NP^3 MS Workflow

A Pipeline for LC-MS/MS Metabolomics Data Process and Analysis
Version - 1.2.1

Cristina F. Bazzano[*] Luiz F. G. Alves[†] Rafael de Felicio[‡] Daniela B. B. Trivella[§]
Guilherme P. Telles[¶]

July, 2025

1. Overview

The NP^3 MS workflow is a software system with a collection of scripts to enhance untargeted metabolomics research focused on drug discovery with optimizations towards natural products.

The workflow is an automatized procedure to cluster (join) and quantify the MS2 spectra (MS/MS) associated with the same ion, which eluted in concurrent chromatographic peaks (MS1), of a collection of samples from LC-MS/MS experiments. It generates a rank of candidate spectra responsible for the observed hits in bioactivity experiments, suggests the number of metabolites present in the samples ([M+H]+ ions) and constructs molecular networks to improve the analysis and visualization of the results.

The NP^3 MS workflow consists of ten major steps:

- Step 1: Metadata table construction describing the input LC-MS/MS samples, bioactivity scores and groups. This is the only step that requires user intervention.
- Step 2: Raw data pre-process, that enriches the MS2 spectra with MS1 chromatographic peak dimensions.
- Step 3: Clustering of pre-processed MS2 spectra into a collection of consensus spectra.
- Step 4: Quantification of consensus spectra per sample and computation of the sample type indicators.
- Step 5: Pairwise similarity comparisons of consensus spectra and cleaning of their quantifications based on the similarity values into a set of clean consensus spectra.
- Step 6: Library spectra identification of consensus spectra against In-Silico predicted MS/MS spectrum of Natural Products Database (ISDB) from the Universal Natural Products Database (UNPD) using the tremolo tool (for Unix OS and positive ion mode only).
- Step 7: Ionization variants annotation among concurrent clean consensus spectra (adducts, neutral losses, multiple charge, dimers/trimers, isotopes and in-source fragmentation are considered), creation of a ionization variant annotation molecular network (IVAMN) and assignment of the most likely [M+H]+ consensus spectra representatives (for positive ion mode only).
- Step 8: Merge of clean consensus spectra quantifications based on the annotated variants and the [M+H]+ representatives (for positive ion mode only).

[*]Brazilian Biosciences National Laboratory (LNBio), National Center for Research in Energy and Materials (CNPEM), 13083-970, Campinas, SP, Brazil and Institute of Computing (IC), University of Campinas (UNICAMP), 13083-852, Campinas, SP, Brazil - cristina.bazzano@lnbio.cnpem.br

[†]Brazilian Biosciences National Laboratory (LNBio), National Center for Research in Energy and Materials (CNPEM), 13083-970, Campinas, SP, Brazil - luiz.alves@lnbio.cnpem.br

[‡]Brazilian Biosciences National Laboratory (LNBio), National Center for Research in Energy and Materials (CNPEM), 13083-970, Campinas, SP, Brazil - daniela.trivella@lnbio.cnpem.br

[§]Brazilian Biosciences National Laboratory (LNBio), National Center for Research in Energy and Materials (CNPEM), 13083-970, Campinas, SP, Brazil - daniela.trivella@lnbio.cnpem.br

[¶]Institute of Computing (IC), University of Campinas (UNICAMP), 13083-852, Campinas, SP, Brazil - gpt@ic.unicamp.br

- Step 9: Correlation between the consensus spectra quantifications and the samples bioactivity scores to rank the candidates responsible for the observed hits in bioactivity experiments. It also computes the quantification grouping.
- Step 10: Creation of a spectra similarity molecular network (SSMN) based on the clean consensus spectra similarity and creation of the protonated networks IVAMN [M+H]+ and SSMN [M+H]+ filtered.

The Steps 2 to 10 can be automatically executed with the NP^3 command **run**. And different results from the NP^3 MS Workflow may be joined using the command **join_jobs**, which automatically execute Steps 3 to 10 with adaptations for joining previous results.

This workflow also contains two interactive commands for data visualization:

- The first extracts chromatogram from raw MS1 data files and save the images to PNG files. Depending on the chosen parameters this can be a total ion chromatogram (TIC), a base peak chromatogram (BPC) or an extracted ion chromatogram (XIC), extracted from each sample/file with customized groups, and m/z and retention time windows.
- The second visualizes and compares MS2 data from MGF files or peak lists. It can also save the images to PNG or SVG files. Currently, it is only supported for Unix OS.

Furthermore, the user can manually identify the resulting consensus spectra against the GNPS [8] spectral libraries and use another separated command of this workflow to join the GNPS identification results to the NP^3 MS workflow quantification tables.

1.1 Data

The main input for the NP^3 MS workflow are LC-MS/MS raw data files in mzXML or mzML formats (mzData format could also be used, but was not vastly tested). It supports only the positive ([M+H]+) ion mode. The negative ([M-H]-) ion mode could also be used, but the Steps 7 and 8 were not vastly tested for it and could lead to undesired results.

The workflow output is organized in folders and depending on the command can contain:

- The pre-processed spectra in MGF files, containing all detected and filtered MS2 enriched with MS1 information.
- The collection of consensus spectra from Step 3 and Step 5 in MGF files
- The consensus spectra quantification results in CSV files, referred to as count tables. The count tables contains a consensus spectra ID, named msclusterID (representing a m/z and a retention time interval), by row and its quantifications and indicators by column. The following count tables are created:
  - Clustering count tables (Step 4)
  - Clean count tables, with the clustering count table redundancies aggregated (Step 5)
  - Merged count tables (Step 8)
- The count tables can also contain the following columns:
  - The libraries identification results (Step 6)
  - The ionization variants annotations grouped by type (Step 7)
  - The list of [M+H]+ representatives, suggesting the number of metabolites in the samples (Step 7)
  - The bioactivity correlation scores, that can be used to rank the consensus spectra. And the quantification grouping (Step 9)
- The molecular networks of annotations and of similarity files from Steps 7 and 10:
  - IVAMN (Step 7)
  - SSMN (Step 10)
  - SSMN filtered (Step 10)
  - IVAMN [M+H]+ (Step 10)
  - SSMN [M+H]+ filtered (Step 10)

The pre-processed data and the clean data from the NP^3 results may be used as input for the joining jobs command.

2. Installation and Setup

Installation with Conda and Mamba

We recommend using Mamba because it's considerably faster, but if you want to use Conda, just replace commands starting with `mamba` by `conda`.

NP³ MS workflow includes a mamba and conda environment file for Unix and Windows, enabling users to install package dependencies.

First, download or clone the workflow repository.

Download the repository:

- Click on the green button named 'Code' and then click on the 'Download ZIP' option.
  - The repository contains the entire Universal Natural Products Database (UNPD), so this download can take a while to finish.
- Extract the zip file

Or **clone** the repository:

- Make sure you have Git installed (instruction in https://github.com/git-guides/install-git)
- Open the command line in the folder you want to place this repository and then type:

```
$ git clone https://github.com/danielatrivella/NP3_MS_Workflow.git
```

Then download and install **miniforge** to install the conda environment with the mamba package included from the following link:

- Miniforge for Linux amd64
- Miniforge for Windows
- More OS and architectures
  - During installation on Windows OS check the option: 'Add to my PATH environment variable'
    * If you do not check this option you have to manually add the conda anda mamba executables to your PATH environment variables
  - To install on Unix OS run the following terminal command in the folder where the Anaconda installer was downloaded

    ```
    $ sh Miniforge3-Linux-x86_64.sh
    ```

    * During installation when asked 'Do you wish the installer to initialize Anaconda3 by running conda init?' answer 'yes'
    * If you answer 'no', then you need to manually type the command `conda init` before using Mamba.
  - If you already have a conda environment installed on your device, you can install the mamba package exclusively in the base environment with the command below, but it is recommended that you do a fresh install removing the old conda environment.

    ```
    $ conda install conda-forge::mamba -y
    ```

For more detailed instructions, visit the conda-forge/miniforge's github

To verify if the Conda and Mamba installation was successful, open a **new** terminal window and type:

```
$ mamba --version
```

If it outputs the `mamba`'s and `conda`'s versions you are good to go.

Now, go the NP³ MS workflow repository folder that you have just downloaded and extracted, and open a terminal window there.

Then, create the NP^3 MS workflow conda environment to automatically install almost all the workflow programs and required packages using the following terminal command:

Unix OS:

```
$ mamba env create -f environment_np3_unix.yml
```

Windows OS:

```
$ mamba env create -f environment_np3_win.yml
```

The NP^3 MS workflow conda environment must be created and activated before running the NP³ MS workflow commands. If the NP^3 MS workflow environment was created successfully, activate it with the following command:

```
$ mamba activate np3
```

You should see the '(np3)' tag as the first thing in your terminal line. Every time you open a new terminal you must execute this command once again in order to activate the NP^3 MS workflow environment before entering the NP^3 MS workflow commands.

Now install the libraries required by the node.js program:

In the terminal execute the following command:

```
$ npm install shelljs@0.8.4 commander@5.1.0
```

Make sure all programs and OS libraries were properly installed and continue to the remaining packages installation. If you could not install the environment or failed in any of the above steps go to the **manual** installation below.

The workflow command **setup** (more instructions below) checks the programs and packages installation and tries to automatically install the missing ones (except for the *node.js* packages). If the workflow command **setup** fails the programs and packages must be manually installed, and then this command must be used to automatically compile the NP3_MSCluster algorithm or it must be compiled manually (more instructions below).

For **automatically** installing the remaining packages and compiling the NP3_MSCluster algorithm, in the NP^3 MS workflow repository folder open a terminal and run the following command:

```
$ node np3_workflow.js setup
```

- If it runs without any ERROR message you are good to go! Otherwise, look for dependencies problems and retry with superuser privileges. If it persists to fail follow the manual installation described below.

For **manually** installing the NP^3 MS workflow dependencies, use the following links to download the required programs and install them. Then, go to the repository folder and run the following commands in the terminal to install the programs' required packages:

- *node.js* (LTS version) - https://nodejs.org/en/download/
  - Also need to install the *npm* package manager (automatically installed with node.js - need to check the option in the installation setup for Windows OS)
  - Terminal command to install the *node.js* packages with *npm*:

  ```
  $ npm install shelljs@0.8.4 commander@5.1.0
  ```

- *make* - https://www.gnu.org/software/make/
- Compilers:
  - *gcc* - https://gcc.gnu.org/
  - *g++* - https://gcc.gnu.org/projects/cxx-status.html
    * These compilers are usually installed for most Unix distribution.
- *R >= 3.6.3* - https://www.r-project.org/
  - Terminal command to install the required *R* packages:

```
$ sudo Rscript src/R_requiriments.R
```

- *Python 3.7* - https://www.python.org/download/releases/3.7/
  - Also install *pip* - https://pypi.org/project/pip/
  - Terminal command to install the required *python 3* packages with *pip*:

```
$ pip install -r src/python_requirements
```

If any installation fails, look for dependencies problems and retry.

To manually **compile** the NP3_MSCluster algorithm, inside the NP^3 MS workflow repository go to the *NP3_MSCluster* folder and in the terminal run:

```
$ make clean
$ make
```

If it runs without any ERROR message you are good to go! Otherwise, look for dependencies problems and retry with superuser privileges.

After any update in the NP^3 MS workflow, it will be signalized in the repository if the NP3_MSCluster algorithm needs to be **recompiled**, using the **setup** command or manually, to make the updates work.

3. How to use

The NP^3 MS workflow is implemented in a command line interface (CLI) capable of automatically running all the workflow steps, except the first. The CLI have a help usage information describing the list of available NP^3 MS workflow commands and their options (examples and more details in the next sections).

Before running any command of the workflow, the metadata table must be manually constructed, defining how the workflow will behave (Step 1). Then, the entire workflow can be executed with the command *run*, which runs Steps 2 to 10 or just Steps 3 to 10, using results obtained separately for Step 2.

We strongly recommend to run the pre-process command (Step 2) separated when running the workflow for the first time in a new dataset. This way the most critical parameters (peak_width and rt_tolerance) can be manually tuned to achieve more accurate results in the LC-MS peak detection algorithm and processing, which also depends on the mass spectrometrer equipment and the data quality. The user should **follow the pre-processing optimization guide** present in section 4.3.4. of this manual to improve the pre-process result, and also inpect the data. The workflow default values have been tested with an UHPLC-MS/MS-qTOF equipment (Acquity HClass Waters UHPLC and ESI-QqTOF Impact II Bruker spectrometer).

For further analyses, after running the entire NP^3 MS workflow, each step can be executed separately on the results obtained previously and with different tolerance parameter values and/or correlation groups, as needed. When running a specific command separately it is important not to modify the original output files, instead create a copy of those files. In this way, you may reproduce the workflow steps as needed when repeating any step.

If you did install the workflow dependencies using the conda environments, always remember to activate the NP^3 MS workflow environment before executing any of the workflow commands. To activate the NP^3 MS workflow environment use the following command:

```
$ mamba activate np3
```

4. NP^3 MS Workflow Commands

The NP^3 MS workflow was developed in a Node.js CLI capable of automatically running the entire workflow, except for the first step. It can also execute each step of the workflow separately, plus the visualization commands and the command to include the GNPS library identification results in the count tables.

The executable is found in the root folder of the NP^3 MS workflow repository (named np3_workflow.js) and can be run with the command:

```
$ node np3_workflow.js [cmd] [options]
```

Where **cmd** are the available NP^3 MS workflow commands that the script can handle (the workflow steps) and **options** are the list of available options for each command.

The list of available commands is described below together with their respectively mandatory options (the full list of options is described in each command section). These lists can also be found by running the following command in the terminal:

```
$ node np3_workflow.js --help
```

Or simple:

```
$ node np3_workflow -h
```

From this point on, when NP^3 MS workflow commands are described, the following conventions will be used: - Angled brackets (e.g., <x>) indicate a required input. - Square brackets (e.g. [y]) indicate an optional input. - The brackets should not be typed while running the command, they are only used to indicate the type of the option (see the examples of the commands).

Commands:

- **setup** : Check if the NP^3 MS workflow dependencies are installed, try to install missing R and python packages and compile the NP3_MSCluster algorithm
- **run** [options] : Steps 2 to 10: Runs the entire NP^3 MS workflow.
  - List of mandatory options:
  - *-n, --output_name* <name> : the job name. It will be used to name the output directory and the results of the final clustering integration step
  - *-m, --metadata* <file> : path to the metadata table CSV file
  - *-d, --raw_data_path* <path> : path to the folder containing the input LC-MS/MS raw spectra data files (mzXML format is recommended)
  - *-o, --output_path* <path> : path to where the output directory will be created
- **pre_process** [options] : Step 2: This command runs the pre-process of the LC-MS/MS raw data. It extracts the list of MS1 peaks with their dimension information (minimum and maximum retention times, the peak area and ID) in each sample, matches the MS2 spectra retention time and precursor m/z against this list and assign to each MS2 spectra a MS1 peak that encompasses it. Additionally, a table with the MS1 peaks without any MS2 spectrum m/z and retention time match are stored in a count table of non-fragmented MS1 peaks.
  - List of mandatory options:
  - *-n, --data_name* <name> : the data collection name for printting in the verbose messages
  - *-m, --metadata* <file> : path to the metadata table CSV file
  - *-d, --raw_data_path* <path> : path to the folder containing the input LC-MS/MS raw spectra data files
- **clustering** [options] : Steps 3 and 4: This command runs the NP3_MSCluster algorithm to perform the clustering of pre-processed MS/MS data into a collection of consensus spectra. Then, it runs the consensus spectra quantification to count the number of spectra and peak area by sample (clustering counts). If necessary, it runs Step 2. And it can also run the library spectra identifications (Step 6) for the collection of consensus spectra.
  - List of mandatory options:

  - *-n, --output_name* : the job name. It will be used to name the output directory and the results of the final clustering integration step
  - *-m, --metadata* : path to the metadata table CSV file
  - *-d, --raw_data_path* : path to the folder containing the input LC-MS/MS raw spectra data files
  - *-o, --output_path* : path to where the output directory will be created

- **clean** [options] : Step 5: This command runs the pairwise comparisons of the collection of consensus spectra (if not done yet) and then runs the cleaning of the clustering counts. It also runs Step 7 to annotate possible ion variants using the new clean count tables and to create the molecular network of annotations, and runs Step 10 to overwrite any old computation of the molecular network of similarities. It can also run the library spectra identifications (Step 6) for the collection of clean consensus spectra.

  - List of mandatory options:
  - *-m, --metadata* <file> : path to the metadata table CSV file
  - *-o, --output_path* <path> : path to the final output data folder, inside the 'outs' directory of the clustering result folder. It should contain the 'mgf' folder and the 'count_tables' folder with the peak area and spectra count tables in CSV files. The job name will be extracted from here
  - *-y, --processed_data_dir* <path> : the path to the folder inside the raw data folder where the pre-processed data (MGFs) were stored.

- **tremolo** [options] : Step 6: (for Unix OS and positive ion mode only) This command runs the tremolo tool, used for spectra matching against In-Silico predicted MS/MS spectrum of Natural Products Database (ISDB) from the UNPD (Universal Natural Products Database). It also includes origin and class information of the compounds from NPClassifier, NPAtlas and ClassyFire.

  - List of mandatory options:
  - *-o, --output_path* : path to where the spectral library search results will be stored
  - *-g, --mgf* : path to the input MGF file with the MS/MS spectra data to be searched and identified

- **annotate_protonated** [options] : Step 7: (for positive ion mode only) This command runs the annotation of possible ionization variants in the clean count tables and creates the molecular network of annotations. It searches for adducts, neutral losses, multiple charges, dimers/trimers, isotopes and in-source fragmentation based on numerical equivalences and chemical rules. Finally, it runs a link analysis in the molecular network of annotations to assign some of the consensus spectra as putative [M+H]+ representatives.
  - List of mandatory options:
  - *-m, --metadata* <file> : path to the metadata table CSV file
  - *-o, --output_path* <path> : path to the output data folder, inside the 'outs' directory of the clustering result folder. It should contain the 'counts_table' folder and inside it the 'clean' subfolder with the clean count tables in CSV files. The name of the output folder will be used as the job name.
- **merge** [options] : Step 8: (for positive ion mode only) This command runs the merge of the clean count tables based on the annotated variants. It creates new symbolic spectra candidates representing the union of each spectra with its annotated variants. By default the merge is only performed for the consensus spectra assigned as a [M+H]+ representative ion, to better account for the quantifications of the true metabolites.
  - List of mandatory options:
  - *-o, --output_path* <path> : path to the output data folder, inside the 'outs' directory of the clustering result folder. It should contain the 'counts_table' folder and inside it the 'clean' subfolder with the clean count tables in CSV files. The job name will be extracted from here
  - *-y, --processed_data_dir* <path> : the path to the folder inside the raw data folder where the pre-processed data (MGFs) were stored.
  - *-m, --metadata* <file> : path to the metadata table CSV file
- **corr** [options] : Step 9: This command runs the bioactivity correlation to rank the consensus spectra based on the scores computed for the selection of samples and bioactivity values present in the metadata table.

  - List of mandatory options:
  - *-b, --metadata* <file> : path to the metadata table CSV file. Used to retrieve the biocorrelation groups
  - *-c, --count_file_path* <file> : path to the count table CSV file
- **mn** [options] : Step 10: This command runs the creation of a molecular network of similarity based on the pairwise spectra similarity value above the given similarity cut-off. Then, a filter is applied on this network to limit the number of neighbors of each node (number of links) to the top K most similar ones and to limit the size of the components to a maximum number of nodes. The final filtered network contains components that represent the most analogous spectra, possible connecting spectra from similar chemical classes.
  - List of mandatory options:
  - *-o, --output_path* <path> : path to the output data folder, inside the outs directory of the clustering result folder. It should contain the 'molecular_networking' folder and inside it the 'similarity_tables' folder. The job name will be extracted from here
- **gnps_result** [options] : This command join the GNPS library identification result from the Molecular Networking (download 'clustered spectra') or the Library Search (download 'All identifications') workflows to the count tables of the NP^3 clustering or clean steps.
  - List of mandatory options:
  - *-i, --cluster_info_path* <path> : If joining the result of a Molecular Networking job, this should be the path to the file inside the folder named 'clusterinfo' of the downloaded output from GNPS. Not used for results coming from the Library Search workflow. (default: " ")
  - *-s, --result_specnets_DB_path* <path> : If joining the result of a Molecular Networking job, this should be the path to the file inside the folder named 'result_specnets_DB'; and if this is the result of a Library Search workflow, this should be the path to the file inside the downloaded folder
  - *-c, --count_file_path* <path> : Path to any of the count tables (peak_area or spectra) resulting from the NP^3 MS workflow clustering or clean steps. If the peak_area is informed and the spectra table file exists in the same path (or the opposite), it will merge the GNPS results to both files
- **chr** [options] : This command runs an interactive prompt to extract chromatogram(s) from raw MS1 data files (mzXML, mzData and mzML) and to save to PNG image files. Depending on the provided parameters this can be a total ion chromatogram (TIC - default), a base peak chromatogram (BPC) or an extracted ion chromatogram (XIC) extracted from each sample/file.
- **spectra_viewer** [options] : (for Unix OS only) This command runs an interactive Web App to visualize and compare MS2 spectra. It receives as input a MGF file or a peak list. It is also possible to manipulate, filter, calculate similarity of the spectra and save PNG or SVG plots.
- **test** [options] : This command runs some use cases to test the NP^3 MS workflow consistency in all steps. This option is intended for debugging purposes, and is not a part of the analysis workflow.

