1 EXTRACTION OF MSX SCANS: MSX_EXTRACTOR.PY

Software required:
Python version: 3.9.7 (https://www.python.org/downloads/release/python-397/), Proteowizard version 3.0.21246 (https://proteowizard.sourceforge.io/)
Python Packages required: tkinter, sys, time, os, pymsfilereader, pathlib[1]

INPUT:

- Dir: folder with raw files to convert.
- IS-ENDO file: pairs of mz for IS and ENDO peptides (Figure 1)

```
434.881286,432.209886
1009.486558,1004.482423
1044.976839,1040.969739
1048.145059,1045.473659
1189.560505,1186.224415
1206.039222,1201.035088
367.931562,365.429495
408.211336,404.204237
409.203661,405.867572
424.722068,420.714969
448.194502,444.187402
```

Figure 1. Example of IS-ENDO File

OUTPUT:

- Mzml files: one mzml file per raw file containing only the desired MSx scans. To be used as input for Skyline.
- Results.txt files: one per raw file, contains the injection times per scan. Required for IT normalization. Copy all txt files into a new folder and label it as "txt".
- Config.txt: one file per raw file, used internally by ProteoWizard. They can be deleted when the conversion is finished.

BEFORE START:

1. Locate the folder that contains the installation of ProteoWizard.
2. Open with a text editor (such as NotePad++) the app "MSx_Extractor.py"
3. Go to line 62 in the python script and update the directory to the one where you have installed ProteoWizard. Remember to keep the "\\" between subdirectories.
4. Copy all the raw files into a new folder.

RUNNING THE APP:

1. Double click on MSx_Extrator.py. Two windows would appear (Figure 2 and 3), one corresponding to the GUI and the python console.

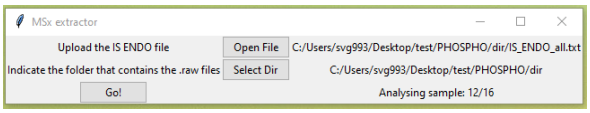

Figure 2. MSx_Extractor.py main window.

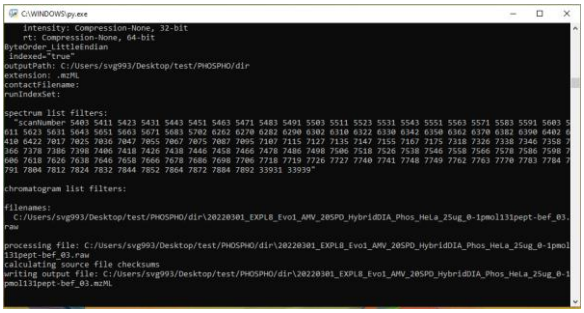

Figure 3. Python console.

[1]For help installing python packages: https://packaging.python.org/en/latest/tutorials/installing-packages/#id18

2. Upload IS ENDO file by clicking on “Open file” and navigating to the location of the file.
3. Indicate the folder that contain the .raw files of interesting by clicking on “Select Dir”.
4. Click “Go!” to start the analysis. A message saying “Analyzing sample n/n” would appear and it would be updated every time a raw file is done. When all raw files have been processed, the message “Done” will appear.
5. During the raw file processing the python console (Figure 3) would show ProteoWizard actions.
6. When done, three files would have been generated per raw file: .mzml, results.txt and config.txt. For convenience in further steps, copy all results.txt files into a new folder.

[1]For help installing python packages: https://packaging.python.org/en/latest/tutorials/installing-packages/#id18

2 IMPORT INTO SKYLINE: EXTRACTION OF THE TRANSITIONS.

Required: Skyline template with the desired transitions (.sky) and fragment library (.blib).

