OpenSWATH Documentation

Release 0.1

OpenSWATH Developers

Sep 27, 2017

Spectral library generation

The OpenSWATH Workflow enables targeted data analysis of data-independent acquisition (DIA) or SWATH-MS proteomic data. The main workflow consists of OpenSWATH, PyProphet, TRIC, IPF and TAPIR. This website provides documentation on installation and application of the tools.

CHAPTER 1

Acknowledgments

The tools and workflows are being developed at the Aebersold Group at IMSB, ETH Zurich and Stanford University with contributions from others. The core components are implemented as part of the OpenMS framework, the PyProphet, and msproteomicstools distributions.

Trans-Proteomic Pipeline

Overview

A tutorial[1] describes individual steps to generate spectral libraries for SWATH-MS using the Trans-Proteomic Pipeline (TPP).

Contact and Support

We provide support separately for the TPP, the msproteomicstools and the OpenMS components of the workflow.

References

SWATHAtlas

Overview

SWATHAtlas is a repository providing spectral libraries generated from endogeneous samples and synthetic peptides. The libraries are preformatted for several commonly employed targeted data analysis tools, e.g. OpenSWATH, Skyline, Spectronaut and PeakView.

[1] Schubert OT, Gillet LC, Collins BC, Navarro P, Rosenberger G, Wolski WE, Lam H, Amodei D, Mallick P, MacLean B, Aebersold R. Building high-quality assay libraries for targeted analysis of SWATH MS data. Nat Protoc. 2015 Mar;10(3):426-41. doi: 10.1038/nprot.2015.015. Epub 2015 Feb 12. PMID: 25675208

Contact and Support

The libraries are processed by different tools and we thus provide support via different channels. If you encounter problems with obtaining the libraries, please contact the SWATHAtlas team, e.g. via the TPP Support Group.

For support regarding `spectrast2tsv.py`, please use the msproteomicstools issue tracker or consider the support channels for the OpenMS components of the workflow.

OpenSWATH

Overview

OpenSWATH[1] is a proteomics software that allows analysis of LC-MS/MS DIA (data independent acquisition) data using the approach described by Gillet et al.[2] and implemented as part of OpenMS[3]. The original SWATH-MS method uses 32 cycles to iterate through precursor ion windows from 400-426 Da to 1175-1201 Da and at each step acquire a complete, multiplexed fragment ion spectrum of all precursors present in that window. After 32 fragmentations (or 3.2 seconds), the cycle is restarted and the first window (400-426 Da) is fragmented again, thus delivering complete "snapshots" of all fragments of a specific window every 3.2 seconds.

The analysis approach described by Gillet et al. extracts ion traces of specific fragment ions from all MS2 spectra that have the same precursor isolation window, thus generating data that is very similar to SRM traces.

Contact and Support

We provide support for OpenSWATH using the OpenMS support channels. Please address general questions to the open-ms-general mailing list.

You can contact the authors Hannes Röst and George Rosenberger.

Installation

OpenSWATH is completely integrated into OpenMS. We suggest to always use the most current development version of OpenMS to get the newest OpenSWATH developments. Alternatively, the OpenMS releases provide installers for all plattforms.

We recommend to compile a nightly snapshot from the OpenMS Git repository on Linux or, if this is not possible, to download the nightly builds for Windows. Installation and compilation instructions can be obtained from the OpenMS documentation.

Tutorial

The core step of OpenSWATH conducts signal processing by data extraction from DIA data. It requires mzXML or mzML files. Since 2017, ProteoWizard natively supports conversion of SCIEX, Thermo and Waters DIA data.

[1] Röst HL, Rosenberger G, Navarro P, Gillet L, Miladinović SM, Schubert OT, Wolski W, Collins BC, Malmström J, Malmström L, Aebersold R. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. Nat Biotechnol. 2014 Mar 10;32(3):219-23. doi: 10.1038/nbt.2841. PMID: 24727770

[2] Gillet LC, Navarro P, Tate S, Röst H, Selevsek N, Reiter L, Bonner R, Aebersold R. Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis. Mol Cell Proteomics. 2012 Jun;11(6):O111.016717. Epub 2012 Jan 18. PMID: 22261725

[3] Röst HL, Sachsenberg T, Aiche S, Bielow C, Weisser H, Aicheler F, Andreotti S, Ehrlich HC, Gutenbrunner P, Kenar E, Liang X, Nahnsen S, Nilse L, Pfeuffer J, Rosenberger G, Rurik M, Schmitt U, Veit J, Walzer M, Wojnar D, Wolski WE, Schilling O, Choudhary JS, Malmström L, Aebersold R, Reinert K, Kohlbacher O. OpenMS: a flexible open-source software platform for mass spectrometry data analysis. Nat Methods. 2016 Aug 30;13(9):741-8. doi: 10.1038/nmeth.3959. PMID: 27575624

The generated mzXML and mzML files are directly compatible with OpenSWATH. Previous workarounds are not necessary anymore and should be avoided.

Four native interfaces exist to use OpenSWATH:

Integrated Executable

Since OpenMS 2.0, the executable `OpenSwathWorkflow` provides a fast and efficient analysis. Most users will want to directly use this workflow.

An extended tutorial describing a complete OpenSWATH analysis workflow using `OpenSwathWorkflow` was recently published[4] and is also available from bioRxiv.

Classic TOPP Workflow

The original "classic" workflow provides different OpenMS TOPP[5] executables that can also be accessed from different supported workflow managers. The OpenMS User Tutorial provides detailed instructions.

