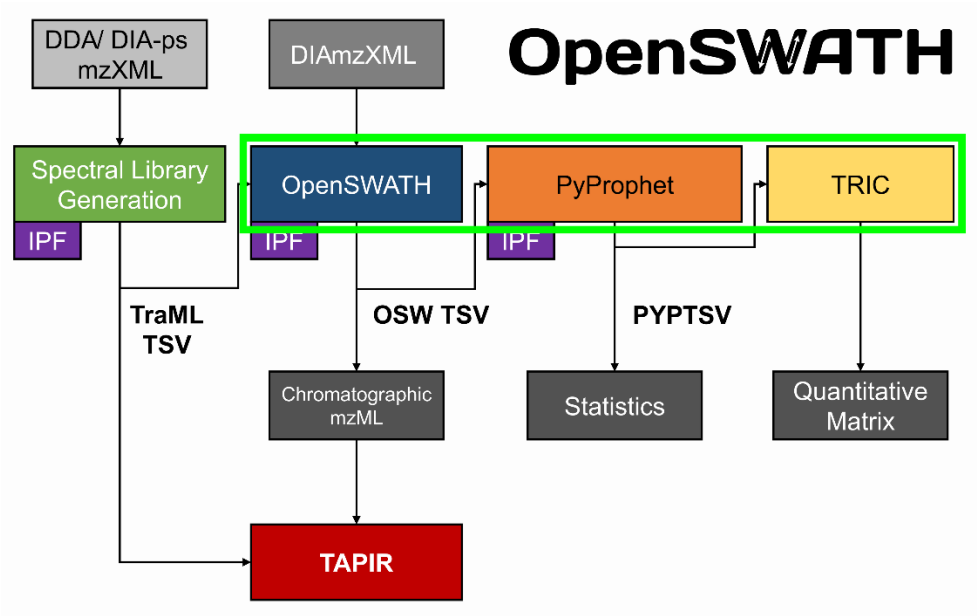
Tutorial 4: DIA data processing using OpenSWATH

1. Introduction

As you can see in the following figure (adapted from the OpenSWATH wiki at http://www.openswath.org), the OpenSWATH workflow comprises three major components namely OpenSWATH [1], PyProphet [2] and TRIC [3].



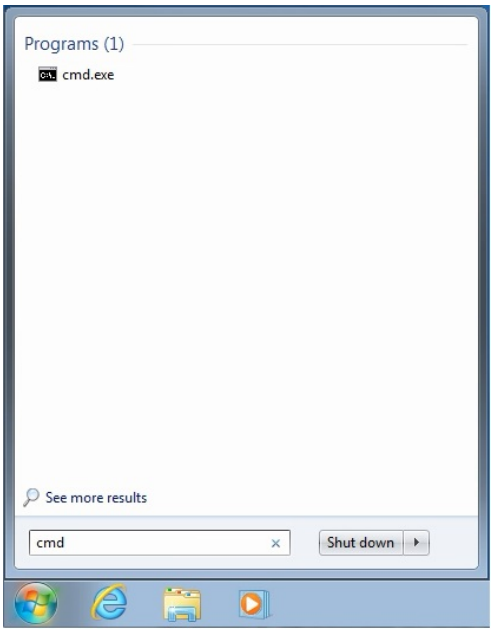
1) **OpenSWATH**
OpenSWATH as a part of OpenMS enables LC-MS/MS DIA data analysis. This proteomic software has been implemented based on a targeted peptide centric approach to assign peak groups according to its prior knowledge, spectral library, generated in tutorial 1 and 3. Please find more detailed information about OpenSWATH at http://www.openswath.org/en/latest/docs/openswath.html

