DIA-NN GUI manual[1]

version 10/10/2020

Contents

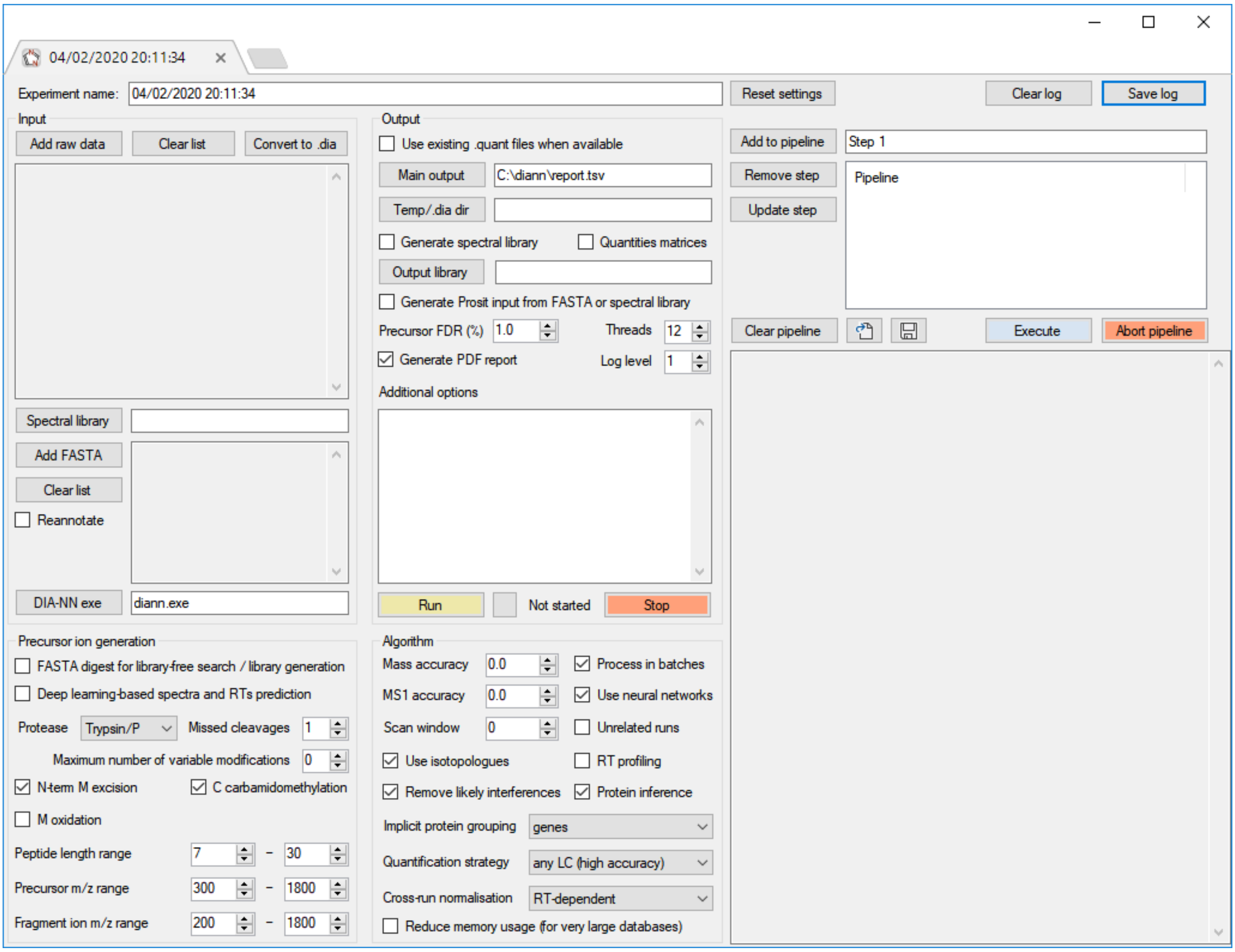
[1] Please email any comments, suggestions, questions or feedback to Vadim Demichev (vadim.demichev[at]gmail.com) or create a new issue on GitHub (https://github.com/vdemichev/diann).

Introduction

DIA-NN is a fast and easy to use tool for processing data-independent acquisition (DIA or SWATH) proteomics data. DIA-NN is designed to do as much as possible automatically, eliminating the need to optimise the processing parameters for each experiment.

