

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
```

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
```

Kneed:

Follow the installation instructions under <https://pypi.org/project/kneed/>

```
pip3 install kneed
```

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/> .

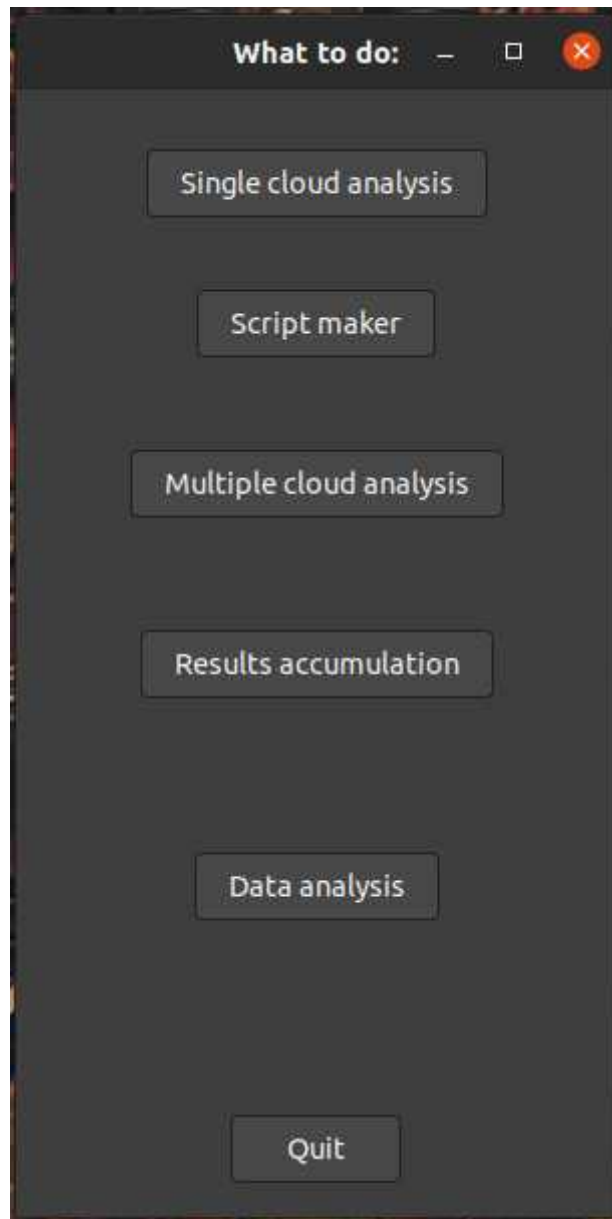
Running the script:

Save 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.

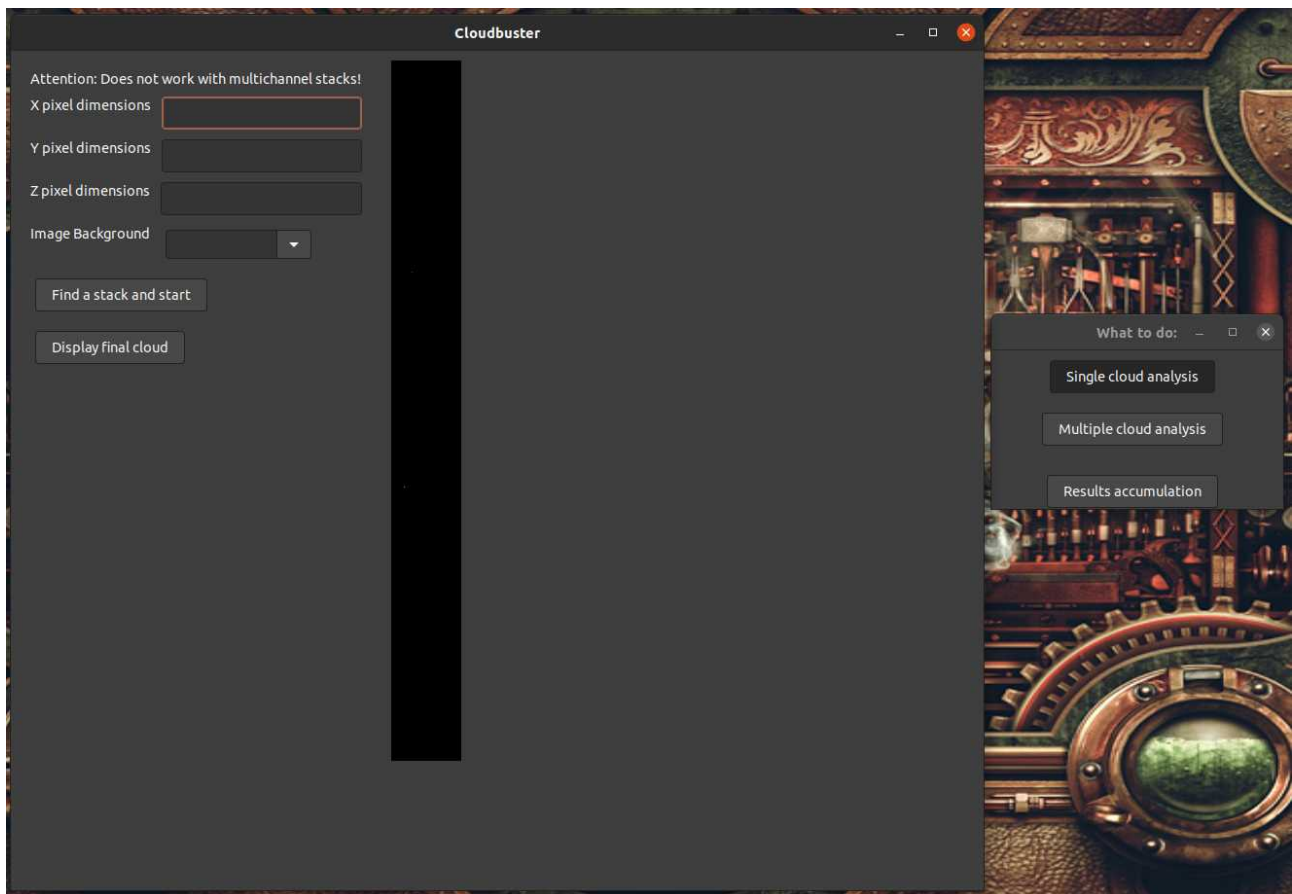
How to use:

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), in which the pixel resolution for X, Y, Z, dimensions have to be added. A selector field requires the information for the background of the image: black or white. Upon pressing the „Find a stack and start“- button, a file selector window will open and the regarding tif stack can be selected. After selection and cofirmation 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. 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).
- ..._ellipsoid.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. All quantification data are stored in the results folder in CSV files that includes the original filename to which identifiers have been added:

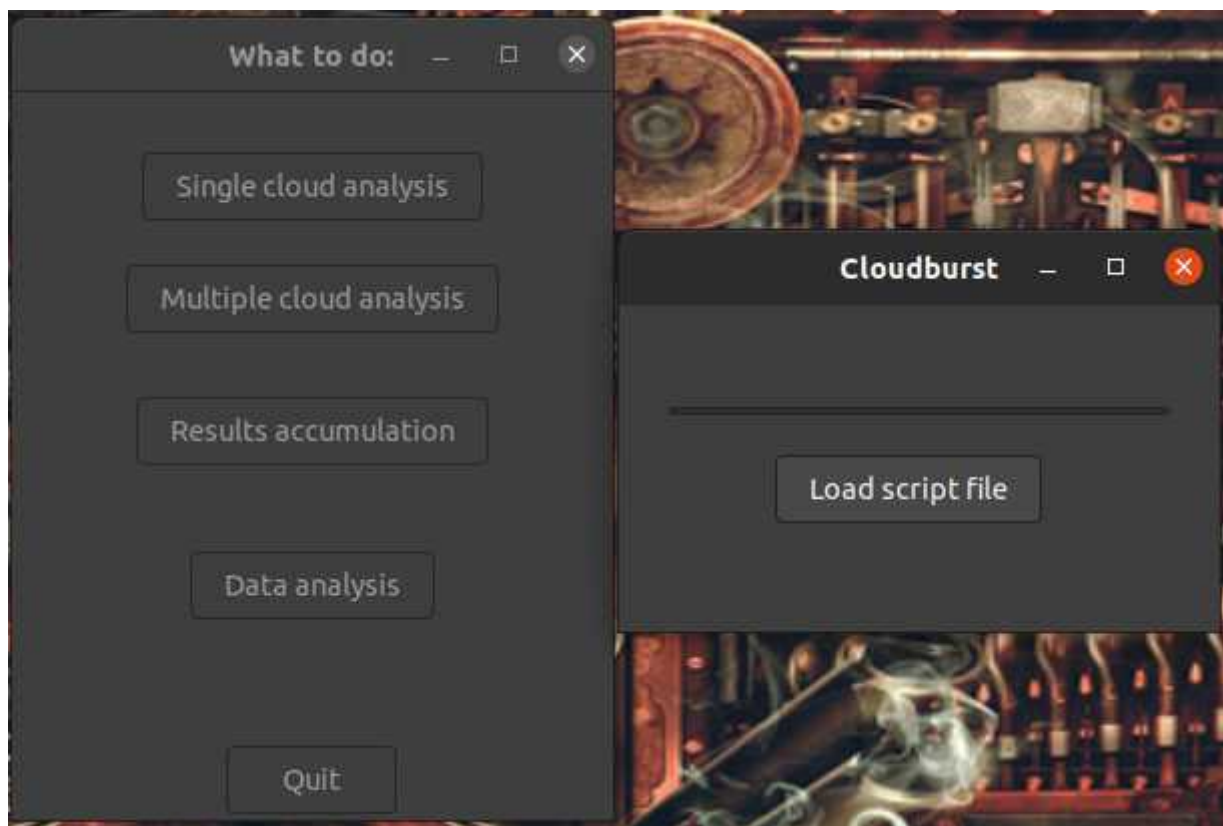
..._Final_Results.csv = results file with spheroid size, separated parts (cells) quantification and extensions quantification.

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

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

One additional „.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.

Masscloud.py (Multiple cloud analysis):



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 background 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.