4.1. Step 1: Construction of a Metadata Table

The metadata table must be a CSV file with columns separated by comma ',' and with the dot '.' as decimal point character. It can be created and edited in EXCEL, LibreOffice CALC or any other spreadsheet program, but must be saved in CSV (UTF-8) format. If any text name in the metadata file contains special characters, the file must be saved with the parameter to quote text cells set to TRUE. After the metadata file was finished it is recommended to open it with a simple text viewer, e.g., notepad, to make sure that the column separator character and the decimal point character were properly set.

A metadata template file is available in the NP^3 MS workflow repository, named 'METADATA_TEMPLATE.csv'.

The metadata table must contain the following mandatory columns (in uppercase):

- FILENAME - the raw data file names (including extensions but not including complete paths). The accepted file format is mzXML or mzML (mzData format could also be used, but was not vastly tested).
- SAMPLE_CODE - unique and syntactically valid (see below) names for each file. The use of the same SAMPLE_CODE in more than one entry will generate an error. A syntactically valid name must start with a letter and consist of letters, numbers, and underscore characters. R reserved words

(https://cran.r-project.org/doc/manuals/r-release/R-lang.html#Reserved-words) are not syntactically valid names. The SAMPLE_CODE names will be used to name and reference the pre-processed files in Step 2 and to name the output count table's columns of Steps 3 to 9.

- DATA_COLLECTION_BATCH - a number (starting with 1 and progressing in ascending order, e.g., 1,2,3,. . .) indicating to which data collection batch each sample file belongs. Files from the same data collection batch should have been collected in the same LC-MS/MS run (e.g., from the same screening or the same plate), ensuring closely related experimental conditions, and must have the same DATA_COLLECTION_BATCH number. See below more details about the order of samples in each batch.
- SAMPLE_TYPE - one among "sample", "blank", "bed", "control" or "hit". If the sample file is a chromatographic blank this column must be set to "blank" (e.g. DMSO, solvents, etc); if the sample is the culture media (containing molecules that should be identified for later exclusion or separation) set it to "bed"; if the sample is a control (e.g. the culture media + the non-elicited microorganism) set it to "control"; if the sample is bioactive and is desired to dereplicate its m/z's, set it to "hit"; and if the file is not a blank sample nor any of the above set it to "sample". Spectra from blank, control, bed and hit samples will be used to signal possible false positive spectra and to dereplicate m/z's that appear in hit samples (see Step 4 details).

The metadata table may contain the following optional columns:

- BIOACTIVITY_<NAME> - zero or more columns starting with the prefix "BIOACTIVITY_" followed by a unique name <NAME> defining different bioactivities values for each sample. Set it to a positive number (greater or equal than zero) indicating the sample bioactivity score, it could be an inhibition or an activation of a target. These values should be normalized, e.g., from 0 to 100, to be used in the correlation computation.
- COR_<NAME> - zero or more columns starting with the prefix "COR_" followed by a unique name <NAME> defining the correlation group. Each one of these columns must have values 0 or 1. If the sample spectra or peak area count is to be used in the COR_<NAME> correlation with each defined bioactivity scores (BIOACTIVITY_<NAME> columns), set it to 1. Otherwise, set it to 0. In Step 9 each one of these columns will result in a new column for each available bioactivity score in the output count tables, containing the bioactivity correlation scores computed with the respectively selected samples. For better correlation results the selected groups of samples should belong to the same bioactivity peak, e.g., a selection that contains very active, middle active and not active samples of a same bioactivity peak (a 'hit'). We strongly recommend to use the blank, bed and control samples in this selection to eliminate false positives. Bioactivity data must be collected from chemical samples at the same concentration. LC-MS/MS data must be collected from the chemical samples at the same concentration. This will guarantee real correlations, especially if the samples and bioactivity data come from a screening of natural product samples.
- GR_<NAME> - zero or more columns starting with the prefix "GR_" followed by a unique name <NAME> defining a set of grouping for the spectra quantification. Each of those columns may have one or more tags to label a group of samples. A column "GR_<NAME>" with groups "<g1>", "<g2>" and "<g3>" will bind to the final count tables the columns "<NAME_g1>", "<NAME_g2>" and "<NAME_g3>" containing the sum of peak areas or number of spectra of the samples that belongs to each group for each m/z row.
  For example, to create a set called **colors** with the groups **red**, **green** and **blue**, add to the metadata the column "GR_colors", tag the samples that belongs to the groups **red**, **green** and **blue** by filling the rows (you may have empty observations if it is necessary) and it will bind the columns **colors_red**, **colors_green** and **colors_blue** to the final count tables.

The user is free to add any additional column to the metadata file, for example to add relevant descriptions for each file. These additional columns will be ignored by the workflow commands, but can be useful to add information related to the job samples. It is important that additional column names are not equal or a prefix of any of the mandatory or optional columns of the metadata file.

4.1.1 Metadata Table Details: The first files of each data collection batch appearing in the given metadata table order, excluding blanks, are used in Step 2 to perform the suggestion of misalignment between the samples (this suggestion can be used in the parameter rt_tolerance of Steps 3 to 7). This information must be taken into account when defining the files order in the metadata table. It is expected that the first file of each data collection batch is the more related file between the different data collection batches, e.g., those that have the same polarity or other characteristic that provides them bigger chances of having more MS1 peaks in common. Files from extracts are a good choice to go first. This will only influence in the proposed retention time tolerance value used to deal with the misalignment between the samples; it will not modify the data (see 4.3.3. Step 2 Details).

The clustering will be performed using the DATA_COLLECTION_BATCH number and the SAMPLE_TYPE information. First, all the samples, excluding blanks, (column "SAMPLE_TYPE" equal to "sample", "hit", "bed" or "control") from the same batch (which have the same DATA_COLLECTION_BATCH number) are clustered in the *data clustering step*. Then all the blank samples (column "SAMPLE_TYPE" equal to "blank") from the same batch are clustered in the *blank clustering step*, in which the retention time is not used to better deal with baseline blanks. Subsequently, all sub-batches (data and blank steps results) from the same data collection batch are clustered in the *data collection batch integration step*. Finally, all batches are clustered in the *final integration step*. This way, spectra that were detected in the same conditions, and thus tend to be more similar, are enriched before being clustered with the spectra from a sample of a different batch, and finally all spectra are clustered.

4.2. Steps 2 to 10: Command *run*

This command runs the entire NP^3 MS workflow (Steps 2 to 10).

For the complete list of options run:

```
$ node np3_workflow.js run --help
```

Options for the command *run*:

- *-n, --output_name* <name> : the job name. It will be used to name the output directory and the results of the final clustering integration step. It must have less than 80 characters.
- *-m, --metadata* <file> : path to the metadata table CSV file
- *-d, --raw_data_path* <path> : path to the folder containing the input LC-MS/MS raw spectra data files
- *-o, --output_path* <path> : path of the output directory
- *-f, --fragment_tolerance* [x] : the tolerance in Daltons for fragment peaks. Peaks in the original MS/MS spectra that are closer than this tolerance are merged in the clustering steps (Step 3). It is also used in the pre-processing (Step 2), in the spectra similarity comparisons and in the cleanning Step 5 (default: 0.05)
- *-z, --mz_tolerance* [x] : this is the tolerance in Daltons for the m/z of the precursor that determines if two spectra will be compared and possibly joined. It is used in the clustering steps (Step 3), in the cleaning (Step 5), in the library identifications (Step 6) and in the annotation of ionization variants (Step 7) (default: 0.025)
- *-p, --ppm_tolerance* [x] : the maximum tolerated m/z deviation in parts per million (ppm) to be used in the pre-processing (Step 2). Typically set to a generous multiple of the mass accuracy of the mass spectrometer. (default: 15)
- *-a, --ion_mode* [x] : the precursor ion mode. One of the following numeric values corresponding to an ion adduct type: '1' = [M+H]+ or '2' = [M-H]- (default: 1)
- *-i, --similarity_function* [x] : the similarity function to be used in the spectra comparison to create the pairwise similarity tables after clustering and clean steps. One of "np3_shifted_cosine" or "spec2vec". If "spec2vec" is selected, the model trained on UniqueInchikey subset (12,797 spectra) is used by spec2vec in the spectra comparison and the matchms library is used to compute the number of matched peaks between the compared spectra; otherwise, the NP3 shifted cosine function is used. (default: "np3_shifted_cosine")

- *-s, --similarity* [x] : the minimum similarity to be considered in the hierarchical clustering of Step 3, starts in 0.70 and decrease to X in 15 rounds. Also used in the Step 5 quantification cleaning (default: 0.55)
- *-g, --similarity_blank* [x] : the minimum similarity to be consider in the hierarchical clustering of the blank clustering steps, starts in 0.70 and decrease to X in 15 rounds. Only used in the clustering of blank samples (column SAMPLE_TYPE equals 'blank' in the metadata table) (default: 0.3)
- *-w, --similarity_mn* [x] : the minimum similarity score that must occur between a pair of consensus spectra to connect them with a link in the molecular network of similarity. Lower values will increase the components sizes by inducing the connection of less related spectra; and higher values will limit the components sizes to the opposite (default: 0.6)
- *-t, --rt_tolerance* [x,y] : tolerances in seconds for the retention time width of the precursor that determines if two spectra will be compared and possibly joined. It is directly applied to the retention time minimum (subtracted) and maximum (added) of the spectra. It enlarges the peak boundaries to deal with misaligned samples or ionization variant spectra. The first tolerance [x] is used in the data, blank and batches integration steps from the clustering Step 3 and in Steps 2 (if no previous result is provided) and 7 (chemical annotations); and the tolerance [y] is used in the final integration step from the clustering Step 3 and in the clean Step 5 (default: 1,2)
- *--min_matched_peaks* [x] : The minimum number of common peaks that two spectra must share to be connected by an edge in the filtered SSMN. Connections between spectra with less common peaks than this cutoff will be removed when filtering the SSMN. Except for when one of the spectra have a number of fragment peaks smaller than the given min_matched_peaks value, in this case the spectra must share at least 2 peaks. The fragment peaks count is performed after the spectra are normalized and cleaned. (default: 6)
- *-k, --net_top_k* [x] : the maximum number of connections for one single node in the molecular network of similarity. A link between two nodes is kept only if both nodes are within each other's X most similar nodes. Keeping this value low makes very large networks (many nodes) much easier to visualize (default: 15)
- *-x, --max_component_size* [x] : the maximum number of nodes that each component of the molecular network of similarity must have. The links of this network will be removed using an increasing cosine threshold until each component has at most X nodes. Keeping this value low makes very large networks (many nodes and links) much easier to visualize. (default: 200)
- *--blank_expansion* [x] : the distance of neighborhood nodes from the blanks in IVAMN to be selected for removal in the final protonated networks. (0) to only remove blanks nodes, (1) to remove nodes directly connected to a blank node, (2 or greater) to remove nodes in a distance equal to 2 or greater from a blank node, or (-1) to remove all possible neighbours and ancestors of a blank node (remove blank clusters) from IVAMN (default: 0)
- *-c, --scale_factor* [x] : the scaling method to be used in the fragmented peak's intensities before any dot product comparison (Steps 3 and 6). Valid values are: 0 for the natural logarithm (ln) of the intensities; 1 for no scaling; and other values greater than zero for raising the fragment peaks intensities to the power of x (e.g. x = 0.5 is the square root scaling). [x] >= 0 (default: 0.5)
- *-l, --parallel_cores* [x] : the number of cores to be used for parallel processing in the spectra pairwise comparison (Step 5). x = 1 for disabling parallelization and x > 2 for enabling it. [x] >= 1 (default: 2)
- *-y, --processed_data_name* [x] : the name of the folder inside the *raw_data_path* where the pre-processed data will be stored. If the given folder does not exist it will be created and the pre-process will be run using the default values of the missing options. Otherwise, it will depend on the *processed_data_overwrite* parameter value (default: "processed_data")
- *-q, --processed_data_overwrite* [x] : A logical "TRUE" or "FALSE" indicating if the pre-processed data present in the *processed_data_name* folder (if it already exists) should be overwritten and pre-processed again (default: "FALSE")
- *--bflag_cutoff* [x] : A positive numeric value to scale the interquartile range (IQR) of the blank spectra basePeakInt distribution and allow spectra with a basePeakInt value below this distribution median plus IQR*bflag_cutoff to be joined with a blank spectrum without relying on the similarity value. Or FALSE to disable it. The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile). The spectra with a basePeakInt value <= median + IQR*bflag_cutoff of the

blank spectra basePeakInt distribution and BFLAG TRUE will be joined to a blank spectrum in the clean step. This cutoff will affect the spectra with BFLAG TRUE that would not get joined to a blank spectra when relying only on the similarity cutoff. This is a turn around to the fact that blank spectra have low quality spectra and thus can not fully rely on the similarity values. (default: 1.5)

- *--noise_cutoff* [x] : A positive numeric value to scale the interquartile range (IQR) of the blank spectra basePeakInt distribution from the clustering Step 3 result and to remove the spectra with a basePeakInt value below this distribution median plus IQR*noise_cutoff after the clean Step 5. Or FALSE to disable it. The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile) of the distribution. When no blank sample is present in the metadata, the full distribution is used. This cutoff will affect the spectra with with a low basePeakInt value that probably are noise features. If the clustering Step 3 results in more than 25000 spectra, the noise cutoff will be applied before the clean Step 5 to prevent a long processing time (default: “FALSE”)
- *-u, --rules* [x] : path to the CSV file following the NP3 rules table format with the accepted ionization modification rules for detecting adducts, multiple charge and dimers/trimers variants, and their combination with neutral losses. To be used by the annotation algorithm (Step 7) (default: “rules/np3_modifications.csv”)
- *-r, --trim_mz* [x] : A logical “TRUE” or “FALSE” indicating if the spectra fragmented peaks around the precursor m/z +-20 Da should be deleted before the pairwise comparisons. If “TRUE” this removes the residual precursor ion, which is frequently observed in MS/MS spectra acquired on qTOFs. (default: “TRUE”)
- *--max_shift* [x] : Maximum difference between precursor m/zs that will be used in the search of shifted m/z fragment ions in the NP3 shifted cosine function. Shifts greater than this value will be ignored and not used in the cosine computation. It can be useful to deal with local modifications of the same compound. (default: 200)
- *-b, --max_chunk_spectra* [x] : Maximum number of spectra (rows) to be loaded and processed in a chunk at the same time. In case of memory issues this value should be decreased. To be used in Steps 6, 7 and 10 (default: 3000)
- *-e, --method* [name] : a character string indicating which correlation coefficient is to be computed. One of “pearson”, “kendall”, or “spearman” (Step 9) (default: spearman)
- *-j, --tremolo_identification* [x] : (not Windows OS’s) A logical “TRUE” or “FALSE” indicating if the Tremolo tool should be used for the spectral matching against the ISDB from the UNPD (Step 5) (default: “TRUE”)
- *-v, --verbose* [x] : for values X>0 show the scripts output information. For values greater or equal to 10 a consistency test of the results is also performed. (default: 0)
- *-h, --help* : output usage information

4.2.1. *run* Results A directory inside the *output_path* named with the *output_name* containing:

- A copy of the *metadata* and the *rules* files and the command line parameters values used in a file named ‘logRunParms’, for reproducibility
- A folder named ‘outs’ with the clustering steps results in separate folders containing:
  - A subfolder named ‘count_tables’ with the Step 4 quantifications in CSV tables named as ’<step_name>_(spectra|peak_area).csv’.
  - Another subfolder named ‘clust’ with the clusters membership files (which SCANS or msclusterID were joined)
  - A third subfolder named ‘mgf’ with the resulting clusters consensus spectra in MGF files
  - A text file named ‘logNP3MSClusterOutput’ with the NP3_MSCluster log output.

The clustering steps results folders are named as ‘B_<DATA_COLLECTION_BATCH>_<X>’ where *DATA_COLLECTION_BATCH* is the data collection batch number in the metadata file of each group of samples and *X* is 0 if it is the result of a *data clustering step* or 1 if it is the result of a *blank clustering step*. The *data collection batch integration step* results are stored in folders named as ‘B_<DATA_COLLECTION_BATCH>’.

The final integration step result is located inside the ‘outs’ directory in a folder named with the *output_name*. This folder contains the final counts and is where the user will find the final results. It also contains the following data:

- Inside the 'count_tables' folder two subfolders named "clean" and "merge" containing CSV tables with the counts from Steps 6, 7, 8 and 9, and the base peak intensity distribution plot of the clustering counts in a PNG image file;
- The 'identifications' folder with the tremolo identification results;
- The 'molecular_networking' folder containing:
  - One subfolder named "similarity_tables" with the pairwise similarity tables (Step 5);
  - Five molecular networks edge files (Steps 7 and 10):
    - The ionization variant annoation molecular network named as: '<*output_name*>_ivamn.selfloops'
    - The complete spectra similarity molecular network named as : '<*output_name*>_ssmn_w_<*similarity_mn*>.selfloops', which contains all links with a similarity value above the cut-off
    - The filtered spectra similarity molecular network named as '<*output_name*>_ssmn_w_<*similarity_mn*>_k_<*net_top_k*>_x_ <*max_component_size*>.selfloops'
    - The IVAMN [M+H]+ named as: '<*output_name*>_ivamn_protonated.selfloops'
    - The SSMN [M+H]+ filtered named as: '<*output_name*>_ssmn_protonated_w_<*similarity_mn*>_k_<*net_top_k*>_x_ <*max_component_size*>.selfloops'
  - One CSV table containing the molecular network of annotations attributes and the assigned [M+H]+ representatives, named as: '<*output_name*>_ivamn_attributes.csv' (Step 7)

4.2.2. *run* Examples Fake examples using the *run* command:

```
$ node np3_workflow.js run --output_name "test_np3" --output_path
  "/path/where/the/output/will/be/stored/" --metadata
  "/path/to/the/metadata/file/test_np3_metadata.csv" --raw_data_path
  "/path/to/the/raw/data/dir/" --fragment_tolerance 0.01
```

```
$ node np3_workflow run -n "test_np3_rt_tol" -o "/path/where/the/output/will/be/stored"
  -m "/path/to/the/metadata/file/test_np3_metadata.csv" -d "/path/to/the/raw/data/dir"
  -t 3.5,5
```

4.3. Step 2: Command *pre_process*

Runs the pre-process of the LC-MS/MS raw data. It extracts the list of MS1 peaks with their dimension information (retention time minimum and maximum, the peak area and ID) in each sample, matches the MS2 spectra retention time and precursor m/z against this list and assign to each spectra a MS1 peak that encompasses it. Additionally, a table with the MS1 peaks without a MS2 spectrum m/z and retention time match are stored in a count table to account for the not fragmented MS1 peaks. Finally, a diagnostic is performed to evaluate the pre-process result followed by a suggestion for better *rt_tolerance* and *peak_width* parameters values, based on the list of fragmented MS1 peaks.

If *max_samples_batch_align* > 0, align the samples to obtain a good guess for the retention time disaliagment between samples of the same data collection batch and between all samples. The alignment does not modify the samples retention time, it's only used to suggest a retention time tolerance value for the following commands (clustering, clean, annotate_protonated and merge).

It generates one MGF by sample containing all the detected MS2 spectra enriched with their respectively matched MS1 peak dimensions, e.g., retention time minimum, maximum, peak area and peak ID (given by the peak detection algorithm). When using the same *raw_data_path* to pre-process the LC-MS/MS raw data files using different metadata tables, a different *processed_data_name* must be used to avoid overwriting useful files and to store the created MGFs in different folders.

Running the pre-process step with a big data collection (number of samples above ~100) can be time consuming and impracticable to repeat this step more than one time to try to improve the parameters values based on the final diagnostic and suggestions. In this case, we recommend the user to build a metadata table with only a small selection of the samples, choosing the ones that could be more representative of the data collection and all blank samples (to correctly remove the blank m/zs). Then, iteratively pre-process these small selection of samples until better results are obtained based on the final diagnostic and suggestions for the parameters values. Finally, the user can run the pre-process for the complete data collection using the parameters that yielded the best result.

