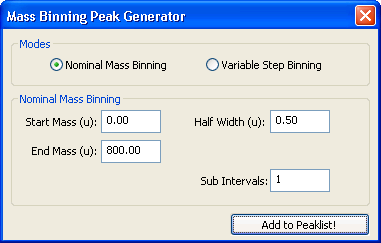
**PCA analysis manual**

1. Export txt data
   1. In the “spectra” window of Surface Lab 6 software → open the set of interesting spectra (either positive or negative)
   2. Activate the 1st spectrum → open “mass calibration” small window → carefully calibrate the spectrum → double click to activate the 2nd to the last spectrum one-by-one → carefully calibrate each spectrum → Re-check each spectrum to make it sure all spectra have been calibrated
   3. Activate the 1st spectrum→ “Peak List” → “Create Nominal Mass Binning” → in the new small window, select the below typical parameters and click “Add to Peaklist”



1. Don’t close the small window → double click to activate the 2nd to the last spectrum one-by-one → create new peak list for each spectrum→ Re-check each spectrum to make it sure all new peak lists have been created
2. Select all spectra → Click ‘Statistics’ button at the left side of the peak list sub-window → a new window jump out after 1-2 minutes, showing the results → select “Area normalized by Total Ion Intensity” → Save the txt file
3. Change default working directory: find pca7.py file and its accompanied PCA file folder, e.g., pca.py file is found in the folder of “C:/Anaconda/scripts/” and the accompanied PCA file folder is “C:/Anaconda/scripts/PCA/”🡪 open pca7.py file with notepad 🡪 in line of the default working directory, “pcaDir= “C:/Anaconda/scripts/PCA/” 🡪 if necessary, adjust directory to make it sure these two directories are same.
4. Group column names of spectra file: Open the .txt data file using Excel → cut the relevant data columns → Paste in a new Excel file → Group spectra by change name of the column → (1) Make it sure the 1st column’s name is “Mass (u)” 🡪 by adding group numbers 01, 02, 03…… (up limit, 99 groups) before the spectra column names based their group number, e.g., the original column names are A, B, C, D, E, F, G, H, I, J, K, L, M, N…; after re-group are 01A, 01B, 01C, 01D, 02E, 02F, 02G, 03H, 03I, 03J, 03K, 03L, 03M, 03N, ….. → save the new file as “Text (Tab delimited) (\*.txt)” file → paste the new file into the folder: C:/Anaconda/scripts/PCA/OriginalData/
5. Rename group names in final figures: go to C:\Anaconda\scripts\PCA\Group Names\ → open Group Names.txt → after 1, 2,3, 4 …, rename or input names you want to show in the final figure for group 01, 02, 03, 04….
6. In “Start menu” find out “Anaconda Powershell Prompt (anaconda).exe”, Right click “Anaconda Powershell Prompt (anaconda).exe” 🡪 Run as administrator 🡪 Input “cd C:\Anaconda\scripts” then press Enter 🡪Input command python pca7.py then press Enter
7. Go to the folder C:/Anaconda/scripts/PCA/Exports/ to check PCA results
   1. 10 combinations of 2-d PCA scores plots of PC1-PC5 with confidence circle
   2. 10 combinations of 2-d PCA scores plots of PC1-PC5 without confidence circle
   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
   7. PC1 to PC10 weight “Percentage of explained variance”
   8. PC1 to PC10 weight “Percentage of explained variance” table

**Note 1:** In 10 combinations of 2-d PCA scores plots of PC1-PC5 with confidence circle**,** 10 combinations of 2-d PCA scores plots of PC1-PC5 without confidence circle, 1-d PC1 to PC5 scores plots, andPC1 to PC5 loadings plots **,** change **coordinate system sizes, figure resolution** 🡪 open pca7.py 🡪 find the following items: line 16-32 🡪 change the parameters 🡪 save it and run again:

'axes.labelsize': '28',

'xtick.labelsize':'28',

'ytick.labelsize':'28',

'lines.linewidth':3,

'lines.markersize':10,

'legend.fontsize': '25',

'dpi = 300'

**Note 2:** The circle is **Confidence Interval** (or “confidence limits”) = 90%. If other values are needed, open pca7.py 🡪 find “nstd = 1.645” 🡪 change “1.645” to the values based on the below table 🡪 save pca7.py and re-run it:

|  |  |
| --- | --- |
| ***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 less **PCs** are required 🡪 open pca7.py 🡪 find the following items 🡪 change the parameters 🡪 save pca7.py and re-run it: (Assume that the number of required PCs is n, modify the red areas highlighted below)

