

mzmine

Documentation

The mzmine Community

This documentation is licensed under the MIT License (MIT).

Table of contents

1. Welcome to the mzmine documentation!	6
1.1 Getting started with mzmine	6
1.2 User-interface gallery	7
1.3 mzmine news	8
1.4 History of mzmine	8
1.5 About this documentation	8
1.6 How to contribute	8
2. Getting Started	10
2.1 Download	10
2.2 System requirements	10
2.3 Install or update mzmine	10
2.4 Running mzmine	10
2.5 Sign in / Sign up	11
2.6 Set user preferences	12
2.7 Quick workflows in the mzwizard	12
2.8 Introduction video	12
2.9 Workflow overviews	13
3. Users management	14
3.1 Sign up / Registration	14
3.2 Offline use	14
3.3 User account console - changing user details	15
3.4 Reset password	15
3.5 Sign up / Registration details	15
3.6 Command-line interface (CLI)	19
3.7 Alternative Sign in / Sign up	19
3.8 mzmine's educational non-profit program	19
3.9 Frequently asked questions (FAQ)	19
4. Learners corner	21
4.1 Useful channels	21
4.2 mzmine videos	21
4.3 General Mass Spectrometry	21
4.4 (Computational) Metabolomics	21
4.5 SIRIUS	21
5. Main window overview	22
5.1 MS data files and feature lists tab	22

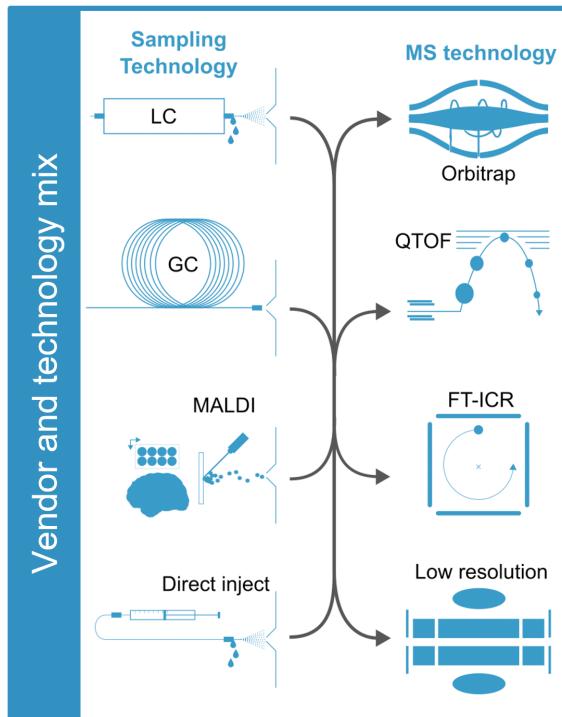
5.2 Main content pane	22
5.3 Main menu	23
5.4 Task overview	23
5.5 Page Contributors	23
6. Processing wizard	24
6.1 mzwizard parameter tabs	25
6.2 [1] Data import	25
6.3 [2] Sample introduction system	25
6.4 [3] Ion mobility instrument	26
6.5 [4] MS instrument, e.g., Orbitrap, QTOF, FTICR	27
6.6 [5] Filters	28
6.7 [6] Annotation	28
6.8 [7] Workflows	29
6.9 [8] Create Batch	30
6.10 Optimizing the wizard workflow	31
7. Data handling (supported formats)	32
7.1 Data conversion to open formats (.mzML / .imzML)	32
7.2 Data sharing (FAIR)	34
7.3 Data repositories	35
7.4 Page Contributors	35
8. Performance & Benchmarking	36
8.1 General remarks	36
8.2 Benchmarking	41
9. Command-line interface (CLI) for headless batch mode	43
9.1 Selecting a user	43
9.2 Windows	43
9.3 macOS	44
9.4 Linux	44
9.5 Argument table	45
10. Integration to other tools	48
10.1 Overview	48
10.2 Page Contributors	51
11. Processing modules	52
11.1 I/O	52
11.2 File merging	84
11.3 Raw data filtering	86
11.4 Mass detection	95
11.5 LC-MS feature detection	110

11.6 LC-IMS-MS feature detection	123
11.7 MS imaging feature detection	128
11.8 MRM feature detection	130
11.9 Smoothing	135
11.10 Resolving	138
11.11 Spectral deconvolution (GC)	151
11.12 CCS Calibration and calculation	154
11.13 MS2 Scan Pairing	157
11.14 Assign MALDI MS2s to features	159
11.15 Isotope filtering	161
11.16 Feature grouping	168
11.17 Feature list filtering	178
11.18 Alignment	191
11.19 Gap filling	203
11.20 Normalization	207
11.21 Precursor mass search	216
11.22 Spectra search	220
11.23 Other parameters	244
11.24 Other detectors (UV)	248
12. Visualization modules	260
12.1 Visualization modules	260
12.2 MS data visualisation	261
12.3 Ion mobility raw data overview (LC-IMS-MS)	268
12.4 Image viewer	270
12.5 Image co-localization	272
12.6 Processed data visualization	275
12.7 Massvoltammogram	304
12.8 Statistics dashboard	308
12.9 Multi detector visualizer	311
12.10 Correlation dashboard	313
13. Workflows	315
13.1 Untargeted LC-MS Workflow	315
13.2 Untargeted LC-IMS-MS Workflow Overview	318
13.3 Untargeted GC-MS Workflow	323
13.4 Imaging Workflow	325
13.5 Batch mode processing	327
13.6 mzwizard	328
13.7 Library generation workflow	329

13.8 SIMSEF Workflow	333
13.9 MRM workflow	335
13.10 Workflow optimization	336
14. Tools	339
14.1 Spatial ion mobility-scheduled exhaustive fragmentation (SIMSEF)	339
15. Additional resources	341
15.1 Installation	341
15.2 System requirements	346
15.3 General terminology	348
15.4 mzmine-specific terminology	351
15.5 Ion mobility spectrometry terminology	354
15.6 Graphical comparison of LC-MS and LC-IMS-MS data	357
15.7 Spectral similarity measures	359
15.8 Other Detector Workflow Glossary	361
15.9 Additional figures	363
16. Contribute	366
16.1 Coding	366
16.2 How to contribute to the MZmine documentation	415
16.3 Markdown template to create online documentation for a processing module	420
16.4 Module name	420
17. Acknowledgements	422
17.1 Related projects	422
17.2 Libraries we use in MZmine	422
18. Troubleshooting	423
18.1 GUI freezes	423
18.2 mzmine does not start	423
18.3 mzML files not importing	423
18.4 Thermo raw files not importing	423
18.5 Scans not sorted in retention time dimension	423
18.6 Scans were recognised as empty	423
18.7 Zero intensity values in scans	423
18.8 Reporting issues	423
19. Changelog	425
19.1 Latest mzmine changes	425
20. Workshops	427

1. Welcome to the mzmine documentation!

mzmine is an open-source and platform-independent software for mass spectrometry (MS) data processing and visualization. The workflow is optimized to allow processing of large-scale mass spectrometry studies. mzmine supports multiple raw data formats, including open formats such as mzML and several vendor-specific file types. Data from various instrumental setups, such as LC-MS, LC-IMS-MS, GC-MS, and even MALDI-(IMS)-MS is supported. Interactive visualization tools allow you to seamlessly survey both raw data and processed results. mzmine allows you to transform your spectral raw data into meaningful feature lists and offers compound annotation approaches by spectral library matching, exact mass searches, rule based lipid annotation, and formula prediction. Furthermore, mzmine allows generation of custom spectral libraries.



mzmine



Vendor data
types and formats



High speed and
throughput



Extract knowledge
from data



Easy to use

Info

When using mzmine, please cite the corresponding paper: Robin Schmid, Steffen Heuckeroth, Ansgar Korf et al. Integrative analysis of multimodal mass spectrometry data in MZmine 3. *Nature Biotechnology* (2023), doi:10.1038/s41587-023-01690-2.

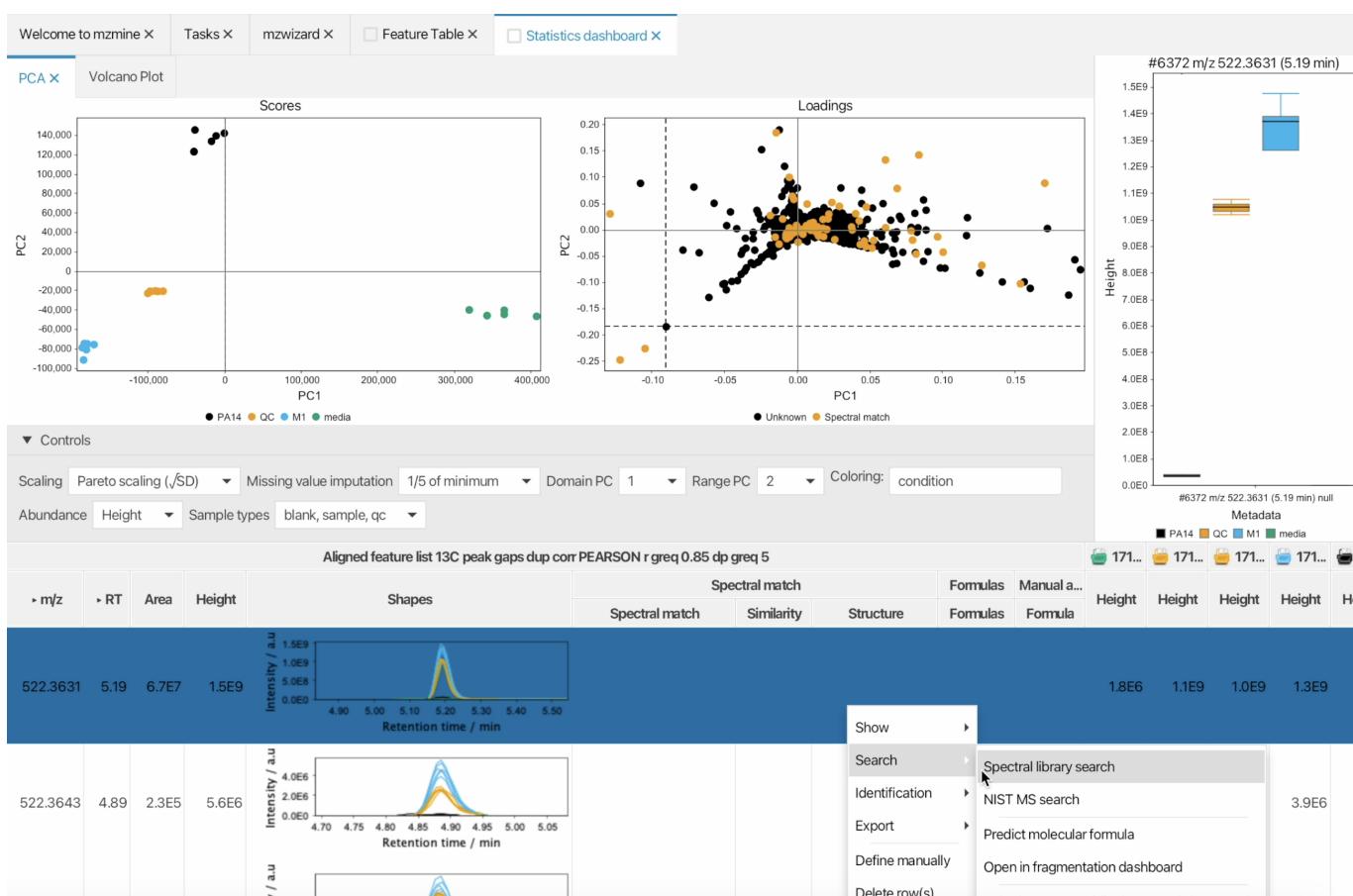
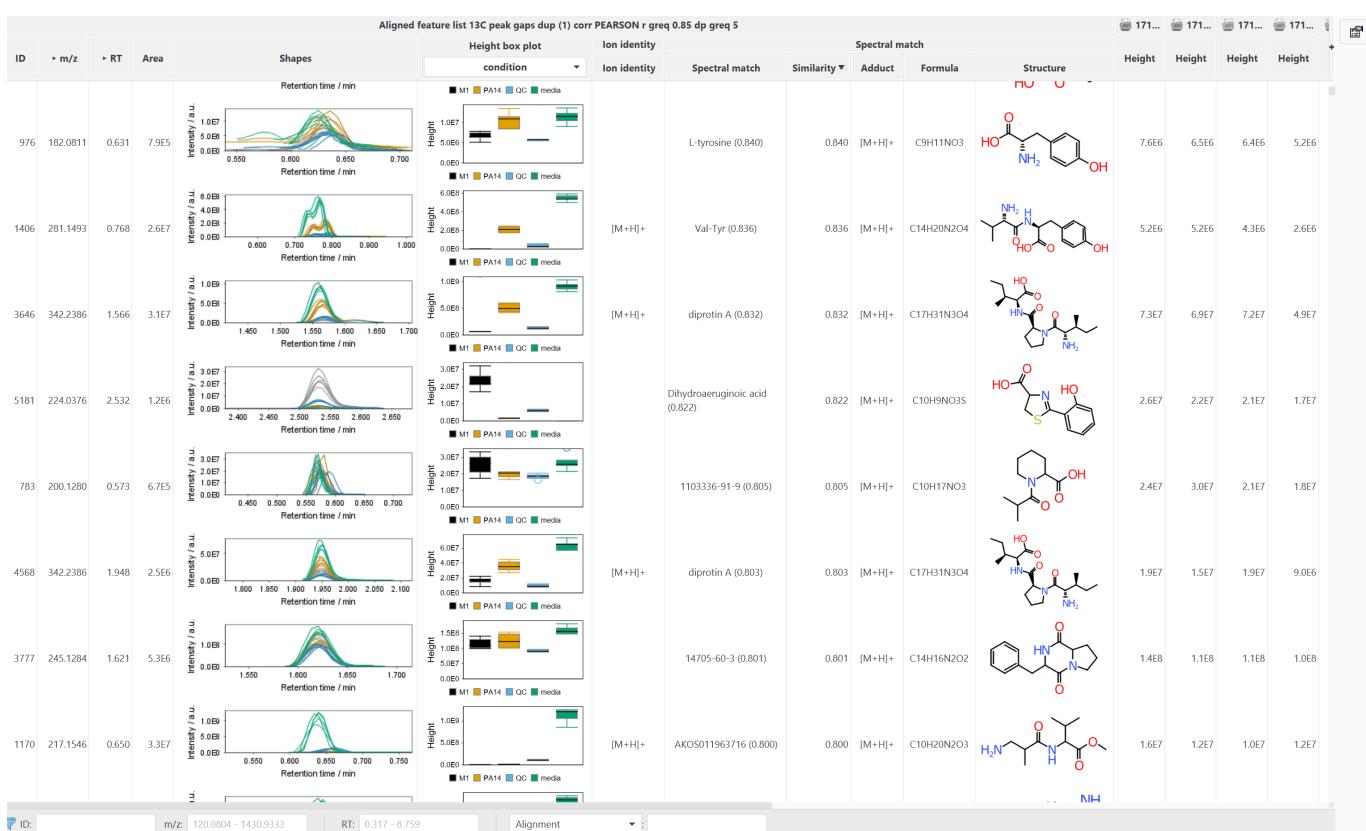
1.1 Getting started with mzmine

For information on how to install and get started with mzmine, check out our [getting started](#) page or [mzmine video tutorials on YouTube](#) and [other videos](#).

A recent mzmine introduction video:



1.2 User-interface gallery



1.3 mzmine news

Check out the latest **mzmine news** where we announce and showcase new functions.

1.4 History of mzmine

Now, mzmine is at major version 4, representing the latest evolution of the platform, delivering unprecedented performance improvements with >10x lower memory footprint and faster processing speeds compared to previous versions. This milestone transforms mzmine into an enterprise-ready solution, capable of handling large studies with complex multidimensional MS data. The mizio team offers support packages while maintaining the open-source philosophy that provides a framework for community members to develop their own ideas faster.



MZmine 3 was developed as a community effort, lead by Robin Schmid, Steffen Heuckeroth, Ansgar Korf, and Tomas Pluskal. It introduced a redesigned [GUI](#), a completely new data structure which provided the flexibility to process any type of mass spectrometry data. This included LC-MS, GC-MS and MS-imaging and introduced ion mobility support (see [LC-IM-MS data visualization](#) and [feature detection](#)). Finally, significant effort was devoted to trace memory issues and bottlenecks, resulting in an unprecedented processing performance and scalability.

Since the introduction of MZmine 2 in 2010, the project has matured into a community-driven, highly collaborative platform and its functions continue to expand based on the users' needs and feedbacks. This has also enabled the tight integration of the mzmine ecosystem with popular third-party software for MS data analysis, such as the [SIRIUS](#) suite for *in-silico* metabolite annotation, the [GNPS](#) platform with Ion Identity Molecular Networking, the [MetaboAnalyst](#) web app for univariate and multivariate statistical analysis, etc.

Such great progress was made possible by the invaluable contribution of many [developers](#) from research labs distributed all over the world!

1.5 About this documentation

Here you can find documentation for both processing and visualization modules in mzmine. Moreover, data processing pipelines for untargeted [LC-MS](#) and [LC-IMS-MS](#) feature detection as well as [MS-Imaging](#) are described and general recommendations are given.

1.6 How to contribute

The mzmine community is always welcoming new developers and contributions! You can contribute by improving existing modules or even adding new features in mzmine! Please, check out our brief [tutorial](#).

You can also contribute to this wiki and help new users to get started with mzmine! See [here](#) how to contribute to the documentation.

Ansgar Korf, Robin Schmid, Steffen Heuckeroth, omokshyna, tdamiani

⌚ November 25, 2025 10:29:24

👤 Ansgar Korf, Robin Schmid, Steffen Heuckeroth, omokshyna, tdamiani

2. Getting Started

2.1 Download

Download mzmine portable versions or installers from [GitHub](#). The software follows semantic versioning (major.minor.patch like 4.8.0). The latest version is always available under: <https://github.com/mzmine/mzmine/releases/latest>

2.2 System requirements

System requirements are available [here](#)

2.3 Install or update mzmine

Download your system-specific installer (e.g., .msi installer on Windows) and double-click to install mzmine. To update mzmine, download a new version installer and install it over the old installation. There is no need to remove the old version. Alternatively, portable versions are distributed as zip archives, which need to be unzipped to reveal the executables (e.g., .exe on Windows). See platform-specific install instructions:

- Installation on [Windows](#)
- Installation on [Linux](#)
- Installation on [macOS](#)

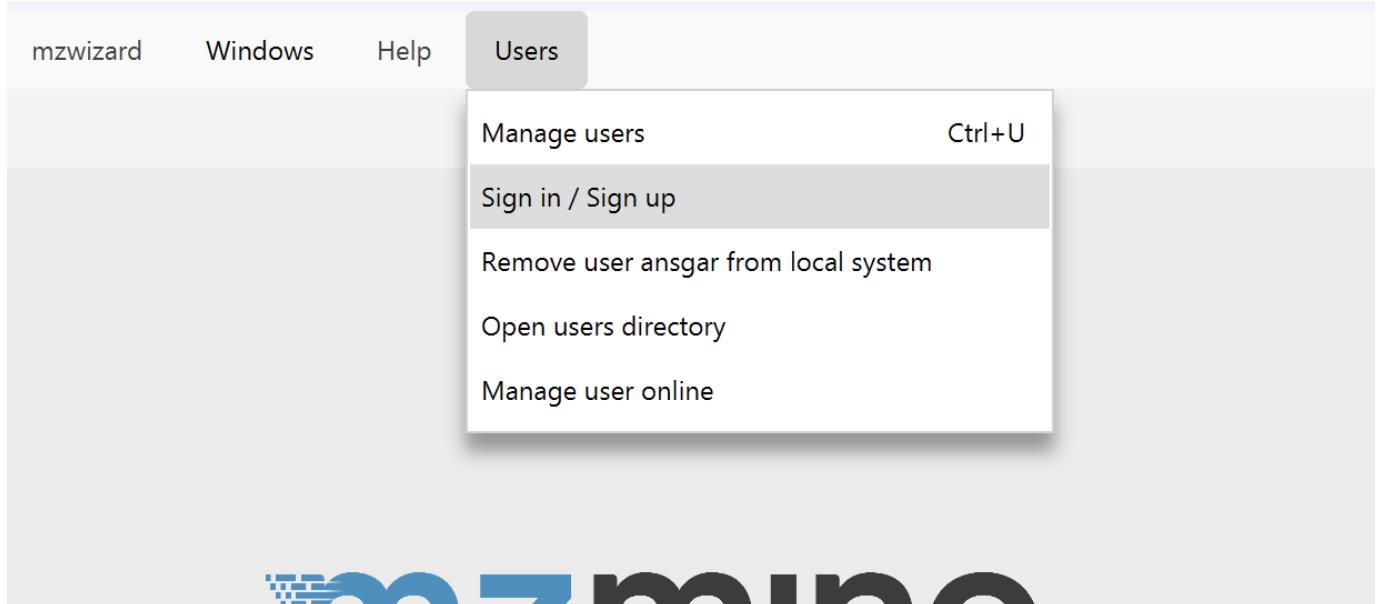
As standalone modern Java software, mzmine has **NO** further requirements and ships with its own specialized Java Runtime. This means the local Java installation has **no** impact on mzmine. Windows and macOS users might be warned that mzmine is not signed or from a trusted source and have to click run anyway.

2.4 Running mzmine

mzmine provides a user-friendly graphical user interface (GUI) that facilitates data exploration, batch optimization, and results validation. If the batch processing is optimized and the goal is to solely produce the output files without the need for a GUI, we recommend running mzmine as a [command-line tool \(CLI\)](#).

2.5 Sign in / Sign up

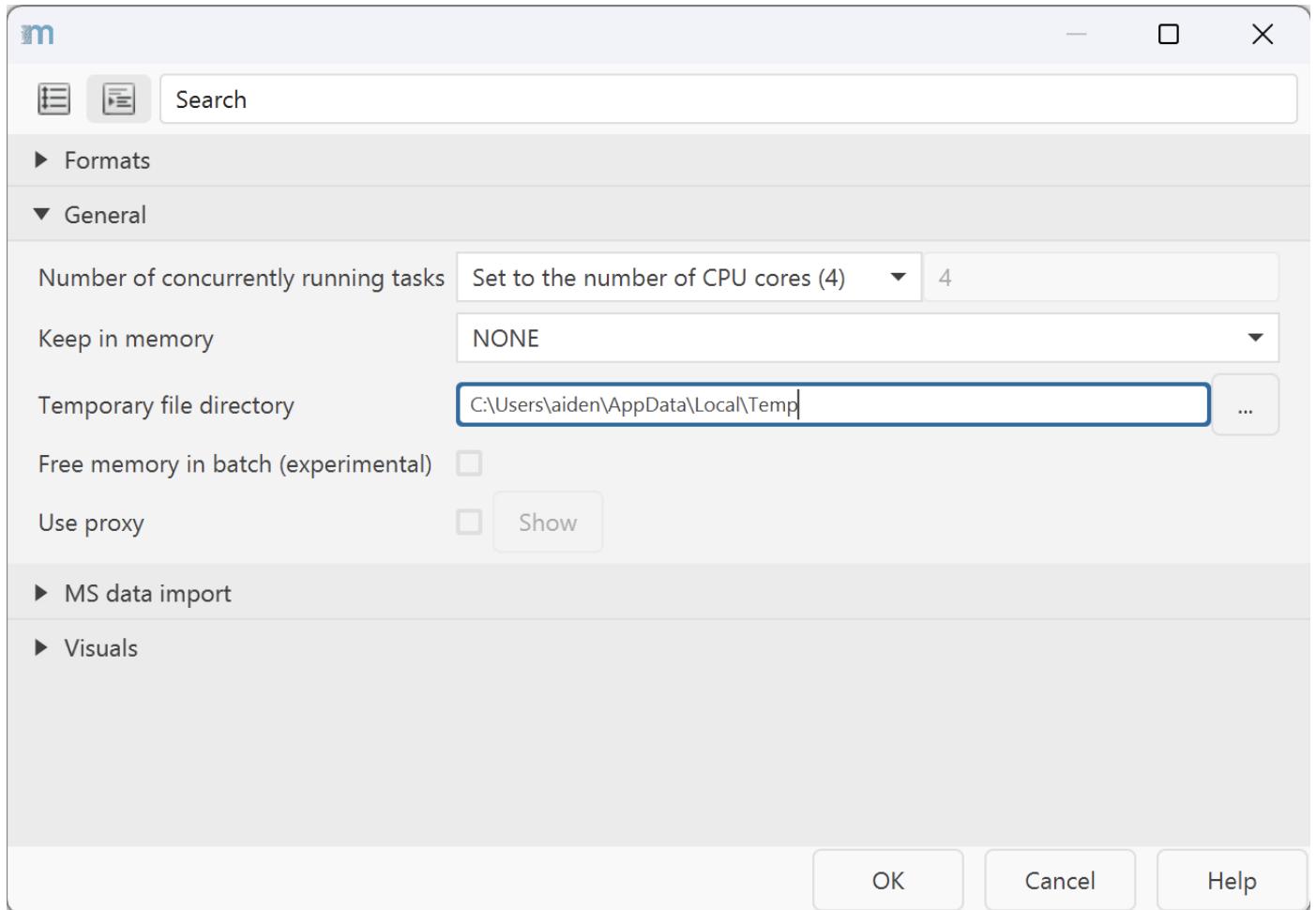
Open the **Users/User management** to sign in to an existing user or sign up for a free user account ([more details](#)).



2.6 Set user preferences

Before creating your first project, we recommend setting up some things.

1. Set a temporary file directory. Go to *Project → Set preferences → Temporary file directory*. This requires a restart to take effect.
2. We recommend setting the directory to an SSD other than your system drive with enough space for fast processing and visualizations.



3. mzmine 2 and mzmine 3 projects cannot be imported due to changes in the data structure.
4. mzmine 2 and mzmine 3 batch files cannot be imported due to parameter optimizations.

You can get familiar with the new GUI here: [Main window overview](#)

2.7 Quick workflows in the mzwizard

As a quick start you can use the **mzwizard** in the main menu or directly on the landing page. A good starting point is watching this [mzmine video tutorial on YouTube](#).

2.8 Introduction video

2.9 Workflow overviews

A quick insight to data processing workflows can be found here:

- [LC-MS workflow](#)
- [LC-IMS-IMS workflow](#)
- [Imaging workflows](#)

[Ansgar Korf, Robin Schmid, Steffen Heuckeroth, amcaraballor, omokshyna](#)

 November 25, 2025 10:29:24

 [Ansgar Korf, Robin Schmid, Steffen Heuckeroth, amcaraballor, omokshyna](#)

3. Users management

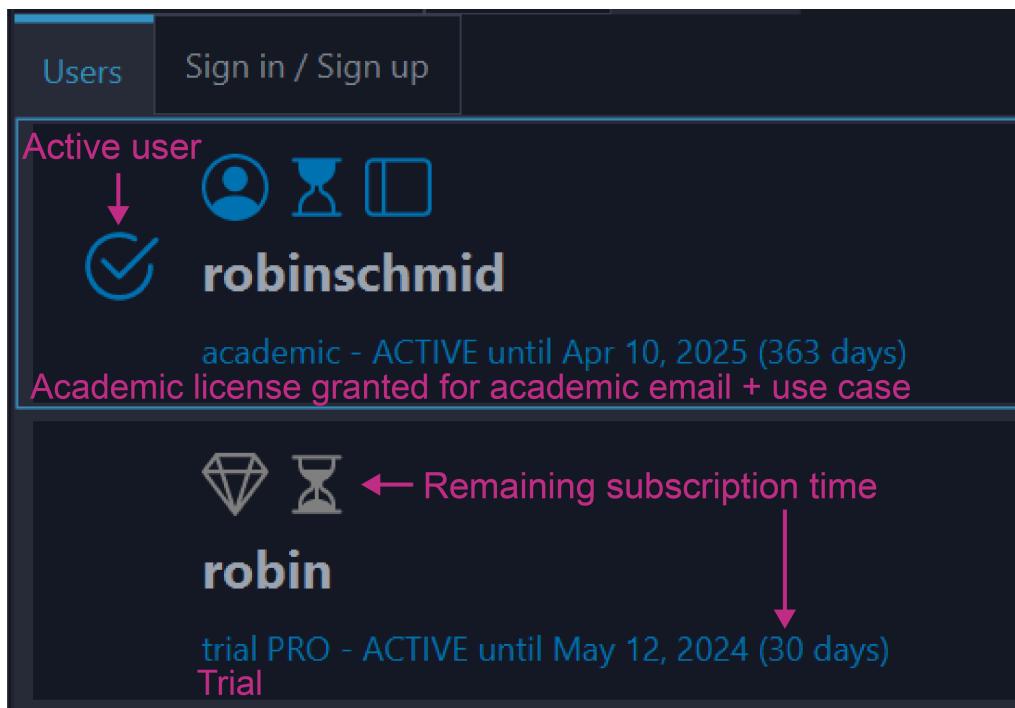
3.1 Sign up / Registration

Users can sign in into an existing account or sign up (register) a new free account within seconds. The sign up is available directly in the mzmine **Users** menu (Manage users). If you experience any issues with the built-in webview, click the external browser button and sign up online. Briefly, the user needs to provide few details including:

- **Username:** Fixed identifier that cannot be changed
- **Define your use case:** The intended use of this user profile, either **commercial use** or ** non-profit academic use**
- **Email:** Either an academic or company email address. It is important to use an academic email address to gain a non-profit academic license
- **Read and accept the Terms and Conditions**

After clicking the register button, a verification email is sent to the provided address. This can take a few minutes.

1. Please open the email
2. Click the verification link that redirects to the user portal
3. Verify the email by **Click here to proceed** to verify your email
4. Return to mzmine and refresh the page (automatically done every 30 seconds)
5. Check your license status in mzmine. Every user will receive a 30-day trial license or a non-profit academic license. Please reach out if your academic institution still needs our automatic validation.



3.2 Offline use

Sign in to your user in mzmine on a computer that is connected to the internet. Click on the menu **User/Open users directory** to open the file explorer at your systems user/.mzmine/.users directory. Copy the corresponding users (.mzuser files) to the new system into the same directory. Alternatively copy the files to the other system and drag-and-drop them into mzmine. The users will be installed. Finally, open the **Users/User management** and select the active user.

3.3 User account console - changing user details

Open the [mzio user account console](#) and sign in with your user credentials. Go to personal information to update any details. When changing the **email** the user is automatically logged out. On the next sign in, the user will receive another verification email to validate email address ownership. Revalidation of the email also triggers a revalidation of the **non-profit academic license**.

The screenshot shows the mzio & mzmine account management interface. At the top, there is a dark header bar with the mzio logo on the left, a "Sign out" button, and a user profile for "Robin Schmid". Below the header, a welcome message reads "Welcome to the mzio & mzmine account management". The main area is divided into three columns:

- Personal info** (Icon: person): Manage basic information and access Personal info.
- Account security** (Icon: shield): Control password and account access, including Signing in and Device activity.
- Applications** (Icon: computer): Track app permission and access Applications.

3.3.1 Changing password & adding Two-factor authentication

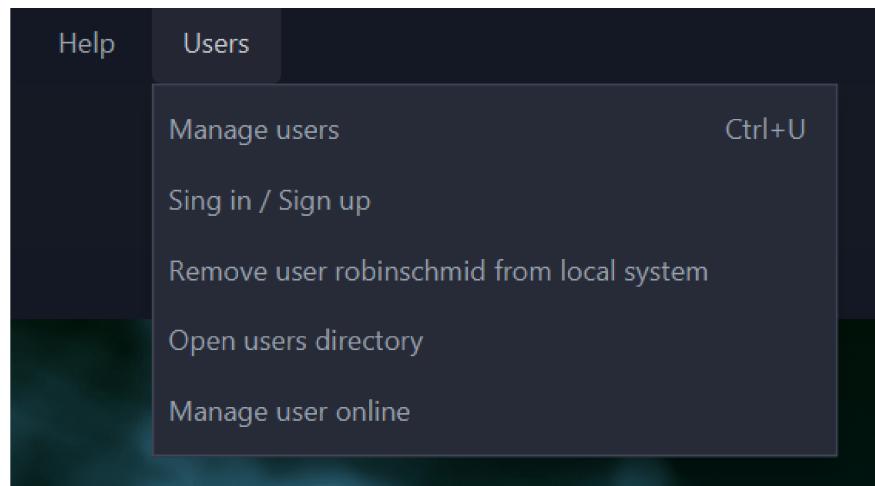
Within the account console, navigate to **Account security/Signing in** and update your password. Two-factor authentication can be activated on the same page. Start the setup and follow the instructions. This requires a mobile authenticator app that will generate a one-time-password for each sign in.

3.4 Reset password

On the sign in page in mzmine or in the [user account console](#) click on **Forgot Password**, enter your username or email. An email for resetting the password will be sent to the users email within a few minutes. Follow the instruction by clicking on the link and entering a new password.

3.5 Sign up / Registration details

Open mzmine and navigate to **Users/Manage users**, alternatively click on the user icon on the mzmine landing page. If you have a user account sign in. Otherwise, click on **register**.



Online user console

Open in external browser

Refresh page
(after email validation)

Sign in to your account

Username or email

Password

Remember me

[Forgot Password?](#)

[Sign In](#)

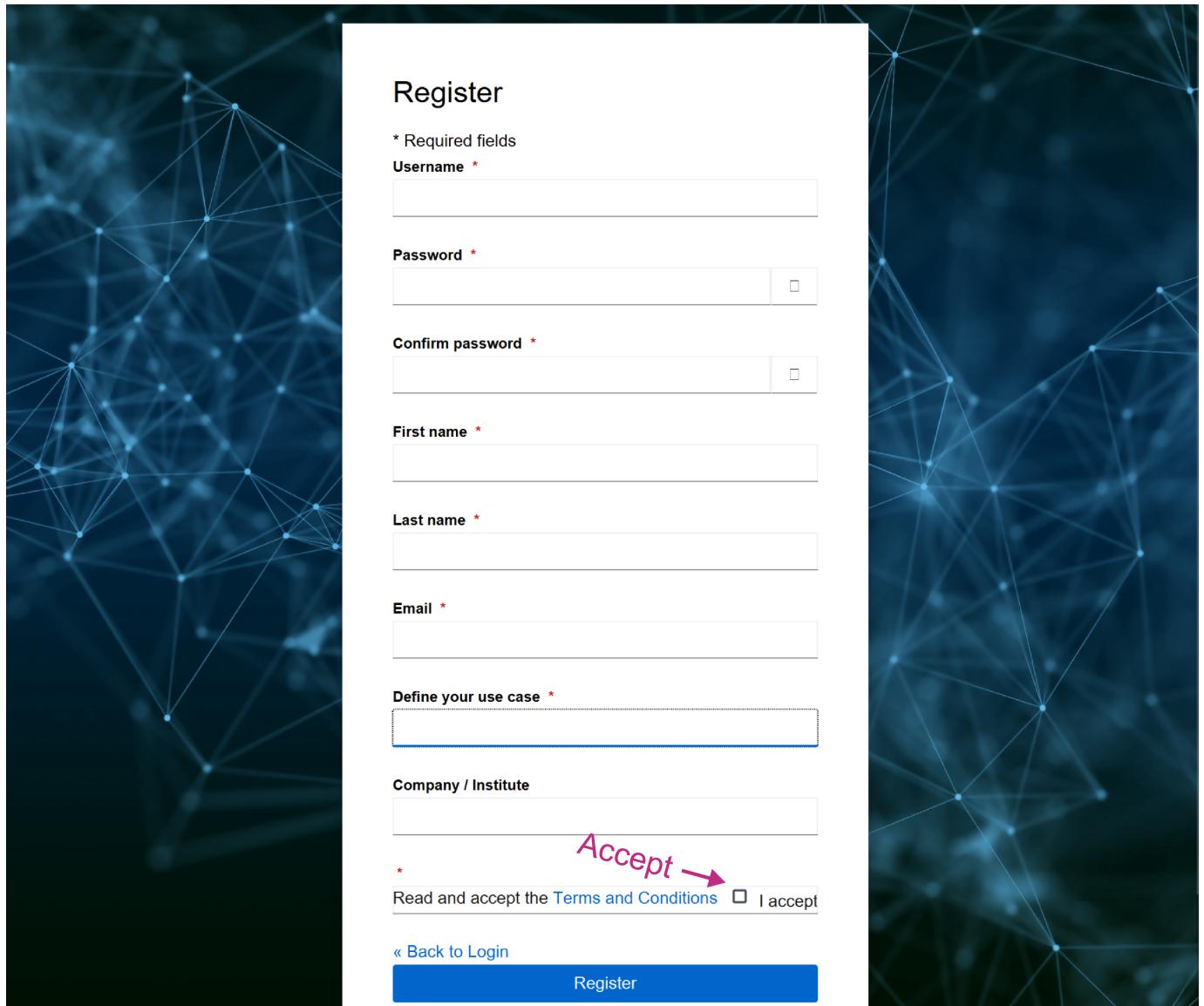
New user? [Register](#) ← [Register for free](#)

Academic User
 Simply sign up for a free research license using your active academic email address. Join the mzmine community now!

Commercial User
 Sign up and test mzmine for free today. Tell us about your experience and [get in touch](#) to unlock full mzminePRO support.

Fill in the registration form and provide all details including:

- **Username:** Fixed identifier that cannot be changed
- **Define your use case:** The intended use of this user profile, either **commercial use** or * *non-profit academic use**
- **Email:** Either an academic, company, or personal email address. It is important to use an academic email address to gain a non-profit academic license
- **Read and accept the Terms and Conditions**



Register

* Required fields

Username *

Password *

Confirm password *

First name *

Last name *

Email *

Define your use case *

Company / Institute

*

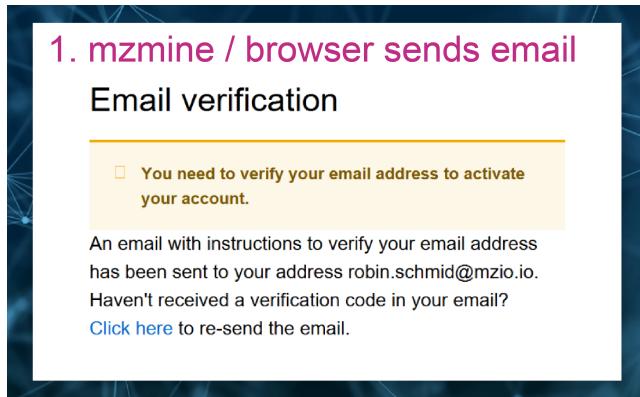
Accept 

Read and accept the [Terms and Conditions](#) I accept

[« Back to Login](#)

Register

After clicking the register button, a verification email is sent to the provided address. This can take a few minutes.



3. Click confirmation link in browser



2. Click verification link in email

Verify email of mzmine account Posteingang x

noreply@mzio.io 21:39 (vor 0 Minuten) ☆ ⓘ ↵ ⋮
an mich ▾

Someone has created an mzio GmbH account to access the mzmine software with this email address. If this was you, click the link below to verify your email address

→ [Link to e-mail address verification](#)

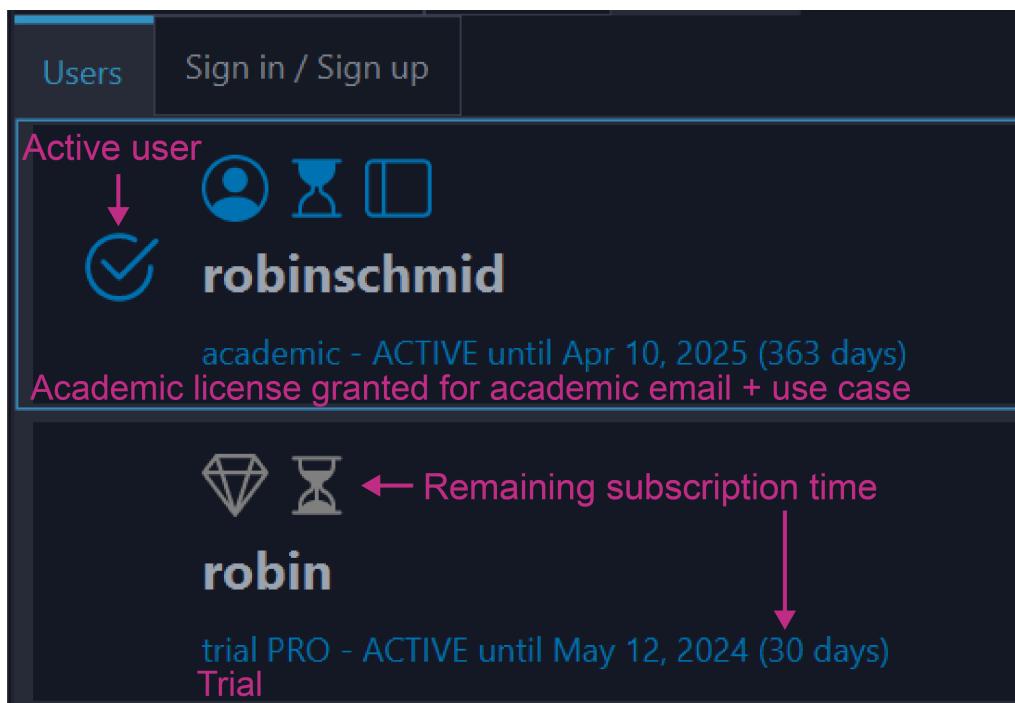
This link will expire within 5 minutes.

If you didn't create this account, just ignore this message.

mzio
mzio GmbH
Altenwall 26, 28195 Bremen, Germany
Phone: +49 151 51882184
E-Mail: info@mzio.io
Web: www.mzio.io
CEO: Dr. Ansgar Korf
Register Court: Bremen, HRB 40042 HB

This message is confidential and intended solely for the addressee. If you are not the intended recipient, any dissemination, use, review, distribution, printing, or copying of the information contained in this e-mail is strictly prohibited.

1. Please open the email
2. Click the verification link that redirects to the user portal
3. Verify the email by **Click here to proceed** to verify your email
4. Return to mzmine and refresh the page (automatically done every 30 seconds)
5. Check your license status in mzmine. Every user will receive a 30-day trial license or a non-profit academic license. Please reach out if your academic institution still needs our automatic validation.



3.6 Command-line interface (CLI)

Users can be logged in via command-line arguments to run mzmine in batch mode. See [CLI tools](#) for more information.

The most important commands on Windows are:

```
# login in browser
"C:\Program Files\mzmine\mzmine_console.exe" -login

# login in the terminal console
"C:\Program Files\mzmine\mzmine_console.exe" -login-console

# list all users
"C:\Program Files\mzmine\mzmine_console.exe" -list-users
```

3.7 Alternative Sign in / Sign up

Alternatively users can register on the mzio.io website <https://auth.mzio.io/realm/mzio/account> by clicking on the **Sign in** button in the top right corner. Then follow the regular self-registration described above. Finally, users will need to sign in to the mzmine software. If the sign in website does not load in mzmine, click the  . Set a proxy and retry with the login screen but if this fails try the alternative sign in by username and password directly. The mzmine CLI also provides the **-login-console** argument for alternative sign in.

3.8 mzmine's educational non-profit program

During your registration in mzmine, you may enter an email address of your educational institution. This will grant automatic access to our academic non-profit program if your institution is already listed with us. If you receive a trial version, please reach out to us and provide any information required to prove the educational non-profit status.

Each year, academic users will need to update their user profile, potentially changing to their current institution and new email address, and to verify their email. This will renew your academic license if your institution is still in our program. Otherwise, users are granted a trial version.

3.9 Frequently asked questions (FAQ)

1. Why is the link to verify my email address missing?

- Your email provider may remove links as a security measure. Please reach out to your postmaster or IT administrator with info@mzio.io in Cc to resolve this issue. Please provide the information that links are missing and that the domain mzio.io can be trusted.

2. Why am I not receiving any verification email?

- Your email provider may falsely block the mzio.io domain. Please reach out to your postmaster or IT administrator with info@mzio.io in Cc to resolve this issue. Please provide the information that the domain mzio.io can be trusted.

3. The login page in mzmine is not loading, what can I do?

- The proxy settings and alternative sign in page may open automatically or is available by clicking on the tool button  on the *Sign in/Sign up* page.
- Check your computers proxy settings and apply them on this page and press update proxy to retry.
- Otherwise, try the alternative sign in by username and password. This requires an existing mzmine user that can also be registered on the website.

4. My academic account expires soon, how to renew my account?

- Once the license is expired, on your next login you will need to update your user details and revalidate your academic email address. If successful this will grant another year in our academic non-profit program.
- During your profile update, you will have the chance to change to your new institution and email address.