Options for the command *pre_process*:

- *-n, --data_name* <name> : the data collection name for verbosity
- *-m, --metadata* <file> : path to the metadata table CSV file
- *-d, --raw_data_path* <path> : path to the folder containing the input LC-MS/MS raw spectra data files (mzXML format is recommended)
- *-y, --processed_data_name* [x] : The name of the output directory that should be created inside the *raw_data_path* to store the processed data. When not using the default directory, this value should be informed in the following *run* or *clustering* commands. (default: processed_data)
- *-t, --rt_tolerance* [x] : the tolerance in seconds used to enlarge the MS1 peaks boundaries and accept as a match all MS2 ions that have a retention time value that is within a MS1 peak range. This value is applied to both sides of the MS1 peaks (RTmin - rt_tolerance and RTmax + rt_tolerance). Tries to overcome bad MS1 peak integrations. (default: 3)
- *-z, --mz_tolerance* [x] : the tolerance in Daltons for matching a MS1 peak m/z with a MS2 spectrum precursor m/z. (default: 0.05)
- *-p, --ppm_tolerance* [x] : the maximal tolerated m/z deviation in consecutive MS1 scans in parts per million (ppm) for the initial ROI definition of the R::xcms::centWave algorithm. Typically set to a generous multiple of the mass accuracy of the mass spectrometer. (default: 15)
- *-a, --ion_mode* [x] : the precursor ion mode. One of the following numeric values corresponding to a ion adduct type: '1' = [M+H]+ or 2' = [M-H]- (default: 1)
- *-e, --peak_width* [X,Y] : two numeric separated by comma ',' without spaces and using decimal point equals dot '.', containing the expected approximate peak width in chromatographic space. Given as a range (min, max) in seconds. The mean value will be used to simulate the width of the fake peaks (see Details section bellow) (default: 2,10)
- *-s, --snthresh* [x] a numeric defining the signal to noise ratio cutoff. (default: 0)
- *-f, --pre_filter* [k,I] two numeric separated by comma ',' without spaces and with decimal point equals dot '.' specifying the prefilter step for the first analysis step (ROI detection) of the R::xcms::centWave algorithm. Mass traces are only retained if they contain at least k peaks with intensity >= I. (default: 1,750)
- *-r, --noise* [x] a numeric allowing to set a minimum intensity required for centroids to be considered in the first analysis step (centroids with intensity < noise are omitted from ROI detection). (default: 500)
- *-c, --mz_center_fun* [name] Name of the function to calculate the m/z center of the chromatographic peak. Allowed are: "wMean": intensity weighted mean of the peak's m/z values, "mean": mean of the peak's m/z values, "apex": use the m/z value at the peak apex, "wMeanApex3": intensity weighted mean of the m/z value at the peak apex and the m/z values left and right of it and "meanApex3": mean of the m/z value of the peak apex and the m/z values left and right of it. (default: wMeanApex3)

- *-u, --integrate_method* [x] Integration method for the XCMS CentWave algorithm. For integrate = 1 peak limits are found through descent on the mexican hat filtered data, for integrate = 2 the descent is done on the real data. The latter method is more accurate but prone to noise, while the former is more robust, but less exact. (default: 2)
- *-g, --fit_gauss* [x] a logical "TRUE" or "FALSE" indicating whether or not a Gaussian should be fitted to each peak. Decreases performance. (default: FALSE)
- *-j, --max_samples_batch_align* [x] The maximum number of not blank samples to be selected in the metadata sequence for the batch alignment. Due to memory issues this value should not exceed 15 samples to avoid crashing the script. If x == 0 disable the alignment. (default: 5)
- *-i, --min_fraction* [x] a numeric defining the minimum fraction of samples in at least one sample group in which the peaks have to be present to be considered as a peak group (feature). Used in the alignment process to define hook peaks. (default: 0.3)
- *-w, --bw* [x] a numeric defining the bandwidth (standard deviation of the smoothing kernel) to be used. This option is passed to the density method used in the alignment process. (default: 2)
- *-b, --bin_size* [x] a numeric defining the size of the overlapping slices in mz dimension. This option is passed to the density method used in the alignment process. (default: 0.05)
- *-x, --max_features* [x] a numeric with the maximum number of peak groups to be identified in a single mz slice. This option is passed to the density method used in the alignment process. (default: 100)
- *-q, --processed_data_overwrite* [x] A logical "TRUE" or "FALSE" indicating if the pre-processed data present in the processed_data_name folder should be overwritten and pre-processed again if it already exists (default: "TRUE")
- *-v, --verbose* [x] for values x>0 show the script output information (default: 0)
- *-h, --help* output usage information

4.3.1. ***pre_process*** Results

A folder named *processed_data_name* inside the *raw_data_path* containing:

- One MGF per sample of the metadata table with the MS2 spectra enriched with their matched MS1 peak chromatographic dimensions. Each pre-processed file is named as "<SAMPLE_CODE>_peak_info.mgf", where <SAMPLE_CODE> is as defined in the metadata file located in the *metadata_path*.
- A count file named "MS1_list_with_MS2.csv" with the quantification by sample of all the MS1 peaks that were assigned to a MS2 ion.
- A count file named "MS1_list_no_MS2.csv" with the quantification by sample of all the MS1 peaks that were not assigned to a MS2 ion, e.g. probably not fragmented MS1 peaks, and thus these peaks will be missing in the final clustering counts. It will help to search for the ion's isotopic distributions.
- A log file named "logPreProcessStatisticsWarning" or "logPreProcessStatistics" with the statistics of the rate of MS2 spectra without a MS1 peak correspondence and guidelines to improve this step result.
- A table file named "log_MS2_no_MS1peak_match.csv" with the informations of the MS2 spectra that did not have a MS1 peak correspondence.
- If *max_samples_batch_align* > 0, a CSV file named "samples_alignment.csv" is created with the maximum misalignment value in seconds for each data collection batch, as defined in the metadata table, and for all samples together. These values serve as a suggestion for the retention time tolerance to be used in the following steps of the workflow. This alignment process does not change the samples retention time.
- A file named "parameters.csv" with the parameter's values used, for reproducibility.

4.3.2. ***pre_process*** Examples:

Pre-processing with the default parameters values and compute the misalignment using at most 6 samples per data collection batch:

```
$ node np3_workflow pre_process --data_name "data_UHPLC_qTOF" --metadata
  "/path/to/the/metadata/file/test_np3_metadata.csv" --raw_data_path
  "/path/to/the/raw/data/dir" --max_samples_batch_align 6 --verbose 1
```

Pre-processing without computing the misalignment and with a different m/z tolerance value:

```
$ node np3_workflow pre_process --data_name "data_UHPLC_qTOF" -m
  "/path/to/the/metadata/file/test_np3_metadata.csv" -d "/path/to/the/raw/data/dir"
  -j 0 -z 0.01
```

4.3.3. *pre_process* Details The pre-process step use the R package XCMS CentWave algorithm [1] to process the LC-MS data and extract the samples' list of MS1 peaks, the XCMS::PeakGroup algorithm [2] to measure the disalignment between the samples, the MSnbase R package [3] to read the raw MS/MS data and a modified version of this package function to write to MGF files the pre-processed MS/MS data enriched with the MS1 peak dimensions information. Four fields are added to the MGF files header to store the MS1 information: RTMIN, RTMAX, PEAK_ID and PEAK_AREA.

The pre-process will enrich the MS2 spectra with a MS1 peak dimension. To do so, each detected MS2 ion's retention time and precursor m/z are matched against its sample list of MS1 peaks. The closest peak that encompasses the spectrum is assigned to it. This match reference is used by the write MGF function to write the peak dimensions information (retention time minimum and maximum, the peak area and the peak ID) in the header of each MS2 ion of the resulting pre-processed MGFs. The peak ID is unique for each MS1 peak, it is given by the CentWave algorithm and has the spectra's sample code of origin added as a suffix. The MS2 retention time is kept as the center of the peak.

Blank samples (in the metadata table SAMPLE_TYPE equals 'blank') receive a different treatment when enriching the spectra. To deal with the baseline problem, e.g., no defined MS1 peak for a given m/z, all MS2 ions of the blank samples are assigned with a baseline peak dimension which encompasses the entire chromatogram from retention time 0 to 1000000 seconds. These allows for more false positive spectra to be removed from the results.

In the matching peak step, depending on the parameter's values used, mostly the *peak_width*, some peaks can be incorrectly integrated by the CentWave algorithm generating false positive and false negative peaks. This can lead to no match between a MS2 ion and a MS1 peak. The reasons for not finding a peak match can be mostly because of three facts: (1) bad defined values for the *peak_width* parameter; (2) "bad" peaks that do not have a good gaussian shape (e.g., jagged peaks, very close intersecting isomers peaks) can make the CentWave algorithm not work properly and wrongly integrate the peaks (since it is an approximated solution) [4,5], or (3) a bad choice of the *mz_tolerance* or the *rt_tolerance* parameters values that could be too small to allow a match. The first and third cases can be modified as needed by the user for each job, and the second case depends on the data quality.

When no MS1 peak match is found for an MS2 spectra, a "fake" peak is created. The mean peak width, provided in the *peak_width* parameter, is used to simulate the fake peak width centered in the MS2 spectra retention time (RTmean +- mean(peak_width)/2) and the spectra intensity is used as the fake peak area (the clustering will simulate the peak area integration by summing the MS2 spectra intensities). When the peak match occurs because of the *rt_tolerance* value, e.g., the MS2 ion retention time is outside the integrated MS1 peak width but within a distance less than the *rt_tolerance* value, the peak width for that spectra is enlarged to encompass the spectra retention time correctly. When more than one match is found, the closest MS1 peak with respect to the peak retention time mean (center of the peak) to the spectra's retention time is chosen as the final match.

When no MS2 spectra receives an integrated MS1 peak, this peak dimensions and quantifications by sample are recorded in a file named "MS1_list_no_MS2.csv" inside the processed data folder. These records will generate a new count file with all MS1 peaks that were not fragmented and thus did not produce a MS2 spectra, and therefore, will be missing in the final clustering counts. This count is aggregated after a clustering job to produce the MS1 count file named with the suffix "_peak_area_MS1.csv", which will add up for the job completeness to try to guarantee that any ion that is detected in the data collection samples will be present in the final counts. These list of not fragmented MS1 peaks will also be used to search for isotopic patterns of multiple charged ions as part of the rules used in the annotation step (Step 7).

When *max_samples_batch_align* > 0, the XCMS::adjustRtime-peakGroups algorithm [2] is used to perform an alignment of the samples MS1 peaks and suggest a value to overcome the samples misalignment. The maximum deviation found between the sample's retention time is used to give a suggestion for the retention time tolerance value that should be used in the following workflow commands (in the clustering, clean and merge step, NOT in the pre-process step). It must be noted that the alignment does not change the samples retention time, it only gives a suggestion of the retention time misalignment between the samples.

The samples alignment is a very computational consuming step and can possibly not work or crash in some computers with limited memory and/or processing capacity. To try to overcome this difficulty three programming decisions were made. First, the blank samples (metadata column SAMPLE_TYPE equals 'blank') are not used in the alignment, mainly because these samples usually have a lot of noise that could lead to a bad behavior of the algorithm. Second, the samples alignment of a same data collection batch can be made using only a given number of samples, defined by the parameter *max_samples_batch_align*, chosen following the not blank samples order in the provided metadata table (the first not blank sample of each data collection batch appearing in the metadata file is then always picked). And third, the alignment between all samples is always made using only one representative sample of each data collection batch, which is also hard coded to be the first not blank sample of each data collection batch appearing in the metadata file. These behaviors must be taken into account when setting the *max_samples_batch_align* value and when defining the samples order in the metadata file. The first sample of each data collection batch in the metadata file should be the more related samples between the data collections, e.g. the ones that have the same polarity or bigger chances of having more peaks in common.

The parameters *min_fraction*, *bin_size*, *bw* and *max_features* should also be tuned to achieve a better alignment result. If the alignment persists to fail it is up to the user to guess a good value for the retention time tolerances of the following commands, and this value should represent the real misalignment between the samples. The alignment can fail if not enough common MS1 peaks are found between the samples, this is a limitation of the used algorithm.

If the alignment succeeds, then the user may evaluate to run the following NP^3 MS workflow commands with the suggested retention time tolerance values. The user can increase or decrease this tolerance, but we recommend to try to keep it in a range that will prevent joining close by peaks (isomers) or splitting a large peak depending on the common *peak_width* found in the raw data. It's up to the users to correctly define this tolerance values.

Finally, a diagnostic is performed to evaluate the pre-process result based on the rate of MS2 spectra without a MS1 peak correspondence (percentage of fake peaks). And then, a suggestion for better *peak_width* and *rt_tolerance* pre-process parameters values is performed to help the user improve this step result when needed. The diagnostic computes the percentage of MS2 spectra without a MS1 m/z range correspondence (named rate_MS2_no_MS1_mz) and the percentage of MS2 spectra without a MS1 retention time range correspondence (named rate_MS2_no_MS1_rt) for each sample and removing blank m/z. If any of these rates is above a cutoff of 5% for a not blank sample, a report is printted informing the most problematic m/zs by sample to facilitate a manual evaluation of these cases by the user. The user should visualize the chromatogram of these m/zs to identify the problem(s), and then, decide to repeat the pre-process step with other parameters values or to accept the result. A descriptive statistics of these rates is also printted to help the user in this decision, when the job have a big number of samples it can be more reasonable to pay more attention to the mean/median values of these rates. Following this diagnostic, a suggestion for better *peak_width* and *rt_tolerance* parameters values is performed based on the peak width distribution of the MS1 list with a MS2 correspondence (not fake peaks) and without blanks (table 'MS1_list_with_MS2.csv' in the pre-process result). The suggestion is computed as follow:

- Retention time tolerance = median / 2
- Peak width range (minimum and maximum) = Q25 - 1.5 * IQR, Q75 + 1.5 * IQR

- The IQR (interquartile range) factors start at 1.5 and are decreased by 0.25 until the peak width suggestions are between the minimum and maximum peak width limit values.

Where the median, Q25, Q75 and IQR values corresponds, respectively, to the median, first quartile (percentile 25th), third quartile (percentile 75th) and interquartile range values of the peak width distribution of the not fake peaks without blanks. Depending on the shape of this distribution, the suggestions can be more or less accurate.

The user should use these suggestions as a guide for choosing better values for the parameters, depending on the diagnostic result and also based on your own analysis/knowlegde of the data (e.g. chromatograms inspection) and the peak width distribution of the real MS1 peaks.

At the end, a diagnostic of possible splitted peaks is also performed. It works by counting the number of MS1 peaks by m/z present in the MS2 data, without blanks. The number of MS1 peaks indicates the number of possible isomers by m/z. The distribution of the number of m/zs that had a given number of MS1 peaks is stored in the file 'log_number_mzs_by_number_peaks.png' located in the pre-process result. This diagnostic will print a warning if there are more than 5% of the m/zs with more than 30 MS1 peaks (possible isomers), which can indicate that real peaks are being splitted in smaller peaks. The user should evaluate if this is really an issue for the dataset being used.

4.3.4. *pre_process* Optimization Guide The Pre-processing step performs the LC-MS data processing (feature finding) with XCMS. It extracts the list of MS^1 peaks for each sample, and then, directly impact in the isomers separation and resolution. The LC-MS processing is highly dependent on the parameters setup. The optimization of this parameters depends on the mass spectrometer equipment, the chromatographic method used and the data quality. With that pointed out, a diagnostic of Pre-process result and a suggestion of values for the **most critical parameters** were implemented to help the user in this optimization. The diagnostic and suggestion are still in a **Beta version**, and should be used as a guide and not as a rule.

In order to optimize the Pre-processing step, the user could follow two strategies: (1) run pre-processing with the default parameters and analyse the diagnostic result and suggestions; or (2) inspect the chromatograms of the dataset first (using a target m/z or randomly pick one) and run the pre-processing with a pre setup of parameters values that best represent what you observed in the real data. **The second option is more recommended**, because the suggestion of values for the most critical parameters will be biased by the parameters values used. The user can also choose to run the entire workflow with this first pre-processing result to examine the final clean quantification tables and decide if they are good enough for its purpose.

The critical parameters for the pre-processing step are:

1. **ppm_tolerance** - the maximal tolerated m/z deviation in consecutive MS1 scans in parts per million (ppm). Typically set to a generous multiple of the mass accuracy of the mass spectrometer;
2. **mz_tolerance** - the tolerance in Daltons for matching the m/z of a MS1 peak with the precursor m/z of a MS2 spectrum;
3. **rt_tolerance** - the tolerance in seconds used to enlarge the MS1 peaks boundaries and accept as a match all MS2 ions that have a retention time value within a listed MS1 peak range;
4. **peak_width** - the expected approximate peak width in chromatographic space (MS1 peaks). Given as a range (min, max) in seconds.

The first two critical parameters are more related to the mass spectrometrer equipment accuracy and the data resolution, while the third and fourth critical parameters are also related to the chromatographic method and the data quality (well defined MS1 peaks). Then, to better define the last two critical parameters, the user must know your data and/or inspect its chromatograms to visualize the MS1 peaks' shape and definition. The MS1 peaks' shape and definition can vary between different datasets coming from the same equipment and can also vary between different m/zs from the same dataset. Because of that, the **last two parameters are the most critical ones**, with the **peak_width** being the most critical of all.

It is also important to notice that the noise removal parameters of Pre-processing, used to remove minority peaks, have very small default values close to the baseline of evaluated samples. For different samples they could also compromise the correspondence between MS^1 and MS^2. By default, Pre-processing set a minimum intensity equal to 500 to consider a MS^1 peak (parameter **noise**) and only consider MS^1 peaks that have at least one ion with 750 of intensity (parameter **pre_filter**). If the minimum required MS^1 intensities are higher than the samples MS^1 baseline, real MS^1 peaks may be removed and the rate of non-correspondence between MS^1/MS^2 may increase (no matches in terms of m/z can be greatly affected here). The user must properly set the noise removal parameters values in Pre-processing to prevent this from happening (the minimum experimental intensity used to detect MS^2 ions is a good upper limit) and to better optimize the other critical parameters.

The user must take the following relations into account when choosing the critical parameters values:

- ppm tolerance:
  - A small value can split MS1 peaks from the same ion into distinct peaks, with close but different m/zs
  - A big value can join MS1 peaks from different ions with different m/zs
- Retention time tolerance and m/z tolerance:
  - A small value can prevent the match between a MS2 spectrum and its respective MS1 peak
  - A big value can assign a MS2 spectrum to a wrong MS1 peak (this can hide a bad MS1 integration processing)
- Minimum peak width:
  - A small value can split short MS1 peaks
  - A big value can exclude very short peaks and/or join adjacent isomers
- Maximum peak width:
  - A small value can split large MS1 peaks
  - A big value can join adjacent isomers

As you can see by now, this optimization will be a balance between the amount of effort that the user is willing to give and how good is the MS1 peaks definition across the different m/zs of a dataset.

At the end of the pre-processing step a diagnostic of its results is performed and stored in a log file that will be named 'logPreProcessStatisticsWarning' if a problem is detected in the diagnostic, or will be simple named 'logPreProcessStatistics' if no problem is detected. This log is saved to the directory named with the 'processed_data_name' inside the 'raw_data_path' folder, e.g. the pre-process output folder.

The pre-process statistics log will start with the statistics of the percentage of MS2 spectra without a MS1 peak correspondence. This is presented using two variables, better described in the last section 4.3.3., which are: (1) **rate_MS2_no_MS1_mz** - the percentage of MS2 spectra without a MS1 m/z range correspondence; and (2) **rate_MS2_no_MS1_rt** - the percentage of MS2 spectra without a MS1 retention time range correspondence. These rates are computed for each sample and removing blank m/zs.

If any of the not blank samples have one of these rates above a cutoff of 5%, a report is printted informing the top 5 problematic m/zs by sample. In these case the user should visualize the chromatogram of one or all (recommended) the printted m/zs to identify the problem(s). These could include bad defined peaks, for example jagged peaks or peaks with a long tail, or peaks with a width different from the values used in the **peak_width** parameter. The NP^3 MS workflow command *chr* can be used for this visualization or any other LC-MS/MS visualization tool. And then, the user have to decide to repeat the pre-process step with other parameters values or to accept this result. A descriptive statistics of these rates (mean, median and quartiles values) is also printted to help the user in this decision. When the job have a big number of samples it can be more reasonable to pay more attention to the mean/median values of these rates instead of expecting all samples to be below the cutoff (the best parameters for a given sample can give bad results in another sample).

At the end of the diagnostic of MS1 and MS2 correspondence, a diagnostic to check if real peaks are being split is also performed. It will count the number of MS1 peaks (possible isomers) by m/z for each sample and aggregate this counts into a single distributions. Then, if more than 5% of the m/zs have more than 30 MS1 peaks a warnings will be printed indicating that real peaks could have being split into multiple smaller peaks. The user must evaluate if this is not expected for the dataset being used, and then choose to increase the maximum peak_width value to prevent this anomaly. The distribution of the number of m/zs by the number of MS1 peaks is stored in the file ‘log_number_mzs_by_number_peaks.png’ for further examination.

Following the diagnostic, a Beta suggestion for the **peak_width** and the **rt_tolerance** parameters values will be executed to help the user in this optimization. It starts by plotting the peak width distribution of the pre-process result, for all the MS1 peaks that had a MS2 correspondence and excluding blank m/zs. These distribution will show the common peak width returned by the LC-MS processing with the given parameters, then its important to notice that it will be biased by the parameters used, but it will also tend to go to the expected peak width for that dataset. For example, if the pre-process was executed with big values in the **peak_width** parameter, it will result in large MS1 peaks being detected even if that does not represent what is found in the dataset, because it will join adjacent peaks to fulfill the given **peak_width** values. And that’s why we recommend the user to start with a pre setup of parameters that better represent your dataset.

Depending on the diagnostic result, the user should choose to follow or not the suggested values and rerun the pre-process if needed. The peak width distribution of the pre-process result is stored in the file ‘MS1_list_with_MS2_noBlank_peak_width_hist.png’ to help the user in this decision.

The user can also look at the table with the list of detected MS1 peaks that had a MS2 correspondence to better understand the pre-process results. This table is named ‘MS1_list_with_MS2.csv’ and is located in the output folder of the pre-process step. When there is a target m/z, it is easier to look for it in this table to check if the results follow what is seeing in the chromatograms. Otherwise, the user could randomly choose a m/z to do it or use the problematic m/zs that appear in the diagnostic when a warning is emitted.

4.4. Step 3 and 4: Command *clustering*

Runs the NP3_MSCluster algorithm to perform the clustering of pre-processed MS/MS data into a collection of consensus spectra. It relies on spectra similarity and chromatographic dimensions. Then, runs the consensus spectra quantification (Step 4) to parse the clustering results and count the number of spectra and of peak area by sample. Finally, it computes appropriate indicators based on the sample’s types. It can also run the library spectra identifications (Step 6), and if necessary, it runs Step 2.