Python interface via pyOpenMS

Many OpenMS classes are also available from within Python. pyOpenMS[6] provides access to most functionality with Python data structures. Example scripts for both OpenMS and OpenSWATH can be found in the OpenMS User Tutorial.

Illumina BaseSpace

OpenSWATH is available as native application on Illumina BaseSpace. Together with SCIEX OneOmics it can analyse SWATH-MS data on the Amazon Cloud.

Tutorial Data

Availability

To learn OpenSWATH, we suggest to use the *M. tuberculosis* dataset published alongside the 2017 Methods Mol Biol. OpenSWATH tutorial[4] which is available from the PeptideAtlas raw data repository with accession number PASS00779.

The SWATH-MS Gold Standard and *Streptococcus pyogenes* data sets (used in the original 2014 Nature Biotechnoly publication) are available from the PeptideAtlas raw data repository with accession number PASS00289.

The Skyline results are available from Skyline Panorama Webserver.

[4] Röst HL, Aebersold R, Schubert OT. Automated SWATH Data Analysis Using Targeted Extraction of Ion Chromatograms. Methods Mol Biol. 2017;1550:289-307. doi: 10.1007/978-1-4939-6747-6_20. PMID: 28188537

[5] Kohlbacher O, Reinert K, Gröpl C, Lange E, Pfeifer N, Schulz-Trieglaff O, Sturm M. TOPP–the OpenMS proteomics pipeline. Bioinformatics. 2007 Jan 15;23(2):e191-7. PMID: 17237091

[6] Röst HL, Schmitt U, Aebersold R, Malmström L. pyOpenMS: a Python-based interface to the OpenMS mass-spectrometry algorithm library. Proteomics. 2014 Jan;14(1):74-7. doi: 10.1002/pmic.201300246. PMID: 24420968

Mycobacterium tuberculosis data

- 3 mzML instrument data files (centroided)
- 3 WIFF raw instrument data files
- Mtb assay library (for OpenMS 2.1)
- Mtb assay library (for older OpenMS)
- Swath windows file for analysis
- iRT assay file (TraML format)

SWATH-MS Gold Standard

- 90 mzXML instrument data files
- 90 WIFF raw instrument data files
- SGS TSV assay library
- SGS TraML assay library
- SGS OpenSWATH results
- SGS Skyline results on Panorama
- SGS manual results

Streptococcus pyogenes

- 4 mzXML instrument data files
- 4 WIFF raw instrument data files
- *S. pyo* TSV assay library
- *S. pyo* TraML assay library
- *S. pyo* OpenSWATH results
- *S. pyo* summary results

References

SQLite-based Workflow

Overview

With the increasing size and number of runs acquired by data-independent acquisition (DIA)-based methods, data analysis algorithms like OpenSWATH[1] are challenged by the size and data formats of the input data. Additionally, re-

[1] Röst HL, Rosenberger G, Navarro P, Gillet L, Miladinović SM, Schubert OT, Wolski W, Collins BC, Malmström J, Malmström L, Aebersold R. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. Nat Biotechnol. 2014 Mar 10;32(3):219-23. doi: 10.1038/nbt.2841. PMID: 24727770

cent extensions like TRIC[2], IPF[3] and error-rate control on different levels and in different contexts[4] produce additional layers of data that could not be ideally represented by the original data structures.

For this reason, we are currently adapting the tools for new SQLite[5] based data formats that represent the data in a non-redundant fashion, require less storage and processing time. The new data formats have been implemented in OpenSWATH and PyProphet and the final results can be exported to the legacy TSV reports.

PQP files represent the data stored in TraML files. OSW files copy the exact data structure of the PQP files and append `feature` tables generated by OpenSWATH. Finally, PyProphet appends `score` tables linked to the `feature` tables. OpenSWATH stores the results of one run in a single OSW file. However, PyProphet can merge OSW files in a non-redundant, non-destructive fashion.

Contact and Support

The new data formats are currently in development and must NOT be used in production environments. We would however be very grateful for testing of the new workflows and reporting of problems and bugs.

You can contact the author George Rosenberger.

Installation

To use the new data formats, please use the following versions of our tools:

OpenMS

Full support for PQP and OSW files is provided in OpenMS/develop, with limited support available since OpenMS 2.2. Please follow the instructions in the *OpenSWATH* tutorial to install OpenMS.

PyProphet

We have developed a new, substantially changed version of PyProphet that integrates the new functionality of both IPF and can conduct error-rate control in different contexts and on different levels. If Python and PIP are configured correctly, the following command can be used to install the development version:

```
pip install git+https://github.com/grosenberger/pyprophet.git@feature/refactoring
```

TRIC

TRIC should be installed according to the *TRIC* installation instructions. The SQLite format is presently not supported, however exporting the legacy format will enable intermediate compatibility.