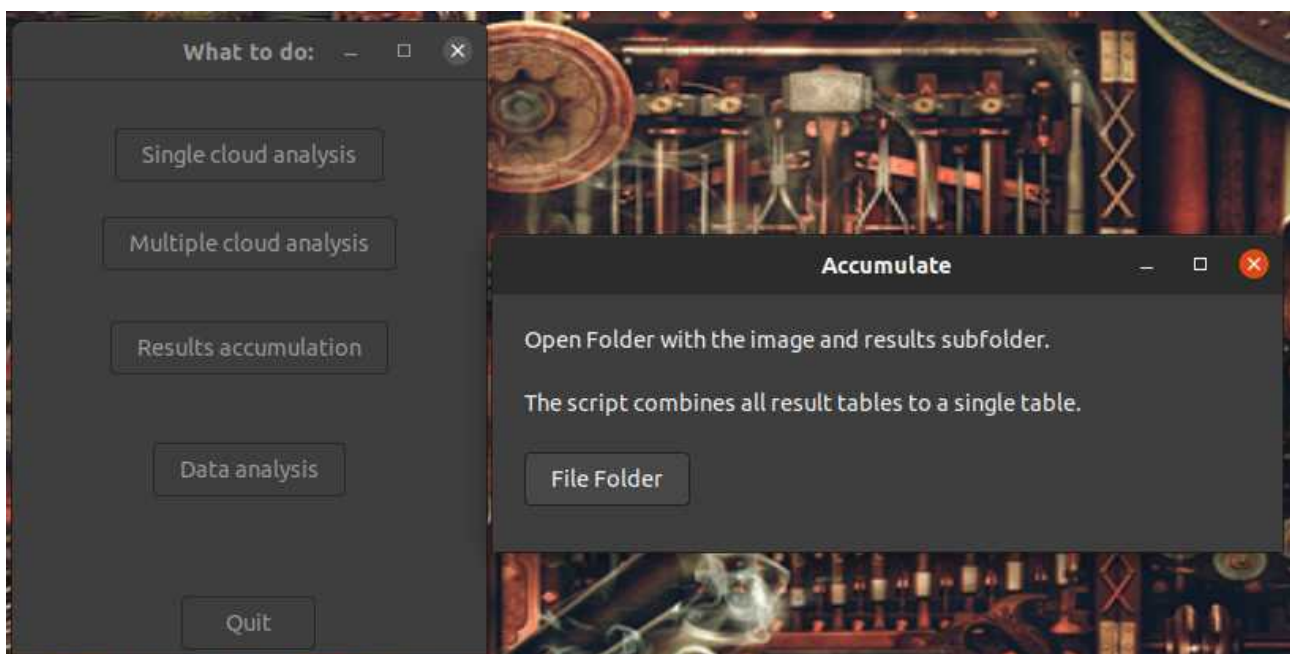
Example:

```
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↓  
↓
```

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“.

To make the generation of the script file easier an additional small script is included in the Cloudbuster folder: scriptmaker.py (Script maker in the menu). It can be used when the image stacks contain a size calibration provided by the aquisition software or added later. Scriptmaker.py only requires text file containing the list of file names to be used. Such a list can be easily created by using the command: `ls > contents.txt` on windows powershell or Linux bash in the regarding folder. Scriptmaker does not provide a GUI. You will have to change the folder names for the contents and skript file in the scriptmaker.py. Also, it will be necessary to install the module `tifffile = pip3 install tifffile`. The script will extract any calibration information from the .tif files and will save a script file readable for MassCloud.

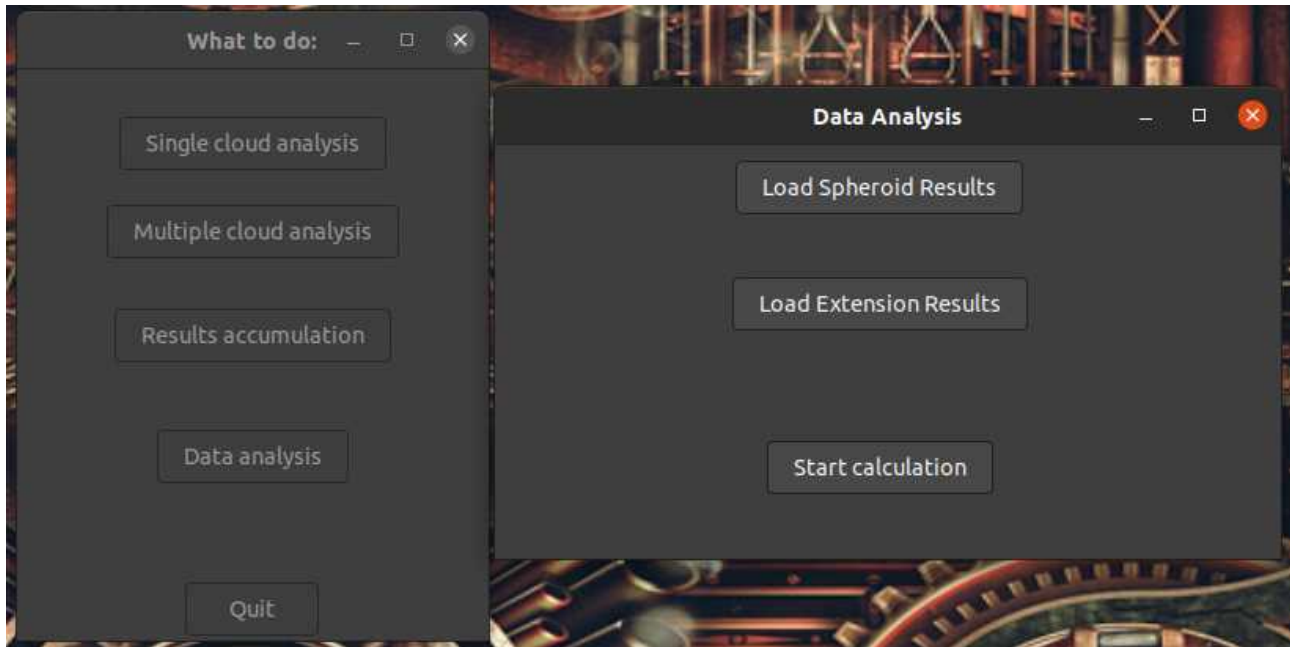
Accumulate.py (Results accumulation):



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“ and „acc_extensions.csv“). These files contain in addition 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.