Options for the command *clustering*:

- *-n, --output_name* <name>: the job name. It will be used to name the output directory and the results from the final clustering integration step
- *-m, --metadata* <file> : path to the metadata table CSV file
- *-d, --raw_data_path* <path> : path to the folder containing the input LC-MS/MS raw spectra data files (mzXML format is recommended)
- *-o, --output_path* <path> : path to where the output directory will be created
- *-f, --fragment_tolerance* [x] : the tolerance in Daltons for fragment peaks. Peaks in the original spectra that are closer than this get merged by the NP3_MSCluster algorithm (default: 0.05)
- *-z, --mz_tolerance* [x] : this is the tolerance in Daltons for the m/z of the precursor that determines if two spectra will be compared and possibly joined. Used in the clustering job and in the library identifications (Step 6) (default: 0.025)
- *-p, --ppm_tolerance* [x] : the maximal tolerated m/z deviation in parts per million (ppm) to be used in the pre-processing step if needed (default: 5)
- *-a, --ion_mode* [x] : the precursor ion mode. One of the following numeric values corresponding to a ion adduct type: ‘1’ = [M+H]+ or ‘2’ = [M-H]- (default: 1)
- *-s, --similarity* [x] : the minimum similarity to be consider in the hierarchical clustering, starts in 0.70 and decrease to X in 15 rounds (default: 0.55)
- *-g, --similarity_blank* [x] : the minimum similarity to be consider in the hierarchical clustering of the blank clustering steps, starts in 0.70 and decrease to X in 15 rounds. Only used in the clustering of blank samples (column SAMPLE_TYPE equals ‘blank’ in the metadata table) (default: 0.3)
- *-t, --rt_tolerance* [x,y] : tolerances in seconds for the retention time width of the precursor that determines if two spectra will be compared and possibly joined. The first tolerance [x] is used in the data, blank and batches integration clustering steps; and the second tolerance [y] is for the final integration step between all samples (default: 1,2)

- *-y, --processed_data_name* [x] : the name of the folder inside the raw_data_path where the pre-processed data was stored. If the given folder does not exist it will be created and the pre-process will be run using the default values of the missing options. Otherwise, it will depend on the processed_data_overwrite parameter value (default: "processed_data")
- *-q, --processed_data_overwrite* [x] : A logical "TRUE" or "FALSE" indicating if the pre-processed data present in the processed_data_name folder should be overwritten and pre-processed again if it already exists (default: "TRUE")
- *-c, --scale_factor* [x] : the scaling method to be used in the fragmented peak's intensities before any dot product comparison (Step 3). Valid values are: 0 for the natural logarithm (ln) of the intensities; 1 for no scaling; and other values greater than zero for raising the fragment peaks intensities to the power of x (e.g., x = 0.5 is the square root scaling). [x] >= 0 (default: 0.5)
- *-e, --method* [name] : a character string indicating which correlation coefficient is to be computed. One of "pearson", "kendall", or "spearman" (default: "spearman")
- *-x, --min_peaks_output* [x] : the minimum number of fragment peaks that a spectrum must have to be outputted after the final clustering step (Step 3). Spectra with less than X peaks will be discarded. x >= 1 (default: 5)
- *-j, --tremolo_identification* [x] : (not Windows OS's) A logical "TRUE" or "FALSE" indicating if the Tremolo tool should be used for the spectral matching against the ISDB from the UNPD (default: "TRUE")
- *-b, --max_chunk_spectra* [x] : Maximum number of spectra to be loaded and processed in a chunk at the same time. In case of memory issues this value should be decreased (default: 3000)
- *-v, --verbose* [x] : for values X>0 show the scripts output information (default: 0)
- *-h, --help* : output usage information

4.4.1. *clustering* Results

A directory inside the *output_path* named with the *output_name* containing:

- A copy of the *metadata* file and the command line parameters values used in a file named 'logRunParms', for reproducibility
- A folder named 'outs' with the clustering steps results in separate folders containing:
  - A subfolder named 'count_tables' with the Step 4 quantifications in CSV tables named as '_(spectra|peak_area).csv'.
  - A subfolder named 'clust' with the cluster's membership files (which SCANS or msclusterID were joined)
  - A subfolder named 'mgf' with the resulting clusters consensus spectra in MGF files
  - A text file named 'logNP3MSClusterOutput' with the NP3_MSCluster log output.

The clustering steps results folders are named as 'B_<DATA_COLLECTION_BATCH>_<X>' where *DATA_COLLECTION_BATCH* is the data collection batch number in the metadata file of each group of samples and* X* is 0 if it is the result of a *data clustering step* or 1 if it is the result of a *blank clustering step*. The *data collection batch integration step* results are stored in folders named as 'B_<DATA_COLLECTION_BATCH>'.

The final integration step result is located inside the 'outs' directory in a folder named with the *output_name*. This folder contains the final counts and is where the user will find the final results. It also contains the tremolo identification results inside the 'identifications' folder.

4.4.2. *clustering* Examples

Fake examples to execute the Clustering command:

```
$ node np3_workflow.js clustering --output_name "test_np3" --output_path
  "/path/where/the/output/will/be/stored" --metadata
  "/path/to/the/metadata/file/test_np3_metadata.csv" --raw_data_path
  "/path/to/the/raw/data/dir"
```

Same example with different retention time tolerances:

```
$ node np3_workflow.js clustering -n "test_np3_rt_tol" -o
  "/path/where/the/output/will/be/stored" -m
```

```
    "/path/to/the/metadata/file/test_np3_metadata.csv" -d
    "/path/to/the/raw/data/dir" -t 3.5,5
```

4.4.3. *clustering* Step 3 Details Tandem mass spectrometry (MS/MS) experiments often generate redundant data sets containing multiple spectra of the same molecule. Clustering of MS/MS spectra takes advantage of this redundancy by identifying multiple spectra of the same ion and replacing them with a single representative spectrum [6].

The NP3_MSCluster is a modified version of the MS-Cluster algorithm [7]. The MS-Cluster is a simple and effective algorithm designed to rapidly process large MS/MS data sets (even in the excess of 10 million spectra), while ensuring the high quality of the resulting clusters. Its optimizations were made for proteomics focusing on processing peptides data. This open-source software was developed in C++ and is available for download at http://peptide.ucsd.edu. It is part of the GNPS workflow [8].

To use the MS-Cluster in the NP^3 MS workflow the bias toward proteomics had to be overcome with modifications toward metabolites processing. The major limitation of its use is that stereo isomers are not resolved as separate clusters, because the original algorithm did not use the retention time for joining the spectra. Another limitation is that this algorithm is not an optimal solution, it approximates a hierarchical clustering leading to fragmented clusters, that is, several distinct clusters containing spectra of a same ion. These fragmented clusters can lead to a wrong count of the detected ions from the same metabolite and the solution for this problem is presented in the section of clean Step 5.

An ion in a LC-MS/MS experiment is detected in a chromatographic peak (MS1 peak), which is characterized by a m/z range, a retention time range and the total intensity (peak area). The MS2 data are the fragmented ions (spectra) of the MS1 peaks. The *pre_process* step is capable of matching the peak dimension information of the MS1 peaks with all detected MS2 spectra, enriching the spectra with chromatographic information and making possible to use this information in the clustering job.

The NP3_MSCluster algorithm was developed to support the clustering of metabolites and to separate stereo isomers. It parses the MS1 peak dimensions information from the NP^3 MS workflow pre-processed MS/MS data and incorporates it in the cluster's attributes, by taking the average of its members peak dimensions. With the use of this additional information, it prevents the join of spectra that have the same m/z but different retention times (descendant from different MS1 peaks, not concurrent), what characterizes stereo isomers.

A tolerance in the spectra retention time range is used to deal with the calibration problem when multiple MS runs are being clustered and the runs were not carefully aligned or can't fully rely in the alignment. Due to the retention time tolerance very close by peaks can still be joined and the user must take this into account when setting this tolerance value. The default parameters values were refined for a UHPLC-MS/MS-qTOF equipment, the UHPLC Acquity HClass Waters and the spectrometer ESI-QqTOF Impact II Bruker.

The NP3_MSCluster algorithm only considerate two clusters within a given mass tolerance as joining candidates if the retention time mean (center of the peak) of one cluster is contained in the retention time range of the other cluster, within a retention time tolerance applied in the peak boundaries. Then the clustering happens as in the original MS-Cluster algorithm. If the cosine similarity of the joining candidates is above the threshold, they will be joined in a consensus spectrum, called the cluster representative. The consensus spectrum peak dimensions attributes are computed by summing the total MS2 intensities of the cluster members and using it to calculate the weighted average of the members peak dimensions values, with the precursor intensity of each cluster member as weight. At the end of the clustering process each consensus cluster will represent the spectrum of one MS1 peak, except for the fragmented clusters limitation and the possible join of very close by peaks (very close by isomers).

The MS-Cluster configurations and some hard coded parameters had to be modified to eliminate its bias toward proteomics. The filtering (*sqs*) and the annotation parameters (*assign-charges* and *correct-pm*) were disabled from the default settings following the authors recommendation [9]. To reduce the loss of information from the detected MS2 spectra, the hardcoded filter of spectra with too few numbers of fragmented peaks was modified in the entire code to only filter spectra with no fragmented peaks. The original algorithm filtered spectra with less than 7 peaks in the input and with less than 15 peaks in the output. A parameter was added to allow the user to control this filtering in the final clustering step, by choosing the minimum number of fragmented peaks that a spectrum must have to be outputted (default to 5). Some other hard coded heuristic parameters were relaxed to allow more comparisons, aiming to reduce the fragmented clusters occurrence with exchange in performance (see file NP3_MSCluster/src/MsCluster/MsClusterIncludes.h for the list of changed parameters).

Scaling peak intensities has been shown to improve the quality of the similarity computations [10]. The scaling method adopted by the MS-Cluster algorithm is the natural logarithm of the peaks' intensities, which they found to be the most suitable for their data [7]. But for our metabolomic data we found that the square root scaling method was the most suitable one and it is the recommended one for dot-product algorithms (cosine) [10,11]. To overcome this limitation a new parameter was added to allow the user to choose which scaling method is to be used in the entire workflow, and the available options are: no scaling; natural logarithm scaling; and scaling to the power of x, where x can assume any value greater than zero (by default set to square root scaling x = 0.5).

Relying in the principle of similarity between experimental spectra [7,10] the samples clustering is performed in steps, based on the defined data collection batches and the samples types, in the metadata table. Thus, the more similar spectra, guaranteed of being detected in the same conditions, are forced to be joined previous and more information be aggregated to them before clustering all the samples spectra together.

The NP^3 MS workflow spectra clustering is performed in four steps:

1. The *data clustering step* - all the not blank samples (column SAMPLE_TYPE equals 'sample', 'bed', 'control' or 'hit' in the metadata table) of each data collection batch are clustered together;
2. The *blank clustering step* - all the blank samples (column SAMPLE_TYPE equals 'blank') of each data collection batch are clustered together, to try to avoid the noise impact of these samples in the remaining clustering steps;
3. The *data collection batch integration step* - the results of the data and the blank clustering steps of each data collection batch are clustered together;
4. And the *final integration step* - the results of all data collection batch integration steps are clustered together to integrate all results.

The NP3_MSCluster results are the MGF with the collection of consensus spectra from the outputted clusters and .clust files (in CSV format) with the cluster's information describing which scans were joined to make up the consensus spectra. In the data and in the blank clustering steps the resulting .clust files contains the spectra detected in the raw files (real SCANS), with the matched peak information, and in the subsequent clustering steps the SCANS are the clusters IDs (named msclusterID) of the consensus spectra created in the previous steps by the workflow Step 4.

4.4.4. *clustering quantification* Step 4 Details The spectra and peak area counts are performed by scanning and parsing the information present in the .clust files from the clustering output to compute the consensus spectra quantifications by sample. It aggregates the information of the clusters members to obtain the total counts (number of spectra and peak area) by sample and the consensus peak profile information. A unique cluster ID is given to the consensus spectra, named msclusterIDs.

In the data and in the blank clustering steps the clusters members are the raw data SCANS, what gives for each cluster the number of spectra (SCANS's) and the peak profile information by sample. In the following integration steps the clusters members are the consensus spectra created in previous steps and the counts only merge and aggregate the values computed previously from the raw data.

The count of spectra quantification is obtained by summing for each cluster the number of MS2 SCANS detected in each sample, what gives the total number of times that a spectrum within a m/z tolerance was detected in a retention time range by sample. The count of peak area quantification is quite different, it is based on the peak's IDs of the cluster's members. The peak ID is used to detected the sample of origin and to parse the pre-processed data to extract the respective peak area, and then for each unique peak ID of the clusters members the count is obtained by summing the peak areas by sample.

The spectra and peak area count of all the blank samples (column SAMPLE_TYPE equals 'blank' in the metadata table) are summed for each m/z (row) and the result stored in a new column named *BLANKS_TOTAL*. The same is performed for control and bed samples (SAMPLE_TYPE equals "control" and "bed", respectively), resulting in two new columns named *CONTROLS_TOTAL* and *BEDS_TOTAL*, respectively. These columns containing the total counts by sample type can help filter false positive candidates and are used to compute the flag indicators described below.

The counting script adds at most three new boolean (TRUE or FALSE) columns named *BFLAG*, *CFLAG* and *BEDFLAG*, if the respective sample type equals blank, control or bed is present in the metadata table. They are computed as follow: if a msclusterID have the column *BLANKS_TOTAL*, *CONTROLS_TOTAL* or *BEDS_TOTAL* greater than zero all spectra that have a m/z's within the given mass tolerance from it's m/z will have the column *BFLAG*, *CFLAG* or *BEDFLAG*, respectively, set to TRUE, otherwise it will be FALSE. These column flags serve as a warning for false positive msclusterIDs, they signalize that there is a very close m/z that was detected in a blank, control or bed sample respectively, and is up to the user to check its correctness.

The 'hit' sample type is used to dereplicate the m/z's that appeared in one of the samples that had a high bioactivity score or that has a m/z close enough to other m/z that appears in one of these samples. The counting scripts adds two new character columns named *DESREPLICATION* and *HFLAG*. The first stores all the hit samples codes in which each msclusterID have a count greater than zero. And the second column signalize all the hit samples codes that have a count greater than zero in any cluster ID with a m/z within the given mass tolerance of each msclusterID m/z (this column contains the first column results). The same way as the other flag columns, the *HFLAG* serve as a warning for false negative clusters, evidencing a spectrum that has a m/z very similar to a spectrum that appeared in a hit sample and is up to the user to check its correctness.

The resulting clustering count table has m rows and at most $20 + n$ columns, where m equals the number of consensus spectra in the end of the clustering job and n equals the number of processed samples present in the metadata file. A description of each column is presented below (as described above the weighted mean values are obtained using the cluster members MS2 summed intensities as weights).

Table 1: NP^3 MS workflow Clustering Count Table Format

Columns	Description	Value Type
msclusterID	the consensus spectrum unique cluster ID	numeric
numSpectra	total number of joined spectra (cluster members), cluster size	numeric
mzConsensus	weighted mean of the cluster members m/z	numeric
rtMean	weighted mean of the cluster members retention time center	numeric
rtMin	weighted mean of the cluster members retention time minimum	numeric
rtMax	weighted mean of the cluster members retention time maximum	numeric
peakIds	the peak IDs of the cluster members concatenated without duplicates	string
scans	the MGF SCANS number with the sample of origin as suffix of the cluster members concatenated	string
basePeakInt	the maximum base peak intensity value of the cluster members	numeric
sumInts	sum of the cluster members MS2 intensities	numeric
<X_1>_<Y>	the count of Y in the sample X_1, where Y can be 'spectra' for the number of spectra or 'area' for the unique sum of peak areas in the sample with SAMPLE_CODE equals X_1 in the metadata table	numeric
...		
<X_n>_<Y>	the other samples counts for this mslusterID	numeric

Columns	Description	Value Type
BLANKS_TOTAL	total number of spectra or peak area that appeared in blank samples	numeric
CONTROLS_TOTAL	total number of spectra or peak area that appeared in control samples	numeric
BEDS_TOTAL	total number of spectra or peak area that appeared in bed samples	numeric
DESREPLICATION	concatenation of all the hit samples codes in which the spectra had a count greater than zero	string
BFLAG	TRUE or FALSE indicating if there is a close m/z to the mzConsensus in any blank sample	boolean
CFLAG	TRUE or FALSE indicating if there is a close m/z to the mzConsensus in any control sample	boolean
BEDFLAG	TRUE or FALSE indicating if there is a close m/z to the mzConsensus in any bed sample	boolean
HFLAG	concatenation of all the hit samples codes that have a count greater than zero in any msclusterID with a mzConsensus within the given mass tolerance of the current cluster (contains the DESREPLICATION column results)	string
peaksList	the concatenated list of the fragmented peaks m/z's of the msclusterID consensus spectrum	string
peaksInt	the concatenated list of the fragmented peaks intensities of the msclusterID consensus spectrum	string

At the end of the clustering step, the distribution of the base peak intensity values (column basePeakInt) is plotted and saved to a file named 'basePeakInt_distribution.png'. A descriptive analysis of the basePeakInt distribution is also computed and saved to a file named 'basePeakInt_distribution_summary.txt'. If there is at least one blank sample in the metadata table, the base peak distribution plot will be colored to highlight the values from the consensus spectra that appear in blank samples and the descriptive analysis will be computed only for these consensus spectra from blank samples. The interquartile range (IQR) is also computed and printed at the end of the descriptive analysis file to help in the setup of the noise cutoff and of the bflag cutoff parameters (to be used in the clean Step 5). Also, vertical lines corresponding to different noise/bflag cutoff values are plotted in the base peak intensity distribution.

4.5. Step 5: Command *clean*

Compute the pairwise comparisons of the consensus spectra (if not done yet) and then clean the clustering counts. It also runs Step 7 to annotate possible ion variants using the new clean tables and to create the molecular network of annotations, and runs Step 10 to overwrite any old computation of the molecular network of similarities. It can also run the library spectra identifications (Step 6) for the collection of clean consensus spectra.

Options for the command *clean*:

- *-m, --metadata* <file> : path to the metadata table CSV file
- *-o, --output_path* <path> : path to the final output data folder, inside the 'outs' directory of the clustering result folder. It should contain the 'mgf' folder and the 'count_tables' folder with the peak area and spectra count tables in CSV files. The job name will be extracted from here
- *-y, --processed_data_dir* <path> : the path to the folder inside the raw data folder where the pre-processed data (MGFs) were stored.
- *-z, --mz_tolerance* [x] : the tolerance in Daltons that determines if two spectra will be compared and possibly joined. It is also used in Step 7 to detect possible ion variants (default: 0.025)
- *-t, --rt_tolerance* [x,y] : tolerances in seconds for the retention time width of the precursor that determines if two spectra will be compared and possibly joined. It is directly applied to the retention

time minimum (subtracted) and maximum (added) of the spectra. It enlarges the peak boundaries to deal with disaligned samples or ionization variant spectra. The first tolerance [x] is used in the annotation Step 7; and the tolerance [y] is used in the clean Step 5 (default: 1,2)

- *-a, --ion_mode* [x] : the precursor ion mode. One of the following numeric values corresponding to an ion adduct type: ‘1’ = [M+H]+ or ‘2’ = [M-H]- (default: 1)
- *-i, --similarity_function* [x] : the similarity function to be used in the spectra comparison to create the pairwise similarity tables after clustering and clean steps. One of “np3_shifted_cosine” or “spec2vec”. If “spec2vec” is selected, the model trained on UniqueInchikey subset (12,797 spectra) is used by spec2vec in the spectra comparison and the matchms library is used to compute the number of matched peaks between the compared spectra; otherwise, the NP3 shifted cosine function is used. (default: “np3_shifted_cosine”)
- *-s, --similarity* [x] : the similarity to be consider when joining clusters and merging their counts in not blank samples (default: 0.55)
- *-g, --similarity_blank* [x] : the similarity to be consider when joining clusters and merging their counts in blank samples (column SAMPLE_TYPE equals ‘blank’ in the metadata table) (default: 0.3)
- *-w, --similarity_mn* [x] : the minimum similarity score that must occur between a pair of consensus spectra to connect them with an edge in the molecular networking. Lower values will increase the component size of the clusters by inducing the connection of less related spectra; and higher values will limit the components sizes to the opposite (default: 0.6)
- *-f, --fragment_tolerance* [x] : the tolerance in Daltons for fragment peaks. Peaks in a cluster spectrum that are closer than this are considered the same. (default: 0.05)
- *--bflag_cutoff* [x] : A positive numeric value to scale the interquartile range (IQR) of the blank spectra basePeakInt distribution from the clustering result and to allow spectra with a basePeakInt value below this distribution median plus IQR*bflag_cutoff to be joined with a blank spectrum during the clean Step 5, without relying on the similarity value. Or FALSE to disable it. The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile) of the distribution. The spectra with a basePeakInt value <= median + IQR*bflag_cutoff (from the blank spectra basePeakInt distribution) and BFLAG TRUE will be joined to a blank spectrum in the clean Step 5. This cutoff will affect the spectra with BFLAG TRUE that would not get joined to a blank spectra when relying only on the similarity cutoff. This is a turn around to the fact that blank spectra have low quality spectra and thus can not fully rely on the similarity values. (default: 1.5)
- *--noise_cutoff* [x] : A positive numeric value to scale the interquartile range (IQR) of the blank spectra basePeakInt distribution from the clustering Step 3 result and to remove the spectra with a basePeakInt value below this distribution median plus IQR*noise_cutoff after the clean Step 5. Or FALSE to disable it. The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile) of the distribution. When no blank sample is present in the metadata, the full distribution is used. This cutoff will affect the spectra with with a low basePeakInt value that probably are noise features. If the clustering Step 3 resulted in more than 25000 spectra, the noise cutoff will be applied before the clean Step 5 to prevent a long processing time (default: “FALSE”)
- *-u, --rules* [x] : path to the CSV file with the accepted ionization modification rules for detecting adducts, multiple charge and dimers/trimers variants, and their combination with neutral losses (Step 7). (default: “rules/np3_modifications.csv”)
- *-c, --scale_factor* [x] : the scaling method to be used in the fragmented peak’s intensities before any dot product comparison (Step 5). Valid values are: 0 for the natural logarithm (ln) of the intensities; 1 for no scaling; and other values greater than zero for raising the fragment peaks intensities to the power of x (e.g. x = 0.5 is the square root scaling). [x] >= 0 (default: 0.5)
- *--min_matched_peaks* [x] : The minimum number of common peaks that two spectra must share to be connected by an edge in the filtered SSMN. Connections between spectra with less common peaks than this cutoff will be removed when filtering the SSMN. Except for when one of the spectra have a number of fragment peaks smaller than the given min_matched_peaks value, in this case the spectra must share at least 2 peaks. The fragment peaks count is performed after the spectra are normalized and cleaned. (default: 6)
- *-k, --net_top_k* [x] : the maximum number of connections for one single node in the similarity molecular networking. An edge between two nodes is kept only if both nodes are within each other’s X most

similar nodes. This restriction is applied to the spectra following the msclusterID's order, so smaller m/z's are limited first. Keeping this value low makes very large networks (many nodes) much easier to visualize (default: 10)

