**PCA analysis manual**

1. The PCA analysis software has been tested most recently using Python 3.8.10 on a Windows 10 machine with the following packages installed:

|  |  |
| --- | --- |
| **Package Name** | **Version Tested** |
| sklearn | 1.3.0 |
| python-docx | 0.8.11 |
| pandas | 2.0.3 |
| numpy | 1.24.4 |
| matplotlib | 3.7.2 |
| customtkinter | 5.2.0 |
| scipy | 1.10.1 |

|  |  |
| --- | --- |
| (optional) ipykernel | 6.25.1 |
| (optional) torch | 2.0.1 |
| (optional) torchvision | 0.15.2 |
| (optional) optuna | 3.3.0 |
| (optional) tqdm | 4.66.1 |
| (optional) seaborn |  |

(The optional packages are only required for some of the Jupyter notebooks in the repository, which are used to test future functionality.) Follow the instructions below to set up Python and the necessary packages on your PC.

1. If on Windows, download the latest Ubuntu version in addition to the Windows Subsystem for Linux (WSL) on your computer. If you are a Mac user, you can skip to the next step.
   1. Download and install Python software with version 3.8.2 or newer at https://www.python.org/downloads/ or by using your preferred package manager between pip and Anaconda.

1.2. Open Terminal and activate your Python environment.

1.3. Check installed packages.

       1.3.1 (Optional) Exit python in the Terminal by typing exit. Now, your path should under computername:~username$  
       1.3.2 type pip list to check the list of packages installed. You need to have packages listed in the table above.

1.4. Install packages (If not already installed).

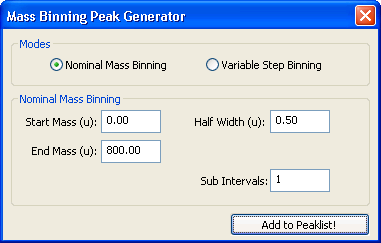
       1.4.1 Enter pip install packagename into Terminal.

       1.4.2 If you cannot install with pip install command, install package manually. For example: Google python-docx to download this package at <https://pypi.org/project/python-docx/>. Manually install this package by typing python3 -m pip install ./downloads/python-docx-0.8.11.tar.gz (Mac version) in the Terminal or typing py -m pip install ./downloads/python-docx-0.8.11.tar.gz (PC version) in the Terminal. (Use the correct path of the file.)

1. Generate the spectra data in .txt format (we use IONTOF’s Surface Lab software as an example to show how to export the unit mass .txt spectral data. If other manufacturers’ instruments are used, please follow their instructions to generate the data.)
   1. In the “spectra” window of Surface Lab 6 or 7 software → open the set of interesting spectra (either positive or negative)
   2. For every spectrum, follow the following steps to calibrate it:

Double-click a spectrum to select it → press F3 (or Spectrum -> Mass Calibration) to enter the mass calibration menu → default calibration points for negative spectra should be H-, C-, C2-, C3- and for positive spectra should be H+, H2+, CH­3+, C­3H3+ → if H- is highlighted red, it is oversaturated and needs to be excluded (which is done automatically), so click “Recalibrate” in the lower right-hand corner → exit the mass calibration menu by pressing ESC

* Note for positive spectra calibration: The 4th default ion, C3H­3+, usually causes large deviations (>100ppm) on the calibration; we should be using CH3O+ instead. To fix this, on each spectra’s mass calibration menu, you should right click this ion 🡪 select “Delete” 🡪 click on the “Find” search bar on the top-middle section of the screen 🡪 type CH3O+ and press enter 🡪 click on the red arrow that pops up on the center panel 🡪 an ID for the species you selected should pop up on the mass calibration menu 🡪 if it is incorrect, tune your result on the zoomed-in spectrum panel below the main panel by clicking the red arrow on that panel 🡪 now CH­3O+ should show up in the mass calibration panel’s text box 🡪 select “Add” 🡪 select “Recalibrate” 🡪 your calibration should have smaller deviations now!
  1. Activate the 1st spectrum→ “Peak List” → “Create Nominal Mass Binning” → in the new small window, copy the parameters below and click “Add to Peaklist”



