A data analysis protocol for quantitative data-independent acquisition proteomics

Pietilä S.[1], Suomi T.[1], Aakko J.[1], Elo L.L.[1,*]

[1] Turku Centre for Biotechnology, University of Turku and Åbo Akademi University, Turku, Finland

* laura.elo@utu.fi

Abstract

Data-independent acquisition (DIA) mode of mass spectrometry, such as the SWATH-MS technology, enables accurate and consistent measurement of proteins, which is crucial for comparative proteomics studies. However, there is lack of free and easy to implement data analysis protocols that can handle the different data processing steps from raw spectrum files to peptide intensity matrix and its downstream analysis. Here, we provide a data analysis protocol, named *diatools,* covering all these steps from spectral library building to differential expression analysis of DIA proteomics data. The data analysis tools used in this protocol are open source and the protocol is distributed at Docker Hub as a complete software environment that supports Linux, Windows and MacOS operating systems.

Keywords

Proteomics, Mass Spectrometry, DDA, DIA, SWATH-MS, Spectral Library, Data Analysis

Running head

Quantitative DIA data analysis protocol

1 Introduction

The current method of choice for large-scale identification and quantification of proteins is liquid chromatography tandem mass spectrometry (LC-MS/MS) ***(1)***. In addition to data-dependent acquisition (DDA) mode of mass spectrometry, there is an increased interest in data independent acquisition (DIA) mode, such as Sequential Windowed Acquisition of All Theoretical Fragment Ion Mass Spectra (SWATH-MS) ***(2)***.

DIA has been suggested to combine the advantage of the high-throughput of DDA proteomics with the benefit of the high reproducibility of targeted analysis, such as selective reaction monitoring (SRM) ***(2, 3)***. In SWATH-MS, all precursors generated from a sample are systematically fragmented within a predetermined mass-to-charge ratio (m/z) and retention-time range. Since the spectra are generated without explicit association between peptide precursors and corresponding fragments, a spectral library is used for the identification of peptides from the data. For building the spectral library, data from samples produced by mass spectrometry in DDA mode can be used.

Here, we provide a comprehensive data analysis protocol, named *diatools*, together with its open source implementation for analysing DIA data. The protocol covers all steps from raw spectrum files to a final result of differentially expressed proteins, with focus on SWATH-MS data. After installation of the required software and preparation of folder structure for the data (Section 3.1), the raw mass spectrum files are converted to required open formats (Section 3.2) and a database FASTA is constructed, which should contain sequences of all possible proteins that can be potentially found from the whole data set (Section 3.3). Section 3.4 then discusses the optional customization of the parameters of the protocol and Section 3.5 illustrates how to run the protocol to build the spectral library, to produce an intensity matrix of the identified peptides for each sample, and to perform differential expression

analysis between sample groups. The spectral library is built as described by Schubert et al. 2015 ***(4)***. The SWATH-MS data is processed using the OpenSWATH software ***(5)*** including a TRIC alignment step ***(6)***.

The *diatools* protocol is distributed as a Docker image at Docker Hub (compbiomed/diatools). Docker is a software technology that provides light-weight virtualized software environment enabling easy implementation of the data analysis protocol ***(7)***. The protocol supports Linux, Windows and MacOS operating systems, with the exception that Windows is needed to convert the raw spectrum files to open formats.

2 Materials

The raw mass spectrum files produced by mass spectrometers are typically in proprietary vendor-specific formats, which need to be converted to open formats before data analysis. The format conversion of the raw mass spectrum files can be done with the ProteoWizard software ***(8)*** on a Windows platform. Otherwise, our *diatools* data analysis protocol and all the required software are distributed as a Docker image and, therefore, can be run on any platform that supports Docker.

The *diatools* Docker image is available at Docker Hub (compbiomed/diatools). Additionally, the source code of our implementation is released under open source General Public Licence (GPL) 3.0 and can be downloaded from GitHub https://github.com/computationalbiomedicine/diatools.git. The Docker image is based on Ubuntu 17.04 operating system and it contains multiple proteomics tools, including OpenMS version 2.3 ***(9)***, Trans-Proteomics Pipeline (TPP) version 5.0 ***(10)***, msproteomicstools version 0.6.0, and ProteoWizard version 3.0.11252 ***(8)***. The downstream statistical analyses are performed using R and the R 3.3.2 environment with the appropriate packages is also included in the Docker image.