- *-x, --max_component_size* [x] : the maximum number of nodes that all component of the similarity molecular network must have. The edges of this network will be removed using an increasing cosine threshold until each network component has at most X nodes. Keeping this value low makes very large networks (many nodes and edges) much easier to visualize. (default: 200)
- *--blank_expansion* [x] : the distance of neighborhood nodes from the blanks in IVAMN to be selected for removal in the final protonated networks. (0) to only remove blanks nodes, (1) to remove nodes directly connected to a blank node, (2 or greater) to remove nodes in a distance equal to 2 or greater from a blank node, or (-1) to remove all possible neighbours and ancestors of a blank node (remove blank clusters) from IVAMN (default: 0)
- *-r, --trim_mz* [x] : A logical "TRUE" or "FALSE" indicating if the spectra fragmented peaks around the precursor m/z +-20 Da should be deleted before the pairwise comparisons. If "TRUE" this removes the residual precursor ion, which is frequently observed in MS/MS spectra acquired on qTOFs. (default: "TRUE")
- *--max_shift* [x] : Maximum difference between precursor m/zs that will be used in the search of shifted m/z fragment ions in the NP3 shifted cosine function. Shifts greater than this value will be ignored and not used in the cosine computation. It can be useful to deal with local modifications of the same compound. (default: 200)
- *-l, --parallel_cores* [x] : the number of cores to be used for parallel processing (Step 5). x = 1 for disabling parallelization and x > 2 for enabling it. [x] >= 1 (default: 2)
- *-e, --method* [name] : a character string indicating which correlation coefficient is to be computed. One of "pearson", "kendall", or "spearman" (default: spearman)
- *-j, --tremolo_identification* [x] : (not Windows OS's) A logical "TRUE" or "FALSE" indicating if the Tremolo tool should be used for the spectral matching against the ISDB from the UNPD (default: "TRUE")
- *-b, --max_chunk_spectra* [x] : Maximum number of spectra (rows) to be loaded and processed in a chunk at the same time. In case of memory issues this value should be decreased (default: 3000)
- *-v, --verbose* [x] : for values X>0 show the scripts output information. (default: 0)
- *-h, --help* : output usage information

4.5.1. ***clean*** Results

One subfolder inside the 'count_tables' folder is created named 'clean' containing:

- A text file named 'analyseCountClusteringClean' with the clean count analyses
- Two CSV files with the clustering counts of spectra and peak area cleaned and annotated, named with the suffix '_clean_ann.csv'
- CSV files with the correlation columns added are also included when there is a biocorrelation result

The 'molecular_networking' folder is also created if not present yet, and inside it is created:

- One subfolder named "similarity_tables" with the clean version of the pairwise table of similarity;
- Five molecular networks edge files (Steps 7 and 10):
  - The ionization variant annoation molecular network named as: '<*output_name*>_ivamn.selfloops'
  - The complete spectra similarity molecular network named as : '<*output_name*>_ssmn_w_<*similarity_mn*>.selfloops', which contains all links with a similarity value above the cut-off
  - The filtered spectra similarity molecular network named as '<*output_name*>_ssmn_w_<*similarity_mn*>_k_<*net_top_k*>_x_ <*max_component_size*>.selfloops'
  - The IVAMN [M+H]+ named as: '<*output_name*>_ivamn_protonated.selfloops'
  - The SSMN [M+H]+ filtered named as: '<*output_name*>_ssmn_protonated_w_<*similarity_mn*>_k_<*net_top_k*>_x_

<*max_component_size*>.selfloops'

- One CSV table containing the molecular network of annotations attributes and the assigned [M+H]+ representatives, named as '<*output_name*>_ivamn_attributes.csv' (Step 7)

Where the 'output_name' is extracted from the 'output_path';

When running Tremolo the 'identification' folder is also created (if not present yet) with the identifications results inside it. These identifications are also added as new columns in the created clean count tables.

4.5.2. *clean* Examples Fake example to execute the Cleaning command:

```
$ node np3_workflow.js clean --metadata
  "/path/to/the/metadata/file/test_np3_metadata.csv"
  --output_path "/path/to/the/output/dir/test_np3/outs/test_np3" -b 1000
```

4.5.3. *clean* Details The clustering (Step 3) identify multiple spectra of the same ion and replace them with a single representative consensus spectrum. The MS-Cluster algorithm is not an optimal solution, so even though it drastically reduces the data size keeping it's quality high, it is not able to remove all the repeated spectra of the same ion, leading to fragmented clusters. In the NP3_MSCluster algorithm, fragmented clusters are clusters with the same m/z within a m/z tolerance and that coeluted in a concurrent chromatographic peak (detected in the same retention time range) within a retention time tolerance.

The clean step is a second clustering strategy that was developed to reduce the remaining redundancies from the clustering result, and to overcome the fragmented clusters limitation. It is an optimal clustering with greedy heuristics that was implemented in R code with auxiliary compiled functions (dlls) in C++.

The *clean* step starts by performing the pairwise comparisons of the consensus spectra. These comparisons result in two symmetric and quadratic matrix (n + 1)x(n + 1), where n is the number of consensus spectra from Step 3 and the remaining 1 row and column are the msclusterIDs (clusters IDs resulting from the clustering step). The upper triangular part of these matrix contains in one the similarity values and in the other the number of matched peaks between all consensus spectra. And their lower triangular part is empty to reduce repeating the same computations. These similarity values are computed using a shifted version of the cosine function [8,9], which we call NP^3 shifted cosine function, or spec2vec [24], using the model trained on UniqueInchikey subset (12,797 spectra). If spec2vec is used, the matchms [25] library is used to compute the number of matched peaks between the compared spectra.

The NP^3 shifted cosine algorithm first matches the same m/z's from fragmented peaks list of the spectra being compared within a mass tolerance; then, for the remaining not matched fragmented peaks of the spectra it adds a shift equals to the difference of the spectra precursor m/z in all fragmented peaks' m/z of the spectrum with the smaller precursor m/z. And finally, it tries to match again the remaining peaks of the spectra within the mass tolerance. The shift is used to overcome a small motif difference that could be present in the ions structure, it will produce a shift in some fragmented peaks m/z's equal to the motif mass. Only motifs greater than the m/z tolerance and smaller than the max_shift parameter value are considered, a precursor m/z difference outside this range will be set to zero, and then, the shift won't be applied. Using the precursor m/z difference in the cosine computation is a heuristic that can increase the similarity score of analogue structures and possible ionization variants (e.g., different adducts, neutral losses, etc). This implementation was inspired by the modified cosine implemented in the GNPS workflow [8].

Next, the *clean* step use the computed similarity table of the resulting consensus spectra to reduce the clustering count tables. It works in the following steps:

1. Data cleaning step: the fragmented clusters are joined if their consensus spectra have a similarity value above the cutoff, or if they share at least one peakId or if they pass the bflag cutoff criteria. The similarity table is reduced based on the performed joins always keeping the maximum similarity value between the aggregated rows and columns (greedy heuristic). This procedure is repeated at most 10 times to take into account the results of the performed aggregations of previous steps, it stops earlier if no join is performed. At the first time of this step, only blank spectra are considered for the joining

and teh bflag cutoff is applied. Finally, a noise cutoff is applied to remove spectra with a low base peak intensity value;

2. Indicators step: the samples types indicators are computed for each row of the just obtained clean counts. All the indicators of Step 4 are computed plus another indicator showing the distance of each spectra to a blank spectrum (when blank samples are present). The joined clusters have their fragmented peak list and intensities aggregated in a consensus that is stored in the clean count table, and saved to the clean MGF file. The IDs of the joined clusters are stored in the clean count tables as well, in a column named 'joinedIds';

In the fragmented clusters search of the Data cleaning step, an additional condition was added in the criteria to join two spectra. It checks if the peak center devition between the spectra is less than 4 times the retention time tolerance or if their peak boundaries deviation is less than 2 times the retention time tolerance. If both of these conditions are not satisfied, the spectra are kept separated. By doind this, we try to prevent joining adjacent isomers that have a high similarity value.

Also in the Data cleaning step, a bflag cutoff was implemented to allow joining spectra from blank samples that do not have a similarity value above the cutoff. This helps to reduce the fragmented clusters with bflag TRUE and a low base peak intensity, that could not fully rely in the similarity values. The bflag cutoff is computed as the median value of the base peak intensity distribution of blank spectra plus the factor informed by the user times the interquartile range (IQR) of this distribution. The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile) of a distribution. The consensus spectra with a basePeakInt value <= median + IQR*bflag_factor (from the blank spectra basePeakInt distribution) and BFLAG TRUE will be joined to a blank spectrum independent of their similarity values.

At the end of the clean step a noise cutoff is applied to remove spectra with a low base peak intensity value. The noise cutoff is computed as the the median value of the base peak intensity distribution of blank spectra plus the factor informed by the user times the interquartile range (IQR) of this distribution. When no blank sample is present in the metadata, the full distribution of the base peak intensity from the clustering counts is used. This cutoff will affect the consensus spectra with a low basePeakInt value that probably are noise features. If the clustering Step 3 resulted in more than 15000 consensus spectra, the noise cutoff will be applied before the clean step to prevent a long processing time.

The base peak intensity distribution plotted at the end of the clustering step helps the users to better define this cutoffs' factors and to better undersand its effect on their dataset.

The count tables, in terms of the number of spectra and of peak area, of the joined clusters are aggregated following the same rules applied in Step 4 and the joined clusters peak information (m/z, retention time mean, minimum and maximum) are computed as the average of the joining clusters values weighted by their intensities (sumInts).

The following columns are added to the count tables in the *clean* step:

Table 2: Clean Count Tables New Columns from the Data Cleaning Step

Columns	Description	Value Type
joinedIDs	the msclusterIDs of the joined spectra in the cleaning step, separated by a ';'	character with concatenated numbers
numJoins	the number of spectra that were joined in the cleaning step	numeric
BLANK_DIST	the distance in the molecular network of similarity from the current spectra to a spectrum of a blank sample. This distance ranges from 0 (if the current spectra appear in a blank sample) to 3 (if there are at least 3 links between the current spectra and a spectrum of a blank sample in the molecular network of similarity). If this value is NA it means that this distance is at least greater than 3, and thus was not computed.	numeric

4.6. Step 6: Command *tremolo*

Runs the library spectra identification of the collection of consensus spectra against the In-Silico predicted MS/MS spectrum of Natural Products Database (ISDB) [16] in positive ion mode for 170 602 Natural Products from the Universal Natural Products Database (UNPD) [17] using the tremolo tool [18]. This tool can only be executed in Unix OS and for data in positive ion mode.

The NPClassifier [21], NPAtlas [22] and ClassyFire [23] solutions were used to add origin and class information to the UNPD compounds. NP^3 MS Workflow will retrieve this information together with the identification results.

A binary version of the tremolo tool is freely available for download in the following link: http://proteomics.ucsd.edu/software-tools/tremolo/.

Options for the command *tremolo*:

- *-o, --output_path* : path to where the spectral library search results will be stored
- *-g, --mgf* : path to the input MGF file with the MS/MS spectra data to be searched and identified
- *-c, --count_files [name]* : optional paths to the count CSV files, separated by a comma and no space, as outputted by the NP^3 MS workflow where the identifications should be added as new columns. The top_k search results for each msclusterID will be added in 8 identification columns generated by tremolo. The count files header must be in the first row. (default: " ")
- *-z, --mz_tolerance* : the tolerance for parent mass search in Daltons. Set a small tolerance for dereplication using parent ion mass as prefilter, keeping in mind the resolution of your data. Increase to the wanted range for variable dereplication search (ex: 100 or 200 Da) Caution as this will also increase calculation times! (default: 0.025)
- *-s, --similarity* : the similarity threshold that determines if two spectra are the same. Should be kept low when using in-silico DB, as recommended by the authors. Typically, 0.2 to 0.3 (default: 0.2)
- *-k, --top_k* : defines the maximal number of results returned by the tremolo tool, e.g. number of identifications by msclusterID
- *-v, --verbose* : for values X>0 show the scripts output information (default: 0)
- *-h, --help* : output usage information

4.6.1. *tremolo* Results Two files are created inside the 'output_path' folder (this folder is created if missing):

- 'logTremolo' a text file with the tremolo tool output information;
- 'tremolo_results.csv' a CSV file with the tremolo search results for each spectra SCANS present in the provided MGF file.

When the count_files are provided the following columns are added to them:

- tremolo_UNPD_IDs : the UNPD IDs of the top_k identified spectra concatenated with a ';'
- tremolo_SMILES : the SMILES of the top_k identified spectra concatenated with a ','. A comma is used here to allow including this information in cytoscape to visualize the nodes annotated identified structure
- tremolo_chemicalNames : the chemical names of the top_k identified spectra concatenated with a ';'
- tremolo_InChIKey : the InChIKey of the top_k identified spectra concatenated with a ';'
- tremolo_molecularFormula : the molecular formula of the top_k identified spectra concatenated with a ';'
- tremolo_molecularWeight : the molecular weight of the top_k identified spectra concatenated with a ';'
- tremolo_CAS : the CAS number of the top_k identified spectra concatenated with a ';'
- tremolo_MQScore : the MQ score computed by tremolo of the top_k identified spectra concatenated with a ';'. Similarity score, evaluates the quality of the match
- tremolo_mzErrorPPM : the m/z error in PPM for the top_k identified spectra concatenated with a ';'. Mass error, evaluates the quality of the match and signalizes analogs
- tremolo_numSharedPeaks : the number of shared peaks with the top_k identified spectra concatenated with a ';'. Shared peaks in the match, evaluates the quality of the match

- tremolo_NPClassifier_superclass : the chemical superclass from NPClassifier of the top_k identified spectra concatenated with a ';'
- tremolo_NPClassifier_class : the chemical class from NPClassifier of the top_k identified spectra concatenated with a ';'
- tremolo_ClassyFire_subclass : the chemical subclass from ClassyFire of the top_k identified spectra concatenated with a ';'
- tremolo_NPClassifier_pathway : the chemical pathway from NPClassifier of the top_k identified spectra concatenated with a ';'
- tremolo_NPClassifier_isglycoside : the chemical is glycoside tag from NPClassifier of the top_k identified spectra concatenated with a ';'
- tremolo_NPAtlas_id : the NPAtlas ID of the top_k identified spectra concatenated with a ';'
- tremolo_NPAtlas_compound_names : the compound names from NPAtlas of the top_k identified spectra concatenated with a ';'
- tremolo_NPAtlas_origin_type : the biological origin type from NPAtlas of the top_k identified spectra concatenated with a ';'
- tremolo_NPAtlas_genus : the biological genus from NPAtlas of the top_k identified spectra concatenated with a ';'
- tremolo_NPAtlas_origin_species : the biological origin species from NPAtlas of the top_k identified spectra concatenated with a ';'

For more details about the Tremolo tool see the src/ISDB_tremolo_NP3/README.md file and its documentation at src/ISDB_tremolo_NP3/Data/doc/Workflow_062016.pdf.

4.6.2. *tremolo* Examples Fake example to execute the Tremolo command:

```
$ node np3_workflow.js tremolo --output_path
  "/path/to/the/output/dir/test_np3/outs/test_np3/identifications" --mgf
  "/path/to/the/output/dir/test_np3/outs/test_np3/mgf/test_np3_all.mgf"
```

Concatenate tremolo results in the count files:

```
$ node np3_workflow.js tremolo -o
  "/path/to/the/output/dir/test_np3/outs/test_np3/identifications"
  -g "/path/to/the/output/dir/test_np3/outs/test_np3/mgf/test_np3_all.mgf" -c
  "/path/to/the/output/dir/test_np3/outs/test_np3/test_np3_spectra_clean_ann.csv,
  /path/to/the/output/dir/test_np3/outs/test_np3/test_np3_peak_area_clean_ann.csv"
```

4.7. Step 7: Command *annotate_protonated*

Annotate possible ionization variants in the clean count tables and creates the ionization variant annotation molecular network (IVAMN) - for positive ion mode only. It searches for adducts, neutral losses, multiple charge, dimers/trimers, isotopes and in-source fragments based on numerical equivalences and chemical rules. Finally, it runs a link analysis in the IVAMN to assign some consensus spectra as putative [M+H]+ representatives.

The number of [M+H]+ representatives offer a suggestion to the number of real metabolites present in the samples.

The nodes of the IVAMN are the set of clean consensus spectra and the links of this network connects two consensus spectra that have a chemical annotation. The links have a direction, pointing from the consensus spectra considered as an ion variant to the consensus spectra considered as a putative [M+H]+ candidate in the respective annotation (e.g., [M+Na]+ -> [M+H]+).

Options for the command *annotate_protonated*:

- *-m, --metadata* <file> : path to the metadata table CSV file

- *-o, --output_path* <path> : path to the output data folder, inside the 'outs' directory of the clustering result folder. It should contain the 'counts_table' folder and inside it the 'clean' subfolder with the clean count tables in CSV files. The job name will be extracted from here
- *-z, --mz_tolerance* [x] : the tolerance in Daltons for matching the numerical rules of detecting adducts, neutral losses, multiple charge, dimers/trimers and isotopes variants (default: 0.025)
- *-f, --fragment_tolerance* [x] : the tolerance in Daltons for matching the numerical rules of in-source fragments and multiple charge isotopic patterns. (default: 0.025)
- *-t, --rt_tolerance* [x] : tolerance in seconds to enlarge the peak boundaries for detecting concurrent variant spectra. (default: "2")
- *-i, --absolute_ms2_int_cutoff* [x] : The absolute intensity cutoff for fragmented MS2 peaks in the interval of 0 to 1000 (default: 15)
- *-a, --ion_mode* [x] : the precursor ion mode. One of the following numeric values corresponding to an ion adduct type: '1' = [M+H]+ or '2' = [M-H]- (default: 1)
- *-u, --rules* [x] : path to the CSV file with the accepted ionization modification rules for detecting adducts, multiple charge and dimers/trimers variants, and their combination with neutral losses. (default: "rules/np3_modifications.csv")
- *-c, --scale_factor* [x] : the scaling method to be used in the fragmented peak's intensities before any dot product comparison (Step 5). Valid values are: 0 for the natural logarithm (ln) of the intensities; 1 for no scaling; and other values greater than zero for raising the fragment peaks intensities to the power of x (e.g. x = 0.5 is the square root scaling). [x] >= 0 (default: 0.5)
- *-b, --max_chunk_spectra* [x] : Maximum number of spectra (rows) to be loaded and processed in a chunk at the same time. In case of memory issues this value should be decreased (default: 3000)
- *-v, --verbose* [x] : for values X>0 show the scripts output information. (default: 0)
- *-h, --help* : output usage information

4.7.1. ***annotate_protonated*** **Results**

It creates inside the 'count_tables/clean' folder:

- Two CSV files with the clean counts of spectra and peak area concatenated with the annotations result as new columns, named with the suffix '_ann.csv'

The 'molecular_networking' folder is created if not present yet, and inside it is created:

- The IVAMN edge file named as '<*output_name*>_ivamn.selfloops';
- One CSV table containing the IVAMN attributes and the assigned [M+H]+ representatives, named as '<*output_name*>_ivamn_attributes.csv'

Where the 'output_name' is extracted from the 'output_path';

4.7.2. ***annotate_protonated*** **Examples**

Fake example to execute the Annotation [M+H]+ command:

```
$ node np3_workflow.js annotate_protonated --metadata
  "/path/to/the/metadata/file/test_np3_metadata.csv"
  --output_path "/path/to/the/output/dir/test_np3/outs/test_np3" -i 5
```

4.7.3. ***annotate_protonated*** **Annotation Algorithm Details**

During the detection of a given metabolite in a MS experiment some ionization variants can occur, what adds more redundancy and ambiguity to the list of consensus spectra. To try to overcome this limitation of the technique a chemical annotation based on chemical rules and numerical equivalences is performed to detected some of the variations in the clean set of consensus spectra and creates the IVAMN with the results.

The ionization variants considered are adducts, neutral losses, multiple charge, multiple charge isotopic patterns, dimers/trimers, isotopes and in-source fragmentation. Adducts and dimers/trimers combined with neutral losses are also considered. Multiple charge isotopic pattern is considered looking at the MS2 and the MS1 list that do not have a consensus spectrum assigned to it, in the last case the detection is only signalize with a flag added to clean count tables.

