
OpenSWATH Documentation

Release 1.0

OpenSWATH Developers

Dec 05, 2018

1	News	3
2	The OpenSWATH Workflow	5
2.1	Docker	5
2.1.1	Overview	5
2.1.2	Installing Docker	5
2.1.3	Running OpenSWATH in Docker	5
2.1.4	Software version information	6
2.2	Binaries	6
2.2.1	Overview	6
2.2.2	OpenMS	7
2.2.3	PyProphet	7
2.2.4	msproteomicstools & TRIC	7
2.3	Sources	7
2.3.1	Overview	7
2.3.2	OpenMS	7
2.3.3	PyProphet	8
2.3.4	msproteomicstools & TRIC	8
2.4	Getting Started	8
2.4.1	Tutorial Data	8
2.4.2	Questions?	8
2.4.3	References	9
2.5	Generic Transition Lists	9
2.5.1	Overview	9
2.5.2	Contact and Support	9
2.5.3	Format	9
2.5.4	Conversion	11
2.5.5	References	11
2.6	Trans-Proteomic Pipeline	11
2.6.1	Overview	11
2.6.2	Contact and Support	11
2.6.3	Tutorial	11
2.6.4	References	12
2.7	Skyline	12
2.7.1	Overview	12
2.7.2	Contact and Support	12

2.7.3	Tutorial	12
2.7.4	References	12
2.8	SWATHAtlas	12
2.8.1	Overview	12
2.8.2	Contact and Support	13
2.9	Peptide Query Parameter Generation	13
2.9.1	PQP Generation for OpenSWATH	13
2.9.2	PQP Generation for IPF	13
2.9.3	Decoy Generation	14
2.9.4	References	15
2.10	OpenSWATH	15
2.10.1	Overview	15
2.10.2	Contact and Support	15
2.10.3	Input	16
2.10.4	Parameters	17
2.10.5	Output	18
2.10.6	Tutorial Data	18
2.10.7	References	19
2.11	PyProphet	19
2.11.1	Overview	19
2.11.2	Contact and Support	20
2.11.3	Tutorial	20
2.11.4	References	23
2.12	Percolator	23
2.12.1	Overview	23
2.12.2	Contact and Support	23
2.12.3	Tutorial	23
2.12.4	References	24
2.13	TRIC	24
2.13.1	Overview	24
2.13.2	Contact and Support	24
2.13.3	Tutorial	24
2.13.4	Data	25
2.13.5	References	26
2.14	IPF	26
2.14.1	Overview	26
2.14.2	Contact and Support	26
2.14.3	Installation	26
2.14.4	Tutorial	26
2.14.5	Data	27
2.14.6	References	28
2.15	TAPIR	28
2.15.1	Overview	28
2.15.2	Contact and Support	28
2.15.3	Installation	28
2.15.4	Tutorial	28
2.15.5	Data	29
2.15.6	References	30
2.16	SWATH2stats	30
2.16.1	Overview	30
2.16.2	Contact and Support	30
2.16.3	References	30

3 Acknowledgments

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

News

Note: 2018-11-07: The Docker image now includes OpenMS 2.4.0 and PyProphet 2.0.1.

Note: 2018-11-07: With the release of OpenMS 2.4.0 and PyProphet 2.0.1, the new OpenSWATH workflow is available in the release branches.

Note: 2018-03-22: We provide an *experimental* Docker image for the latest development version of the OpenSWATH workflow.

Note: 2017-12-28: The tools of the OpenSWATH Workflow now provide *experimental* support for new SQLite-based file formats.

The OpenSWATH Workflow

2.1 Docker

2.1.1 Overview

Docker is a flexible and popular container management platform. With only few steps, you can obtain a full OpenSWATH installation including all dependencies on Windows, macOS and Linux. The OpenSWATH docker container will continuously be updated to the latest development version.

Users who just want to run the OpenSWATH workflow on their desktops can follow the instructions below to obtain an installation-free, always-updated workflow. It will run with nearly native speed on desktops, clusters and cloud environments.

2.1.2 Installing Docker

First, install [Docker Community Edition](#) for either Windows, macOS or Linux. On Linux, most current distributions provide this out-of-the-box. After installation, ensure that you assign the number of CPUs and RAM that you would like to use with OpenSWATH (e.g. 4 CPUs, 8GB RAM) and make sure to share your local drives. Please follow the instructions for [Windows](#) and [macOS](#).

2.1.3 Running OpenSWATH in Docker

Make sure that Docker is up and running in the background. On macOS or Linux, start a Terminal session. On Windows, start PowerShell, CMD or similar. Execute the following steps within the console:

```
# Download OpenSWATH image (openswath/openswath:0.2.0)
docker pull openswath/openswath:0.2.0
```

This will download the latest version of the OpenSWATH Docker image and cache it on your machine.

Note: Official containers for OpenMS and related tools are available via [BioContainers](#). The image provided here provides compatible release or development versions including related tools in an integrated container.

```
# Generate tutorial container (osw_tutorial) and log in
docker run --name osw_tutorial --rm -v ~/Desktop/:/data -i -t openswath/openswath:0.2.
↪0
```

This command will start a container based on the OpenSWATH image and map the local volume `~/Desktop/` (from your desktop) to `/data/` (to your container). It will open a Bash command line within the container for you to control the individual components of the workflow. If you want to exit the session, just type `exit` to return to your console.