1. Edit the following parameters in the "Transition settings tab" (see figure below for details):
    - Filter tab: remove "b" ions from ion types. They cannot be used for quantification since they are shared between the heavy and light peptides.
    - Instrument tab: set "method match tolerance m/z" to 0.001.
    - Full-Scan:
        - MS1 filtering: none
        - MS2 filtering: PRM

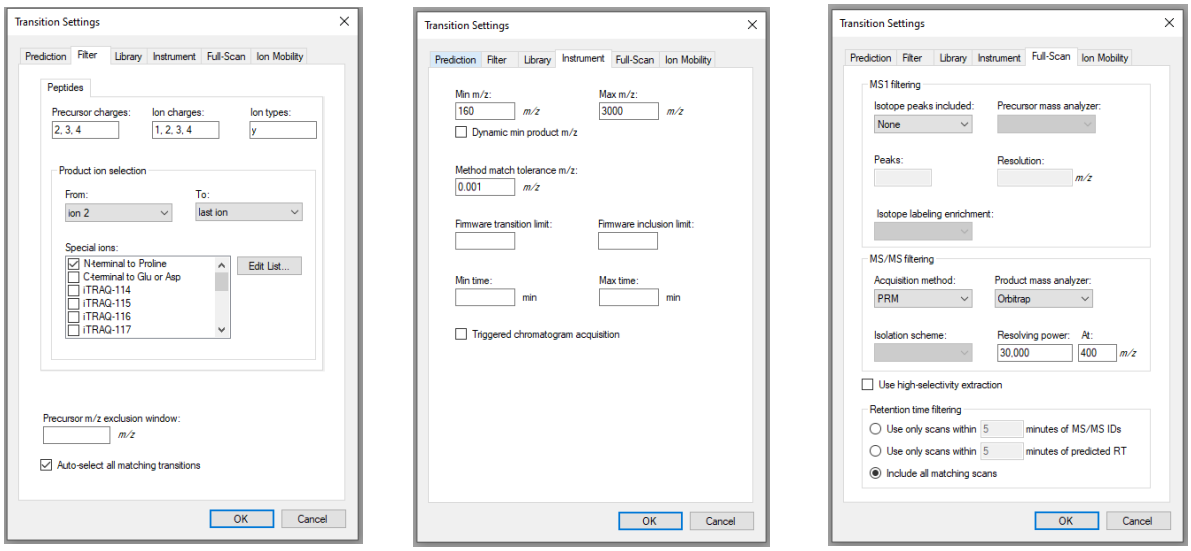

Figure 4. Transition settings tab in Skyline with specific parameters required for the succesfull import of the processed mzml files.

2. Import mzml files: File > Import > Results
3. Open Document grid (Alt+3 or View>Document Grid). Edit a document to contain the following information and safe the report as "Quantification_IS_ENDO_forHybrid".
    a. Peptide.
    b. Protein. IMPORTANT: This header would indicate in the next step the IDs for each IS/ENDO pair. In the case of more than one peptide per protein, modify the header to "ProteinName", and rename "Peptide" column to "Protein".
    c. Replicate.
    d. Raw Intensities.
    e. Raw Times.
    f. Fragment Ion.
    g. Product Mz.
    h. Isotope Label Type.
    i. Mass Error PPM.
    j. Peptide Peak Found Ratio.
    k. Raw Spectrum Ids.
4. Export Report "Quantification_IS_ENDO_forHybrid" (.csv format).

[1]For help installing python packages: https://packaging.python.org/en/latest/tutorials/installing-packages/#id18

5. Open Document grid again and load the table named “All_Precursors_Initial_Survey”. Edit the table to contain only two columns:
    a. Protein Name. IMPORTANT: This header would indicate in the next step the IDs for each IS/ENDO pair. In the case of more than one peptide per protein, choose the appropriate column for each unique pair (i.e: Modified Peptide Sequence). The content needs to match the content of the “Protein” column from the Quantification report.
    b. Precursor Mz.
6. Export Report “All_Precursors_Initial_Survey” (.csv format).