The annotation algorithm considers every consensus spectrum as a putative [M+H]+ candidate and parses the spectra that are in concurrent chromatographic peaks (same retention time interval with a tolerance applied in the peak boundaries) in search of m/z shifts corresponding to known annotations (numerical rules with a tolerance), and with the use of additional chemical rules in some cases (see table below for more details). Only the consensus spectra that appears in at least one data collection batch in common can be annotated. This algorithm also computes the m/z error of the numerical rules (within the tolerance) and the retention time error (between the peak boundaries – retention time minimum and maximum) associated to each annotation. And two flags named 'multicharge_ion' and 'isotope_ion' are created. The 'multicharge_ion' flag signalize potential multiple charge ions without the need of the mono charge ion detection (which can be absent), it indicates if the ion has a multiple charge isotopic pattern (a m/z shift of 0.5 for double charge and of 0.33 for triple charge variants) detected in the MS1 or MS2 lists. The 'multicharge_ion' flag receives a value of 2 if the isotopic pattern and the mono charge variant were detected in the MS1 or MS2 list, 1 if the isotopic pattern was detected but the mono charge is absent, and 0 otherwise. The 'isotope_ion' flag signalize potential isotope ions, it receives a value of 1 if the respective spectrum was annotated as a isotope ion, and 0 otherwise.

The annotation algorithm guides its behavior using the provided rules table, where the user specifies in the correct format (described below) all ionization variants that should be detected. Except for in-source fragmentation and multiple charge isotopic patterns that are always considered. It's important to notice that the annotations are based on chemical evidence and numerical equivalence of the used rules, and thus they can be wrong in some cases or some true annotations that are not covered could be missing. A fine tune of the tolerance parameters (for precursor m/z, retention time deviation and similarity cutoff) and of the defined annotations in the rules table should be manually performed by the user to reduce false positive and false negative rates, according to the experimental conditions.

The NP^3 MS workflow default rules table of accepted ionization modifications can be found inside the 'rules' folder in the 'np3_modifications.csv' CSV file. It must contain the adducts, neutral losses, multiple charge, dimers/trimers and carbon isotopes variants to be considered and which adducts or dimers/trimers should also be considered in combination with a neutral loss of water or ammonia. A copy of this file should be used in order to select a subset of the default rules and/or to add new ones. These new created file must follow the rules format described in the two tables below and be passed to the workflow commands to be used in place of the default rules table.

The NP^3 MS workflow rules table must be defined following the format of the table below with the same column names, where X, Y and Z are values provided by the user:

Table 3: NP^3 MS workflow Rules Table Format

ion	mzdiff	charge	neutral_loss_h2o	neutral_loss_nh3	neutral_loss	sim_cutoff
[**Z**M+**X**]**Y**+	The m/z difference to the monoisotopic m/z ([M]+) after taking *Y* and *Z* into account	+Y	0 or 1, if 1 and the ion is an adduct or dimer/trimer variant also considerates it combined with a neutral loss of water (i.e., *X-H2O*), else nothing happen	0 or 1, if 1 and the ion is an adduct or dimer/trimer variant also considerates it combined with a neutral loss of ammonia (i.e., *X-NH3*), else nothing happen	0 or 1, if the ion modification is a neutral loss set it as 1, otherwise as 0 (if it is an adduct or dimer/trimer combined with a neutral loss set as 0)	A numeric ranging from 0 to 1. The similarity cut-off value to accept the annotation, for adducts, neutral losses and isotopes. Multiple charge variants do not rely on the similarity value. The spectra must have a similarity value greater or equal to this value to be annotated
[**Z**M-**X**]**Y**-	The m/z difference to the the monoisotopic m/z ([M]-) after taking *Y* and *Z* into account	-Y	0 or 1	0 or 1	0 or 1	A numeric ranging from 0 to 1

The 'ion' column will be used to label the detected annotations, spaces are removed and ignored. *X* must be a string with the ionization modification to be considered (e.g. for adducts Na, NH4, 2K-H or for neutral losses H-H2O, H-NH3, H-H2O-NH3). *Y* is the ion charge, it is used to define double or triple charge variants and when *Y=1* it must be omitted in the 'ion' column. *Z* is the number of repetitions of a monomer, used to define dimers or trimers variants, otherwise it must be omitted in the 'ion' column. The mzdiff column is defined in the same way as the 'Mass' column present in the Fiehn Lab Mass Spectrometry Adduct Calculator [12] and the numerical rules follow their standard. The difference in the NP^3 MS workflow annotations is that the calculations are done assuming that the current spectra being annotated is a [M+H]+ or a [M-H]- ion, and so the standard ionization mode is hard coded to be subtracted from the spectra m/z, but the rules definition in the table are the same.

The following table describes the chemical and numerical rules used by the annotation algorithm to detect each type of possible ionization variant and the correct format of their annotation label (to be placed in the 'ion' column).

Where:

- M : is the m/z of the current spectrum being annotated (considered as a [M+H]+ or a [M-H]- ion)

- m : is the m/z of any other spectra that appears in a concurrent retention time interval and in at least one data collection batch in common with the current spectrum M
- ion_mode : is the ionization mode informed by the user (+1 for [M+H]+ and -1 for [M-H]-) which is converted to the hydrogen mass equals 1.00783 or -1.00783 Da
- mz_tol : is the precursor m/z tolerance in Daltons informed by the user (default 0.025)
- fragment_tol : is the MS2 fragmented peaks m/z tolerance in Daltons informed by the user (default 0.025)
- m_H2O : is the mass of the water molecule equals 18.01056 Da
- m_NH3 : is the mass of the ammonia molecule equals 17.03052 Da
- m_iso : is the mass difference between the carbon isotopes 13C - 12C equals 1.0033 Da
- abs(**expression**) : is the function that returns the absolute value of the expression inside the parentheses, e.g., the positive value

Table 4: NP^3 MS workflow Chemical and Numerical Rules for Ionization Variants Annotation

Variant Type	Annotation Format	Rules	Description
Adduct	[M+X]+ or [M+X-H2O]+ or [M+X-NH3]+	abs(M-ion_mode-m-m_X) <= mz_tol OR abs(M-ion_mode-m-m_X-m_H2O) <= mz_tol OR abs(M-ion_mode-m-m_X-m_NH3) <= mz_tol AND the spectra have a similarity value greater or equal than the 'sim_cutoff' value of the respective annotation	m_X is the mass of the adduct X
Multiple Charge Isotopic Pattern	[M+X]Y+ isotopic	abs((M-ion_mode) - m) - 1/Y <= fragment_tol AND the peak area of the greater m/z is less than 2/3 of the other spectra peak area	Y is the charge of the multiple charge variant, only Y = 2 or 3 are considered. Multiple charge variants do not rely on the similarity value; This annotation is always considered
Multiple Charge	[M+X]Y+	((M-ion_mode)/Y + m_X) - m <= mz_tol AND the presence of the isotopic pattern where an m/z equals m+1/Y must be present in the count tables (the not fragmented MS1 peaks count table is also checked)	m_X is the mass of the adduct X and Y is the charge of the multiple charge variant, only Y = 2 or 3 are considered. Multiple charge variants do not rely on the similarity value.

Variant Type	Annotation Format	Rules	Description
Dimer or Trimer	[Z*M+X]+ or [Z*M+X-H2O]+ or [Z*M+X-NH3]+	(Z*(M-ion_mode) + m_X) - m <= mz_tol AND the monomer must not have fragmented peaks greater than the precursor m/z plus a gap for neutral losses AND one of the following is true (1) the dimer or trimer do not have fragmented peaks between the m/z of the monomer and its precursor m/z plus a gap for neutral losses and it has no fragmented peaks greater than its precursor m/z plus a gap for neutral losses OR (2) the candidates have a spectra similarity greater or equal than the 'sim_cutoff' value of the respective annotation	m_X is the mass of the adduct X (less m_h2o or m_nh3 when X-H2O or X-NH3) and Z is 2 if it is a dimer and 3 if it is a trimer; this annotation is only considered if no multiple charge variant was detected between spectra M and m
In-source Fragmentation	fragment	m appears in the list of fragmented peaks of M, within a tolerance equals 'fragment_tol' AND m < M - 2*iso_mass AND with an intensity greater than a MS2 baseline cut-off (15 is used by default) AND the spectra have a trimmed similarity value greater or equal than 0.2	the fragmented peaks intensity are normalized from 0 to 1000 and then scaled. The scale is also applied to the baseline cut-off before using it; A trimmed similarity is computed by first trimming the fragmented peaks of both spectra using the smaller precursor m/z less 2*m_iso and then computing their cosine similarity, the maximum similarity value between the shifted cosine and the trimmed cosine is used; This annotation is always considered
Neutral Loss	[M+H-X]+	abs(M-m-m_X) <= mz_tol AND the spectra have a similarity value greater or equal than the 'sim_cutoff' value of the respective annotation	m_X is the mass of the neutral loss X, e.g. if X=H2O then m_X = m_h2o. The 'neutral_loss' column must be set to 1

Variant Type	Annotation Format	Rules	Description
Isotope	[M+X]+	(M-m) - m_iso * X <= mz_tol AND the spectra have a similarity value greater or equal than the 'sim_cutoff' value of the respective annotation	X is an integer representing the carbon isotope deviation to be considered in the precursor m/z, e.g. X=1 for C13-C12 (m_iso difference) and X=2 for C14-C12 (2*m_iso difference). We recommend setting X <= 2 and a high 'sim_cutoff'

Only the rules for the positive ion mode are present in the above table, but the equivalent rules are also applied when the negative ion mode is selected but they were not vastly tested for this ion mode and thus could lead to undesired results.

The annotations of ionization variants detected for each msclusterID (row) being considered as a [M+H]+ ion is stored in the count tables in 6 new columns, one for each type of annotation, except for neutral losses that are grouped with adducts, named as: 'adducts', 'isotopes', 'dimers', 'multiCharges', 'fragments' and 'analogs.' Where 'analogs' annotate the concurrent spectra that have a similarity value greater or equal than 0.7 and did not receive any annotation from the spectra of the respective row.

The annotations are stored in those columns in the following format:

<ann> (sim <sim_value> - mzE <mz_error> - rtE <rt_error>)[<variantID>]{<#samples>}

Where,

ann is the detected annotation, following the format present in Table 4

sim_value is the spectra similarity value. If the annotation is 'fragment' the similarity value is the trimmed spectra similarity.

mz_error is the m/z error related with the numerical rule of the respective annotation

rt_error is the retention time error between the spectra peak boundaries

variantID is the variant spectra msclusterID in the clean count table

#samples is the number of common samples between the spectra, the number of samples that both spectra appear

If more than one annotation of a same type is present for a given spectra, they are concatenated using a ';' character.

4.7.4. *annotate_protonated* Ionization Variant Annotation Molecular Network (IVAMN) and Protonated Assignment Algorithm Details After the annotation algorithm the ionization variant annotation molecular network (IVAMN) is created based on the detected chemical annotations. The nodes of this network are the clean consensus spectra and the links of this network connects two consensus spectra that have a chemical annotation. The links have a direction, pointing from the consensus spectra considered as an ion variant to the consensus spectra considered as a putative [M+H]+ candidate in the respective annotation (e.g., [M+Na]+ -> [M+H]+).

The IVAMN .selfloop file have the following columns, comma separated:

- "msclusterID_source" : the number of the msclusterID of the source node (ionization variant)

- "msclusterID_target" : the number of the msclusterID of the target node (considered [M+H]+ ion)
- "cosine" : the similarity value between the source and the target node
- "annotation" : the annotation detected between the source and the target node
- "mzError" : the m/z numerical rule error associated with the detected annotation
- "rtError" : the retention time error between the peak boundaries of the source and target nodes
- "numCommonSamples" : the number of samples that both the source and the target node appear
- "componentIndex" : the component index of the source and target nodes

Finally, an algorithm to assign some of the clean consensus spectra as the most likely [M+H]+ (protonated) candidates is applied in the IVAMN. This algorithm performs a link analysis in IVAMN to assign some nodes as representative [M+H]+ in each component (set of interconnected nodes) of this network. It works as follows:

1. First it runs the PageRank [13], a link analysis algorithm;
2. Then, for each component of IVAMN it selects the node with the highest score in the PageRank algorithm as a putative [M+H]+ representant.
    - The nodes signalized as possible multiple charge ions are excluded from this selection, as well as ions annotated as neutral losses of a multiple charge node.
    - In the case of a tie in the PageRank score between nodes that are in a same cycle (e.g., that have links pointing to each other), the node pointed by the link of the annotation with the lowest m/z error is used to select the putative [M+H]+ representant of the current iteration. If the tie is between nodes that are not in a same cycle, the node with the lowest ID is selected first;
3. Next it removes all the nodes ancestors to the node selected as representant (e.g., all nodes that have a path to this node) from the current component
4. Repeats steps 2. and 3. until all nodes of the current component are removed (have a putative [M+H]+ representant that covers it).

A validation metric is computed for each node assigned as a representative [M+H]+ candidate, equals the sum of the numerical rules' errors of the annotations of all its ancestors. The result of this algorithm is a list of consensus spectra that represent the most likely [M+H]+ candidates of the collection, and thus, can be used as a suggestion of the number of real metabolites present in the samples. Some ambiguities may occur in the assignment of the putative [M+H]+ representants in the components of the molecular network of annotations, due mostly to possible missing/mistaken annotations and the PageRank solution, and a more robust solution to this limitation will be addressed in the next version of this workflow.

The following columns are added to the count tables in the *annotate_protonated* step:

Table 5: Clean and Annotated Count Tables New Columns from the Protonated Assignment Step

Columns	Description	Value Type
protonated_representative	1 if the respective spectrum was assigned as a [M+H]+ representative, 0 otherwise	numeric
protonated_mzError_sum	if protonated_representative is 1, equals the sum of the numerical rules' errors of the annotations of all its ancestors in the molecular networking of annotations, NA otherwise	numeric
protonated_rtError_sum	if protonated_representative is 1, equals the sum of the retention times errors of the annotations of all its ancestors in the molecular networking of annotations, NA otherwise	numeric

An attributes table is also created for the molecular network of annotations containing the nodes information (m/z and retention times), the 'multicharge_ion' flag, the columns of Table 5 and the following columns:

- "in_degree" : the number of incoming connections of the node
- "total_degree" : the number of incoming and outgoing connections of the node
- "pagerank" : the PageRank algorithm scores. It computes a ranking of the nodes in the network based on the structure of the incoming links
- "number_ancestors" : the number of ancestor nodes of the current node
- "protonated_num_ancestors_edges" : the number of connections between all the ancestor's nodes of the current node

4.8. Step 8: Command *merge*

Runs the merge of the clean count tables based on the annotated variants (for positive ion mode only). It creates new symbolic spectra candidates representing the union of each spectra with its annotated variants. This union is performed for each type of annotation (adducts + neutral losses, multiple charges, dimers/trimers, isotopes and in-source fragments) and by combining all of them together, what can lead to at most 31 new symbolic spectra by consensus spectra.

By default, the merge is only performed for the consensus spectra assigned as a [M+H]+ representative, to better account for the quantifications of the true metabolites.

Options for the command *merge*:

- *-o, --output_path* <path> : path to the output data folder, inside the 'outs' directory of the clustering result folder. It should contain the 'counts_table' folder and inside it the 'clean' subfolder with the clean count tables in CSV files. The job name will be extracted from here
- *-y, --processed_data_dir* <path> : the path to the folder inside the raw data folder where the pre-processed data (MGFs) were stored.
- *-m, --metadata* <file> : path to the metadata table CSV file
- *-p --merge_protonated* [x] : A boolean TRUE or FALSE indicating if only the [M+H]+ representative consensus spectra should be merged. If FALSE merge all msclusterID's (default: "TRUE")
- *-e, --method* [name] : a character string indicating which correlation coefficient is to be computed. One of "pearson", "kendall", or "spearman" (default: spearman)
- *-v, --verbose* [x] : for values X>0 show the scripts output information. (default: 0)
- *-h, --help* : output usage information

4.8.1. *merge* Results

One subfolder inside the 'count_tables' folder is created named 'merge' containing:

- Two CSV files with the cleaned and annotated counts of spectra and peak area merged and new symbolic clusters added as new rows, named with the suffix '_merged_ann.csv';
- CSV files with the correlation columns added are also included when there is a biocorrelation result.

4.8.2. *merge* Examples

Fake example to execute the Merging command:

```
$ node np3_workflow.js merge --output_path "/path/to/the/output/dir/test_np3/outs/test_np3"
```

4.8.3. *merge* Details

The merge step extends the clean count tables with new rows containing symbolic clusters resulting from the aggregation by row of each spectra with its annotated variants. It contains all the clean results plus new symbolic spectra created from the merged annotations, with a quantification that could be more representative of a true metabolite (merged count tables). By default, the merge is only performed for the consensus spectra assigned as a [M+H]+ representative.

The merge algorithm is performed by consensus spectra, following these steps: first all annotated isotopes variants are merged and results in new symbolic spectra, with their counts aggregated as in Step 4 (summing the number of spectra and summing the unique peak areas); then all annotated adducts and neutral losses are merged, the new symbolic spectra from the previous merge (isotopes) are also considered; next the same is done for dimers/trimers, multiple charge and finally for fragments. The merge can result in too many new symbolic clusters, for a single spectrum the maximum is 31 new symbolic clusters. We recommend to use the correlation score to filter the relevant new symbolic clusters and to set the parameter merge_protonated to TRUE to only apply the merge algorithm to a subset of the consensus spectra. The merge is not applied to spectra that appears in blank samples.

It's up to the user to identify relevant symbolic clusters. Since the annotations are based on evidences of the considered chemical rules, some symbolic clusters may include the aggregation of wrong annotations and must be evaluated carefully.

The 'peakLists' and 'peakInts' columns are also recomputed aggregating the merged spectra fragmented peaks lists similar to how it is done in the clean Step 5. The new symbolic spectra are not exported to a MGF file because their lists of fragmented peaks could be very noisy and were not vastly tested. In the next releases of the workflow, we plan to develop a more sophisticated heuristic to perform the merge and to provide a more relevant and easier to use result. Still, the new symbolic spectra can improve the biocorrelation score of metabolites that were detected as different types of ionization variants.

The following columns are added to the count tables in the *merge* step:

Table 6: Merge Count Tables New Columns

Columns	Description	Value Type
mergedIDs_all	the msclusterIDs of all merged spectra concatenated by a ';'. The first msclusterID of this list is from the spectrum that received the merge	character
mergedIDs_adducts	the msclusterIDs of all adducts and neutral losses variants merged concatenated by a ';'	character
precursorMz_adducts	the mzConsensus of all adducts and neutral losses variants merged concatenated by a ';'	character
mergedIDs_dimers	the msclusterIDs of all dimers/trimers variants merged concatenated by a ';'	character
precursorMz_dimers	the mzConsensus of all dimers/trimers variants merged concatenated by a ';'	character
mergedIDs_multiCharges	the msclusterIDs of all multiple charge and multiple charge isotopic variants merged concatenated by a ';'	character
precursorMz_multiCharges	the mzConsensus of all multiple charge and multiple charge isotopic variants merged concatenated by a ';'	character
mergedIDs_fragments	the msclusterIDs of all fragments variants merged concatenated by a ';'	character
precursorMz_fragments	the mzConsensus of all fragments variants merged concatenated by a ';'	character

4.9. Step 9: Command *corr*

Runs the R script to compute for each consensus spectra the biocorrelation score between its count of spectra or peak area by sample and the samples bioactivity score. The biocorrelation is performed for each correlation group defined in the metadata table (columns named with the prefix "BIOACTIVITY_" and "COR_"). The correlation can be performed with any of the following methods: 'pearson', 'kendall' or 'spearman' (default).

The bioactivity score can be used to rank the consensus spectra and to select the most possible candidates responsible for the observed hits in bioactivity experiments. This information can also be visualized in the molecular networks and applied for the discovery of active compounds, as done in [15].

Before the biocorrelation computations, this command computes the grouping of the quantification defined in the metadata table (column named with prefix "GR_"). These groups may be used to aggregate any characteristics of the samples.

Options for the command *corr*:

- *-b, --metadata* <file> : path to the metadata table CSV file. Used to retrieve the biocorrelation groups and quantification grouping. For a joined job, this must be the original samples metadata table.
- *-c, --count_file_path* <file> : path to the count table CSV file
- *-e, --method* [name] : a character string indicating which correlation coefficient is to be computed. One of "pearson", "kendall", or "spearman" (default: spearman)
- *-b, --bio_cutoff* [name] : a bioactivity cutoff value. Bioactivities in the metadata table that are less than the bio_cutoff value will be set to zero. (default: 0)
- *-v, --verbose* [x] : for values X>0 show the scripts output information (default: 0)
- *-h, --help* : output usage information

4.9.1. *corr* Results A CSV table in the count_file directory named as '<count_file_name>_corr_<method>.csv', where the count_file_name is the name of the provided count_file table. The new tables contain the original count tables data plus new columns with the correlation scores of each consensus spectrum for each correlation group and bioactivity score, as defined in the metadata table. It may also contain new columns for the quantification grouping, if defined in the metadata.

Another table is also created as a copy of the first table, named with the suffix 'bioAct' as '<count_file_name>_corr_<method>_bioAct.csv', containing new rows at the beginning of the table with the bioactivities scores of each sample placed above the original counts table header (first rows).

In the metadata table the columns with a bioactivity score must be named with the prefix "BIOACTIVITY_" and the columns with the correlation groups (samples to be used in each correlation) must be named with the prefix "COR_". The correlation score will produce NA values with warnings if the counts of the selected samples are all equal 0 or if the bioactivity of the selected samples are all the same, and it will produce "CTE" if the counts of the selected samples have the same values (standard deviation equals 0).