 November 25, 2025 10:09:50

 Ansgar Korf, Robin Schmid, Steffen Heuckeroth

4. Learners corner

4.1 Useful channels

- mzio YouTube channel
- mzmine project
- Robin Schmid
- Functional Metabolomics Lab with Daniel Petras
- Dorresteijn lab

4.2 mzmine videos

- mzmine at VMOL webinars
- mzmine at CMFI webinars
- mzmine at 2nd international summer school on non-targeted metabolomics 2022
- Coding with mzmine

4.3 General Mass Spectrometry

- Mass spectrometry introduction by Corinna Brungs, Summer School Copenhagen, 2023

4.4 (Computational) Metabolomics

- 3rd international summer school on non-target metabolomics (clinical focus), 2023
- 2nd international summer school on non-targeted metabolomics, 2022
- CMFI seminars
- GNPS workshops
- Lab tutorials by functional metabolomics lab

4.5 SIRIUS

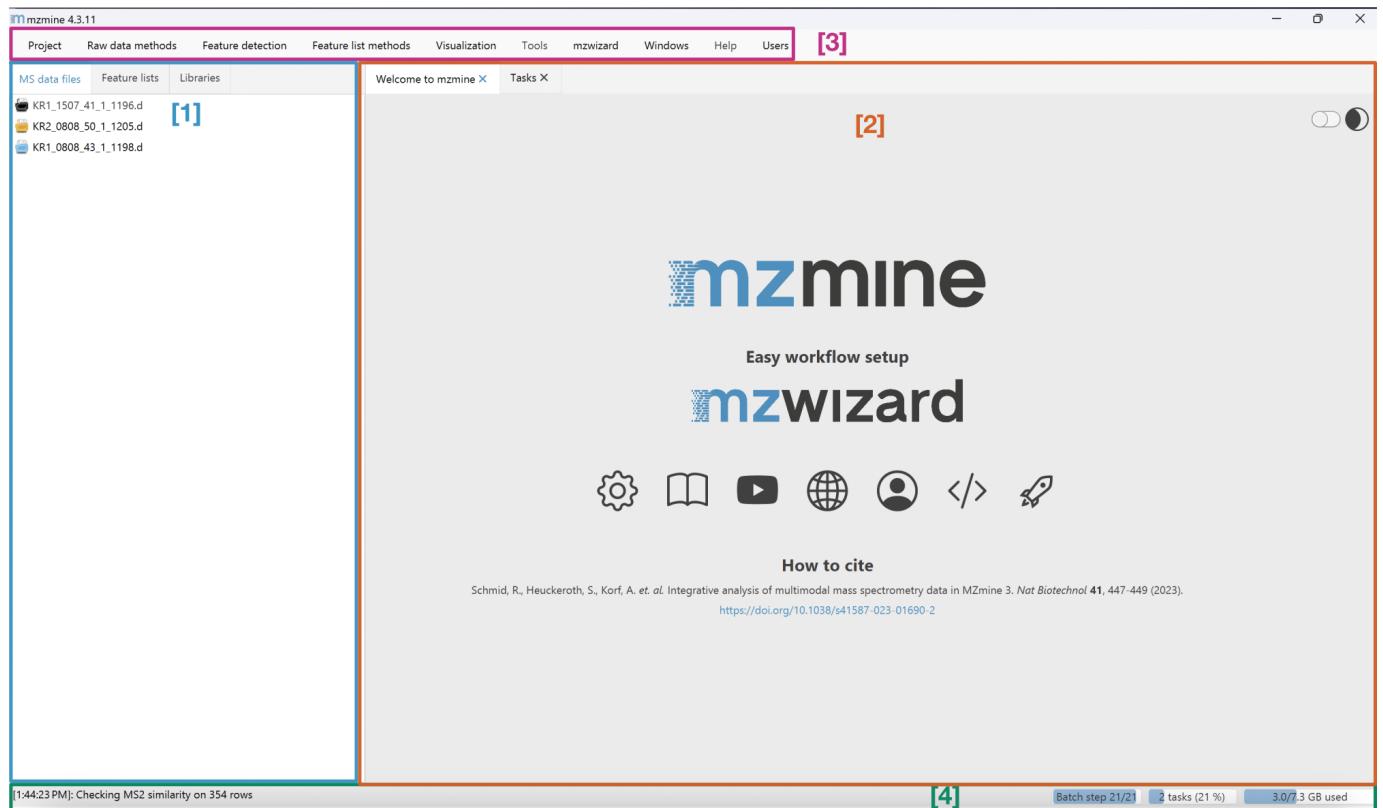
- SIRIUS 5

 December 6, 2024 16:39:03

 Robin Schmid

5. Main window overview

The mzmine main window is made up of mainly four important building blocks.



5.1 MS data files and feature lists tab

[1]: The (raw) ms data, Feature list, and Libraries tabs. Here you can find your imported raw data files, processed feature lists, and imported spectral libraries.

Tip

You can also import files by dragging & dropping them to the ms data tab.

5.2 Main content pane

[2]: Landing page and main content pane. Visualisations such as a raw data overview or a feature list can be viewed here. This pane can also contain multiple tabs. Every tab can also be opened in a new separate window by right-clicking on the header.

Tip

The landing page has many short cuts to helpful tools e.g. the mzwizard.

5.3 Main menu

[3]: The main menu. Here you can find methods to import and process your data files and feature lists and visualise the results. Projects can be saved and preferences can be set. Furthermore, the user management as well as the documentation can be accessed.

5.4 Task overview

[4]: The progress of the current tasks and batch are displayed and the latest logged activity is displayed. A detailed task overview can be found in the task tab in the main content pane.

5.5 Page Contributors

Ansgar Korf, Steffen Heuckeroth

 October 8, 2024 16:27:00

 Ansgar Korf, Steffen Heuckeroth

6. Processing wizard

mzwizard

The processing wizard reduces the configuration of complex standardized workflows into few parameters. Those values are used to estimate or calculate all other parameters from spectral processing over feature detection and alignment to annotation and data export. The mzwizard organizes the different parts of the instrumental setup to define a workflow split up into: Sample introduction, IMS, MS instrument, and workflow. Ideas for a workflow? We are open for ideas and contributions. Feel free to reach out if your workflow might be a candidate for a wizard setup.

Processing wizard



This workflow produces **Feature lists** as their result. Feature lists are a central part in visualizing the results in the **Statistics dashboard** and in the **Interactive network visualizer**.

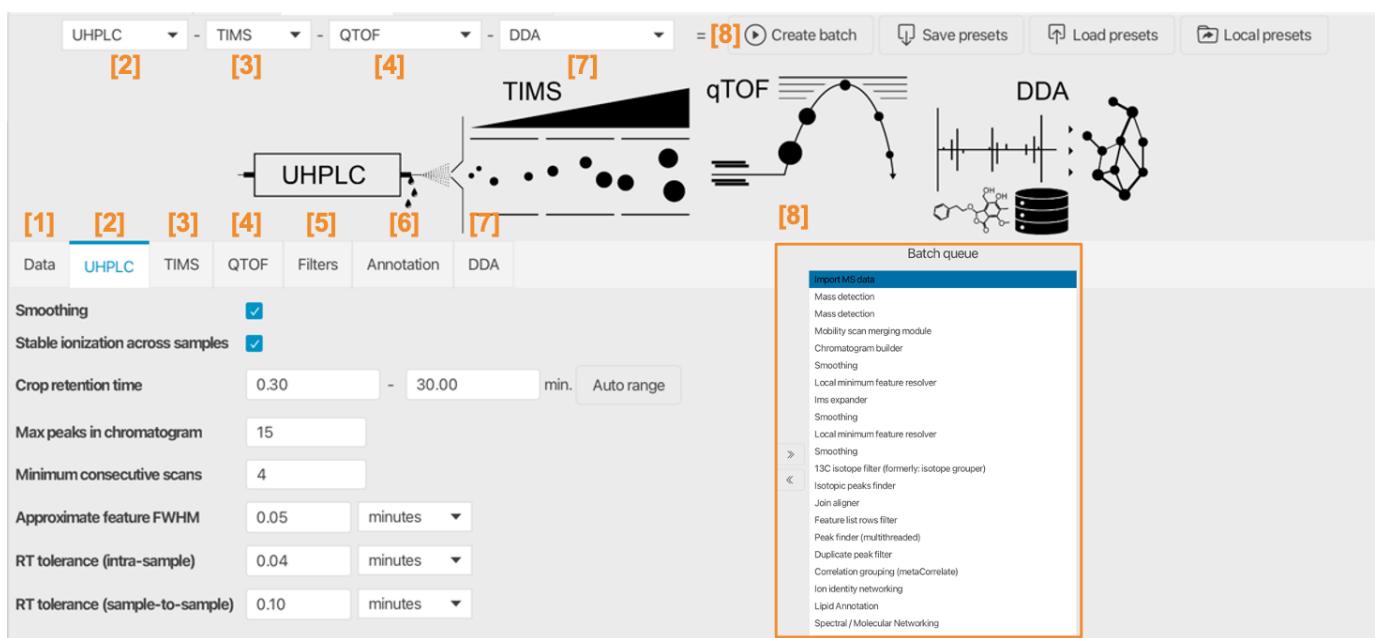


The mzwizard is split into sections, most importantly, the MS part and the sample introduction. Parameters here reflect the sensitivity, resolution, and accuracy of these parts of the hyphenation. Selecting one of the default presets actually populates the initial values.



The defaults are only suggestions and different acquisition methods and instruments produce different noise levels etc. The **raw data overview** and **aligned feature lists** are useful to optimize these parameters.

6.1 mzwizard parameter tabs



6.2 [1] Data import

Specify all data files that need to be processed. You can drag & drop the data files into the text field.

6.3 [2] Sample introduction system



Depends on the selected sampling system, e.g. **MALDI**, **HPLC**, **DESI**

6.3.1 Chromatography-based

Smoothing

Apply smoothing to the chromatograms. Smoothing should not be used for data low MS1 acquisition rates (e.g. below 5 data points per chromatographic peak/feature).

Influences batch steps: [Smoothing](#)

Stable ionization across samples

Used during feature grouping of adducts and other ions of the same molecule. Only use if the matrix (e.g., salt content, same culture medium, only fresh OR sea water samples) and ionization efficiencies are comparable across the whole study.

Influences batch steps: [Correlation grouping](#)

Crop retention time

Crops the chromatograms at these retention time values. Useful to cut off the start and the end of the chromatograms. The start is often diverted into the waist and the end often contains the cleanup procedure.

Influences batch steps: [Chromatogram building](#) [Local minimum feature resolver](#) (Calculation of [chromatographic threshold](#))

Max peaks in chromatogram

An estimate of the number of isomers and isobaric ions in the chromatograms. Used to estimate the percentage of data points that hold useful data for the feature resolving step. (Chromatographic threshold in local minimum resolver).

Influences batch steps: [Local minimum feature resolver](#) (Calculation of [chromatographic threshold](#))

Minimum consecutive scans

Only keep chromatograms and features with at least X data points in retention time dimension.

Influences batch steps: [Local minimum feature resolver](#), [Smoothing](#), [Gap filling](#)

Approximate feature FWHM

The full-width at half maximum of peaks in retention time dimension. Best extracted from the feature tables of already processed test datasets or from the raw data overview.

Influences batch steps: [Local minimum feature resolver](#), ([Minimum search range](#))

RT tolerance (intra-sample)

Retention time tolerance to group adducts and isotopes of the same molecule. The comparison is performed within each individual sample, usually leading to small variance.

Influences batch steps: [Isotope grouping](#), [Correlation grouping](#)

RT tolerance (sample-to-sample)

Retention time tolerance to align features across all samples. Dependent on retention time shifts.

Influences batch steps: [Join Aligner](#), [Gap filling](#), [Local compound database search](#)

6.3.2 Imaging based



Supported techniques: **MALDI, SIMS, DESI, LDI**

Minimum number of pixels

The minimum number of pixels containing the same m/z. Influences batch steps: [Image Builder](#)

6.4 [3] Ion mobility instrument

Smoothing

Applies smoothing to the ion mobilograms. This should be enabled for long ramp times in TIMS instruments.

Influences batch steps: [Smoothing](#)

Minimum consecutive scans

The number of consecutive scans/datapoints in a valid ion mobilogram and feature. Increase this for longer ramps in TIMS instruments.

Influences batch steps: [Smoothing \(mobility dimension\)](#), [Local minimum resolver \(mobility dimension\)](#)

Approximate feature FWHM

The full-width at half maximum for IMS features with regard to the IMS dimension.

Influences batch steps: [Local minimum resolver \(mobility dimension\)](#)

6.5 [4] MS instrument, e.g., Orbitrap, QTOF, FTICR

Noise threshold (MS1/MS2)

Choose the mass detector from the drop down menu. Choose the **Factor of lowest signal** for both centred or profile data where each spectrum contains noise signals, often represented by many signals at the same low intensity (spectral grass). This may correspond to static noise or single counts. If the spectra are free of noise, for example because they were already filtered, use an **absolute intensity** threshold.

Depending on the selected mass detector, separate absolute noise levels or factors are defined to threshold spectra of MS level 1 and 2 (or above). So the MS2 noise level is used for MSn data with level > 1. These parameters can be optimized by looking at the spectral raw data in the raw data overview.



We recommend the **Factor of lowest signal** setting for Orbitrap instruments and the **Absolute intensity** setting for TOF instruments.

Influences batch steps: [Mass detection \(MS1, MSn\)](#) 1/5th of the value is used for [ion mobility scans](#).

Minimum feature height

The minimum height of chromatograms and features.

Influences batch steps: [Chromatogram builder](#), [Local minimum feature resolver \(RT dimension\)](#), [Local minimum feature resolver \(IMS dimension\)](#), [DIA MS2 grouping \(todo documentation\)](#), [RT Calibration](#)

Scan-to-scan m/z tolerance

Relative and absolute m/z tolerance. Always applies the maximum tolerance based on the m/z this means that smaller and higher values are stronger affected by the absolute and relative tolerance, respectively. Used to find the same signal in different scans when connecting chromatograms. Reflects on the mass accuracy between scans in the same raw data file.

Influences batch steps: [Mobility scan merging](#), [Chromatogram builder](#), [Gap filling](#), [Lipid annotation \(MS2\)](#), [Spectral library search](#), [Spectral/ Molecular Networking](#), [DIA MS2 grouping \(todo documentation\)](#)

Feature-to-feature m/z tolerance

Relative and absolute m/z tolerance. Always applies the maximum tolerance based on the m/z this means that smaller and higher values are stronger affected by the absolute and relative tolerance, respectively. Used to group isotopes and adducts of the same molecule. Those m/z values are already averaged over their features and should have lower m/z differences than the scan to scan tolerance.

Influences batch steps: [Isotope filter](#) [Isotope finder](#) [Ion identity networking](#) [Duplicate filter](#)

Sample-to-sample m/z tolerance

Relative and absolute m/z tolerance. Always applies the maximum tolerance based on the m/z this means that smaller and higher values are stronger affected by the absolute and relative tolerance, respectively. Used to align features across samples. Those m/z values are already averaged over their features but originate from different samples.

Influences batch steps: [Join Aligner Local compound database search Lipid annotation \(MS1\)](#)

6.6 [5] Filters

Original feature list

Options to either *keep* or *remove* the original feature lists. *Keep* is valuable during workflow optimization whereas *remove* saves resources and allows for more performant processing or large datasets. (See section on performance).

Influences batch steps: all batch steps that create a new feature list.

Min samples per aligned feature

Only keep aligned features that were detected in at least n samples. This parameter should usually scale with the sampleset size and if samples are relatively similar from their compounds. Uses the maximum of an absolute and relative value.

Influences batch steps: [Feature list rows filter](#)

Only keep features with 13C

Detect isotope pattern and only keep feature with valid 13C isotope pattern.

Influences batch steps: [Feature list rows filter](#)

6.7 [6] Annotation

Local compound database

- Specify the database file (csv or tsv format).
- Select precursor m/z option to either
- calculate from given neutral mass (or formula/structure)
- use provided precursor m/z in column
- Filenamne column is only used for the library generation workflow
- Columns map the table column headers to the internal names in mzmine
- Select data types present in local compound database

Annotate lipids

Select if you want to annotate lipids. Lipid database scope can be defined in the [Lipid annotation](#).

Spectral library files

Select all spectral libraries to import and use during spectral library matching to annotate compounds in the final aligned feature list.

Influences batch steps: [Spectral library search](#)

Influences batch steps: [Local compound database](#)

6.8 [7] Workflows

Most of the parameters in the workflow section define data output and some workflow specific parameters.

6.8.1 General parameters

- Apply [spectral networking](#) (FBMN/IIMN): Will compare all MS2 spectra across features to form molecular networks by spectral similarity.
- Define an export path and base file name, e.g., "D:\analysis\date_project" this will create a new folder and save all files from export modules there. Each module will add a specific suffix to the file name.
- Export for molecular networking (e.g., GNPS, FBMN, IIMN, MetGem): Will export all files for molecular networking
- Export for SIRIUS: Will export all files needed for SIRIUS
- Export annotation graphics: Exports annotations like spectral library matches, lipid matches, etc. to graphical reports. Contains options to also export chromatographic/ion mobility shapes, images, and other plots.

6.8.2 Workflow parameters and descriptions

DDA

The data-dependent acquisition workflow is the default non target workflow. We recommend to also use this workflow for targeted analysis and combine it with the [Local compound database search](#) or [Spectral library search](#) annotation modules.

DIA

The data-independent acquisition workflow is can process any type of MSe, broad band CID, or all ion fragmentation experiments. Add a minimum DIA correlation coefficient and a minimum number correlated data points in of MS2 data points.

Library generation

More method and contributor **metadata** is required to build spectral libraries. This workflow produces reference libraries with options to filter and merge spectra.

Imaging



This workflow is only available for the sampling techniques: **MALDI, LDI, DESI, SIMS**

Select if you want to co-localize images.

Target plate



This workflow is only available for the sampling technique: **MALDI** (Bruker timsTOF only)

Select if you want to co-localize images.

Spectral deconvolution



This workflow is only available for the sampling technique: **GC-EI**

Select the minimum number of signals in a deconvoluted spectrum

6.9 [8] Create Batch

Select on the "Create batch" button to open a batch queue. Each step in the batch queue can be modified via double click. Find more information how to modify a [batch queue](#). Click on "OK" to start the processing.

The screenshot shows the 'Batch queue' dialog box. At the top right, there is a 'Batch mode' button. The left side is labeled 'Batch queue' and contains a tree view of modules:

- Project**
 - Project I/O
 - Open project
 - Save project
 - Save project as
 - Project
 - Clear project
 - Project metadata
 - Project metadata import
- Spectral data**
 - Raw data import
 - Import MS data
 - Bruker TDF file import
 - iMZML file import
 - mzData file import
 - mzML file import via MSdk
 - mzXML file import
 - netCDF file import
 - Thermo RAW file import
 - ZIP/gZIP file import
 - Import spectral library
 - Spectral library to feature list
 - Raw data methods

Below the tree view is a search bar with a clear button (X) and a placeholder 'Search'.

The right side is labeled 'Batch queue' and lists a series of steps:

- Import MS data
- Mass detection
- Mass detection
- Mobility scan merging module
- Chromatogram builder
- Smoothing
- Local minimum feature resolver
- Ims expander
- Smoothing
- Local minimum feature resolver
- Smoothing
- 13C isotope filter (formerly: isotope grouper)
- Isotopic peaks finder
- Join aligner
- Feature list rows filter
- Peak finder (multithreaded)
- Duplicate peak filter
- Correlation grouping (metaCorrelate)
- Ion identity networking
- Lipid Annotation
- Spectral / Molecular Networking

At the bottom of the dialog are several buttons: 'Set parameters', 'Clear', 'Redefine outputs', 'Load', 'Save', 'Load last...', 'Intermediate feature lists' (with a 'REMOVE' button), 'Apply to batch', 'Cancel', 'OK', and 'Check'.

6.10 Optimizing the wizard workflow

While the presets of the processing wizard will produce good results out of the box for many instrument types and combinations, it may be worth optimizing specific workflow steps to achieve even better results. There are two ways to optimize the wizard workflow.

1. By optimizing the parameters in the wizard steps.
2. By optimizing individual steps in the [created batch](#).

A step-by-step approach is detailed in the [workflow optimization guide](#).

Ansgar Korf, Robin Schmid, Steffen Heuckeroth, Steffen Heuckeroth, corinnabrungs

 October 21, 2025 15:45:16

 Ansgar Korf, Robin Schmid, Steffen Heuckeroth, Steffen Heuckeroth, corinnabrungs

7. Data handling (supported formats)

mzmine supports both **open** (e.g., .mzML, .mzXML, .imzML, .netCDF) and **proprietary** formats from Bruker Daltonics (.d and .tdf/tsf). Raw data files from vendors must be converted into an open format prior to the import. **This conversion can be applied automatically during the import, if the user has MSConvert installed.** If you want to convert the files yourself, see the sections below.

The **recommendations** for the data handling are the conversion of the raw data to centroided .mzML data files, **except** for timsTOF data (native .tdf and .tsf inside the Bruker .d folder), and the conversion of MS imaging data to .imzML, except for the timsTOF fleX MS imaging data.

7.1 Data conversion to open formats (.mzML / .imzML)

7.1.1 MSConvert (ProteoWizard) to mzML

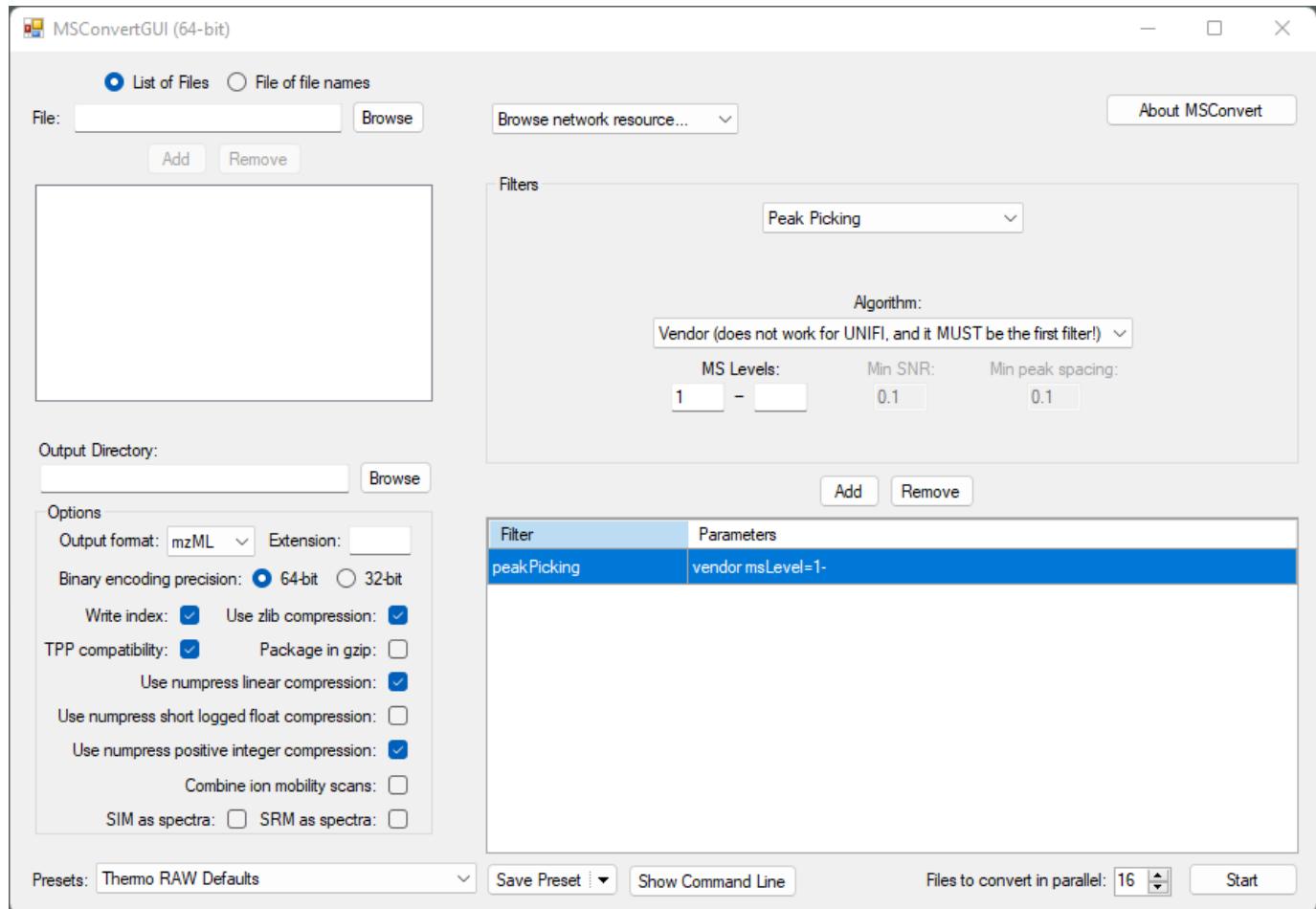
Info

mzmine can use MSConvert automatically. Make sure to setup the MSConvert installation path in the mzmine preferences. (only supported on Windows)

MSConvert supports the conversion of AB SCIEX, Agilent, Bruker, Shimadzu, Thermo Scientific, and [Waters](#) raw data. More information about the formats can be found in the [ProteoWizard Documentation for Users](#). Furthermore, profile data can be centroided to reduce the file size and memory consumption, which can be important when processing larger datasets.

Recommended conversion settings for high-resolution MS data to mzML (newer than mzXML). Use **64 bit ** for m/z to retain accuracy**. Compression is optional** but reduces the size significantly without much information loss - this might be instrument dependent though.

The **peak picking** filter needs to be the first in the list - otherwise MSConvert might default to another algorithm than the selected vendor-specific one. **Beware** that previous versions of MSConvert had the titleMaker filter at the first position and the peak picking algorithm was added manually and the position needed to be switched accordingly to peak picking first.



7.1.2 MSConvert command line interface

Example Microsoft windows command to convert all .raw to .mzML. Can be in a .bat-batchfile located together with the raw data files. By default, the *m/z* is encoded in **64 bit**, whereas the intensity is encoded in **32 bit**. NumpressPic compresses the intensity, whereas the numpressLinear compresses the *m/z* and RT and a [toler] can be set. The peakPicking can be set to specific MS levels (here MS1- for multistage fragmentation)

```
"C:\Users\**USERNAME**\AppData\Local\Apps\ProteoWizard 3.0.22166.28b1b7b 64-bit\msconvert.exe" *.raw --filter "peakPicking true 1-" --zlib --numpressPic --
numpresslinear -v -o mzml
pause 1
```

More details can be found in the [ProteoWizard documentation](#).

7.1.3 ThermoRawFileParser

It is used to convert ThermoFisher .raw files into .mgf, .mzML, .parquet. This converter is important if an internal calibrant was used (e.g., EASY-IC). This mass is excluded in the FreeStyle view, whereas MSConvert remains all signals in the mzML, including the calibrant. If those masses together with some flagged signals by Thermo, should be removed use this converter with the option --excludeExceptionData.

Note

mzmine can use the ThermoRawFileParser automatically to import your data without conversion. In the preferences (CTRL+P) set the "Thermo data import" to "Thermo raw file parser" instead of MSConvert. The raw file parser is supported on Mac, Linux, and Windows.

Example for command line interface with the exclusion of exception data:

```
"C:\Users\**USERNAME**\AppData\Local\Apps\ThermoRawFileParser\ThermoRawFileParser.exe" *.raw -d=INPUT_DIRECTORY -o=".\\mzml_ThermoRawFileParser_excl_data/" --excludeExceptionData
pause 1
```

More details can be found on [GitHub](#).

Alternatively, you can use the ThermoRawFileParserGUI to convert Thermo files to mzml without having to use the CLI: <https://github.com/compomics/ThermoRawFileParserGUI>

7.1.4 Bruker

Bruker data is supported in .tdf, .tsf, and .baf format without prior conversion. Calibrations for .tdf and .tsf are directly imported if they are present.

Recalibrated baf files

For Bruker recalibrated data in .baf format, it is recommended to perform **export analysis to mzXML** using the Bruker's Data Analysis software (for automation, apply Bruker's processing script during data acquisition) prior to importing to mzmine. Compared to the mzML and mzData formats, the mzXML format best retains the full scan definition and sample recalibration properties. In mzML the MS1 scans were recalibrated but the MS2 scan definition of the precursor *m/z* did stay unchanged, leading to differences between MS1 and MS2.

The same problem is observed with Waters data. The solution is to use a precursor *m/z* values corrector by applying a script on .mzML or .mzXML files after conversion. The script and the manual how to use it can be found [here](#).

7.1.5 Waters

Waters recently released a tool called **Waters data connect**, which allows conversion of DDA, DIA, and HD-DDA data to mzML. Lock mass correction is applied during the conversion. We also recommend to enable centroiding (2D peak picking).

The tool is available for [UNIFI](#) and [MassLynx](#). Additional information can be found [here](#)

Alternatively, you can try to import the .raw file directly into mzmine, if you have MSConvert installed on your computer.

7.2 Data sharing (FAIR)

Nowadays, modern scientific knowledge requires the comprehensive description and accessibility of experimental and computational approaches used in a study. Therefore, the results should be **Findable, Accessible, Interoperable, and Reusable (FAIR)**. This is critical in support of both the initial reviewing of a research article, and to ensure its reproducibility and impact. When publishing, the following information should be added in the method/data availability section:

- Input files

Provide the input files that were used for the study by using the deposition number given by the data repository (see Data repositories). Make sure to provide the raw data file and if used the converted (and centroided) format (both can be stored in the same repository). + Metadata table file

This file should describe the list of input files and their respective experimental information. Best if already stored within the data repository.

- mzmine Version

The mzmine version, for example mzmine 4.3.0, that was used for processing the data and describes the major.minor.patch version. Stable releases are provided through GitHub permanently. In contrast, the mzmine development versions are not permanently available. When using such a version for a publication either make sure the results are the same with the previous or next release, or upload and provide a link to the version on a permanent storage. + mzmine .mzbatch file

This batch contains all parameters of the processing. Starting with MZmine 3.4.0, batch files also contain information on the version that created the file. However, the version should be still denoted in the manuscript. This batch can be created a posteriori from any feature list by **right click → Feature list information → Show batch**. This batch file contains all steps that were used to create this specific feature list. + Result files

These files, e.g., feature lists or .mgf files, can be used for direct statistical (re-)analysis and spectral (re-)annotations. + mzmine project file

Although this file can be very large, sharing it remains the best approach to make your data 100% FAIR.

7.3 Data repositories

Data files should be uploaded to a public mass spectral data repository. Popular repositories are [GNPS/MassIVE](#), [MetaboLights](#), and [MetabolomicsWorkbench](#). All detailed information about the submission requirements and needed information can be found on each webpage. If possible and used for data processing, upload both the original raw and the converted (centroided) data files. The converted (open format) file enables the easy integration in other tools, while the original raw data might contain metadata and profile data that remains uncovered by the converted format. Additionally, a metadata file should be added, containing information about the input files and their experimental information.

7.4 Page Contributors

Ansgar Korf, Robin Schmid, Steffen Heuckeroth, corinnabrungs, elnurgar, li-chin

 September 8, 2025 09:46:32

 Ansgar Korf, Robin Schmid, Steffen Heuckeroth, corinnabrungs, elnurgar, li-chin

8. Performance & Benchmarking

8.1 General remarks

This section contains information on how to tune mzmine for different systems. Generally these points can improve the throughput and memory constraints:

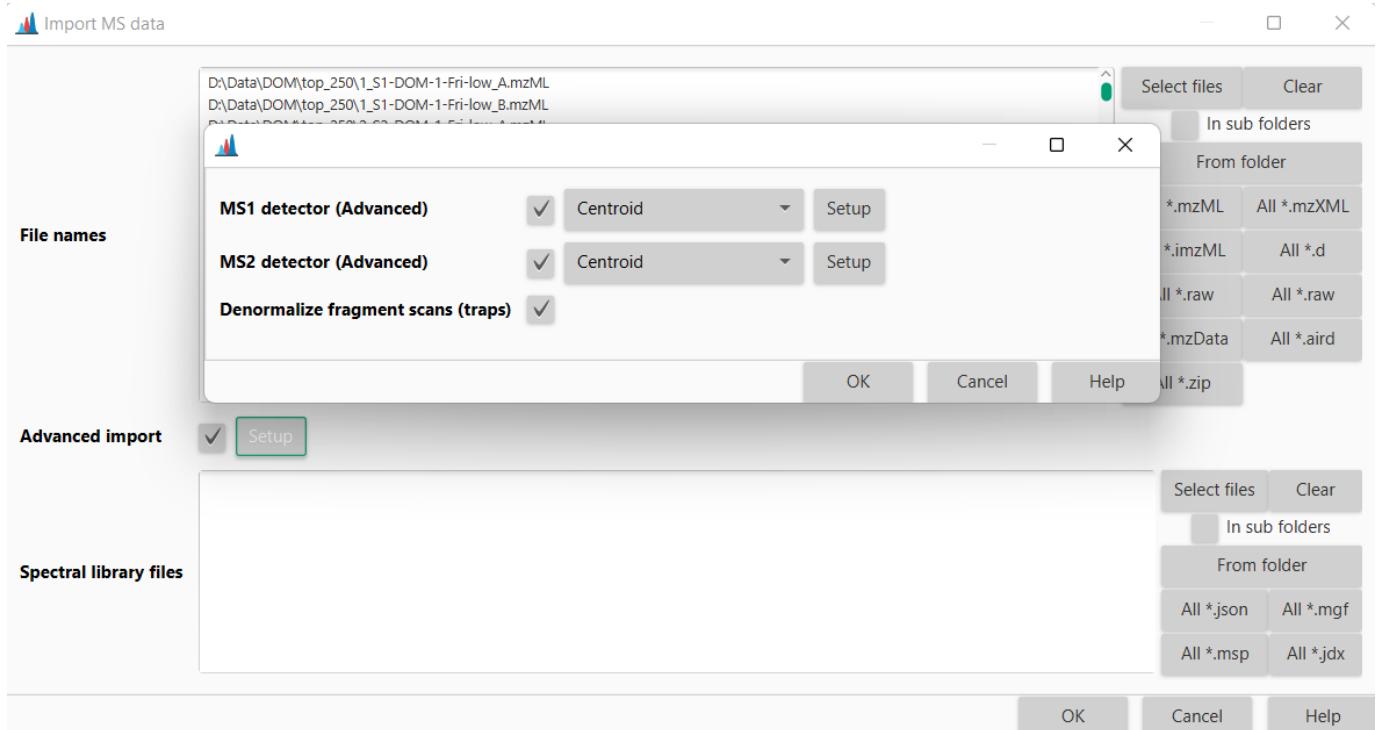
- Check the latest **Processing Wizard** workflows that may contain new modules and options to speed up and enrich your analysis. You may want to remove unnecessary steps.
- Make sure to set the **temporary** files directory to a **fast local drive**, e.g., SSD
- **Convert files** to centroided open formats, e.g., mzML
- Run mzmine in **headless mode** as a **command-line tool** to avoid the graphical-user interface (GUI). Just provide an .mzbatch file and process on any computer or server.
- Use the **advance data import** to reduce the amount of data that needs to be loaded. This will apply mass detection during the import.
- Optimize your mzmine workflow in the batch mode, considering your data and study design:
- Increase minimum feature height, noise levels, and other feature constraints. If a feature was detected in at least one sample - gap-filling will try to fill the gaps of missed features later.
- Use **Join aligner** and **Peak finder (multithreaded)** gap-filling for large datasets. Both modules are optimized for performance.
- Apply **Rows filter** between alignment and gap-filling with a minimum number of samples filter: "How many samples do I need for a significant feature?"
- Set all **Handling original feature lists** parameters to *PROCESS IN PLACE* or *REMOVE*
- Upgrade the **RAM** of your system or run on a server

8.1.1 Tuning modules

Advanced data import

The advanced data import provides the option to apply mass detection during import and only loading those filtered mass spectra. This is a great way to reduce the need for memory mapping of raw spectra that are not used in most workflows (most use the masslists detected by Mass detection step as a filtered centroid version of spectra). This option is great for large datasets and for profile mode data.

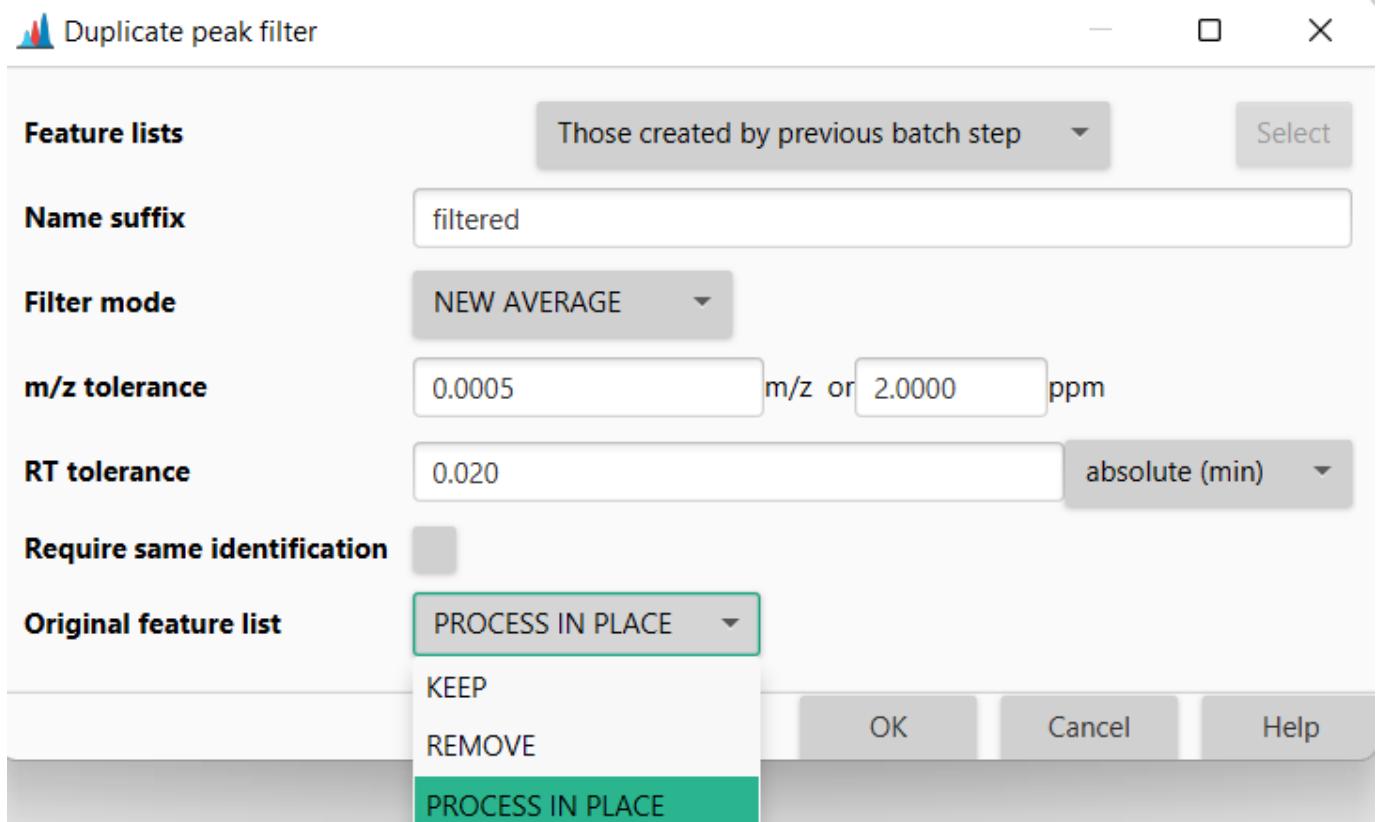
1. Open the Import MS data parameters
2. Activate Advanced import and press Setup
3. Define the MS1 and MS2 detectors as described in the Mass detection step
4. Remove all other Mass detection steps from the batch configuration, usually applied after data import



Handling original feature lists

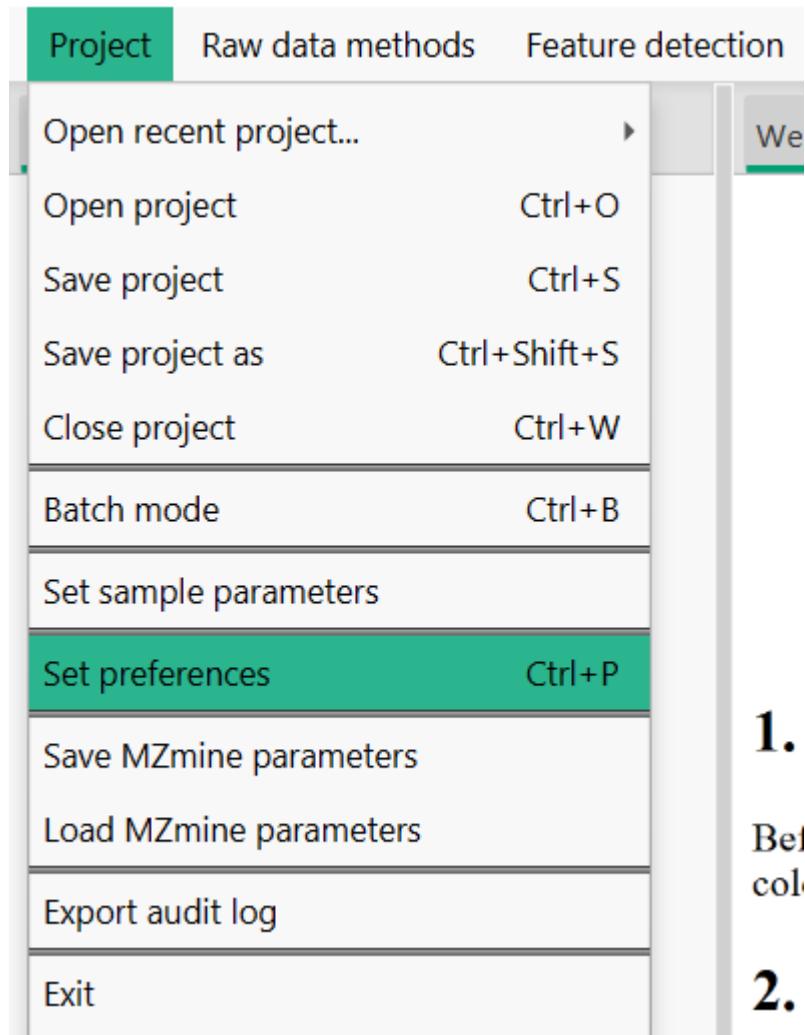
Most processing steps in mzmine create new feature lists and have options to handle the original lists.

- **KEEP:** Use during optimization of the workflow. All feature lists are kept for comparison purposes and to backtrack issues.
- **REMOVE:** Remove the original feature lists. Better for performance and memory constraints on large datasets.
- **PROCESS IN PLACE:** Only available for some modules like the duplicate filter or the rows filter. Even better performance than REMOVE.



8.1.2 Preferences

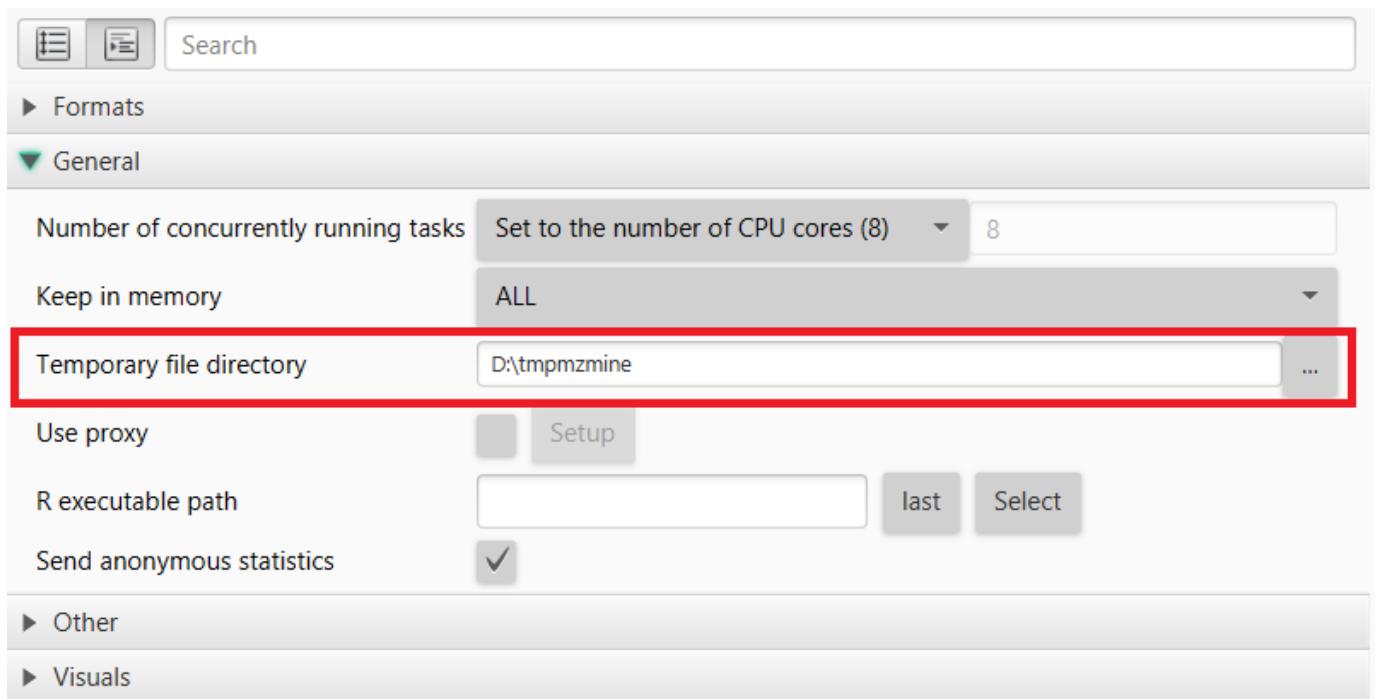
The preferences can be changed in mzmine's graphical user interface by accessing *File/Set preferences* from the menu. The choices will be stored in a (hidden) *.mzmine3.conf* file in the user's home directory (Windows: *C:\Users\USERNAME*) once mzmine is closed.



Temporary files

mzmine will create multiple temporary files at various times of the processing stage, e.g., when importing spectral data, running mass detection, or creating feature lists. These files will be stored in a folder that can be specified in the preferences.

We recommend putting this folder on an SSD drive, ideally an M.2 for the best performance. Until version < 4.3 the temporary files were deleted when mzmine was closed (Mac & Linux) or when a new session is started (Windows). With mzmine version 4.3 > temporary files are deleted as soon as possible.



Memory options



The parameter **Keep in Memory** defines what data is kept in memory (RAM) or otherwise memory mapped to the temp directory.

- Generally this setting should be *none* (**default**).
- If memory is no issue this option might be set to *all* process all spectral and feature data in memory.
- The option *masses_features* keeps centroid mass lists and features in memory while memory mapping raw spectral data.
- The option *mass_lists* will keep only mass lists in RAM, while memory mapping the raw spectral data and features.

8.1.3 Logs

Currently, the logs are written to the *mzmine_0_0.log* file in the user's home directory. Please submit your log files together with any issues on [GitHub](#).

8.1.4 Maximum memory

Relative memory limits (default)

The maximum Java heap size (the main part of the RAM available to mzmine) is set to 80%. This is usually a good value, considering that mzmine and its Java Virtual Machine (JVM) will use memory extending over this 80% threshold for specific tasks. There is one way to change the maximum heap size before starting mzmine, however, it requires administrator access.