DIA-NN distribution contains DiaNN.exe, which is a command line tool, and DIA-NN.exe, a GUI wrapper for this tool. The GUI works by launching DiaNN.exe and displaying its output. A report (with precursor ion and protein identification and quantification information) is generated by the command line tool. Optionally, a spectral library can be generated. Any analysis that can be performed with the GUI can also be performed with the command line tool and vice versa. Here we provide detailed information on data formats supported by DIA-NN and consider the ways to fine-tune its performance.

GUI window



Experiment name: 04/02/2020 20:11:34

□ Reannotate

DIA-NN exe: diann.exe

□ Use existing .quant files when available

Main output: C:\diann\report.tsv

□ Generate spectral library □ Quantities matrices

□ Generate Prosit input from FASTA or spectral library

Precursor FDR (%) 1.0 Threads 12

☑ Generate PDF report Log level 1

Step 1

□ FASTA digest for library-free search / library generation

□ Deep learning-based spectra and RTs prediction

Protease Trypsin/P Missed cleavages 1

Maximum number of variable modifications 0

☑ N-term M excision ☑ C carbamidomethylation

□ M oxidation

Peptide length range 7 – 30

Precursor m/z range 300 – 1800

Fragment ion m/z range 200 – 1800

Mass accuracy 0.0 ☑ Process in batches

MS1 accuracy 0.0 ☑ Use neural networks

Scan window 0 □ Unrelated runs

☑ Use isotopologues □ RT profiling

☑ Remove likely interferences ☑ Protein inference

Implicit protein grouping genes

Quantification strategy any LC (high accuracy)

Cross-run normalisation RT-dependent

□ Reduce memory usage (for very large databases)

The GUI window features four panels used to specify the data and processing settings: **Input**, **Output**, **Precursor ion generation** and **Algorithm**. To the right, there is a pipeline editor at the top and a panel used to display the output of the command-line tool at the bottom. Extra commands entered in the **Additional options** text box (**Output** panel) will also be passed to the command line tool; quotes can be used when passing options but are not required unless file names featuring multiple adjacent spaces need to be specified. **See the project description on GitHub for the information on DIA-NN's command line syntax and some useful commands that are not described in this manual**. Most GUI elements feature tooltips with a short description of their function, activated by hovering a mouse cursor over the element.

Input

Sciex .wiff, Thermo .raw, .mzML and .dia (native DIA-NN format) files are supported. Reading Sciex .wiff files requires the DIA-NN.Wiff.dll file (included in the DIA-NN distribution) as well as the Sciex DLL libraries to be present in the same folder as DiaNN.exe. For example, one can first install ProteoWizard (which includes the Sciex DLLs) and then install DIA-NN in the same folder using the installer provided (DIA-NN-Setup.msi). Reading Thermo .raw files requires Thermo MS File Reader to be installed. Please use specifically the version MS File Reader 3.0 SP3[*]. MSX-DIA .raw files are currently not supported. It is highly recommended that .mzML files are centroided, preferably using the vendor centroiding algorithm for both MS1 and MS2 spectra. DIA-NN has been tested on .mzML files produced from Sciex .wiff and Thermo .raw by MSConvert GUI (part of the ProteoWizard project) with 32-bit binary precision, vendor centroiding enabled and all other options except "Write index" disabled.

DIA-NN can convert raw files to its own .dia format. This format allows very quick loading, essentially limited only by the hard drive speed. Converted files are placed either in the same location as the raw files, or to a folder specified using the **Temp/.dia dir** field in the **Output** panel.

Spectral library

Used to specify the spectral library. The library is required unless a library-free search against a FASTA database is to be performed. In that case, if specified, the spectral library will be used to enhance and speed up library-free search (experimental). DIA-NN accepts spectral libraries in a plain text format (tab-separated or comma-separated, i.e. .tsv, .csv, .xls or .txt) as well as in its own compact binary format (.speclib).

Libraries in the PeakView format as well as libraries produced by FragPipe, TargetedFileConverter (part of OpenMS), exported from Spectronaut (Biognosys) in the .xls format or generated by DIA-NN itself are supported "as is". Support for NIST .msp and SpectraST .sptxt libraries (both must have fragment ion types & charges annotated) is experimental. Use `--sptxt-acc` to set the fragment filtering mass accuracy (in ppm) when reading .sptxt/.msp libraries.