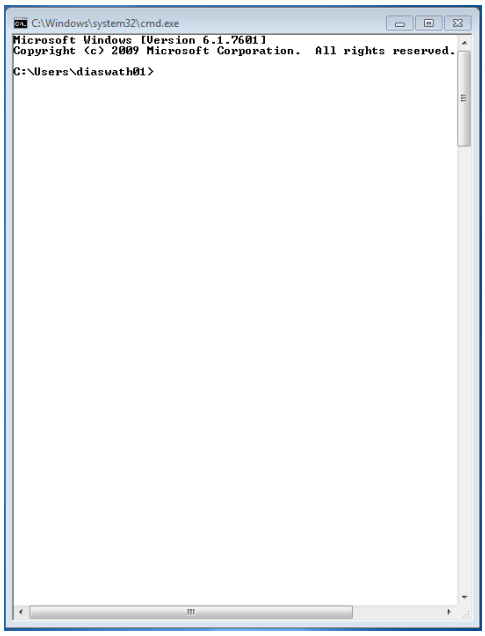
2) **PyProphet**
PyProphet is a python-based and optimized reimplementation of the mProphet algorithm, which can analyze large scale datasets generated by OpenSWATH statistically. The conventional version of PyProphet is NOT recommended for very large sample size due to the accumulation of the false targets across samples. To overcome the issue, a new version has been developed, jumbo PyProphet (JPP), which can control the error rate over thousands of samples and conditions globally. In this tutorial, we would focus on the conventional version but the instruction for JPP is provided in appendix 1. Please find more detailed information about PyProphet at http://www.openswath.org/en/latest/docs/pyprophet.html

3) **TRIC**
TRIC is an alignment software included in msproteomicstools, a python package, to integrate various information of each run with a graph-based method. Please find more detailed information about TRIC at http://www.openswath.org/en/latest/docs/tric.html

2. Run OpenSWATH to assign peak groups

- Start by opening the Windows terminal namely *cmd.exe*
  - **Note!** Just to remind you from the previous tutorials. Please click on the start button and search for “cmd.exe”





- To run OpenSWATH, we need to convert the raw files (.wiff) to the mzXML format in either profiled or centroid mode. As this step has been explained in the tutorial overview, we here would use the centroid version of the files directly.
- Please change your working place to the following.

```
C:

cd \DIA_Course\Tutorial4_OpenSWATH\
```

- Create a folder for the intermediate files. Here with the “FOR” loop, we can create six folders attributed to the runs once. Alternatively, one can creates all the six folders one by one.

```
mkdir TMPDIR

FOR %f IN (\DIA_Course\Data\DIA_data\*) DO (mkdir
"\DIA_Course\Tutorial4_OpenSWATH\TMPDIR\%~nf")
```

- Run the following command to analyze each run separately. The parameters have been explained below. Again here we will use a “FOR” loop to be able to analyze all the runs one after each other. Obviously, you can run the main part of the command for each run separately. Then you need to repeat the very same command six times and each time change the input and output file. The expression %f contains the complete file path plus file name (e.g. \DIA_Course\Data\DIA_data\ lgillet_I150211_008_cent_thresh2.mzXML). When using the expression %~nf this only extracts the file name from the file path saved in %f, in this example case lgillet_I150211_008_cent_thresh2.

- **Note!** Please run this command before reading about the parameters as it takes quite a bit of time to be done.

```
FOR %f IN (\DIA_Course\Data\DIA_data\*.mzXML) DO (OpenSwathWorkflow
-in %f -out_tsv %~nf_OSW.tsv -tr
\DIA_Course\Tutorial1_Library\SpecLib_cons_openswath_decoy.TraML -
tr_irt hroest_DIA_iRT.TraML -tempDirectory .\TMPDIR\%~nf\ -
readOptions cache -batchSize 1000 -threads 8 -mz_extraction_window
30 -ppm -min_upper_edge_dist 1 -extra_rt_extraction_window 100)
```