Find the **mzmine/app/mzmine.cfg** file in the mzmine install directory or portable version. The portable version allows this file to be edited. In the installed Windows version, this file is write protected, which needs to be changed under **File/Properties/Security/** select Users and click Edit to grant write access. Now change the *MaxRAMPercentage* to grant more RAM.

Note: You'll need to modify the **mzmine_console.cfg** if you run mzmine_console (on Windows).

```
java-options=-XX:InitialRAMPercentage=10
java-options=-XX:MaxRAMPercentage=80
```

Absolute memory limits

The relative RAM percentages are the recommended best practice when tuning the JVM. This way mzmine will run well on smaller and larger systems. Alternatively, absolute memory constraints can be defined as start RAM (-Xms) and maximum RAM (-Xmx). For this, remove the two lines above from the config file and add the two below by defining the RAM, e.g., 32g for 32 GB.

```
java-options=-Xms32g
java-options=-Xmx32g
```

Container memory limits

mzmine is built on modern Java technology and fully supports containerization. It automatically respects container memory limits, making it ideal for deployment in containerized environments (Docker, Kubernetes, etc.). When running mzmine in a container, the JVM will detect and honor the container's memory constraints.

Containerization is particularly useful on systems with large amounts of RAM, as it provides a clean way to limit memory usage without modifying the application. This ensures mzmine only uses the allocated resources, preventing memory overflow issues on shared systems.

8.1.5 Limit number of cores

For HPC applications, limiting the number of threads may not be sufficient. In case the usage of the actual number of cores shall be limited, the java option

```
java-options=-XX:ActiveProcessorCount=10
```

in the **mzmine/app/MZmine.cfg** in the mzmine install directory or portable version. (see [Maximum memory](#))

Ansgar Korf, Robin Schmid, Steffen Heuckeroth, omokshyna

 December 10, 2025 07:34:08

 Ansgar Korf, Robin Schmid, Steffen Heuckeroth, omokshyna

8.2 Benchmarking

mzmine simplifies benchmarking through its batch mode. Any batch workflow running in mzmine will measure the run time and output the results in the log file as tab-separated tabular data.



Open the log file through **Help>Show log file** and scroll to the bottom after running a batch workflow to find the tabular benchmarking results.

Memory



42.0 GB RAM



Throughput (speed)

3.5 samples/min



3.9 GB RAM

71.6 samples/min

11x improved memory use

20x speed up

8.2.1 General considerations

mzmine performance tuning

mzmine is tuned for higher processing speeds. This means that if more RAM is available, mzmine may use additional RAM for faster processing. The **performance options** allow constraining the memory available to mzmine, which is a good option on larger servers with much installed RAM. The defaults should work great on most hardware by auto-scaling the used RAM.

The challenge to measure used random access memory (RAM)

Monitoring RAM usage of modern applications running on modern operating systems is challenging. mzmine brings its own **Java Virtual Machine** (JVM) that runs mzmine and continuously optimizes the performance during its runtime. Memory is dynamically managed by the JVM. mzmine will allocate more memory if available and not contested by other processes.



Windows Task Manager lacks the tools to measure the used RAM and will always show the memory managed by the JVM instead. This value mostly reflects how much memory is available to mzmine but has no real value in measuring its performance.

Garbage collection (GC) is a JVM process to reclaim memory from unused objects like intermediate processing results. By default, mzmine only runs GC when there are memory constraints. An option in the preferences enables GC after each batch processing step. This is usually only needed for more accurate measurements during benchmarking.

Manual garbage collection

The user may run **manual garbage collection** by clicking the **RAM usage bar** in the bottom right corner of the graphical user interface (GUI). After running manual GC, the memory bar will show the used memory and the logs will print a message like below:

```
Freeing unused memory
Used heap memory after manual GC: 0.26 GB
```

8.2.2 Example benchmark output

Processing time

The processing time is measured as elapsed seconds between starting and finishing each step in a batch workflow. The total time is reported as the last line in the output table.

Memory (RAM)

The **usedHeapGB** in the table below provides a readout of how much RAM was used by mzmine after finishing a batch processing step. It is only available after activating it in the preferences. See below. After running a memory benchmark batch from the GUI, it is still advised to run a manual garbage collection step to get the final used memory. This value is usually very close to the last print-out in the table, but sometimes a bit lower still. In headless command line mode the only readout is the table and there is no manual GC option.



Activate the RAM measurements in the preferences by selecting the **Free memory in batch** option. This option will run the internal memory cleanup, i.e. the JVM garbage collection (GC), after every step. This will have a small impact on the processing speed, so for benchmarking processing throughput this option is best disabled.

Table output in the log file

step	secondsToFinish	name	usedHeapGB
1	1.567	Import MS data	0.21
2	0.357	Mass detection	0.23
3	3.502	Chromatogram builder	0.32
4	0.409	Local minimum feature resolver	0.28
5	0.846	Join aligner	0.29
0	7.064	WHOLE BATCH	0.29

⌚ December 9, 2025 15:21:55

👤 Robin Schmid

9. Command-line interface (CLI) for headless batch mode

Command-line arguments offer a variety of options that generally override the corresponding parameters in the preferences. The CLI option for mzmine offers to run one batch file (-batch). This batch file can be easily generated by another script or can be a standardized batch file for an analytical method that is applied to multiple datasets.

The most convenient way to apply a batch file to other datasets is to specify the **-input** and **-output** arguments to process all import files and to redirect all outputs to another directory. See examples below for how to process all .mzML files in a folder and to redirect outputs to a new directory and base filename.

9.1 Selecting a user

Running mzmine in CLI mode requires a **user**. Either login via graphical-user interface (GUI) or via command-line argument **-login** (web browser support required) or via **-login-console**. Then locate the user file in your system USER directory under `./mzmine/users/`. The most recently active user is automatically selected for mzmine processing, but an `.mzuser` file can be passed in as command-line argument to **-user**.

9.2 Windows

For command-line mode, the `mzmine_console.exe` is preferred as it will print the logging output to the console.

An easy way to start mzmine with arguments is to create a shortcut to the `mzmine.exe` or the `mzmine_console.exe`, right-click, and define the target with additional arguments. This example runs mzmine in batch mode (headless), imports the specified batch file, overrides the memory management to **none** (which is the default), effectively using memory mapping to store and access spectral, centroid, and feature data from temporary files stored in the defined temp directory. By leaving out the *memory* or *temp* arguments, the values stored in the current *preferences* file will be used, or the default values if no *preferences* file was found.

Login to a user in command-line mode

```
# login in browser
"C:\Program Files\mzmine\mzmine_console.exe" -login

# or in the terminal
"C:\Program Files\mzmine\mzmine_console.exe" -login-console
```

Process all .mzML files in a folder by the provided batch file, redirect output files to results folder and use a base filename, various export modules add a specific suffix

```
"C:\Program Files\mzmine\mzmine_console.exe" -user "D:\user\testuser.mzuser" -batch "D:\batch\my_batch_file.mzbatch" -input "D:\Data\*.mzML" -output "D:\Results\basename"
```

Defining the temp folder, otherwise uses the one specified in local configuration

```
"C:\Program Files\mzmine\mzmine_console.exe" -user "D:\user\testuser.mzuser" -batch "D:\batch\my_batch_file.mzbatch" -temp "D:\tmpmzmine"
```

Start mzmine batch with memory mapping (DEFAULT)

```
"C:\Program Files\mzmine\mzmine_console.exe" -user "D:\user\testuser.mzuser" -batch "D:\batch\my_batch_file.mzbatch" -memory none -temp "D:\tmpmzmine"
```

Start mzmine batch on machines with enough memory (RAM) with **-memory all** this will disable memory mapping to temporary files.

```
"C:\Program Files\mzmine\mzmine_console.exe" -user "D:\user\testuser.mzuser" -batch "D:\batch\my_batch_file.mzbatch" -memory all -temp "D:\tmpmzmine"
```

9.3 macOS

If the app was installed to the *Applications* folder, run this command in the terminal to run a batch file:

```
/Applications/mzmine.app/Contents/MacOS/mzmine -user "path/to/testuser.mzuser" -batch /path/to/batch_file.mzbatch  
/Applications/mzmine.app/Contents/MacOS/mzmine -user "path/to/testuser.mzuser" -batch /path/to/batch_file.mzbatch -input "/path/to/data/*.mzML"
```

9.4 Linux

If mzmine was installed (see instructions) correctly, run the code below to start mzmine in GUI mode or CLI batch mode:

```
/opt/mzmine/bin/mzmine  
# some older versions or bad installations may be in:  
# /opt/mzmine-linux-installer/bin/mzmine  
  
# login and run batch  
/opt/mzmine/bin/mzmine -login-console -batch /path/to/batch_file.mzbatch
```

9.5 Argument table

Argument	Options (default)	Description	Available since
-h and -help	no argument	show help	
-u and -user	a path, e.g., "D:\myuser.mzuser"	Path to an .mzuser file to login a user. This file is copied to the users directory. To obtain a user file, login in the mzmine graphical user interface (GUI) or command line interface (CLI). User files can be found in your system USER directory in ./mzmine/users/ after login. A user can be used in an offline environment for a certain duration.	
-login	no argument	Use the command-line to login to a user. This will open the login website in the system internet browser, if supported, or prompts an input for the user credentials into the console. After successful login, a user file will be copied to your system USER directory in ./mzmine/users/. And the current user will be saved to the configuration. The created user file can be accessed with the -user argument on the next startup.	
-login-console	no argument	Use the command-line to login to a user. This option prompts the user for username and password. After successful login, a user file will be copied to your system USER directory in ./mzmine/users/. And the current user will be saved to the configuration. The created user file can be accessed with the -user argument on the next startup.	
-list-users	no argument	List all users available on this system. The users are read from the system USER directory ./mzmine/users/.	
-b and -batch	a path, e.g., "D:\batch.mzbatch"	Path to batch file that will be run in command-line mode.	
-i and -input	a txt file or glob pattern, e.g., "D:\Data*.mzML"	input data files. Either defined in a .txt text file with one file per line or by glob pattern matching. To match all .mzML files in a path: -i "D:\Data*.mzML"	
-l and -libraries	a txt file or glob pattern, e.g., "D:\Data*.json"	spectral library files. Either defined in a .txt text file with one file per line or by glob pattern matching. To match all .json or .mgf files in a path: -libraries "D:\Data*.json"	
-o and -output	An output path and base filename, e.g., "D:\basename"	The base file name will be extended by default module names to export all files.	
-m and -memory	none , all, features, centroids, raw, masses_features	Defines what data is kept in memory (RAM) or otherwise memory mapped to the temp directory. Generally this setting should be <i>none</i> . If memory is no issue this option might be set to <i>all</i> process all spectral and feature data in memory. The option <i>masses_features</i> keeps centroid mass lists and features in memory while memory mapping raw spectral data.	
-t and -temp	a path, e.g., "-temp D:\tmpmzmine\"	The defined directory should be on a fast drive (usually SSD > HDD > network drive) with enough free space. Local drives are usually preferred. mzmine uses memory mapping to efficiently store and	

Argument	Options (default)	Description	Available since
		access spectral and feature data. This can lead to a considerable temporary consumption of disk space. Make sure that the selected drive has enough space (maybe 20 GB + 1 GB/10 files; generously over estimated).	
-p and -pref	a path, e.g., "D:\config.mzconfig"	An mzmine configuration file in xml format. The general preferences.	
-threads	int number of threads or 'auto'	replace the preference parameter with either 'auto' to detect the number of available cores - or an int number of threads for processing.	
-ignore-parameter-warnings	no argument	When a .mzbatch file from an older mzmine version was loaded that may have an influence on the mzmine processing, the cli mode will fail by default, printing a message of the affected parameters. If this argument is set, the warnings will be ignored and the batch will be run anyway.	4.6.14

Robin Schmid, Steffen Heuckeroth, omokshyna

⌚ May 16, 2025 10:33:40

-avatar Robin Schmid, Steffen Heuckeroth, omokshyna

10. Integration to other tools

10.1 Overview

MZmine3 data can be easily exchanged with and integrated into other software and web services, allowing to perform additional downstream analysis. Multiple export modules in MZmine3 allow to transfer its results with great flexibility.

Links to the detailed documentation and specifications are provided in the table.

Tool	Doc	Applicable MZmine module(s)	Format	Description
SIRIUS	Doc	Export to SIRIUS	.mgf	Contains MS2 spectra and MS1 or correlated MS1 spectra for each feature list row
GNPS-FBMN GNPS-IIMN	FBMN IIMN	Export/submit GNPS-FBMN	.mgf _quant.csv _edges.csv	- Contains one representative MS2 spectrum per feature - The feature list as an intensity (area/height) matrix - The additional IIN edges between features
GNPS-GC-MS	Doc	Export to GNPS-GC-MS	.mgf _quant.csv	- Contains one representative correlated GC-MS1 spectrum per feature - The feature list as an intensity (area/height) matrix
MetaboAnalyst	Doc	Export to MetaboAnalyst	.csv	Metadata grouped feature intensity matrix (area/height)
R, Python general purpose export		Export to CSV (legacy) Export to CSV (modular)	.csv	Export any row (averaged) and feature (per sample) types to a tabular format. Export all columns from the feature table to a tabular format.
tidyMass	Doc	Export to CSV (legacy)	.csv	Export any row (averaged) and feature (per sample) types to a tabular format
MetDNA	Doc	GNPS-FBMN/IIMN export	.csv	Export any row (averaged) and feature (per sample) types to a tabular format
NeatMS	Doc	Export to CSV (legacy)	.csv	Export any row (averaged) and feature (per sample) types to a tabular format
NetID	Doc	Export to CSV (legacy)	.csv	Export any row (averaged) and feature (per sample) types to a tabular format
mzRAPP	Doc	Export to CSV (legacy)	.csv	Export any row (averaged) and feature (per sample) types to a tabular format
MetGem	Doc	MSP Export MGF Export	.msp .mgf	
TIMA	Doc	Export/submit GNPS-FBMN	_quant.csv _edges.csv	- The feature list as an intensity (area/height) matrix - The additional IIN edges between features
Inventa	Doc	Export/submit GNPS-FBMN / Export to Sirius	.mgf _quant.csv _edges.csv	- Contains one representative MS2 spectrum per feature - The feature list as an intensity (area/height) matrix

Tool	Doc	Applicable MZmine module(s)	Format	Description
				- The additional IIN edges between features
MEMO	Doc	Export/submit GNPS-FBMN	.mgf	- Contains one representative MS2 spectrum per feature
SLAW	Doc	Export/submit GNPS-FBMN	.mgf _quant.csv _edges.csv	- Contains one representative MS2 spectrum per feature - The feature list as an intensity (area/height) matrix - The additional IIN edges between features
FERMO	Doc	Export/submit GNPS-FBMN	.mgf _quant.csv	- Contains one representative MS2 spectrum per feature - The feature list as an intensity (area/height) matrix

10.2 Page Contributors

[Adriano Rutz](#), [Justin van der Hooft](#), [Robin Schmid](#), [Steffen Heuckeroth](#), [omokshyna](#)

 October 31, 2024 12:58:55

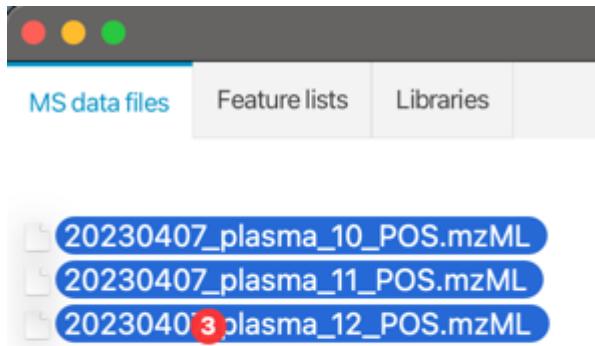
 [Adriano Rutz](#), [Justin van der Hooft](#), [Robin Schmid](#), [Steffen Heuckeroth](#), [omokshyna](#)

11. Processing modules

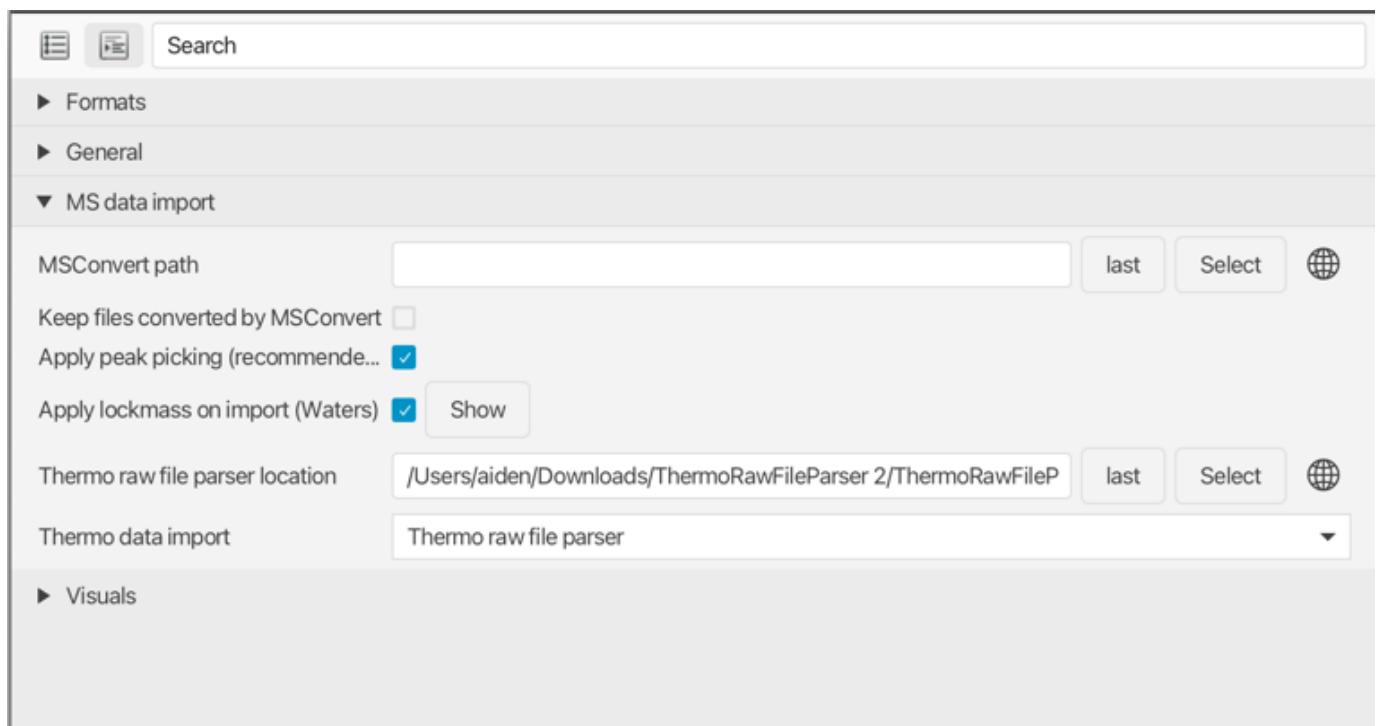
11.1 I/O

11.1.1 Data import

You can simply drag & drop the raw data into the MS data files tab of the main window. Alternatively, you can use the **Import MS data** module.



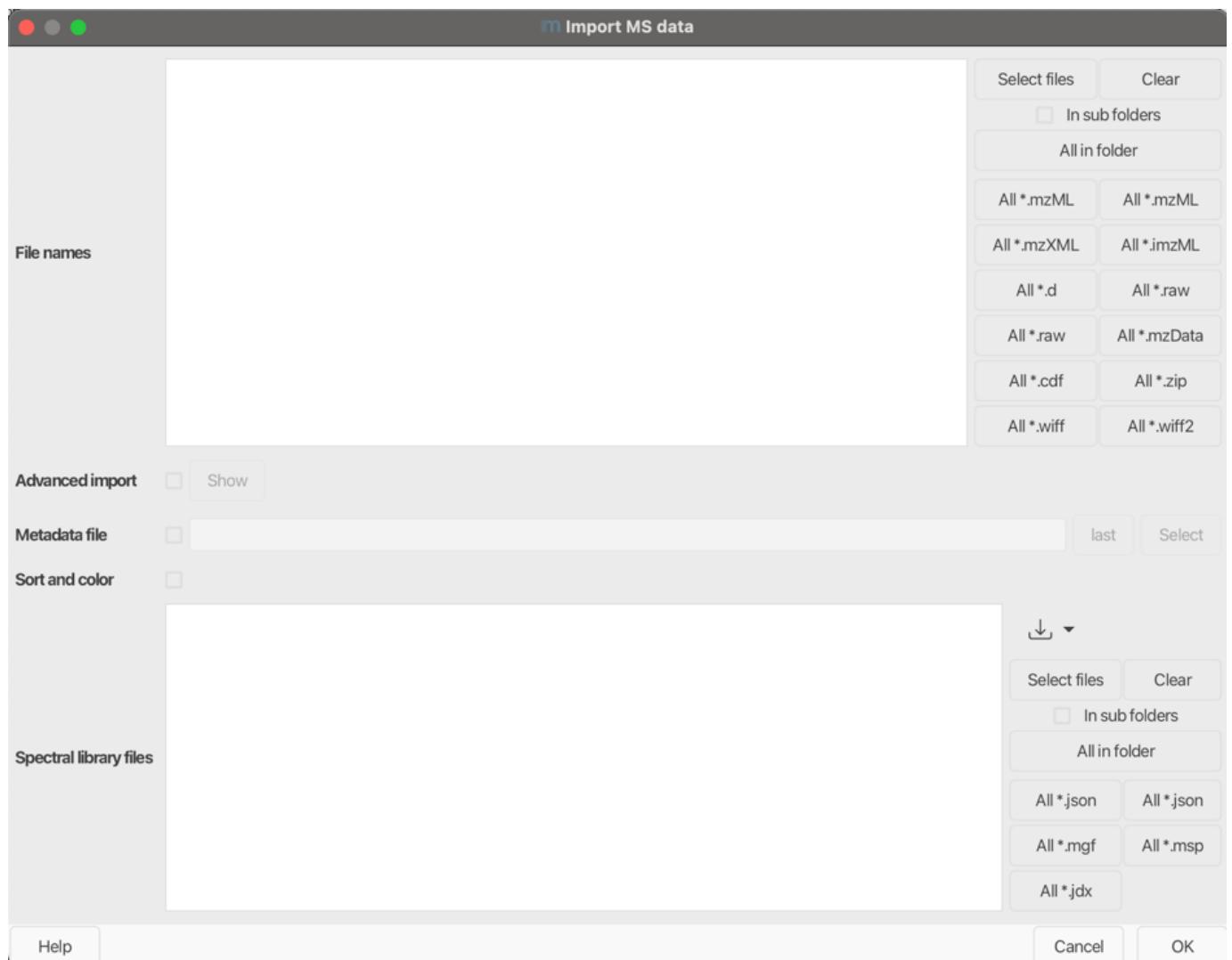
mzmine can use MSConvert for raw data conversion automatically. Make sure to setup the MSConvert installation path in the mzmine preferences.



MS data

DESCRIPTION

Raw data can be imported via [Raw data methods → Import MS data](#).



FILE NAMES

Click **Select files** or choose the preferred data format (All.) and browse the desired directory. All data files stored therein will be imported. The enabled option in sub folders* searches all files in the whole subdirectory tree.

ADVANCED IMPORT

Scan filter

You can set up the scan filter to reduce the overall data. Options include e.g. polarity, or retention time range.

Crop MS1 m/z

You can filter for a specified m/z range.

MS1 detector and MS2 detector



When processing very large datasets (hundreds to thousands of data files), the mass detection can be run during the import. All signals below the specified noise level are not imported (altering the raw data on the import), thus saving memory and processing time.

Set the mass detection parameters as described in the Mass detection section.

Warning

While it is slightly more efficient to do this during data import, only the processed data is stored and the unprocessed data is discarded. Only use this method if you have tested that the final results are the same, compared to splitting the data import and mass detection into two separate steps.

Caution

If you use a mzbatch file created by the mzwizard, make sure to remove the mass detection steps if you apply MS1 and MS2 detection during the import.

Denormalize fragment scans (traps)

This option denormalizes scans with MS level > 1, acquired with trap instruments, including Orbitrap or trapped ion mobility spectrometry (TIMS). When enabled, each scan is multiplied with the injection time resulting in "raw" intensities. If the injection time is missing in the data file, the scans remain unprocessed. This option is encouraged before spectral merging to reduce the intensity differences between scans acquired with different injection times.

METADATA FILE

Select this option, if you have a .csv or .tsv file with your metadata. If you need a template it is recommended to import all your raw data files into mzmine and then navigate to  **Project → Sample metadata**. Select the export button in the bottom toolbar. The **Format mzmine internal** is recommended.

SORT AND COLOR

Warning

This option requires metadata

If selected, the raw data will be colored based on *sample_type*, which covers **blank**, **QC**, and **sample**.

SPECTRAL LIBRARY FILES

Select all spectral library files you want to import.

[Ansgar Korf](#), [Steffen Heuckeroth](#), [corinnabrungs](#), [omokshyna](#)

 November 18, 2024 14:13:48

 [Ansgar Korf](#), [Steffen Heuckeroth](#), [corinnabrungs](#), [omokshyna](#)

11.1.2 Data export

Description

Export scans - Spectra/mass list to mgf, txt, msp and mzML export

This module exports scans or mass lists. MZmine allows several types of data export:

- to *.mzML,
- to *.netCDF,
- spectra/mass list to mgf, txt, msp and mzML export,
- MS(n) trees export.

Parameters

Raw data files

One or more raw data files that contain the scans/mass lists

Optional mass list

If checked, mass lists are exported instead of raw scans

File

The destination

Format

- mgf: MASCOT generic format - useful for SIRIUS
- txt: Plain text format
- msp: NIST search format
- mzML: Open format standard for MS data

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

11.1.3 Feature list import modules

mzTab import

DESCRIPTION

≡ Feature list methods → Import feature list → mzTab

This module imports the contents of a mzTab file into a feature list. If links to raw data files are available in the mzTab file, these can automatically be imported together with the feature list.

PARAMETERS

mzTab files

Names of the files to be imported

Import raw data files?

Optional parameter

If selected, raw data files will also be imported (if they are available).

If the raw data files are not available, empty raw data files will be generated by MZmine.



Please note that missing files may lead to issues with other modules that need access to the scan data.

mzTab-m import

DESCRIPTION

≡ Feature list methods → Import feature list → mzTab

This module is highly similar to the previous one and imports the contents of a mzTab-m file into a feature list. If links to raw data files are available in the mzTab-m file, these can automatically be imported together with the feature list.

The raw data import searches for the files in the location provided in the mzTab-m file. If the files are not present, the search continues in the same folder as that of mzTab-m file. If the files are still not found, the import module checks if the files are present in compressed form (.gz or .zip). If no files are found after that, the empty rawFiles are created in the end.

PARAMETERS

mzTab-m files

Names of mzTab-m files to be imported.

Import raw files too?

Optional parameter

If selected, raw data files will be imported if they are available. Files will be loaded from the **msrun** location defined in the mzTab-m file or from the folder containing the mzTab-m file. If the raw data files are not available, empty raw data files will be generated by MZmine.



Please note that missing files may lead to issues with other modules that need access to scan data.

CSV**DESCRIPTION****≡ Feature list methods → Import feature list → csv**

This module imports features present from a csv file.

Warning

Imported csv file should follow a specific format described below.

This kind of format can be obtained by using **xcms library** in R to detect features using Centwave algorithm on a mzXML file.

The R code returns a list of features with following columns which can be exported to a csv file to be used as an import for this module. Column Numbering starts from 0, and **0th column is ignored**.

Column Name	Column No.
peak_mz	1
mzMin	2
mzMax	3
peak_rt	4
rtMin	5
rtMax	6
maxO	9

where peak_mz - feature m/z value, peak_rt - feature retention time value, max 0 - intensity value.

Warning

To use this module, a user has to input the raw data file, which was used to generate peakList from the R code.

Example R code that generates the csv file

```
mzfile <- system.file("standards.mzXML")
library(xcms)
mzraw <- readMSData(files = mzfile, mode = "onDisk")
cpw <- CentWaveParam(ppm = 15, peakwidth = c(5,20), snthresh = 10)
mzData <- findChromPeaks(object = mzraw, param = cpw)
featurelist <- as.data.frame(xcms::chromPeaks(mzData))
write.csv(featurelist, "featureListFile.csv")
```

REFERENCES**Quote**

1. Smith CA, Want EJ, O'Maille G, Abagyan R, Siuzdak G. XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. Analytical chemistry. 2006 Feb 1;78(3):779-87. DOI: 10.1021/ac051437y

PARAMETERS**CSV**

CSV files in the described format to import

Raw data files

Raw data files that were used to generate imported feature lists.

[omokshyna](#)

 November 21, 2022 11:52:23

 [omokshyna](#)

11.1.4 Feature list export modules

This section describes general format export modules. For the description of export modules for the specific software, please, refer to [Data export to other software](#).

// # TODO

CSV

DESCRIPTION

≡ Feature list methods → Export feature list → CSV

This module allows to export obtained feature lists to csv (comma-separated values) file format.

PARAMETERS

Field separator

Character(s) used to separate fields in the exported file. "," by default.

Identification separator

Character(s) used to separate multi object columns in the exported file. ";" by default.

Remove empty columns

Optional parameter

Removes empty columns during data export.

Filter rows

Allows user to export only those rows that have associated MS/MS data or are annotated.

CSV (legacy MZmine 2)

DESCRIPTION

≡ Feature list methods → Export feature list → CSV (legacy MZmine 2)

This module is a legacy module from MZmine2 and, similarly to the previously described module, exports the contents of the feature list(s) into a csv file, which can later be processed by MS Excel or other tools. The produced CSV files cannot be imported back to MZmine. Missing values such as heights and areas of undetected features are exported as 0.

PARAMETERS

Filename

Name of file where the exported data is saved

Field separator

Columns in the new CSV file will be separated by this character (coma by default)

Export common elements

Which row elements (such as row ID, row retention time, or best ion annotation) from the feature list will be exported

Export data file elements

Which feature elements (such as feature status, name, or tailing factor) will be exported.

Export quantitation results and other information

Optional parameter

If checked, all feature-associated data will be exported.

Identification separator

Character to separate multiple identification results (if they are available) for a feature.

Filter rows

Allows user to export only those rows that have associated MS/MS data or are annotated.

Export all annotations to CSV

DESCRIPTION

≡ Feature list methods → Export feature list → Export compound annotations to csv

Export all annotations from spectral library matching, local compound DB CSV matching, and others.

PARAMETERS

Filename

Base file name of all annotation file (Use {} to fill in the feature list name when exporting multiple feature lists at once).

Top N per method

Export the top N annotations per annotation method.

Venn diagram export

Description

≡ Feature list methods → Export feature list → Venn diagram export

Exports a feature list to a csv that can be plotted as a venn diagram by other software such as [VennDis](#).

Parameters

Directory

Choose a directory to export the feature list to.

Export gap filled as detected

Optional parameter

If checked, gap filled features will be exported as detected, otherwise they will be marked as undetected.

mzTab export

DESCRIPTION

≡ Feature list methods → Export feature list → mzTab

This module exports the contents of the feature list into a mzTab file, which can later be processed by other tools.

PARAMETERS

Feature lists

Name of the feature list(s) to be exported.

Filename

Name of file where the exported data is saved.

Include all peaks*Optional parameter*

If selected, the features with unknown identity will also be exported to the mzTab file.

mzTab-m export**DESCRIPTION****≡ Feature list methods → Export feature list → mzTab-m**

This module exports the contents of the feature list into a mzTab-m file, which can later be processed by other tools.

PARAMETERS**Feature lists**

Name of the feature list(s) to be exported

Filename

Name of file where the exported data is saved

Include all peaks*Optional parameter*

If selected, the features with unknown identity will also be exported to the mzTab file.

SQL export**DESCRIPTION****≡ Feature list methods → Export feature list → SQL database**

This module allows direct export of the feature lists into MySQL or PostgreSQL database servers. If you would like to use another database then please add its JDBC driver (jar file) to the MZmine lib/ folder.

PARAMETERS**Feature lists**

Name of the feature list(s) to be exported

JDBC connection string

Identification of the database according to the JDBC standard. For example:

```
jdbc:mysql://localhost/dbname?dbname=mysql&password=mysqlpw
```

Database table

Name of the table, into which data will be exported.

Export columns

The mapping between the database table columns ("Table column") and MZmine data types ("Export data type"). For some data types an additional value ("Export value") must be specified, such as for the "Constant value" type.

Export empty feature list*Optional parameter*

If selected, an empty feature list will be exported. All columns, other than raw data file and constants, will be assigned null values.

MSP Export

DESCRIPTION

Feature list methods → Export feature list → MSP file (ADAP)

This module exports fragmentation spectra of peaks in the feature list into **MSP** format file. In addition to the spectra, the following feature characteristics are saved into the file:

- Name Name of the peak as displayed in the peak list
- DB# Row number in the peak list
- RT *Optional* Retention time of a feature
- ANOVA_P_VALUE (optional) P-value calculated by the module Data analysis -> One-way ANOVA Test
- Num Peaks Number of peaks in the spectrum

PARAMETERS

Feature lists

Name of the feature list(s) to be exported

Filename

Name of file where the exported data is saved

Add retention time

If selected, each MSP record will contain feature's retention time.

Add ANOVA p-value (if calculated)

If selected, each MSP record will contain the One-way ANOVA p-value if such p-value is present

Integer m/z

If selected, fractional m/z values will be rounded to the closest integer, and a new intensity will be calculated based on the merging mode:

- Maximum: new intensity is the maximum of merged intensities,
- Sum: new intensity is the sum of merged intensities.

MGF Export

DESCRIPTION

Feature list methods → Export feature list → MGF file (ADAP)

This module exports fragmentation spectra of features in the feature list into **MGF (Mascot Generic Format)** format file.

PARAMETERS

Feature lists

Name of the feature list(s) to be exported

Filename

Name of file where the exported data is saved

Representative m/z

Choose the representative m/z values of a cluster. The available options are: - As in feature table, - Highest m/z, - Or maximum intensity.

Fractional m/z values

Optional parameter

If selected, fractional m/z values will be saved. If not, all fractional m/z values will be rounded to the closest integer.

Warning

In case of rounding to integer, the corresponding features will be merged.

Integer m/z

When features are merged, the resulting elution profile is calculated by one of the following rules: - Maximum. In the resulting elution profile, the intensity at retention time t is the maximum of the intensities at t across the merging elution profiles. - Sum. In the resulting elution profile, the intensity at retention time t is the sum of the intensities at t for all merging elution profiles.

Export feature network to csv

DESCRIPTION

Feature list methods → Export feature list → Export feature network to csv

Export results from correlation/MS annotation as networks to csv.

PARAMETERS

Filename

Base file name of all edge files (Use {} to fill in the feature list name when exporting multiple feature lists at once).

Export row relationships

Optional parameter

Export all relationships of different rows to files. The relationships that can be exported include: - MS1 feature correlation - Ion identity network - MS2 cosine similarity - MS2 neutral loss cosine similarity - MS2 modified cosine similarity (GNPS)

Combine to one file

Optional parameter

If chosen, all output is combined into one file. Otherwise, a separate file is exported per each relationship type.

Export IIN edges

Optional parameter

Export all edges of Ion Identity Networks (IIN).

Export IIN relationship edges

Optional parameter

Export relationships between Ion Identity Networks (IIN).

Filter rows

Limit the exported rows to those with MS/MS data and/or annotated rows.

Spectral library batch generation

DESCRIPTION

Feature list methods → Export feature list → Spectral library batch generation

This module allows to generate spectral library using metadata sheet.

PARAMETERS**Feature lists**

Name of the feature list(s) to be exported.

Export file

Name of a local library file where the exported data is saved.

Export format

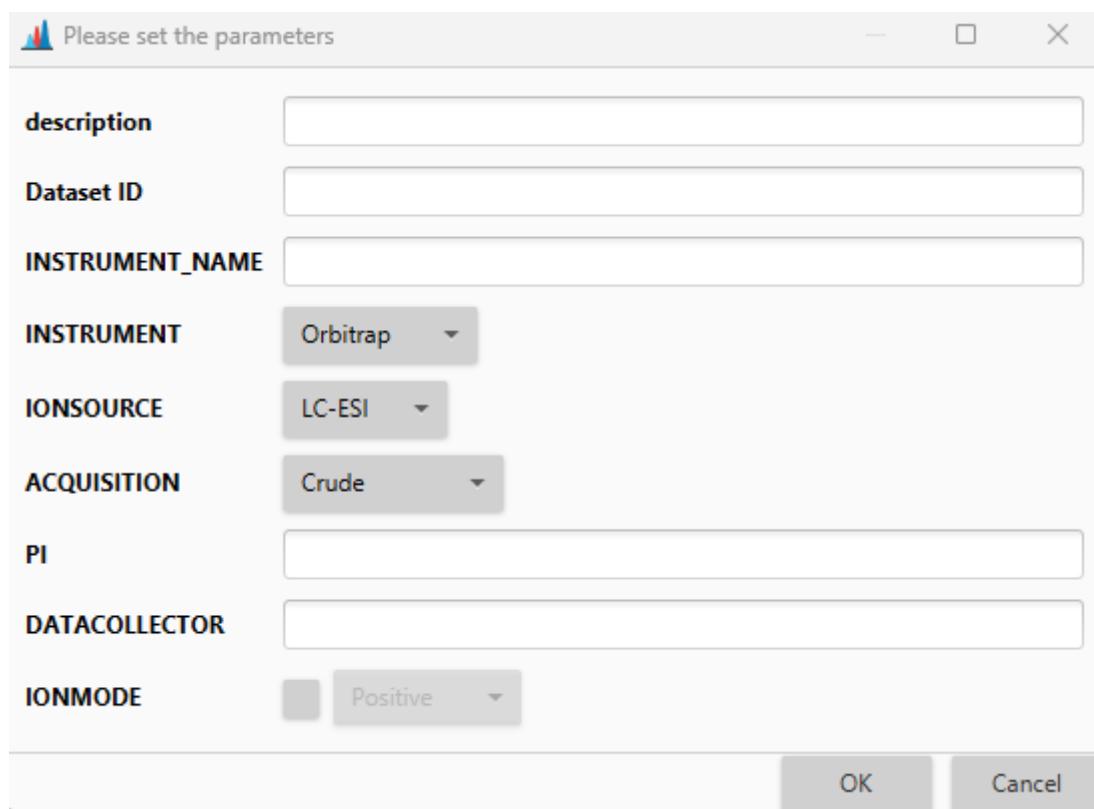
File format for the export. Available options: - MZmine json (**recommended**), - And NIST msp.

Metadata

Metadata for all entries.

Metadata setup

In this dialog user can setup all dataset-related data, including the description, ID (either from MassIVE, MetaboLights, or Metabolomic Workbench), specifications of the instrument, information on researchers involved in the dataset creation, and ion mode.

**m/z tolerance (merging)**

Optional parameter

The tolerance used to group signals during merging of spectra. If selected, spectra from different collision energies will be merged.

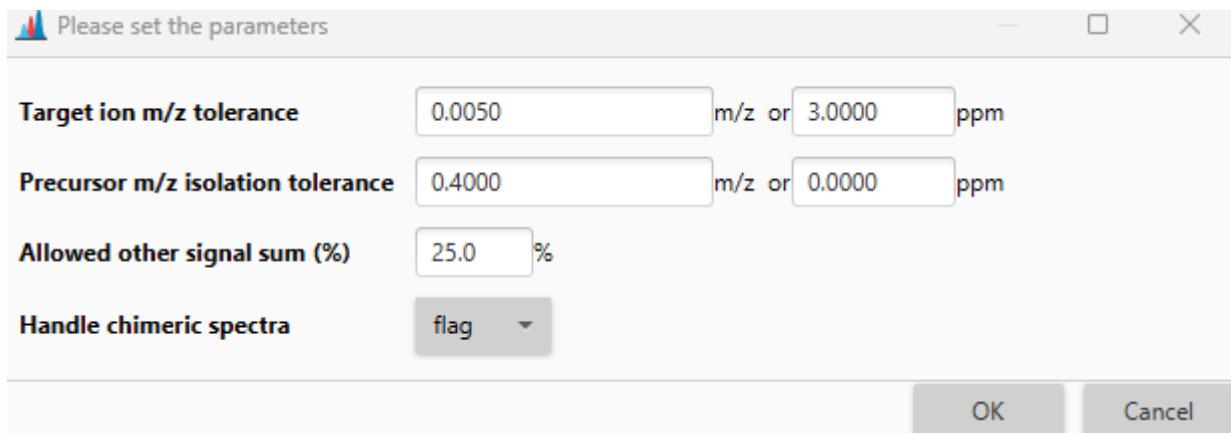
Handle chimeric spectra

Optional parameter

Options to identify and handle chimeric spectra with multiple MS1 signals in the precursor ion selection.

Handle chimeric spectra setup

This dialog allows to set parameters for handling chimeric spectra. The spectrum detected as chimeric will be flagged if this option is chosen.

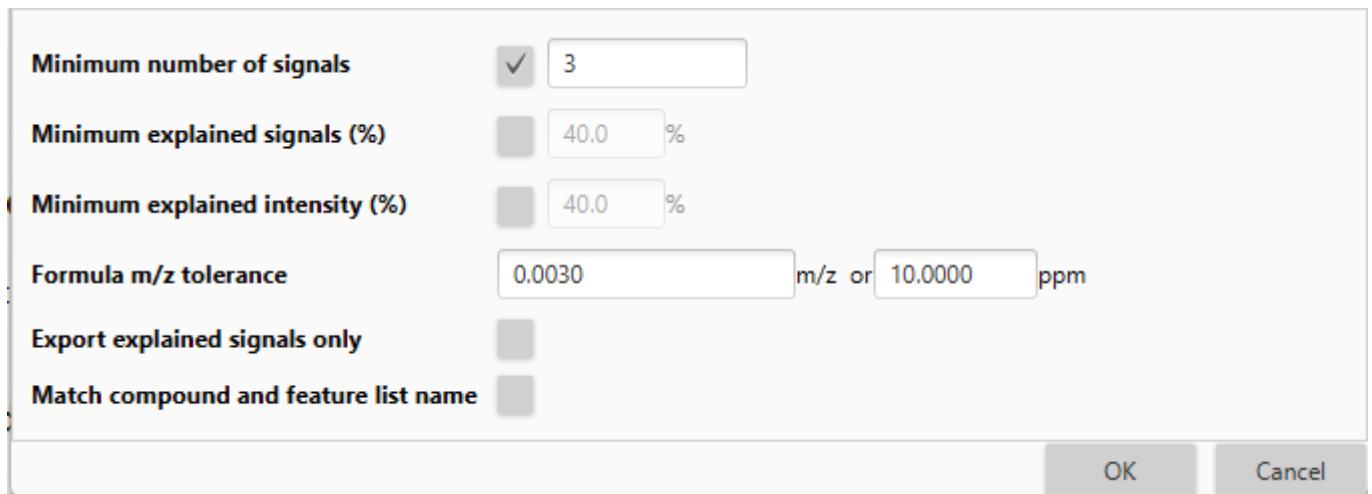


Quality parameters

Quality parameters for MS/MS spectra to be exported to the library.

Quality parameters spectra setup

The following dialog allows to set the criteria for the MS/MS export to the library, such as minimum number of signals, percentage of explained signals or intensity, etc.



Robin Schmid, omokshyna

⌚ April 6, 2023 09:55:53

👤 Robin Schmid, omokshyna

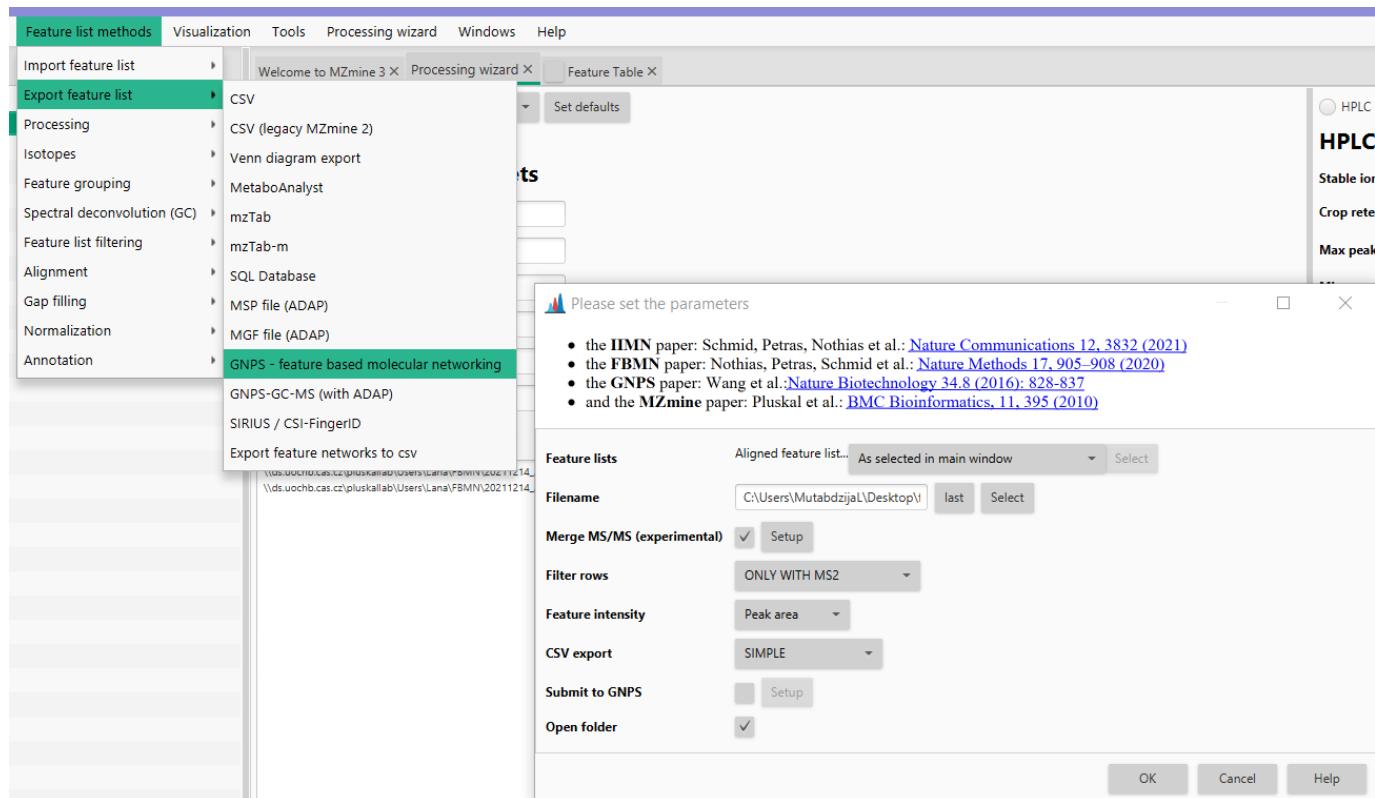
11.1.5 Data export to other software

GNPS-FBMN/IIMN export

DESCRIPTION

≡ Feature list methods → Export feature lists → GNPS - feature based molecular networking.