For libraries generated by other means, DIA-NN needs to be told the header names (separated by commas) (for the columns it requires) using the `--library-headers` command. Use the * symbol instead of the name of a header to keep its recognition automatic. See below the descriptions of the respective columns (in the order the headers need to be specified).

- **Modified peptide sequences.** Outside parentheses, all symbols apart from letters are ignored. Modifications are specified within parentheses ("()" or "[]") after the modified amino acid or (e.g. "[Acetyl (Protein N-term)]") before the first amino acid. Recognition of some common modifications is built in. Custom modifications can be specified using the `--mod` command, e.g. "`--mod UniMod:5,43.005814`" will add carbamylation (KR) to the list of modifications recognized by DIA-NN. Specifying "label" after the modification mass will tell DIA-NN that the modification in question is an isotopic label, e.g. "`--mod UniMod:259,8.014199,label`" would correspond to Lysine +8 SILAC label (SILAC recognition is built-in though, so no need to add it separately). DIA-NN does not need to recognise modifications if fragment charge and type (either y-series or b-series) information is provided (this is usually the case) or if decoys are provided in the library (see below). The `--clear-mods` command makes DIA-NN forget all the built-in modifications (spectral library-based search only), so that the output would contain modification names as specified in the spectral library, whereas by default DIA-NN produces output with modification names in UniMod format. Alternatively, the `--original-mods` command only disables conversion of modifications to the UniMod format. The `--full-unimod` command loads the complete UniMod modification database and disables conversion.

- **Precursor charge.**

[*] MS File Reader 3.0 SP3 download link: https://thermo.flexnetoperations.com/control/thmo/login?nextURL=%2Fcontrol%2Fthmo%2Fdownload%3Felement%3D6306677

- **Precursor m/z.**
- **Reference retention time.** There are no restrictions on the retention time scale used here.
- **Fragment ion m/z.**
- **Relative intensity of the fragment ion.** There are no restrictions on the scale used here.

The following optional headers can also be specified:

- **UniProt identifiers** of the proteins matched to the peptide (separated by semicolon). These will be used for peptide annotation as well as protein quantification. If a FASTA file is specified, UniProt identifiers will be searched against this file and annotated with protein and gene names. If the library contains no column specifying which peptides are proteotypic (see below), DIA-NN will consider any peptide matched to a single UniProt identifier proteotypic. DIA-NN does not check the validity of the UniProt identifiers provided, i.e. arbitrary text can be used instead.
- **Protein names.** These will be used for annotation if a FASTA file is not provided and protein inference is turned off.
- **Gene names.** These will be used for annotation if a FASTA file is not provided and protein inference is turned off.
- **Proteotypicity.** The value should be 1 if the peptide is to be considered proteotypic (i.e. there is only a single protein from which it can originate). DIA-NN calculates protein q-values using proteotypic peptides only.
- **Decoy.** Indicates whether the peptide is a decoy. If there are decoy peptides in the library, DIA-NN uses these and does not generate its own decoys.
- **Fragment ion charge.** If this column is not provided, DIA-NN infers the charges itself, although this inference relies on recognition of all peptide modifications and is not guaranteed to be correct in all the cases. However, DIA-NN needs to know fragment ion charges if (1) the library contains modifications DIA-NN does not recognise, and these have not been added with the `--mod` command, and decoys are not provided in the library or (2) DIA-NN is told to consider isotopologues of the peptides and fragments in the library (this is enabled by default).
- **Fragment ion type.** Should be ‘y’ or ‘b’. DIA-NN needs to know the fragment ion type if the library contains modifications DIA-NN does not recognize, and these have not been added with the `--mod` command, and decoys are not provided in the library.
- **Fragment series number**, **Fragment loss type**, **Q-value.**
- **Elution Group.** This column specifies a unique string identifying the elution group, e.g. the set of all precursor ions which are expected to have exactly the same retention time. If not provided, DIA-NN infers elution groups automatically, assuming ‘co-elution’ of precursors corresponding to the same peptide with different charge states or isotopic labels (relying on the built-in SILAC recognition and on label information provided when supplying the `--mod` command – see above). The (experimental) `--peak-translation` command will make DIA-NN force the same retention time for precursors in an elution group, improving identification performance. Heavy/light ratios (equivalent to absolute quantities when used with spike-in quantification standards, e.g. PlasmaDive or PQ500) will also be calculated (more information in the description of the Label.Ratio output column, see below).
- **An indicator whether the fragment ion should be excluded when quantifying precursors**. When using the `--restrict-fr` option, some fragments will not be used for quantifying precursors, based on the value in this library column (True or False). When used in conjunction with `--no-fr-selection`, thus disabling automatic selection of fragments for quantification, this allows to precisely control how