Within the running container, you can execute all commands as you would in a native environment:

```
# Execute OpenSwathWorkflow in docker
OpenSwathWorkflow --help

# Execute PyProphet in docker
pyprophet --help

# Execute TRIC in docker
feature_alignment.py --help
```

All data that will be stored in `~/Desktop`, will be available in `/data/`. For example, we can process and write back the files like this:

```
OpenSwathWorkflow \
-in /data/data.mzML \
-tr /data/library.tsv \
-tr_irt /data/iRT_assays.TraML \
-swath_windows_file /data/SWATHwindows_analysis.tsv \
-sort_swath_maps -batchSize 1000 \
-readOptions cacheWorkingInMemory -tempDirectory /tmp/ \
-use_ms1_traces \
-mz_extraction_window 50 -ppm \
-mz_correction_function quadratic_regression_delta_ppm \
-TransitionGroupPicker:background_subtraction original \
-RTNormalization:alignmentMethod linear \
-Scoring:stop_report_after_feature 5 \
-out_tsv /data/osw_output.tsv \
```

2.1.4 Software version information

Please find further information at the [GitHub repository](#).

2.2 Binaries

2.2.1 Overview

Precompiled or prepackaged versions of OpenMS, PyProphet and msproteomicstools are readily available for most supported platforms.

2.2.2 OpenMS

OpenSWATH is completely integrated into OpenMS. The current releases can be obtained from [GitHub](#). To make use of the latest developments, consider using the [nightly builds for Windows and macOS](#).

Instructions for compilation and installation can be obtained from the [OpenMS documentation](#).

2.2.3 PyProphet

PyProphet requires Python 2.7 or Python 3. Windows users should install Anaconda. Mac and Linux users should be able to install PyProphet directly from [GitHub](#). It is strongly recommended to install PyProphet within a [virtualenv](#).

```
# Install dependencies
pip install numpy scipy scikit-learn pandas numexpr statsmodels Click matplotlib
↪seaborn

# Install PyProphet release version
pip install pyprophet

# Alternative: Install PyProphet development version
pip install git+https://github.com/PyProphet/pyprophet.git@master
```

2.2.4 msproteomicstools & TRIC

msproteomicstools 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 msproteomicstools through PyPI:

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

You can alternatively download the msproteomicstools 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
```

2.3 Sources

2.3.1 Overview

The source code of OpenMS, PyProphet and msproteomicstools is provided on [GitHub](#) under the 3-clause BSD license. Please refer to the individual repositories for information regarding obtaining, compiling and installing the components.

2.3.2 OpenMS

OpenSWATH is fully integrated within OpenMS. Please obtain the latest version of OpenMS from [GitHub](#).

Instructions for compilation and installation can be obtained from the [OpenMS documentation](#).

2.3.3 PyProphet

Please obtain the latest version of PyProphet from [GitHub](#).

The instructions for the installation of the PyProphet *Binaries* is applicable here as well.

2.3.4 msproteomicstools & TRIC

Please obtain the latest version of msproteomicstools from [GitHub](#).

The instructions for the installation of the msproteomicstools *Binaries* is applicable here as well.

2.4 Getting Started

Note: OpenSWATH is part of the OpenMS project, therefore if you install OpenMS you will get OpenSWATH automatically. Please refer to our [installation instructions](#).

To get started with the DIA or SWATH-MS analysis, we recommend to follow the recently published step-for-step tutorial¹ describing a complete OpenSWATH analysis workflow using `OpenSwathWorkflow` available from [bioRxiv](#). In addition, you can find in-depth information about the different tools in a SWATH analysis on this page.

2.4.1 Tutorial Data

You can access the tutorial data (a *M. tuberculosis* dataset) used in the 2017 Methods Mol Biol. OpenSWATH tutorial¹ [here](#). The data contains

- 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)

2.4.2 Questions?

Please address questions directly to the [mailing list](#).

¹ 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

2.4.3 References

2.5 Generic Transition Lists

2.5.1 Overview

OpenSWATH¹ supports generic transition lists in tab- or comma-separated value (TSV/CSV) formats. TargetedFileConverter is part of OpenMS² and is available for Windows, macOS and Linux.

Transitions lists from *Trans-Proteomic Pipeline* and *Skyline* can be directly converted. Alternatively, it is possible to use the generic transition list format, described below.

2.5.2 Contact and Support

We provide support for TargetedFileConverter using the [OpenMS support channels](#). Please address general questions to the open-ms-general mailing list.

2.5.3 Format

The input to TargetedFileConverter can come either directly from *Trans-Proteomic Pipeline* or *Skyline* or the generic transition list format. The columns should be separated either by comma or tab and the files should be saved with Windows or Unix line endings.

Required Columns

The following columns are required:

- PrecursorMz * (float)
- ProductMz * (float; synonyms: FragmentMz)
- LibraryIntensity * (float; synonyms: RelativeFragmentIntensity)
- NormalizedRetentionTime * (float; synonyms: RetentionTime, Tr_recalibrated, iRT, RetentionTime-CalculatorScore) (normalized retention time)

Targeted Proteomics Columns

For targeted proteomics, the following additional columns should be provided:

- ProteinId ** (free text; synonyms: ProteinName)
- PeptideSequence ** (free text, sequence only (no modifications); synonyms: Sequence, StrippedSequence)
- ModifiedPeptideSequence ** (free text, should contain modifications; synonyms: FullUniModPeptideName, FullPeptideName, ModifiedSequence)
- PrecursorCharge ** (integer, contains the charge of the precursor; synonyms: Charge)

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 Röst HL, Sachsenberg T, Aiche S, Bielow C, Weissner 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

- `ProductCharge **` (integer, contains the fragment charge; synonyms: `FragmentCharge`)
- `FragmentType` (free text, contains the type of the fragment, e.g. “b” or “y”)
- `FragmentSeriesNumber` (integer, e.g. for y7 use “7” here; synonyms: `FragmentNumber`)

Warning: `ModifiedPeptideSequence` should contain modifications in UniMod (preferred) (`(. (UniMod:1)PEPC (UniMod:4)PEPM (UniMod:35)PEPR. (UniMod:2)`) or TPP (`(n[43]PEPC[160]PEPM[147]PEPRc[16]`) formats. N- and C-terminal modifications are indicated after a pre- or suffix ..

Targeted Metabolomics Columns

For targeted metabolomics, the following fields are also supported:

- `CompoundName **` (synonyms: `CompoundId`)
- `SMILES`
- `SumFormula`

Grouping Columns

OpenSWATH uses grouped transitions to detect candidate analyte signals. These groups are by default generated based on the input, but can also be manually specified:

- `TransitionGroupId` (free text, designates the transition group [e.g. peptide] to which this transition belongs; synonyms: `TransitionGroupName`, `transition_group_id`)
- `TransitionId` (free text, needs to be unique for each transition [in this file]; synonyms: `TransitionName`, `transition_name`)
- `Decoy` (1: decoy, 0: target, i.e. no decoy; determines whether the transition is a decoy transition or not, synonyms: `decoy`, `isDecoy`)
- `PeptideGroupLabel` (free text, designates to which peptide label group (as defined in MS:1000893) the peptide belongs to)
- `DetectingTransition` (1: use transition to detect peak group, 0: don't use transition for detection; synonyms: `detecting_transition`)
- `IdentifyingTransition` (1: use transition for peptidofom inference using IPF, 0: don't use transition for identification; synonyms: `identifying_transition`)
- `QuantifyingTransition` (1: use transition to quantify peak group, 0: don't use transition for quantification; synonyms: `quantifying_transition`)

Warning: If you are unsure about these columns, `TargetedFileConverter` and the downstream tools `OpenSwathAssayGenerator` and `OpenSwathDecoyGenerator` will generate them automatically for you.

Optional Columns

Optionally, the following columns can be specified but they are not actively used by OpenSWATH:

- CollisionEnergy (float; synonyms: CE)
- Annotation (free text, e.g. y7)
- UniprotId (free text; synonyms: UniprotID)
- LabelType (free text, optional description of which label was used, e.g. heavy or light)

Fields indicated with * are strictly required while fields indicated with ** are only required in the specific context (proteomics or metabolomics).

2.5.4 Conversion

Once the generic transition list is generated by the user (or TPP or Skyline), peptide query parameters can be derived by the OpenSWATH tools part of OpenMS. Please follow the instructions on how to install *OpenSWATH* prior to the next steps.