Statistics.py (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 „Load Spheroid Results“ button a file selection dialog will open and you will need to open the „acc_results.csv“ file here. Next you will need to open the corresponding „acc_extensions.csv“ file. Upon pressing the „Start calculation“ 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.
- K-Means Clustering = Cluster analysis of the data using K-Means, 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.

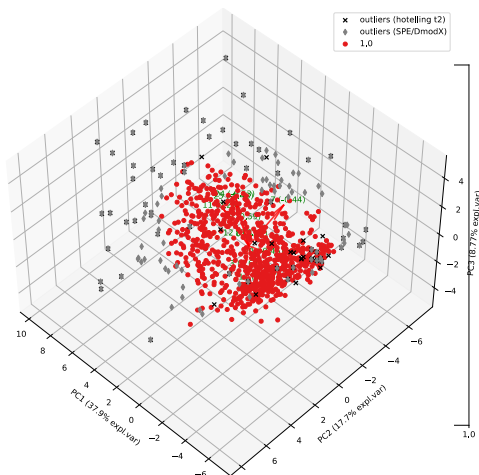
„Hier_Clu.csv“ = Sample names and associated ranking according to hierarchical clustering.

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

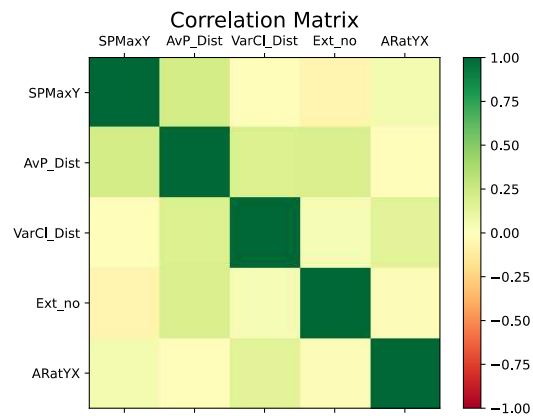
„K_means_raw.csv“ = Raw data from k-means cluster analysis.

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

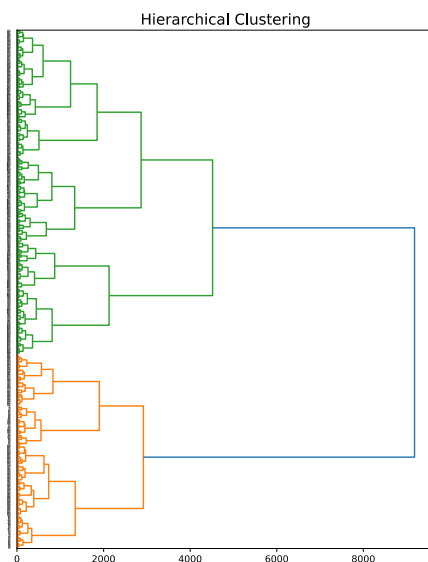
PCA:



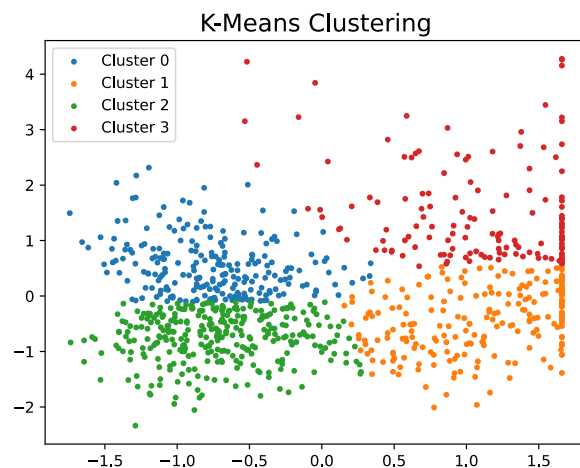
Correlation Analysis:



Hierarchical Clustering:



K-Means Clustering:



Abbreviations in result files (..._Final_Results.csv, acc_results.csv):

AMaxX =	X-extension of full cloud file	VarCI_Dist =	Variance of sub-cloud dist. to central spheroid
AMaxY =	Y-extension of full cloud file	MaxCI_Dist =	Maximum of sub-cloud dist. to central spheroid
AratYX =	Ratio of Y/X extensions (full cloud)	PixResol =	Pixel (voxel) calibration value in μm or equivalent
AratYZ =	Ratio of Y/Z extensions (full cloud)	ExNumber =	number of all extensions
AratXZ =	Ratio of X/Z extensions (full cloud)	Brim =	number of extensions with brim charact.
A_Hint =	Crude shape estimate full cloud	Broad =	number of extensions with broad charact.
Sep_Comp =	Count of separatable sub-clouds	Narrow =	number of extensions with narrow charact.
SpMaxX =	X-extension of full central spheroid	Small =	number of small extensions
SPMaxY =	Y-extension of full central spheroid	Maxlength =	max. length of extensions
SPRatYX =	Ratio of Y/X extensions (spheroid)	AvMaxlength =	average of all max. points of extensions
SPRatYZ =	Ratio of Y/Z extensions (spheroid)	Avlength =	average length of extensions
SPRatXZ =	Ratio of X/Z extensions (spheroid)	MaxSurface =	largest extension surface area found
Sp_Hint =	Crude shape estimate spheroid	AvSurface =	average extension surface area
MedCI_Dist =	Median sub-cloud dist. to central spheroid	AvMedlength =	median length of extensions
AvCI_Dist =	Average sub-cloud dist. to central spheroid	AvVarlength =	variance of extension lengths