precursors are quantified in a particular run, independently from other runs in the experiment. The `--gen-fr-restriction` option allows to annotate the current spectral library with fragment exclusion information, based on the runs being analysed (fragments least affected by interferences are selected for quantification, why the rest are excluded). Note that DIA-NN would only use the top 6 fragments (based on their reference library intensities) detectable in a particular run (e.g. their m/z values being within its MS/MS range) for quantifying precursors (and some of these might be excluded based on the library annotation). If all fragments selected for quantification in the cross-run manner happen to have zero intensities, the precursor quantity would thus be set to zero.

Thus, `--library-headers *,*,Precursor_Mz,*,Fragment_Mz,*,Protein_Ids`, for example, would instruct DIA-NN to recognise "Precursor_Mz" and "Fragment_Mz" as the headers for the columns containing precursor and fragment ion m/z values, and "Protein_Ids" as the header for the column containing UniProt identifiers. Importantly, rows describing fragment ions corresponding to the same precursor must be grouped together (this is the case with all software-generated libraries).

After loading a library in a text format, DIA-NN always automatically saves it in the compact .speclib format to the same folder. Any text/.speclib library can also be exported to an OpenMS-compatible text format by selecting the **Generate spectral library** option in the **Output** panel and running DIA-NN without specifying any raw data files (as otherwise DIA-NN would only save precursors identified in these files and would refine the spectra and retention times based on these identifications – see below). Note that when saving libraries in text format, DIA-NN rounds all real numbers to 8 significant figures. Only fragments annotated with the fragment type and charge information or fragments recognized by DIA-NN automatically (i.e. y/b-series fragments, potentially with a single H2O, NH3 or CO neutral loss) will be exported. Commands `--min-fr` and `--max-fr` allow to specify the minimum and maximum numbers of fragments per precursor in the exported library. For example, `--export-library --min-fr 6 --max-fr 12` will instruct DIA-NN to discard all precursors with less than 6 fragments annotated and retain only top 12 fragments, based on their reference intensities. Such filtering might be useful e.g. to reduce the size of libraries produced by Prosit. Usually, there is little benefit in using more than 6 fragments and no or almost no benefit in using more than 12.

FASTA

One or several FASTA files can optionally be specified. These will be used for protein annotation. If the library does contain information on protein IDs associated with each peptide, these should match the protein sequence IDs in the FASTA database, e.g. if the FASTA database is in the UniProt format, the library should also contain UniProt IDs. If the library does not contain protein information (e.g. Prosit output library) or the **Reannotate** option is used, protein information will be extracted from the FASTA database by digesting the latter *in silico* (using the currently specified protease settings, see below) and matching library peptides to the digest.

If library-free search is used (activated by enabling FASTA digest in the **Precursor ion generation** panel), the FASTA files will be digested *in silico* to generate a spectral library, which DIA-NN will then use to analyse the acquisitions.

Optionally, peptides considered when loading FASTA databases can be filtered using the `--fasta-filter` command, e.g. "`--fasta-filter C:/human_peptides.fasta`". Peptides can be provided as a fasta file (e.g. DIA-NN recognises the format of peptide lists from PeptideAtlas builds) or as a simple list of stripped sequences (one per line); all lines starting with '>' are ignored. Filtering allows to dramatically reduce the library-free search space, speeding up the analysis and improving identification performance. The `--force-swissprot` command instructs DIA-NN to only consider SwissProt proteins (in UniProt format).

Output

During the analysis, DIA-NN generates a .quant file containing identification and quantification information for each run analysed. By default, these .quant files are placed to the same location as the raw files. Thus, it is not recommended to analyse the same raw data files simultaneously with multiple instances of DIA-NN (unless .quant files are saved to a separate location or stored in memory – see below), although multiple instances of

DIA-NN can be used to analyse different experiments. The size of the .quant files is roughly proportional to the number of precursors identified at the given FDR setting and is negligible unless a library-free search without any FDR filtering is used (not recommended). The .quant files can be reused to speed up reanalysis of the data, but only if the spectral library and FASTA files specified as well as the FDR filtering settings are exactly the same as the settings that were used to generate these .quant files.