This module connects MZmine feature finding results to the [GNPS](#) workflows for [Feature-based Molecular Networking \(FBMN\)](#) and [Ion Identity Molecular Networking \(IIMN\)](#).



Using this module, the user can export the feature list needed for the manual submission to GNPS' feature based molecular networking (GNPS FBMN) or directly submit the job to the GNPS platform from MZmine. In both cases, two files are created:

1. Quantification table (CSV file) which contains the features and their associated information (e.g., average m/z, retention time, and each feature's area or height).
2. MS/MS spectral summary (.MGF file) which contains one representative MS/MS spectrum for each row in the feature list.
3. A [supplementary edges file](#) with related ion identities (if ion identity networking was performed).

REFERENCES

- IIMN:** Schmid R., Petras D., Nothias LF, et al. [Ion Identity Molecular Networking for mass spectrometry-based metabolomics in the GNPS Environment](#). Nat. Comm. 12, 3832 (2021).
- FBMN:** Nothias, L.-F., Petras, D., Schmid, R. et al. [Feature-based molecular networking in the GNPS analysis environment](#). Nat. Methods 17, 905–908 (2020).
- GNPS:** Wang, M. et al. [Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking](#). Nat. Biotechnol. 34, 828–837 (2016).

PARAMETERS**Filename**

Name to be given to the output files (.MGF and .CSV). In this field, the user can either write the path where they want to save the file, or click "select", navigate into the desired output folder, write the output name in the "file name" field and click save. Once that is done, the path should be visible in the Filename field in the GNPS export module.

Merge & select fragment scans

This parameter controls how fragment spectra are filtered, merged, and selected for downstream analysis (see [detailed description](#)). Briefly, either choose preset based spectral merging, input scans without merging, or an advanced setup for more options.

For GNPS export, mzmine recommends using the preset **Single scan: Merged across energies** as GNPS requires one MS2 per feature.

Filter rows

In the final output files, the user can select to export all the rows without any filters applied, rows only with MS/MS spectra, rows with MS/MS and Ion Identity (it gives MS/MS and the adduct information) and rows with MS/MS or Ion Identity. Normally, for FBMN you want to retain features with MS/MS spectra.

Feature intensity

The user can either select peak area or peak height which will then be displayed in the quantification table.

CSV export

The user can choose between **simple**, **comprehensive**, or **all**. Difference is in the amount of information that is present in the quantification table. Simple resembles the legacy format from the MZmine 2 export. Both options can be used for FBMN in GNPS other tools might rely on the simple MZmine 2 style output.

Submit to GNPS

This option allows any user to directly submit FBMN/IIMN jobs to GNPS. The password and user name are optional and are sent without encryption (until the server has moved to its final location with https). The input files uploaded to GNPS with the "Submit to GNPS" option are not saved on your GNPS user account. These files are deleted on monthly basis, which prevent future cloning of the job and retrieval of the files. Use the "standard" interface of FBMN for persistant jobs and more options. Or log into your GNPS account and click on **Clone to latest version** for a job submitted via direct interface.

Open folder

Opens the export folder.

GNPS GC-MS (with ADAP)**DESCRIPTION**

≡ Feature list methods → Export feature list → Export to GNPS GC-MS (with ADAP)



This module should be run on aligned feature list after applying one of the spectral deconvolution methods (see ADAP manual).

The GNPS GC-MS module exports all files needed for the GC-MS workflow on the GNPS web platform. This includes the **feature quantification table (CSV file)** and the **spectral summary (.MGF file)** with representative clustered GC-EI-MS spectra for the row in an (aligned) feature list.

REFERENCES**Quote**

1. Smirnov et al.: J. Proteome Res 2017, DOI: 10.1021/acs.jproteome.7b00633
2. Wang et al.: Nature Biotechnology 34.8 (2016): 828-837.
3. Pluskal et al.: BMC Bioinformatics, 11, 395 (2010)

PARAMETERS**Feature lists**

Name of the feature list(s) to be exported.

Filename

Base name for the output files

Representative m/z

The m/z value that is exported to the mgf and used in GNPS GC-MS. Options: - As in feature table, - highest m/z, - heighest intensity.

Feature intensity

Intensity in the quantification table (csv file). Available options are: - Peak area, - Peak height.

Open folder

Optional parameter

Opens the folder containing the output files.

SIRIUS/ CSI-FingerID**DESCRIPTION****≡ Feature list methods → Export feature list → SIRIUS / CSI-FingerID**

The module allows the export of spectral features to Sirius from a selected feature list or an aligned feature list in .MGF format. The generated .MGF file can be directly processed with Sirius (version 3.4 minimum).

Sirius uses both the deconvoluted isotopic pattern for each feature and the corresponding MS2 spectrum to rank molecular formula candidates. Spectral features exported consist of a pair of MS1 and MS2 spectrum for each row. Note that the MS1 spectrum can contain the deconvoluted isotopic peaks.

Warning

A masslist with mass detection performs on MS level 2 has to be generated first and selected.

This module can be used in combination with the Export for GNPS module and imported back into GNPS molecular networks. Please refer to the following documentation for a detailed tutorial:

REFERENCES**Quote**

1. **MZmine2 paper:** Pluskal et al.: BMC Bioinformatics, 11, 395 (2010)
2. K. Dührkop, et al. "Sirius 4: a rapid tool for turning tandem mass spectra into metabolite structure information", Nat methods, 2019.

Instructions

Before using the module, perform the following:

- Deconvolute isotopic peaks: Feature list methods → Isotopes → 13C isotope filter
- Filter out peak without MS/MS spectrum associated. MZmine2 Menu: Peak list methods / Filtering / Peak list row filter / Keep only peaks with MS2 scan (GNPS)
- Reset the peak number ID. MZmine2 Menu: Peak list methods / Filtering / Peak list row filter / Reset the peak number ID
- Perform mass detection on MS level 2 and select. MZmine2 Menu: Raw data methods / Mass detection [Make sure to use a noise threshold consistent with the MS level 2]

PARAMETERS**Feature lists**

Name of the feature list(s) to be exported.

Filename

Base name for the output files

Merge & select fragment scans

This parameter controls how fragment spectra are filtered, merged, and selected for downstream analysis (see [detailed description](#)). Briefly, either choose preset based spectral merging, input scans without merging, or an advanced setup for more options.

For SIRIUS handles merging internally, mzmine recommends exporting either **All input scans** or using a simple preset for merging in mzmine.

m/z tolerance

m/z tolerance to exclude duplicates in correlated spectrum.

Renumber IDs

Optional parameter

Resets the IDs (uses the row ID otherwise).

Only rows with annotation

Optional parameter

Only export rows with an annotation (run MS annotate or metaMScorrelate).

Exclude empty MS/MS spectra

Optional parameter

Do not export empty MS/MS spectra (only features with MS/MS spectrum with at least 1 signal are exported).

Exclude multiple charge

Optional parameter

Do not export multiply charged rows.

Exclude multimers

Optional parameters

Do not export rows that were annotated as multimers (2M) (run MS annotate or metaMScorrelate).

MetaboAnalyst export

DESCRIPTION

≡ Feature list methods → Export feature list → MetaboAnalyst export

This module exports the contents of the feature list into a csv file formatted. This csv file can be easily uploaded into the **Statistical Analysis** tools of the **MetaboAnalyst** service (metaboanalyst.ca).



Prior to export, samples must be grouped by a sample metadata column that can be defined manually or better loaded from a metadata csv/tsv table file. (see ≡ Project → Sample metadata). Each group must contain at least 3 samples (raw data files).

Format of exported CSV file

The exported file is formatted as shown below. Each feature is represented by its ID in the feature list, m/z value (formatted according to MZmine configuration), retention time value (formatted according to MZmine configuration), and the preferred identity (name).

```
"Sample", "ko15", "ko16", "ko18", "ko19", "ko21", "ko22", "wt15", "wt16", "wt18", "wt19", "wt21", "wt22"
"Label", "KO", "KO", "KO", "KO", "KO", "WT", "WT", "WT", "WT", "WT", "WT"
"1:258.110@13.4 (name1)", 147887.53, 451600.71, 65290.38, 56540.93, 85146.33, 162012.44, 175177.08, 82619.48, 51951.61, 69198.22, 153273.47, 98144.28
"2:137.045@9.4 (name2)", 1778569, 1567038, 1482796, 1039130, 1223132, 1072038, 1950287, 1466781, 1572679, 1275313, 1356014, 1231442
"4:184.0733@13.7 (name4)", 237993.6, 269714, 201393.4, 150107.3, 176989.7, 156797, 276541.8, 222366.2, 211717.7, 186850.9, 188285.9, 172348.8
...
```

PARAMETERS

Filename

Name of exported CSV file for MetaboAnalyst. If the file already exists, it will be overwritten.

Metadata grouping

Sample metadata column that will be used to obtain group information for each sample (e.g. control vs disease). Please import sample metadata from csv/tsv files or define them manually.

UPLOADING TO METABOANALYST

In MetaboAnalyst, please select the Statistical Analysis module.

After selecting the Statistical Analysis module, please set the parameters as shown below. In the "Data File" field, please select the CSV file generated by this mzmine module.

1) Upload your data

Comma Separated Values (.csv) :

Data Type: Concentrations Spectral bins Peak intensity table

Format: Samples in columns (unpaired)

Submit

Data File: Choose File mzmine_exported_file.csv

For details about MetaboAnalyst functions and capabilities, please refer to their documentation.

Tidymass

Tidymass can import the csv feature table from MZmine, using `convert_mzmine2mass_dataset` function. For more details see [Tidymass tutorial](#).

Robin Schmid, Steffen Heuckeroth, omokshyna

 November 25, 2025 10:29:24

 Robin Schmid, Steffen Heuckeroth, omokshyna

11.1.6 Batch spectral library generation

≡ Feature list methods → Export feature list → Spectral library batch generation

After matching with a compound database table, only annotated spectra are exported with this module. This enables the automatic generation of mass spectral libraries with different data format provided. Different quality parameters and how to handle chimeric spectra can be set to increase the spectral quality in the new spectral library and to flag chimeric spectra. Furthermore, different information about the acquired data can be added, for example, instrument, who acquired the data,... Single best spectra but also merged spectra are exported (same energy, multiple collision energies, or multistep fragmentation)

Parameters

Export file

The path and name of the exported library spectra.

Export format

Here, you can set the export format. It mostly depends on your next steps and where to import the library. Every format works, but we recommend the .json format.

MS level filter

If data were acquired with multistep fragmentation, the MS2 or all MSn data can be exported individually. If you only acquired MS2 data or want to export MSn, the setting can remain MSn

Metadata

Various information about the acquired data can be stored here. For example, who acquired the data, which instrument was used, etc. On the description information about the compounds or library can be given. The Ion mode is only needed, when the polarity is not saved in the data files.

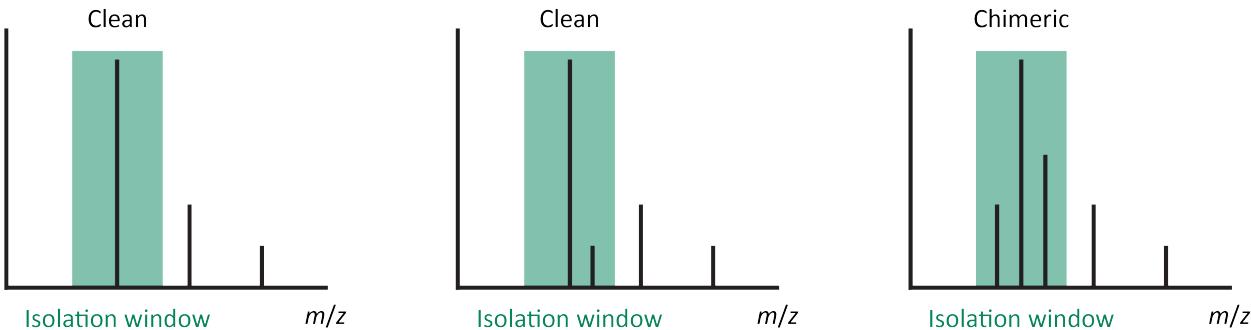
Merge & select fragment scans

This parameter controls how fragment spectra are filtered, merged, and selected for downstream analysis (see [detailed description](#)). Briefly, either choose preset based spectral merging, input scans without merging, or an advanced setup for more options.

For library generation, mzmine recommends the simple preset **Representative scans or MSn tree**.

Handle chimeric spectra

This option checks the MS1 scans for chimeric precursor selection. A minimum for this purity can be set and how to handle chimeric spectra.



Target ion m/z tolerance

This is the mass tolerance for the precursor signal. The tolerance is usually smaller than the precursor isolation window and depends on the accuracy and resolution of the instrument.

Precursor m/z isolation tolerance

This should be the set precursor isolation window, to check for co-isolated signals in this mass window. Be aware, that some QTOF instruments have a wider isolation window than a unit, resulting in co-isolation of isotope signals.

Minimum precursor purity

MS1 spectra with a precursor signal explaining less than X% of the total signal intensity within the set precursor isolation window will be flagged as chimeric.

Chimeric spectra

Here, spectra that are flagged as chimeric can be skipped or flagged. We suggest that flagging is a better option, as the flagging state will be exported and can be removed afterwards as it contains still information about the precursor ion. But matches need to be inspected carefully.

QUALITY PARAMETERS

Quality parameter can be set before exporting the spectra. Only spectra that fullfil those criteria will be exported.

Minimum number of signals

Number of signals an MSn scan must contain to be exported. Depends also on the fragmentation pattern of the compounds (very rich pattern, more signals, less informative spectra less signals). At least 3 signals might be a good compromise if the compounds show a very different fragmentation behavior.

Minimum explained signals (%)

The number of explained signals within the MSn scan based on a subformula level, without taking the chemical structure into account.

Minimum explained intensity (%)

The minimum number of the explained spectra intensity. This increases the value of more abundant signals that should be explained in a fragmentation spectrum based on the subformula level for an annotated compound. Low abundant signals can correspond to unfiltered noise, ect. and shouldn't be taken into account in the same way as high abundant signals. Therefore, we recommend this setting before the explained signal one.

Formula m/z tolerance

This is the m/z tolerance to assign MSn signals to a subformula.

[Robin Schmid, corinnabrungs](#)

 January 6, 2025 06:05:58

 [Robin Schmid, corinnabrungs](#)

11.1.7 Export scans (feature list)

≡ Feature list methods → Export feature lists → Export scans (feature list)

A module to export spectra in different formats like mgf, msp, json (mzmine) with many options to customize the scan selection and output. One application of this module is to export fragment scans merged on various levels. Furthermore, this module allows to generate "blank" or "sample" libraries of unknown spectra with added identifiers from sample wide metadata.

Parameters

Export file

The path and name of the exported spectra.

Export format

Here, you can set the export format. It mostly depends on your next steps and where to import the library. Every format works, but we recommend the .json format.

Metadata

Various information about the acquired data can be stored here. For example, who acquired the data, which instrument was used, etc. On the description information about the compounds or library can be given. The Ion mode is only needed, when the polarity is not saved in the data files.

Sample wide metadata

This is metadata that is imported as project metadata for each sample, usually from a tsv or csv table. This parameter allows to map the values from specific columns to all scans of these samples. So if a spectrum was merged from two samples, the compound name or entry description will contain the values from all source samples.

This option was added to allow for **blank libraries** or **sample libraries**. Measure various blanks like "well plate, manufacturer XY" and load this information as sample metadata. Then define the column to be used as compound name or description to export all scans from this sample labelled as potentially originating from this blank.

Skip annotated features

If true, annotated features will be skipped and only "unknown" features will be exported. This is useful when all annotated features were already exported by the **Spectral library batch generation**.

Export MS1

Option to export MS1 scans, either the most intense MS1 across all samples or a correlated MS1 of isotopologue and adduct signals. The MS1 data may be exported together with fragment scans or into a separate file.

Intensity normalization

Options to apply intensity normalization to all signals. Scientific format is preferred as it best accounts for many orders of magnitude.

EXPORT FRAGMENT SCANS

Handle chimeric spectra

Score the isolation purity and handle chimeric precursor ion isolation. See full description of all parameters in **Spectral library batch generation**.

Merge & select fragment scans

This parameter controls how fragment spectra are filtered, merged, and selected for downstream analysis (see [detailed description](#)). Briefly, either choose preset based spectral merging, input scans without merging, or an advanced setup for more options.

Minimum signals

Threshold to exclude low informative spectra from export.

ADVANCED**Compact USI**

Universal spectrum identifiers:

- Compact USI are non canonical USI for merged spectra and export a list of USI for each sample one, where all source scan number are reduced to number ranges as in 1-5,9 for the scans from 1 to 5 and scan 9. This greatly reduces the redundancy compared to regular USI.
 - Regular USI export a list of USI, one for each source scan of a merged spectrum.
-

Robin Schmid

 January 6, 2025 06:05:58

 [Robin Schmid](#)

11.1.8 Export MRMs to TSV

⚠ Warning

To process multiple reaction monitoring (MRM) data, the MRM service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

The Export MRMs to TSV module exports MRM-specific metrics to a tsv or csv table. An exemplary export is shown below:

id	feature_list	datafile	q1_mz	q3_mz	is_quantifier	area	height	rt
43	Aligned featuri	FILE_A	568	92	false	0.00E+00	4.47E+01	5.7424
43	Aligned featuri	FILE_A	568	568	true	6.83E+02	1.68E+04	5.7424
43	Aligned featuri	FILE_B			false			
44	Aligned featuri	FILE_A	400	400	true	1.07E+05	5.06E+06	5.8409
44	Aligned featuri	FILE_B	400	400	true	1.24E+05	6.23E+06	5.8409

Figure 1: Example export result with [Skip missing features](#) disabled.

Parameters

Feature lists

Select the feature lists that shall be exported.

Export file

Select the file to which the export should export to.

Skip missing features

Select if the export should skip missing features, if a feature was detected in file A but not in file B. An example of the behaviour is shown in the screenshot (Figure 1) above. Here, the [Skip missing features](#) option was disabled, creating an empty line for the feature of ID 43 in FILE_B. The export contains the ID of the row the feature belongs and was aligned to, but no q1 or q3 masses, no area, no height, and no RT, since the feature was not detected.

If the [Skip missing features](#) option is enabled, this line would not exist.

Robin Schmid, Steffen Heuckeroth

⌚ November 25, 2025 10:29:24

👤 Robin Schmid, Steffen Heuckeroth

11.1.9 Export feature data

This module will export the data of all features in a feature list to a file.

≡ Feature list methods → Export feature list → Export feature data

Parameters

Feature lists

Select the feature list(s) you want to export to a file. If multiple feature lists shall be exported, make sure to use the placeholder `{}` in the [Export file name](#).

Export file name

Specify the name of the file to export the feature data into. Use `{}` to automatically insert the name of the respective exported feature list.

Export format

Specify the export format. XML is currently the only available option.

An example XML file may look like this (Some entries were removed to keep it more or less concise):

```
<?xml version='1.0' encoding='UTF-8'?>
<featurelist
    featurelistname="Aligned feature list 13C peak gaps dup corr PEARSON r greq 0.85 dp greq 5"
    numberofrows="4204" date="2025/05/16 12:15:41">
    <row id="49">
        <datatype type="area">3262872.0</datatype>
        <datatype type="rt">0.32390267</datatype>
        <datatype type="mz_range">[175.11827087402344;175.1194610595703]</datatype>
        <datatype type="charge">1</datatype>
        <datatype type="fragment_scans">
            <scan scantype="simplescan" rawdatafile="171103_PMA_TK_media_02.mzML" scanindex="151"/>
            <scan scantype="simplescan" rawdatafile="171103_PMA_TK_media_03.mzML" scanindex="153"/>
        </datatype>
        <datatype type="feature_shape">true</datatype>
        <datatype type="alignment_scores">
            <subtypes>
                <datatype type="rate">0.3043478</datatype>
                <datatype type="aligned_features_n">7</datatype>
                <datatype type="align_extra_features">5</datatype>
                <datatype type="weighted_distance_score">0.8496631</datatype>
                <datatype type="mz_diff_ppm">4.4820127</datatype>
                <datatype type="mz_diff">7.84884589847934E-4</datatype>
                <datatype type="rt_absolute_error">0.021971852</datatype>
                <datatype type="ion_mobility_absolute_error">0.0</datatype>
            </subtypes>
        </datatype>
        <datatype type="rt_range">[0.30016744;0.44724196]</datatype>
        <datatype type="mz">175.11891174473593</datatype>
        <datatype type="spectral_db_matches">
            <feature_annotation annotation_type="spectral_library_annotation">
                <spectraldatabaseentry library_file="MoNA-export-LC-MS-MS_Spectra.json (150211 spectra)">
                    <mzs>
                        60.05536;70.06634;72.08131;112.0877;114.1043;115.0878;116.0718;130.0988;157.1094;158.0937;175.1232
                    </mzs>
                    <intensities>
                        50.218311;69.503389;4.296692;4.05231;1.387287;1.208214;38.124055;16.18238;2.837249;14.222061;100.0
                    </intensities>
                    <databasefieldslist>
                        <entry name="INSTRUMENT_TYPE">LC-ESI-QTOF</entry>
                        <entry name="MS_LEVEL">MS2</entry>
                        <entry name="INCHI">
                            InChI=1S/C6H14N4O2/c7-4(5(11)12)2-1-3-10-6(8)9/h4H,1-3,7H2,(H,11,12)(H4,8,9,10)/t4-/m0/s1
                        </entry>
                        <entry name="PRECURSOR_MZ">175.118957519531</entry>
                        <entry name="POLARITY">positive</entry>
                        <entry name="ION_TYPE">[M+H]+</entry>
                        <entry name="ION_SOURCE">ESI</entry>
                        <entry name="DATA_COLLECTOR">Isabel Meister, Romanas Chaleckis</entry>
                        <entry name="SMILES">O=C(O)C(N)CCNC(=N)N</entry>
                        <entry name="PRINCIPAL_INVESTIGATOR">Isabel Meister, Romanas Chaleckis</entry>
                        <entry name="INCHIKEY">ODKSFYDXXFIFQN-BPYZUCNSA-N</entry>
                        <entry name="FORMULA">C6H14N4O2</entry>
                        <entry name="NAME">Arginine; LC-TDDA; CE10</entry>
                        <entry name="RT">12.65</entry>
                        <entry name="INSTRUMENT">Agilent 6550 iFunnel</entry>
                        <entry name="COLLISION_ENERGY">10</entry>
                    </databasefieldslist>
                </spectraldatabaseentry>
            </feature_annotation>
        </datatype>
    </row>
</featurelist>
```

```

<spectralsimilarity>
  <similarityfunction>Weighted cosine similarity</similarityfunction>
  <overlappingpeaks>10</overlappingpeaks>
  <score>0.991997389209482</score>
  <explainedLibraryIntensity>1.0</explainedLibraryIntensity>
  <libraryspectrum>
    <mzs>
      60.05536;70.06634;72.08131;112.0877;114.1043;115.0878;116.0718;130.0988;157.1094;158.0937
    </mzs>
    <intensities>
      50.218311;69.503389;4.296692;4.05231;1.387287;1.208214;38.124055;16.18238;2.837249;14.222061
    </intensities>
  </libraryspectrum>
  <queryspectrum>
    <mzs>

```

60.05648689221877;70.06589447026973;72.08146690621223;112.08713629620883;114.10296712134367;115.0865468821093;116.0707957060524;130.09748713358292;157.1084853
0129516;158.09225213474215

```

    </mzs>
    <intensities>

```

1.700311924990654E7;2.792031528575635E7;5692692.274521589;2063768.8710607588;505617.66608789563;535806.9913217761;1.6561022981089354E7;6705487.050384879;10575
48.0377922803;5742073.7107543945

```

    </intensities>
    <queryspectrum>
      <alignedspectrumlist numvalues="2">
        <alignedspectrum>
          <mzs>
            60.05536;70.06634;72.08131;112.0877;114.1043;115.0878;116.0718;130.0988;157.1094;158.0937
          </mzs>
        <intensities>
          50.218311;69.503389;4.296692;4.05231;1.387287;1.208214;38.124055;16.18238;2.837249;14.222061
        </intensities>
      </alignedspectrum>
      <alignedspectrum>
        <mzs>

```

60.05648689221877;70.06589447026973;72.08146690621223;112.08713629620883;114.10296712134367;115.0865468821093;116.0707957060524;130.09748713358292;157.1084853
0129516;158.09225213474215

```

        </mzs>
        <intensities>

```

1.700311924990654E7;2.792031528575635E7;5692692.274521589;2063768.8710607588;505617.66608789563;535806.9913217761;1.6561022981089354E7;6705487.050384879;10575
48.0377922803;5742073.7107543945

```

        </intensities>
        <alignedspectrum>
        <alignedspectrumlist>
      </spectralsimilarity>
      <ccserror>NULL_VALUE</ccserror>
      <testeddmz>175.11891174473593</testeddmz>
      <testeddrt>0.32390267</testeddrt>
      <scan scantype="simplemergedmsmspectrum" mslevel="2" merge_spec_type="SAME_ENERGY"
        ce="30.0" mergingtype="MAXIMUM" rawdatafile="171103_PMA_TK_media_05.mzML">
        <msmsinfo type="ddamsmsinfo" precursormz="175.118453979492" charge="1" energy="30.0"
          activationtype="HCD" mslevel="2" isolationwindow="[174.618453979492;175.618453979492]" />
      <mzs>

```

60.05648689221877;70.06589447026973;72.08146690621223;112.08713629620883;114.10296712134367;115.0865468821093;116.0707957060524;130.09748713358292;157.1084853
0129516;158.09225213474215;175.118826887203

```

        </mzs>
        <intensities>

```

1.700311924990654E7;2.792031528575635E7;5692692.274521589;2063768.8710607588;505617.66608789563;535806.9913217761;1.6561022981089354E7;6705487.050384879;10575
48.0377922803;5742073.7107543945;2.654460362855494E7

```

        </intensities>
        <scans rawdatafile="171103_PMA_TK_media_03.mzML">{153}[]</scans>
        <scans rawdatafile="171103_PMA_TK_media_06.mzML">{149}[]</scans>
        <scans rawdatafile="171103_PMA_TK_media_04.mzML">{144}[]</scans>
        <scans rawdatafile="171103_PMA_TK_media_05.mzML">{150}[]</scans>
        <scans rawdatafile="171103_PMA_TK_media_02.mzML">{151}[]</scans>
      </scans>
      </feature_annotation>
    </datatype>
    <datatype type="intensity_range">[171891.52;7.5953896E7]</datatype>
    <datatype type="feature_group">27</datatype>
    <datatype type="id">49</datatype>
    <datatype type="height">7.5953896E7</datatype>
    <feature rawdatafile="171103_PMA_TK_M1_01.mzML">
      <datatype type="area">28286.621</datatype>
      <datatype type="rt">0.32012287</datatype>
      <datatype type="mz_range">[175.11862182617188;175.11878967285156]</datatype>
      <datatype type="best_ms1_scan_number">
        <scan scantype="simplescan" rawdatafile="171103_PMA_TK_M1_01.mzML" scanindex="144"/>
      </datatype>
      <datatype type="fwhm">0.019634774</datatype>
      <datatype type="rt_range">[0.30645204;0.34960598]</datatype>
      <datatype type="feature_state">ESTIMATED</datatype>
      <datatype type="datafile">171103_PMA_TK_M1_01.mzML</datatype>
      <datatype type="mz">175.11867991636373</datatype>
      <datatype type="intensity_range">[290797.88;1358260.6]</datatype>
      <datatype type="asymmetry_factor">2.1566434</datatype>
      <datatype type="feature_data">
        <feature_data>
          <simpleiontimeseries>

```

```

<scans numvalues="7">136;140;144;148;152;156;160</scans>
<intensities numvalues="7">
  0.0;584109.4375;1358260.625;1170476.875;665673.75;290797.875;0.0
</intensities>
<mzs numvalues="7">
  175.11877432337528;175.1186981201172;175.11862182617188;175.11865234375;175.11878967285156;175.1187744140625;175.11877432337528
</mzs>
<rts numvalues="7">
  0.30645203590393066;0.3132686913013458;0.3201228678226471;0.32709363102912903;0.3339269161224365;0.3407540023326874;0.3496059775352478
</rts>
</simpleiontimeseries>
</feature_data>
</datatype>
<datatype type="tailing_factor">1.5783217</datatype>
<datatype type="height">1358260.6</datatype>
</feature>
<feature rawdatafile="171103_PMA_TK_M1_02.mzML">
  <datatype type="area">26188.146</datatype>
  <datatype type="rt">0.31954175</datatype>
  <datatype type="mz_range">[175.11877432337528;175.1194610595703]</datatype>
  <datatype type="best_ms1_scan_number">
    <scan scantype="simplescan" rawdatafile="171103_PMA_TK_M1_02.mzML" scanindex="144"/>
  </datatype>
  <datatype type="fwhm">0.02036497</datatype>
  <datatype type="rt_range">[0.30587724;0.34912074]</datatype>
  <datatype type="feature_state">ESTIMATED</datatype>
  <datatype type="datafile">171103_PMA_TK_M1_02.mzML</datatype>
  <datatype type="mz">175.11930236793012</datatype>
  <datatype type="intensity_range">[222693.89;1360981.8]</datatype>
  <datatype type="asymmetry_factor">1.7352959</datatype>
  <datatype type="feature_data">
    <feature_data>
      <simpleiontimeseries>
        <scans numvalues="7">136;140;144;148;152;156;160</scans>
        <intensities numvalues="7">
          0.0;881861.4375;1360981.75;924455.9375;405910.3125;222693.890625;0.0
        </intensities>
        <mzs numvalues="7">
          175.11877432337528;175.1192169189453;175.11924743652344;175.1194305419922;175.11929321289062;175.1194610595703;175.11877432337528
        </mzs>
        <rts numvalues="7">
          0.3058772285120392;0.3126917779445648;0.3195417523384094;0.3263646066188812;0.33320000767707825;0.34002700448036194;0.34912073612213135
        </rts>
      </simpleiontimeseries>
    </feature_data>
  </datatype>
  <datatype type="tailing_factor">1.5823284</datatype>
  <datatype type="height">1360981.8</datatype>
</feature>
<feature rawdatafile="171103_PMA_TK_M1_03.mzML">
  <datatype type="area">19992.443</datatype>
  <datatype type="rt">0.32646403</datatype>
  <datatype type="mz_range">[175.11877432337528;175.1190643310547]</datatype>
  <datatype type="best_ms1_scan_number">
    <scan scantype="simplescan" rawdatafile="171103_PMA_TK_M1_03.mzML" scanindex="148"/>
  </datatype>
  <datatype type="fwhm">0.021426953</datatype>
  <datatype type="rt_range">[0.30597037;0.34899095]</datatype>
  <datatype type="feature_state">ESTIMATED</datatype>
  <datatype type="datafile">171103_PMA_TK_M1_03.mzML</datatype>
  <datatype type="mz">175.1189857310156</datatype>
  <datatype type="intensity_range">[269273.7;865271.25]</datatype>
  <datatype type="asymmetry_factor">1.0992143</datatype>
  <datatype type="feature_data">
    <feature_data>
      <simpleiontimeseries>
        <scans numvalues="7">136;140;144;148;152;156;160</scans>
        <intensities numvalues="7">
          0.0;367347.625;859838.75;865271.25;523678.3125;269273.6875;0.0
        </intensities>
        <mzs numvalues="7">
          175.11877432337528;175.11904907226562;175.11891174316406;175.11898803710938;175.1190185546875;175.1190643310547;175.11877432337528
        </mzs>
        <rts numvalues="7">
          0.30597037076950073;0.31278908252716064;0.31963691115379333;0.3264640271663666;0.33330148458480835;0.34013262391090393;0.34899094700813293
        </rts>
      </simpleiontimeseries>
    </feature_data>
  </datatype>
  <datatype type="tailing_factor">1.0496072</datatype>
  <datatype type="height">865271.25</datatype>
</feature>
</row>
</featurelist>

```

 May 19, 2025 13:08:10

 Steffen Heuckeroth

11.1.10 Annotation graphics export

Export in batch mode (head less, CLI)

Warning

To export the annotation graphics in head less mode, a display needs to be configured to render the graphics. If you are running on a desktop computer, this is usually the case. If you are running on a server, e.g., ubuntu, you can use GTK and run it via Xvfb.

Note

This export only covers annotated features. Features without an annotation will not be exported.

Parameters

Export directory

Set a parent directory to which the graphics shall be exported. A sub directory with the name of the selected feature list will be created by the export module. An individual file will be created for each of the exported graphics. The file names follow this pattern:

```
%d-%s_mz-%s_score-%s_%s_match-%d
```

which will be replaced like this:

```
<featureid>-<graphic type>_mz-<feature mz>_score-<match score>_<additional note>_match-<match number>
```

Feature lists

Select the feature lists to export

Number of matches to export

Define how many matches per annotation type (supported: spectral library matches and lipid annotations) shall be exported to a graphic. The match number will refer to the ranking in an annotation type.

DPI scaling factor

Defines a scaling factor for pixels in exported png images.

Export shapes

If selected, chromatogram charts will be exported for each row.

Export mobilograms

If selected, mobilogram charts will be exported for each row.

Export images

If selected, each feature will be exported as ion image (for imaging data files).

Export lipid matches

If selected, a chart of the matched lipid signals in the MS2 spectrum will be exported.

Export spectral library matches

If selected, the spectral library match will be exported as a mirror plot with metadata information. The png export will contain more metadata information, but cannot be edited. The pdf export can be edited using a vector graphics tool.

[Export PDF](#)

If selected, the plots will be exported as PDF.

[Export PNG](#)

If selected, the plots will be exported as PNG.

[Chart export parameters](#)

Define chart export parameters.

 November 25, 2025 10:29:24

 Robin Schmid, Steffen Heuckeroth

11.2 File merging

11.2.1 Merge raw data files

Description

 Raw data methods → File merging → Raw data file merging

This module merges all raw data files into a new. For example to combine positive and negative scans with MS2, all from different raw data files.

Method parameters

Raw data files

Raw data files the module will take as an input.

Mode

- MERGE PATTERN:

Merge files based on a grouping identifier which can be a name suffix or prefix. (e.g., Sample_A_1, Sample_B_1: Use AFTER LAST _ to combine these files)

- MERGE SELECTED:

Merge all selected files to a new

Grouping identifier position

Search for the specific group identifier before the first or after the last marker.

Position marker

The marker that splits the specific group identifier from the rest of the file names.

MS2 marker

If a raw data file has this marker in its name, it will only be used as a source of MS2 (MSn) scans. All MS1 scans of this file will be discarded.

Suffix

Suffix to be added to the new file name.

omokshyna

 November 24, 2022 21:13:07

 omokshyna

11.2.2 Mobility scan merging

Description

≡ Raw data methods → File merging → Mobility scan merging

This module merges mobility scans in each **single ion mobility data file** at the same retention time to a summed frame spectrum.

The merged frame spectrum is used if a *.mzML file is imported. The merged frame spectrum is required to gain access to MZmine's LC-MS functionality.

! This step uses the centroided and thresholded data produced by the [mass detection](#) step.

! This step is not required when importing native **Bruker .tdf or .tsf** data from .d folders. When importing native Bruker data, a merged spectrum for the frame is created automatically by the vendor library.

Parameters

Raw data files

Raw data files the module will take as an input.

Noise level

Data points beyond the defined noise level threshold will be ignored.

Merging type

The way to calculate intensities. Intensities can be either averaged, summed, or the maximum value can be chosen.

m/z weighing

Chosen function is used to weigh m/z values by their intensities. The available options are:

- None,
- Linear,
- log10,
- log2,
- square root,
- or cube root.

Scans

Selects the scans that should be included.

m/z tolerance

Maximum allowed difference between two m/z values in order for them to be considered the same.

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [omokshyna](#)

11.3 Raw data filtering

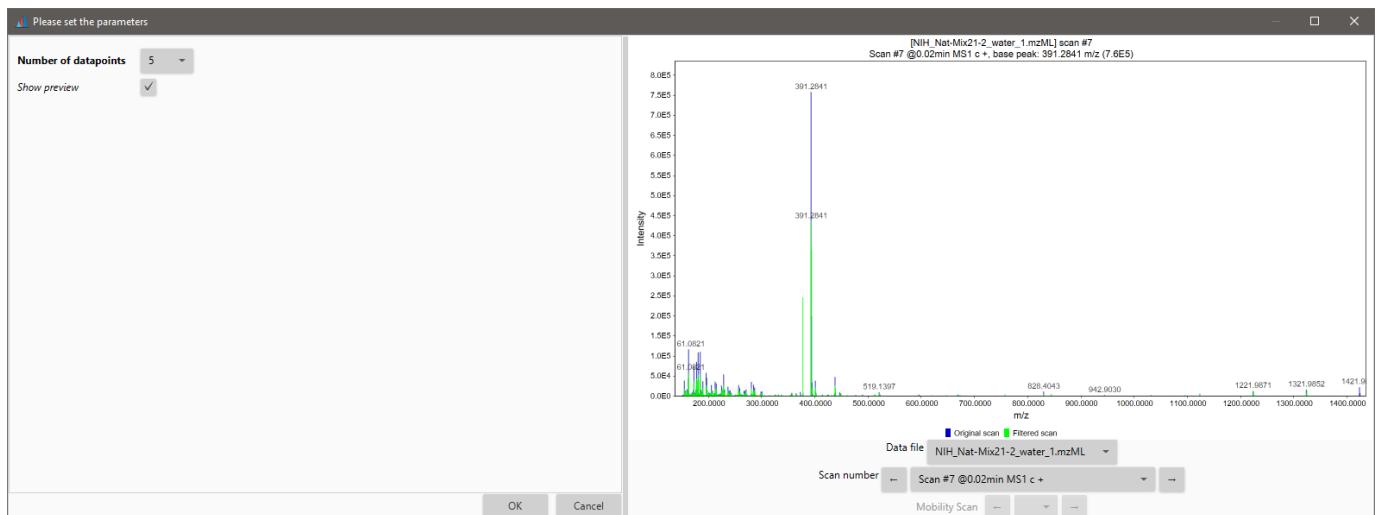
11.3.1 Scan by scan filtering

Description

≡ Raw data methods → Raw data filtering → Scan by scan filtering

This module can be used to preprocess data in each scan using various filters.

The preview shows the superposed spectra of the scan before and after the Savitzky-Golay filter is applied.



Filters

MEAN FILTER

For each data point, the filter assigns to it the intensity average of all the datapoints inside the user defined window. The window is centered in the mass value of this data point.

Parameters

Window length

One-sided length of the m/z smoothing window.

SAVITZKY GOLAY FILTER

The Savitzky-Golay smoothing filter was first described in 1964 by Abraham Savitzky and Marcel J. E. Golay.

The Savitzky-Golay method essentially performs a **local polynomial regression (of degree k)** on a series of values (of at least $k+1$ points which are treated as being equally spaced in the series) to determine the smoothed value for each point.

http://en.wikipedia.org/wiki/Savitzky_Golay_smoothing_filter

Parameters

Number of datapoints

This number can be 5, 7, 9, 11, 13 or 15.

RESAMPLING FILTER

Each scan is divided in m/z bins whose length is defined by the user in the parameters. The mass of the new data point will be in the middle of each m/z bin's space. Its intensity is the average of the intensity of all the data points inside the bin.

Parameters

m/z bin length

The length of m/z bin.

ROUND RESAMPLING FILTER

All data points in each scan is shifted to the nearest rounded integer (**ion number**). The mass of the new data point will be on the rounded value.

If several data points are competing for the same rounded value / mass, the new data point's intensity is the average of all the competing data points.

Unless "**Sum duplicate intensities**" is checked, leading to a single new data point with intensity equal to the sum of the intensities of all the competing data points.

 If the scan is not centroided, its spectrum is first turned into centroid using a default "local maxima" algorithm, then the same algorithm as described above is applied.

Parameters

Sum duplicate intensities

Sums ions count (intensity) of m/z peaks competing for being rounded at same m/z unit. If unchecked, the intensities are averaged rather than summed.

Remove zero intensity m/z peaks

Clear all scans spectra from m/z peaks with intensity equal to zero.

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

11.3.2 Crop filter

Description

≡ Raw data methods → Raw data filtering → Crop filter

This module performs cropping of raw data files based on the user-defined parameter range defined by user. This allows user to obtain a new raw data file that contains only the information from the range of interest.

Parameters

Scans

Define scan filtering parameters, which include:

- Scan number,
- Base filtering number,
- Retention time,
- Mobility,
- MS level,
- Scan definition (matching some pattern),
- Polarity,
- and Spectrum type.

m/z

m/z boundary of the cropped region

Suffix

This string is added to filename as suffix

Remove source file after filtering

If checked, only filtered file version is stored.

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [omokshyna](#)

11.3.3 Baseline correction

Description

≡ Raw data methods → Raw data filtering → Baseline correction

This module performs baseline correction on raw data files. It is designed to **compensate for gradual shifts** in the chromatographic baseline by detecting the baseline and then subtracting it from the raw data intensity values.

The module proceeds as follows for each raw data file passed to it:

- The full range of m/z values present in the raw data is divided into a series of bins of a specified width (see m/z bin width).
- For each bin a chromatogram is constructed from the raw data points whose m/z values fall within the bin. This chromatogram may be either the **base peak chromatogram** or **total ion count (TIC) chromatogram**.
- The raw intensity values of each data point in a bin are corrected by subtracting the bin's baseline. Subtraction of baseline intensity values proceeds according to the type of chromatogram used to determine the baseline.

If the **base peak chromatogram** was used then the corrected intensity values are calculated as follows:

$$\{I_{corr}\} = \max(0, I_{orig} - I_{base})$$

If the **TIC chromatogram** was used then the corrected intensity values are calculated as follows:

$$\{I_{corr}\} = \max(0, I_{orig} * (1 - I_{base}/I_{max}))$$

where $\{I_{orig}\}$, $\{I_{base}\}$, $\{I_{max}\}$, and $\{I_{corr}\}$ are the original, baseline, maximum and corrected intensity values, respectively, for a given scan and m/z bin.

If $\{I_{base}\}$ is less or equal to zero then no correction is performed, i.e. $\{I_{corr}\} = I_{orig}$.

- A new raw data file is generated from the corrected intensity values.

Parameters

Filename suffix

The text to append to the name of the baseline corrected raw data file.

Chromatogram type

TIC: total ion count, i.e. summed intensities per scan, or

Base peak intensity: maximum intensity per scan.

MS-level

MS level to which to apply correction. Select "0" for all levels.

Use m/z bins

Baselines can be calculated and data points corrected per m/z bin or to the entire raw data file. If no binning is performed then a single chromatogram is calculated for the entire raw data file and its baseline used to correct the full data file. No binning is very quick but much less accurate and so is only suitable for fine-tuning the smoothing and asymmetry parameters.

m/z bin width

The width of the m/z bins if binning is performed (see use m/z bins). Smaller bin widths result in longer processing times and greater memory requirements. Avoid values below 0.01.

Correction method

The width of the m/z bins if binning is performed (see use m/z bins). Smaller bin widths result in longer processing times and greater memory requirements. Avoid values below 0.01.

R engine

This option allows you to choose between two Java libraries to communicate with R - RServe or RCaller.

Remove source file after baseline correction

Whether to remove the original raw data file once baseline correction is complete.

Correction methods

More information on correction methods is available in [CRAN description of baseline package](#)

ASYMMETRIC BASELINE CORRECTOR

This corrector estimates a baseline using asymmetric least squares and subtracts it from the data.

Additional parameters**Smoothing**

The smoothing factor ($>= 0$), generally ranges from 1E5 to 1E8. The larger this factor is, the smoother the baseline.

Asymmetry

Default value is 1E-3. The weight ($0 <= p <= 1$) for points above the trend line, whereas $1-p$ is the weight for points below it. Naturally, p should be small for estimating baselines.

ROLLING BALL CORRECTOR

The corrector estimates a trend based on the **Rolling Ball algorithm**, and subtracts it from the raw data intensity values. (Ideas from **Rolling Ball algorithm for X-ray spectra by M.A.Kneen and H.J. Annegarn**. Variable window width has been left out).

Additional parameters**wm (number of scans)**

Width of local window for minimization/maximization (in number of scans).

ws (number of scans) Width of local window for smoothing (in number of scans).**PEAK DETECTION BASELINE CORRECTOR**

The corrector estimates a trend based on the Peak Detection algorithm, and subtracts it from the raw data intensity values. Peak detection is done in several steps sorting out real peaks through different criteria. Peaks are removed from spectra and minimums and medians are used to smooth the remaining parts of the spectra. (A translation from **Kevin R. Coombes et al.'s MATLAB code** for detecting peaks and removing baselines).

Additional parameters**left (number of scans)**

Smallest window size for peak widths (in number of scans).

right (number of scans)

Largest window size for peak widths (in number of scans).

lwin (number of scans)

Smallest window size for minimums and medians in peak removed spectra (in number of scans).

rwin (number of scans)

Largest window size for minimums and medians in peak removed spectra (in number of scans).

snminimum

Minimum signal to noise ratio for accepting peaks.

mono

Monotonically decreasing baseline if 'mono' > 0 .

multiplier

Internal window size multiplier.