For running the *diatools* data analysis protocol, we recommend having at least 128 GB of RAM depending on the number of samples and the size of the sequence database (FASTA). The Docker image requires at least 30 GB of free disk space. Additionally, mass spectrometry raw data take typically a lot of space and, therefore, depending on the data, multiple terabytes of disk space might be required to store the input files.

The *diatools* protocol assumes that Biognosys iRT kit peptides are used in the laboratory protocol. The default settings are for the Q Exactive HF mass spectrometer (Thermo Fisher

Scientific, Waltham, Massachusetts, USA) but data from other instruments can be used as well by adjusting the parameters accordingly.

3 Methods

This chapter describes the steps of our *diatools* data analysis protocol, including data conversion, peptide identification, quantification, and differential expression analysis for the acquired data. A schematic illustration of the protocol is shown in **Figure 1**.

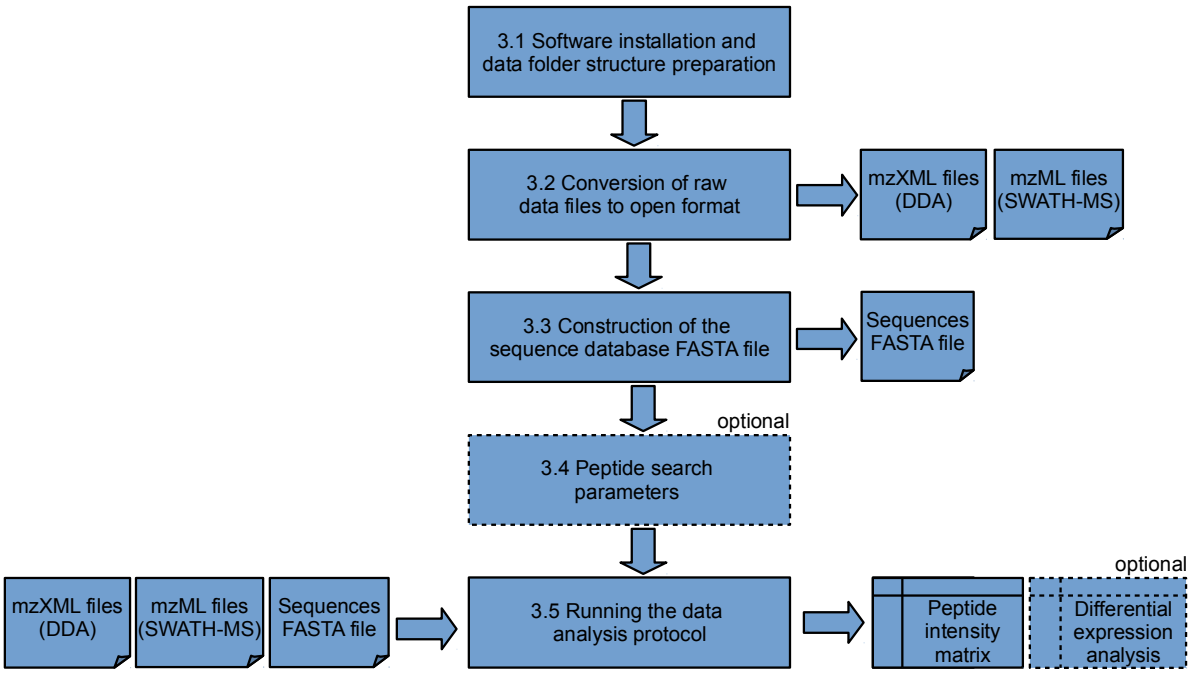

Figure 1. The *diatools* data analysis protocol for quantitative data-independent acquisition proteomics data. Protocol steps are shown with their corresponding section numbers and their input/output files. Optional steps are marked with dashed line.

3.1 Software installation and data folder structure preparation

Install the ProteoWizard software to the Windows machine that is used to convert the raw data files to an open format. The ProteoWizard software can be downloaded from http://proteowizard.sourceforge.net/.

Install Docker on the machine on which the data analysis protocol will be run. Docker installation package for Linux, Windows and MacOS can be downloaded from

www.docker.org. If the Linux distribution is Ubuntu, RedHat or CentOS Linux, the Docker can be installed from their respective software repositories (see **Note 1**). For Windows, the latest version 10 is recommended. On Windows, allow Docker to access the needed drive (for example C:) from the Docker settings.