[2] Röst HL, Liu Y, D'Agostino G, Zanella M, Navarro P, Rosenberger G, Collins BC, Gillet L, Testa G, Malmström L, Aebersold R. TRIC: an automated alignment strategy for reproducible protein quantification in targeted proteomics. Nat Methods. 2016 Sep;13(9):777-83. doi: 10.1038/nmeth.3954. Epub 2016 Aug 1. PMID: 27479329

[3] Rosenberger G, Liu Y, Röst HL, Ludwig C, Buil A, Bensimon A, Soste M, Spector TD, Dermitzakis ET, Collins BC, Malmström L, Aebersold R. Inference and quantification of peptidoforms in large sample cohorts by SWATH-MS. Nat Biotechnol. 2017 Aug;35(8):781-788. doi: 10.1038/nbt.3908. Epub 2017 Jun 12. PMID: 28604659

[4] Rosenberger G, Bludau I, Schmitt U, Heusel M, Hunter CL, Liu Y, MacCoss MJ, MacLean BX, Nesvizhskii AI, Pedrioli PGA, Reiter L, Röst HL, Tate S, Ting YS, Collins BC, Aebersold R. Statistical control of peptide and protein error rates in large-scale targeted data-independent acquisition analyses. Nat Methods. 2017 Sep;14(9):921-927. doi: 10.1038/nmeth.4398. Epub 2017 Aug 21. PMID: 28825704

[5] http://sqlite.org/

Tutorial

The general workflow is very similar to the original OpenSWATH workflow with a few minor changes:

1. Peptide Query Parameter Generation

Peptide query parameters should be generated identically as described previously, including appended decoys. Optionally, `OpenSwathAssayGenerator` can append peptide query parameters for IPF. The final TraML should then be converted to a peptide query parameter (PQP) file und `TargetedFileConverter` from OpenMS:

```
TargetedFileConvert -in assays_ipf_decoys.TraML -out assays_ipf_decoys.pqp
```

2. Targeted data extraction using OpenSWATH

The next step is conducted using OpenSWATH.

```
OpenSwathWorkflow
-in MSDATA.mzXML.gz \
-tr assays_ipf_decoys.pqp \
-out_osw MSDATA_RESULTS.osw
[OTHER PARAMETERS]
```

The workflow is executed identically as before, with the only change being that the PQP file is used `-tr assays_ipf_decoys.pqp` and an OSW file is exported `-out_osw MSDATA_RESULTS.osw`.

3. Statistical validation using PyProphet

PyProphet is then applied to the OSW files. Importantly, the updated version has changed substantially internally and in terms of the command line interface. Several different commands can be run to consecutively to do the analysis:

```
pyprophet merge --out=merged.osw \
--subsample_ratio=1 *.osw
```

This command will merge and optionally subsample multiple files. If a set of runs should be analyzed in an experiment-wide fashion, we recommend to conduct this step. If semi-supervised learning is too slow, create an additional merged file with a smaller `subsample_ratio`. The model will be stored in the output and can be applied to the full file.

```
pyprophet score --in=merged.osw --level=ms2
```

The main command will conduct semi-supervised learning and error-rate estimation in a fully automated fashion. `--help` will show the full selection of parameters to adjust the process. The default parameters are recommended for SCIEX TripleTOF 5600/6600 instrument data, but can be adjusted in other scenarios. The parameter `--level` can be set to `ms2`, `ms1` or `transition`. If MS1 or transition-level data should be scored, the command is executed three times, e.g.:

```
pyprophet score --in=merged.osw --level=ms1 \
score --in=merged.osw --level=ms2 \
score --in=merged.osw --level=transition
```

Importantly, PyProphet will store all results in the input OSW files. This can be changed by specifying `--out`. However, since all steps are non-destructive, this is not necessary.

If IPF should be applied after scoring, the following command can be used:

```
pyprophet ipf --in=merged.osw
```

To adjust the IPF-specific parameters, please consult `pyprophet ipf --help`.

To conduct peptide inference in run-specific, experiment-wide and global contexts, the following command can be applied:

```
pyprophet peptide --in=merged.osw --context=run-specific \
peptide --in=merged.osw --context=run-specific \
peptide --in=merged.osw --context=global
```

This will generate individual PDF reports and store the scores in a non-redundant fashion in the OSW file.

Analogously, this can be conducted on protein-level as well:

```
pyprophet protein --in=merged.osw --context=run-specific \
protein --in=merged.osw --context=run-specific \
protein --in=merged.osw --context=global
```

Finally, we can export the results to legacy OpenSWATH TSV report:

```
pyprophet export --in=merged.osw --out=legacy.tsv \
```

By default, IPF results will be used. This can be disabled by setting `--no-ipf`.

References

PyProphet

Overview

PyProphet[1] is a reimplementation of the mProphet[2] algorithm for targeted proteomics. It is particularly optimized for analysis of large scale data sets generated by OpenSWATH or DIANA.

Contact and Support

We provide support for PyProphet on the GitHub repository.

You can contact the authors Uwe Schmitt, Johan Teleman, Hannes Röst and George Rosenberger.

Installation

PyProphet is currently available in two flavors:

[1] Teleman J, Röst HL, Rosenberger G, Schmitt U, Malmström L, Malmström J, Levander F. DIANA–algorithmic improvements for analysis of data-independent acquisition MS data. Bioinformatics. 2015 Feb 15;31(4):555-62. doi: 10.1093/bioinformatics/btu686. Epub 2014 Oct 27. PMID: 25348213

[2] Reiter L, Rinner O, Picotti P, Hüttenhain R, Beck M, Brusniak MY, Hengartner MO, Aebersold R. mProphet: automated data processing and statistical validation for large-scale SRM experiments. Nat Methods. 2011 May;8(5):430-5. doi: 10.1038/nmeth.1584. Epub 2011 Mar 20. PMID: 21423193

PyProphet

PyProphet is the main Python package. It is also available from PyPI.