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

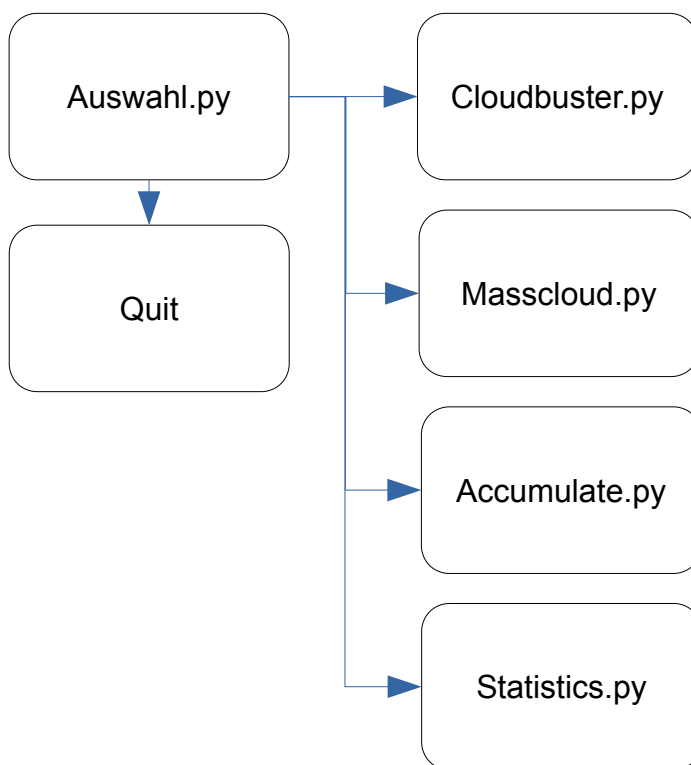
Result files are stored in the subfolder „results“.

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

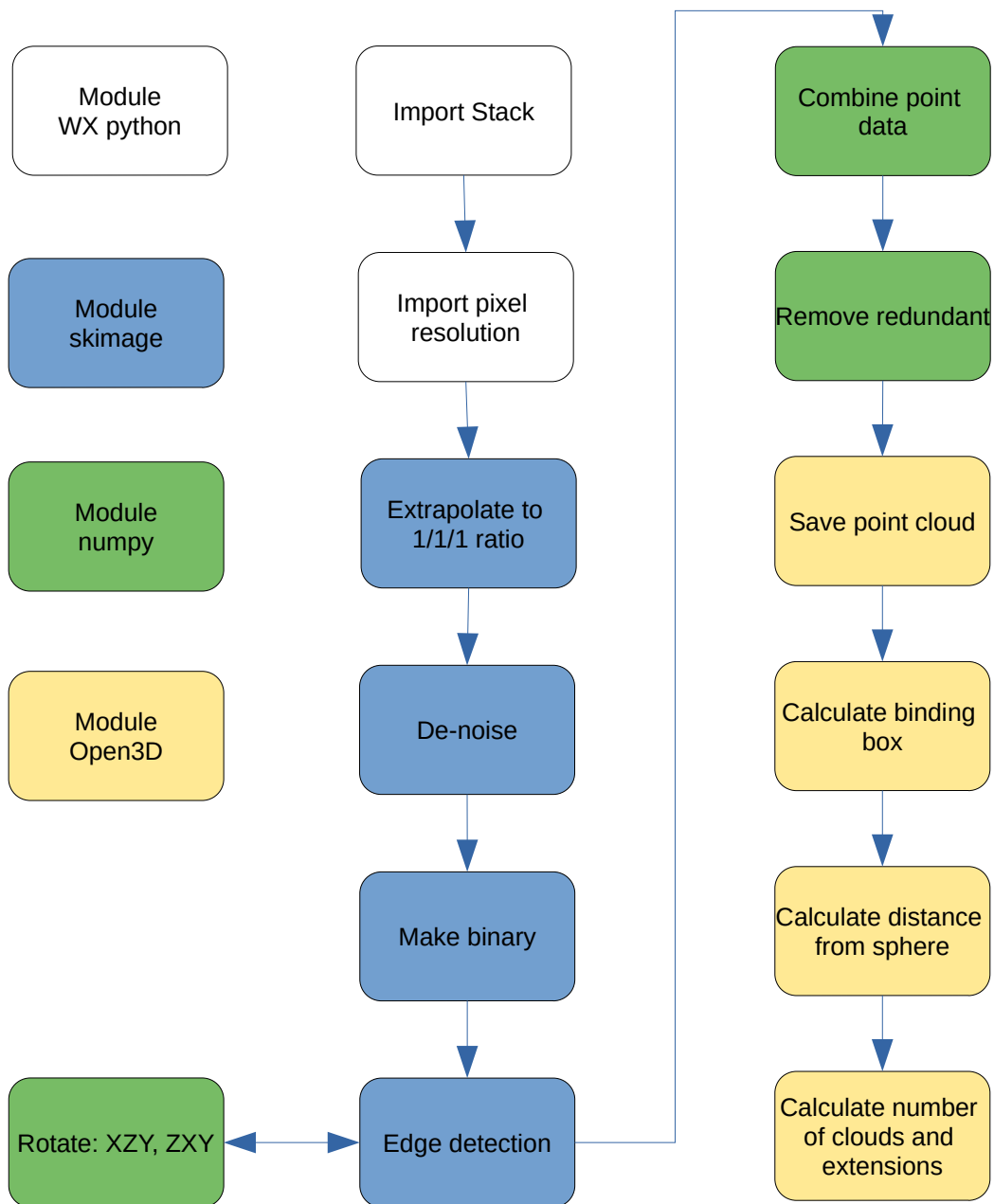
- ➔ filename.ply = Basic point cloud.
- ➔ filename_color.ply = color coded identified parts.
- ➔ filename_ellipsoid.ply = fitted ellipsoid point cloud.
- ➔ filename_extensions.ply = identified extensions point cloud only.
- ➔ filename_largest.ply = largest (central) spheroid point cloud only.