- In the command above we use the spectral library that was generated from the DDA files. Alternatively, you can also use the library generated with using DIA-Umpire (Tutorial 3). The alternative command is below – it is exactly the same but with an alternative path to the spectral library:

```
FOR %f IN (\DIA_Course\Data\DIA_data\*.mzXML) DO (OpenSwathWorkflow
-in %f -out_tsv %~nf_OSW.tsv -tr \DIA_Course\Tutorial3_DIAUmpire
\SpecLib_cons_openswath_decoy.TraML -tr_irt hroest_DIA_iRT.TraML -
tempDirectory .\TMPDIR\%~nf\ -readOptions cache -batchSize 1000 -
threads 8 -mz_extraction_window 30 -ppm -min_upper_edge_dist 1 -
extra_rt_extraction_window 100)
```

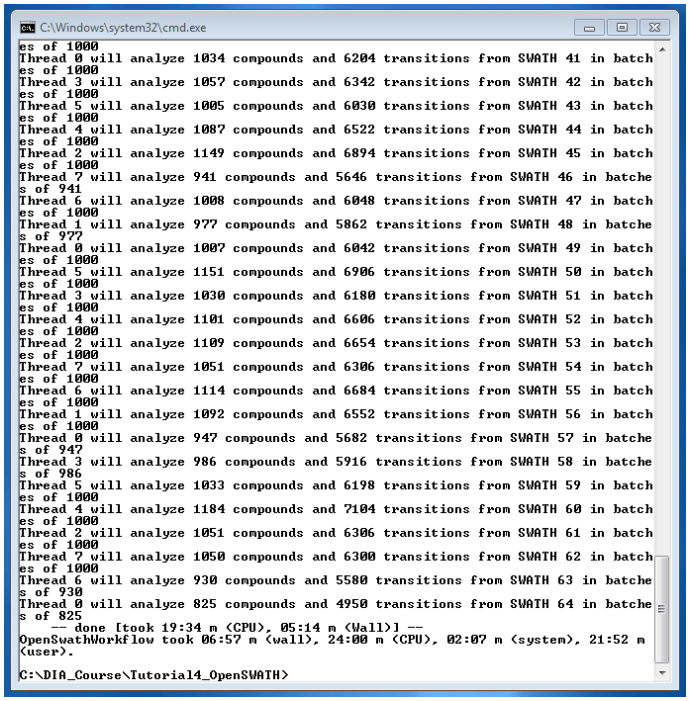
- **Note!** You will see some warnings in the terminal about 'electrospray ionization' and 'ill formed absolute of relative sourceFile'. You can ignore these.

Parameters and Options	Description
-in	Input file (valid format: mzXML, mzML)
-out_tsv	Output file (valid format: tsv)
-tempDirectory	To specify where the temporary files should be saved
-tr	An assay library containing mass spectrometric and chromatographic coordinates for peptides. This file has been generated during tutorial 1 and 3 (valid format: TraML, tsv, csv)
-readOption	Whether to run OpenSWATH directly on the input data, cache data to disk first or to perform a datareduction step first. If you choose cache, make sure to also set tempDirectory (default: 'normal' valid: 'normal', 'cache', 'cacheWorkingInMemory')
-batchSize	The batch size of chromatograms to process (0 means to only have one batch, sensible values are around 500-1000) (default: '0' min: '0')
-min_rsq	Minimum r-squared of RT peptides regression (default: '0.95')

-min_coverage	Minimum relative amount of RT peptides to keep (default: '0.6')
-Scoring:Scores:use_dia_scores	Use the DIA (SWATH) scores (default: 'true' valid: 'true', 'false')
-mz_extraction_window	Extraction window used (in Thomson, to use ppm see -ppm flag) (default: '0.05' min: '0')
-ppm	M/Z extraction_windows is in ppm
-threads	Set the number of threads allowed to be used by the TOPP tool
-min_upper_edge_dist	Minimal distance to the edge to still consider a precursor, in Thomson (default: '0')
-tr_irt	Transition file for the iRT peptides (valid format: TraML)
-extra_rt_extraction_window	Output an XIC with a RT-window that by this much larger (e.g. to visually inspect a larger area of the chromatogram) (default: '0' min: '0')
-rt_extraction_window	Only extract RT around this value (-1 means extract over the whole range, a value of 600 means to extract around +/- 300 s of the expected elution) . (default: '600')

- **Note!** You can find all the parameters by running the following commands. `OpenSwathWorkflow -help` or `OpenSwathWorkflow --helphelp` for the full list

- You can already see that OpenSWATH is running on your console. You don't necessarily need to know what those reports mean. However, they would be very helpful for any potential troubleshooting. Here you can see a screenshot of a successful OpenSWATH DIA data extraction on mzXML files.