RUBBER BAND CORRECTOR

The corrector estimates a trend based on the Rubber Band algorithm (which determines a convex envelope for the spectra - underneath side), and subtracts it from the raw data intensity values.

Additional parameters**noise**

Ignored if "auto noise" is checked. Noise level to be taken into account.

auto noise

Determine noise level automatically (from lower intensity scan).

df Degree of freedom.

spline

Logical indicating whether the baseline should be an interpolating spline through the support points or piecewise linear.

bend factor

Does nothing if equals to zero. Helps fitting better with low "**df**".

💡 Try starting with value around 5E4.

LOCAL MINIMA + LOESS CORRECTOR

The corrector estimates a trend based on Local Minima + LOESS (smoothed low-percentile intensity), and subtracts it from the raw data intensity values.

Additional parameters**method**

"**loess**" (smoothed low-percentile intensity) or "**approx**" (linear interpolation).

bw

The bandwidth to be passed to loess.

breaks

Number of breaks set to m/z values for finding the local minima or points below a certain quantile of intensities; breaks -1 equally spaced intervals on the log(m/z) scale.

break width (number of scans)

Width of a single break. Usually the maximum width (in number of scans) of the largest peak.

⚠ Overrides "**breaks**" value.

qntl

If 0, find local minima; if >0 find intensities < qntl*100th quantile locally.

Requirements

⚠ This module relies on the local installation of R statistical computing software and several R packages.

Quick install

The whole thing can be setup as follows by running the following code in R:

```
install.packages(c("Rserve", "ptw", "baseline", "hyperSpec"))
source("http://bioconductor.org/biocLite.R")
biocLite("PROcess")
```

Details

- Rserve (All correctors): provides an interface between MZmine and R. To install Rserve from CRAN packages run R and enter:
install.packages("Rserve")
- ptw (Asymmetric corrector): parametric time-warping provides the asymmetric least-squares implementation. To install ptw run R and enter:
install.packages("ptw")
- baseline (RollingBall and PeakDetection correctors): provides a trend based on "Rolling Ball" and "Peak Detection" algorithms implementation. To install baseline run R and enter:
install.packages("baseline")
- hyperSpec (RubberBand corrector): provides a trend based on "Rubber Band" algorithm (which determines a convex envelope for the spectra) implementation. To install hyperSpec run R and enter:
install.packages("hyperSpec")
- PROcess (Local Minima + LOESS corrector): provides the local minima search + LOESS (smoothed low-percentile intensity) implementation. To install PROcess run R and enter:
source("http://bioconductor.org/biocLite.R") biocLite("PROcess")

References

- [1] Boelens, H.F.M., Eilers, P.H.C., Hankemeier, T. (2005) "Sign constraints improve the detection of differences between complex spectral data sets: LC-IR as an example", Analytical Chemistry, 77, 7998 - 8007.
- [2] Rserve "A TCP/IP server which allows other programs to use facilities of R"
<https://rforge.net/Rserve>.
- omokshyna

 November 24, 2022 21:13:07

 omokshyna

11.3.4 Align scans (MS1)

Description

≡< Raw data methods → Raw data filtering → Align scans (MS1)

This module aligns scans for small fluctuations correlating consecutive scans.

Parameters

Suffix

This string is added to filename as suffix.

Horizontal scans

Number of scans to be considered in the correlation (to the left and to the right of the scan being aligned).

Max Vertical Alignment

Maximum number of shifts to be compared. This depends on equipment, normally this should be 1.

Minimum height

Minimum intensity to be considered for align correlation.

If chromatogram height is below this level, it is not used in the correlation calculation.

Correlation in log

Transform intensities to *log* scale before comparing correlation.

Remove previous files

Remove processed files to save memory.

omokshyna

⌚ November 24, 2022 21:13:07

👤 omokshyna

11.3.5 Scan smoothing (MS1)

Description

≡ Raw data methods → Raw data filtering → Scan smoothing (MS1)

This module averages intensity values within a time-scan frame.

Parameters

Suffix

This string is added to filename as suffix.

Time (min)

Time span over which intensities will be averaged in the same m/z over scans.

⚠ The max between this and Scan Span will be used.

Scan span

Number of scan in which intensities will be averaged in the same m/z.

⚠ The max between this and Time Span will be used.

m/z tolerance

m/z range in which intensities will be averaged.

⚠ The max between this and m/z points will be used. If both equal 0 no m/z smoothing will be performed.

m/z min points

Number of m/z points used to smooth.

⚠ The max between this and m/z tol will be used. If both equal 0 no m/z smoothing will be performed.

Min height

Minimum intensity of the highest data point in the chromatogram.

If chromatogram height is below this level, it is not used in the average calculation.

Remove previous files

Remove processed files to save memory.

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [omokshyna](#)

11.4 Mass detection

11.4.1 Mass detection

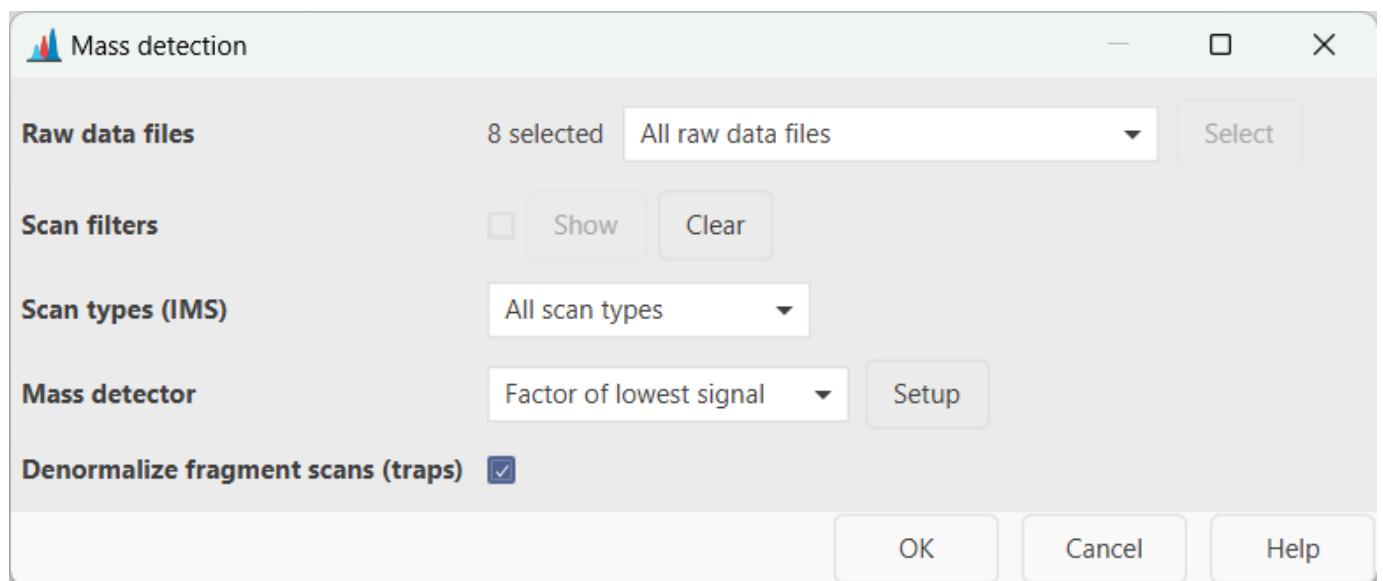
Description

≡ Raw data methods → Mass detection → Mass detection

The mass detection module generates a [mass list](#) (*i.e.* list of m/z values and corresponding signal intensities) for each scan, in each raw data file. During the mass detection, noise filtering is performed based on a user-defined threshold (see [Setting the noise level](#)). Additionally, profile raw data are centroided.

The available algorithms are described [here](#).

Parameters



Raw data files

Select the input raw data files for the mass detection. All the imported data files can be processed in bulk (*i.e.* *All raw data files*), or a subset can be selected directly from the *MS data files* panel (*i.e.* *As selected in the main window*) or based on the filename (*i.e.* *File name pattern*). As an alternative, the files' directory can be also specified (*i.e.* *Specific raw data files*). Finally, if the *Those created by previous batch step* option is selected, MZmine takes the output of the last processing step as input. This option is only available for [batch processing](#).

Scans

Select (or filter out) the MS scans to be processed. Several filters are available (*Select filters* button). A scan number, RT and mobility range can be set (*i.e.* *Scan number*, *Retention time* and *Mobility* options); only the scans belonging to the defined range(s) will be processed. The *Base Filtering Integer* option allows to process one every-N scans. The *Scan definition* field can be used to filter scans based on the scan's description normally included in the raw file's metadata (*e.g.* FTMS). Scans can also be filtered by *MS level* (*i.e.* 1, 2, ..., N), polarity and spectrum type (*i.e.* Centroided, Profile and Thresholded).

Scan types (IMS)

This parameter applies only to IM data and determines if *mobility scans*, *frame scans* or both (*i.e.* *All scan types*) are processed. For more details about *mobility* and *frame scans*, see [here](#).

Tip. Since *frame scans* are obtained by merging multiple *mobility scans*, the noise thresholds will likely be different. However, only one noise level can be set per processing step. Therefore, if one wants to run the mass detection for *mobility* and *frame scans* using different noise cutoffs, two module calls are required. As an alternative, mass detection can be performed only

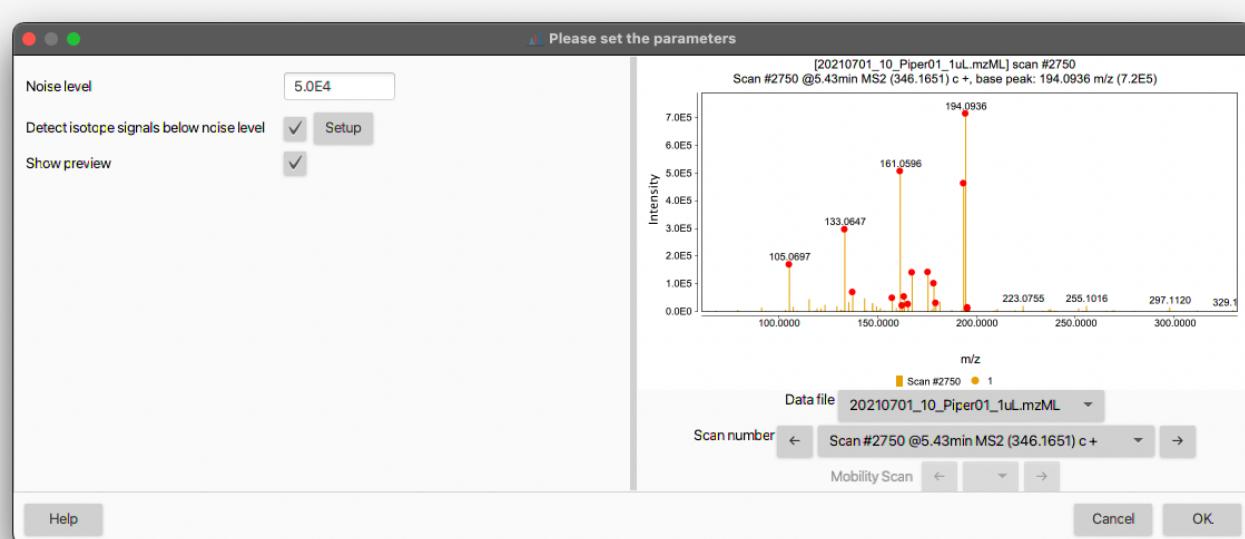
on the *mobility scans* by selecteing the appropriate noise level. *Mobility scans* can then be merged into *frame scans* with a dedicated module.

Mass detector

Select the algorithm to be used for the mass detection. Several mass detection algorithms are available and can be selected in the drop-down menu. The choice depends on the raw data characteristics (profile/centroded, mass resolution, etc.). The *Centroid* algorithm must be used for already-centroded data. A step-by-step guide to convert profile into centroded data is provided in the [GNPS documentation](#). Other algorithms are available for profile raw data and are described in more details [here](#). The *Exact mass* algorithm is highly recommended for profile HRMS data. When *Auto* is selected, the *Centroid* and *Exact mass* algorithms are used by default for centroded and profile data, respectively.

Setting the noise level

All the mass detection algorithms allow to set a threshold for the noise filtering (*i.e.* *Noise level*) by hitting the *Setup* button next to the *Mass detector* field. A dialog box like the following will open up:



The noise threshold can be entered either in standard or scientific notation. By checking the *Show preview* box, an interactive visualization panel will open to help the user to adjust the noise level (see also [How do I determine the noise level in my data?](#)). The red dots denotes the mass signals retained in the mass list according to the set threshold. Different data files and scan numbers can be visualized using the corresponding drop-down menus.

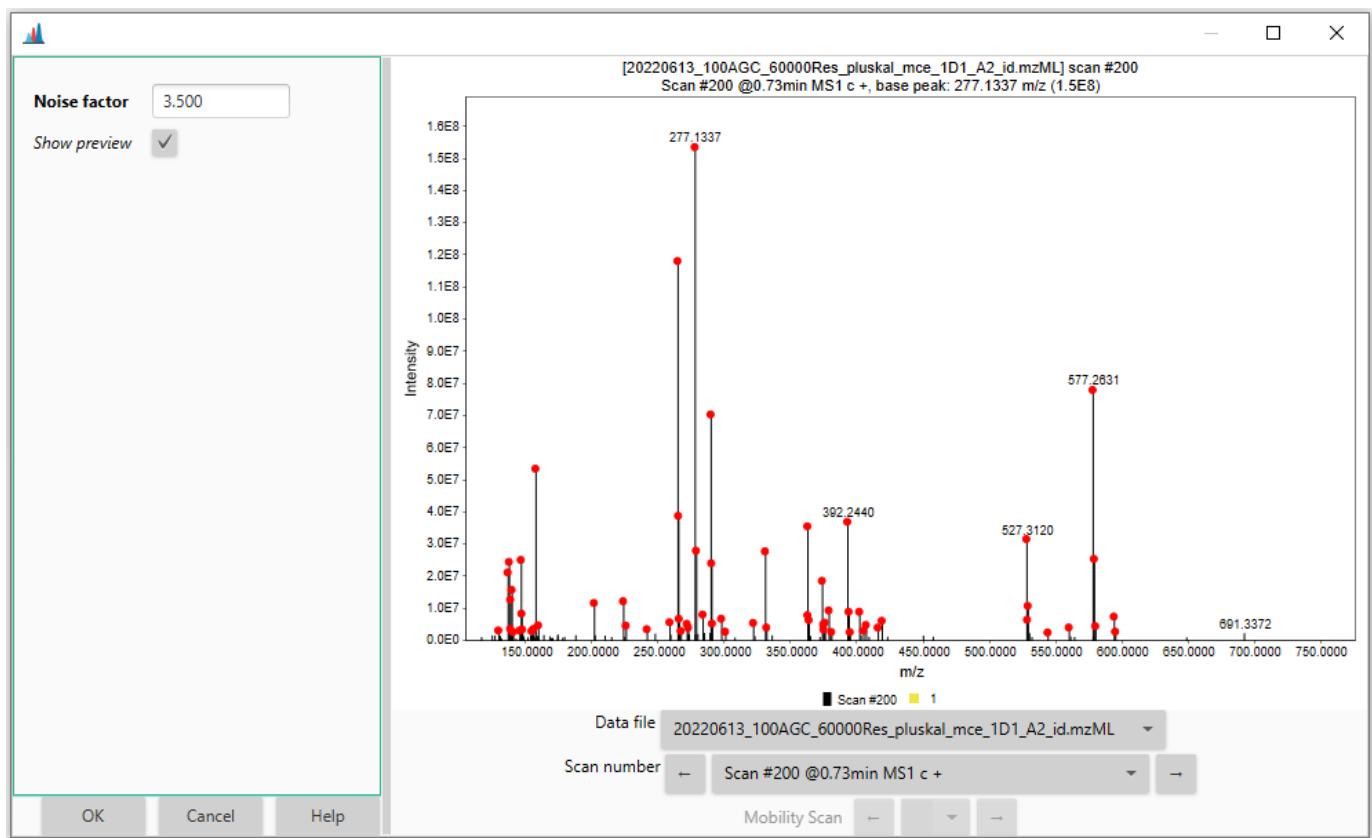
Denormalize fragment scans (traps)

This option denormalizes scans with MS level > 1, acquired with trap instruments, including Orbitrap or trapped ion mobility spectrometry (TIMS). When enabled, each scan is multiplied with the injection time resulting in "raw" intensities. If the injection time is missing in the data file, the scans remain unprocessed. This option is encouraged before spectral merging to reduce the intensity differences between scans acquired with different injection times.

How do I determine the noise level in my data?

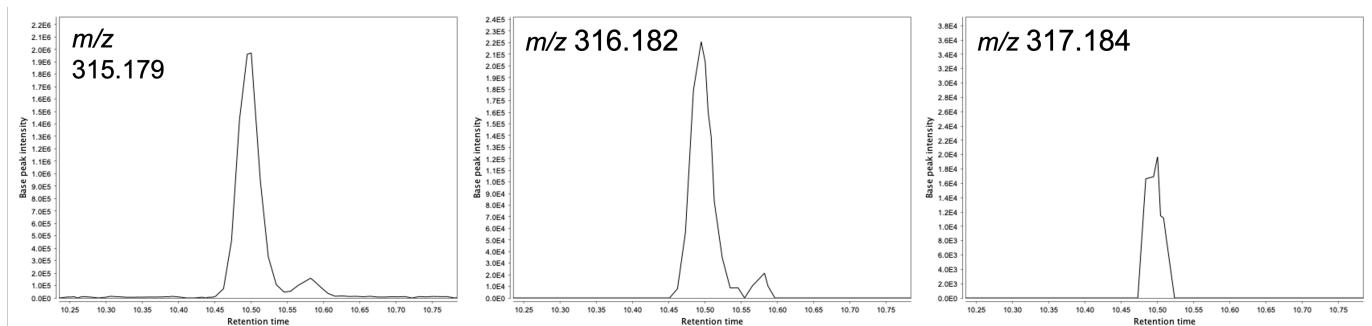
The background noise level largely depends on the mass spectrometer and detector type. For example, Orbitrap instruments normally provides higher signal intensities than TOF devices. To provide some numbers, while 1.0E2 - 1.0E3 could be an appropriate noise level for TOF analyzers, the same would be overly low for Orbitrap instruments (which normally require 1.0E4 - 1.0E5).

The best way to determine the instrumental noise level is undoubtedly by looking at the raw data. The background noise (often referred to as "grass" in technical jargon) is characterized by several signals having the same intensity and no clear pattern among them (see Figure).



Since these signals are produced by electrical and/or mechanical noise, rather than actual ions, they should be excluded from the mass detection and downstream data processing. The red dashed line in the figure corresponds to a hypothetical noise level (1.0E4 in this case) that would filter out most of the "grass"-type noise from the mass detection.

Another way, more relevant for the feature detection, to determine the noise level consists of picking a mid-intensity LC peak and extract the EICs of its 13C isotopes. When the chromatographic peak shape starts to deteriorate, it means we are approaching the instrument detection limit (see Figure).



Such approach can also be useful to determine other parameters in the feature detection such as the **Group intensity threshold** and **Min highest intensity** parameters in the [Chromatogram builder](#) module.

Robin Schmid, Steffen Heuckeroth, corinnabrungs, omokshyna, tdamiani

⌚ July 30, 2025 10:13:34

👤 Robin Schmid, Steffen Heuckeroth, corinnabrungs, omokshyna, tdamiani

11.4.2 Mass detection algorithms

Mass detection can be done with the following six algorithms:

- Factor of lowest signal
- Centroid
- Exact mass
- Local maxima
- Recursive threshold
- Wavelet transform
- Auto

Factor of lowest signal

 This mass detector is suitable for centroid or profile data.

Choose the **Factor of lowest signal** for both centred or profile data where each spectrum contains noise signals, often represented by many signals at the same low intensity (spectral grass). This may correspond to static noise or single counts. This algorithm removes all data points below a spectrum's lowest intensity multiplied by a factor. If the spectra are free of noise, for example because they were already filtered, use an **absolute intensity** threshold.

Centroid

 This mass detector is suitable for already centred data.

Centroid algorithm assumes that each signal above a given noise level is a detected ion. Here, the noise level is set as an absolute fixed value for each MS_n level.

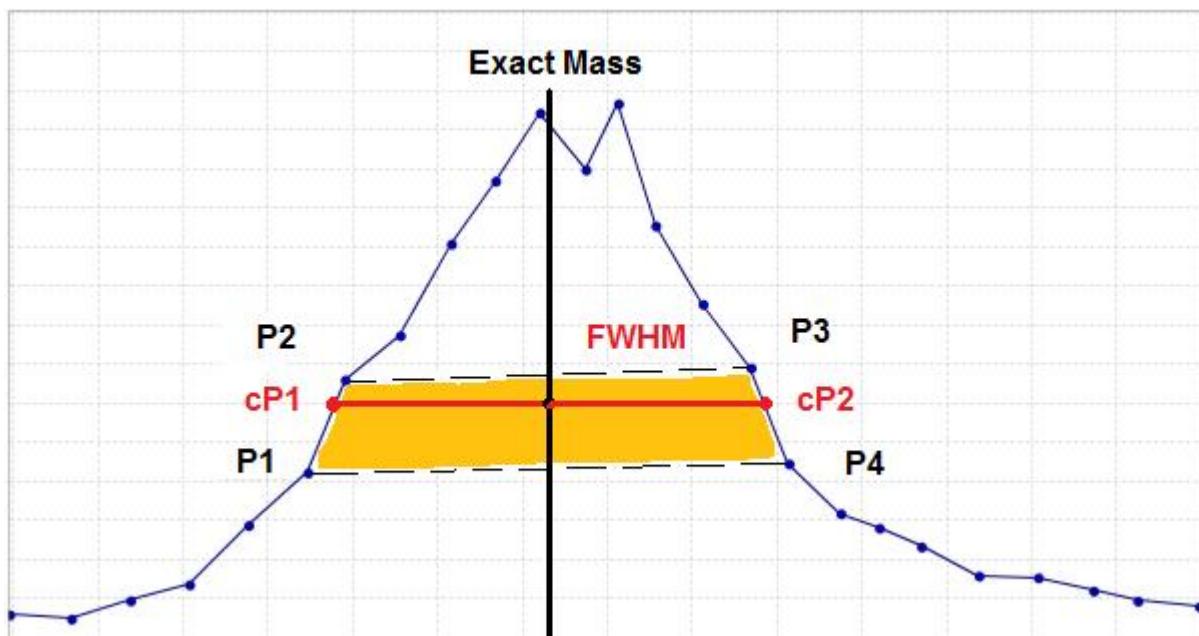
Exact mass

 The exact mass algorithm is highly recommended for profile MS data.

This mass detector first searches for all local maxima within the spectrum, which then form candidate ions.

This method calculates the exact mass of a peak using the **FWHM** (full width at half maximum) concept and linear equation ($y = mx + b$). FWHM is the difference between the two values of the independent variable at which the dependent variable equals half of its maximum value.

First, the method locates the data points located nearest to the peak center at half of the maximum intensity (P₁, P₂, P₃, P₄). With these four points it calculates two points (cP₁, cP₂) that define the width of the peak. The exact mass is then obtained as the center of the width.



This method is suitable for high-resolution MS data, such as provided by FTMS instruments.

- If the continuous data is too noisy, one can use recursive threshold algorithm.

Local maxima

This very simple mass detector detects all local maxima within the spectrum, except the signals below the given noise level. The practical usability of this method on real MS data is limited, but it is useful to demonstrate and understand the functionality of mass detection using the preview plot.

Recursive threshold

The algorithm finds all m/z ranges within the given limit in a recursive way.

Initially, it looks at the whole range of data points. If the m/z width of this range is not within given limits, a minimum data point is found and used to split the range in two parts. The same algorithm is then applied recursively on each part. Recursion continues until all m/z ranges fitting into the given width limits are found.

Final m/z values are determined as local maxima of the identified m/z ranges.

This mass detector is suitable for continuous data, which has too much noise for the Exact mass detector to be used, but which shows a consistent width of m/z peaks.

Additional parameters

Min m/z peak width: Minimum acceptable peak width in m/z.

Max m/z peak width: Maximum acceptable peak width in m/z.

- Recursive threshold method can be used with the noisy continuous data that shows a consistent width of m/z peaks.

Wavelet transform

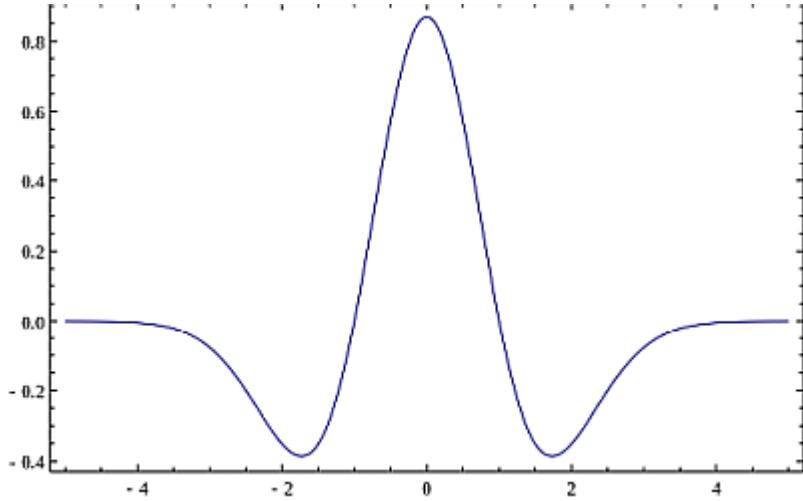
The method uses the **Mexican Hat wavelet model** of the continuous wavelet transform (**CWT**) algorithm.

The search of mass spectrum peaks is executed in three steps. First, the data point intensity is converted into wavelet domain. Second, all the local maxima of the calculated wavelet are found. Finally, m/z peaks (ions) are declared in those points, where the wavelet has a local maximum. The m/z peak is formed with the selected data point (mass and intensity) using the wavelet and all

surrounding data points. The final m/z value of the ion is calculated as an average of m/z values of surrounding data points weighted by their intensity.

Mathematical model

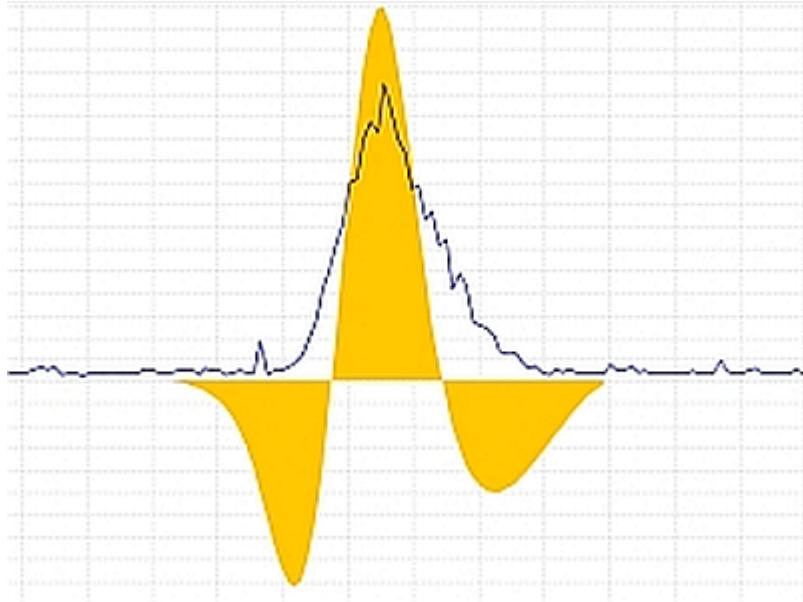
In numerical analysis, the Mexican hat wavelet is the normalized second derivative of a Gaussian function.



$$\psi(t) = \frac{1}{\sqrt{2\pi}\sigma^3} \left(1 - \frac{t^2}{\sigma^2} \right) e^{-\frac{t^2}{2\sigma^2}}$$

The parameter $\psi(t)$ is the intensity of each data point in the curve, and σ corresponds to the standard deviation.

To simplify the process of wavelet calculation, the original function is transformed into two parts, where W_c is the wavelet coefficient and y is the intensity of the wavelet at certain point. In the following formula, $\psi(t)$ corresponds to the Wavelet window size (%) parameter.



$$W_c = \left[\frac{2}{\sqrt{3}} \right] \left[\frac{1}{4} \right] \left[1 - t^2 \right] e^{-\frac{t^2}{2}}$$

$$y = \lim_{LL \rightarrow UL} (W_c x)$$

The lower (LL) and upper (UL) limits, where the Mexican Hat wavelet is evaluated, are from -5 until 5. The incremental step used in this range is the result of limits range division by 60,000.

Additional parameters

Scale level

Number of wavelet coefficients to use in m/z feature detection. Serves as the scale factor that either dilates or compresses the wavelet signal.

When the scale factor is relatively low, the signal is more contracted, which results in a more detailed resulting graph and more noisy peaks are detected. On the other hand, when the scale factor is high, the signal is stretched out, which means that the resulting graph will be less detailed with a smoothed signal.

Wavelet window size (%) The size of the window used to calculate the wavelet signal. When the size of the window is small, more noisy peaks can be detected. The proper value of this parameter may help to avoid the undesired noise peaks.

 The Wavelet transform mass detector is particularly suitable for low-resolution and noisy data.

Auto

Auto mass detector recognizes if the spectrum is of profile or centroided data type and applies centroid or exact mass algorithms correspondingly.

[corinnabrungs](#), [omokshyna](#), [tdamiani](#)

 January 22, 2024 08:55:07

 [corinnabrungs](#), [omokshyna](#), [tdamiani](#)

11.4.3 FTMS shoulder peak filter

Description

! This module should be used after mass detection step is performed.

Raw data methods → Mass detection → FTMS shoulder peak filter

Raw data obtained from FTMS (Fourier Transform Mass Spectrometer) instruments often contains false signals around high-intensity m/z peaks, called "**shoulder peaks**". These signals are residues of the Fourier Transform function and their intensity is usually below 5% of the main (true) m/z peak.

The FTMS shoulder peaks filter attempts to remove these false signals. Ions in the mass lists (generated previously by the Mass detector module) are processed in the order of decreasing intensity. A peak model (shape) is built around each ion peak using given function and resolution, and those m/z peaks which fall below the model are considered to be shoulder peaks and therefore are removed.

The method offers three theoretical peak models.

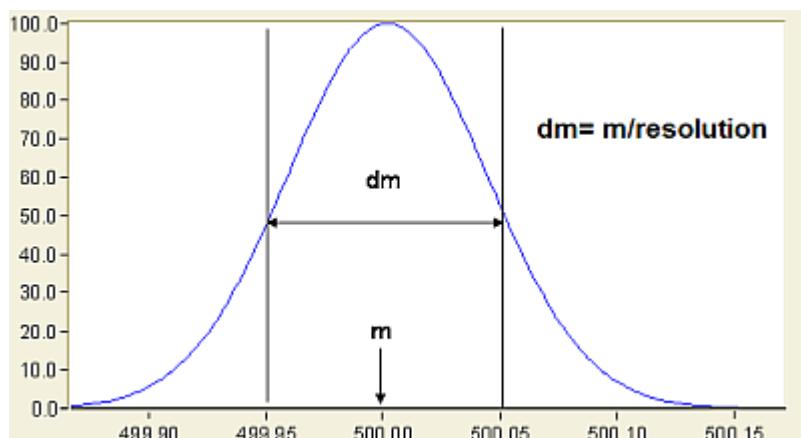
Example of running the shoulder peaks filter on LTQ Orbitrap data:



Parameters

Mass resolution of the data

Defines the width of the model, which should be equal to the estimated resolution of the peaks in the raw data. Mass resolution is the dimensionless ratio of the mass of the peak divided by its width. Peak width is taken as the full width at half maximum intensity (FWHM).



Peak model function

Defines the shape of the model function, as described below. Peaks under the curve of this peak model will be removed.

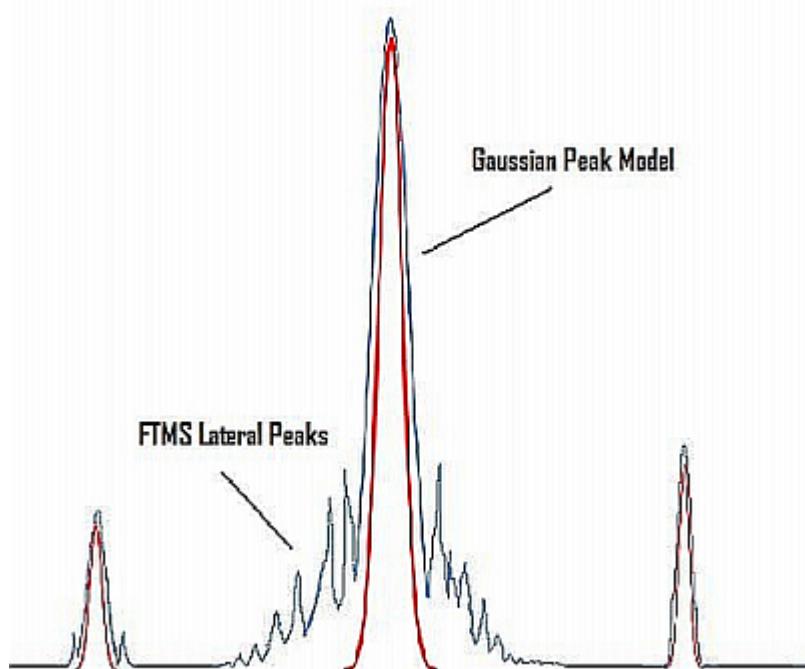
Available peak models

GAUSSIAN PEAK MODEL

The Gaussian peak model is a characteristic symmetric "bell shape curve" that quickly falls off towards plus/minus infinity, described by the following formula.

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$$

The parameter "a" is the height of the curve's peak, "b" is the position of the center of the peak, and "c" controls the width of the "bell".

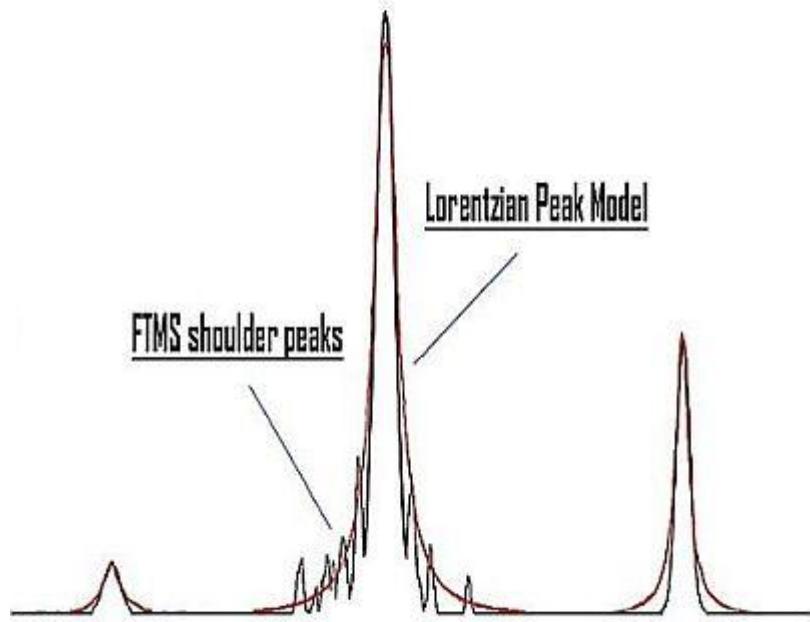


LORENTZIAN PEAK MODEL

The Lorentzian function (Cauchy-Lorentz distribution) is used for this model. The Lorentzian peak model is described by the following formula:

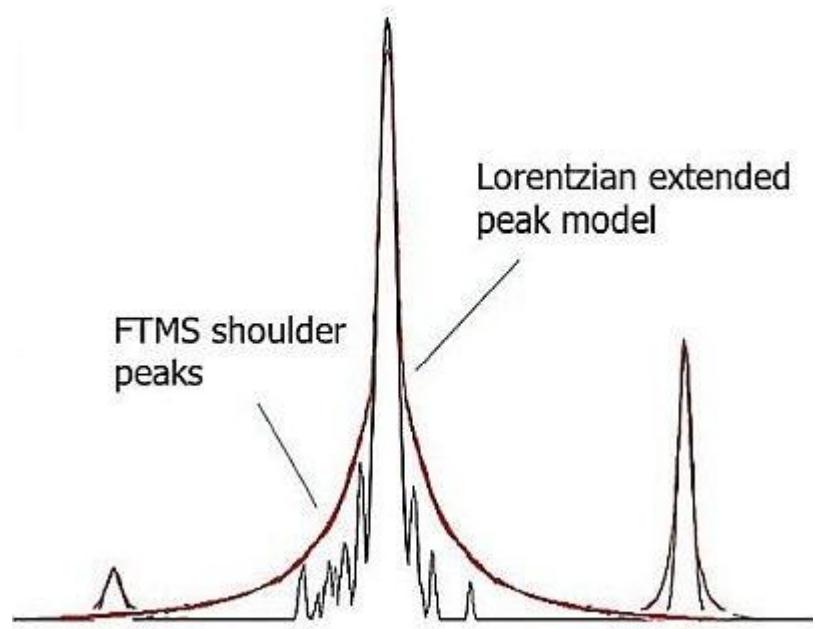
$$f(x; x_0, \gamma, I) = \frac{I}{1 + \frac{\gamma^2}{4} \left(\frac{(x - x_0)^2}{\gamma^2} + 1 \right)}$$

Where "x0" is the location parameter, specifying the location of the peak of the distribution, and "y" is the scale parameter which specifies the width of the peak.



[LORENTZIAN EXTENDED PEAK MODEL](#)

This model uses the same mathematical formula as the Lorentzian peak model, but the lower part of the model (below 5% of the intensity) is extended. The width of the peak below 5% intensity is calculated from another Lorentzian peak with 5% of the resolution of the main peak.



⌚ September 23, 2022 17:08:14

👤 omokshyna

11.4.4 Mass calibration

Description

! This module should be used after mass detection step is performed.

≡ Raw data methods → Mass detection → Mass calibration

Mass calibration module estimates the measurement error of each detected mass and calibrates them using reference libraries of ions through three main processes:

1. Peak matching with library of reference and extraction of errors,
2. Estimation of overall mass bias, and
3. Mass calibration of detected masses.

Parameters

Raw data files

The raw datafiles to calibrate. Each selected datafile is calibrated independently in a separate task.

Mass list name

Name of the mass lists to be calibrated. The mass lists must be previously generated for each scan by the Mass detector module.

Intensity threshold

Determines the intensity of the peaks used for matching against the library of ions and subsequent mass calibration. Only mass peaks with intensity above this threshold will be considered for calibration. This is useful to avoid certain noise peaks that could have been picked in the Mass Detection module. To consider all peaks, the Intensity Threshold needs to be set with a value equal or lower than the previously used in the Mass Detection module.

Duplicate Error Filter

Removes duplicate ions with the same m/z value independently of their retention time. If enabled, for a specific detected exact mass present in different scans (not necessarily consecutive), only a single ion with that exact mass value will be considered for calibration. This filter performs for the full list of masses and does not consider RT difference between ions.

Reference Library of ions

Selects the library used for ion matching and determination of mass errors. SCL and UCL libraries are available.

• SCL-only parameters

Standard Calibrant Library file

File with a list of ion formulas and retention times (xls, xlsx and csv (tab, comma or semicolon delimited files are supported)). This list should contain ions that are expected to be detected in the samples. Files need to contain a first column with the retention time in minutes and a second column with the ionic formula strings. Optionally, column 3 is used as a name of the standard and column 4 is used the m/z. If only a m/z but no name shall be provided, column 3 must exist, but be empty. All rows in the file must have the same length. Sample standards list file:

```
1.4,C3H7COO-,butyric acid,1337.1337
1.4,C3H7COO--,1337.1337
1.4,C3H7COO--,
```

Retention time tolerance (only for SCL)

Maximum difference in retention time between an actual measured ion and a calibrant to consider a match.

• UCL-only parameters

Ionization mode

Ionization mode for which an appropriate universal calibrants list is used.

m/z tolerance

Maximum allowed difference in m/z between an actual measured ion and a calibrant to consider a match.

Overall Mass Bias Estimation

Measurement mass errors are calculated based on the matching of detected ions against the library, a distribution of errors is built, and measurement bias is estimated. Because not all ion matches can be considered as correct, the calibration model will automatically identify the high-density mass error range (mass error range with larger number of matches) from the generated distribution of matching errors.

 If both parameters (*Primary High-Density Range of Errors size* and *Error range tolerance*) are set to zero, all errors obtained after matching against the reference library of ions are used for calibration of peaks.

- **High-Density Range of Errors**

Primary High-Density Range of Errors size

Determine the range (in PPM) containing most mass errors after matching the detected ions with the ion calibration library. Use zero to skip this step, in such case the distribution is split into subranges containing all the errors within the error tolerance and the largest subrange is used.

Error Range Tolerance

Maximum distance (in PPM error) between the maximum and minimum thresholds of the Primary High-Density Range of Errors and the consecutive error to allow the extension of the error range. Determines how far the distribution range of errors will be extended to extract the errors used for mass calibration. The module will include any consecutive matched error from the most populated error range found within the established error range tolerance (i.e., 0.1PPM).

This process continues until the algorithm does not find any consecutive error within the Error Range Tolerance value. Use zero to skip this step and no extension will be computed.

- **Percentile Range of Errors**

Percentile range

The module calculates the Interquartile Range (IQR) from the overall distribution of errors to extract those errors to be used for mass calibration of peaks. IQR can be modified (by default determined at 25th and 75th percentiles). In such case, errors distributed below 25th or above 75th percentile will not be considered for mass calibration of peaks.

Mass Calibration method

Method used for mass calibration. Described in more details [below](#).

- **Arithmetic mean**

- **KNN regression**

- *Nearest neighbors' percentage*

Percentage of nearest neighbors used for error prediction.

- **OLS regression**

- *Polynomial degree*

The degree of polynomial trend used, the summand powers of the polynomial will be the OLS regression features. Use 0 for constant component, 1 for linear, 2 for quadratic and so on.

- *Exponential feature*

When selected, an exponential feature $\backslash(\exp(x/10)\backslash)$ is included.

- *Logarithmic feature*

When selected, logarithmic feature $\backslash(\ln(x)\backslash)$ is included.

REFERENCE LIBRARY OF IONS

First, mass lists (MS1) from raw data files are matched against a reference library and mass measurement errors are calculated.

This module can support two different matching strategies:

- **Standard Calibrant Library (SCL)** (recommended method): file needs to be provided by the user in xls, xlsx or csv format). The file needs to include retention time (RT) information and the ion formula of a collection of ions that are expected to appear in the samples analyzed with a known chromatographic method. Ion formulas format of the SCL will depend on the ionization method used to analyze the samples. See below in the “Parameters” section a detailed description for the library format.

Using a SCL, the matching of measured peaks the matching of measured peaks against the library is performed using both RT and exact m/z. RT and m/z error tolerances are needed to be defined.

 Note: To ignore RT parameter when using SCL for mass calibration, the parameter can be set with an equal or larger value of the chromatography length (i.e., 30 min).

- **Universal Calibrant Library (UCL)** include a collection of ions often found in mass spectrometry experiments. The module has two different lists in positive (+ve) and negative (-ve) ionization modes to be chosen by the user.

Universal calibrant lists are based on **Keller et al. 2008 Analytica Chimica Acta 627:71-81** and **Hawkes et al. 2020 Limnology and Oceanography Methods 18:235-258**. MZmine includes both those libraries.

Matching of detected ions against the UCL library is performed using m/z values alone independently where the ions are appearing along the chromatography and only m/z error tolerance will be needed.

More details on mass calibration method

 To estimate mass measurement bias more accurately, we can model the trend exhibited by the error size vs m/z value relation obtained by matching the mass peaks. With the estimation model we can shift/calibrate the mass peaks at different particular m/z values more accurately.

The module supports two main modes for mass calibration:

- Arithmetic mean:

This method uses the arithmetic mean of the extracted errors of the overall bias estimation.

Calibration of peaks will be performed globally based on a single overall bias value. This method is especially recommended for datasets with low number of extracted errors (i.e., blank samples).

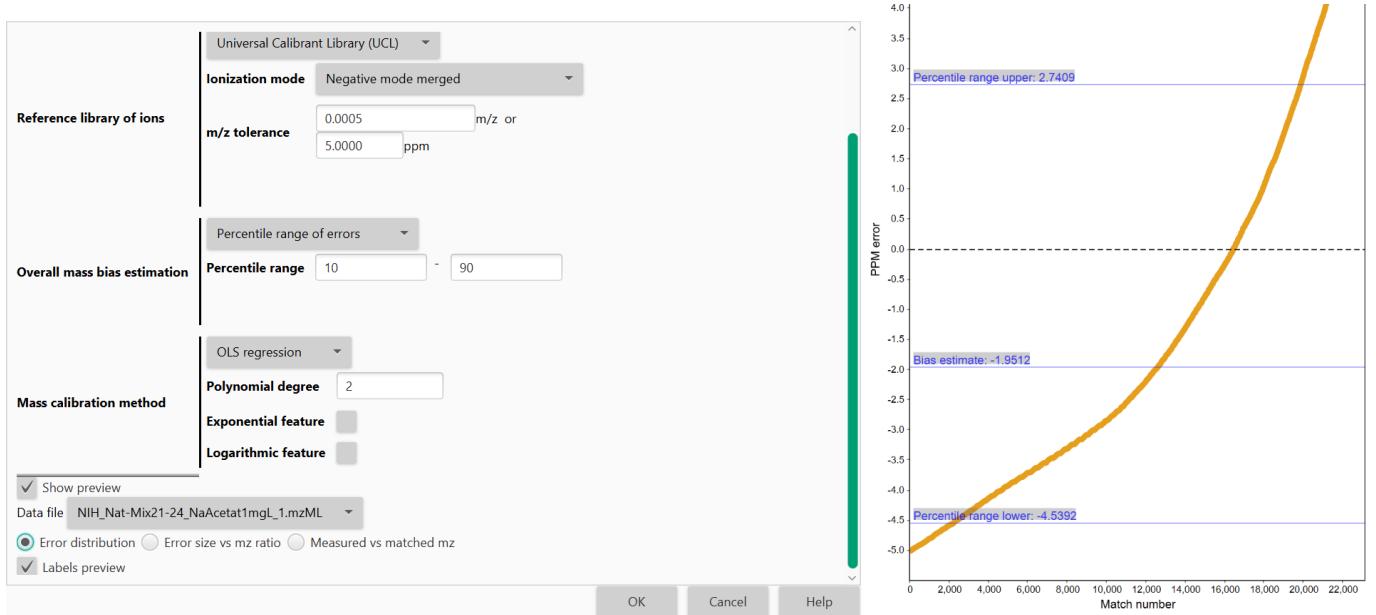
- Regression mode:

This mode models a trend from the direct relationship of error size (in PPM) vs. measured m/z of detected peaks. Mass lists will be calibrated according to the estimated model. The mass calibration module supports two different methods of regression: **OLS** and **KNN**.

