# DIA-NN manual

## Table of Contents

## Introduction

DIA-NN is an open-source software tool that incorporates all stages of raw DIA/SWATH-MS proteomics data processing in a single program. DIA-NN is designed to be easy to use. It is also very fast and has relatively low hardware requirements, allowing for processing of large-scale experiments on average desktops/laptops. As input, DIA-NN takes a set of raw data files (in .mzML or its own .dia format) and a spectral library (in a simple text format). As output, it produces precursor ion identifications (exact retention times) along with precursor and protein quantities. Cross-run normalisation and cross-run retention time profiling are supported.

## Downloading

DIA-NN Windows x64 executable can be downloaded here (tested on Windows 7 & 10), DIA-NN source code is available here. DIA-NN does not require installation.

## Building

A Visual Studio 2017 solution file is provided along with the source code for building on Windows. Windows was chosen as the primary platform for the development of DIA-NN due to the fact that most contemporary vendor software as well as software for conversion of vendor raw data formats into .mzML is Windows-only. However, it should be possible (not tested) to build DIA-NN on Linux, as all the libraries it relies on are Linux compatible. In order to build DIA-NN on OS X or in case of a difficulty with building it on Linux, .mzML support can be disabled (comment the "define MZML" line in /src/diann.cpp).

## Running

DIA-NN can be run as any Windows Console application. For example, navigate to the folder in which DIA-NN executable has been placed, right click on the empty space, choose "New" and then "Text Document". Double click on the resulting text file and type "diann.exe [commands]", where [commands] is the list of commands to be passed to DIA-NN (see below). Close the file and rename it to change the file extension to .bat (to do this, make sure that file extensions are displayed by Windows). To run DIA-NN with the commands specified, double click on this file.

Minimal set of commands required by DIA-NN comprises the `--f` (for each file to be processed) and `--lib` commands. The order of commands supplied to DIA-NN can be arbitrary.

```
diann.exe --f C:\Data\run1.mzML --f C:\Data\run2.mzML --lib library.csv
```

The above command will instruct DIA-NN to process run1.mzML and run2.mzML located in C:\Data\ using library.csv as the spectral library. The output report file quant.csv will be generated in the current working directory (the one with diann.exe).

In the process, DIA-NN will also create files run1.mzML.quant and run2.mzML.quant in the same directory. These files are small in size (typically orders of magnitude smaller than the raw data files) and contain the precursor identifications in an internal binary format. All the information DIA-NN needs to generate the final report is contained in these files. Therefore,

multiple runs can be processed separately, with information from the respective .quant files quickly compiled at a later stage (see the `--use-quant` and `--report-only` commands below).

The memory requirements of DIA-NN are proportional to the spectral library size. Processing of typical yeast 20-40 min gradient runs with a 30000 precursor spectral library requires approximately 3Gb RAM. Processing of complex mixed human-yeast-*E.coli* 2-hour gradient runs (which we used to benchmark the quantification performance of DIA-NN) with a spectral library containing 45000 precursors might require more than 10Gb RAM. However, the amount of RAM required can be reduced dramatically with the use of .ref files that contain expected high confidence identifications (see the `--update-ref` and `--use-ref` commands below).

## Raw data files

Raw mass spectrometry data files should be converted to the centroided .mzML data format. This is easily achieved with MSConvert (GUI or command line tool), which is part of the ProteoWizard project. In our tests, we converted Sciex .wiff files with MSConvert GUI (vendor peak picking turned on for MS levels 1 and 2, 32-bit binary precision, all other options turned off). Although DIA-NN is capable of accessing .mzML files, we recommend further converting the data to its internal .dia format. This reduces the time needed to open the files from over a minute to several seconds. Conversion is performed by calling DIA-NN with the `--convert` command.

```
diann.exe --f C:\Data\run1.mzML --f C:\Data\run2.mzML --convert
```

The above command will convert run1.mzML and run2.mzML, generating run1.mzML.dia and run2.mzML.dia in C:\Data\.