Background:

The aim of the project was the analysis of the morphology of generated 3D spheroids and migratory cells escaping from the spheroids to characterise, for example, the effect of anti-migratory inhibitors in depth. We had previously reported on distinct changes in cell morphology in single migratory cells in 3D systems after treatment with various small molecule inhibitors (Ketchen et al). The current python based workflow developed as part of this work transforms slice based 3D images into a point cloud structure and delivers a basic quantification of spheroids with emphasis on the 3D nature of the structure. To address the requirement to analyse single image stacks as well as series of image stacks for statistical analysis resulted in a set modular python files that can be started individually. A simple selection menu, Auswahl.py has been implemented to allow the user to choose between the required scripts:

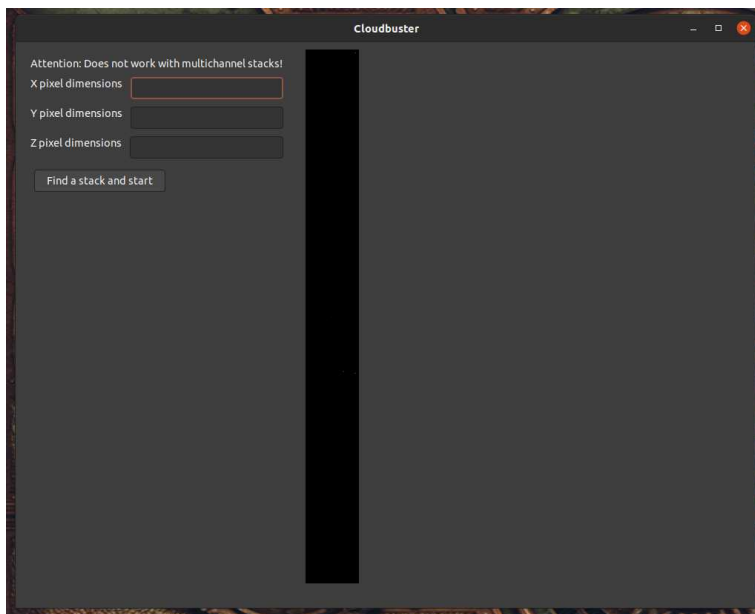


Cloudbuster.py as a single image stack analysis script and Masscloud.py for the sequential analysis of multiple image stacks provide 25 morphological parameters for each image stack. The basic workflow of Cloudbuster.py and Masscloud.py is identical:



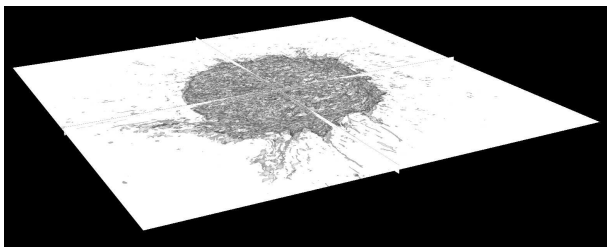
In case of Masscloud.py this workflow will be repeated for a list of image stacks.

It requires the entry of pixel dimensions in all three directions:

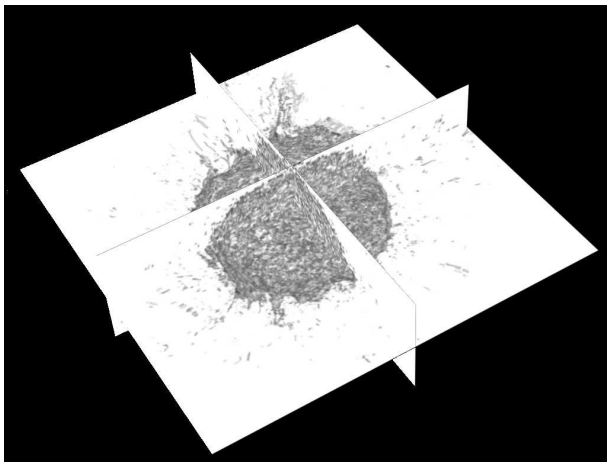


The software first extrapolates the initial image stack (A) to a 3D stack with even dimension ratios using the python skimage module (B). It then applies a transformation to a binary stack, also performed by the python skimage module (Sezgin & Sankur 2004) :