An alternative location for .quant files can be provided by specifying the **Temp/.dia dir** in the **Output** panel. The `--no-quant-files` command instructs DIA-NN to store .quant files in memory, without saving to disk. These options allow to analyse raw data files stored in a write-protected folder, e.g. in a network location.

Main output

DIA-NN saves its report as a simple table in the tab-separated format (.tsv). Default output header names can be replaced (in the full report only) with those specified with the `--output-headers` command, analogous to `--library-headers` (see above). Along with the main report, DIA-NN produces a file [main report name].genes.tsv containing only gene-level information. The `--compact-report` command makes DIA-NN retain only the most important columns in the report, thus reducing the file size. The command `--report-lib-info` adds library fragment annotation to the report. Choosing **Quantities matrices** will generate precursor/protein x samples expression level matrices, in addition to the main report. These contain 'Normalised' quantities (see below) for precursors and protein groups, MaxLFQ quantities for genes. By default, 1% q-value is used at both precursor and protein (group) level, this can be adjusted using the `--matrix-qvalue` command, e.g. `--matrix-qvalue 0.02` would set the q-value threshold to 2%.

Protein.Group

DIA-NN aims to reduce the number of proteins matched to each precursor ion. Maximal parsimony approach is implemented using a greedy set cover algorithm. Furthermore, TrEMBL proteins can be omitted, if the precursor in question is also matched to at least one SwissProt protein (this can be disabled with `--no-swissprot`). If protein inference is turned off, protein grouping provided in the spectral library is used instead of DIA-NN's algorithm.

Protein.Names and Genes

Protein names and genes corresponding to proteins listed in the Protein.Group column.

Protein.Q.Value

The best q-value across all proteins matched to the precursor ion. Protein q-values are calculated separately for all protein isoform names, protein names or genes (not protein groups or gene groups) – depending on the implicit protein grouping setting. Only proteotypic peptides are used to calculate q-values. Thus, when DIA-NN reports the number of proteins identified at 1% FDR (for each run separately, once it is analysed), it underestimates the number of protein groups that are actually detected. Importantly, proteins with no proteotypic peptides found always have q-value equal to 1.0 (= 100%).

PG.Q.Value and GG.Q.Value

Analogous to Protein.Q.Value but calculated for the entire protein group/gene group, rather than for uniquely identified proteins only. All precursors matched to the group are considered. For most datasets it does make sense to filter the final report at 1%-5% PG.Q.Value or GG.Q.Value. DIA-NN itself can also be instructed to do that using the `--protein-qvalue` command, e.g. `--protein-qvalue 0.01`.

Quantities

PG.Quantity – protein group (i.e. the set of proteins listed in Protein.Group) quantity, Genes.Quantity – gene group (i.e. the set of genes listed in Genes) quantity. Basic quantification is performed using the top N method (with N = 1 by default), i.e. the intensities of the top N precursor ions (matched to the protein/gene group) identified at 1% FDR are summed for each run separately. The `--top` command allows to specify N, e.g. `--top 3` would set N to 3. Normalised – the same quantities with cross-run normalisation applied (on the precursor level). The first normalisation step is performed globally using the top 40% least variable (across runs) precursors identified at 1% FDR. These figures can be adjusted, e.g. `--norm-fraction 0.3 --norm-qvalue 0.001`.

Subsequently, local (in terms of RT; can be turned off with `--global-norm` – see below) normalisation is performed.

DIA-NN also calculates normalised gene group quantities using the MaxLFQ algorithm: Genes.MaxLFQ. Another quantity, Genes.MaxLFQ.Unique, is calculated using proteotypic peptides only. For most experiments it is recommended to use one of these MaxLFQ quantities (they are usually significantly better than quantities obtained with the Top N method). For experiments of several hundred samples or more it makes sense, however, to first apply batch correction on the precursor level (using R/Python), filter precursors based on their detection rates/CV values in QC samples, and quantify proteins afterwards (e.g. using the "diann" (recommended, https://github.com/vdemichev/diann-rpackage) or "iq" R packages).