Currently PyProphet requires Python 2.7 and several dependencies. Windows users should install Anaconda, Mac and Linux users should be able to install PyProphet directly from PyPI:

```
pip install pyprophet
```

PyProphet-cli aka Jumbo-PyProphet

To deal with larger data sets and to provide error rate control on the level of peptide sequences and proteins for different contexts (run-specific, experiment-wide and global), an extension of PyProphet is in development[4]. It is optimized to analyse hundreds of runs simultaneously and builds on IBM LSF or OpenLava workflow managers, but the steps can also be executed independently. It can be installed from PyPI:

```
pip install pyprophet
pip install pyprophet-cli
pip install pyprophet-brutus
```

PyProphet-cli can be adapted to other workflow managers by development of lightweight modules replacing `pyprophet-brutus`.

Tutorial

PyProphet

An extended tutorial describing a complete OpenSWATH analysis workflow including PyProphet was recently published[3] and is also available from bioRxiv.

PyProphet-cli aka Jumbo-PyProphet

If the three modules have been properly configured, PyProphet jobs can be submitted using the following command:

```
pyprophet-cli run_on_brutus \
--data-folder="/tmp/openswath_results/" \
--data-filename-pattern="openswath_output_*.tsv" --sample-factor=0.1 --job-count=10 \
--extra-args-prepare --extra-group-column=ProteinName \
--extra-args-score --lambda=0.8
```

The example works as following:

- –data-folder: /tmp/openswath_results/ contains 10 files, openswath_output_0.tsv - openswath_output_9.tsv.
- –data-filename-pattern: This regular expression is used to grab the correct files.
- –sample-factor: This value can be anything from 0 - 1. We recommend to use 1/(#runs), here 1/10=0.1.
- –job-count: Specifies the number of parallel jobs to submit.
- –extra-args-prepare –extra-group-column=ProteinName: Also compute protein-level q-values

[4] Rosenberger G, Bludau I, Schmitt U, Heusel M, Hunter CL, Liu Y, MacCoss MJ, MacLean BX, Nesvizhskii AI, Pedrioli PGA, Reiter L, Röst HL, Tate S, Ting YS, Collins BC, Aebersold R. Statistical control of peptide and protein error rates in large-scale targeted data-independent acquisition analyses. Nat Methods. 2017 Sep;14(9):921-927. doi: 10.1038/nmeth.4398. Epub 2017 Aug 21. PMID: 28825704

[3] Röst HL, Aebersold R, Schubert OT. Automated SWATH Data Analysis Using Targeted Extraction of Ion Chromatograms. Methods Mol Biol. 2017;1550:289-307. doi: 10.1007/978-1-4939-6747-6_20. PMID: 28188537

- –extra-args-score –lambda=0.8: Set lambda to 0.8 for q-value estimation.

There are further parameters that can be set, please refer to:

```
pyprophet-cli --help
```

Alternatively, if pyprophet-brutus-driver is not available or for integration with other workflow managers, it is also possible to execute all steps independently. In the following example, 3 example runs are used:

1. Prepare data

```
pyprophet-cli prepare --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
--work-folder=/tmp/pyprophet_work/ --separator="tab" --extra-group-column="ProteinName
↪"
```

2. Subsample

```
pyprophet-cli subsample --data-folder="/tmp/openswath_results/" --data-filename-
↪pattern="*.tsv" \
--work-folder="/tmp/pyprophet_work/" --separator="tab" --job-number 1 --job-count 3 --
↪sample-factor=0.4 &
pyprophet-cli subsample --data-folder="/tmp/openswath_results/" --data-filename-
↪pattern="*.tsv" \
--work-folder="/tmp/pyprophet_work/" --separator="tab" --job-number 2 --job-count 3 --
↪sample-factor=0.4 &
pyprophet-cli subsample --data-folder="/tmp/openswath_results/" --data-filename-
↪pattern="*.tsv" \
--work-folder="/tmp/pyprophet_work/" --separator="tab" --job-number 3 --job-count 3 --
↪sample-factor=0.4 &
```

3. Semi-supervised learning

```
pyprophet-cli learn --work-folder="/tmp/pyprophet_work/" --separator="tab" --ignore-
↪invalid-scores
```

4. Scoring

```
pyprophet-cli apply_weights --data-folder="/tmp/openswath_results/" --data-filename-
↪pattern="*.tsv" \
--work-folder="/tmp/pyprophet_work/" --separator="tab" --job-number 1 --job-count 3 &
pyprophet-cli apply_weights --data-folder="/tmp/openswath_results/" --data-filename-
↪pattern="*.tsv" \
--work-folder="/tmp/pyprophet_work/" --separator="tab" --job-number 2 --job-count 3 &
pyprophet-cli apply_weights --data-folder="/tmp/openswath_results/" --data-filename-
↪pattern="*.tsv" \
--work-folder="/tmp/pyprophet_work/" --separator="tab" --job-number 3 --job-count 3 &
```

5. Statistical validation

- Run-specific context

```
pyprophet-cli score --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
--work-folder="/tmp/pyprophet_work/" --result-folder="/tmp/pyprophet_result_run_
↪specific" --separator="tab" \
--job-number 1 --job-count 3 --lambda=0.4 --statistics-mode=run-specific --overwrite-
↪results &
pyprophet-cli score --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
```

```
--work-folder="/tmp/pyprophet_work/" --result-folder="/tmp/pyprophet_result_run_
↪specific" --separator="tab" \
--job-number 2 --job-count 3 --lambda=0.4 --statistics-mode=run-specific --overwrite-
↪results &
pyprophet-cli score --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
--work-folder="/tmp/pyprophet_work/" --result-folder="/tmp/pyprophet_result_run_
↪specific" --separator="tab" \
--job-number 3 --job-count 3 --lambda=0.4 --statistics-mode=run-specific --overwrite-
↪results &
```