- **OLS (ordinary least squares)** regression minimizes the mean squared error between the predicted trend and the datapoints in the dataset. Available features include power features (polynomial trend), logarithmic feature and exponential feature. **By default**, linear trend is fitted. This mode is suitable for datasets with enough data and exhibit a clear and strong trend (Error vs. m/z).
- **KNN (K-Nearest Neighbors)** regression finds the average value of the K nearest neighbors. In this module, the number of neighbors is defined by a percentage set by the user of all the errors present in the dataset. The K closest neighbors are thus found by the absolute difference of the m/z values within such percentage. Then the arithmetic mean of the neighbors' errors is calculated for each individual error and will serve as an error estimate for a specific m/z. This method is suitable for datasets with enough data and a trend between mass error vs. m/z is not clear. Therefore, KNN regression allows the trend to match the dataset closely without introducing additional assumptions on how the variables are related.

 Overfitting problems at large m/z values (>800) can occur when modeling the errors with regression as those regions commonly have less matches against the reference libraries of ions. If regression is modelled mainly by matches with small m/z ions (<400), it is recommended to use the arithmetic mean for mass calibration. This also applies to blank samples were the number of matched ions is commonly low and regression can produce overfitting problems.

Examples



This module was initially created during a GSoC 2020 project with MZmine by Łukasz Fiszer and MZmine team.

Steffen Heuckeroth, omokshyna

⌚ January 14, 2025 09:19:53

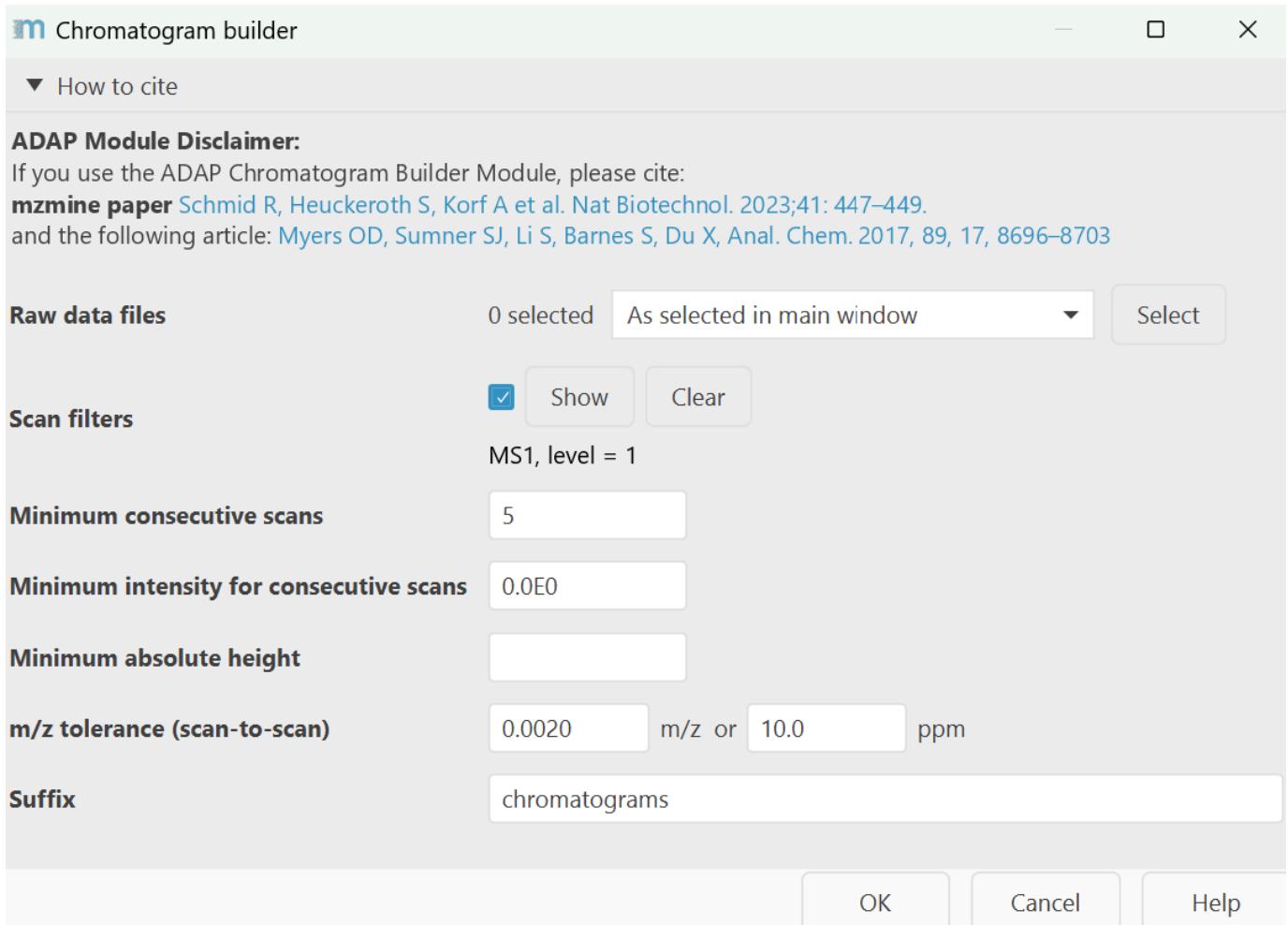
👤 Steffen Heuckeroth, omokshyna

11.5 LC-MS feature detection

11.5.1 Chromatogram builder

Description

≡ Feature detection → LC-MS → Chromatogram builder



The *Chromatogram builder* module is one of the LC-MS feature detection algorithms provided by mZmine. The module essentially builds an [EIC](#) for each *m/z* value that was detected over a minimum number of consecutive scans in the LC-MS run. Each data file is processed individually. The [mass list](#) associated to each MS1 scan in a data file (see [Mass detection](#) module) are taken as input and a [feature list](#) is returned as output. Since a mass list must be available, the *Mass detection* module must be run first.

The *Chromatogram builder* algorithm operates as follows:

- Only MS1 scans are processed.
- All the data points are extracted from all the MS1 scans in a data file and sorted in order of decreasing intensity.
- The processing starts from the most intense data point. Since no EICs have yet been created, a new EIC is initialized and associated to the corresponding m/z value.
- The processing proceeds with the second-highest data point. The corresponding m/z is checked to determine if it "belongs" to the existing EIC (based on the user-defined tolerance, *i.e.* "Scan to scan accuracy (m/z)" parameter).
- If yes, then the data point is added to the EIC and the EIC-associated m/z is updated. Otherwise, a new EIC is initialized.
- The process repeats iteratively until all the data points have been processed and a set of EICs has been created.
- Finally, the EICs are checked according to the user-defined parameters (*i.e.* minimum number of data points and intensity). The EICs matching the requirements are retained in the *feature list*, whereas the rest are discarded.

The so-built EICs can then be resolved into individual features by one of the deconvolution algorithms provided by mzmine (*e.g.* [Local minimum resolver](#) module).

Parameters

Raw data files

Select the input raw data files for chromatogram building. Mass lists associated with the data files will be automatically selected. See option descriptions in [Mass detection](#) module.

Scan filters

Select (or filter out) the MS scans to be processed. Although setting the $MS\ level = 1$ is usually enough for this module, several filters are available (see option descriptions in [Mass detection](#) module). For example, specific RT ranges (*e.g.* dead volume, equilibration time, calibration segments, *etc.*) can be excluded from the processing by setting the corresponding filter.



For polarity switching data, choose one polarity to produce chromatograms in either ion mode.

Minimum consecutive scans

Minimum number of consecutive MS1 scans where an m/z must be detected with a non-zero intensity in order for the corresponding EICs to be considered valid and retained in the feature list.

This parameter largely depends on the chromatographic system setup (*e.g.* HPLC vs UHPLC) and the acquisition rate (*a.k.a.* MS scan speed) of the mass spectrometer. The best way to optimize this setting is by manually inspecting the raw data and determining the typical minimum number of data points of the LC peaks. Usually, no less than 4-5 should be used.

Minimum intensity for consecutive scans

The minimum signal intensity that the group scans (see previous parameter) must exceed in order for the corresponding EICs to be considered valid and retained in the feature list.

A good starting point for this parameter is 3 times the noise level used in the [Mass detection](#), if the instrumental noise is used as cutoff. See also [How do I determine the noise level in my data?](#) for more details.

Minimum absolute height

Minimum intensity that the highest point in the EIC must exceed in order for the corresponding trace to be considered valid and retained in the feature list. This parameter mainly depends on the mass spectrometer characteristics (*e.g.* Orbitrap instruments normally provides higher signal intensities than TOF devices) as well as the overall goal of the processing. Overly low intensity thresholds normally leads to a larger number of background signals being retained as features, extending the overall processing time. On the other hand, overly high thresholds may lead to low-intensity features being erroneously discarded.

 A good starting point for this parameter is 7-10 times the noise level used in the [Mass detection](#), if the instrumental noise is used as cutoff. See also [How do I determine the noise level in my data?](#) for more details.

m/z tolerance (scan-to-scan)

Maximum allowed difference between an EIC-associated m/z and a new data point to be added to the existing EIC trace. It is essentially the maximum allowed mass accuracy deviation between consecutive data points in the EICs. The tolerance can be specified as absolute tolerance (in m/z), relative tolerance (in ppm), or both. When both are specified, the tolerance range is calculated using the maximum between the absolute and relative tolerances.

 This is an [m/z tolerance \(scan-to-scan\)](#), and it depends on the mass accuracy, resolution and stability of the instrument. The best way to optimize this parameter is by manually inspecting the raw data and determining the typical fluctuation of the accurate mass measurement over consecutive scans. A good starting point is 0.002-0.005 m/z and 5-10 ppm for Orbitrap instruments, while 0.005 m/z and 10-15 ppm can be used for TOF devices.

Suffix

String added to the filename as suffix when creating the corresponding feature list.

Robin Schmid, corinnabrungs, omokshyna

 July 30, 2025 10:13:34

 Robin Schmid, corinnabrungs, omokshyna

11.5.2 MSn tree feature list builder

≡ Feature detection → MSn → MSn tree feature list builder

This module is an alternative to the [Chromatogram builder](#) for creating chromatograms based on MS2 spectra or MSn trees.

1. Groups all MS2 spectra are by an m/z tolerance.
2. Adds MSn, multi-stage fragmentation, spectra to the MS2 groups, if they are present.
3. Builds an extracted ion chromatogram (EIC) for each group around their first MS2 scans precursor m/z and the m/z tolerance.
4. The EICs may be empty if the precursor m/z reported in the MS2 scan metadata does not match the MS1 data or if the m/z tolerance is too narrow.
5. Even empty EICs will be reported in the resulting feature list with their respective precursor m/z and fragmentation scans.

Parameters

Raw data files

The selected data files to process.

EIC scans

A filter to select scans (usually MS1) for building extracted ion chromatograms.

m/z tolerance (scan-to-scan)

Defines the fluctuation of an individual m/z peak in between scans. For TOF instruments, 0.005 m/z or 20 ppm is a good starting point, while Orbitraps or FT-ICRs can achieve narrower signals depending on the set mass resolution.

Suffix

This suffix is added to the produced feature list.

Robin Schmid

⌚ July 30, 2025 10:00:24

👤 Robin Schmid

11.5.3 GridMass

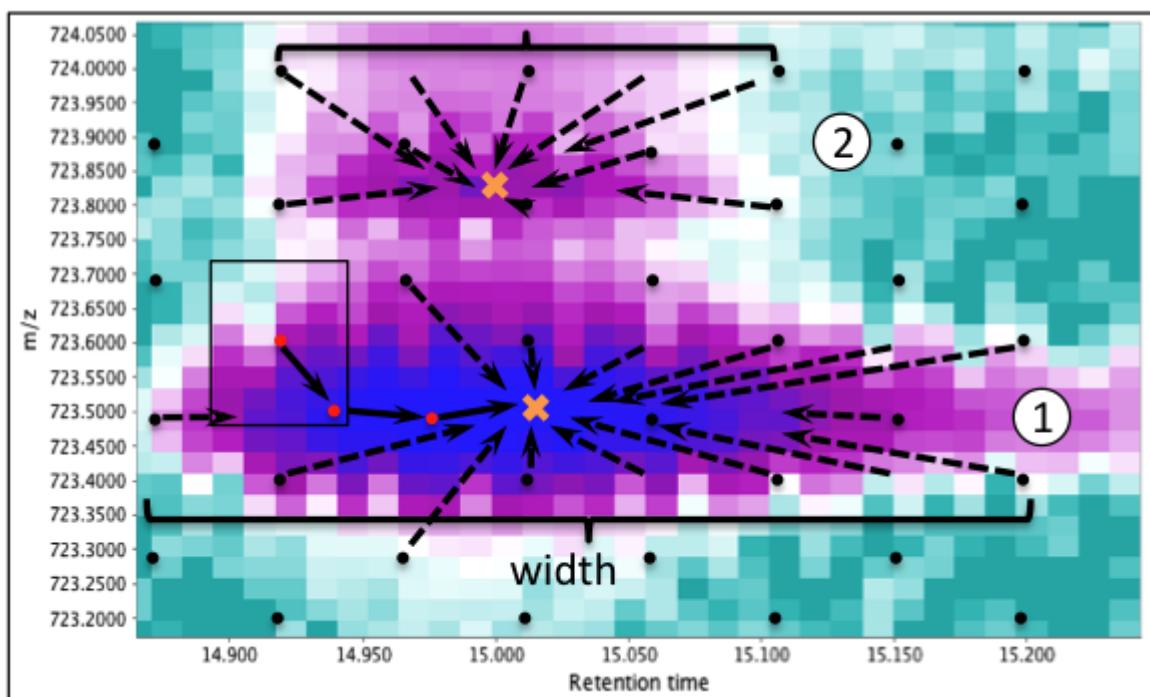
Description

Feature detection → LC-MS → GridMass

GridMass is an algorithm to detect peaks analyzing 2D data. It directly generates a peak list, which can then be operated using the peak list methods.

ALGORITHM

To detect the position and boundaries of masses, the GridMass algorithm first generates a grid of equally spaced probes covering the entire chromatographic area. A representative section is shown in the figure below.



- Each probe (black dots in Figure) explores a rectangular region around it to find a local maximum.
- The probe location is moved to the local maxima to further search for a higher value.
- The procedure is performed until no higher values exist within the exploring rectangle.
- This local maximum is then defined as a **feature**, which contains information of the m/z, the time, and the intensity detected.

A putative trajectory of a probe is shown in the figure. All probes converging to the same feature provide an estimation of its **boundaries**.

Consequently, different features represent different masses.

💡 This procedure is highly sensitive and specific for smooth surfaces. However, given that in real chromatographic data a certain level of noise and artifacts are present, in MZmine additional criteria (such as possibility to choose time windows, additional smoothing, etc.) were implemented.

A summary of the algorithm is as follows:

1. Ignore artifact spectra in time domain. In chromatography, it is typical to find a peak near the injection time, corresponding to metabolites that show no interaction with the column in the particular gradient. While this is not an artifact per se, given the myriad of signals present and the nature of the detector, the resulting peak is a strong source of artifacts that later affect analysis. To avoid this, the user may enter a list of time ranges to be ignored. The controlling parameter is ignore times whose format is time1-time2, time3-time4, ... Alternatively, the user may crop these data before processing, for example using "Raw Data→Filtering→Data Set" and ignore setting this parameter. Therefore, this step is optional.
2. Generate equally spaced probes over the mass-time space. To generate the grid, the parameters used are m/z tolerance and minimum width. The gap in the m/z dimension between probes is set to m/z tolerance multiplied by 2 or minimum to 1e-6 and are intercalated between scans. The gap in the time dimension is calculated by time associated to scans, which is estimated by the minimum width divided by 4 (down to a minimum of 1 scan).
3. Move each probe to corresponding local maximum until convergence. Each probe explores its surroundings (limited by the positions of other probes) to locate the highest intensity value, then after updating its position, it keeps exploring the surrounding until a local maximum is reached. To speed up the procedure, generate only interesting features above a certain level of noise, and limit the number of reported features, only intensities higher than minimum height threshold are considered.
4. Generate features by merging probes with similar 2D positions. Many probes will reach the same maximum that must correspond to the same feature. In addition, experimental chromatographic data is noisy and non-smooth, which may generate local maxima very close to each other. Therefore, probes whose difference in m/z is lower than the m/z tolerance and whose difference in time is lower than minimum width are merged. Then, from all probes reaching the same maximum, the m/z assigned to the feature corresponds to the highest observed intensity. The width of the feature is estimated from the probes with the lowest and highest time. To form the peak and estimate its area, the highest value in each scan is used.
5. Remove features whose width is out of a range. Features having large or very low width are likely to be artifacts. To avoid this, all features out within the range given by the parameters minimum width and maximum width (in minutes) are removed.
6. Remove features of similar mass and high cumulative times. Chemical noise or large blurs are characterized by generating many features of similar mass, similar intensities, and separated by short times. To avoid these artifacts, we merge features whose m/z difference is lower than m/z tolerance and whose intensity ratio (higher/lower) is higher than an intensity similarity ratio parameter. Once merged, the removal implemented in step 5 is performed on merged features.

Parameters

Suffix

Suffix to be added to peak list name.

Minimum height

Only intensities larger than this minimum are considered.

m/z tolerance

Maximum distance in m/z from the expected location of a peak.

Min-max width time (min)

Time range for a peak to be recognized as a 'mass'. The optimal value depends on the chromatography system setup. Check 2D raw data to determine typical time spans.

Smoothing time (min)

Time window used to smooth the signal before detection.

Smoothing m/z

m/z window used to smooth the signal before detection.

False+: Intensity similarity ratio

Ratio between features to be recognized as the same. This is highly useful to detect artifacts.

False+: Ignore times

Ranges of time to be ignored by the method. This can be avoided if the region is previously cleaned using the crop option.
Format: timeA-timeB, timeC-timeD,...

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

11.5.4 Targeted feature detection

Description

Feature detection → LC-MS → Targeted feature detection

This algorithm opens a *.csv file with a list of peaks and searches for each peak in the selected raw data file. The most crucial parameters are m/z tolerance and Retention time tolerance**, which define the window where the algorithm should find the new peak. It is centered in the m/z average and retention time average of the source peak list. Once the best candidate is found inside the window, its shape in RT direction is also checked.

The *.csv file should have three columns:

- The first column should contain the expected M/Z,
- the second column the expected RT,
- and the third the peak name.

 Each peak should be in a different row.

Parameters

Name suffix

Suffix to be added to the peak list name.

Peak list file

Path of the csv file containing the list of peaks to be detected.

Field separator

Character(s) used to separate fields in the peak list file.

Ignore first line

Check to ignore the first line of peak list file.

Intensity tolerance

This value sets the maximum allowed deviation from expected shape of a peak in chromatographic direction.

Noise level

The minimum intensity level for a data point to be considered part of a chromatogram. All data points below this intensity level are ignored.

m/z Tolerance

Maximum allowed m/z difference to find the peak

Retention time tolerance

Maximum allowed retention time difference to find the peak

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

11.5.5 Feature table



Columns in the table can be displayed or hidden on demand, using the quick column menu on the "+" ** button on the top right or the ... button in the lower right menu. More configuration options are available from the * *parameter menu button in the right lower corner.

Tip

The shapes column shows the retention time feature shapes. This column and other graphic columns may be hidden when opening larger tables with many samples. Activate any column from the quick column menu or the configuration.

The **Feature tables** include several groups of columns:

- Feature summary columns:** All columns from left until the sample-specific columns. They define summary statistics for each feature list row like the unique ID, average m/z, maximum height, total number of fragment scans, spectral library matches, lipid matches, etc.
- Sample-specific columns:** Have a sample icon and name on top and they might be hidden by default for large datasets. They can be accessed through the quick column selection (all columns with the Feature: prefix) or via configuration.
- Box plots are available based on height and area and add a dropdown selection to choose the metadata column for sample grouping. Double-click to reach the statistics dashboard.

Tip

Double-click on different cells like an annotation structure, the box plots, or the shapes to open other visualizers and tools.

- State column describes the feature status in a chosen file using several color-coded options:

- Blue - a feature was found in a primary feature detection,
- Gray - a feature was found in a secondary feature detection (e.g., using Gap filling, after alignment),
- Black - a feature was defined manually.

Configuration options

Quick column selection offers quick toggle buttons to show / hide groups of columns. Also the **Compact table** button helps to get a standard smaller table.

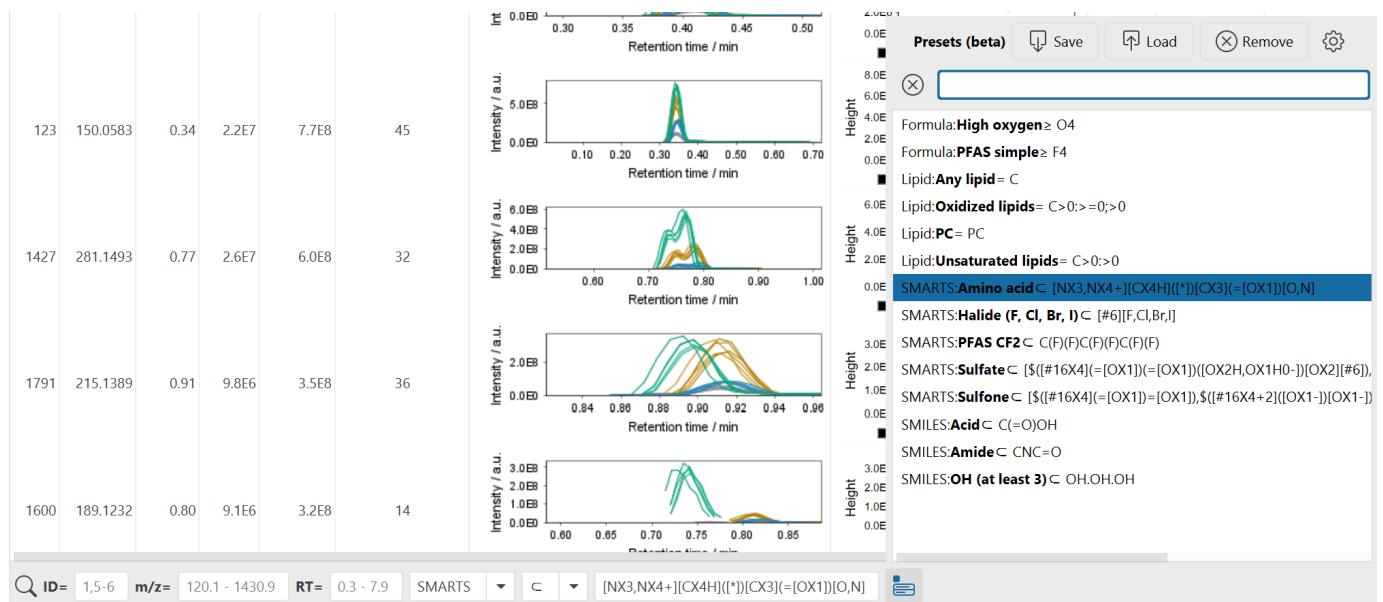
The configuration dialog provides parameters to control the **Feature summary columns** and the **Sample-specific columns**. Also, there are options for the default feature intensity and chart-specific options.

Right-click menu

Open a context menu by right-clicking on any cell in the table. The column cell may change the available options depending if the column is a **Feature summary column** or **Sample-specific column**. There are various visualizers and other tools available from this menu.

Check out the **Show/Feature overview IIMN networks** for a composition of annotations, networks, and more.

Feature list filters



In the bottom panel, there are multiple filters to focus on specific properties of your feature list rows like annotations or retention time and m/z.

- ID:** Filters rows by their ID column either by single ID, ranges (e.g., 1-10), lists (e.g., 1,2,3), or a mix 1,5-9.
- m/z:** Filter the row m/z column. A single value is interpreted as a range with this significance like 200.1 will range from 200.1 - 200.2 and 200.15 will range 200.15-200.16. Or enter ranges directly like 200 - 500.
- RT:** Filter the row retention time (RT) column. A single value is interpreted as a range with this significance like 1 will range from 1 - 2 and 2.5 will range 2.5-2.6. Or enter ranges directly like 1 - 10.

FLEXIBLE FEATURE LIST FILTER

This filter is defined by a search type, matching mode, and textual query. In general, annotation search types will match against any annotation like spectral library match, lipid match, or other feature annotations from other modules. All annotations are considered, this means that sometimes a row will match the filter but the reason is not the primary annotation visible but a secondary one that may be hidden in the list of, e.g., library matches.

This filter is also available in the **Feature list rows filter** module to apply the filter to a feature list and retain its results.

Matching modes

Matching modes define either direct equality or inequality or define minima / maxima. A special case is **C** ("contains") which is used for substructure matching and substring matches. This is usually used for substring name matching where the whole

substring needs to be contained, but there are also other substring matching modes in **all** and **any** that match each word in a query separated by space, semicolon (;), or comma (,) symbols.

Substring ("contains") matching types:

- **C**: The target contains the whole query string
- **all**: The target contains **all words** of the query, separated by space, semicolon, or comma symbols
- **any**: The target contains **any word** of the query, separated by space, semicolon, or comma symbols

Search types

- **Comment (row)**: Searches in the Comment column for each row. This column represents comments on the feature list row level. Useful for tag-based filtering in combination with substring search * *any* or *all** (see below).
- **Comment (row)**: Searches in the Comment column and additional **JSON** for each compound annotation. The comment and json columns can contain additional information like user-specific fields. This filter combines well with substring search **any** or **all**.
- **Ion identity ID**: Find all rows of the same ion identity ID
- **Ion type**: The adduct, in source fragment, or other ion type detected.
- **Compound name**: Any annotation's compound name matches, best to use a substring match
- **IUPAC name**: Any annotation's compound name matches, best to use a substring match
- **Formula**: Molecular formula, options to define a minimum formula
- **Formula range**: Molecular formula range by defining a minimum and maximum formula and the number of atoms need to be within these bounds. e.g., C6Br - OBr3 matches every formula that has at least 6C 1Br and at max 1O 3Br.
- **SMILES / InChI**: Matches exact structures or substructures defined by SMILES or InChI. For example, SMILES CNC=O matches amides
- **SMARTS**: Matches substructures and is more flexible than SMILES or InChI. [Theory Examples](#)
- **Lipid**: Lipid filter is designed for mzmine's rule-based lipid matching module but also works for other annotations like library matches, if the compound name uses the standard lipid short nomenclature. See examples below.
- **Fragment scans**: Filter for a minimum number of fragment scans in the fragment scans column, which all fragment scans across all samples.

Tag-based filtering

In the feature table, the **Comment** column provides the option to add comments or tags to each row, describing the whole row instead of just a single compound annotation. This field can be used to tag rows and the flexible feature filter below and in the feature table or in the [Feature list rows filter](#) can search for multiple tags in search modes **any** or **all**.

In the **any** or **all** search modes, the query is split into words (tags) by all space, semicolon, and comma symbols.



Enter tags into the **Comment** column to describe each feature table row. Use the Comment (row) filter with mode **any** or **all** to match tags entered into the query field. Separate tags in the query by spaces, commas, or semicolons. **Example:** Enter comment *media, impurity* in a row column and then search for tags by entering query: *impurity* or *impurity, media* (order of tags does not matter in modes any/all).

Lipid filter

Enter the following query to match specific lipid matches:

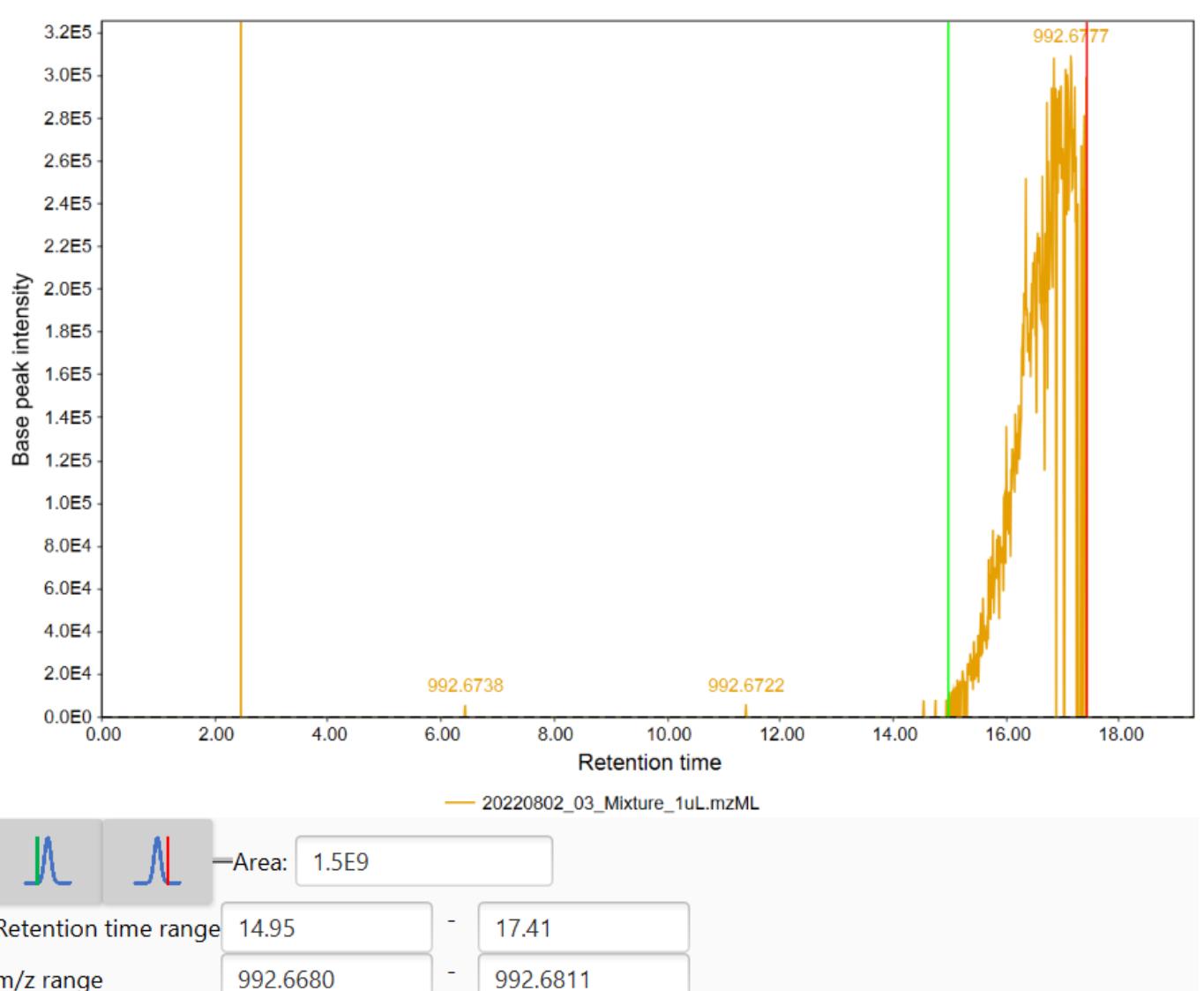
- C for any lipid class
- PC only define class
- C20:2 for 20 C and 2 double bonds
- PC20:2 also specifying the lipid class as PC
- C>20:2 for more than 20 C and exactly 2 double bonds
- C>20:>2 also more than 2 double bonds
- PC18:2_18:0 defining chains
- PC>18:>2_>18:0 defining ranges as well
- PC20:2 - PC40:6 defines ranges with min and max lipid

Manual feature integration



Manual integration was moved to the Integration dashboard that can be opened from the quick module access (CTRL + F or hit double shift). The integration dashboard can also be opened by right click on a feature list and then **Integration dashboard**.

The manual feature integration directly from the feature table is outdated and we advice to use the **Integration dashboard**. The old dialog still works though. The old dialog can be found by opening a feature table, right clicking on a cell that is within a specific sample (right side of the table) and then **Define manually**.



Robin Schmid, omokshyna

⌚ October 21, 2025 15:45:16

👤 Robin Schmid, omokshyna

11.6 LC-IMS-MS feature detection

11.6.1 IMS Expander

11.6.2 Description

≡ Feature detection → LC-IMS-MS → IMS expander.

The IMS expander searches for data points in mobility scans for existing features.

⚠ This requires prior chromatogram building (see [Chromatogram builder](#) and resolving in retention time dimension (see [Resolving](#)).

PARAMETERS

m/z tolerance

If selected, a tolerance will be applied to the feature's detected m/z while searching for data points in mobility dimension. Otherwise, the accepted m/z range is determined by the feature's m/z distribution in accumulated frame spectra.

Recommended setting: selected, 0.003 m/z and 15 ppm

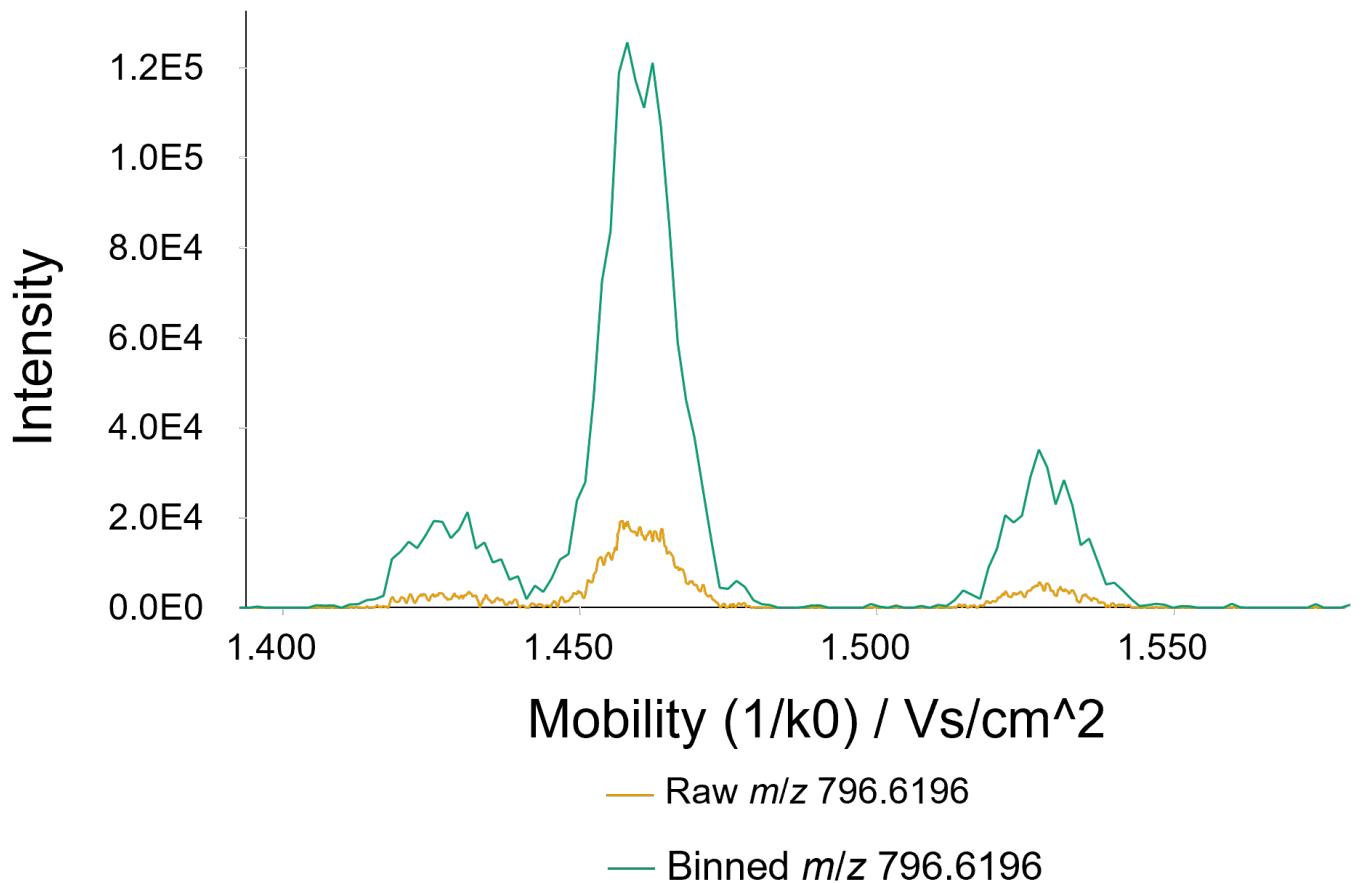
Raw data instead of thresholded

Enables searching in mobility scan raw data instead of the thresholded (=mass detected) data. Only possible for centroid raw data files.

Override default mobility bin width (scans)

If selected, the default number of binned mobility scans can be overridden. Useful for data with high mobility resolution.

BINNED MOBILOGRAM EXAMPLE



Robin Schmid, Steffen Heuckeroth, omokshyna

⌚ July 30, 2025 10:13:34

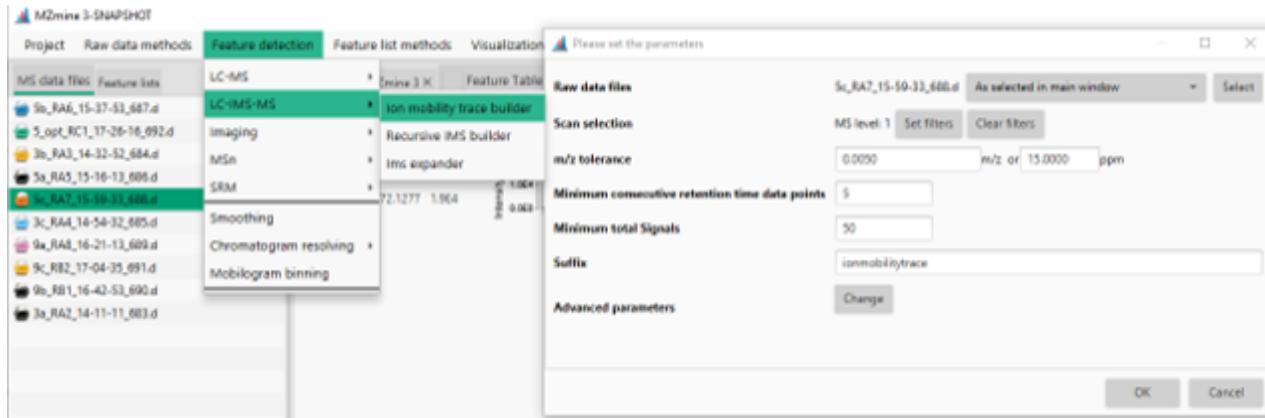
👤 Robin Schmid, Steffen Heuckeroth, omokshyna

11.6.3 Ion mobility trace builder

Description

≡ Feature detection → LC-IMS-MS → Ion mobility trace builder.

The **Ion mobility trace builder** will build ion mobility traces from the raw data. Alternatively, the **Recursive IMS builder** can be used, which requires less RAM but takes longer.



Parameters

Scan selection

The scan selection parameter specifies the scans that shall be processed for feature detection. Usually, setting the ms level to 1 is sufficient. If a calibration segment is present, it can be cut out via the retention time filter in the scan selection.

m/z tolerance

The **m/z tolerance** specifies the scan-to-scan tolerance for ion mobility traces. This tolerance window may need to be set higher than for classic LC-MS feature detection (e.g. to 0.005 m/z and 15-20 ppm instead of 10 ppm) due to lower intensities therefore less accuracy in individual mobility scans compared to LC-MS scans. Note that the overall accuracy is achieved via LC-IMS-MS is the same due to the higher number of scans.

Minimum consecutive retention time data points

This parameter specifies the number of consecutive detections of the same m/z value in a chromatographic peak (rt dimension only). This means that a single m/z has to be detected in, e.g. 5 frames with an intensity higher than zero. This parameter helps to filter noise. Consecutive detections in the mobility dimension do not affect this parameter. Usually no less than 5 should be set here if the MS1 acquisition rate is sufficient.

Minimum total signals

Specifies the total number of peaks in the mobility dimension in all mobility scans. Every "dot" in an ion mobility trace represents a single datapoint.

Advanced parameters

⚠ For most applications, these parameters do not need to be set/changed.

For high mobility resolved data the mobilograms might become noisy due to a few ions reaching the detector at the same time. By default, the number of binned scans is set to cover about 0.0008 Vs/cm² per bin. The effect of binning can be seen [here](#). If you are unsure about the nature of your data, you can perform trace building with the standard parameters and apply/preview the binning afterwards via the **Feature detection → Mobilogram binning** module.

Override default TIMS binning width (Vs/cm²)

The binning width in mobility units of the selected raw data file.

Travelling wave binning width (ms)

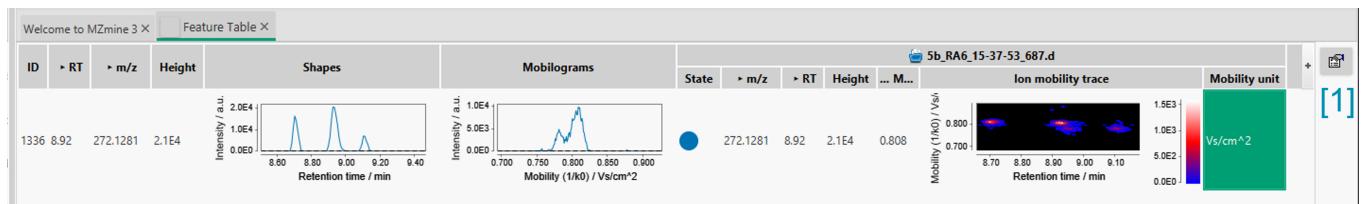
The binning width in mobility units of the selected raw data file.

Drift tube binning width (ms)

The binning width in mobility units of the selected raw data file.

Processing result

After performing ion mobiltiy trace detection, a feature list is created in the feature list tab (see [feature lists tab](#)). In the feature table, multiple columns are created. The displayed columns can be set via the button on the right of the feature table ([1]).



The **shapes** displays an EIC of the ion mobility trace (intensities summed in rt dimension). The **mobilograms** column shows a mobilogram for the ion mobility trace (intensities summed in mobility dimension). The shapes and projections can be smoothed and resolved. However, the ion mobility trace is always represented by the raw data and remains unaltered. After resolving, the shapes and mobilograms have to be recalculated from the raw data, which is why the smoothing is lost after resolving.

[Steffen Heuckeroth, omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 Steffen Heuckeroth, omokshyna

11.6.4 Recursive IMS builder

Description

≡ Feature detection → LC-IMS-MS → Recursive IMS builder.

Builds m/z traces for ion mobility spectrometry data.

Parameters

m/z tolerance

The m/z tolerance to build ion traces. The tolerance is specified as a +- tolerance.

m/z 500.000 with a tolerance of 0.01 will allow m/z change from 499.99 to 501.01.

Minimum consecutive retention time data points

The minimum number of consecutive detections in frames (retention time dimension).

Minimum number of data points

The minimum number of consecutive detections in frames (retention time dimension).

Advanced parameters

Allows adjustment of internal binning parameters for mobilograms. The default values are set to 1.

Override default TIMS binning width (Vs/cm²)

The binning width in mobility units of the selected raw data file.

Travelling wave binning width (ms)

The binning width in mobility units of the selected raw data file.

Drift tube binning width (ms)

The binning width in mobility units of the selected raw data file.

omokshyna

⌚ November 24, 2022 21:13:07

👤 omokshyna

11.7 MS imaging feature detection

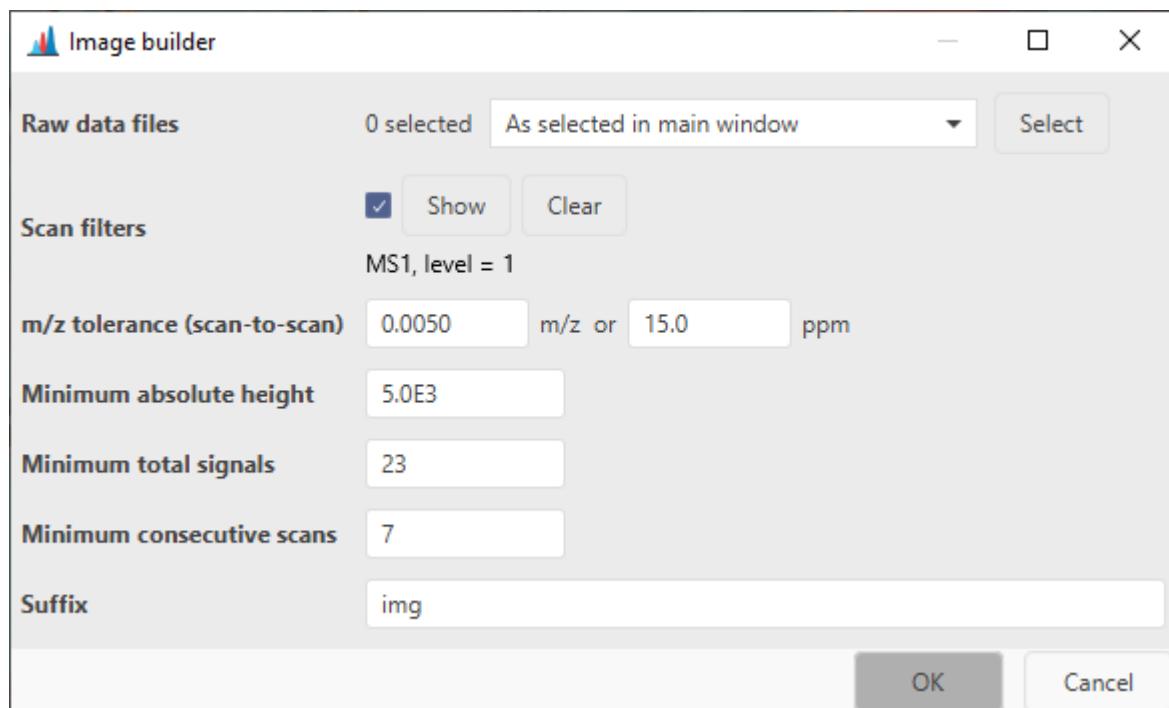
11.7.1 Image builder

Description

≡ Feature detection → Imaging → Image builder

The image builder will create ion image features from mass spectrometry (MS) imaging raw data. Similar to the [LC-MS feature detection](#), a [feature list](#) will be produced, containing images for every detected m/z.