Line 68-69：per\_var=np.round(pca.explained\_variance\_ratio\_\*100,decimals=2)[:n]

labels=['PC'+str(x) for x in range(1,n+1)]

Line 92: pca\_data=pca\_data[:,:n]

Line 111: pca\_df.to\_csv(pcaDir+'Exports/PC1-n\_SCORE\_TABLE.txt')

Line 147-148: for j in range(1, n+1):

for k in range(j+1, n+1):

Line 169-170: for j in range(1, n+1):

for k in range(j+1, n+1):

Line 204: for pc in range(1,n+1):

Line 242-243: loading\_scores=pca.components\_[:n,:]

loadingTable=pd.DataFrame(loading\_scores.T,index=mass,columns=[1,2,3,4,5…,n])

Line 248: for i in range(0,n)

Line 296: loadingTable=pd.DataFrame(loading\_scores.T,indx=mas.columns=['PC1', 'PC1', 'PC1'，'PC1', 'PC1', …, 'PCn']

Line 300: loadingTable.to\_csv(pcaDir+'Exports/PC1-n\_loadingTable.txt')

**Note 4:** In 10 combinations of 2-d PCA scores plots of PC1-PC5 with confidence circle, 10 combinations of 2-d PCA score plots of PC1-PC5 without confidence circle and 1-dPC1 to PC5 score plots, change **symbol sizes, colors** 🡪 open pca7.py 🡪 find the following items: line 136-145 🡪 change the parameters 🡪 save pca7.py and run again

**Note 5:** In10 combinations of 2-d PCA scores plots of PC1-PC5 with confidence circle, change  **figure scale**, **label fontsize, fontname, fontweight and circle linestyle, linewidth, facecolor, alpha**: open pca7.py 🡪 find the corresponding parameters: line 169-201 🡪 change the parameters 🡪 save pca7.py and run again.

**Note 6:** In 10 combinations of 2-d PCA scores plots of PC1-PC5 without confidence circle, change **figure scale, label fontsize, fontname, fontweight**: open pca7.py 🡪 find the corresponding parameters: line 147-167 🡪 change the parameters 🡪 save pca7.py and run again.

**Note 7:** In PC1 to PC5 single PC scores plots, change **figure scale, label fontsize, fontname, fontweight**: open pca7.py 🡪 find the corresponding parameters: line 204-228 🡪 change the parameters 🡪 save pca7.py and run again.

**Note 8:** InPC1 to PC5 loadings plots, change **figure scale,** **fetch number, label fontsize, fontname, fontweight and bar size, color, text fontsize**: open pca7.py 🡪 find the corresponding parameters: line 240-270 🡪 change the parameters 🡪 save pca7.py and run again.

**Note 9:** In PC1 to PC10 weight “Percentage of explained variance” plots, change **figure scale, ticklabel size, label name, fontsize, fontname, fontweight and bar size, color**: open pca7.py 🡪 find the corresponding parameters: line 68-87 🡪 change the parameters 🡪 save pca7.py and run again.

**Note 10:** In PC1-PC5 loadings plots, change **the labeling numbers of m/z**: open pca7.py 🡪 find the corresponding parameters: line 244 🡪 change the parameters 🡪 save pca7.py and run again. (Assume that the number of required labeling numbers is m, modify the red areas highlighted below)

fetchn=m

**Note 11:** In PC1-PC5 loadings tables, change **the** **numbers of +/- loadings**: open pca7.py 🡪 find the corresponding parameters: line 245 🡪 change the parameters 🡪 save pca7.py and run again. (Assume that the number of required numbers of +/- loadings is l, modify the red areas highlighted below)

fetchn\_more=l

**Section 1: Formatting Raw Data File**

The following section provides 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 whole number integers of the unit mass.

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

* The general column name format takes the form ‘XXXX’
* The first two characters are the sample number
  + For the first sample of interest, XX = 01
  + There can be up to 99 samples simultaneously analyzed
* The last two characters consider whether the spectra are in positive or negative ion mode, and the measurement location number
  + 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 the third sample at the fourth location analyzed in negative mode: 03N4

The following screenshot shows an example of the data formatting:

**Table

Description automatically generated with medium confidence**

**Note 3:** Save the formatted data as a text file in SIM\_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 SIM\_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:

**Graphical user interface, text

Description automatically generated**

**Section 3: 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 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 in Terminal or equivalent location to execute Python code. If successfully run, the complete PCA/SIMS spectra analysis report can be found in 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 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 re-run 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 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: 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 the unit mass for the signals which correspond to SiOx in line 443

Text, letter

Description automatically generated