Once Docker is installed, download the data analysis environment with the command:

```
docker pull compbiomed/diatools
```

On the machine where the data analysis is done, create a folder called *dataset* with the following subfolders under it:

- config
- DDA
- DIA
- Ref
- out

3.2 Conversion of raw data files to open format

Use the ProteoWizard tool to convert the raw data files to an open format.

To convert the raw DDA files to mzXML format, open Windows Command Prompt and go to the folder containing the DDA raw data. Run *qtofpeakpicker* from ProteoWizard to pick peaks and to convert the raw files to mzXML format:

```
FOR %i IN (*.raw) DO "\Program Files\ProteoWizard\ProteoWizard
3.0.11252\qtofpeakpicker.exe" --resolution=2000 --area=1 --
threshold=1 --smoothwidth=1.1 --in %i --out %~ni.mzXML
```

The default settings in the protocol are according those by Schubert et al. 2015 ***(4)***. If the ProteoWizard install location or version is different from the present protocol, modify the command accordingly.

To convert the raw SWATH-MS files to mzML format, use the MSConvert program from the ProteoWizard software with the following options:

- Output format: mzML
- Extension: empty
- Binary encoding precision: 64bit
- Write index: checked
- TPP compatibility: checked
- Use zlib compression: unchecked
- Package in gzip: unchecked
- Use numpress linear compression: unchecked
- Use numpress short logged float compression: unchecked
- Use numpress short positive integer compression: unchecked
- Only titleMaker filter

Copy the DDA mzXML files to the *dataset/DDA* folder and the SWATH-MS mzML files to the *dataset/DIA* folder.

3.3 Construction of the sequence database FASTA file

Create a sequence database in FASTA format that consists of all proteins that may exist in the sample set under analysis. The FASTA file is used to construct the spectral library by searching the DDA files against it. Create a FASTA file that contains the following protein or peptide sequences:

- Proteins of interest (for example Swiss-Prot Human)

- IRT peptides[1] (Biognosys|iRT-Kit_WR_fusion)
- Peptides related to lysis (Uniprot ID: Q7M135)
- Digestion enzyme (typically Trypsin (Uniprot ID: P00761))
- Possible contaminants.

Do not generate decoy sequences to the FASTA file manually. They are automatically generated by the protocol by reversing the peptide/protein sequences. Copy the FASTA file to the *dataset/ref* folder and name it as sequences.fasta.

3.4 Peptide search parameters

The default parameters of the protocol are for the nanoflow HPLC system (Easy-nLC1200, Thermo Fisher Scientific) coupled to the Q Exactive HF mass spectrometer (Thermo Fisher Scientific) equipped with a nano-electrospray ionization source. The device and lab protocol specific default settings are listed below:

- Precursor mass tolerance: 10 ppm
- Fragment ion tolerance: 0.02 Da
- Cleavage site: Trypsin_P
- Fixed modification: Carbamidomethyl (C)
- Variable modification: Oxidation (M)

If another type of instrument is used, these settings need to be customized (see **Note 2**).

3.5 Running the data analysis protocol

Open terminal prompt and set the working directory to *dataset/out* folder, where the LOCALPATH refers to the path to the previously created folder structure*:*

```
cd /LOCALPATH/dataset/out
```

[1] https://biognosys.com/media.ashx/irtfusion.fasta

Run the data analysis protocol with the following command:

```
docker run --rm \
-v /LOCALPATH/dataset/:/dataset \
--workdir /dataset/out \
-u $(id -u):$(id -g) \
compbiomed/diatools \
/opt/diatools/dia-pipeline.py \
--in-DDA-mzXML ../DDA/*.mzXML \
--in-DIA-mzML ../DIA/*.mzML \
--db ../ref/sequences.fasta \
--use-comet \
--use-xtandem
```

On a Windows platform, the path to the dataset is given in the following form: "`-v //c/LOCALPATH/dataset:/dataset`", where c is the drive letter. On Linux platform, Docker might be available only to superusers. In that case, add sudo command before the docker command.

To perform the optional differential expression analysis between sample groups, the groups must be provided using an additional parameter in the command:

```
--design-file <designFilename>
```

The design file must be defined as a tab-separated file (see **Table 1** for an example), where the column *Filename* refers to the SWATH-MS filename of a sample, the column *Condition* is the group to which the sample belongs, the column *BioReplicate* refers to the biological replicate, and the column *Run* to the MS run.

