Contents

# Introduction

Tandem mass spectrometry is a technique frequently used for small molecule identification. Automated structure elucidation is usually performed by spectral library search. Building a local high-quality spectral library is an essential step thus often lacking in metabolomics and pharmaceutical laboratories. This is often due to the data confidentiality (e.g drug metadata).

MergeION fills these gaps and enables building local spectral libraries without sharing them in public domains. It works by extracting MS1 and MS2 scans from one or multiple raw chromatogram files according to m/z (and retention time) provided by users. They are then merged into a GNPS-style spectral library combining user-provided metadata. It is compatible with mzML/mzXML format acquired on Thermo, Water or Bruker instruments, in either DDA (Data-driven acquisition) or targeted MS/MS-mode.

In addition, several spectral search algorithms are available, allowing users to search an unknown spectrum in their local database or public databases (i.e. drug structures in GNPS, MASSBANK and DrugBANK).

# Installation and loading

The library is available from Bioconductor (http://www.bioconductor. org).

*# Install BiocManager if it has not been installed previously:* **if** (!requireNamespace("BiocManager", quietly=TRUE)) install.packages("BiocManager")

*# Install MergeION:*   
install.packages("remote")   
install.packages("Rcpp")   
BiocManager::install("multtest")   
Sys.setenv(R\_REMOTES\_NO\_ERRORS\_FROM\_WARNINGS="true") BiocManager::install("daniellyz/MergeION2")

The package is loaded using

***Library(MergeION)***

All dependencies are loaded automatically.

# Input files

## LC/MS data

MergeION handles high-resolution LC/MS spectra in mzML format in centroid1 or in profile mode. If the software doesn’t allow output in centroided data. It is recommended to convert the data to the file format mzML; please refer to <https://ccms-ucsd.github.io/GNPSDocumentation/fileconversion/> for conversion to mzML.

The data in the examples come from the Rpackages: Rmassbank and CompMS2Miner which are measured on LTQ Orbitrap XL and unspecified HRMS devicerespectively.

## Metadata

A compound list in CSV format is required to identify all compounds unambiguously. The CSV file is required to have at least the following columns, which are used for further processing and must be named correctly (but present in any order).

| **Algorithm** | **MergeION (Default)** | **compMS2Miner** | **RMassBank** |
| --- | --- | --- | --- |
| Input Data format | mz(X)ML | mz(X)ML | mz(X)ML |
| Input Acquisition mode | Targeted + DDA | Targeted + DDA | Targeted |
| Input Metadata (Compulsory) | ID + PEPMASS | ID + PEPMASS + RT | ID + SMILES + RT + Filename |
| Input Metadata (Optional) | RT, Filename | Filename |  |
| Search Method | Highest TIC scan | Multiple scan summation | Multiple scan aggregation |
| Spectrum POST-processing | - | Dynamic filtering | Recalibration T/F |
| Extra function | Isomer detection | Isomer detection | Fragment formula annotation |

## Settings

A number of different settings influences the algorithms. They are partly parameters for data processing and partly constants used for annotation. The function proposes three data processing algorithms to pick up MS1/MS2 scans from DDA or targeted mode LC-MS/MS data, merge them into a spectral library and create a spectral similarity-based molecular network. For the arguments to add in each function it is recommended to load information using: ?library\_generator

The big difference between the algorithms is in the post-processing. MergeION doesn’t apply any post-processing whilst; CompMS2Miner applies dynamic filtering, and RMassBank recalibrates. In dynamic filtering it allows the user to create selections from all components of an analysis based on which the whole document will be filtered. You will only see data in your components that meet the filter criteria. In the first step of RMassBank, MSMS spectra of compounds are extracted from raw LC-MS data files, the MSMS spectra are recalibrated using assigned fragment formulas, and effectively denoised by using only annotated peaks (plus peaks which can be manually added.) In the second step, the processed, recalibrated, cleaned data is prepared for submission to a MassBank database. Compounds are first automatically annotated using information from the Chemical Translation Service (CTS). After manually checking and fixing the annotations, the information is compiled together with the spectral data into MassBank records, which can then be uploaded to a MassBank database.