## Spectral library

Spectral libraries exported from Spectronaut (Biognosys AG) (converted from .xls to .csv with e.g. Microsoft Excel) or created by ConvertTraMLtoTSV (which is part of OpenMS) (should be in tab-separated text (.tsv) format only) can be used by DIA-NN directly.

In general, DIA-NN expects a spectral library that is a text file, in which each line, except for the header (that contains column names), represents a certain fragment of a precursor ion. Columns with the following information are expected to be present in the file: protein group ID, modified peptide sequence, precursor ion charge, precursor ion m/z, precursor retention time, fragment ion m/z, relative fragment ion intensity, and (optionally), an indicator (1 or 0) whether the precursor ion is a decoy. If the column names do not match those in one of the formats mentioned above, custom column names can be specified with the `--library-headers` command, after which column names are listed separated by commas.

```
diann.exe --f C:\Data\run1.mzML --f C:\Data\run2.mzML --library-headers
Protein,ModifiedPeptide,PrecursorCharge,PrecursorMz,RT,FragmentMz,FragmentI
ntensity --lib library.csv
```

The above command will make DIA-NN look for the specified headers instead of the standard ones in the spectral library file library.csv.

The following peptide modifications are supported by default: cysteine carbamidomethylation (UniMod:4), methionine oxidation (UniMod:35) and protein N-terminal acetylation (UniMod:1). Custom modifications of individual amino acids can be added with the `--mod` command (see below). If a set of decoy precursors matching the target precursors is provided (and hence there is the respective column in the library file), then DIA-NN will not need to recognise modifications and thus specifying them will be unnecessary.

## Commands

### 1. Essential

```
--f <data file>
```

Specifies a data file to be processed. Use --f for each file to be processed.

```
--lib <spectral library file>
```

Specifies the spectral library.

### 2. Auxiliary

```
--threads <thread number>
```

Specifies the number of CPU threads to be used.

```
--convert
```

With this option DIA-NN converts .mzML files (specified using the --f command) to the .dia format. Unlike .mzML files, .dia files can be loaded quickly (seconds), so it is recommended to convert files that are going to be analysed multiple times.

```
--ext <string>
```

Add a string to the end of each file name (specified with --f).

```
--prefix <string>
```

Add a string to the beginning of each file name.

```
--library-headers <protein id header>,<modified peptide
header>,...,<fragment intensity header>
```

Looks for the specified headers instead of the standard ones in the spectral library. Headers are specified in the order: protein ID, modified peptide sequence, precursor ion charge, precursor ion m/z, precursor retention time, fragment ion m/z, relative fragment ion intensity, and (optionally), an indicator (1 or 0) whether the precursor ion is a decoy.

```
--mod <modification name>,<modification mass>
```

Add a modification with the specified name and mass delta to the list of modifications.

```
--out <output file>
```

Specifies the output file; by default, the output is saved to quant.csv in the current working directory.

```
--output-headers <file name header>,...,<RT header>
```

Specifies the column names in the output file to be used instead of the standard ones. The order of the names is: file name, protein group ID, protein group quantity, modified peptide sequence, precursor ion ID, precursor ion charge, q-value, precursor ion quantity, inferred retention time.

`--verbose <N>`

The detail level for the processing progress information reported by DIA-NN. Default N = 1. N = 0 corresponds to no output, N = 5 corresponds to all the information being displayed. When cross-run neural network training is used (`--global-nn`), N = 5 leads to high RAM usage. N = 4 or 5 might also slightly slow down the processing.

`--use-quant`

Use existing .quant files when available.

`--report-only`

Generate a report using the existing .quant files for all the files specified with `--f`.

### 3. Algorithms

`--global-nn`

Use cross-run neural network training.

`--nn-bagging <N>`

Train N neural networks in parallel. Default N = 12. The more the better. Different CPU threads are used to train different networks, so there is no performance penalty for the use N less or equal to the number of threads.