Label.Ratio
Experimental. Only calculated when using `--peak-translation`. An estimate of the ratio between the quantity of the precursor and the sum of quantities of all other precursors in the same elution group and with the same charge (i.e. the only difference between the precursors is the isotopic label). DIA-NN aims to estimate the ratio only with the use of fragments that are (i) not shared by the respective differentially labelled peptides (ii) least affected by interferences. Label.Ratio can be used for improved relative quantification using SILAC or for absolute quantification using spike-in quantification standards. For LabelRatio to be calculated with high precision, the labelled and unlabelled precursors should have the same spectra (i.e. if the light peptide has fragment y7-H2O annotated with reference intensity 0.6, the heavy one should also feature y7-H2O with reference intensity 0.6 – and vice versa: this is logical, as one would expect identical fragmentation of precursors which differ only in the presence of heavy isotopes).

Quantity.Quality
A metric which reflects the reliability of the precursor quantity calculated for the particular run. Should be interpreted the following way: values close to 1.0 likely (but not with 100% confidence) reflect high quality signal; values significantly below 1.0 and especially values below 0.5 reflect low quality signal. This can be used to filter unreliable quantities from the output.

Extra information
CScore and Decoy.CScore columns provide information on the scores for each precursor ion and the respective decoy, which were used to calculate the q-values. Fragment.Quant.Raw, Fragment.Quant.Corrected and Fragment.Correlations columns list raw and interference-corrected fragment intensities and correlation scores (higher = better) (the fragments being ordered by their library intensities from highest to lowest). This information allows to easily direct DIA-NN output to scripts that perform custom protein inference, quantification, etc.

Quality control
Along with the main report, DIA-NN produces a file [main report name].stats.tsv with some quality control information for all the samples in the experiment. For example, it contains such information as the precursor and protein identification numbers, total MS1 and MS2 signal, total quantity (sum of all precursor quantities), average full width at half maximum (FWHM) for chromatographic peaks, expressed both as the number of SWATH scans and as time in minutes, mass accuracy correction data as well as retention time prediction accuracy, average precursor length, charge and the number of missed tryptic cleavages. The generation of this file can be turned off with the `--no-stats` command.

Generating a spectral library

Precursor ions identified at the specified FDR threshold will be used to generate a new spectral library. The new spectral library is saved in an OpenMS-compatible format. This feature allows to (i) generate spectral libraries directly from DIA data using library-free search (see below); (ii) optimise the spectra and retention times in a spectral library for the specific LC-MS setup: this can significantly decrease the numbers of missing values. DIA-NN attempts to use the reference (e.g. iRT) retention time scale in the newly generated library whenever possible, e.g. when analysing with a spectral library, using deep learning-based spectra / RTs prediction or using a "training" library (see below). On the other hand, library-free analysis without any spectra / RTs prediction method will result in experimental retention times being used instead. Sometimes, experimental RTs might be better than the reference alignment generated by DIA-NN, e.g. when analysing consecutive injections of exactly the same sample (e.g. in gas-phase fractionation mode) on a stable LC. To force saving experimental RTs in the newly generated library, use the `--out-measured-rt` command.

In most cases, when performing library-free analysis of multiple relatively heterogeneous samples, it is better to use two-step analysis: the first step involves creating a spectral library directly from these runs, while the second – using this library to reanalyse the runs (RT Profiling should be off during both steps). Such a method is particularly beneficial if some proteins of interest exhibit large fold changes between different runs of the experiment, although it might result in slightly less conservative FDR values being reported. When applying this technique, the final output can also be filtered using the Lib.Q.Value column, which reflects global FDR calculated for the particular precursor during the library-generation step. For example, in our large-scale plasma experiments we currently consider only precursors with Lib.Q.Value <= 0.005.

Note: information stored in the Lib.Q.Value column is currently not retained when the library is converted to .speclib. The original library in the .tsv format should therefore be used (recommended) in order for DIA-NN to take into account library q-values when assessing the confidence of peptide and protein IDs.

Generating Prosit input

This option allows to generate a .csv file to be used as input for the Prosit online tool (currently available at https://www.proteomicsdb.org/prosit/) from either a FASTA database (digested *in silico* using the settings specified in the **Precursor ion generation** panel) or a spectral library. Prosit analysis should be exported in Spectronaut-compatible format (this produces a comma separated file but without the .csv extension, so just rename myPrositLib.spectronaut to e.g. myPrositLib.csv). In most cases, however, it is preferable to use the deep learning-based spectra/RTs prediction algorithm integrated in DIA-NN (see below).