```
# Import from SpectraST MRM
TargetedFileConverter -in transitionlist.mrm -out transitionlist.TraML

# Import from Skyline or generic transition list format (TSV)
TargetedFileConverter -in transitionlist.tsv -out transitionlist.TraML

# Import from Skyline or generic transition list format (CSV)
TargetedFileConverter -in transitionlist.csv -out transitionlist.TraML
```

2.5.5 References

2.6 Trans-Proteomic Pipeline

2.6.1 Overview

The [Trans-Proteomic Pipeline](#) provides a complete workflow to generate spectral libraries suitable for OpenSWATH. Data sets acquired in data-dependent acquisition (DDA) or data-independent acquisition (DIA; preprocessed by DIA-Umpire²) can be searched by spectrum-centric scoring against a reference FASTA database using several supported search engines. After statistical validation using PeptideProphet, iProphet & MAYU, a consensus spectral library is generated by SpectraST, which serves as final input for the OpenMS tool TargetedFileConverter.

2.6.2 Contact and Support

We provide support separately for the [TPP](#), the [msproteomicstools](#) and the [OpenMS](#) components of the workflow.

2.6.3 Tutorial

A comprehensive tutorial¹ describes the individual steps to generate spectral libraries for SWATH-MS using the [Trans-Proteomic Pipeline \(TPP\)](#). Please follow the tutorial until the generation of SpectraST spectral libraries. If you are

² 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

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

using DIA data, follow the [DIA-Umpire tutorial](#) to generate pseudo-spectra first, which can then be processed using the TPP.

In the last step, import the SpectraST consensus library (format `sptxt` or `splib`) and convert it to a MRM transition list:

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

Convert then the MRM transition list to a TraML spectral library and follow the remaining steps in the *Generic Transition Lists* section.

2.6.4 References

2.7 Skyline

2.7.1 Overview

Skyline¹ is the most popular analysis software for targeted proteomics. It can be used to export spectral libraries suitable for OpenSWATH.

2.7.2 Contact and Support

Skyline is supported by a very active community via their [message board](#).

For TargetedFileConverter, we provide support via the [OpenMS support channels](#).

2.7.3 Tutorial

The Skyline team provides extensive documentation and [tutorials](#) and their website. Please follow their instructions to generate spectral or chromatogram libraries. To export a library for use in OpenSWATH, select `File` and then `Report`. You can then import the `OpenSWATH.skyr` [report definition file](#), which will directly export all necessary columns.

Convert the Skyline transition list to a TraML spectral library and follow the remaining steps in the *Generic Transition Lists* section.

2.7.4 References

2.8 SWATHAtlas

2.8.1 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 MacLean B, Tomazela DM, Shulman N, Chambers M, Finney GL, Frewen B, Kern R, Tabb DL, Liebler DC, MacCoss MJ. Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics*. 2010 Apr 1;26(7):966-8. doi: 10.1093/bioinformatics/btq054. Epub 2010 Feb 9. PMID: 20147306

2.8.2 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.

2.9 Peptide Query Parameter Generation

2.9.1 PQP Generation for OpenSWATH

After importing transitions lists from an upstream workflow (e.g. *Trans-Proteomic Pipeline*, *Skyline* or *Generic Transition Lists*), the transitions can then be optimized using a set of heuristic rules¹:

```
OpenSwathAssayGenerator -in transitionlist.TraML \
-out transitionlist_optimized.TraML \
-swath_windows_file swath64.txt \
```

Please note that the SWATH windows file should be of the following format (tab-separated), including header:

```
lower_offset upper_offset
400 425
424 450
...
...
```

If necessary, the rules for transition selection can be modified with the following parameters:

```
OpenSwathAssayGenerator <other parameters>
-min_transitions 6 \
-max_transitions 6 \
-allowed_fragment_types b,y \
-allowed_fragment_charges 1,2,3,4 \
-enable_detection_specific_losses \
-enable_detection_unspecific_losses \
-precursor_mz_threshold 0.025 \
-precursor_lower_mz_limit 400 \
-precursor_upper_mz_limit 1200 \
-product_mz_threshold 0.025 \
-product_lower_mz_limit 350 \
-product_upper_mz_limit 2000 \
```

2.9.2 PQP Generation for IPF

If IPF scoring² should be conducted, the following parameters should be considered in addition to the others above:

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

² 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 peptidofoms 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

```
OpenSwathAssayGenerator <other parameters>
-enable_ipf \
-unimod_file unimod_phospho.xml \
```

Unimod contains descriptions of more than 1400 post-translational modifications and represents the standard database. However, many modification types are annotated with residue modifiabilities that go beyond the canonical set (e.g. phosphorylation (S,T,Y,D,H,C,R,K) instead of (S,T,Y)).

For the purpose of site-localization, it is thus very important to provide a modified (restricted) Unimod file to `OpenSwathAssayGenerator`. This file can be created by editing the file `unimod.xml` to only the desired modifications and residue modifiabilities. We provide an [example](#) for Phosphorylation (S,T,Y), Carbamidomethyl (C), Oxidation (M) and SILAC (R, K). Please note that also usually fixed PTMs like Carbamidomethyl and Oxidation need to be set.

The generation of identification transitions can be adjusted if necessary by the following parameters:

```
OpenSwathAssayGenerator <other parameters>
-enable_ipf \
-unimod_file unimod_phospho.xml \
-max_num_alternative_localizations 10000 \
-disable_identification_ms2_precursors \
-disable_identification_specific_losses \
-enable_identification_unspecific_losses \
-enable_swath_specificity \
```

`OpenSwathAssayGenerator` excludes peptides that can have too many combinations of alternative site-localization (track the process by setting `-debug 10`). If 10000 alternative peptidofoms are too few, consider increasing this parameter.

By default, unfragmented precursors are extracted from the SWATH maps and used for scoring by IPF, this can optionally be disabled (`-disable_identification_ms2_precursors`). Specific losses (e.g. for Phosphorylation) are used by default and improve specificity; unspecific losses are not recommended to use.

In scenarios with extremely small precursor isolation windows (e.g. < 1 Th), `-enable_swath_specificity` can be used to skip the precursor inference step of IPF. This is not recommended in general.

2.9.3 Decoy Generation

Decoys can then be appended using `OpenSwathDecoyGenerator`:

```
OpenSwathDecoyGenerator -in transitionlist_optimized.TraML \
-out transitionlist_optimized_decoys.TraML \
```

Warning: If you used non-default parameters in `OpenSwathAssayGenerator` (i.e. `-product_mz_threshold`, `-allowed_fragment_types`, `-allowed_fragment_charges`, `-enable_detection_specific_losses` or `-enable_detection_unspecific_losses`), make sure to also specify them for `OpenSwathDecoyGenerator`. The flag `--help` will show a list of all options.

You can then convert the TraML to a PQP file:

```
TargetedFileConverter -in transitionlist_optimized_decoys.TraML \
-out transitionlist_optimized_decoys.PQP \
```

This processed spectral library (including decoys) is the input for *OpenSWATH*.

2.9.4 References

2.10 OpenSWATH

2.10.1 Overview

OpenSWATH¹ 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 OpenMS³. 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.

The OpenSwathWorkflow executable is currently the most efficient way of running OpenSWATH^{1,2} and it is available through OpenMS³. An extended tutorial describing a complete OpenSWATH analysis workflow using OpenSwathWorkflow was recently published⁴ and is also available from [bioRxiv](#) with its associated [dataset](#).

The OpenSwathWorkflow implements the OpenSWATH analysis workflow as described in¹ and provides a complete, integrated analysis tool without the need to run multiple tools consecutively.

It executes the following steps in order:

- Reading of the raw input file (provided as mzML, mzXML or sqMass) and RT normalization transition list
- Computing the retention time transformation using RT normalization peptides
- Reading of the transition list
- Extracting the specified transitions
- Scoring the peak groups in the extracted ion chromatograms (XIC)
- Reporting the peak groups and the chromatograms

2.10.2 Contact and Support

We provide support for OpenSWATH using the [OpenMS support channels](#). Please address general questions to the [mailing list](#).

You can contact the authors [Hannes Röst](#) and [George Rosenberger](#).

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, Nahsen 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 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. [bioRxiv](#).

2.10.3 Input

The input to OpenSwathWorkflow are provided using the following files:

- `in` raw input file (provided as `mzML`, `mzXML` or `sqMass`)
- `tr` transition list (spectral library)
- `tr_irt` an optional transition file containing RT normalization coordinates
- `swath_windows_file` an optional file specifying the *analysis* SWATH windows

Mass spectrometric data

The input file `in` is generally a single `mzML`, `mzXML` or `sqMass` file (converted from a raw vendor file format using `ProteoWizard`).

Spectral library

The spectral library `tr` is a spectral library either in `.tsv`, `.TraML` or `.PQP` format (where the `TSV` or `PQP` format is recommended). Further information in generating these files can be found in the *Generic Transition Lists* section.

Retention time normalization

The retention time normalization peptides are provided using the optional parameter `tr_irt` in `TraML` format. We suggest to use the `IRTassays.TraML` file provided in the tutorial dataset, if the Biognosys `iRT-kit` was used during sample preparation.

If the `iRT-kit` was not used, it is highly recommended to use or generate a set of endogenous peptides for RT normalization. A recent publication⁵ provides such a set of `CiRT` peptides suitable for many eukaryotic samples. The `TraML` file from the supplementary information can be used as input for `tr_irt`. Since not all `CiRT` peptides might be found, the flag `RTNormalization:estimateBestPeptides` should be set to improve initial filtering of poor signals. Further parameters for optimization can be found when invoking `OpenSwathWorkflow --helphelp` under the `RTNormalization` section. Those do not require adjustment for most common sample types and `LC-MS/MS` setups, but might be useful to tweak for specific scenarios.

SWATH windows definition

The `SWATH` windows themselves can either be read from the input files, but it is recommended to provide them explicitly in tab-delimited form. Note that there is a difference between the *SWATH window acquisition scheme settings* and the *SWATH window analysis settings*:

The *acquisition* settings tell the instrument how to acquire the data and how to filter the transitions (see section *Peptide Query Parameter Generation*).

The *analysis* settings on the other hand specify from which precursor isolation windows to extract the data. Note that the analysis windows should not have any overlap.

We suggest to use the `SWATHwindows_analysis.tsv` file provided in the tutorial dataset for 32 windows of 25 Da each.

⁵ Parker SJ, Rost H, Rosenberger G, Collins BC, Malmström L, Amodei D, Venkatraman V, Raedschelders K, Van Eyk JE, Aebersold R. Identification of a Set of Conserved Eukaryotic Internal Retention Time Standards for Data-independent Acquisition Mass Spectrometry. *Mol Cell Proteomics*. 2015 Oct;14(10):2800-13. doi: 10.1074/mcp.O114.042267. Epub 2015 Jul 21. PMID: 26199342

2.10.4 Parameters

Caching of mass spectrometric data

Due to the large size of the files, OpenSwathWorkflow implements a caching strategy where files are cached to disk and then read into memory SWATH-by-SWATH. You can enable this by setting `-readOptions cacheWorkingInMemory -tempDirectory /tmp` where you would need to adjust the temporary directory depending on your platform.

Other potentially useful options you may want to turn on are `batchSize` and `sort_swath_maps`.

Chromatographic parameters

The current parameters are optimized for 2 hour gradients on SCIEX 5600 / 6600 TripleTOF instruments with a peak width of around 30 seconds using iRT peptides. If your chromatography differs, please consider adjusting `-Scoring:TransitionGroupPicker:min_peak_width` to allow for smaller or larger peaks and adjust the `-rt_extraction_window` to use a different extraction window for the retention time.

Mass spectrometric parameters

In m/z domain, consider adjusting `-mz_extraction_window` to your instrument resolution, which can be in Th or ppm (using `-ppm`). In addition to using the iRT peptides for correction of the retention time space, OpenSWATH can also use those peptides to correct the m/z space with the option `-mz_correction_function quadratic_regression_delta_ppm`. For quantification, it can be beneficial to enable background subtraction using `-TransitionGroupPicker:background_subtraction original` as described in the software comparison paper⁶.

MS1 and IPF parameters

Furthermore, if you wish to use MS1 information, use the `-use_ms1_traces` flag, assuming that your input data contains an MS1 map in addition to the SWATH data. This is generally recommended. If you would like to enable IPF transition-level scoring and your spectral library was generated according to the IPF instructions, you should set the `-enable_uis_scoring` flag.

Example

Therefore, a full run of OpenSWATH may look like this:

```
OpenSwathWorkflow.exe
-in data.mzML -tr library.tsv
-tr_irt iRT_assays.TraML
-swath_windows_file SWATHwindows_analysis.tsv
-sort_swath_maps -batchSize 1000
-readOptions cacheWorkingInMemory -tempDirectory C:\Temp
-use_ms1_traces
-mz_extraction_window 50 -ppm
-mz_correction_function quadratic_regression_delta_ppm
-TransitionGroupPicker:background_subtraction original
-RTNormalization:alignmentMethod linear
```

(continues on next page)

⁶ Navarro P, Kuharev J, Gillet LC, Bernhardt OM, MacLean B, Röst HL, Tate SA, Tsou CC, Reiter L, Distler U, Rosenberger G, Perez-Riverol Y, Nesvizhskii AI, Aebersold R, Tenzer S. A multicenter study benchmarks software tools for label-free proteome quantification. Nat Biotechnol. 2016 Nov;34(11):1130-1136. doi: 10.1038/nbt.3685. Epub 2016 Oct 3.

(continued from previous page)