`--nn-epochs <N>`

Train the run-specific neural network for N epochs. Default N = 50.

`--nn-global-epochs <N>`

Train the cross-run neural network for N epochs. Default N = 50.

`--nn-hidden <N>`

Set the number of hidden layers in the run-specific neural network to N. Default N = 2.

`--nn-global-hidden <N>`

Set the number of hidden layers in the cross-run neural network to N. Default N = 5.

`--nn-learning-rate <X>`

Set the learning rate (per epoch) to X. Default X = 10.0.

`--nn-reg <X>`

Set the L2 regularisation strength to X. Default X = 0.0.

`--standardise`

Use standardisation of the input for neural network training.

`--window <N>`

Force the size of the chromatogram scan window to be equal to 2N+1 scans.

`--no-window-inference`

Instead of chromatogram scan window size inference based on averaged peak width, wide window calculated based on the data acquisition duration will be used.

`--individual-windows`

Chromatogram scan window sizes will be inferred separately for different runs. By default, the inference is performed for the first run in the list only and then the calculated window size is used for all the runs.

`--mass-acc <X>`

Set mass accuracy to X (in parts per million). Default X = 20.0.

`--zero-mass-delta`

Disable mass calibration. This speeds up the program considerably when not using a set of reference precursors. Not recommended if the instrument has not been calibrated.

`--iter <N>`

Total number of iterations used to refine elution peak selection. Default N = 8.

`--nn-iter <N>`

Use neural network classifier starting with iteration number N (with the first iteration having number 0). Default N = 6.

`--no-nn`

Turn the neural network classifier off: only linear classifier will be used. This is recommended for data files in which one expects to identify a very low number of precursor ions (tens or hundreds), as a neural network requires a substantial number of true IDs to be trained on.

`--test-proportion <X>`

Set the fraction of library precursor ions that will be used as the test dataset. Default X = 0.25.

`--no-test-dataset`

Use all the library precursor ions to train the classifier.

`--norm-fraction <X>`

Fraction of top precursor ions used for cross-run normalisation. Default X = 0.4.

`--norm-qvalue <X>`

Q-value cutoff for precursor ions used for cross-run normalisation. Default X = 0.001.

`--no-norm`

No cross-run normalisation.

`--quant-qvalue`

Q-value cutoff used to rank precursors for cross-run precursor quantification.

```
--rt-profiling
```

Use cross-run retention time profiling.

```
--profiling-qvalue <X>
```

Q-value cutoff for retention time profiling. Default X = 0.01.

```
--update-ref
```

Process a run and generate a .ref file in the same folder as the spectral library. If, for example, the library file name is "library.csv", the file will be named "library.csv.ref". This file will contain a list of confidently identified precursor ions that can be later used to speed up the data processing (with none or negligible difference in the identification performance) and reduce RAM usage.

```
--use-ref
```

Use a .ref file generated previously (for the spectral library in question). This will speed up the processing, as elution peaks for all other precursors (not listed in the .ref file) will be searched for only within a certain retention time window. The amount of RAM required will also be reduced significantly.

```
--ref <file name>
```

Specify a text file with a list of precursor ions to be used instead of a .ref file. The format of the text file should be the same as that of the spectral library.

**4. Grouping commands**

```
--cfg <config file>
```

Specifies a file with a set of commands. Grouping commands in a file is often convenient: for example, commands specifying processing settings and raw experiment files can be placed in separate config files.

nn-global-10-yeast.txt:

```
--threads 4 --nn-epochs 10 --global-nn --lib yeast_spectral_library.csv
```

experiment1.txt:

```
--ext .mzML --prefix C:\Data\ --f run1 --f run2 --out
C:\Processed\experiment1.csv
```

run-diann.bat:

```
diann.exe --cfg nn-global-10-yeast.txt --cfg experiment1.txt
```

**5. Other commands** for fine tuning of DIA-NN can be found in the arguments() function (/src/diann.cpp).