### Usage

There are different parameters that need to be tweaked in order to run the function library\_generation

First the input has to be defined an example for the tutorial this would be;

The input library can be NULL when not using a previous library

* Input\_library = “Here file”

Then the lcms\_files and meta\_data files should be added. Beware that the lcms\_files should be in mzXML/mzML file and metadata in comma delimited CSV file.

* lcms\_files = c(“File1”, “File2”)
* metadata\_file = “Filename”

Then the processing algorithm can be selected by changing the number in between the brackets.

* processing.algorithm = c("Default", "compMS2Miner", "RMassBank")[1]

After adding the files polarity and mass level should be defined by the number in between the brackets.:

* polarity = c("Positive", "Negative")[1]
* mslevel = c(1, 2)

Furthermore, the following threshold parameters can be defined

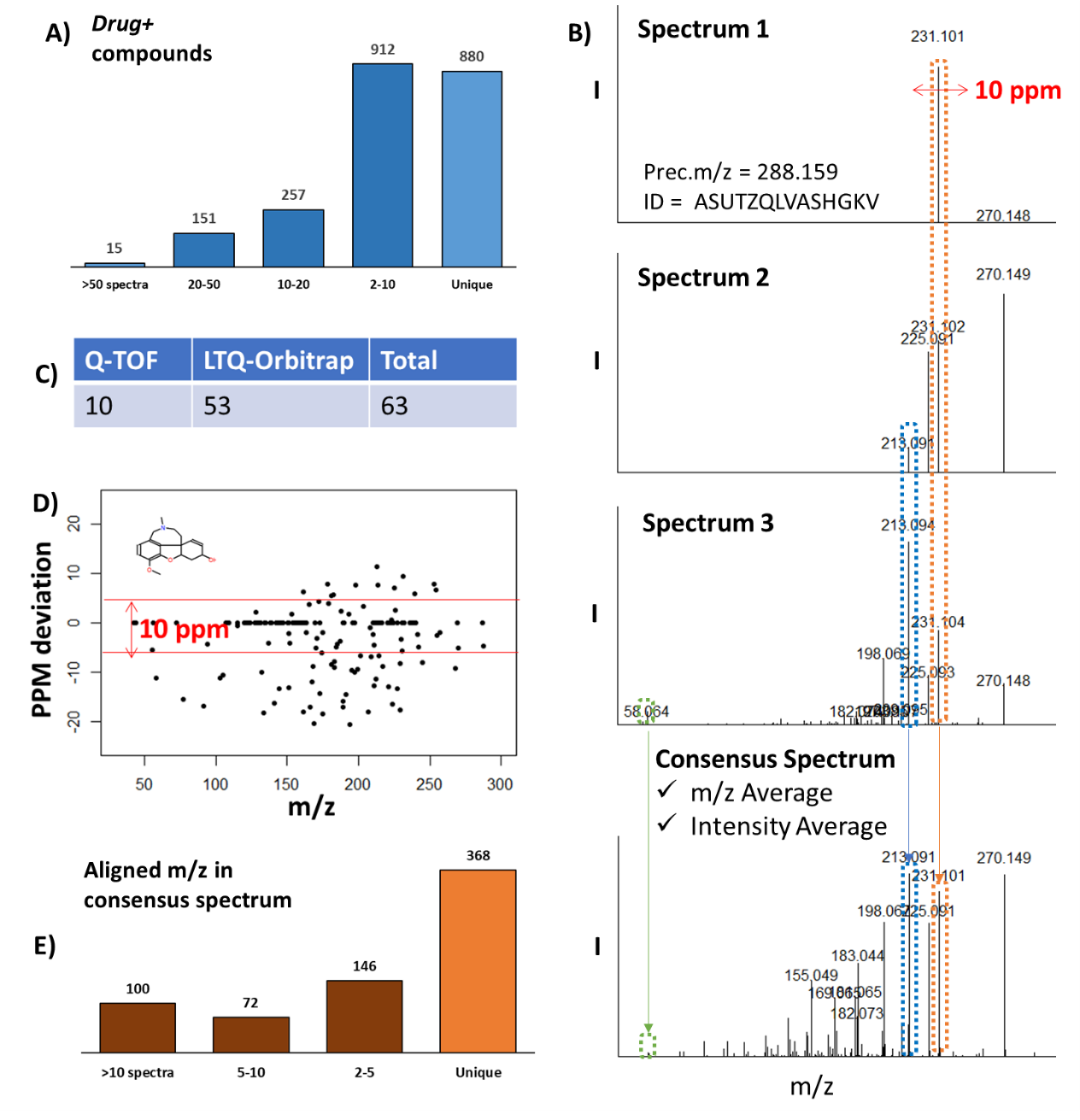
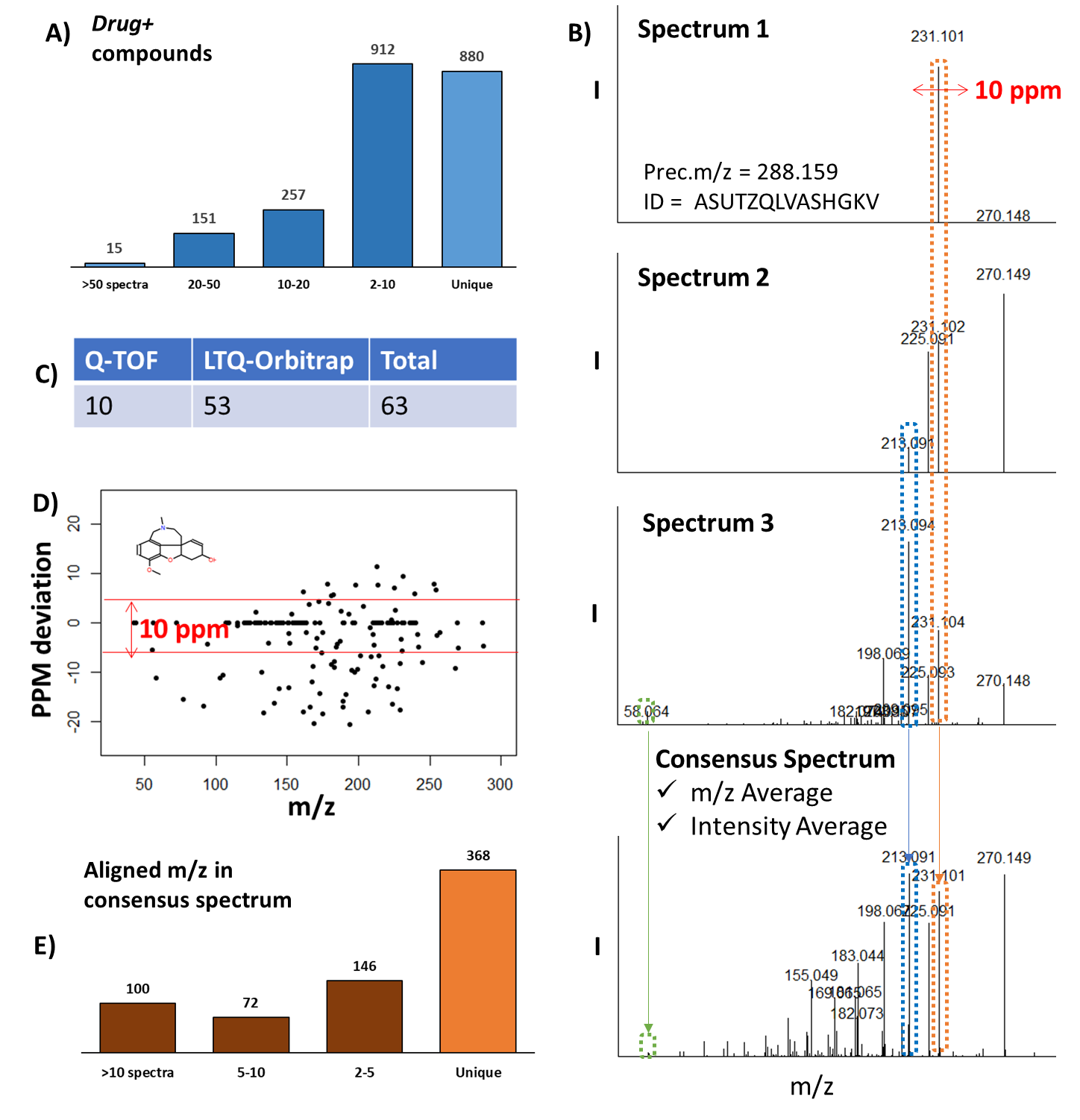
* params.search = list(
  + mz\_search = 0.01,
  + ppm\_search = 10,
  + rt\_seach = 15, rt\_gap = 30)