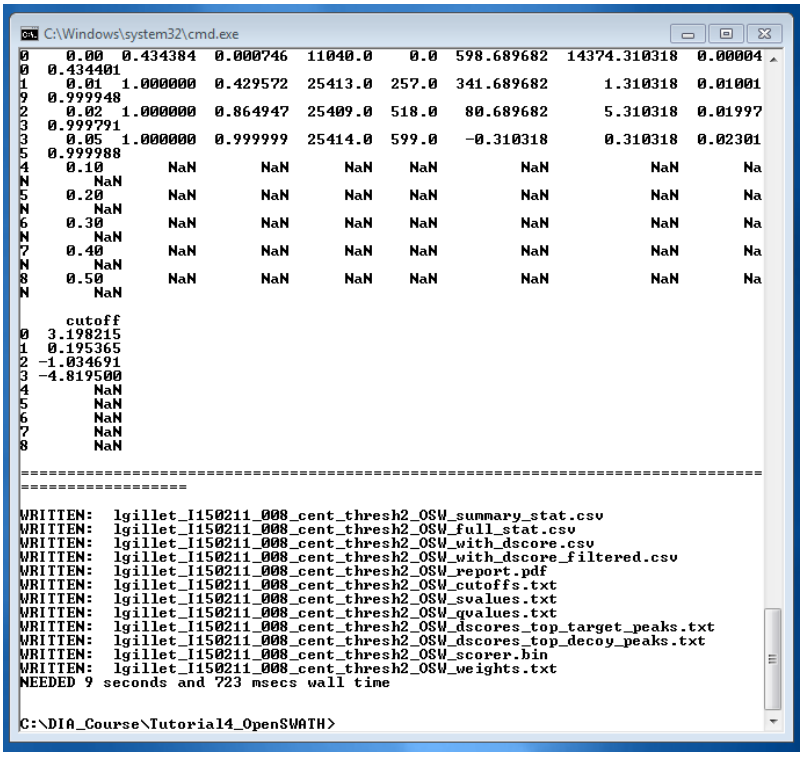
3. PyProphet to analyze the assigned peaks statistically

- Once OpenSWATH is finished you can run PyProphet on each output file

```
FOR %f IN ("*_OSW.tsv") DO (pyprophet --d_score.cutoff="1" --
ignore.invalid_score_columns %f)
```

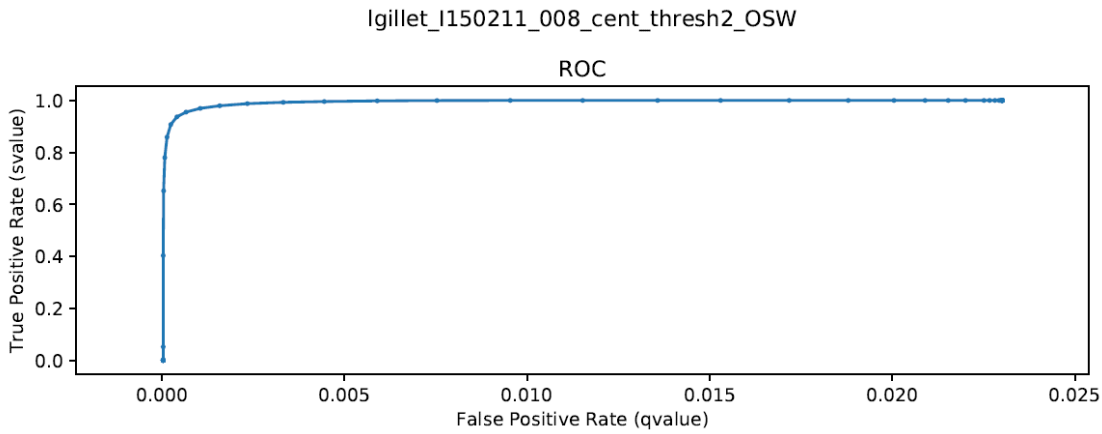
Parameters and Options	Description
--d_score.cutoff=	We can filter out the data based on d score which results in higher performance in terms of speed.
--ignore.invalid_score_columns	Ignore score columns which contain NaN or infinity values