Parameters



SCAN FILTERS

Sets the scans to be used for image generation. The default is MS level = 1 without additional filtering.

M/Z TOLERANCE (SCAN-TO-SCAN)

Defines the fluctuation of an individual m/z peak in between scans. For TOF instruments, 0.005 m/z or 20 ppm is a good starting point, while Orbitraps or FT-ICRs can achieve narrower signals depending on the set resolution.

MINIMUM ABSOLUTE HEIGHT

The minimum intensity an ion image feature must have in at least one point to be considered and retained in the final feature list.

MINIMUM TOTAL SIGNALS

The minimum amount of pixels, an ion image must be detected in. Consider your total image pixel size when setting this parameter.

MINIMUM CONSECUTIVE SIGNALS

The minimum consecutive signals for an ion image feature to be detected.

SUFFIX

The suffix for the created feature list.

Steffen Heuckeroth

 May 19, 2023 12:42:20

 Steffen Heuckeroth

11.8 MRM feature detection

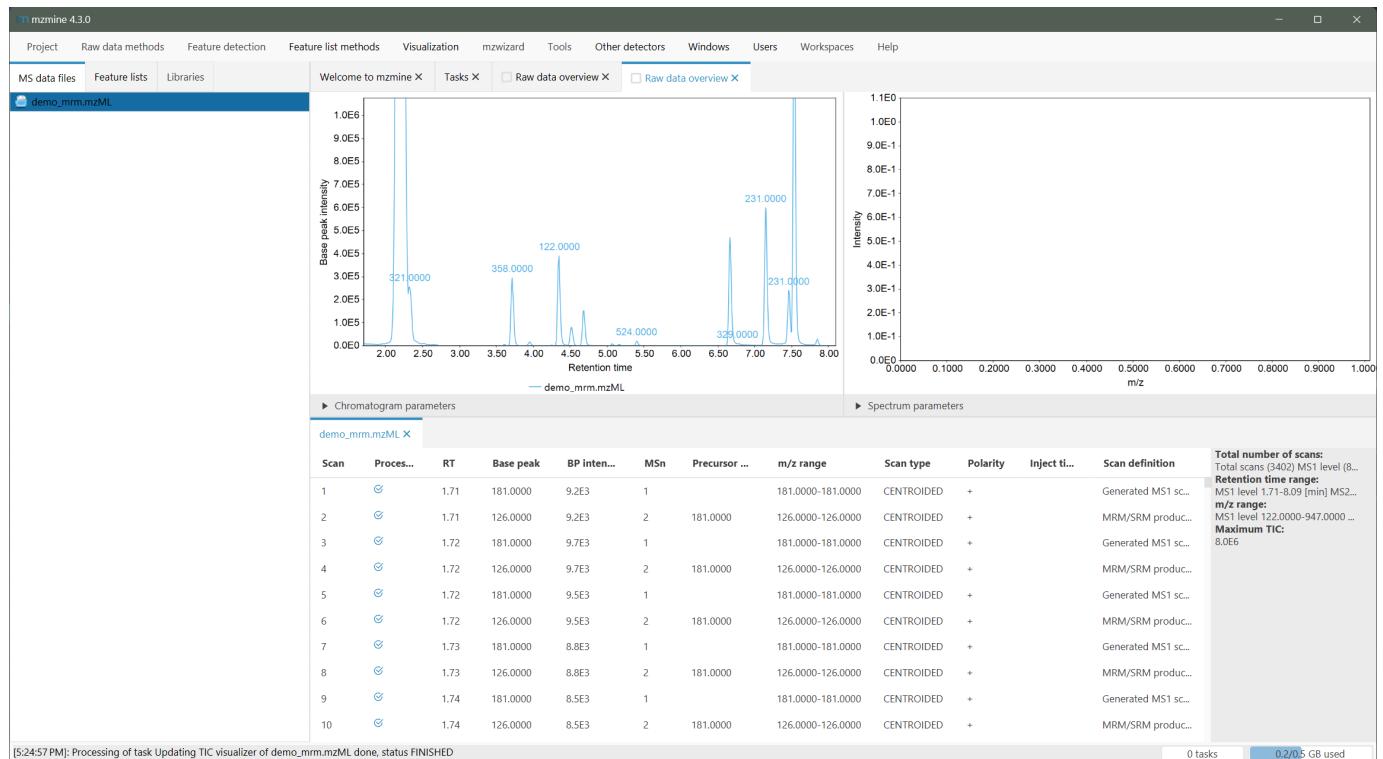
11.8.1 MRM to scans



To process multiple reaction monitoring (MRM) data, the MRM service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

≡ MRM → MRM to scans and chromatograms

Since Triple-Quadrupole MRM or SRM workflows typically produce one or more transition chromatograms per compound instead of individual spectra, the data must be pre-processed to fit into the regular mzmine workflow. This is achieved by this module.



Parameters

Raw data files

Select the raw data files that the MRM processing shall be applied to.

Build chromatograms

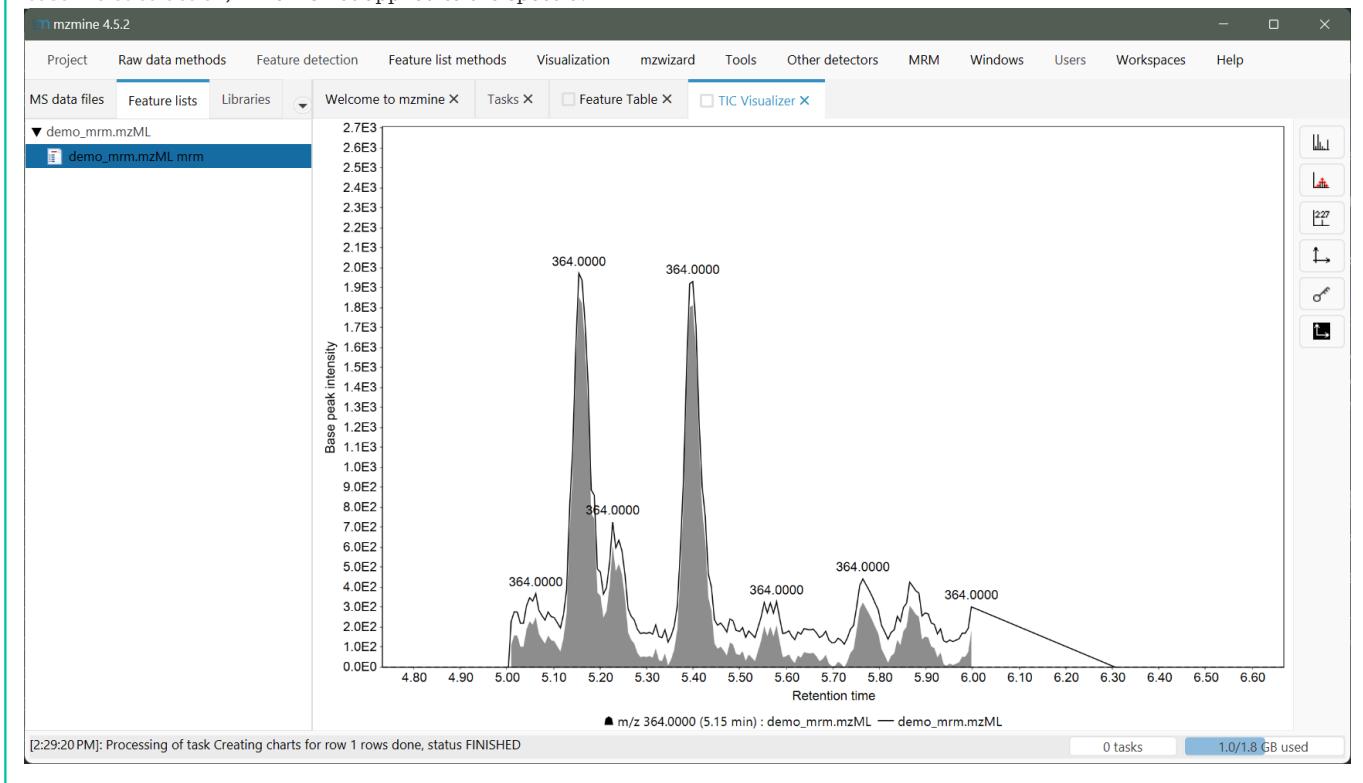
If enabled, this module automatically generates feature lists with chromatograms for the selected data files, based on the grouped transitions (see [transition grouping](#)). These features will contain the chromatographic data of all transitions of a single compound (see [chromatogram generation](#)). This is enabled by default and the recommended way to generate chromatograms for MRM data files.

Static baseline subtraction

If enabled, the lowest non-zero intensity of a transition will be subtracted along the retention time (recommended).

Tip

Visualizing features of MRM data will not "fill" the whole area of the extracted XIC, which is accessible from the feature table context menu. This is caused by the summation of all transitions to compute the MS1 intensity (see [Pseudo MS1 generation](#)) and the [static baseline subtraction](#), which is not applied to the spectra.



Algorithm

Transition grouping

During the conversion, all transitions of the same Q1 precursor m/z (+- 0.01) and within the same retention time window (at least 90% overlap) will be grouped together. For example, Q1 m/z 531.00 will be grouped with transitions 531.00 \rightarrow 429.00 and 531.00 \rightarrow 214.00, but not with 531.02 \rightarrow 212.00.

Pseudo MS2 generation

After all compounds have been grouped, the transitions of a single compound are used to generate pseudo MS2 spectra. The MS2 spectra will be generated for the retention time window, in which the set of transitions was acquired. At each dwell cycle, the respective intensities of the transitions will be interpreted as the intensity of the Q3 m/z in the pseudo MS2. E.g. if the intensity of 531.00 \rightarrow 214.00 was 150 and of 531.00 \rightarrow 429.00 was 125 at retention time 5.00 min, the MS2 spectrum with precursor 531.00 will contain two signals. One with m/z 214.00 and intensity 150, and one with m/z 429.00 and intensity 125.

Pseudo MS1 generation

After all pseudo MS2s were generated, the cycle time of the instrument is determined (which ions are fragmented in a single cycle). With this information, the generated pseudo MS2s are grouped by their cycles. To generate a pseudo MS1 spectrum, all precursor (Q1) m/z s are represented by the accumulated intensity of all their recorded transitions (= signals in the pseudo MS2). However, these pseudo MS1 and MS2 spectra are only used for visualisation purposes and to stay consistent with the mzmine data structure. The default feature detection and integration of the MRM workflow runs on the generated chromatograms (see [chromatogram generation](#)).

Chromatogram generation

Chromatograms are generated for a single grouped compound (see [transition grouping](#)). A feature list is generated from these compounds. Since a feature may now have multiple transitions, the most intense transition is set as the quantifier by default and

used to represent the feature with its chromatographic shape. Since a static background may be present, the lowest non-zero intensity may be subtracted using the [static baseline subtraction](#) parameter (recommended).

 [Robin Schmid, Steffen Heuckeroth](#)

 November 25, 2025 10:29:24

 [Robin Schmid, Steffen Heuckeroth](#)

11.8.2 Set MRM quantifiers

⚠ Warning

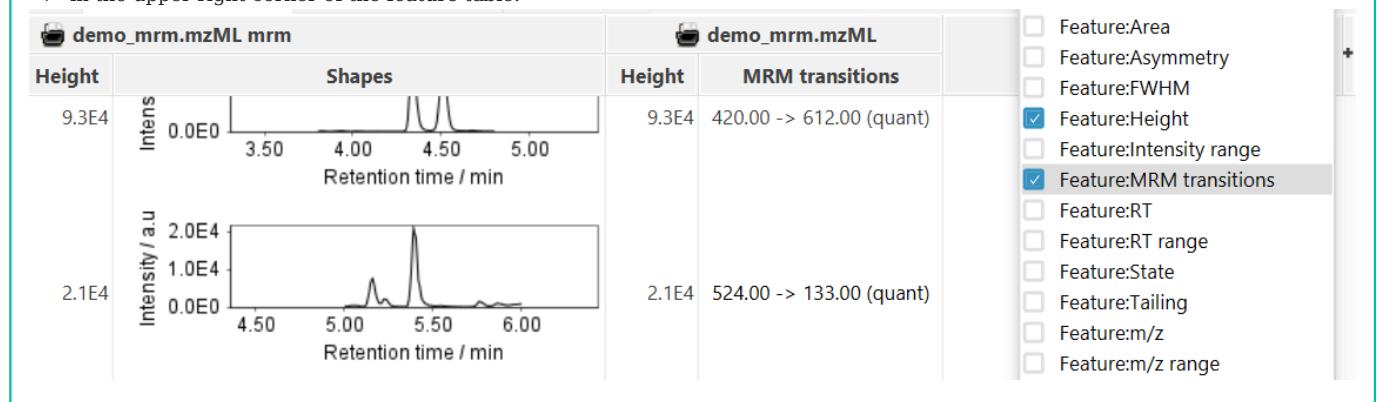
To process multiple reaction monitoring (MRM) data, the MRM service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

≡ MRM → Set MRM quantifiers

The [MRM to Scans module](#) automatically sets the most intense transition as the quantifier ion. However, some MRMs may be more challenging and shall be quantified by a different transition. Here, the Set MRM quantifiers module can help to select the specific ions needed for accurate quantification.

Tip

You can look at the current quantifier ion mass by enabling the "Feature:MRM transitions" in the feature table by selecting the small "+" in the upper right corner of the feature table.



Parameters

Feature lists

Select the feature lists in which the quantifier ion shall be re-set.

Quantifier ion list

Select a file that contains the necessary quantifier ion information in a tabular format (comma, tabulator, semicolon delimited). The file may contain additional columns that are not necessary here.

Tip

Since the [quantifier ion list](#) only selects the necessary information from the file, you can use the same file for the [Local compound database search](#) or in the MRM wizard.

Column headers

Set the column headers in the [quantifier ion list](#) file. mzmine will search for the respective headers in the file and read the data from it. The file may contain additional columns that are not necessary here. All columns (Q1 m/z , Quantifier m/z , RT) are necessary.

RT tolerance

Set an RT tolerance between the entry in the [quantifier ion list](#) and the detected RT in the feature list. This is the option to set different quantifiers to different features of the same precursor m/z .

m/z tolerance

Set an m/z tolerance between the entry in the [quantifier ion list](#) and the m/z in the feature list. Usually, the m/z is the set m/z in the acquisition software and does not fluctuate as it would in HRMS, where the actual m/z value is part of the detection.

Robin Schmid, Steffen Heuckeroth

 November 25, 2025 10:29:24

 Robin Schmid, Steffen Heuckeroth

11.9 Smoothing

11.9.1 Description

 This module should be used after feature detection step is performed.

Feature detection → Smoothing

Smoothing is an optional feature that is used depending on the noisiness of the data. Smoothing allows to approximate a peak shape to the "ideal" shape defined by the target function of the used algorithm.

In MZmine 3, two algorithms can be used for smoothing:

- Savitzky-Golay,
- and Loess smoothing.

Savitzky-Golay smoothing

This smoothing method is also implemented in **Raw data methods → Raw data filtering → Scan by scan filtering**. Its brief description is available in that [section](#). For more details see [1].

 The benefit of Savitzky-Golay is its efficiency. Due to a fixed user-defined window size and equal spacing, the weights are also fixed. As a result, the local fit need be solved only once.

Loess smoothing

LOESS (locally weighted smoothing) or **LOWESS (Locally Weighted Scatterplot Smoothing)** is a non-parametric method that relies only on a smoothing parameter value and the degree of polynomial without a predefined function. Each polynomial is fitted locally depending on the defined bandwidth.

More details on LOESS smoothing can be found in [2].

 Can be slightly slower than the Savitzky-Golay algorithm (due to the repeated local fitting) but also a bit more precise.

11.9.2 References

1. A. Savitzky and M. J. E. Golay, Anal. Chem., 36, 1627 (1964). DOI: 10.1021/ac60214a047
2. William S. Cleveland & Susan J. Devlin (1988) Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting, Journal of the American Statistical Association, 83:403, 596-610, DOI: 10.1080/01621459.1988.10478639

11.9.3 Parameters

FEATURE LISTS

Feature lists that the module will take as an input.

SMOOTHING ALGORITHM

Choose if Savitzky-Golay or LOESS smoothing will be used.

Savitzky-Golay additional parameters

Retention time smoothing

Number of data point to smooth in retention time dimension. Defines window size and coefficients used in smoothing.

Mobility smoothing

Number of data point to smooth in mobility dimension.

LOESS additional parameters

Retention time width (scans)

Number of scans to smooth in retention time dimension. Used to calculate a fraction of source points, which is subsequently used to calculate LOESS "bandwidth". "Bandwidth or "smoothing parameter" determines how much of the data is used to fit each local polynomial.

 Higher values of smoothing parameter lead to the smoother output but simultaneously to a larger loss of information.

Mobility width (scans)

Number of data point to smooth in mobility dimension. The points about previous parameter apply here as well.

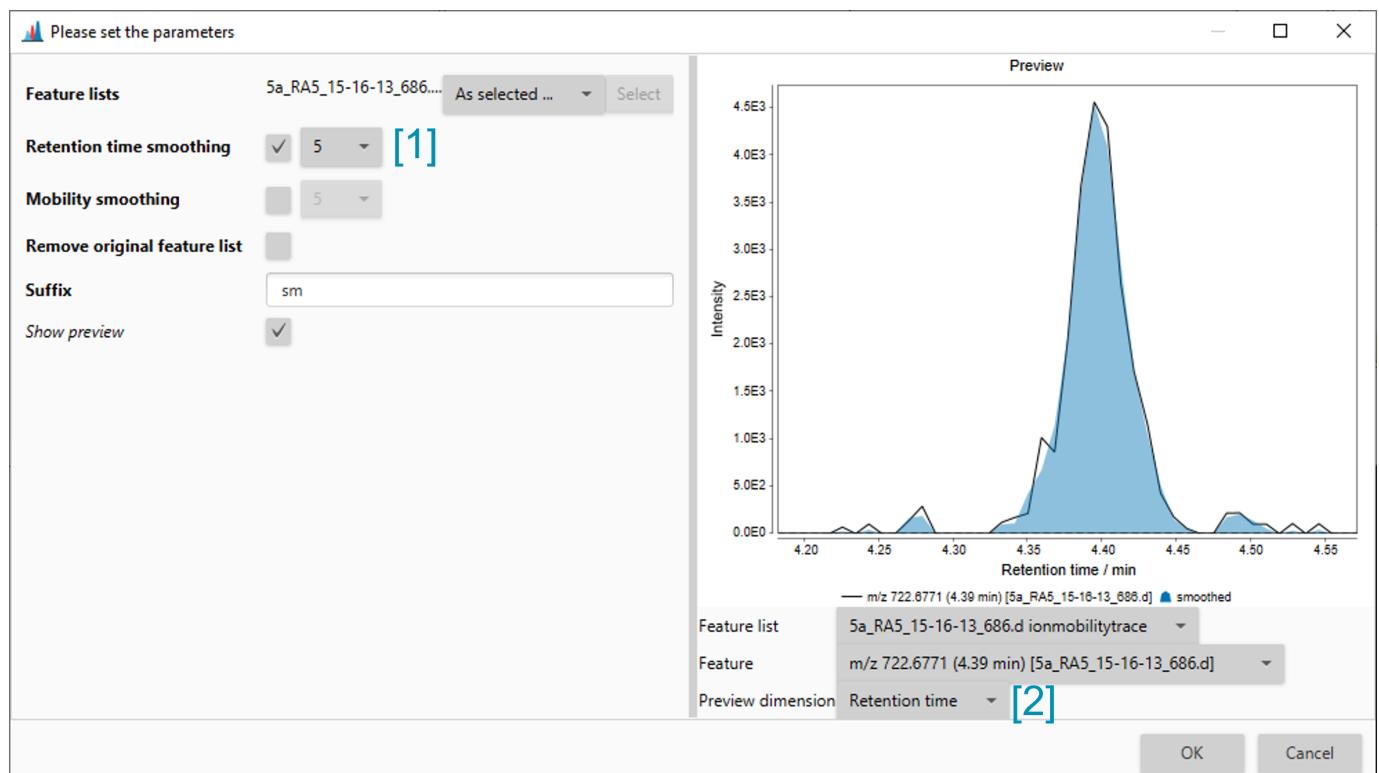
11.9.4 Optimizing smoothing parameters using the preview

Retention time dimension

Smoothing chromatograms is optional. The necessity of smoothing in RT dimension is determined by the noisiness of chromatographic peaks. These can be influenced by the overall spray stability, instrument accumulation times, transfer efficiency and many more.

The number of data points to be smoothed in rt dimension can be set at [1] (see image below). Note that the correct preview dimension is selected at [2] (see image below).

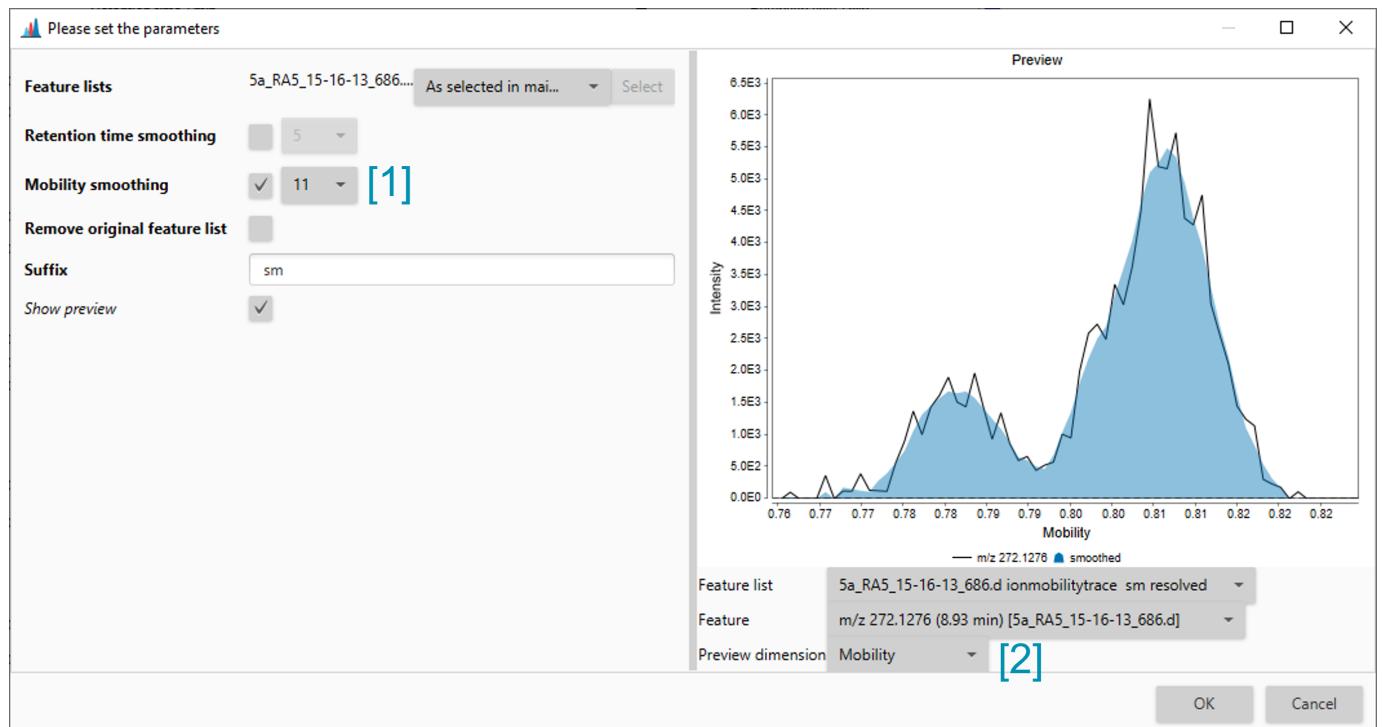
For large batch modes, the **Remove original feature list** parameter should be selected. While parameters are being optimised, this is not recommended, because removing a feature list cannot be undone.



Mobility dimension

After resolving a feature in RT dimension, the mobilograms will be recalculated from the raw data (the resolved ion mobility trace). Therefore, a smoothing step is necessary if the data requires it. The smoothing dialog is opened via **Feature detection** → **Smoothing**

Select to smooth the mobility dimension [1] and select it as preview dimension [2]. The filter with depends on the number of spectra acquired in the observed mobility range. Usually, a value between 5 and 15 should be appropriate.



Steffen Heuckeroth, omokshyna

⌚ November 24, 2022 21:13:07

👤 Steffen Heuckeroth, omokshyna

11.10 Resolving

11.10.1 Local Minimum Resolver

Description

≡ Feature detection → Chromatogram resolving → Local minimum resolver

During the EICs building, overlapping and partially co-eluting features are retained as single features in the feature list (see, for example, [Chromatogram builder](#)). As a local minimum in the EIC trace might correspond to the valley between two adjacent, partially-resolved peaks, the **Local minimum resolver (LMR)** utilizes such minima to split "shoulder" LC peaks into individual features (*i.e.* [chromatographic resolving](#)).

The algorithm examines all the data points in the EIC trace starting from the earliest RT. A scan window is set (see **Minimum search range RT/Mobility** parameter) and centered around the examined data point.

A data point is considered a **local minimum** if it is the lowest intense point within the scan window. When a local minumum is found, a set of user-defined intensity and feature duration requirements is checked. If they are fulfilled, the original overlapping peaks are split into new, distinct features.

 The **LMR** is particularly suitable for LC-MS data with little noise and nice peak shapes.

 With the implementation of ion mobility (IM) support in MZmine 3, this module was expanded and can now be applied over both the RT and IM dimensions (see [Resolving the ion mobility dimension](#)).

RESOLVING THE ION MOBILITY DIMENSION

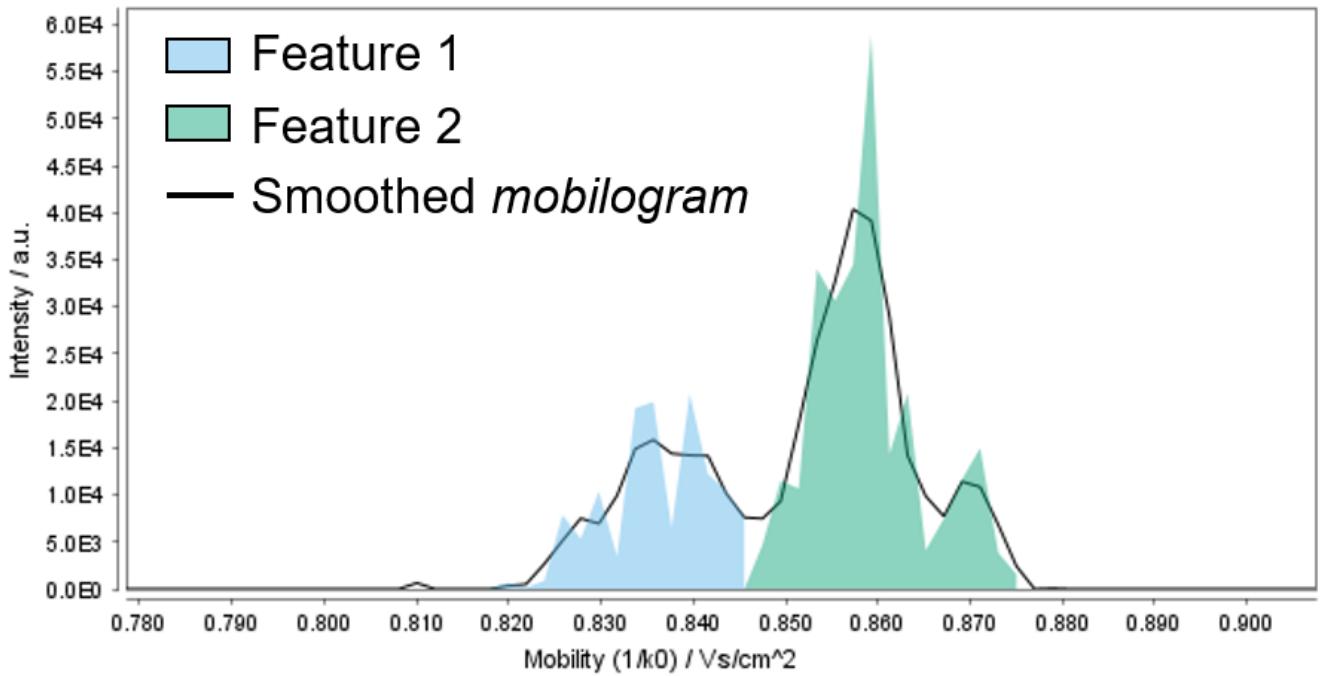
This same module can be used to resolve features co-eluting in the RT dimension, based on their ion mobility. The same concepts apply as in the resolution of the RT dimension.

However, [mobilograms](#) are examined instead of EIC traces and the same settings used for the RT dimension might not be optimal when resolving IM data. In particular, the following aspects should be born in mind:

1. While [frame scans](#) are examined over the RT dimension, [mobility scans](#) are considered over the IM dimension.

As explained [here](#), [frame scans](#) are essentially obtained by merging the [mobility scans](#) acquired over an IM accumulation. Therefore, it might be necessary to adjust parameters like **Minimum absolute height** or **Min ratio of peak top/edge** to account for the lower signal intensity of [mobility scans](#).

1. [Mobilograms](#) are recalculated from raw data, even though a [smoothing](#) step was previously applied. Non-smoothed [mobilograms](#) tend to be more jagged than regular EIC traces (see Figure). Therefore, some parameters (*e.g.* **Min search range** and **Min ratio of peak top/edge**) should be adjusted accordingly.



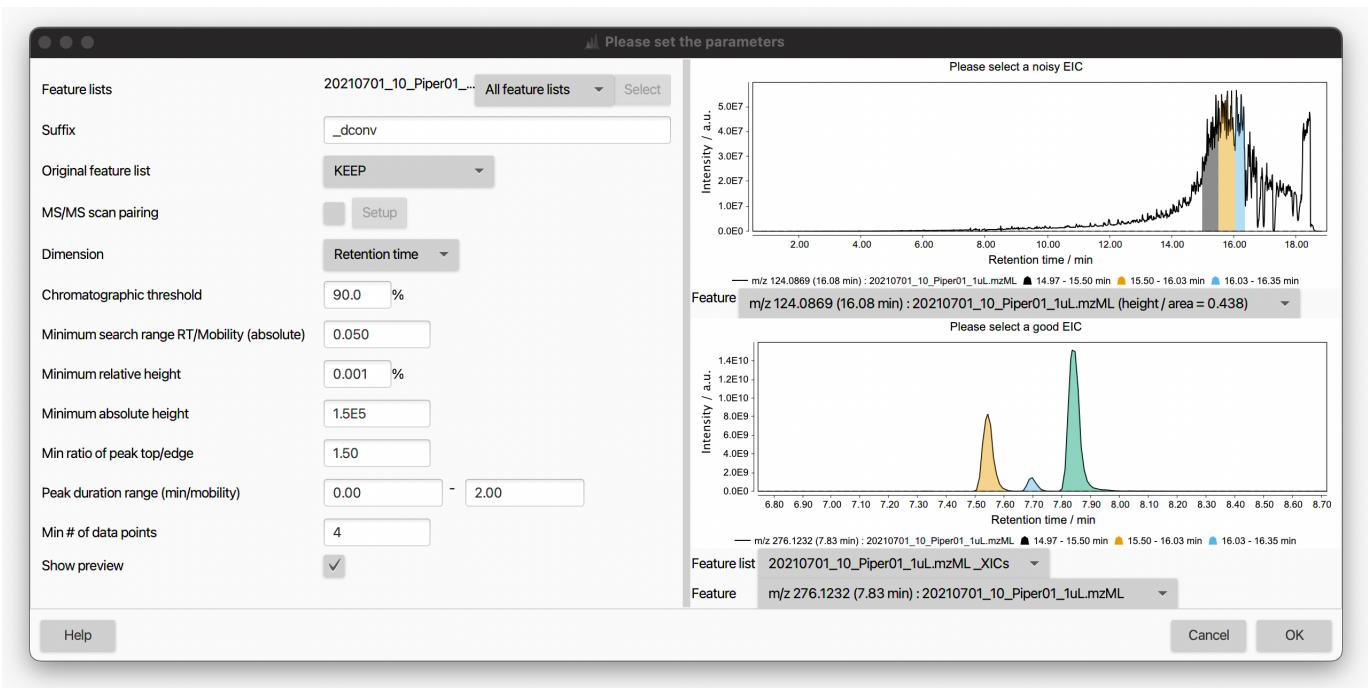
1. Mobilograms contain fewer scans (e.g. $\approx 400\text{-}1000$ per frame, depending on instrument type and acquisition settings), compared to regular EICs (e.g. ≈ 4500 scans for 15 minutes LC run and scan rate of 0.2 seconds). Therefore, a lower **chromatographic threshold** (e.g. 80%) is recommended to avoid relevant data points in the mobilogram being discarded.

On the other hand, a single feature in the IM dimension is normally made up of more data points than regular LC peaks, due to the different timescale the IM separation is performed on (see [here](#) for more details).

Therefore, a higher *Min # of data points* can be set when [resolving the ion mobility dimension](#) to filter out noisy features.

1. Different vendors use different units of mobility. For instance, TIMS express ions' mobility as Vs/cm^2 , whereas [time dispersive IM devices](#) (DTIMS and TWIMS) use the ions' drift time (expressed in milliseconds). TIMS values are numerically smaller than DTIMS or TWIMS; therefore, the **minimum search range** parameter should be adjusted accordingly.

Parameters



Suffix

String added as suffix to when creating the new feature list(s).

Original feature list

Defines the processing.

Standard is to KEEP the original feature list and create a new processed list.

REMOVE saves memory.

PROCESS IN PLACE is an advanced option to process directly in the feature list and reduce memory consumption more - this might come with side effects, apply with caution.

MS/MS scan pairing

Pair MS/MS fragmentation spectra collected in DDA mode to the resolved features. This is optional at this stage as the same can be done later in the pipeline using a separate module. See [MS2 scan pairing](#) documentation for more details.

Dimension

Dimension to be resolved. Select *Retention time* or *Mobility* to run the module over the RT or IM dimension, respectively.

Chromatographic threshold

Percentage of data points in the EIC removed before local minima search. This represents an important filter for noisy chromatogram and significantly reduces the processing time. The algorithm finds the intensity value (threshold) that leaves the specified percentage of data points in the EIC trace below the given value. All such data points are removed.

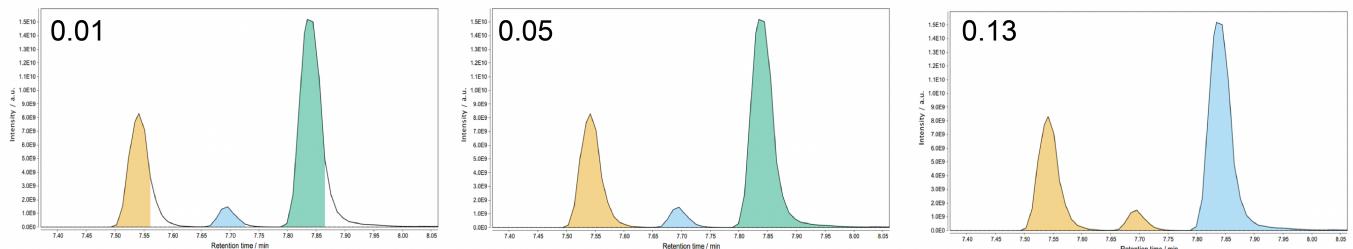
For example, a **Chromatographic threshold** = 50% will discard the lowest-intense 50% data points in the EIC trace.

💡 It must be noted that the algorithm examines the EICs throughout the entire RT range (*i.e.* also the zero data points are considered). Therefore, we recommend to set this value rather high (*e.g.* 90-95%) and lower it only if needed.

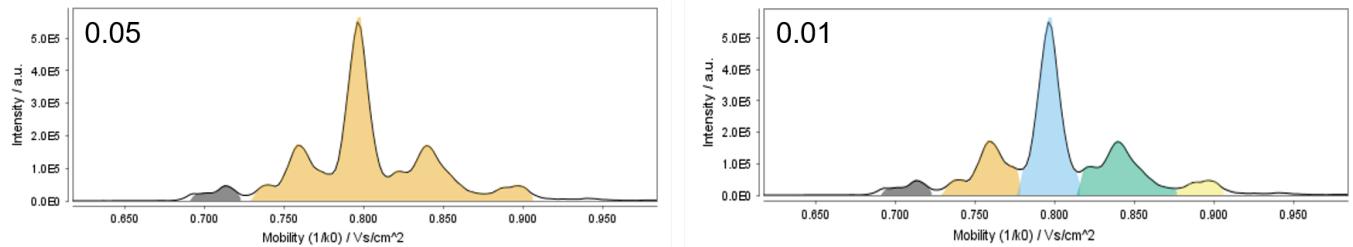
💡 When resolving the ion mobility dimension, we recommend to lower this settings to no more than 80% since [mobilograms](#) contains less data points than regular LC traces.

Minimum search range RT/Mobility (absolute)

Size of the RT, or mobility, window examined for local minimum search. An overly narrow search range can cause peak edges to be cut off, whereas a too wide search ranges might lead to an incomplete resolution of narrowly eluting peaks (see example below).



💡 A shorter *Minimum search range* is generally needed when [resolving the ion mobility dimension](#).



Minimum relative height

Minimum relative intensity (with respect to the highest data point in the EIC) a peak needs to reach to be retained as a feature. This parameter can be used in combination with the **Minimum absolute height** setting as to filter the resolved features to be retained.

Many users prefer to rely only on the *Minimum absolute height* as it is more straightforward to set. To do so, set the *Minimum relative height* = 0 and the parameter will be ignored.

💡 Modern mass spectrometers provide linear dynamic ranges up to 5 orders of magnitude. If we take an Orbitrap device with a detector saturation around 1.0E10 intensity, a *Minimum relative height* = 0.001 would correspond to a 1.0E5 minimum intensity.

Minimum absolute height

Minimum absolute intensity a peak needs to reach to be retained as a feature. This parameter is very similar to the [Min highest intensity](#) settings in the Chromatogram builder module and the same concepts apply.

💡 When resolving the RT dimension, the same value used as [Min highest intensity](#) in the EICs building can normally be used here.

💡 While [frame scans](#) are examined over the RT dimension, [mobility scans](#) are examined over the IM dimension. Therefore, this parameter might need to be adjusted accordingly when [resolving the ion mobility dimension](#).

Min ratio of peak top/edge

Minimum ratio between the intensity of the highest (apex) and side (left and right 'edges') points of a peak, to retain it as a feature. The peak edges have to be X times less intense than the peak apex for the feature to be retained.

The purpose of this parameter is to reduce the detection of false local minima when the examined trace (EIC or *mobilogram*) is not smooth. In general, this mainly affects low intensity and not-baseline-resolved signals

💡 This parameter can best be optimized using the *Show preview* option. We recommend values between 1.7 (not baseline separated) and 2 to start the optimisation.

Peak duration range (min/mobility)

Range of acceptable peak length expressed in minutes (RT dimension) or absolute units (mobility dimension). This parameter can be used to filter out noisy features based on their overly short (or long) duration.

Minimum scans (data points)

Minimum number of data points a resolved peak needs to have to be considered valid and retained as a feature. This parameter can be used along with the **Peak duration range** setting as peak duration constraint to filter out noisy features.

💡 This parameter is very similar to the [Min group size in # of scans](#) settings in the Chromatogram builder module and the same value can normally be used here (usually, no less than 4-5).

💡 A feature in the IM dimension is normally made up of more data points than regular LC peaks. Therefore, a higher *Min # of data points* can be set when [resolving the ion mobility dimension](#) to filter out noisy features.

Show preview

By checking this box, an interactive visualization panel will open to help the user to adjust the algorithm parameters. Two EIC traces can be displayed simultaneously in two sub-panels to assess the impact of chosen settings on both "good" and "noisy" EIC traces. The feature list and EIC traces to display can be selected from the corresponding drop-down menus. A noisy EIC can generally be found by sorting the feature table by decreasing area, or by looking at the height/area ratio provided for each feature in the top sub-panel (noisy EIC tend to have low height/area ratios). We recommend optimising the parameters on good EICs and checking the results of these parameters with a noisy EIC.

Adriano Rutz, Robin Schmid, Steffen Heuckeroth, Steffen Heuckeroth, corinnabrungs, omokshyna, tdamiani

⌚ October 16, 2025 12:14:55

👤 Adriano Rutz, Robin Schmid, Steffen Heuckeroth, Steffen Heuckeroth, corinnabrungs, omokshyna, tdamiani

11.10.2 Wavelet resolver



This feature resolver is currently in beta phase (11.2025). (Added after mzmine 4.8.5)



Wavelet resolver vs. [Local minimum resolver](#):

While the local minimum resolver is the quicker resolver, it may struggle with chromatograms with high baseline and peaks on top said baseline. The wavelet resolver takes longer to compute, but may outperform the local minimum.

Parameters

[Signal-to-noise threshold](#)

Minimum required signal-to-noise ratio of a peak. The noise is calculated by the standard deviation (STD) or median absolute deviation (MAD) (see [noise calculation](#) and [noise window factor](#)). The signal height (S) is determined by the peak height minus the average of the two edges.

$$\begin{aligned} \text{Signal-to-noise} &= S/N \\ S &= I_{\text{peakTop}} - \frac{I_{\text{leftEdge}} + I_{\text{rightEdge}}}{2} \\ N &= \text{STD}(\text{Background}) \text{ or } \text{MAD}(\text{Background}) \end{aligned}$$

[Top to edge \(SNR override\)](#)

If a potential feature does not match the SNR threshold, check if the top/edge ratio is above this threshold.

[Minimum height](#)

Minimum absolute required of a signal to be retained as a feature.

[Noise calculation](#)

Choose a method to calculate the noise around a potential signal. Other closely eluting signals are not used for noise/background calculation.

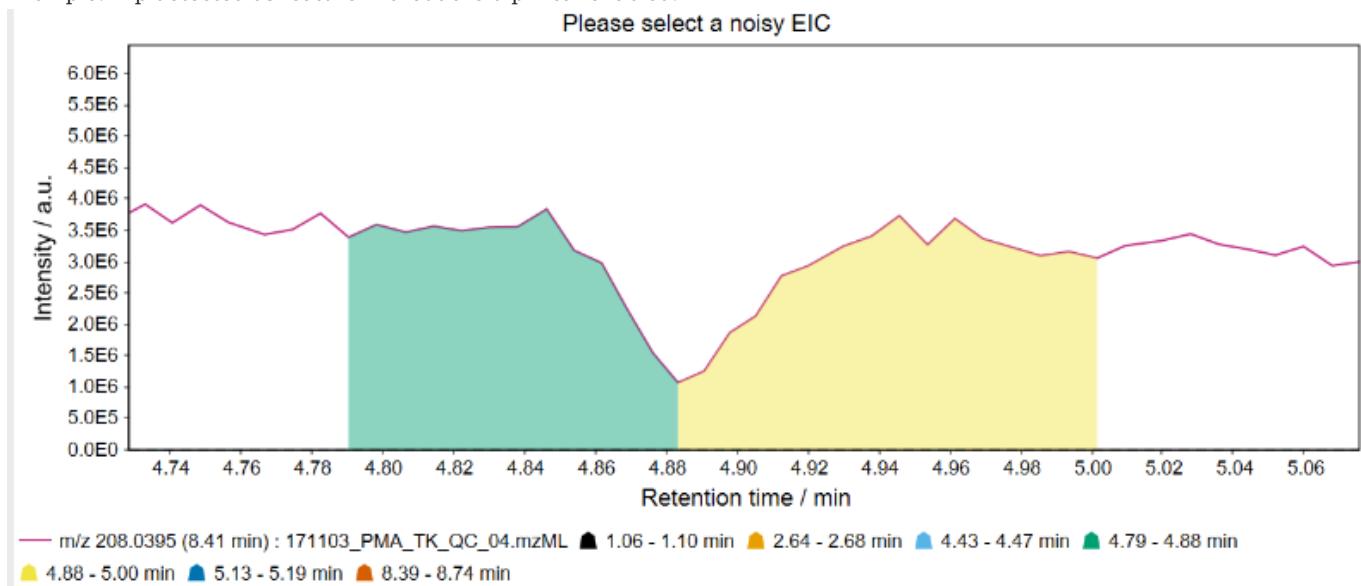
Default: Standard Deviation

[Dip filter](#)

Filters V-shaped dips in the baseline that are caused by instable sprays and may behave like an unresolved double peak.

Default: enable for LC-MS, disable for GC-EI-MS

Example: Dip detected as feature without the dip filter enabled.



Advanced parameters

Scales

Scales (widths) of the wavelets. Multiple wavelets allow detection of signals of varying widths. More scales increase the processing time. Larger scales allow recognition of wider peaks, small scales allow recognition of narrower peaks. This scales with the number of points per peak, but should not be seen as a 1:1 factor. (A scale of 1 will not cause recognition of peaks with only 1 signal).