* 1. Don’t close the pop-up window → right-click the text box that says “Nominal Mass Binning” in the submenu found in the lower left-hand corner→ select “Add To” -> “All Spectra” → recheck a few sample spectra to make sure all new peak lists have been created → now close the pop-up window
  2. Now right click the dialog box next that says “Nominal Mass Binning” again → select “Replace All Existing Peak Lists Of -> All Spectra to delete the extraneous “List 1” for all other spectra → also make sure to delete “List 1” for the current spectra if it is still there → this list can cause problems later, so MAKE SURE to delete it now!
  3. Select all spectra → click the “Peak Statistics On Selected Spectra” button at the left side of the peak list sub-window → a new window should pop up after 1-2 minutes, showing the results → select “Area normalized by Total Ion Intensity” → save the .txt file

1. Change default working directory: find main.py and the folder in which it exists—this should be the path of the location in which you downloaded the PCA analysis software followed by “pca-analysis/SIMS\_PCA/SIMS\_PCA/src” 🡪 open main.py with notepad or your chosen IDE 🡪 on line 18 of main.py, change the value of pcaDir to ”<PATH\_TO\_PCA>/pca-analysis/SIMS\_PCA/SIMS\_PCA”
2. Group column names of spectra file: Open the .txt data file using Excel → remove any irrelevant columns and rows (i.e., we probably don’t need the top- or bottom-most two rows or the stats columns at the end (mean, median, etc.) → make sure the columns are in order: Do this by highlighting all cells and going to “Data” -> “Sort” -> add “Sort by” “Row 1”, then select “Options” -> “Sort left to right” -> click OK → (1) make sure the 1st column’s name is “Mass (u)” 🡪 edit column headers; currently, they might be something like <No Sample Name> (01-N1) \*, <No Sample Name> (01-N2) \*…..; change them to 01-N1, 01-N2, 01-N3, 01-N4, 01-N5, 01-N6, 02-N1, 02-N2, ….. using CTRL+H to find and replace any extraneous character expressions → save the new file as “sims\_data.txt” → paste the new file into the pca data folder (<PATH\_TO\_PCA>/pca-analysis/SIMS\_PCA/SIMS\_PCA/sims-data/OriginalData)
3. Now open the file again in Notepad (preferably Notepad++ for RegEx editing purposes) → if there are any blank lines, remove them using either Notepad++’s features (select “Edit” -> “Line Operations -> “Remove Empty Lines”) or a Python script → also remove any trailing whitespace using Notepad++ by selecting “Edit” -> “Blank Operations” -> “Trim Trailing Space” → once you are certain that no pesky extraneous whitespaces are remaining, save the file and exit (if you have ANY formatting differences from what the code expects, you could get an error when doing PCA analysis later; usually, \*\*\*Error! Cannot Recognize Data!\*\*\*. Here’s a snapshot from a sample .txt data file immediately after it was generated by Surface Lab 7:

Table

Description automatically generated with medium confidence

and after being cleaned up:

Text, table

Description automatically generated

1. In the start menu, find the Terminal corresponding to your machine 🡪 Run it as administrator 🡪 Input “cd <PATH\_TO\_PCA>/pca-analysis/SIMS\_PCA/SIMS\_PCA/src” then press Enter 🡪 Input the command python main.py, then press Enter
2. Go to the folder (<PATH\_TO\_PCA>/pca-analysis/SIMS\_PCA/SIMS\_PCA/sims-data/output\_sample to check PCA results
   1. 10 combinations of 2-d PCA scores plots of PC1-PC5 with confidence circles
   2. 10 combinations of 2-d PCA scores plots of PC1-PC5 without confidence circles
   3. PC1-PC5 single PC scores plots
   4. PC1-PC5 scores tables for all spectra
   5. PC1 to PC5 loadings plots
   6. PC1-PC5 top 20 loadings tables

**Note 1:** In the 10 combinations of 2-d PCA scores plots of PC1-PC5 with confidence circles**,** 10 combinations of 2-d PCA scores plots of PC1-PC5 without confidence circles, 1-d PC1 to PC5 scores plots, andPC1 to PC5 loadings plots**,** to change plot characteristics such as the **font type** and **size** along with the plot **label** and **marker** attributes**:** 🡪 go to the folder <PATH\_TO\_PCA>/pca-analysis/SIMS\_PCA/SIMS\_PCA/src/pca\_sims and open plotting.py🡪 find the following lines: line 18-28 🡪 change the parameters below 🡪 save it and run again:

'font.family':'serif',

    'font.serif':'Times New Roman',

    'axes.labelsize': '28',

    'xtick.labelsize':'28',

    'ytick.labelsize':'28',

    'lines.linewidth':3,

    'lines.markersize':10,

    'legend.fontsize': '25',

**Note 2:** The circles on the graphs are specified by a **Confidence Interval** (or “confidence limits”) of 90%. If other values are needed, open plotting.py 🡪 find “nstd = 1.645” on line 128 🡪 change “1.645” to the desired value based on the below table 🡪 save plotting.py and rerun the main python script main.py:

|  |  |
| --- | --- |
| ***Confidence Interval*** | ***Z*** |
| 80% | 1.282 |
| 85% | 1.440 |
| 90% | 1.645 |
| 95% | 1.960 |
| 99% | 2.576 |
| 99.5% | 2.807 |
| 99.9% | 3.291 |

**Note 3:** If more or fewer **PCs** are required 🡪 open main.py 🡪 find line 53 and change pca\_sims.plot\_pca\_result() to pca\_sims.plot\_pca\_result(max\_pcacomp=n), where n is the desired number of components 🡪 save main.py

**Note 4:** In the 10 combinations of 2-d PCA scores plots of PC1-PC5 with confidence circles, 10 combinations of 2-d PCA score plots of PC1-PC5 without confidence circles, and 1-dPC1 to PC5 score plots, to change **symbol sizes** or **colors** 🡪 open plotting.py 🡪 find lines 130-137 🡪 replace the current strings with the desired ones according to the matplotlib documentation 🡪 save plotting.py and run main.py again

markern = [ ',','o','H','^','v','x','1','D','X','2','3','4','8','s','p','.',

                    'h','+','d','|','\_',0,1,2,3,4,5,6,7,8,9,10,11]

        colorn = ['purple','green','blue','brown','red',

                    'teal','orange','magenta','pink',

                    'gray','violet','turquoise','yellow',

                    'lavender','tan','cyan','aqua','yellowgreen','chocolate',

                    'coral','fuchsia','goldenrod','indigo',

                    'grey','darkorange','rosybrown','palegreen','deepskyblue']

**Note 5:** In PC1-PC5 loadings tables, to change **the** **numbers of +/- loadings**: open pca\_sims.py and find line 235 in addition to opening plotting.py and finding line 258 🡪 change fetchn\_more=20 in both places to the desired number of loadings 🡪 save both files and run main.py again.

**Section 1: Formatting the Raw Data File**

The following section provides detailed guidance to the user on how to properly format the raw data .txt file. Careful consideration of the formatting is necessary for the code to execute.

**Note 1:** The first column name is ‘Mass (u)’ and the values in the column should be unit masses; i.e., whole integers.

**Note 2:** The remaining column names should be adjusted as follows:

* The general column name format should take the form ‘XXX-XX’
* The first three digits are the sample group number
  + For the first sample of interest, XXX = 479
  + There can be up to 999 samples simultaneously analyzed
* The last two characters specify whether 1) the spectra are positive or negative ions and 2) the measurement location number, respectively
  + For example, if there are six locations analyzed for the first sample, and the data is from negative ion mode for the first location, then XX = N1
* Additional example: Column name for sample 479 at the fourth location analyzed in negative mode: 479-N4

The following screenshot shows an example of the raw data file to illustrate proper formatting:

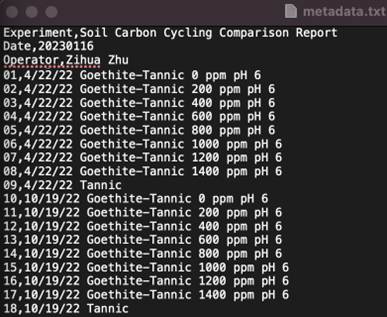
Text, table

Description automatically generated

**Note 3:** Save the formatted data as a text file in pca/SIMS\_PCA/SIMS\_PCA/sims-data/OriginalData

**Section 2: Formatting the Metadata File**

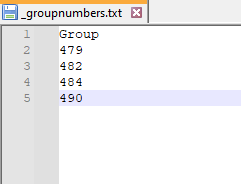
To properly correspond the sample name to the number inputted into the txt file from Section 1, the metadata.txt file (found in <PATH\_TO\_PCA>/SIMS\_PCA/SIMS\_PCA/sims-data/OriginalData) should be appropriately formatted. The following image shows an example; a key consideration is to not have any spaces after the commas in the txt file:

****

The leftmost column should contain the sample group numbers, and the rightmost column should contain the sample descriptions. These will be used to label the samples on the PCA plots found in the report. If they are incorrectly numbered, your sample descriptions won’t match up with the samples in the actual raw data file.