- **Note!** You can find all the parameters by running the following command. `C:\Python27\Scripts\pyprophet.exe –help`
- Here, there is a screenshot of a successful PyProphet run on the OpenSWATH output files.

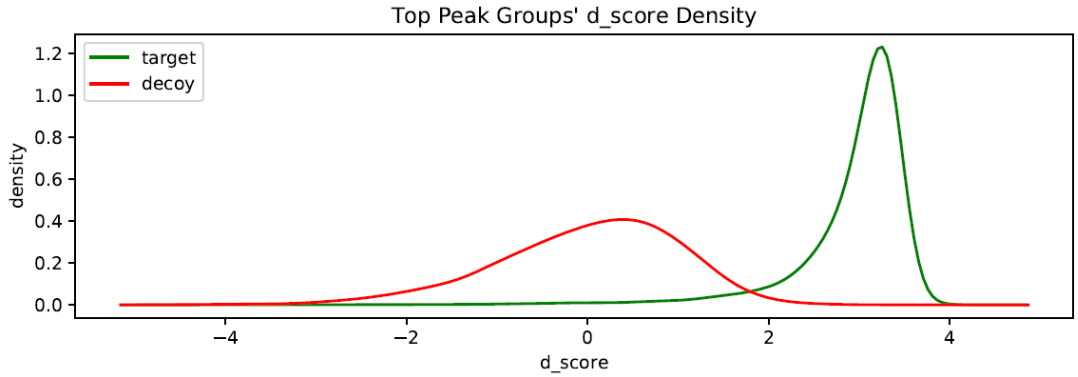


- One of the most important generated file is the pdf report file (_report.pdf) where you can get an overview about each of your file. According to the ROC curve plot, you can realize the sensitivity of the method depicted on the Y axis as a function of false positive rate (1-specificity) on the X axis. Please find more information about ROC curves at https://en.wikipedia.org/wiki/Receiver_operating_characteristic

- **Question!** How can you interpret the plot if the curve was close to the x=y line?



- You can also look at the density plot where you can find how well targets and decoys are separated from each other.



4. TRIC to integrate runs based on various types of data

- TRIC can be run on either "_with_dscore_filtered.csv" or "_with_dscore.csv" files. Please note that if you don't specify any d_score cutoff when running PyProphet in part 3, these two files will be exactly the same. For any large sample size, we recommend to run TRIC on the filtered version.

```
feature_alignment.py --in
"lgillet_I150211_008_cent_thresh2_OSW_with_dscore_filtered.csv"
"lgillet_I150211_009_cent_thresh2_OSW_with_dscore_filtered.csv"
"lgillet_I150211_010_cent_thresh2_OSW_with_dscore_filtered.csv"
"lgillet_I150211_011_cent_thresh2_OSW_with_dscore_filtered.csv"
"lgillet_I150211_012_cent_thresh2_OSW_with_dscore_filtered.csv"
"lgillet_I150211_013_cent_thresh2_OSW_with_dscore_filtered.csv" --
out feature_alignment_pyProphet_TRIC_lowess_localMST.tsv --
file_format openswath --method LocalMST --max_rt_diff 30 --
target_fdr 0.01 --max_fdr_quality 0.05 --mst:useRTCorrection True --
mst:Stdev_multiplier 3.0 --alignment_score 0.0001 --realign_method
lowess --matrix_output_method full  --dscore_cutoff 1.0 --
frac_selected 0 --disable_isotopic_grouping --out_matrix
feature_alignment_pyProphet_TRIC_lowess_localMST_outmatrix.tsv --
out_meta feature_alignment_pyProphet_TRIC_lowess_localMST_meta.tsv
```