By default, the false discovery rate (FDR) used by the *diatools* protocol for the peptide identifications is 0.01. However, the FDR threshold can be adjusted by the user (see **Note**

3). The number of parallel processing threads used is four by default, but the user can choose different numbers of threads according to hardware resources (see **Note 4**). For the downstream differential expression analysis, the data are median normalized and differential expression analysis is performed for all possible pairs of sample groups using the PECA R-package ***(11)*** available from Bioconductor ***(12)***..

Once the data analysis run has completed successfully, the output folder *dataset/out* contains two tab-separated data files: DIA-peptide-matrix.tsv and DIA-protein-matrix.tsv. These files contain the peptides and proteins with their respective intensity values for each sample. The files can be opened with MS Excel or with LibreOffice Calc. The output folder contains also files of intermediate results written by various external tools run by the protocol as wells as a log.txt file which includes details on the run. The log can be used for troubleshooting if the run fails.

If the optional differential expression analysis is performed, those results are stored as tab-separated files in the *dataset/out* folder with the compared groups as filenames. The result files contain for each identified protein the protein name, the value of the test statistic (*t*), the number of peptides per protein (*n*), the significance *p*-value (*p*), and the estimated false discovery rate (*p.fdr*). In addition to performing the differential expression analysis by running the *diatools* protocol, it is possible to perform the differential expression analysis separately using the peptide intensity file produced by the protocol (see **Note 5**).

4 Notes

Note 1: Docker installation under Ubuntu, RedHat or CentOS. If the analysis is done on Ubuntu, RedHat or CentOS Linux distributions, the Docker can be installed from the software repository.

In Ubuntu, use the following shell command:

```
apt-get install docker.io
```

For Ubuntu, it is also convenient to add user to a system group called *docker*, which makes it possible to run the Docker without a *sudo* command.

In RedHat/CentOS, use the following commands:

```
yum install docker
systemctl enable docker
systemctl start docker
```

Note 2: Customization of peptide search parameters. The default parameters of the *diatools* protocol are for the nanoflow HPLC system (Easy-nLC1200, Thermo Fisher Scientific) coupled to the Q Exactive HF mass spectrometer (Thermo Fisher Scientific) equipped with a nano-electrospray ionization source. If another type of instrument is used, the settings need to be customized by editing the Comet and X!Tandem search engine parameters. This can be done by modifying the Comet and X!Tandem configuration files comet.params.template and xtandem_settings.xml, respectively, which are distributed with the protocol. Copy the modified files to the *dataset/config* folder and give the following extra parameters when running the protocol:

```
--comet-cfg-template config/comet.params.template
--xtandem-cfg-template config/xtandem_settings.xml
```

Note 3: Adjusting the false discovery rate (FDR) for the peptide identifications. By default, the protocol uses 0.01 as a FDR threshold for the spectral library building. For the TRIC alignment step, 0.01 is used as target and 0.05 as max threshold. These values can be adjusted by adding the following extra parameters when running the *diatools* protocol:

```
--library-FDR
--feature-alignment-FDR
```

For instance, the following parameter instructs to use 0.05 as the FDR threshold for spectral library building:

```
--library-FDR 0.05
```

Similarly, the following parameter instructs to use 0.01 as target and 0.02 as max threshold for the TRIC alignment:

```
--feature-alignment-FDR 0.01 0.02
```

Note 4: Adjusting the number of parallel processing threads. Currently, the protocol uses a maximum of four threads by default to process the data. If the protocol is run on a high-end desktop or on a server, the number of threads can be increased to correspond to the CPU core count. It speeds up the analysis, but also increases the amount of consumed RAM. For example, the following parameter increases the thread count to 20:

```
--threads 20
```

Note 5: Separate differential expression analysis using the peptide intensity file. In addition to performing the differential expression analysis by running the *diatools* protocol, it is possible to perform the analysis separately using the peptide intensity file produced by the protocol (peptide-intensity-matrix.tsv).