```
-Scoring:stop_report_after_feature 5  
-out_tsv osw_output.tsv
```

Troubleshooting

If you encounter issues with peak picking, try to disable peak filtering by setting `-Scoring:TransitionGroupPicker:compute_peak_quality false` which will disable the filtering of peaks by chromatographic quality. Furthermore, you can adjust the smoothing parameters for the peak picking, by adjusting `-Scoring:TransitionGroupPicker:PeakPickerMRM:sgolay_frame_length` or using a Gaussian smoothing based on your estimated peak width. Adjusting the signal to noise threshold will make the peaks wider or smaller.

2.10.5 Output

The OpenSwathWorkflow produces two types of output:

- identified peaks
- extracted chromatograms

the identified peaks can be stored in tsv format using `-out_tsv` (recommended), in SQLite format using `--out_osw` (experimental) or in a featureXML format using `-out_features` (not recommended).

the extracted chromatograms can be stored in mzML format using `out_chrom` with an `.mzML` extension. By default the produced mzML file will be numpress compressed, but can be converted to regular mzML using the OpenMS FileConverter. Alternatively, output can be written in `.sqMass` format, which is a SQLite-based format (experimental).

2.10.6 Tutorial Data

Availability

To learn OpenSWATH, we suggest to use the *M. tuberculosis* dataset published alongside the 2017 Methods Mol Biol. OpenSWATH tutorial⁴ 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 Biotechnology publication) are available from the PeptideAtlas raw data repository with accession number [PASS00289](#).

The Skyline results are available from [Skyline Panorama Webserver](#).

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

2.10.7 References

2.11 PyProphet

2.11.1 Overview

PyProphet¹ is a reimplementation of the mProphet² algorithm for targeted proteomics. It is particularly optimized for analysis of large scale data sets generated by OpenSWATH or DIANA.

This description represents the new SQLite-based workflow that is available since OpenMS 2.4.0 and PyProphet 2.0.1. This version includes the IPF³ and large-scale data set optimizations⁴. You can alternatively follow the instructions for the pyprophet_legacy.

¹ Telean 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

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 peptidofoms 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

2.11.2 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](#).

2.11.3 Tutorial

Merging

Generate OSW output files according to section `openswath_workflow`. PyProphet is then applied to one or several such SQLite-based reports. Several different commands can be run to consecutively do the analysis:

```
pyprophet --help
pyprophet merge --help
```

This command provides an overview of all available commands to manipulate OSW input files. Further instructions are available for the individual commands.

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

In most scenarios, more than a single DIA / SWATH-MS run was acquired and the samples should be compared qualitatively and/or quantitatively with the OpenSWATH workflow. After individual processing with OpenSWATH and the identical spectral library, the files can be merged by PyProphet.

This command will merge and optionally subsample multiple files. Please note that the experiment-wide context on peptide query-level is applied to merged files, whereas the run-specific context is used with separate OSW files⁴.

If semi-supervised learning is too slow, or the run-specific context is required, 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(s).

Scoring

```
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.

When using the IPF extension, the parameter `--level` can be set to `ms2`, `ms1` or `transition`. If MS2 and MS2 information should be considered together, `--level` can alternatively be set to `ms1ms2`. 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
```

The scoring steps on MS1 and transition-level have some dependencies on the MS2 peak group signals. The parameter `--ipf_max_peakgroup_rank` specifies how many peak group candidates should be assessed in IPF. For example, if this parameter is set to 1, only the top scoring peak group will be investigated. In some scenarios, a set of peptide query parameters might detect several peak groups of different peptidofoms that should be independently identified. If the parameter is set to 3, the top 3 peak groups are investigated. Note that for higher values (or very generic applications), it might be a better option to disable the PyProphet assumption of a single best peak group per peptide query. This can be conducted by setting `--group_id` to `feature_id` and will change the assumption that all high scoring peak groups are potential peptide signals.

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.

IPF

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`. If MS1 or MS2 precursor data should not be used, e.g. due to poor instrument performance, this can be disabled by setting `--no-ipf_ms1_scoring` and `--no-ipf_ms2_scoring`. The experimental setting `--ipf_grouped_fdr` can be used in case of extremely heterogeneous spectral library, e.g. containing mostly unmodified peptides that are mainly detect and peptidofoms with various potential site-localizations, which are mostly not detectable. This parameter will estimate the FDR independently grouped according to number of site-localizations.

Several thresholds (`-ipf_max_precursor_pep`, `-ipf_max_peakgroup_pep`, `-ipf_max_precursor_peakgroup_pep`, `-ipf_max_transition`) are defined for IPF to exclude very poor signals. When disabled, the error model still works, but sensitivity is reduced. Tweaking of these parameters should only be conducted with a reference data set.

Contexts & FDR

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=experiment-wide \
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=experiment-wide \
protein --in=merged.osw --context=global
```

Exporting

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

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

By default, both peptide- and transition-level quantification is reported, which is necessary for requantification or SWATH2stats. If peptide and protein inference in the global context was conducted, the results will be filtered to 1% FDR by default. Further details can be found by `pyprophet export --help`.

Warning: By default, IPF results on peptidofom-level will be used if available. This can be disabled by setting `--ipf=disable`. The IPF results require different properties for TRIC. Please ensure that you want to analyze the results in the context of IPF, else, use the `--ipf=disable` or `--ipf=augmented` settings.