[1]For help installing python packages: https://packaging.python.org/en/latest/tutorials/installing-packages/#id18

3 INJECTION TIME NORMALIZATION, QUANTIFICATION AND VISUALIZATION OF RESULTS.

Software required: R (v4.0)
R packages required: shiny, shinyFiles, shinycssloaders, dplyr, ggplot2, data.table, gridExtra, tidyr, ggpubr, MESS, config

1. Copy the file app.R and config.yml to your computer into a folder called "Shiny-APP".
   The file "config.yml" contains certain parameters to filter the data do the quantification. There data in the file can be used as a default, but it can be modified if needed. The parameters are:
   ppm: max error mass allowed. Default: 10 ppm
   n_y: min number of y fragments required for quantification. Default: 1
   ppfr: peptide peak found ratio from Skyline. Default: 0.5

2. Open R (it can be the R console or Rstudio) and type:

```
library(shiny)
setwd("C:/Users/xxxx/xxxx/dir_shiny_app/")
#Indicate the location of the folder "Shini-APP"
runApp("Shiny-APP")
```

   or click on "Run App" on the right top corner.

3. A shinyApp will open in a new window (Figure 5 and 6).
   Input:
   (1) directory where the txt files generated in the first step are stored.
   (2) common pattern in the raw files. If the raw files share a common pattern and it has been removed during Skyline processing, indicate it here.
   (3) Maximum injection time (IT) used during hybridDIA acquisition
   (4) Precursor MZ list: csv file that contains the mz for the heavy and light peptides together with their id (phospho-site, peptide).
   (5) Quantification results: csv files exported from Skyline in the previous step. When this document is loaded, two selection menus will appear, one for the sample and another for the peptide/phospho-site.

   Output:

   (6) Plot phospho-site: this button will plot the XIC of the IS and ENDO peptide for the selected phospho-site in the selected sample before and after IT normalization. Also, it will show a profile plot of the relative intensity (calculate from the ratio IS/ENDO) of that peptide across samples.
   (7) Print Heatmap: this button will plot the relative intensity (calculates from the ratio IS/ENDO) of all peptide across samples in the form of a heatmap. (In the "All results" tab).
   (8) AUC IS, AUC ENDO: these buttons will export a table with the AUC for the heavy (IS) and light (ENDO) counterparts for all peptides, normalized by IT.

[1]For help installing python packages: https://packaging.python.org/en/latest/tutorials/installing-packages/#id18