- Experiment-wide context

```
pyprophet-cli score --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
--work-folder="/tmp/pyprophet_work/" --result-folder="/tmp/pyprophet_result_
↪experiment_wide" --separator="tab" \
--job-number 1 --job-count 3 --lambda=0.4 --statistics-mode=experiment-wide &
pyprophet-cli score --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
--work-folder="/tmp/pyprophet_work/" --result-folder="/tmp/pyprophet_result_
↪experiment_wide" --separator="tab" \
--job-number 2 --job-count 3 --lambda=0.4 --statistics-mode=experiment-wide &
pyprophet-cli score --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
--work-folder="/tmp/pyprophet_work/" --result-folder="/tmp/pyprophet_result_
↪experiment_wide" --separator="tab" \
--job-number 3 --job-count 3 --lambda=0.4 --statistics-mode=experiment-wide &
```

- Global context

```
pyprophet-cli score --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
--work-folder="/tmp/pyprophet_work/" --result-folder="/tmp/pyprophet_result_global" --
↪separator="tab" \
--job-number 1 --job-count 3 --lambda=0.4 --statistics-mode=global &
pyprophet-cli score --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
--work-folder="/tmp/pyprophet_work/" --result-folder="/tmp/pyprophet_result_global" --
↪separator="tab" \
--job-number 2 --job-count 3 --lambda=0.4 --statistics-mode=global --overwrite-
↪results &
pyprophet-cli score --data-folder="/tmp/openswath_results/" --data-filename-pattern=
↪"*.tsv" \
--work-folder="/tmp/pyprophet_work/" --result-folder="/tmp/pyprophet_result_global" --
↪separator="tab" \
--job-number 3 --job-count 3 --lambda=0.4 --statistics-mode=global --overwrite-
↪results &
```

References

TRIC

Overview

TRIC[1] is an alignment software for targeted proteomics (SRM or SWATH-MS) data. TRIC uses a graph-based alignment strategy based on non-linear retention time correction to integrate information from all available runs. The input consists of a set of csv files derived from a targeted proteomics experiment generated by OpenSWATH (using either mProphet or PyProphet) or generated by Peakview.

There are two basic running modes available. The first one uses a reference-based alignment where a single run is chosen as a reference and all other runs are aligned to it. This is a useful choice for a small number of runs that are chromatographically similar. The second mode generates a guidance tree based on chromatographic similarity of the input runs and uses this tree to align the targeted proteomics runs (the nodes in the tree are runs and the edges are pairwise alignments). Generally this mode is better for a large number of runs or for chromatographically dissimilar samples.

Contact and Support

We provide support for TRIC on the GitHub repository.

You can contact the author Hannes Röst.

Installation

TRIC requires Python 2.7 and can be installed through `pip`. On Microsoft Windows you will first have to install Python (the easiest way to do this is to download Anaconda). You can then install TRIC through PyPI:

```
pip install numpy
pip install msproteomicstools
```

This will install TRIC.py which you can then execute. You can also download the TRIC release directly from PyPI. To obtain the latest development version, please download the code from GitHub. If you are using Microsoft Windows and Anaconda, it is possible that BioPython does not properly install and you may have to install it through Anaconda:

```
conda install biopython
```

Tutorial

After installing TRIC, please familiarize yourself with the TRIC Tutorial. All command line parameters and their effects are explained in the tutorial and the associated tutorial paper (Röst et al). Currently, the recommended parameters for TRIC are:

```
feature_alignment.py
--in file1_input.csv file2_input.csv file3_input.csv
--out aligned.csv
--method LocalMST --realign_method lowess_cython --max_rt_diff 60
```

[1] Röst HL, Liu Y, D'Agostino G, Zanella M, Navarro P, Rosenberger G, Collins BC, Gillet L, Testa G, Malmström L, Aebersold R. TRIC: an automated alignment strategy for reproducible protein quantification in targeted proteomics. Nat Methods. 2016 Sep;13(9):777-83. doi: 10.1038/nmeth.3954. Epub 2016 Aug 1. PMID: 27479329

```
--mst:useRTCorrection True --mst:Stdev_multiplier 3.0
--target_fdr 0.01 --max_fdr_quality 0.05
```

An extended tutorial describing a complete OpenSWATH analysis workflow including TRIC was recently published[2] and is also available from bioRxiv.

Data

Availability

The TRIC Gold Standard, the *Streptococcus pyogenes* data sets and the iPSC datasets are available from the PeptideAtlas raw data repository with accession number PASS00788.

The Skyline results are available from the same repository where a .sky and .sky.view file are provided.

TRIC Gold Standard

The TRIC Gold Standard dataset contains a set of manually validated aligned peptides and can be found in the `./ManualValidation` folder on the FTP server.

- 16 WIFF raw instrument data files
- 1 Skyline file with manually picked data
- 1 CSV file with the manually picked peaks (Skyline export)
- The TRIC results
- Python script used to compare manual with TRIC data

Streptococcus pyogenes

- 16 WIFF raw instrument data files
- 1 Assay library in TraML and CSV format
- 1 iRT library in TraML and CSV format (use instead of default iRT)
- 16 OpenSWATH output files (results/openswath)
- 1 TRIC output file using local MST parameters as described in the paper
- 1 unaligned output matrix (noalign_all_1pcnt.csv)

human iPSC

- 8 WIFF raw instrument data files
- 1 Assay library in TraML and CSV format
- OpenSWATH output files
- 1 TRIC output file using local MST parameters as described in the paper

[2] Röst HL, Aebersold R, Schubert OT. Automated SWATH Data Analysis Using Targeted Extraction of Ion Chromatograms. Methods Mol Biol. 2017;1550:289-307. doi: 10.1007/978-1-4939-6747-6_20. PMID: 28188537

References

IPF

Overview

IPF (Inference of PeptidoForms)[1] is an extension to the OpenSWATH[2] workflow to increase the specificity of the analysis to the level of peptidoforms (modified peptides with specific site-localization) across multiple runs. IPF is fully implemented as part of OpenMS[3] and PyProphet[4] and compatible with the downstream alignment algorithm TRIC[5].