Scaling up

When moving to larger data sets that include 10-1000s of runs, the workflow described above might take a lot of time. For such applications, especially when the analysis is run on HPC infrastructure (cloud, cluster, etc.) we have implemented steps that can mostly parallelize on the level of independent runs:

In the first step, we will generate a subsampled classifier that is much faster to learn:

```
# Here we recommend to set subsample_rate to 1/N, where N is the number of runs.
# Example for N=10 runs:
pyprophet merge --out=model.osw --subsample_ratio=0.1 *.osw
```

In case of very large data sets, this step can alternatively be parallelized:

```
# Here we recommend to set subsample_rate to 1/N, where N is the number of runs.
# Example for N=10 runs:
for run in run_*.osw
do
run_subsampled = ${run}s # generates .osws files
pyprophet merge --out=${run_subsampled} --subsample_ratio=0.1 $run
done

pyprophet merge --out=model.osw *.osws
```

We then learn a classifier on MS1/MS2-level and store the results in `model.osw`:

```
pyprophet score --in=model.osw --level=mslms2
```

This classifier is then applied to all individual runs in parallel:

```
for run in run_*.osw
do
pyprophet score --in=$run --apply_weights=model.osw --level=mslms2
done
```

We then extract the relevant data for global scoring to generate a tiny file:

```
for run in run_*.osw
do
run_reduced = ${run}r # generates .oswr files
pyprophet reduce --in=$run --out=${run_reduced}
done
```

Next, global peptide and protein-level error rate control is conducted by merging the `oswr` files:

```
pyprophet merge --template=model.osw --out=model_global.osw *.oswr

pyprophet peptide --context=global --in=model_global.osw

pyprophet protein --context=global --in=model_global.osw
```

Now we backpropagate the global statistics to the individual runs:

```
for run in run_*.osw
do
pyprophet backpropagate --in=$run --apply_scores=${run_reduced}
done
```

We can then export the results with confidence scores on peptide-query-level (run-specific context), peptide sequence level (global context) and protein level (global context) in parallel:

```
for run in run_*.osw
do
pyprophet export --in=$run
done
```

2.11.4 References

2.12 Percolator

2.12.1 Overview

Percolator^{1,2} is a popular algorithm for statistical validation of shotgun proteomic data. Using the OpenMS TOPP tool `PercolatorAdapter`, Percolator can score the results from OpenSWATH and thus be used instead of PyProphet.

2.12.2 Contact and Support

OpenSWATH support in `PercolatorAdapter` is currently in development and must NOT be used in production environments. We would however be very grateful for testing of the tool and reporting of problems and bugs.

We provide support for `PercolatorAdapter` using the [OpenMS support channels](#). Please address general questions to the [open-ms-general](#) mailing list.

Please address any general Percolator inquiries to the [Percolator team](#).

2.12.3 Tutorial

PercolatorAdapter

`PercolatorAdapter` is available in the OpenMS development branch. To convert the results of `PercolatorAdapter` to an OpenSWATH TSV report, the SQLite-enabled PyProphet version is required. Please install these versions according to the instructions in the [Binaries](#) section.

After installation, `PercolatorAdapter` can be run on the OSW results using the following commands:

```
# Score on MS2-level
PercolatorAdapter -in_osw openswath_results.osw -out openswath_results.osw \
-osw_level ms2

# Score on MS1-level
PercolatorAdapter -in_osw openswath_results.osw -out openswath_results.osw \
-osw_level ms1

# Score on transition-level
PercolatorAdapter -in_osw openswath_results.osw -out openswath_results.osw \
-osw_level transition
```

¹ Käll L, Canterbury JD, Weston J, Noble WS, MacCoss MJ. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat Methods*. 2007 Nov;4(11):923-5. Epub 2007 Oct 21. PMID: 17952086

² The M, MacCoss MJ, Noble WS, Käll L. Fast and Accurate Protein False Discovery Rates on Large-Scale Proteomics Data Sets with Percolator 3.0. *J Am Soc Mass Spectrom*. 2016 Nov;27(11):1719-1727. Epub 2016 Aug 29. PMID: 27572102

PyProphet

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

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

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

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

2.12.4 References

2.13 TRIC

2.13.1 Overview

TRIC¹ 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.

2.13.2 Contact and Support

We provide support for TRIC on the [GitHub repository](#).

You can contact the author [Hannes Röst](#).

2.13.3 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
--mst:useRTCorrection True --mst:Stdev_multiplier 3.0
--target_fdr 0.01 --max_fdr_quality 0.05
```

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

An extended tutorial describing a complete OpenSWATH analysis workflow including TRIC was recently published² and is also available from [bioRxiv](#).

2.13.4 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

2.13.5 References

2.14 IPF

2.14.1 Overview

IPF (Inference of PeptidoForms)¹ is an extension to the OpenSWATH² 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 OpenMS³ and PyProphet⁴ and compatible with the downstream alignment algorithm TRIC⁵.

2.14.2 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](#).

2.14.3 Installation

IPF is fully integrated within the tools of the OpenSWATH workflow. Please follow the [OpenSWATH](#), [PyProphet](#) and [TRIC](#) installation instructions for the latest development branches. The current instructions are written for the new SQLite-based workflow. You can alternatively follow the instructions for the `ipf_legacy`.

2.14.4 Tutorial

Running IPF requires to modify the parameters of several steps of tools part of the OpenSWATH workflow:

1. Peptide Query Parameter Generation

IPF requires a spectral library generated from DDA (or DIA pseudo spectra, e.g. from DIA-Umpire⁷) data. The input can come for example from *Trans-Proteomic Pipeline*, *Skyline* or can be provided in the form of *Generic Transition Lists*. The underlying PSMs do not need to be site-localized, as IPF will assess site-localization independently. However, a site-localized spectral library might provide better peptide query parameters.

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, Bielew 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 Teلمان 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

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

The first step uses `OpenSwathAssayGenerator` to append *in silico* identification transitions to the spectral library. The required parameters (including residue modifiability) and considerations are described in the section [Peptide Query Parameter Generation](#). The spectral library should also be appended with decoys and converted to a PQP file.

2. Targeted data extraction using OpenSWATH

The next step is conducted using OpenSWATH. Follow the IPF-specific instructions in the section `openswath_workflow`. Important is to enable MS1 and transition-level scoring by setting the parameters `-use_ms1_traces` and `-enable_uis_scoring` for `OpenSwathWorkflow`. Make sure to use the PQP spectral library as input and write an OSW file as output.

3. Statistical validation using PyProphet

PyProphet is then applied to the OpenSWATH results. Follow the IPF-specific instructions in the [PyProphet](#) section. Export a legacy TSV report for analysis with 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
```

Note that IPF does not report decoys, which is the reason why `max_fdr_quality` must be set.