Parameters and Options	Description
--in	Input files (valid format: tsv)
--out	Output file
--file_format	Input file format (openswath, mprophet, peakview)
--method	Method to use for the clustering (best_overall, best_cluster_score, global_best_cluster_score, global_best_overall, LocalMST, LocalMSTALLCluster)
--max_rt_diff	Maximum difference in RT for two aligned features
--target_fdr	If parameter estimation is used, which target FDR should be optimized for. If set to lower than 0, parameter estimation is turned off.
--max_fdr_quality	Extension m-score score cutoff, peakgroup of this quality will still be considered for alignment during extension
--mst:useRTCorrection	Use aligned peakgroup RT to continue threading
--mst:Stdev_multiplier	How many standard deviations the peakgroup can deviate in RT during the alignment.
--alignment_score	Minimal score needed for a feature to be considered for alignment between runs
--realign_method	RT alignment method (diRT, linear, splineR, splineR_external, splinePy, lowess, nonCVSpline, CVSpline, Earth)
--matrix_output_method	Which columns are written besides Intensity (none, RT, score, source, full)
--dscore_cutoff	Discard all peakgroups below this d-score
--frac_selected	Do not write peakgroup if selected in less than this fraction of runs (range 0 to 1)
--disable_isotopic_grouping	Disable grouping of isotopic variants by peptide group label
--out_matrix	Matrix containing one peakgroup per row (.csv, .tsv, .xlsx)
--out_meta	Outfile containing meta information

- **Note!** You can find all the parameters by running the following command.
  `c:\Python27\Scripts\feature_alignment.py -help`

- **Question!** Once the TRIC alignment is done, please open the output file (*_localMST.tsv) via Excel and count the number of protein IDs for each organism.
- **Note!** You already have done a similar assignment in the first tutorial.

References:

[1] Röst HL, Rosenberger G, Navarro P, Gillet L, Miladinović SM, Schubert OT, Wolski W, Collins BC, Malmström J, Malmström L, et al.: **OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data [Internet]**. *Nat. Biotechnol.* 2014, **32**:219–223.

[2] Reiter L, Rinner O, Picotti P, Hüttenhain R, Beck M, Brusniak M-Y, Hengartner MO, Aebersold R: **mProphet: automated data processing and statistical validation for large-scale SRM experiments [Internet]**. *Nat. Methods* 2011, **8**:430–435.

[3] Röst HL, Liu Y, D'Agostino G, Zanella M, Navarro P, Rosenberger G, Collins BC, Gillet L, Testa G, Malmström L, et al.: **TRIC: an automated alignment strategy for reproducible protein quantification in targeted proteomics [Internet]**. *Nat. Methods* 2016, **13**:777–783.

Appendix

As it was mentioned in the introductory part, we highly recommend to use the new version of PyProphet so-called Jumbo-PyProphet (JPP) for analysis of any large sample size, or when using very large or organism-scale spectral libraries where many target peptides in the library will not be detectable in the sample. There are two major difference between PyProphet and Jumbo-PyProphet. First, the model generated by JPP using a semi-supervised algorithm is only made once based on all the runs and not for each run separately. Second, the generated model can be applied to the files with three different statistical modes. One of them is called "global" where the error rate can be controlled globally rather than file specific. Hence, we would analyze the data with either run-specific or experiment-wide context (depends on the sample type you want to analyze) and then the results will be filtered based on the results of the global analysis. Please find more detailed information about JPP at http://www.openswath.org/en/latest/docs/pyprophet.html

To run JPP through the windows terminal, *cmd.exe*, please follow the steps below.