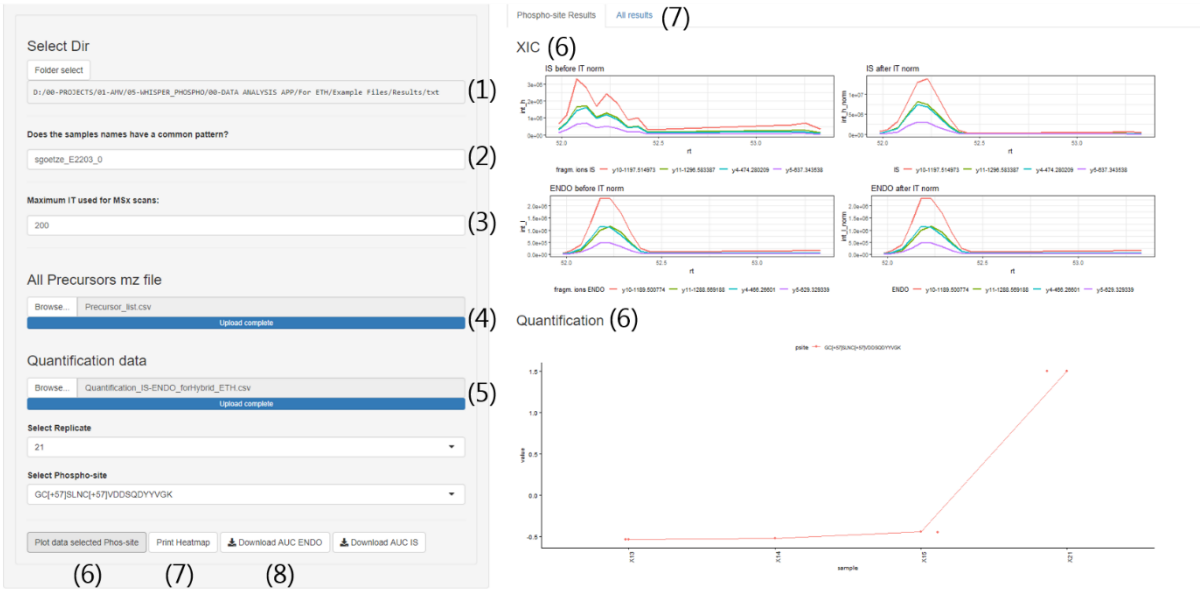

Figure 5. Shiny App interface. On the left panel: option menu to upload required files. Once Quantification data file is uploaded the selection tabs "Selection Replicate" and "Select phospho-site" appears for the user to choose what to plot. On the right panel: (top) XIC of peptide GC[+57]SLNC[+57]VDDSQDYYVGK for sample "21", (bottom)Relative Quantification of that peptide among the samples compared.

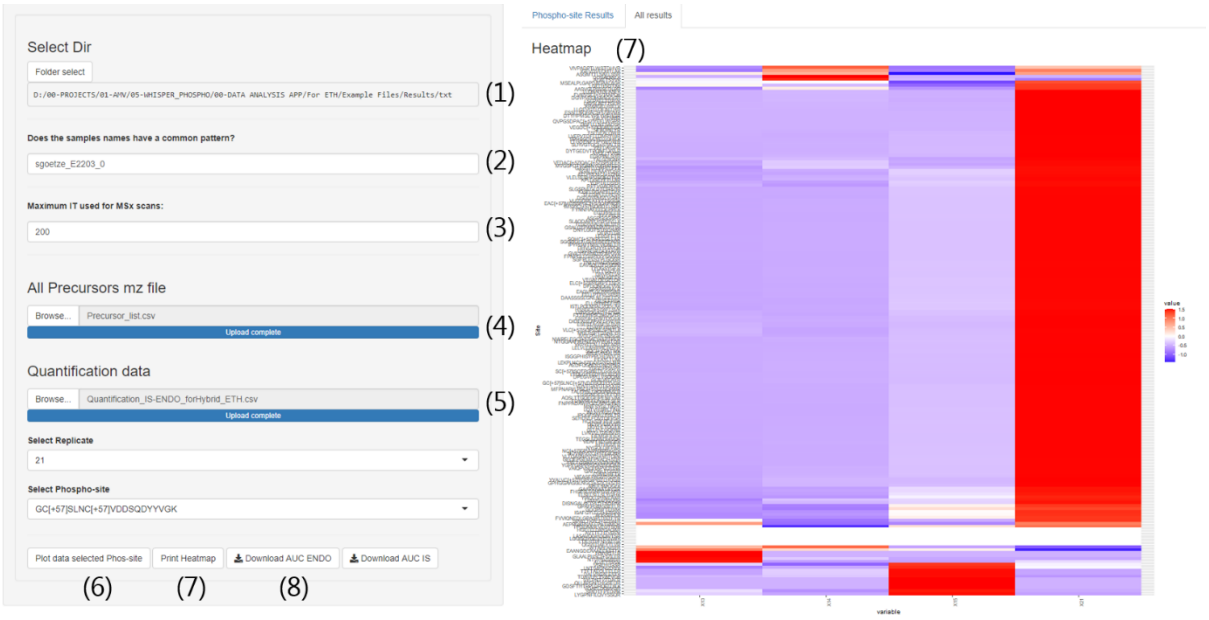

Figure 6. Shiny App Interface. On the right panel: Heatmap with the relative quantification across samples of all peptides monitored in the loaded hybrid DIA experiment.

[1]For help installing python packages: https://packaging.python.org/en/latest/tutorials/installing-packages/#id18