4.9.2. *corr* Examples Fake example to execute the Biocorrelation command:

```
$ node np3_workflow.js corr --metadata "/path/to/the/metadata/file/test_np3_metadata.csv"
  --count_file "/path/to/the/metadata/file/np3_job_spectra.csv"
```

4.10. Step 10: Command *mn* (Molecular Networking)

Creates a spectra similarity molecular network (SSMN) which connects spectra based on the pairwise spectra similarity value above the given similarity cut-off. Then, a filter is applied on this network to remove links between spectra that have less peaks in common than the minimum number of matched peaks, to limit the number of neighbors of each node (number of links) to the top K most similar ones and to limit the size of the components to a maximum number of nodes. The final filtered SSMN contains components that represent the most analogous spectra, possible connecting spectra from similar chemical classes. At the end, a [M+H]+ analysis is executed if IVAMN is present, and results in the protonated IVAMN and protonated SSMN filtered.

The nodes of the SSMN are the set of clean consensus spectra and the links of this network connects the spectra that have a similarity value above the *similarity_mn* cutoff and that passed the filters. The annotations from Step 7 are also included in the SSMN as labels when present.

Options for the command *mn*:

- *-o, --output_path* <path> : path to the output data folder, inside the outs directory of the clustering result folder. It should contain the 'molecular_networking' folder and inside it the 'similarity_tables' folder. The job name will be extracted from here
- *-w, --similarity_mn* [x] : the minimum similarity score that must occur between a pair of consensus MS/MS spectra in order to create an edge in the molecular networking. Lower values will increase the component size of the clusters by inducing the connection of less related MS/MS spectra; and higher values will limit the components sizes to the opposite (default: 0.6)

- *--min_matched_peaks* [x] : The minimum number of common peaks that two spectra must share to be connected by an edge in the filtered SSMN. Connections between spectra with less common peaks than this cutoff will be removed when filtering the SSMN. Except for when one of the spectra have a number of fragment peaks smaller than the given min_matched_peaks value, in this case the spectra must share at least 2 peaks. The fragment peaks count is performed after the spectra are normalized and cleaned. (default: 6)
- *-k, --net_top_k* [x] : the maximum number of connections for one single node in the similarity molecular networking. An edge between two nodes is kept only if both nodes are within each other's [x] most similar nodes. This restriction is applied to the spectra following the msclusterID's order, so smaller m/z's are limited first. Keeping this value low makes very large networks (many nodes) much easier to visualize (default: 10)
- *-x, --max_component_size* [x] : the maximum number of nodes that all component of the similarity molecular network must have. The edges of this network will be removed using an increasing cosine threshold until each network component has at most X nodes. Keeping this value low makes very large networks (many nodes and edges) much easier to visualize. (default: 200)
- *--blank_expansion* [x] : the distance of neighborhood nodes from the blanks in IVAMN to be selected for removal in the final protonated networks. (0) to only remove blanks nodes, (1) to remove nodes directly connected to a blank node, (2 or greater) to remove nodes in a distance equal to 2 or greater from a blank node, or (-1) to remove all possible neighbours and ancestors of a blank node (remove blank clusters) from IVAMN (default: 0)
- *-b, --max_chunk_spectra* [name] : Maximum number of spectra (rows) to be loaded and processed in a chunk at the same time. In case of memory issues this value should be decreased (default: 3000)
- *-v, --verbose* [x] : for values X>0 show the scripts output information. (default: 0)
- *-h, --help* : output usage information

4.10.1. *mn* Results Four files are created inside the 'molecular_networking' folder with the SSMN, the filtered SSMN, the protonated IVAMN and the protonated SSMN filtered:

- The complete spectra similarity molecular network named as :
  '<*output_name*>_ssmn_w_<*similarity_mn*>.selfloops', which contains all links with a similarity value above the cut-off
- The filtered spectra similarity molecular network named as
  '<*output_name*>_ssmn_w_<*similarity_mn*>_k_<*net_top_k*>_x_
  <*max_component_size*>.selfloops'
- The IVAMN [M+H]+ named as:
  '<*output_name*>_ivamn_protonated.selfloops'
- The SSMN [M+H]+ filtered named as:
  '<*output_name*>_ssmn_protonated_w_<*similarity_mn*>_k_<*net_top_k*>_x_
  <*max_component_size*>.selfloops'

Where the 'output_name' is extracted from the 'output_path';

4.10.2. *mn* Examples Fake example to execute the Molecular Network of spectra similarity command:

```
$ node np3_workflow.js mn --output_path "/path/to/the/output/dir/test_np3/outs/test_np3"
  --similarity_mn 0.7 --verbose 1
```

4.10.3. *mn* Details The molecular networks are a great way to visualize the table counts results. The resulting .selfloops edges files can be opened in graph visualization tools, such as Cytoscape [19] and Gephi [20]. Then the count tables can also be loaded to the networks to add the other nodes information present in these tables, by matching their msclusterIDs.

The creation of the spectra similarity molecular network (SSMN) is based on the pairwise similarity comparison of the clean consensus spectra using the shifted cosine similarity score or spec2vec scores (see Step 5 details), ranging from 0 (totally dissimilar) to 1 (completely identical). The nodes of the SSMN are the clean consensus spectra and the links of this network connect two consensus spectra that have a similarity score greater or equal to the provided cosine similarity cut-off (default to 0.6 is used).

The SSMN is further filtered by removing the links between spectra with a number of commom peaks that do not respect the minimum number of matched peaks parameter cut-off (default to at least 6 matched peaks), then limiting the number of links of each node to the top K most similar neighbors (default K equals 15 was used) and pruning the remaining links with an increasing cosine similarity cut-off to limit the components sizes to a provided maximum number of nodes (default of 200 nodes is used). These filtering was based in the GNPS molecular networking analysis code [8].

If the resulting molecular network of similarity is too dense (many nodes and links forming a giant single component), the user can try to restrict the filters with a bigger similarity and minimum number of common peaks cut-offs (similarity_mn and min_matched_peaks parameters) and a smaller top K neighbors and component size cut-offs (net_top_k and max_component_size parameters).

The SSMN .selfloop files have the following columns, comma separated:

- “msclusterID_source” : the number of the msclusterID of the source node spectrum
- “msclusterID_target” : the number of the msclusterID of the target node spectrum
- “cosine” : the similarity value between the source and the target node
- “num_matched_peaks”: the number of common peaks between the source and target node spectra
- “annotation” : the existing annotation between the source and the target node (this can be missing for some node pairs)
- “num_peaks_source”: the number of fragment peaks in the source node spectrum after normalization and cleaning
- “num_peaks_target”: the number of fragment peaks in the target node spectrum after normalization and cleaning
- “componentIndex” : the component index of the source and target nodes

The SSMN is an undirected graph. The source and the target nodes could be shifted, their order do not matter but the source node has always a smaller msclusterID than the target node due to the implemented algorithm.

At the end, a [M+H]+ analysis is executed if IVAMN is present, and results in the protonated networks. It creates de SSMN [M+H]+ filtered and the IVAMN [M+H]+ networks without blanks. It uses as input the complete SSMN and the IVAMN networks, the IVAMN attribute table (protonated information) and the clean table. It works as follow:

1. In IVAMN the blank nodes are removed together with its neighbours and ancestors (or successors, ignores the links direction) if argument blank_expansion != 0, else only remove the blank nodes;
2. Then the [M+H]+ are selected in the IVAMN with no blanks, resulting in the IVAMN [M+H]+ network;
3. The nodes from IVAMN [M+H]+ are used to select the final nodes from the SSMN, resulting in the SSMN [M+H]+ network;
4. Finally, the SSMN [M+H]+ is filtered using the min_matched_peaks, top_k and max_component size arguments, resulting in the SSMN [M+H]+ filtered.

4.11. Command ***join_jobs***

Command to join NP^3 jobs (results from the *run* or the *join_jobs* commands) into a single united job. Concatenate different jobs without the need of running them all together again. Uses a different metadata, called *metadata_join*, defining the jobs to be joined and their reference codes, the names of their original metadata and pre processing directory. It uses the clean results from the provided NP^3 jobs and execute the main pipeline from Steps 3 to 10 with some modifications and adaptations, except for Step 8 which is skipped.

The following subsections define how to create the metadata_join table, its expected format, the data organization to execute the *join_jobs* command and details on its methods.

The *join_jobs* command may be useful for processing growing libraries, which will have new datasets being included from time to time; or for processing very large jobs, which may be divided into smaller jobs and then joined by chunks with a smaller memory footprint (divide and conquer strategy).

This command performs the clustering of all the clean data from the provided jobs together, then proceed the workflow to the clean step, compare the final joined consensus spectra pairwise, next update the original ionization annotations with the final joined clean consensus spectra and merge the original IVAMNS, recompute the [M+H]+ and keep some of the original protonated representatives, compute the biocorrelation and groupings using the original samples metadata with all original jobs together and finally compute the molecular networking with the joined result.

The *join_jobs* can be used to join the results from multiple original jobs and also from previous joined jobs with a new original or joined job. There is no limit for joining NP^3 results, coming from the *run* or the *join_job* commands. The user must only ensure that all sample's code are unique across the different original jobs being joined (SAMPLE_CODE column from the original samples' metadata).

Options for the command *join_jobs*:

- *-n, –output_name* <name> : the job name. It will be used to name the output directory and the results from joining the jobs. It must have less than 80 characters.
- *-m, –metadata_join* <file> : path to the metadata table CSV file defining the jobs to be joined. Different format, see manual.
- *-d, –jobs_data_path* <path> : path to the folder containing the input jobs result to be joined, this should contain their previous NP3 result, named accordingly to what is specified in the metadata_join. Their clean mgf and quantification tables will be used.
- *-y, –pre_processed_dir_path* <path> : path to the folder containing the input jobs pre processing result, this should contain all the original jobs previous NP3 pre processing result in separated folders named accordingly to what is specified in the metadata_join.
- *-o, –output_path* <path> : path to where the output directory will be created
- *-f, –fragment_tolerance* [x] : the tolerance in Daltons for fragment peaks. Peaks in the original spectra that are closer than this get merged by the NP3_MSCluster algorithm. Also used in the pre process (Step 2) (default: 0.05)
- *-z, –mz_tolerance* [x] : this is the tolerance in Daltons for the m/z of the precursor that determines if two spectra will be compared and possibly joined. Used in the clustering job and in the library identifications (Step 6) (default: 0.025)
- *-p, –ppm_tolerance* [x] : the maximal tolerated m/z deviation in parts per million (ppm) to be used in the pre-processing step if ran (default: 5)
- *-t, –rt_tolerance* [x,y] : tolerances in seconds for the retention time width of the precursor that determines if two spectra will be compared and possibly joined. It is directly applied to the retention time minimum (subtracted) and maximum (added) of the spectra. (default: 2)
- *-a, –ion_mode* [x] : the precursor ion mode. One of the following numeric values corresponding to a ion adduct type: '1' = [M+H]+ or '2' = [M-H]- (default: 1)
- *-i, –similarity_function* [x] : the similarity function to be used in the spectra comparison to create the pairwise similarity table after clustering and clean steps. If "spec2vec" is selected, the model trained on UniqueInchikey subset (12,797 spectra) is used by spec2vec in the spectra comparison and the matchms library is used to compute the number of matched peaks between the compared spectra; otherwise, the NP3 shifted cosine function is used. One of "np3_shifted_cosine" or "spec2vec". (default: "np3_shifted_cosine")
- *-s, –similarity* [x] : the minimum similarity to be consider in the hierarchical clustering, starts in 0.70 and decrease to X in 15 rounds (default: 0.6)

- *-g, –similarity_blank* [x] : the minimum similarity to be consider in the hierarchical clustering of the blank clustering steps, starts in 0.70 and decrease to X in 15 rounds. Only used in the clustering of blank samples (column SAMPLE_TYPE equals 'blank' in the metadata table) (default: 0.3)
- *–bflag_cutoff* [x] : A positive numeric value to scale the interquartile range (IQR) of the blank spectra basePeakInt distribution from the clustering result and to allow spectra with a basePeakInt value below this distribution median plus IQR*bflag_cutoff to be joined with a blank spectrum during the clean Step 5, without relying on the similarity value. Or FALSE to disable it. The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile) of the distribution. The spectra with a basePeakInt value <= median + IQR*bflag_cutoff (from the blank spectra basePeakInt distribution) and BFLAG TRUE will be joined to a blank spectrum in the clean Step 5. This cutoff will affect the spectra with BFLAG TRUE that would not get joined to a blank spectra when relying only on the similarity cutoff. This is a turn around to the fact that blank spectra have low quality spectra and thus can not fully rely on the similarity values. (default: 1.5)
- *–noise_cutoff* [x] : A positive numeric value to scale the interquartile range (IQR) of the blank spectra basePeakInt distribution from the clustering Step 3 result and to remove the spectra with a basePeakInt value below this distribution median plus IQR*noise_cutoff after the clean Step 5. Or FALSE to disable it. The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile) of the distribution. When no blank sample is present in the metadata, the full distribution is used. This cutoff will affect the spectra with with a low basePeakInt value that probably are noise features. If the clustering Step 3 resulted in more than 25000 spectra, the noise cutoff will be applied before the clean Step 5 to prevent a long processing time (default: "FALSE")
- *-c, –scale_factor* [x] : the scaling method to be used in the fragmented peak's intensities before any dot product comparison (Step 3). Valid values are: 0 for the natural logarithm (ln) of the intensities; 1 for no scaling; and other values greater than zero for raising the fragment peaks intensities to the power of x (e.g. x = 0.5 is the square root scaling). [x] >= 0 (default: 0.5)
- *-e, –method* [name] : a character string indicating which correlation coefficient is to be computed. One of "pearson", "kendall", or "spearman" (default: "spearman")
- *-x, –min_peaks_output* [x] : the minimum number of fragment peaks that a spectrum must have to be outputted after the final clustering step. Spectra with less than x fragmented peaks will be discarded. x >= 1 (default: 5)
- *-j, –tremolo_identification* [x] : (not Windows OS's) A logical "TRUE" or "FALSE" indicating if the Tremolo tool should be used for the spectral matching against the ISDB from the UNPD (default: "TRUE")
- *-r, –trim_mz* [x] : A logical "TRUE" or "FALSE" indicating if the spectra fragmented peaks around the precursor m/z +-20 Da should be deleted before the pairwise comparisons. If "TRUE" this removes the residual precursor ion, which is frequently observed in MS/MS spectra acquired on qTOFs. (default: "TRUE")
- *–max_shift* [x] : Maximum difference between precursor m/zs that will be used in the search of shifted m/z fragment ions in the NP3 shifted cosine function. Shifts greater than this value will be ignored and not used in the cosine computation. It can be useful to deal with local modifications of the same compound. (default: 200)
- *-l, –parallel_cores* [x] : the number of cores to be used for parallel processing in Step 5 spectra comparison. x = 1 for disabling parallelization and x > 2 for enabling it. x >= 1 (default: 2)
- *-w, –similarity_mn* [x] : the minimum similarity score that must occur between a pair of consensus MS/MS spectra in order to create an edge in the molecular networking. Lower values will increase the component size of the clusters by inducing the connection of less related MS/MS spectra; and higher values will limit the components sizes to the opposite (default: 0.6)
- *-k, –net_top_k* [x] : the maximum number of connection for one single node in the similarity molecular

networking. An edge between two nodes is kept only if both nodes are within each other's [x] most similar nodes. Keeping this value low makes very large networks (many nodes) much easier to visualize (default: 15)

- *-x, –max_component_size* [x] : the maximum number of nodes that all component of the similarity molecular network must have. The edges of this network will be removed using an increasing cosine threshold until each network component has at most X nodes. Keeping this value low makes very large networks (many nodes and edges) much easier to visualize. (default: 200)
- *–min_matched_peaks* [x] : The minimum number of common peaks that two spectra must share to be connected by an edge in the filtered SSMN. Connections between spectra with less common peaks than this cutoff will be removed when filtering the SSMN. Except for when one of the spectra have a number of fragment peaks smaller than the given min_matched_peaks value, in this case the spectra must share at least 2 peaks. The fragment peaks count is performed after the spectra are normalized and cleaned. (default: 6)
- *–blank_expansion* [x] : the distance of neighborhood nodes from the blanks in IVAMN to be selected for removal in the final protonated networks. (0) to only remove blanks nodes,
(1) to remove nodes directly connected to a blank node, (2 or greater) to remove nodes in a distance equal to 2 or greater from a blank node, or (-1) to remove all possible neighbours and ancestors of a blank node (remove blank clusters) from IVAMN (default: 0)
- *-b, –max_chunk_spectra* [x] : Maximum number of spectra to be loaded and processed in a chunk at the same time. In case of memory issues this value should be decreased (default: 3000)
- *-v, –verbose* [x] : for values X>0 show the scripts output information (default: 0)
- *-h, –help* : display help for command

4.11.1. *join_jobs* Results

A directory inside the *output_path* named with the *output_name* containing:

- A copy of the *metadata_join* file and the command line parameters values used in a file named 'logRunParms', for reproducibility.
- Two automatically created metadata tables containing:
  - One the original samples concatenated in a single file named 'original_samples_METADATA.csv';
  - And another with the original NP^3 jobs that were joined in this process in a file named 'original_jobs_METADATA_JOIN.csv'. For reproducibility and future joins with this result.
- A folder named 'outs' with the clustering result of the joined jobs in a single sub folder named with the *output_name* containing:
  - A sub folder named 'count_tables' with the Step 4 quantification in CSV tables named as '_(spectra|peak_area).csv'. And inside it the clean tables in a folder named 'clean'.
  - Another sub folder named 'clust' with the clusters membership files (which SCANS or msclusterID were joined).
  - A third sub folder named 'mgf' with the resulting clean consensus spectra in MGF files.
  - A fourth sub folder named 'identifications' with the tremolo identification results in a csv table.
  - A fifth sub folder named 'molecular_networking' with the molecular networking of this joined job, both SSMN and IVAMN.
  - A text file named 'logNP3MSClusterOutput' with the NP3_MSCluster log output.

4.11.2. *join_jobs* Examples

Join original jobs A and B.

```
$ node np3_workflow.js join_jobs --output_name "test_join_a_b" --output_path
"/path/where/the/output/will/be/stored" --metadata_join
"/path/to/the/metadata_join/file/test_np3_join_a_b_metadata.csv"
--pre_processed_dir_path "/path/to/the/dir/with/joining/jobs/pre/process/results"
--jobs_data_path "/path/where/the/joining/jobs/output/is/stored"
```

Join the joined job from A and B with original job C.

```
  $ node np3_workflow.js join_jobs -n "test_join_ab_c" -o
"/path/where/the/output/will/be/stored" -m
"/path/to/the/metadata_join/file/test_np3_join_ab_c_metadata.csv" -y
"/path/to/the/dir/with/joining/jobs/pre/process/results" -d
"/path/where/the/joining/jobs/output/is/stored" -t 3.5,5 -v 10
```

4.11.3. *join_jobs* Metadata_join creation

The *metadata_join* table defines the list of NP^3 jobs to be united, it defines the joining jobs reference names, original samples' metadata name and original pre processing result. It also defines if the joining jobs came from a *run* (called here original job) or a *join_jobs* (called here joined job) command result. This table has the same format as the original samples' metadata, it must be a CSV file with columns separated by comma ',' and with the dot '.' as decimal point character. It can be created and edited in EXCEL, LibreOffice CALC or any other spreadsheet program, but must be saved in CSV (UTF-8) format. If any text name in the metadata_join file contains special characters, the file must be saved with the parameter to quote text cells set to TRUE. After the metadata_join file was finished it is recommended to open it with a simple text viewer, e.g., notepad, to make sure that the column separator character and the decimal point character were properly set.

A metadata_join template file is available in the NP^3 MS workflow repository, named 'METADATA_JOIN_TEMPLATE.csv'. It uses the Bra346 and the MA9 datasets as examples, provided in the NP^3 paper.

The metadata_join table must contain the following mandatory columns (in uppercase):

- JOBNAME - the name of the job to be joined. This name must be equal to the *output_name* used to obtain this job result, it must match the job final results naming. This name will be used to retrieve the job data to perform the joining.
- JOB_CODE - unique and syntactically valid (see below) name for each job. The use of the same JOB_CODE in more than one entry will generate an error. A syntactically valid name must start with a letter and consist of letters, numbers, and underscore characters. R reserved words (https://cran.r-project.org/doc/manuals/r-release/R-lang.html#Reserved-words) are not syntactically valid names. The JOB_CODE name will be used to reference the original NP^3 jobs and to keep the joined msclusterIDs track.
- METADATA_NAME - the name of the original samples' metadata table used to execute the job, if this job is a result of the *run* command (JOINED_JOB column equals 0). This must be the complete file name with extension of the samples metadata, it will be used to automatically retrieve this data from the original job result. If this job is a result of the *join_jobs* command (JOINED_JOB column equals 1), this metadata name will not be used, instead the automatically created metadata with the original samples united will be used by default.
- PRE_PROCESSED_DATA_NAME - the name of the folder with the pre processing result of the original job. It will be used to retrieve the pre processed data of the job. If this is a joined job (JOINED_JOB column equals 1), this column is skipped.
- JOINED_JOB - A 0 or 1 column value to define if this job is the result of a original job coming from the *run* command execution (0) or if it is the result of a joined job coming from another join_jobs command execution (1).

The user is free to add any additional column to the metadata_join file, for example to add relevant descriptions for each job. These additional columns will be ignored by the workflow commands, but can be useful to add information related to the jobs being joined. It is important that additional column names are not equal or a prefix of any of the mandatory or optional columns of the metadata_join file.