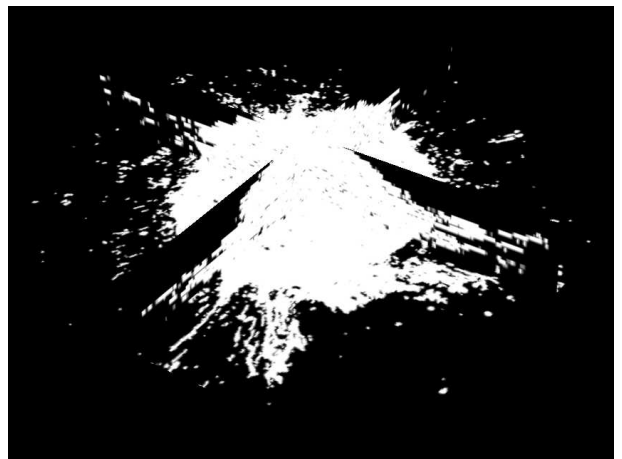
A



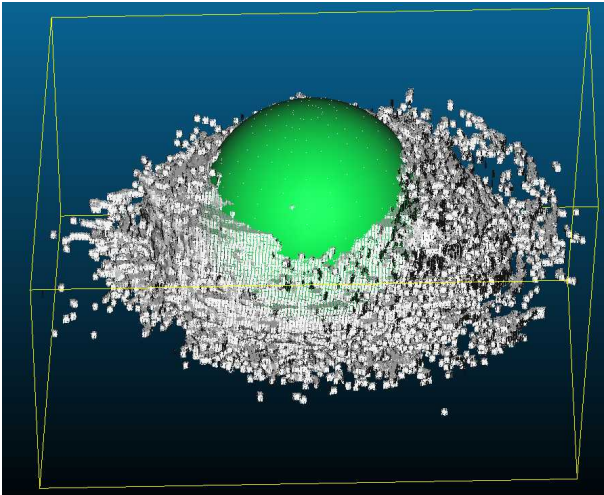
B



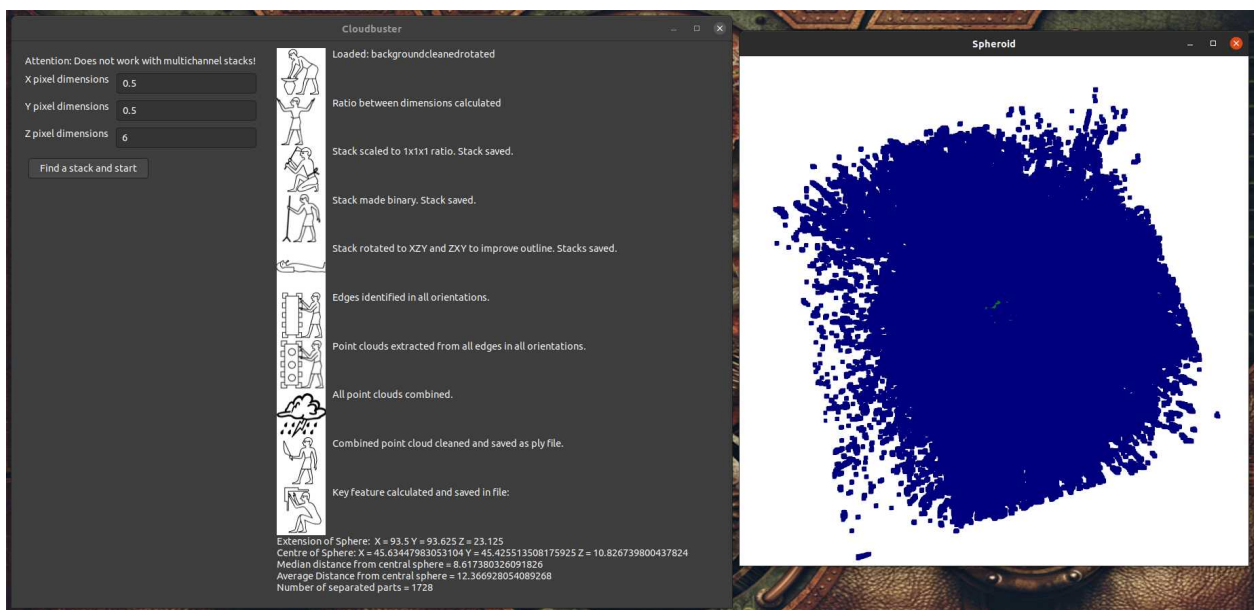
C



To reduce data volume for the analysis a skimage edge detection algorithm (Canny, 1986) is applied, where the original stack matrix is rotated twice with the numpy module to cover the surface as complete as possible. The extracted edge positions of all three orientations are combined to a single array. Redundant points are removed from the array with the numpy module and the result is saved as a point cloud in PLY format with the open3D module (Zhou et al 2018) (Meshlab representation):