Preprocessing can be done by the RMassBank algorithm,

* params.ms.preprocessing = list(normalized = TRUE, baseline = 1000, relative = 0.1,

max\_peaks = 200, recalibration = 0),

Consensus library can be generated based on the following parameter

* params.consensus = list(
  + consensus = FALSE,
  + consensus\_method = c("consensus", "common\_peaks", "most\_recent")[1],
  + consensus\_window = 0.02),
* 
* **Figure 6 *consensus spectra:*** *three different spectra are formed to one consensus spectra based on m/z average and intensity average.*

Networking

params.network = list(network = FALSE, similarity\_method = "Cosine", min\_frag\_match =

6, min\_score = 0.6, topK = 10, reaction\_type = "Metabolic", use\_reaction = FALSE),

Optional parameters that can be added to the sample.

params.user = list(sample\_type = "", user\_name = "", comments = "")

# 4 The workflow

## 4.1 Library generation

### 4.1.1 MergeION Algorithm

For MergeION it is important that the CSV file with metadata contains; ID + PEPMASS.

library\_generator(

input\_library = NULL,

lcms\_files = “Test\_file.mgf”,

metadata\_file = “Metadata.csv”,

polarity = c("Positive", "Negative")[1],

mslevel = c(1, 2),

add.adduct = TRUE,

processing.algorithm = c("Default", "compMS2Miner", "RMassBank")[1],

params.search = list(mz\_search = 0.01, ppm\_search = 10, rt\_seach = 15, rt\_gap = 30),

params.ms.preprocessing = list(normalized = TRUE, baseline = 1000, relative = 0.1,

max\_peaks = 200, recalibration = 0),

params.consensus = list(consensus = FALSE, consensus\_method = c("consensus",

"common\_peaks", "most\_recent")[1], consensus\_window = 0.02),

params.network = list(network = FALSE, similarity\_method = "Cosine", min\_frag\_match =

6, min\_score = 0.6, topK = 10, reaction\_type = "Metabolic", use\_reaction = FALSE),

params.user = list(sample\_type = "", user\_name = "", comments = "")

)

## 4.2 Library query

The Library\_query function in MergeION can be used to search in the generated libraries. This function consists out of two parts. First, the settings for the input; and secondly, the settings for the searching parameters.

### 4.2.1 Settings for input

**Input\_library**, similar as in the generation this will be a library in rdata format. Here it is best to load the library previously with load(“library name.rdata”) into the Renvironment to reduce loading times.

**Query\_spectrum,** Two-column data matrix. Two columns represent m/z and intensity of query tandem spectrum. At least 3 valid peaks should be provided. If provided in a file use query\_file = “filename”.

### 4.2.2 Settings for searching

Threshold values for **params.search** can be defined,

mz\_search = 0.01, ppm\_search = 10, rt\_seach = 15, rt\_gap = 30

and for the **params.query.sp**, here the mode for scoring, precursor mass, polarity, minimal fragment matching can be defined. For the example shown below these are:

(prec\_mz = 235.1805, use\_prec = T, polarity = "Positive", method = "Cosine", min\_frag\_match = 6, reaction\_type = "Metabolic")

### 4.2.3 Example

Peaklist 🡪 Example data for lidocaine

#load Libraries and Data

library(MergeION)

library(R.utils)

load("JANSSEN\_POS.rdata")

#Library query  
qspec <- read.delim("inputdata.txt", sep = ",")  
results = library\_query(

input\_library = JANSSEN\_POS,

query\_expression = "IONMODE=Positive",

query\_spectrum = qspec,

query\_file = NULL,

params.search = list(mz\_search = 0.01, ppm\_search = 10, rt\_seach = 15, rt\_gap = 30),

params.query.sp = list(prec\_mz = 235.1805, use\_prec = T, polarity = "Positive", method =

"Cosine", min\_frag\_match = 6, reaction\_type = "Metabolic")

)