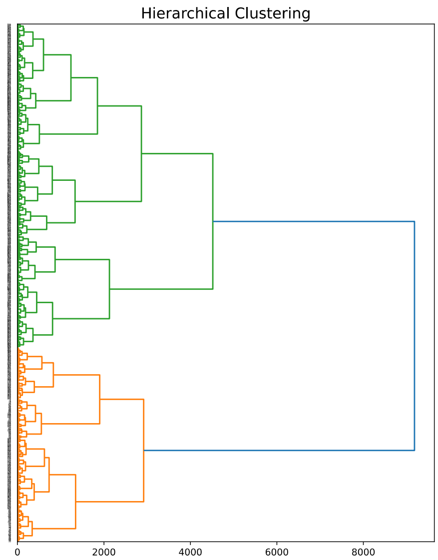
PCA:



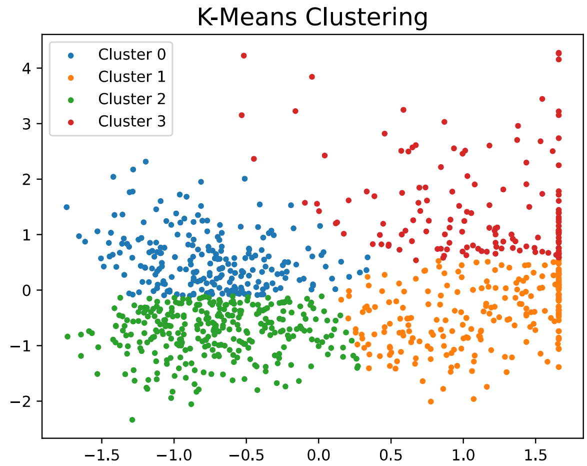
Correlation Analysis:



Hierarchical Clustering:



SB-Scan Clustering:



For the abbreviations in result files see Parameter_List.pdf

Additional results files will be generated for individual extensions and particles for each spheroid file (..._indExt_results.csv, ..._ind_parts_results.csv).

Result files are stored in the subfolder „results“.

Generated 3D point clouds are stored in the subfolder „3D_files“:

- ➔ filename.ply = Basic point cloud.
- ➔ filename_color.ply = color coded identified parts.
- ➔ filename_fit.ply = fitted geometric point cloud.
- ➔ filename_extensions.ply = identified extensions point cloud only.
- ➔ filename_largest.ply = largest (central) spheroid point cloud only.