2.14.5 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 14-3-3 β 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](#).

2.14.6 References

2.15 TAPIR

2.15.1 Overview

TAPIR¹ 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.

2.15.2 Contact and Support

We provide support for TAPIR on the [GitHub repository](#) through GitHub issues or you can contact the author [Hannes Röst](#).

2.15.3 Installation

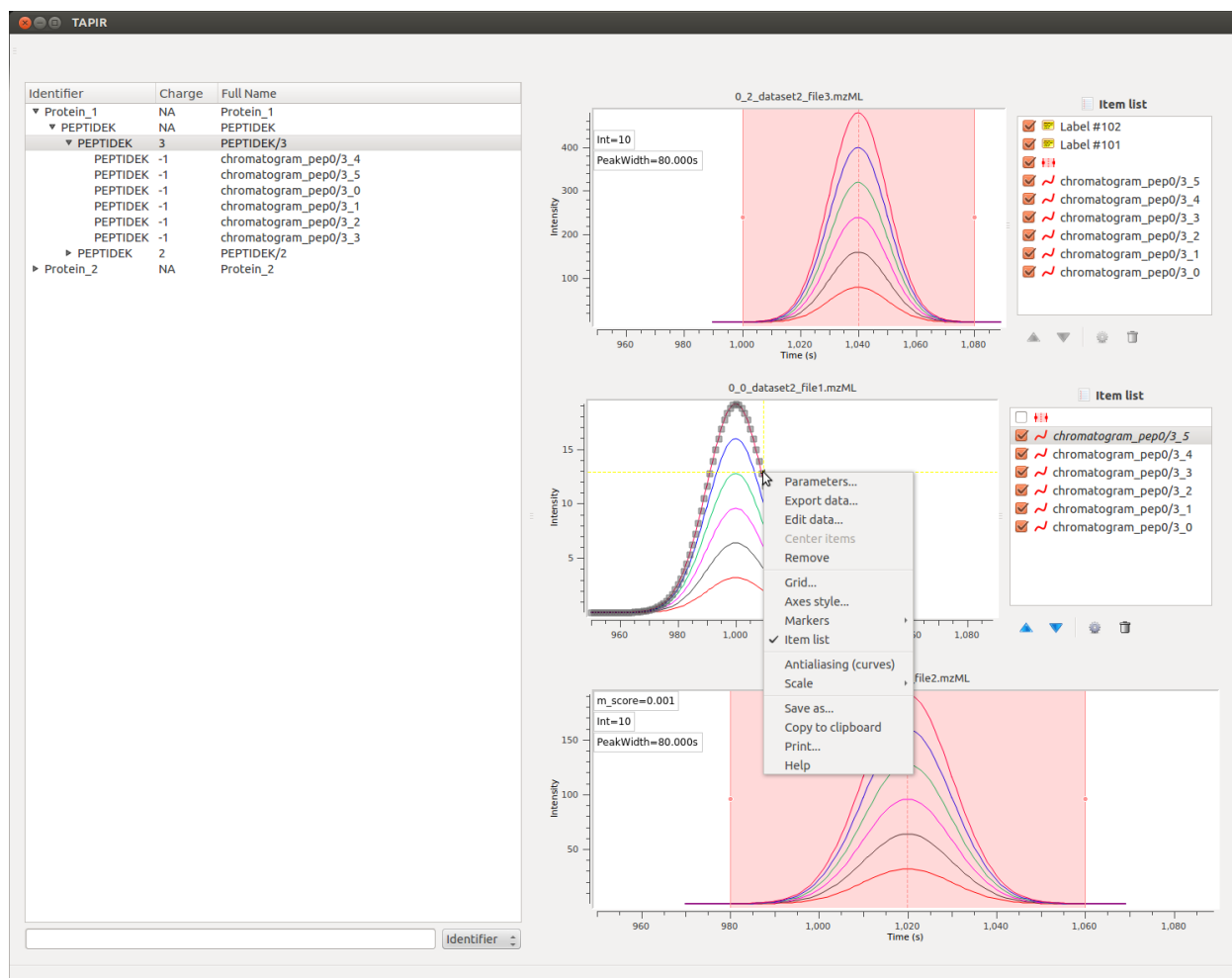
You can download binaries [from here](#) (Mac OSX and 64 bit Microsoft Windows). The source code is available from [Github](#) which allows source-based installation. Please follow the instructions found there for manual installation or installation on a Linux system.

Note: for a successful 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.

2.15.4 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.

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



2.15.5 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 publication² 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

2.15.6 References

2.16 SWATH2stats

2.16.1 Overview

[SWATH2stats](#)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.

2.16.2 Contact and Support

You can contact the authors [Peter Blattmann](#) and [Moritz Heusel](#).

2.16.3 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

CHAPTER 3

Acknowledgments

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