First, the peptide intensity data are transformed into a suitable format using the R/Bioconductor package SWATH2stats. To install SWATH2stats, open R and enter:

```
source("https://bioconductor.org/biocLite.R")
biocLite("SWATH2stats")
```

To read in the peptide intensity data, the following commands can be used:

```
library(data.table)
library(SWATH2stats)
data <- data.frame(fread(file=" peptide-intensity-matrix.tsv",
sep='\t', header=TRUE))
```

Next, get rid of unneeded columns (line 1), remove the iRT peptides that are used for retention time normalization (line 2), filter out rows corresponding to multiple proteins (line 3), and shorten the protein names (line 4):

```
data$run_id <- basename(data$filename)
data <- reduce_OpenSWATH_output(data)
data <- data[grep('iRT', data$ProteinName, invert=TRUE),]
data <- data[grep('^1/', data$ProteinName),]
data$ProteinName <- sapply(strsplit(data$ProteinName, "\\|"),
function(x) unlist(x)[2])
```

To define the sample groups, read in the design matrix, after which the data can be converted to a suitable format for the statistical analysis with PECA:

```
design <- read.table('design.txt', sep="\t", header=TRUE,
stringsAsFactors=FALSE)
data <- sample_annotation(data, design)
data <- convert4PECA(data)
```

PECA determines differential protein expression using directly the peptide-level measurements, instead of the common practice of using pre-calculated protein-level values. Differential expression statistic is first calculated for each measured peptide, after which the protein-level significance is estimated by taking the number of identified peptides per protein into account. To install PECA, open R and enter:

```
source("https://bioconductor.org/biocLite.R")
biocLite("PECA")
```

For DIA data, the reproducibility-optimized test statistic (ROTS) ***(13)*** is the suggested statistic to be used within PECA ***(14)***. This is done by setting the test parameter as `rots` when calling the function. The following commands perform the differential expression analysis for all possible pairs of sample groups:

```
library(PECA)
comb <- combn(unique(design$Condition),2)
for(i in 1:ncol(comb)) {
   group1 <- paste(design$Condition,
      design$BioReplicate,sep="_")
      [design$Condition==comb[1,i]]
   group2 <- paste(design$Condition,
      design$BioReplicate,sep="_")
```

```
      [design$Condition==comb[2,i]]
   peca.out <- PECA_df(data, group1, group2,
      id="ProteinName", normalize="median",
      test="rots", progress=TRUE)
   write.table(peca.out,
     file=paste("PECA_",comb[1,i],
     "-",comb[2,i],".txt",sep=""),
     sep="\t", quote=FALSE, row.names=TRUE, col.names=NA)
}
```

References

1. Aebersold R and Mann M (2003) Mass spectrometry-based proteomics. Nature 422:198–207
2. Gillet LC, Navarro P, Tate S, et al (2012) 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 11:O111.016717
3. Huang Q, Yang L, Luo J, et al (2015) SWATH enables precise label-free quantification on proteome scale. Proteomics 15:1215–23
4. Schubert OT, Gillet LC, Collins BC, et al (2015) Building high-quality assay libraries for targeted analysis of SWATH MS data. Nat Protoc 10:426–41
5. Röst HL, Rosenberger G, Navarro P, et al (2014) OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. Nat Biotechnol 32:219–223
6. Röst HL, Liu Y, D'Agostino G, et al (2016) TRIC: an automated alignment strategy for reproducible protein quantification in targeted proteomics. Nat Methods 13:777–83
7. Merkel D (2014) Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux J 2014
8. Chambers MC, Maclean B, Burke R, et al (2012) A cross-platform toolkit for mass spectrometry and proteomics. Nat Biotechnol 30:918–20
9. Sturm M, Bertsch A, Gröpl C, et al (2008) OpenMS - an open-source software framework for mass spectrometry. BMC Bioinformatics 9:163
10. Deutsch EW, Mendoza L, Shteynberg D, et al (2010) A guided tour of the Trans-Proteomic Pipeline. Proteomics 10:1150–9
11. Suomi T, Corthals GL, Nevalainen OS, et al (2015) Using Peptide-Level Proteomics Data for Detecting Differentially Expressed Proteins. J Proteome Res 14:4564–4570
12. Huber W, Carey VJ, Gentleman R, et al (2015) Orchestrating high-throughput genomic analysis with Bioconductor. Nat Methods 12:115–121

13. Elo LL, Filén S, Lahesmaa R, et al (2008) Reproducibility-optimized test statistic for ranking genes in microarray studies. IEEE/ACM Trans Comput Biol Bioinform 5:423–31

14. Suomi T and Elo LL (2017) Enhanced differential expression statistics for data-independent acquisition proteomics. Sci Rep 7:5869

Table 1. Example design file.

Filename	Condition	BioReplicate	Run
Sample1.mzML	Treatment	1	1
Sample2.mzML	Treatment	2	2
Sample3.mzML	Treatment	3	3
Sample4.mzML	Control	1	4
Sample5.mzML	Control	2	5
Sample6.mzML	Control	3	6