Contact and Support

We provide support for IPF using the OpenMS support channels. Please address general questions to the open-ms-general mailing list.

You can contact the author George Rosenberger.

Installation

IPF requires the installation of several tools:

OpenMS

IPF is available since OpenMS 2.1. Please follow the instructions in the *OpenSWATH* tutorial to install OpenMS.

Percolator

IPF presently requires QVALITY[6] to be installed, which is part of Percolator. Please install and export the binaries according to the Percolator instructions.

[1] Rosenberger G, Liu Y, Röst HL, Ludwig C, Buil A, Bensimon A, Soste M, Spector TD, Dermitzakis ET, Collins BC, Malmström L, Aebersold R. Inference and quantification of peptidoforms in large sample cohorts by SWATH-MS. Nat Biotechnol. 2017 Aug;35(8):781-788. doi: 10.1038/nbt.3908. Epub 2017 Jun 12. PMID: 28604659

[2] Röst HL, Rosenberger G, Navarro P, Gillet L, Miladinović SM, Schubert OT, Wolski W, Collins BC, Malmström J, Malmström L, Aebersold R. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. Nat Biotechnol. 2014 Mar 10;32(3):219-23. doi: 10.1038/nbt.2841. PMID: 24727770

[3] Röst HL, Sachsenberg T, Aiche S, Bielow C, Weisser H, Aicheler F, Andreotti S, Ehrlich HC, Gutenbrunner P, Kenar E, Liang X, Nahnsen S, Nilse L, Pfeuffer J, Rosenberger G, Rurik M, Schmitt U, Veit J, Walzer M, Wojnar D, Wolski WE, Schilling O, Choudhary JS, Malmström L, Aebersold R, Reinert K, Kohlbacher O. OpenMS: a flexible open-source software platform for mass spectrometry data analysis. Nat Methods. 2016 Aug 30;13(9):741-8. doi: 10.1038/nmeth.3959. PMID: 27575624

[4] Teleman J, Röst HL, Rosenberger G, Schmitt U, Malmström L, Malmström J, Levander F. DIANA–algorithmic improvements for analysis of data-independent acquisition MS data. Bioinformatics. 2015 Feb 15;31(4):555-62. doi: 10.1093/bioinformatics/btu686. Epub 2014 Oct 27. PMID: 25348213

[5] Röst HL, Liu Y, D'Agostino G, Zanella M, Navarro P, Rosenberger G, Collins BC, Gillet L, Testa G, Malmström L, Aebersold R. TRIC: an automated alignment strategy for reproducible protein quantification in targeted proteomics. Nat Methods. 2016 Sep;13(9):777-83. doi: 10.1038/nmeth.3954. Epub 2016 Aug 1. PMID: 27479329

[6] Käll L, Storey JD, Noble WS. QVALITY: non-parametric estimation of q-values and posterior error probabilities. Bioinformatics. 2009 Apr 1;25(7):964-6. doi: 10.1093/bioinformatics/btp021. Epub 2009 Feb 4. PMID: 19193729

PyProphet

IPF requires a specific branch of PyProphet. If Python and PIP are configured correctly, the following command can be used to install the latest version:

```
pip install git+https://github.com/grosenberger/pyprophet.git@feature/ipf
```

To work properly, PyProphet needs to find the `qvality` executable in the main path.

TRIC

TRIC should be installed according to the *TRIC* installation instructions.

Tutorial

Running IPF requires following several steps:

1. Peptide Query Parameter Generation

IPF requries a spectral library generated from DDA (or DIA pseudo spectra, e.g. from DIA-Umpire[7]) data. The protocol is described in more detail in the *Trans-Proteomic Pipeline* tutorial.

Using SpectraST, the spectral library `db_consensus.splib` is converted to a MRM and then to a TraML file:

```
# This will generate the file db_assays.mrm
spectrast -cNdb_assays -cICID-QTOF -cM db_consensus.splib

ConvertTSVToTraML -in db_assays.mrm -out db_assays.TraML
```

Then, a TraML file containing the detection and identification transitions is being generated. At this step, the residue modifiability needs to be defined in the OpenMS resource directory. For this purpose, the files `CHEMISTRY/PSI-MOD.obo` and `CHEMISTRY/unimod.xml` can be manually modified. An example for phosphorylation can be obtained from the ProteomeXchange repository. The location of the modified `OpenMS resource directory` needs to be supplied by setting `OPENMS_DATA_PATH` for `OpenSwathAssayGenerator`:

```
OPENMS_DATA_PATH=~/modified_path/share \
OpenSwathAssayGenerator -in db_assays.TraML \
-out db_assays_ptms.TraML \
-swath_windows_file swath64.txt \
-allowed_fragment_charges 1,2,3,4 \
-enable_ms1_uis_scoring \
-max_num_alternative_localizations 2000 \
-enable_identification_specific_losses \
-enable_identification_ms2_precursors
```