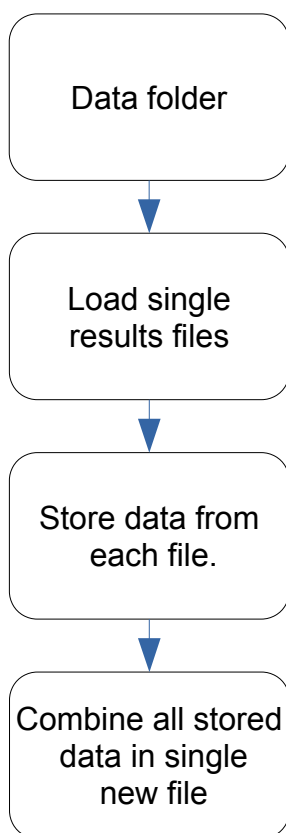
The open3D module is also used for the quantification of the XYZ spheroid extension and, after addition of a central sphere with minimum radius, the average/median distance of points to the central sphere structure. For the quantification of separated clouds (i.e. cells or smaller spheroids) the open3D library also is used:



Intermediate results are individually saved for further investigation. The numeric results are stored with the addition of “_Final_Results.txt” to the original filename. This specialised

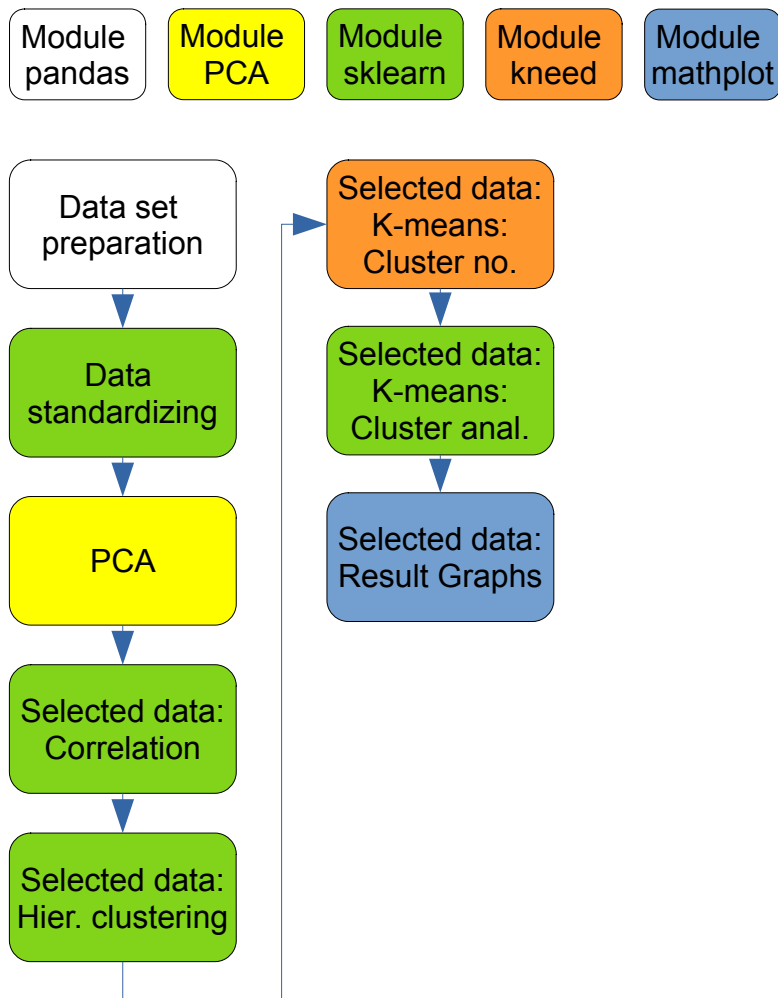
python script bridges the gap between slice based image analysis and 3D based entity analysis without providing, while avoiding the need extensive background knowledge from the user. The handling of the script is therefore straight forward, resulting in indicative values for the evaluation of the spheroid quickly.

To evaluate differential treatments, like inhibitors, obviously larger number of experiments are necessary and thus larger number of image stacks need to be analysed. To prevent the repetitive call of Cloudbuster.py, Masscloud.py has been introduced. Additionally resulting data from several image stacks are collected to two single .csv files with accumulate.py. The two files separate overall morphology from extension data of the isolated central spheroid:



Data analysis is provided by the statistics.py script. The script uses basic machine learning modules to identify the most significant parameters and the resulting relations of the involved spheroids to each other. In a first step PCA is executed using all numerical data parameters. The 5 dimensions (parameters) with the strongest contribution to the overall variance are selected and subjected to a correlation analysis. The 5 most important parameters can serve as indicators for the spheroid behavior and the correlation analysis can provide information how these parameters are related to each other. The highest rated parameter is used for a hierarchical clustering algorithm that allow an estimate of how the samples are related to each other. This is refined by a k-means clustering, including an

estimate for the best number of clusters. Data are sequential passed from one analysis to the next and documented with graphs in .svg format and raw data in .csv format:



Literature:

Qian-Yi Zhou, Jaesik Park, Vladlen Koltun: Open3D: A Modern Library for 3D Data Processing. ArXiv:2018; 1801.09847

Canny, J.: A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986

Sezgin M. and Sankur B.: (2004) "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation" Journal of Electronic Imaging, 13(1): 146-165 DOI:10.1117/1.1631315

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Finding a “Kneedle” in a Haystack: Detecting Knee Points in System Behavior. Ville Satopa † , Jeannie Albrecht† , David Irwin‡ , and Barath Raghavan§

†Williams College, Williamstown, MA ‡University of Massachusetts Amherst, Amherst, MA

§ International Computer Science Institute, Berkeley, CA