Generating PDF report

This will instruct the GUI to launch a script (dia-nn-plotter.exe) to produce a PDF report (visualising various quality control metrics) from the DIA-NN output, once the analysis is finished. Condition and replicate IDs are inferred from the raw file names using the following procedure: after removing the common prefix and suffix, the last integer number is excised from the file names, with what remains serving as the condition identifier. For example, when analyzing C:/Raw/A1.wiff, C:/Raw/A2.wiff, C:/Raw/B1.wiff and C:/Raw/B2.wiff, DIA-NN will identify two conditions: A and B. Performance metrics, including CV values, are calculated and visualized for the conditions separately.

Precursor ion generation

FASTA digest for library-free search / library generation

This will instruct DIA-NN to *in silico* generate a spectral library from the FASTA file provided and use it to analyse the data files (if provided). In addition to the analysis report, a new spectral library can be generated (**Generate spectral library** option in the **Output** panel, see above). If a spectral library is provided in addition to the FASTA file, it will be used (experimental) to enhance and speed up the library-free search. In this mode, q-values for precursor ions that belong to the library will be calculated separately. Also see the “Generating a spectral library” section above for the description of two-step library-free methods.

Deep learning-based spectra and RTs prediction

The Windows version of DIA-NN features a deep neural network for prediction of spectra and retention times. The network has been trained on a tryptic digest (which did include C-terminal peptides ending with amino acids other than K or R) with lengths 7 - 30, up to 2 missed cleavages and up to 1 oxidised methionine. However, the network is also expected to demonstrate high prediction accuracy with longer peptides or non-tryptic peptides, e.g. it has been checked to perform well when predicting spectra of chymotryptic peptides. It is highly recommended that this network is used for library-free searches. If a spectral library is provided while FASTA digest is turned off, this feature allows to replace spectra and retention times in the library with predicted ones – this can sometimes improve performance, if the library in question is optimised for a very different LC-MS setup or just does not have enough fragments annotated.

If generation of spectral library is enabled, an *in silico* spectral library produced will be saved in the .speclib format. For example, if “lib.tsv” is specified as the **Output library** (**Output** panel), the library will be saved to “lib.predicted.speclib” (the file size is roughly proportional to the number of precursors: just over 200Mb per 1 million precursors). This can then be reused in regular library-based search without the need to carry out prediction again.

The speed of the neural network (implemented using PyTorch and running on CPU) strongly depends on the SIMD instructions supported by the CPU (DiaNN.exe prints this information when launched), thus benefiting considerably from running on fairly recent CPU models. On Intel i9-7940X it needs < 3 minutes per million precursors. For comparison, yeast sequence database yields just over 1 million and human – just under 5 million precursors, when processed with the default DIA-NN settings. Importantly, since floating point math operations produce very slightly different results when executed using different SIMD instructions, prediction results are slightly different on old and new CPU models. To fully describe the experiment when using this functionality, one thus needs to specify the CPU model or at least its instruction set.

Of note, DIA-NN still supports the approach for spectra and RTs prediction featured in its earlier versions, namely, the use of a “training” library (this approach is significantly inferior in comparison to the deep learning-based predictor). Such a library can now be specified using the `--learn-lib [library file]` command and used instead of the deep learning predictor (e.g. on Linux), however this is not recommended unless there is just absolutely no way to use the deep learning predictor.

Protease

Several built-in options are available for the *in silico* digest. Alternatively, one can specify custom cleavage specificity, e.g. “`--cut K*,R*,!*P`” corresponds to a canonical tryptic digest. Here, cleavage sites (pairs of amino acids) are listed separated by commas, ‘`*'` indicates any amino acid, and ‘!’ indicates that the respective site will not be cleaved. The default number of missed cleavages is set to 1 and it is not recommended to increase this value, unless there are very solid reasons for this based on the design of the experiment.

Modifications

In general, it is not recommended to enable variable modifications, as this slows down analysis considerably and expands the search space, potentially negatively affecting identification performance. In some cases, however, allowing for one or two oxidised methionines is OK. It is essential that the removal of likely interferences is enabled (see the Algorithm settings below), if e.g. deamidation is enabled and the runs have been acquired on an instrument that does not allow to effectively distinguish between deamidated ions and heavy isotopologues.

Arbitrary user-defined modifications can be added using the `--fixed-mod` and `--var-mod` commands, e.g. –`-var-mod UniMod:5,43.005814,KR` will enable lysine and arginine carbamylation as a variable modification. Similarly to the use of `--mod` (see above), adding “`,label`” would indicate it to DIA-NN that the modification is an isotopic label. Use lower-case letters for the amino acids to restrict the modification to the N-terminus of the peptide.