- Prepare data

```
pyprophet-cli.exe prepare --data-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\ --work-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\TMPDIR\ --data-filename-
pattern=*_OSW.tsv --separator=tab --extra-group-column=ProteinName
```

- Subsample

```
FOR /l %i in (1,1,6) DO (pyprophet-cli.exe subsample --data-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\ --data-filename-
pattern=*_OSW.tsv --work-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\TMPDIR\ --separator=tab --
job-number %i --job-count 6 --sample-factor=0.17)
```

- Semi-supervised learning

```
pyprophet-cli.exe learn --work-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\TMPDIR\ --separator=tab --
ignore-invalid-scores
```

- Scoring

```
FOR /l %i in (1,1,6) DO (pyprophet-cli.exe apply_weights --data-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\ --data-filename-
pattern=*_OSW.tsv --work-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\TMPDIR\ --separator=tab --
job-number %i --job-count 6)
```

- Statistical validation
    - Run-specific context

```
mkdir JPP

cd JPP

mkdir run_specific
```

```
FOR /l %i in (1,1,6) DO (pyprophet-cli.exe score --data-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\ --data-filename-
pattern=*_OSW.tsv --work-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\TMPDIR\ --result-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\JPP/run_specific\ --
separator=tab --job-number %i --job-count 6 --lambda=0.8 --
statistics-mode=local --overwrite-results)
```

- Experiment-specific context

```
mkdir experiment_wide

FOR /l %i in (1,1,6) DO (pyprophet-cli.exe score --data-
folder=C:/DIA_Course/Tutorial4_OpenSWATH/ --data-filename-
pattern=*_OSW.tsv --work-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\TMPDIR\ --result-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\JPP\experiment_wide\ --
separator=tab --job-number %i --job-count 6 --lambda=0.8 --
statistics-mode=local-global --overwrite-results)
```

- Global context

```
mkdir global

FOR /l %i in (1,1,6) DO (pyprophet-cli.exe score --data-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\ --data-filename-
pattern=*_OSW.tsv --work-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\TMPDIR\ --result-
folder=C:\DIA_Course\Tutorial4_OpenSWATH\JPP\global\ --separator=tab
--job-number %i --job-count 6 --lambda=0.8 --statistics-mode=global
--overwrite-results)
```

- Filtering the either run-specific or experiment-wide results based on the global analysis.
  - **Caution!** A customized python script, JPP2TRIC.py, has been provided to filter the data. It makes a global protein list according to the global analysis and desired false discovery rate (FDR) you have supplied. Thereafter, your either run-specific or experiment-wide results will be filtered accordingly. Please follow the steps below to filter your data.

  - Please copy JPP2TRIC.py into your JPP folder. This can be found on the cloud storage (link on your VM desktop or USB stick called DIA_Course) in the 'Tutorial4_OpenSWATH' directory or on your USB stick in the same directory.

  - Now you can run the script with the following command.

```
c:\Python27\python.exe JPP2TRIC.py
```

  - You are asked for the full path of your global directory. Please enter the following.

```
c:\DIA_Course\Tutorial4_OpenSWATH\JPP\global
```

  - Now you are asked for the either run-specific or experiment-wide folder.

- If you want to filter your run-specific data, please enter the following path

```
c:\DIA_Course\Tutorial4_OpenSWATH\JPP\run_specific
```

- If you want to filter your experiment-wide data, please enter the following path

```
C:\DIA_Course\Tutorial4_OpenSWATH\JPP\experiment_wide
```

- Please enter the protein FDR you are aiming at (e.g. 0.01 for 1%)

```
0.01
```

- Once the script is done, you can find your global protein list in the global directory and your filtered data in your run-specific/experiment-wide folder
- **Caution!** In case you check your final protein lists – do not get confused by the large number of decoys. Remember, decoy counting is NOT valid in this approach as long as you do not correct for the fraction of false targets!
- **Note!** Please continue with the TRIC alignment where you need to specify your filtered files generated here by JPP as TRIC's input files

We would like to thank SystemsX for supporting the Zurich DIA / SWATH Course 2017.