We then append decoys to the library:

```
OPENMS_DATA_PATH=~/modified_path/share \
OpenSwathDecoyGenerator -in db_assays_ptms.TraML \
-out db_assays_ptms_decoys.TraML \
-method shuffle \
```

[7] Tsou CC, Avtonomov D, Larsen B, Tucholska M, Choi H, Gingras AC, Nesvizhskii AI. DIA-Umpire: comprehensive computational framework for data-independent acquisition proteomics. Nat Methods. 2015 Mar;12(3):258-64, 7 p following 264. doi: 10.1038/nmeth.3255. Epub 2015 Jan 19. PMID: 25599550

```
-append \
-mz_threshold 0.1 \
-remove_unannotated
```

2. Targeted data extraction using OpenSWATH

The next step is conducted using OpenSWATH.

```
OPENMS_DATA_PATH=~/modified_path/share \
OpenSwathWorkflow -min_upper_edge_dist 1 \
-mz_extraction_window 0.05 \
-rt_extraction_window 600 \
-extra_rt_extraction_window 100 \
-min_rsq 0.95 \
-min_coverage 0.6 \
-use_ms1_traces \
-enable_uis_scoring \
-Scoring:uis_threshold_peak_area 0 \
-Scoring:uis_threshold_sn 0 \
-Scoring:stop_report_after_feature 5 \
-tr_irt DIA_iRT.TraML \
-tr db_assays_ptms_decoys.TraML \
-threads 8 \
-in MSDATA.mzXML.gz \
-out_tsv MSDATA_RESULTS.tsv
```

Important is to set the parameters `-use_ms1_traces` and `-enable_uis_scoring` to extract the additional identification transitions and precursor signals using OpenSWATH.

3. Statistical validation using PyProphet

PyProphet is then applied to the OpenSWATH results:

```
pyprophet --target.overwrite \
--final_statistics.emp_p \
--qvality.enable \
--qvality.generalized \
--ms1_scoring.enable \
--uis_scoring.enable \
--d_score.cutoff=100000 \
--semi_supervised_learner.num_iter=20 \
--xeval.num_iter=20 \
--ignore.invalid_score_columns \
--uis_scoring.expand_peptidoforms MSDATA_RESULTS.tsv
```

It generates reports on several different levels. Important for TRIC are the files that end with `*_uis_expanded.csv`. IPF attaches several columns, e.g. `PosteriorFullPeptideName`, which contains the peptidoform sequence of the best scoring peptidoform. The column `pfqm_score` represents the peptidoform q-value, whereas `pf_score` represent the posterior probability. After running IPF, the `m_score` column is equal to `pfqm_score` to enable alignment by TRIC.

4. Multi-run alignment using TRIC

TRIC can be applied to the IPF results with the following command:

```
feature_alignment.py --in *_uis_expanded.csv \
--out feature_alignment.csv \
--out_matrix feature_alignment_matrix.csv \
--file_format openswath \
--fdr_cutoff 0.01 \
--max_fdr_quality 0.2 \
--mst:useRTCorrection True \
--mst:Stdev_multiplier 3.0 \
--method LocalMST \
--max_rt_diff 30 \
--alignment_score 0.0001 \
--frac_selected 0 \
--realign_method lowess_cython \
--disable_isotopic_grouping
```

Data

Availability

The synthetic phosphopeptide reference mass spectrometry proteomics data is available from PRIDE/ProteomeXchange with the data set identifier PXD004573.

The enriched U2OS phosphopeptide mass spectrometry proteomics data is available from PRIDE/ProteomeXchange with the data set identifier PXD006056.

The 14-3-3β phosphopeptide interactomics mass spectrometry proteomics data is available from PRIDE/ProteomeXchange with the data set identifier PXD006057.

The twin study mass spectrometry proteomics data is available from PRIDE/ProteomeXchange with the data set identifier PXD004574.

References

TAPIR

Overview

TAPIR[1] is a visualization software for chromatographic data obtained by mass spectrometry. It provides efficient visualization of high-throughput targeted proteomics experiments.

The TAPIR software is a fast and efficient Python visualization software for chromatograms and peaks identified in targeted proteomics experiments. The input formats are open, community-driven standardized data formats (mzML for raw data storage and TraML encoding the hierarchical relationships between transitions, peptides and proteins).

TAPIR is scalable to proteome-wide targeted proteomics studies (as enabled by SWATH-MS), allowing researchers to visualize high-throughput datasets. The framework integrates well with existing automated analysis pipelines and can be extended beyond targeted proteomics to other types of analyses.

[1] Röst HL, Rosenberger G, Aebersold R, Malmström L. Efficient visualization of high-throughput targeted proteomics experiments: TAPIR. Bioinformatics. 2015 Jul 15;31(14):2415-7. doi: 10.1093/bioinformatics/btv152. Epub 2015 Mar 18. PMID: 25788625

Contact and Support

We provide support for TAPIR on the GitHub repository.

You can contact the author Hannes Röst.