The samples' quantification groupings, correlations and bioactivities will be retrieved from the samples' metadata of the original jobs. These information is not defined here. The original samples metadata automatically created in the setup of this command may be used a posteriori to add new biocorrelations or quantification groupings to the joining results, and the command *corr* may be used to recompute them.

4.11.4. *join_jobs* Data Organization The data organization to execute the *join_jobs* command must contain three things (defined by mandatory parameters):

1. A metadata_join table defining the jobs to be united and following the format and mandatory columns defined in the previous section (parameter *metadata_join*).
2. A folder containing the results of the NP^3 jobs to be united, these may be results from a *run* or a previous *join_jobs* execution, in the last case all the original NP^3 jobs must also be included in this folder. All jobs defined in the metadata_join must be placed inside this folder. The data to be united will be retrieved from here (parameter *jobs_data_path*).
3. And a folder containing the results of the pre processing of the original NP^3 jobs to be united or united by any included joined job. These data will be used to correctly compute the peak area of the final joined clean consensus spectra (paramater *pre_processed_dir_path*).

In a project intended to continue grow with new jobs being collected and processed, this structure must be kept to continuous join a new processed dataset.

4.11.5. *join_jobs* Details The join_jobs command uses the main NP^3 MS Workflow steps (from command *run*) to unite results coming from different NP^3 results. Some steps needed to be adapted (Steps 4 and 5) or replaced (step 7) to allow joining original NP^3 results.

The inputs to the join_jobs command are the pre processed data and the clean data from other NP^3 results, coming from a *run* execution (called original job) or from a *join_jobs* execution (called joined job). The clean data used is the clean MGF file, the clean count tables and the annotations of ionization variants. The ionization variants are not recomputed here because they are directly related to the original samples which were collected together, and instead, the original annotations are maintained from the original jobs, joined and updated here.

The join_jobs command starts with a setup process to retrieve the metadata information of the original NP^3 jobs being joined and to create two default metadatas: the first is the original samples metadata, named "original_samples_METADATA.csv", which concatenates the original samples of all original jobs being joined. The format of the samples metadata follow the format of the original metadata presented in section 4.1 which is Step 1 of the NP^3 workflow. It uses the column METADATA_NAME from the metadata_join to retrieve all the samples metadata from the original jobs. And the other is the original joined jobs metadata, named "original_jobs_METADATA_JOIN.csv", which concatenates the list of all original NP^3 jobs being joined. Its format follows the metadata_join format. If there is a joined job in the current join_jobs execution (JOINED_JOB == 1 in the provided metadata_join), its original joined jobs metadata is retrieved and concatenated with the list of new original jobs or other joined jobs metadata in the metadata_join. The same happens with the original samples metadata when a joined job is present. With this two metadatas, the reference to the original jobs is automatically kept throughout different executions of the join_jobs commands, even when the executions recursively call the join of previous joined jobs. With the original samples metadata created, a check is performed to guarantee that all samples' code are unique among the jobs being joined (column SAMPLE_CODE must be unique here), otherwise the join will fail and the user must correct this to proceed.

The join_jobs processing workflow starts with the clustering (Step 3) of the clean MGF of all jobs being joined. The clustering step reduces all the clean MGF of the provided jobs to a single clustered MGF and the command proceed to the quantification by original samples (Step 4) of the clustered consensus spectra that were joined. The quantification step was adapted here to use the clean tables from the jobs being joined and merge their counts for the clustered spectra. It uses as reference for the original jobs the JOB_CODE, present in the original joined jobs metadata, and concatenates it with the original msclusterID of each clean consensus spectra present in the clean counts and store this information in a new column named "joinedJobsIDs". For an already joined job being joined again, this column is kept for the reference. The quantification performs the count by number of spectra as a simple sum of the clean consensus spectra that were joined in the clustering step and performs the count by peak area using the pre processed data of all original samples being joined. It uses the "scans" column to retrieve the reference to the original peaks information and the original samples metadata to retrieve all the original SAMPLES_CODEs present in the pre-processed result, and then, correctly compute the peak area of the joined clustered consensus spectra. The peak area computation was adapted to allow retrieving the peaks information from multiple pre processed data, the PRE_PROCESSED_DATA_NAME from the metadata_join is used here for reference.

Next, the join_jobs workflow proceeds to the cleaning (Step 5), which is executed similar to the main workflow. The only exception is the peak area computation which was adapted for multiple pre processed data. Then, the clean joined data are compared pairwise and the joined clean MGF is identified against UNPD by tremolo (Step 6), and follow to the annotate_protonated (Step 7).

The annotate_protonated step was replaced here by a procedure to merge the ionization annotations of the original jobs being joined and then to recompute the [M+H]+ while keeping some of the original protonated representatives. The join_jobs flow of Step 7 is as follow: first it retrieves the IVAMNs of all the original jobs and map the msclusterIDs to the new joined ids, selfloops are removed and duplicated edges are merged. The edges attributes are recomputed with the new similarity values, the mzError receives the mean value of the merged edges and the rtError is recomputed again for the joined msclusterIDs. The annotations rules are applied again and removes any invalid annotation (similarity cutoff), any resulting empty edge is removed. This result in the final joined IVAMN, selfloops are added again at the end for the missing nodes. Then, the find protonated script is executed for the final joined IVAMN, resulting in a new list of [M+H]+ ions (column protonated_representatives = 1). This list is further merged with the original protonated_representatives that have an in-degree > 0 in the final joined IVAMN (keep [M+H]+ with valid annotations), this prevent loosing relevant information from the original jobs. Finally, a new script parses the information present in the final joined IVAMN and writes them as annotation columns to the joined clean tables, with a similar format to the main flow. The only missing column here is the 'analogs' one, which is not recomputed in the new procedure. The rest of the ionization annotation columns and the protonated_representative are written to the clean tables and to the joined IVAMN attribute table.

Next, the join_jobs workflow skips Step 8, the merging may result in too much additional data and may be executed separated by the user at the end of this procedure if needed. To execute the merge with the join_jobs result the user must inform the created original samples metadata as the metadata parameter. Then, the join_jobs command proceeds to the biocorrelation (Step 9), using the automatically created original samples metadata to perform the correlations and any grouping present in the original metadatas. At the end of this command, the user may add new correlations, bioactivities or quantification groupings to the original samples metadata and execute this step again (using command *corr*).

The join_jobs workflow ends with the molecular networking (Step 10) similar to the main flow. The data organization of the output is similar to the main flow and the additional steps may be executed the same way.

4.12. Command *gnps_result*

Join the GNPS library identification result from the Molecular Networking (download 'clustered spectra') or the Library Search (download 'all identifications') workflows to the count tables of the NP3 clustering or clean steps (Steps 3 or 5)

Options for the command *gnps_result*:

- *-i, --cluster_info_path* <path> : If joining the result of a Molecular Networking job, this should be the path to the file inside the folder named 'clusterinfo' of the downloaded output from GNPS. Not used for results coming from the Library Search workflow. (default: " ")
- *-s, --result_specnets_DB_path* <path> : If joining the result of a Molecular Networking job, this should be the path to the file inside the folder named 'result_specnets_DB'; and if this is the result of a Library Search workflow, this should be the path to the file inside the downloaded folder
- *-c, --count_file_path* <path> : Path to any of the count tables (peak_area or spectra) resulting from the NP^3 MS workflow clustering or clean steps. If the peak_area is informed and the spectra table file exists in the same path (or the opposite), it will merge the GNPS results to both files
- *-h, --help* : output usage information

4.12.1. ***gnps_result* Results**

The following columns with the GNPS results are added to the count tables: “gnps_SpectrumID”, “gnps_Adduct”, “gnps_Smiles”, “gnps_InChIKey”, “gnps_CAS_Number”, “gnps_Compound_Name”, “gnps_LibMZ”, “gnps_MZErrorPPM”, “gnps_MQScore”, “gnps_LibraryQualityString”, “gnps_SharedPeaks”, “gnps_Organism”, “gnps_npclassifier_superclass”, “gnps_npclassifier_class”, “gnps_npclassifier_subclass” and “gnps_npclassifier_pathway”. See the GNPS documentation for the description of these columns (https://ccms-ucsd.github.io/GNPSDocumentation/spectrumcuration/#adding-single-spectra).

If there is more than one GNPS result for a single msclusterID the results are concatenated with a ‘;’, except for the “gnps_Smiles” column which is concatenated with a ‘,’ (ease visualization in cytoscape).

4.12.2. ***gnps_result* Examples**

Fake example to use the GNPS Identification Join command:

```
$ node np3_workflow.js gnps_result --cluster_info_path
  "/path/to/the/output/dir/GNPS_result/clusterinfo/file"
  --result_specnets_DB_path
  "/path/to/the/output/dir/GNPS_result/result_specnets_DB/file.tsv"
  --ms_count_path "/path/to/the/output/NP3/count_files/count.csv"
```

4.12.3. ***gnps_result* Details**

The GNPS library identifications results should be obtained with the collection of consensus spectra present in the NP^3 MS workflow output MGF files, from the clustering (Step 3) or the clean (Step 5) results. In the cleaning Step 5 some consensus spectra can be joined, mostly due to the fragmented clusters case. Then, if the clean count table is informed for a library identification result obtained with the MGF from the clustering Step 3, the *gnps_result* will aggregate the identification results of the joined consensus spectra from Step 5 into a single result, using as reference the column ‘joinedIDs’.

In the GNPS Molecular Networking (MN) workflow result, the ‘clusterinfo’ file contains the column ‘SpecIdx’ which is equal to the SCANS numbers present in the NP^3 MS workflow MGF file used for the identification job. The SCANS numbers of the MGF files are equal to the msclursterIDs of the NP^3 MS workflow count tables and some of then can be aggregated in the clean count tables, this information is present in the ‘joinedIDs’ column. The column ‘ClusterIdx’ of the GNPS MN output 'clusterinfo' file is equivalent to the column ‘#Scan’ of the 'result_specnets_DB' file, and they are used to join the information of both files to obtain for each identification result the correct reference (the ‘SpecIdx’ column) to the msclusterIDs present in the NP^3 MS workflow count tables.

In the GNPS Library Search (LS) workflow result, the ‘#Scan’ column present in the output file is equal to the SCANS numbers of the MGF files. These SCANS are equal to the msclusterIDs of the NP^3 MS workflow count tables. In this case, the join is directly performed.

4.13. Command *chr*

Runs the R interactive script to extract chromatogram(s) from raw MS1 data files (mzXML, mzData and mzML) and saves to PNG image(s) file(s). Depending on the provided parameters this can be a total ion chromatogram (TIC), a base peak chromatogram (BPC) or an extracted ion chromatogram (XIC) extracted from each sample/file. In the interactive prompt is possible to select a mz and/or retention time window to better visualize interesting parts of the sample’s chromatograms. The plots can be grouped by data collection batch, by a selection of samples, or all the samples can be plotted together.

Options for the command *chr*:

- *-n, --data_name* [name] the data collection name. Used for verbosity (default: -)
- *-b, --metadata* <file> path to the metadata table CSV file
- *-d, --raw_data_path* <path> path to the folder containing the input LC-MS/MS raw spectra data in mzXML, mzData and mzML format
- *-h, --help* output usage information

4.13.1. *chr* Results PNG image(s) file(s) with the extracted chromatogram(s) of the selected sample(s) and grouped as specified in the options.

4.13.2 *chr* Examples Fake example to execute the MS^1 Viewer command:

```
$ node np3_workflow.js chr --metadata "/path/to/the/metadata/file/test_np3_metadata.csv"
  --data_name "test_np3" --raw_data_path "/path/to/the/raw/data/directory"
```

```
$ node np3_workflow.js chr -b "/path/to/the/metadata/file/test_np3_metadata.csv"
  -d "/path/to/the/raw/data/directory"
```

4.14. Command *spectra_viewer*

This command runs the Streamlit interactive Web App to visualize and compare MS2 spectra. It receives as input a MGF file or a peak list. Currently, it is only supported for Unix OS. It is also possible to manipulate, filter, calculate similarity of the spectra and save PNG or SVG plots.

Options for the command *spectra_viewer*:

- *-p, --port* [port_number] localhost server port number (default: "8501")
- *-h, --help* output usage information

4.14.1. *spectra_viewer* Examples: Example to execute the MS^2 spectra viewer application:

```
$ node np3_workflow.js spectra_viewer --port 8080
```

4.15. Command *test*

Run the NP^3 MS Workflow test suit. I consists of some use cases to test the workflow consistency in all steps. It can also be used to test if the installation was successful.

It may take a long time; 30 minutes is expected. This option is intended for debugging purposes, and is not a part of the analysis workflow.

Options for the command *test*:

- *-p, --pre_process* [x] : 'TRUE' or 'FALSE' to test the pre-process step. (default: "FALSE")
- *-t, --tremolo* [x] : 'TRUE' or 'FALSE' to test the tremolo step. (default: "FALSE")
- *-s, --skip* [x] : Skip to test x (default: 1)
- *-h, --help* output usage information

4.15.1. *test* Results: The NP^3 jobs for the tested datasets. They are stored inside the test/Bra346 folder in the repository. The metadata table files present in the test/Bra346 folder are used in the testing cases and should not be modified.

4.15.2. *test* Examples: Example to execute the test suit from NP^3 MS Workflow:

```
$ node np3_workflow.js test
```

References

1. Tautenhahn, R., Böttcher, C. & Neumann, S. Highly sensitive feature detection for high resolution LC/MS. BMC Bioinformatics 9, 504 (2008). https://doi.org/10.1186/1471-2105-9-504 (XCMS::CentWave Algorithm)

2. Colin A. Smith, Elizabeth J. Want, Grace O'Maille, Ruben Abagyan and Gary Siuzdak. "XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification" Anal. Chem. 2006, 78:779-787 (XCMS::PeakGroup and XCMS::adjustRtime-peakGroups algorithms)
3. Gatto L, Lilley KS. MSnbase-an R/Bioconductor package for isobaric tagged mass spectrometry data visualization, processing and quantitation. Bioinformatics. 2012 Jan 15;28(2):288-9. doi: 10.1093/bioinformatics/btr645. Epub 2011 Nov 22. PMID: 22113085 (MSnbase R package)
4. Owen D. Myers, Susan J. Sumner, Shuzhao Li, Stephen Barnes, and Xiuxia Du. Detailed Investigation and Comparison of the XCMS and MZmine 2 Chromatogram Construction and Chromatographic Peak Detection Methods for Preprocessing Mass Spectrometry Metabolomics Data. Analytical Chemistry 2017 89 (17), 8689-8695 DOI: 10.1021/acs.analchem.7b01069 (XCMS results investigation)
5. Myers OD, Sumner SJ, Li S, Barnes S, Du X. One Step Forward for Reducing False Positive and False Negative Compound Identifications from Mass Spectrometry Metabolomics Data: New Algorithms for Constructing Extracted Ion Chromatograms and Detecting Chromatographic Peaks. Anal Chem. 2017 Sep 5;89(17):8696-8703. doi: 10.1021/acs.analchem.7b00947. Epub 2017 Aug 17. PMID: 28752754. (XCMS results investigation)
6. Ramsay, D.. "Applications of Clustering Algorithms in the Analysis of Mass Spectrometry Data." (2017).
7. Frank AM, Bandeira N, Shen Z, Tanner S, Briggs SP, Smith RD, Pevzner PA. Clustering millions of tandem mass spectra. J Proteome Res. 2008 Jan;7(1):113-22. doi: 10.1021/pr070361e. Epub 2007 Dec 8. PMID: 18067247; PMCID: PMC2533155. (MS-Clsuter)
8. Wang, M., Carver, J., Phelan, V. et al. Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking. Nat Biotechnol 34, 828–837 (2016). https://doi.org/10.1038/nbt.3597 (GNPS)
9. Watrous, J., Roach, P., Alexandrov, T., Heath, B., Yang, J., Kersten, R., Voort, M., Pogliano, K., Gross, H., Raaijmakers, J., Moore, B., Laskin, J., Bandeira, N., & Dorrestein, P. (2012). Mass spectral molecular networking of living microbial colonies. Proceedings of the National Academy of Sciences, 109(26), E1743–E1752.
10. Stein SE, Scott DR. Optimization and testing of mass spectral library search algorithms for compound identification. J Am Soc Mass Spectrom. 1994 Sep;5(9):859-66. doi: 10.1016/1044-0305(94)87009-8. PMID: 24222034.
11. van den Berg, R.A., Hoefsloot, H.C., Westerhuis, J.A. et al. Centering, scaling, and transformations: improving the biological information content of metabolomics data. BMC Genomics 7, 142 (2006). https://doi.org/10.1186/1471-2164-7-142
12. "Mass Spectrometry Adduct Calculator" Fiehn Lab, 2016, https://fiehnlab.ucdavis.edu/staff/kind/Metabolomics/MS-Adduct-Calculator/. Accessed 01 December 2020
13. Page, Lawrence; Brin, Sergey; Motwani, Rajeev and Winograd, Terry, The PageRank citation ranking: Bringing order to the Web. 1999 http://dbpubs.stanford.edu:8090/pub/showDoc.Fulltext?lang=en&doc=1999-66&format=pdf
14. Jon Kleinberg, Authoritative sources in a hyperlinked environment Journal of the ACM 46 (5): 604-32, 1999. doi:10.1145/324133.324140. http://www.cs.cornell.edu/home/kleinber/auth.pdf.
15. Louis-Félix Nothias, Mélissa Nothias-Esposito, Ricardo da Silva, Mingxun Wang, Ivan Protsyuk, Zheng Zhang, Abi Sarvepalli, Pieter Leyssen, David Touboul, Jean Costa, Julien Paolini, Theodore Alexandrov, Marc Litaudon, and Pieter C. Dorrestein. Bioactivity-Based Molecular Networking for the Discovery of Drug Leads in Natural Product Bioassay-Guided Fractionation Journal of Natural Products 2018 81 (4), 758-767 DOI: 10.1021/acs.jnatprod.7b00737
16. Allard PM, Péresse T, Bisson J, Gindro K, Marcourt L, Pham VC, Roussi F, Litaudon M, Wolfender JL. Integration of Molecular Networking and In-Silico MS/MS Fragmentation for Natural Products Dereplication. Anal Chem. 2016 Mar 15;88(6):3317-23. doi: 10.1021/acs.analchem.5b04804. Epub 2016 Mar 1. PMID: 26882108 (ISDB)
17. Gu J, Gui Y, Chen L, Yuan G, Lu H-Z, Xu X (2013) Use of Natural Products as Chemical Library for Drug Discovery and Network Pharmacology. PLoS ONE 8(4): e62839. https://doi.org/10.1371/journal.pone.0062839 (UNPD)
18. Mingxun Wang and Nuno Bandeira. Spectral Library Generating Function for Assessing Spectrum-

Spectrum Match Significance. Journal of Proteome Research 2013 12 (9), 3944-3951. DOI: 10.1021/pr400230p (tremolo)

19. Lopes CT, Franz M, Kazi F, Donaldson SL, Morris Q, Bader GD. Cytoscape Web: an interactive web-based network browser. Bioinformatics. 2010 Sep 15;26(18):2347-8. doi: 10.1093/bioinformatics/btq430. Epub 2010 Jul 23. PMID: 20656902; PMCID: PMC2935447
20. Bastian M., Heymann S., Jacomy M. (2009). Gephi: an open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media
21. Kim HW, Wang M, Leber CA, Nothias LF, Reher R, Kang KB, van der Hooft JJJ, Dorrestein PC, Gerwick WH, Cottrell GW. NPClassifier: A Deep Neural Network-Based Structural Classification Tool for Natural Products. J Nat Prod. 2021 Oct 18. doi: 10.1021/acs.jnatprod.1c00399. Epub ahead of print. PMID: 34662515. (NPClassifier)
22. van Santen, J. A.; Poynton, E. F.; Iskakova, D.; McMann, E.; Alsup, T. A.; Clark, T. N.; Fergusson, C. H.; Fewer, D. P.; Hughes, A. H.; McCadden, C. A.; Parra Villalobos, J.; Soldatou, S.; Rudolf, J. D.; Janssen, E. M.-L.; Duncan, K. R.; Linington, R. G.* "The Natural Products Atlas 2.0: A Database of Microbially-Derived Natural Products", Nucleic Acids Research, 2021, ASAP. 10.1093/nar/gkab941 (NPAtlas)
23. Djoumbou Feunang Y, Eisner R, Knox C, Chepelev L, Hastings J, Owen G, Fahy E, Steinbeck C, Subramanian S, Bolton E, Greiner R, and Wishart DS. ClassyFire: Automated Chemical Classification With A Comprehensive, Computable Taxonomy. Journal of Cheminformatics, 2016, 8:61. DOI: 10.1186/s13321-016-0174-y (ClassyFire)
24. F Huber, L Ridder, S Verhoeven, JH Spaaks, F Diblen, S Rogers, JJJ van der Hooft, "Spec2Vec: Improved mass spectral similarity scoring through learning of structural relationships", bioRxiv, https://doi.org/10.1101/2020.08.11.245928, https://github.com/iomega/spec2vec_gnps_data_analysis?tab=readme-ov-file (spec2vec)
25. F. Huber, S. Verhoeven, C. Meijer, H. Spreeuw, E. M. Villanueva Castilla, C. Geng, J.J.J. van der Hooft, S. Rogers, A. Belloum, F. Diblen, J.H. Spaaks, (2020). matchms - processing and similarity evaluation of mass spectrometry data. Journal of Open Source Software, 5(52), 2411, https://doi.org/10.21105/joss.02411 (matchms)