**Section 3: Formatting the Group Numbers File**

The file \_groupnumbers.txt, located in <PATH\_TO\_PCA>/pca-analysis/SIMS\_PCA/SIMS\_PCA/sims-data/OriginalData, must also be formatted properly before the software is run. The file should contain the word “Group” followed by a list of three-digit numbers, one on each line, as in the following example:

****

These represent the desired group numbers (from the top row in the raw data file) to be analyzed. If you would like to include all data in the PCA analysis, then make sure to write all sample numbers to this file; in this case, you would need to include 479, 480, 482, 483, 484, 489, 490, and 491—eight numbers in total. However, since only 4 numbers are given, only data from columns whose headers contain one of the sample numbers above will be included in the PCA analysis and subsequent component tables found in the report. Make sure that you include at least one three-digit number, and remember to double-check that each is actually one of your sample numbers. If you do happen to accidentally include a sample number that doesn’t exist, it will simply be ignored by the software.

**Section 4: Report Generation**

Once the raw data and metadata files have been properly formatted and saved, make the following changes to the main.py file (located in <PATH\_TO\_PCA>/SIMS\_PCA/SIMS\_PCA/src) to analyze the data:

**Note 1:** Indicate whether in positive or negative ion mode:

****

**Note 2:** Specify the raw data txt file path and name:

****

**Note 3:** Once the main.py file is updated and saved, execute it again in the Terminal or equivalent location to execute Python code. If successfully run, the complete PCA/SIMS spectra analysis report can be found in <PATH\_TO\_PCA>/SIMS\_PCA/SIMS\_PCA/output\_sample/report.docx.

**Section 4: Peak Assignment Table in Report**

Once the PCA code is successfully run, a table of peak assignments can be found in the report. The peak assignments are taken from the ‘negative\_doc\_mass\_record.csv’ and ‘positive\_doc\_mass\_record.csv’ documents found in <PATH\_TO\_PCA>/SIMS\_PCA/SIMS\_PCA/sims-data. These are the default peak assignments, but the user can manually input different assignments for a given unit mass based on the context of the sample.

For example, in positive ion mode, the signals of C2H3+, Al+, and CNH+ all correspond to a unit mass of 27. Therefore, the peak assignment table can be updated manually from the mass record files to reflect different options of peak assignments for the same unit mass. This enables the user to select peak assignments that are most relevant to the context of the sample analyzed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Default Mass Record Document | | | → | User Updated Mass Record Document | | |
| Unit Mass | Assignment | Document Mass | Unit Mass | Assignment | Document Mass |
| 27 | C2H3+ | 27.02 | 27 | C2H3+,  Al+,  CNH+ | 27.02,  26.98,  27.01 |

If the mass record documents are updated, the user must rerun the python script to have the updated peak assignments appear in the exported report document.

**Section 5: Adding Mass Combination Rules for Similar Chemical Species**

The following section provides guidance on how the user can incorporate rules to assign mass values to similar chemical species. For example, for SiOx-based signals, there can be a variety of mass values that are correspondent. Referring to the ‘negative\_doc\_mass\_record.csv’, values related to SiOx are tabulated.

|  |  |  |
| --- | --- | --- |
| Unit Mass | Assignment | Document Mass |
| 60 | SiO2- | 59.9673 |
| 61 | SiO2H- | 60.9751 |
| 76 | SiO3- | 75.9622 |
| 77 | SiO3H- | 76.97 |
| 136 | Si2O5- | 135.9289 |
| 137 | Si2O5H- | 136.9368 |

**Note 1:** To add additional assignments for a particular chemical species, the user must add information regarding unit mass, signal assignment, and document mass into the mass record file

* For positive ion signals, make additions to ‘positive\_doc\_mass\_record.csv’
* For negative ion signals, make additions to ‘negative\_doc\_mass\_record.csv’
* These files can be found in <PATH\_TO\_PCA>/SIMS\_PCA/SIMS\_PCA/sims-data/

**Note 2:** Once these documents are updated for their respective signals, the user must update the python file which employs the mass combination rules by specifying the relevant unit masses.

* The python file to update is found here: <PATH\_TO\_PCA>/SIMS\_PCA/SIMS\_PCA/src/pca\_sims/pca\_sims.py
* The following snapshot is a sample code in the pca\_sims.py file (line 442-460)
* The user should add unit masses for signals corresponding to the SiOx group in line 443:

Text, letter

Description automatically generated