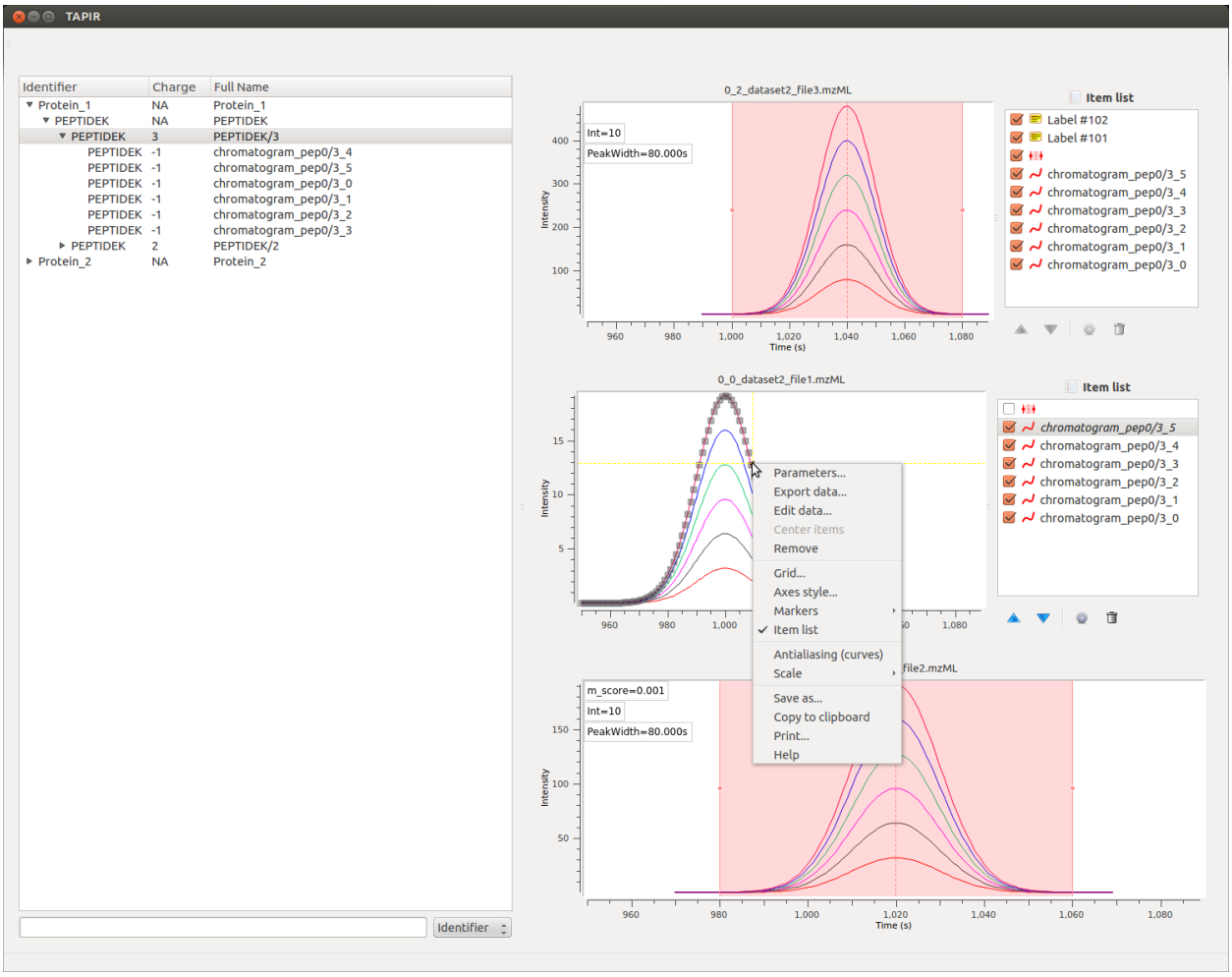
Installation

You can download binaries for Mac and Microsoft Windows (64 bit) directly from this website. The source code is available from Github which allows source-based installation. Please follow the instructions found there for manual installation or installation on a Linux system.

Note: for a successfull installation on Mac OS X, extract the provided file and drag it into the `Applications` folder. You may need to allow execution of the software if you see a warning that TAPIR is from an "unidentified developer". Simply go to System Preferences, click on "Security & Privacy" and in the "General" tab allow the execution of TAPIR.

Tutorial

The TAPIR software is highly flexible and interactive, allowing for investigation of single data traces and data points. Each graph item can be selected and inspected individually, allowing for customization of the visualization and production of publication-quality figures. Data can be exported as an image or in table format and used for further analysis; individual traces can be removed or re-added and all graph settings (such as color, line width, line style etc.) are fully customizable. The implementation relies on guiqwt for these features.



Data

Availability

You can download a small sample dataset. A larger, real-life dataset can be obtained by downloading these five files (this might take a while since the whole dataset is ca 5 GB)

Dataset 1 (Cond 1)

Dataset 2 (Cond 1)

Dataset 3 (Cond 2)

Dataset 4 (Cond 2)

Peak description file

This dataset is retrieved from the original OpenSWATH publication[2] and the two conditions (0% and 10%) refer to the treatment of *S. pyogenes* with human plasma. For each condition, two biological replicates are available.

[2] Röst HL, Rosenberger G, Navarro P, Gillet L, Miladinović SM, Schubert OT, Wolski W, Collins BC, Malmström J, Malmström L, Aebersold R. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. Nat Biotechnol. 2014 Mar 10;32(3):219-23. doi: 10.1038/nbt.2841. PMID: 24727770

References

SWATH2stats

Overview

SWATHstats[1] is intended to transform SWATH data from the OpenSWATH software into a format readable by other statistics packages while performing filtering, annotation and FDR estimation.

Contact and Support

You can contact the authors Peter Blattmann and Moritz Heusel.

References

[1] Blattmann P, Heusel M, Aebersold R. SWATH2stats: An R/Bioconductor Package to Process and Convert Quantitative SWATH-MS Proteomics Data for Downstream Analysis Tools. PLoS One. 2016 Apr 7;11(4):e0153160. doi: 10.1371/journal.pone.0153160. eCollection 2016. PMID: 27054327