Precursor ion mass range

This allows to restrict the range of precursors being considered, e.g. to the range covered by the SWATH cycle of the instrument. Allows to slightly reduce memory consumption. Further, precursor charge considered can be controlled (experimental) using the `--min-pr-charge` and `--max-pr-charge` commands, e.g. `--min-pr-charge 2 --max-pr-charge 3`.

Fragment ion mass range

This allows to restrict the range of fragment ion masses being considered.

Algorithm

Mass accuracies can be determined automatically (when set to the default 0.0) or specified by the user. Initially, DIA-NN performs a preliminary search with mass accuracy set to 100ppm, followed by mass correction. If the masses reported by the instrument are guaranteed not to be that much off (e.g. the instrument is regularly calibrated), this setting can be reduced, e.g. to 30ppm with "`--mass-acc-cal 30`". This can sometimes produce a noticeable speed up for library-free searches.

The 'Scan window' setting allows to specify the radius of the retention time window as measured in MS2 cycles. The optimal setting of scan window is usually approximately equal to the average number of data points per peak (total, not FWHM). If set to 0, DIA-NN will determine it automatically: for the first run of the experiment only, unless the 'Unrelated runs' option is selected.

DIA-NN can use an ensemble of neural networks to calculate q-values. By default, these are trained for a single epoch, minimizing the chance of overfitting. It is not recommended to disable the removal of likely interferences when using neural networks. The number of epochs can be increased by setting `--nn-epochs`, and the number of neural networks in the ensemble – by setting `--nn-bagging` (more = better and slower; default = 12). To almost completely eliminate any potential overfitting and allow for an increased number of epochs, neural networks can be used in a "cross-validation mode", when each network is only used to score peptide-spectrum matches that have not been used for its training. However, in this case it is desirable to also increase the number of networks, slowing down the analysis. For example, "`--nn-cross-val --nn-bagging 96 --nn-epochs 5`".

The implicit protein grouping option allows to specify the proteotypicity definition that will be used by DIA-NN. For example, if set to the default "genes", DIA-NN will consider a peptide proteotypic, if all of its associated proteins correspond (according to the FASTA files provided) to a single gene. Proteotypicity definition affects protein q-value calculation, protein inference and grouping.

If it is known that no chromatographic peak broadening has occurred during the course of the experiment (this can be checked by looking at the peak width data in the .stats.tsv quality control report or the PDF report - these have been described above), it might be beneficial to switch the quantification strategy to "robust LC", to somewhat improve the quantification precision. The "high accuracy" mode enables interference removal from fragment elution curves, thus making quantities more reliable (i.e. giving more confidence that what is being quantified is indeed the peptide of interest and not the peptide of interest plus some unknown interfering peptide). However, this introduces some extra noise, so if the goal is to obtain low CV values, the alternative "high precision" mode would be more suitable.

DIA-NN always calculates "normalised" quantities in addition to the raw quantities. The "Global" cross-run normalisation algorithm acts under the assumption that most precursors are not differentially expressed. Its "RT-dependent" version allows to correct for SPE/ion spray efficiency issues that might affect different runs to a different extent. "Signal-dependent" normalisation (available in addition to the RT-dependent) might be highly useful for minimising batch effects in large-scale experiments. However, signal-dependent normalisation should be used with caution, as it acts under the assumption that most precursors are not differentially expressed, within any precursor intensity range containing > 500 precursors identified at 1% FDR. This is expected to be true for most experiments. Still, if one, for example, were to generate samples by mixing cell lysates of several species in different proportions, this assumption might not hold.

Pipelines

Pipeline support allows to set up automatic analysis of a series of experiments, using a separate set of settings for each. The **Add to pipeline** button adds the current set of settings (including all input and output file names) to the pipeline. Selecting one of the pipeline steps (with a mouse) loads the respective set of settings. The **Update step** button overwrites the selected pipeline step with the current set of settings. Buttons with pictograms to the right from the **Clear pipeline** button can be used to open and save .pipeline files. Note that the .pipeline format is not guaranteed to retain compatibility across different versions of DIA-NN GUI.

The pipeline functionality is also convenient for saving/loading "base settings" for particular types of experiments.