If disabled/default: 0.5, 1.0, 1.5, 2.0, 3.0, 5.0, 8.0, 10.0

Noise window factor

The window around a potential signal to calculate the signal-to-noise ratio in. The window is defined by this factor multiplied by the peak width. Only signals that do not belong to a potential other peak are used for noise calculation. At least 6 signals must be found to calculate the noise by the selected [noise calculation](#) method. Otherwise the top-to-edge ratio of the signal must match the selected SNR.

If disabled/default: 3 is used

Required fitting scales

Minimum number of fitting wavelet scales for a peak to be retained.

If disabled/default: 2 Note: Enable and set to 1 to disable this filter

Apply robustness iteration

Regions of potential peaks are not used for noise calculation. This may cause too many regions being excluded. Applying a second iteration allows more robust noise calculation for signals in crowded regions of the chromatogram.

Default: true

Edge detection

Define an algorithm to detect the edges of a peak. Local minimum: Continues the search for a minimum as long as the next point (within a tolerance depending on peak width) is lower than the previous. Absolute minimum: Continues the search as long as the next point (within a tolerance depending on peak width) is lower than the previous absolute minimum. Slope: Continues the search until an increasing slope is reached. Uses varying window width depending on peak width.

If disabled/default: Absolute minimum is used

Allow sign changes every N points

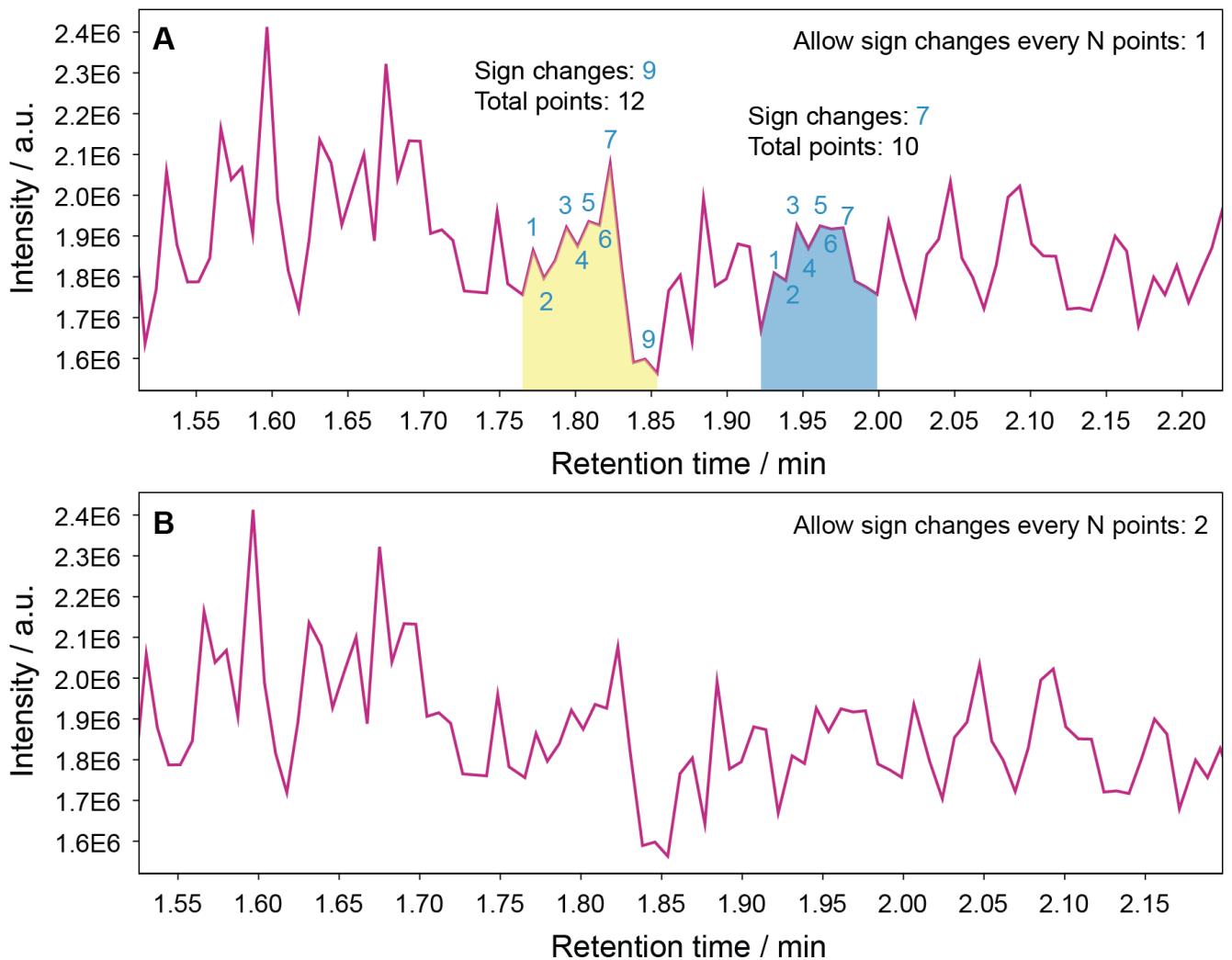
Removes noisy signals with lots of up/down movement.

Filters by the number of sign changes ($|S|$) in the slope between two points in a potential feature. If the slope changes from positive to negative or vice versa, it is counted as a sign change. Since a peak is expected to have a high point, one sign change is expected per potential peak and subtracted from the number of counted sign changes. The number of additional sign changes ($|S-1|$) is then divided by the total number of data points in a peak ($\frac{|S-1|}{N_{\text{points}}}$). This number is then multiplied by the value of this parameter (p). We call the result of this formula the relative jaggedness (J_R). If the relative jaggedness is ≤ 1 , the potential peak is discarded.

$$(J_R = \frac{|S-1|}{N_{\text{points}}} \cdot p)$$

If disabled: 2 is used. To not use this filter enable and set to 1.

The effect of the filter is visualized here:



The yellow signal in panel **A** shows 9 sign changes for 12 data points, the blue signal shows 7 sign changes for 10 data points. The jaggedness is thereby calculated as:

$$(J_{R,yellow} = \frac{|(9-1)|}{12} \cdot p = \frac{8}{12} \cdot p = \frac{2}{3}p) \quad (J_{R,blue} = \frac{|(7-1)|}{10} \cdot p = \frac{6}{10} \cdot p = 0.7p)$$

For a parameter value (p) of 1, both signals are retained, as ($J_R \leq 1$). If a sign changes is only allowed every ($p = 2$) points, it results in ($J_R > 1$) in both cases.

(Note: S/N had to be set to 2 to allow these signals to be picked up at all.)

Maximum ratio of similar height signals in background

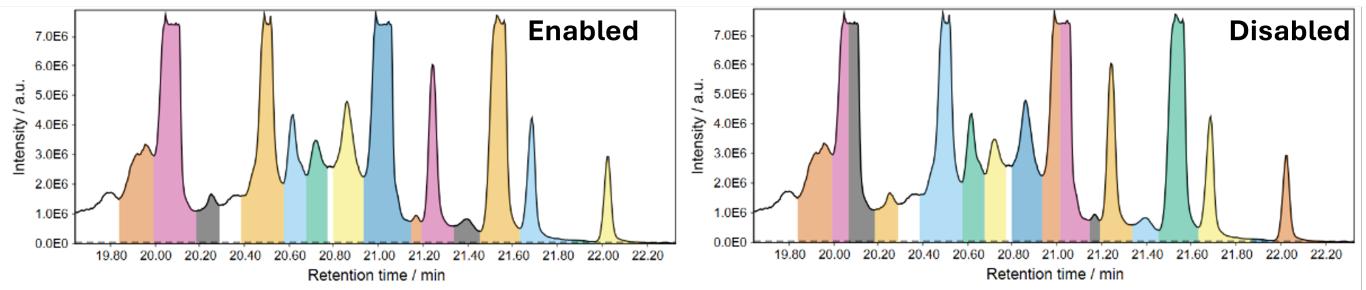
Noisy baselines may pass the SNR filter due to low standard deviation if the baseline is relatively stable. This filter checks if there are a lot of signals with a similar absolute height (80%) as a potential peak. If that is the case, the peak is removed.

If disabled: (0.10) will be used as default. Note: A value of 1 disables this filter. Note: It may make sense to disable this filter (set to 1) to detect signals that are placed on an increasing baseline.

Saturation filter

Detects if a peak has a flat top signal due to saturation of the MS detector.

Default: true



⌚ December 11, 2025 11:45:16

👤 Steffen Heuckeroth

11.10.3 Baseline resolver

Description

≡ Feature detection → Chromatogram resolving → Baseline resolver

A very simple method that can be used to demonstrate the functionality of chromatogram resolving.

- First, the algorithm removes the lowest part of the chromatogram below a **Baseline level** specified by the user.
- Remaining peaks above the baseline level are recognized if they fulfill the height and duration requirements.

Parameters

Suffix

This string is added to feature list name as suffix

Original feature list

Defines the processing.

Standard is to KEEP the original feature list and create a new processed list.

REMOVE saves memory.

PROCESS IN PLACE is an advanced option to process directly in the feature list and reduce memory consumption more - this might come with side effects, apply with caution.

MS/MS scan pairing

Set MS/MS scan pairing parameters. For more details see [MS2 scan pairing](#)

Min peak height

Minimum acceptable feature height (absolute intensity)

Peak duration range

Range of acceptable feature durations

Baseline level

Level below which all data points of the chromatogram are removed (absolute intensity)

Min # of data points

Minimum number of data points on a feature

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [omokshyna](#)

11.10.4 Noise amplitude resolver

Description

≡ Feature detection → Chromatogram resolving → Noise amplitude resolver

This method is suitable for chromatograms with significant background noise of varying intensities. It works in a similar way to the Baseline cut-off method, but sets the baseline level individually for each chromatogram, depending on the amplitude of signal noise.

The baseline level is calculated as follows:

- The intensity range of the chromatogram is divided into bins of the user-specified size (the Noise amplitude parameter)
- The bin with the highest number of data points is found. This bin represents the intensity level of the noise signal.
- The baseline level is set to the intensity of the bin with the most data points

Parameters

Suffix

This string is added to feature list name as suffix

Original feature list

Defines the processing.

Standard is to KEEP the original feature list and create a new processed list.

REMOVE saves memory.

PROCESS IN PLACE is an advanced option to process directly in the feature list and reduce memory consumption more - this might come with side effects, apply with caution.

MS/MS scan pairing

Set MS/MS scan pairing parameters. For more details see [MS2 scan pairing](#)

Min peak height

Minimum acceptable height (intensity) for a feature

Peak duration range

Range of acceptable feature durations

Amplitude of noise

This value is the intensity amplitude of the signal in the noise region

Min # of data points

Minimum number of data points on a feature.

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [omokshyna](#)

11.10.5 Savitzky-Golay resolver

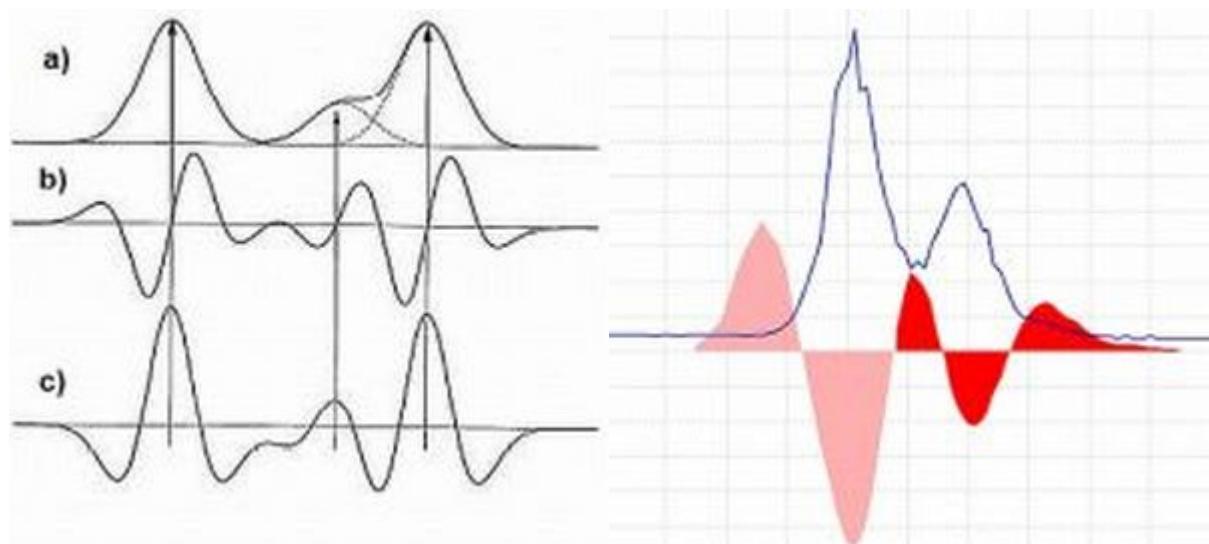
Description

≡ Feature detection → Chromatogram resolving → Savitzky Golay resolver

This method uses the **Savitzky-Golay polynomial** [1] to calculate the smoothed second-derivative of the chromatogram's intensities.

The following figure (left) presents the shape of a) a Gaussian peak, b) the first derivative, and c) the second derivative.

The figure on the right side shows how the signal (blue line) may be divided into individual chromatographic peaks by observing the second derivative.



References

1. A. Savitzky and M. J. E. Golay, Anal. Chem., 36, 1627 (1964). DOI: [10.1021/ac60214a047](https://doi.org/10.1021/ac60214a047)

Parameters

Suffix

This string is added to feature list name as suffix

Original feature list

Defines the processing. Standard is to KEEP the original feature list and create a new processed list.

REMOVE saves memory.

PROCESS IN PLACE is an advanced option to process directly in the feature list and reduce memory consumption more - this might come with side effects, apply with caution.

MS/MS scan pairing

Set MS/MS scan pairing parameters. For more details see [MS2 scan pairing](#)

Min peak height

Minimum acceptable feature height (absolute intensity)

Peak duration range

Range of acceptable feature durations

Derivative threshold level

Minimum acceptable intensity in the second derivative for feature recognition

Min # of data points

Minimum number of data points on a feature.

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

11.11 Spectral deconvolution (GC)

11.11.1 GC-EI spectral deconvolution

Description

≡ Feature list methods → Spectral deconvolution (GC) → GC-EI spectral deconvolution

This ADAP inspired method groups resolved features into pseudo spectra, representing deconvoluted GC-EI spectra. It clusters features based on retention time and other parameters to construct fragmentation spectra for analytes.

info !!!

This module requires resolved features. Run this module after feature resolving, e.g. Local Minimum Resolver.

The following steps are performed depending on the selected Deconvolution algorithm:

DECONVOLUTION ALGORITHM: RT GROUPING AND SHAPE CORRELATION

All features are clustered based on their retention times. The user can specify the maximum retention time tolerance and the minimum number of signals in a pseudo spectrum. Features are initially sorted by intensity. Each feature is then assigned to potential clusters within the specified retention time tolerance. If a feature falls within the RT range of multiple clusters, shape correlation is performed to determine the most suitable cluster. Clusters that do not meet the minimum number of signals are removed.

DECONVOLUTION ALGORITHM: HIERARCHICAL CLUSTERING

All features are clustered based on their retention times. The user can specify the maximum retention time tolerance and the minimum number of signals in a pseudo spectrum. Features are initially sorted by intensity. Each feature is placed in an initial cluster. Clusters are then paired based on their RT distance, and a priority queue is used to manage these pairs. If all features in a cluster pair fall within the specified RT tolerance, the clusters are merged. This process continues until no more clusters can be merged. Clusters that do not meet the minimum number of signals are removed.

Also read: References

Smirnov A, Jia W, Walker D, Jones D, Du X: ADAP-GC 3.2: Graphical Software Tool for Efficient Spectral Deconvolution of Gas Chromatography—High-Resolution Mass Spectrometry Metabolomics Data. *J. Proteome Res* 2017, DOI: [10.1021/acs.jproteome.7b00633](https://doi.org/10.1021/acs.jproteome.7b00633)

Parameters

Deconvolution algorithm

RT GROUPING AND SHAPE CORRELATION

Retention Time Tolerance (min)

Maximum allowed difference between two retention time values for them to be clustered together.

Minimum Number of Signals

Minimum number of deconvoluted signals in a pseudo spectrum.

Minimum shape similarity

Minimum shape similarity of a feature to be added to a pseudo spectrum. Features not meeting the shape similarity (Pearson correlation) are not considered in the pseudo spectrum.

HIERARCHICAL CLUSTERING

Retention Time Tolerance (min)

Maximum allowed difference between two retention time values for them to be clustered together.

Minimum Number of Signals

Minimum number of deconvoluted signals in a pseudo spectrum.

Exclude m/z-values

Optional parameter Optionally, the user can specify a list of deprecated m/z values such that EIC peaks with those m/z values could not be chosen as model peaks. It is possible to specify single m/z values as well as ranges of m/z values. For example: 1-50, 73, 100.

Suffix

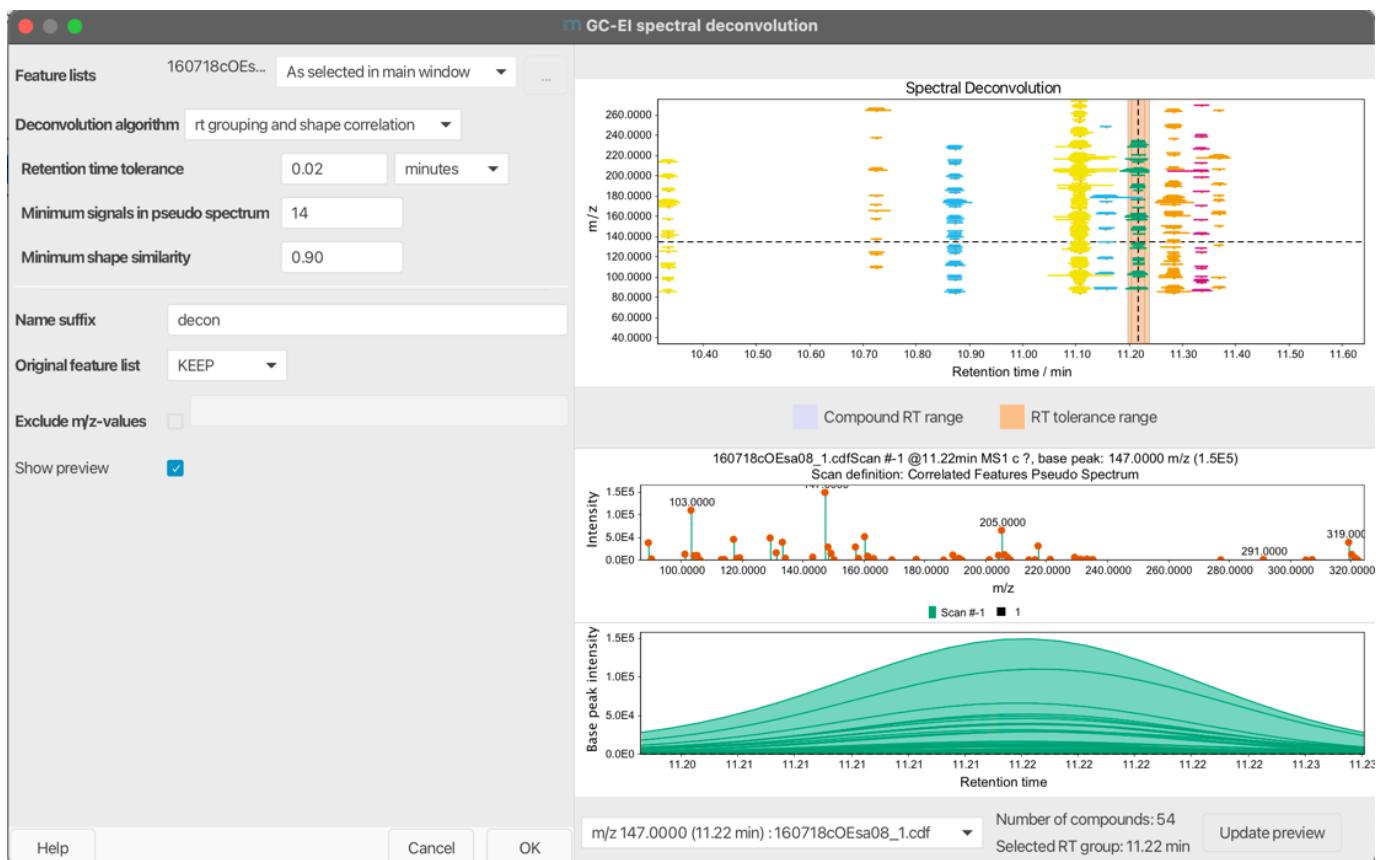
String to add to feature list name as a suffix.

Remove original feature list

If checked, the original feature list will be removed.

Preview

The preview dialog displays the deconvolution results for the selected feature list. Click on "Update preview" to create the plots. The upper plot displays grouped features by color as compounds. The grey area around a selected group represents the RT range of the most abundant feature. The orange area indicates the RT tolerance set in the parameters. The plots at the bottom show pseudo spectra based on the grouped features and an overlay of the grouped features for quality control. At the bottom, the user can select a grouped compound based on the representative feature. Next to the combo box is a short summary of the selected feature and the total number of groups.



Ansgar Korf

⌚ November 23, 2024 08:29:26

👤 Ansgar Korf

11.12 CCS Calibration and calculation

11.12.1 Description

The modules described in this section can be found in **Feature list methods → Processing → Internal reference CCS calibration / External CCS calibration / Calculate CCS values.**

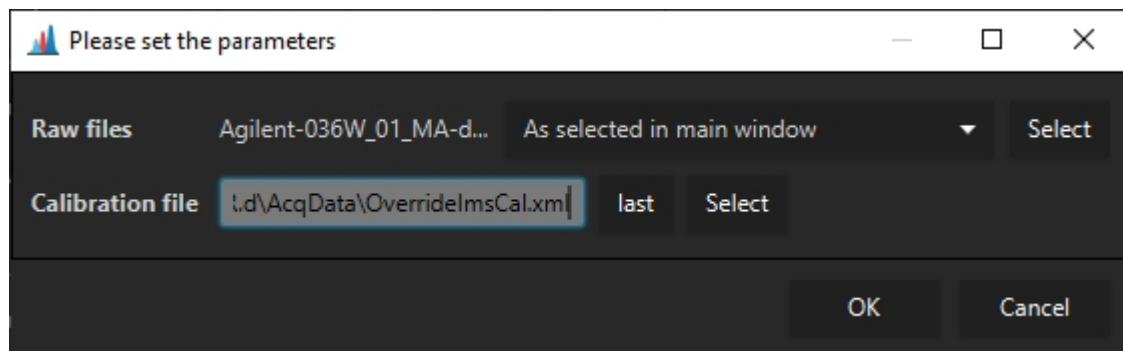
Accurate determination of CCS values requires a valid CCS calibration and molecule charge states to be detected. A CCS calibration can be either imported or created from internal reference.

- **timsTOF** raw data can be recalibrated using data analysis and imported in MZmine. The recalibrated data will be used by default. (see [Calculating CCS values](#))
- **mzML** raw data requires the determination of a calibration function from the raw data (e.g. as detected features) or as import from an external file. (see [importing an external CCS calibration](#) and [reference CCS calibration](#))

11.12.2 Importing an external CCS calibration

Agilent calibration data can be imported from the "OverrideImsCal.xml" file in the Agilent raw data folder. Waters calibration data can be imported from the "mob_cal.csv" file in the Waters raw data folder. The "_extern.inf" file is also required, but will be read automatically when the "mob_cal.csv" is selected.

The calibration import is accessed via **Feature list methods -> Processing -> External CCS Calibration**. Then select the calibration "OverrideImsCal.xml"/"mob_cal.csv" from the raw data folder, and select the raw data files the calibration should be applied to.

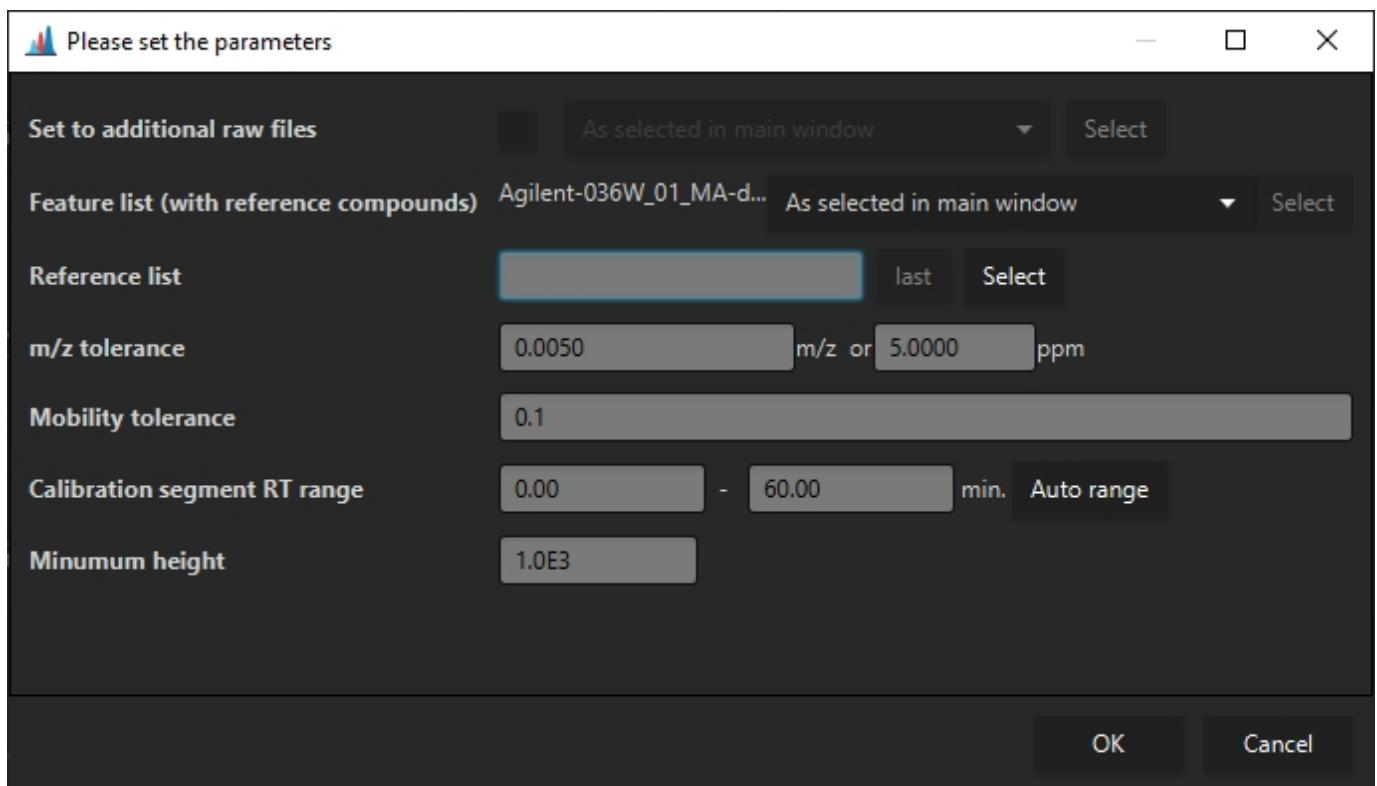


11.12.3 Reference CSS calibration

If a mobility calibrant is infused during an HPLC run of every sample, a CCS calibration can be calculated on a per-raw file basis. (Common procedure on Bruker devices) Otherwise, a single run can be used to calibrate multiple files.

Please note that this is currently only supported for TIMS and DTIMS data.

The calibration module can be accessed via **Feature list methods -> Processing -> Internal reference calibration**.



Parameters

SET TO ADDITIONAL RAW FILES

If a calibration calculated from a single feature list shall be applied to multiple other raw files, the raw files can be selected here. This requires only a single raw file to be selected.

FEATURE LIST (WITH REFERENCE COMPOUNDS)

Specifies (a) feature list(s) that contains the reference compounds. If multiple feature lists are selected, every feature list will be searched for reference compounds, and the calibration will be used for the raw data files in the particular feature list. This means that no raw data file may be selected. (Cannot set multiple calibrations to a single raw file)

If a single feature list is selected, the calibration may be applied to additional raw data files via the **Set to additional raw files** parameter.

REFERENCE LIST

Specifies a ".csv" reference list of for CCS calibrant ions. Must contain the columns "mz", "mobility", "ccs", "charge". Columns must be separated by ";". The ion mode may be specified via the charge of the ion, e.g., as 1 or -1. Only the correct polarity will be used to calculate the calibration.

M/Z TOLERANCE

The m/z tolerance for the reference compounds.

MOBILITY TOLERANCE

The mobility tolerance to detect the reference compounds.

CALIBRATION SEGMENT RT RANGE

Specifies the rt range that shall be searched for calibrant ions. Usually either the beginning or end of a HPLC run.

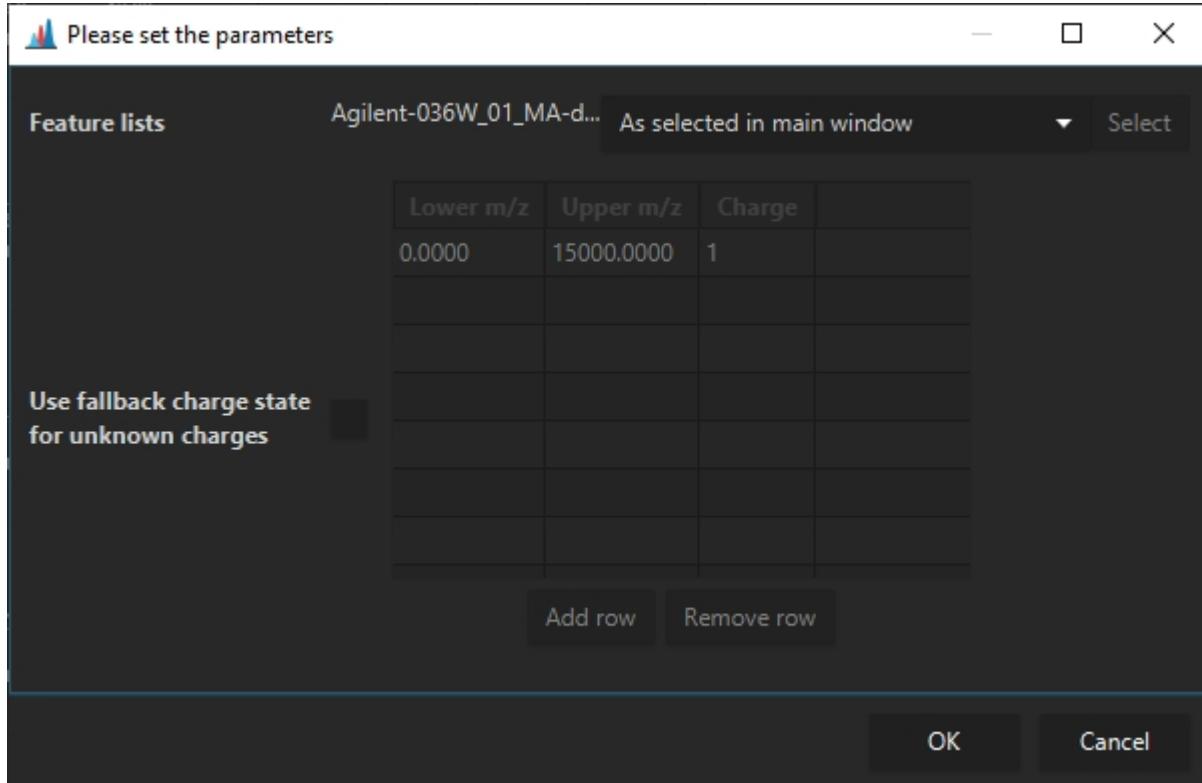
MINIMUM HEIGHT

A minimum intensity for reference compounds to be used as calibrant signals for determination of the calibration.

11.12.4 Calculating CCS values

After a calibration as been set (Agilent/Waters/Bruker mzML) (Bruker tdf works out-of-the-box) CCS values can be calculated via **Feature list methods -> Processing -> Calculate CCS values**.

Here, a default charge state may be set, in case it could not be determined. Otherwise, the charge state determined via the isotope pattern will be used.



Steffen Heuckeroth, omokshyna

⌚ November 24, 2022 21:13:07

👤 Steffen Heuckeroth, omokshyna

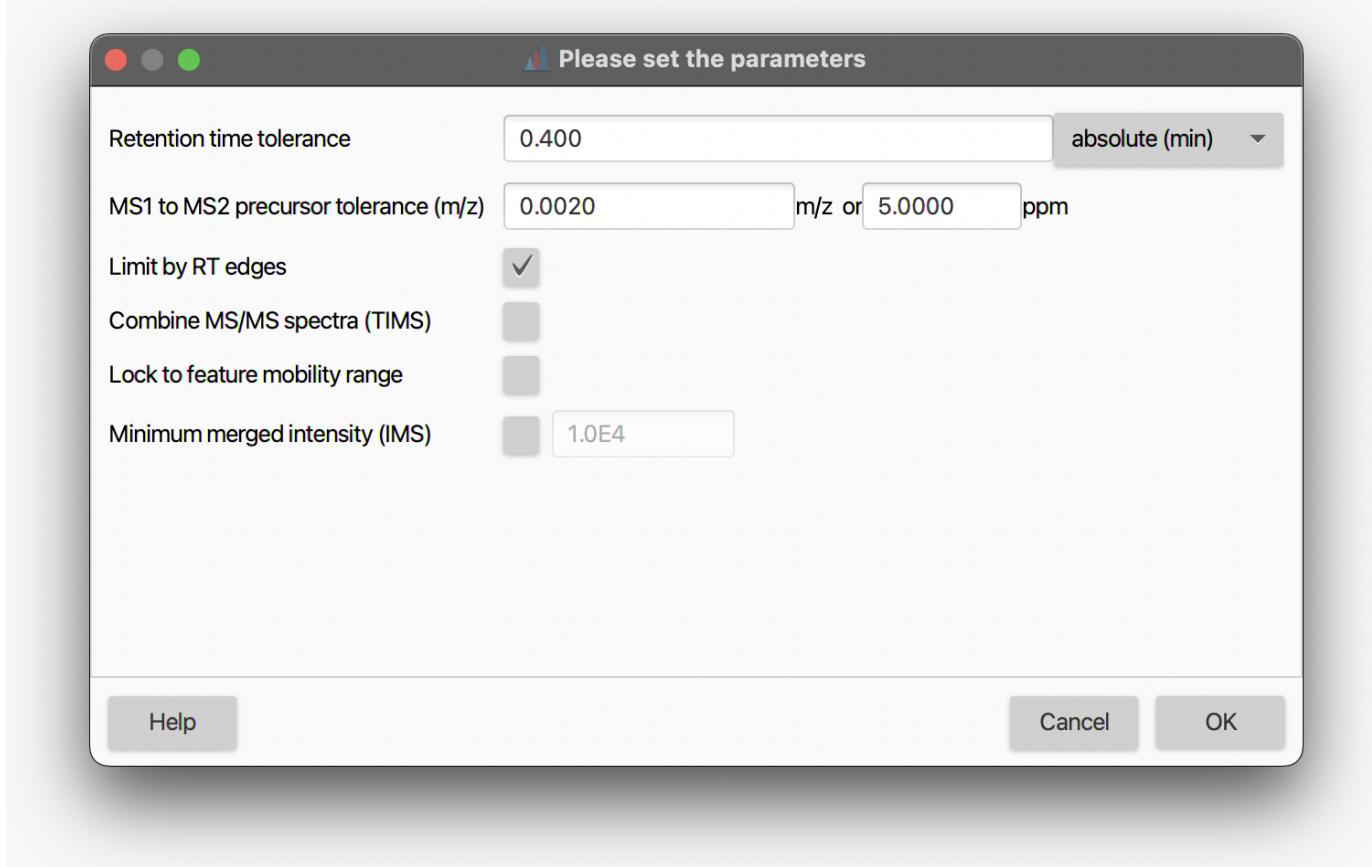
11.13 MS2 Scan Pairing

11.13.1 Description

 Feature list methods → Processing → Assign MS2 to feature

This module allows to pair MS2 scans with features. It assigns all MS2 scans within range to all features in chosen feature list.

11.13.2 Parameters



RETENTION TIME TOLERANCE

The maximum offset between the highest point of the chromatographic peak and the time the MS2 was acquired.

MS1 TO MS2 PRECURSOR TOLERANCE (M/Z)

Describes the tolerance between the precursor ion in a MS1 scan and the precursor m/z assigned to the MS2 scan.

LIMIT BY RT EDGES

Use the feature's edges (retention time) as a filter.

COMBINE MS/MS SPECTRA (TIMS)

If checked, all MS/MS spectra assigned to a feature will be merged into a single spectrum.

LOCK TO FEATURE MOBILITY RANGE

If checked, only mobility scans from the mobility range of the feature will be merged.

 This is usually not needed. However, if isomers/isobars elute at the same retention time and are close in mobility, the MS/MS window might be larger than the peak in mobility dimension and thus cause chimeric MS/MS spectra.

MINIMUM MERGED INTENSITY

If an ion mobility spectrometry (IMS) feature is processed, this parameter can be used to filter low abundant peaks in the MS/MS spectrum, since multiple MS/MS mobility scans need to be merged together.

[omokshyna, tdamiani](#)

 November 24, 2022 21:13:07

 [omokshyna, tdamiani](#)

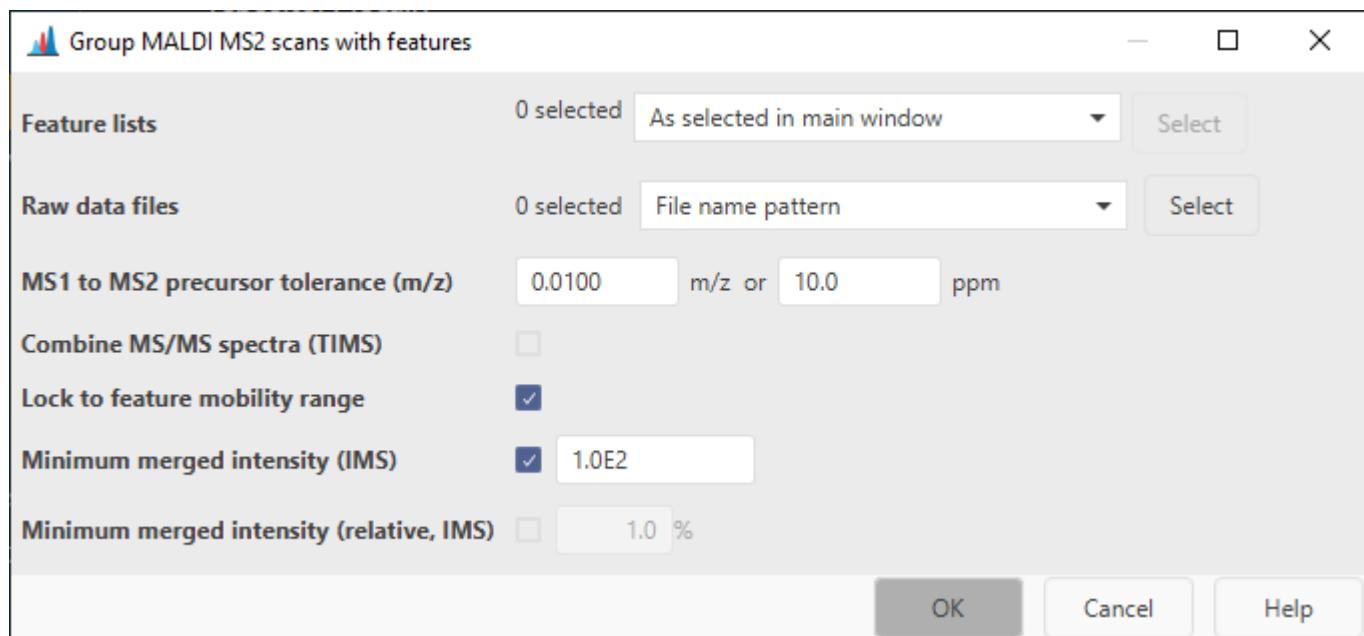
11.14 Assign MALDI MS2s to features

11.14.1 Description

≡ Feature list methods → Processing → Assign MALDI MS2s to features

This module allows to pair MS2 scans with features. It uses the spot information stored in the MS1 feature list and the MS2 spectra, the precursor m/z, and the ion mobility windows to pair the MS2s. Be careful when processing multiple imaging datasets with different geometries (multiple flexImaging run files). Among different geometries, multiple spots can have the same coordinate, as they are just stored as a location. This can lead to wrong assignments for abundant species. Make sure to only select the correct MS2 files, e.g., with a more specific name pattern.

11.14.2 Parameters



FEATURE LISTS

Select the feature lists to pair the MS2s to

RAW DATA FILES

The raw data files containing the MS2 spectra. By default, "File name pattern" with the pattern "*_MSMS*" is selected, due to the naming of the MS2 files. More specific patterns, such as " *RUN1*_MSMS*" can be used.

MS1 TO MS2 PRECURSOR TOLERANCE (M/Z)

Describes the tolerance between the precursor ion in a MS1 scan and the precursor m/z assigned to the MS2 scan.

COMBINE MS/MS SPECTRA (TIMS)

If checked, all MS/MS spectra assigned to a feature will be merged into a single spectrum. Individual spectra are retained. Usually this is not required, because MS2s can be merged during library matching.

LOCK TO FEATURE MOBILITY RANGE

If checked, only mobility scans from the mobility range of the feature will be merged.

⚠ This is usually not needed. However, if isomers/isobars elute at the same retention time and are close in mobility, the MS/MS window might be larger than the peak in mobility dimension and thus cause chimeric MS/MS spectra.

MINIMUM MERGED INTENSITY

If an ion mobility spectrometry (IMS) feature is processed, this parameter can be used to filter low abundant peaks in the MS2 spectrum, since multiple MS/MS mobility scans need to be merged together.

[Steffen Heuckeroth](#)

 November 13, 2023 11:26:25

 [Steffen Heuckeroth](#)

11.15 Isotope filtering

11.15.1 ^{13}C isotope filter

Description

≡ Feature list methods → Isotopes → ^{13}C isotope filter (formerly: isotope grouper)

MZmine carries out the feature detection steps of *chromatogram building* and subsequent *resolving* considering **all** the signals stored in the *mass lists*. As a consequence, signals generated by isotopologues of the same chemical entity (see *isotopic pattern*) are detected as distinct features and included in the *feature lists*, representing redundant information for the downstream data analysis. This issue ordinarily occurs for C-containing molecules, where the ^{13}C isotope signals can be easily detected ($^{13}\text{C}/^{12}\text{C} \approx 1.1\%$).

The *^{13}C isotope filter* module (formerly called *Isotope grouper*) aims at filtering out the features corresponding to the ^{13}C isotopes of the same analyte. The algorithm consider each feature individually and checks for the presence of potential ^{13}C -related peak(s) in the *feature lists*. When an isotope pattern meeting the user-defined tolerances (e.g. *m/z*, RT) and requirements (e.g. *monotonic shape*) is found, the information is saved, and only the feature corresponding to the e.g. most intense isotope is retained in the *feature list*. It must be noted that ^{13}C peaks are searched within the *feature list*, and not in the raw data.

Parameters

The dialog box has a title bar "Please set the parameters". It contains the following fields:

- Feature lists:** A dropdown menu set to "Those created by previous batch step" with a "Select" button.
- Name suffix:** An input field containing "_diso".
- m/z tolerance:** A dropdown menu showing "0.001" followed by "m/z or 3.5000 ppm".
- Retention time tolerance:** An input field showing "0.010" with a dropdown menu set to "absolute (min)".
- Mobility tolerance:** An input field showing "0.008".
- Monotonic shape:** A checked checkbox.
- Maximum charge:** An input field showing "1".
- Representative isotope:** A dropdown menu set to "Most intense".
- Never remove feature with MS2:** A checked checkbox.
- Original feature list:** A dropdown menu set to "KEEP".

At the bottom are "Help", "Cancel", and "OK" buttons.

Name suffix

String added as suffix when creating the new feature list(s).

m/z tolerance

Maximum allowed difference between the measured and the predicted m/z of the (potential) ^{13}C isotope to be grouped as isotopologues. The tolerance can be specified as absolute tolerance (in m/z), relative tolerance (in ppm), or both. When both are specified, the tolerance range is calculated using the maximum between the absolute and relative tolerances.

 We recommend to set a fairly strict m/z tolerance to reduce the risk of discarding false ^{13}C isotopes.

Retention time tolerance

Maximum allowed RT difference between the feature and its (potential) ^{13}C isotope to be grouped as isotopologues.

 Isotopologues should exhibit identical chromatographic behaviour and thus produce overlapping LC peak shapes. Therefore, a strict RT tolerance can be used to reduce the risk of discarding false ^{13}C isotopes.

Mobility tolerance

If enabled and the mobility dimension was recorded, potential ^{13}C isotopes will be grouped as isotopologues only if their mobility difference is within the defined tolerance.

 The same principle seen for the RT tolerance apply to the IM dimension. Isotopologues should exhibit identical IM separation; therefore, a strict mobility tolerance can be used to reduce the risk of discarding false ^{13}C isotopes.

Monotonic shape

If true, a monotonically decreasing trend of the isotope pattern (typical of the ^{13}C isotope pattern of small molecules) is required for the filtering.

Maximum charge

Maximum charge state considered to predict the ^{13}C isotopes' m/z . If a value > 1 is set, all the cahrge states are evaluated and the one providing the most isotope features detected is used for the filtering.

Never remove feature with MS2

If checked, potential ^{13}C -related features will not be discarded if [associated to a MS2 spectrum](#).

Original feature lists

Keep or remove the input feature list(s). The *PROCESS IN PLACE* option directly filter the input feature list and performs better in terms of memory usage; therefore, it is recommended over *REMOVE*, when available.

[lalalana5, omokshyna, tdamiani](#)

 November 24, 2022 21:13:07

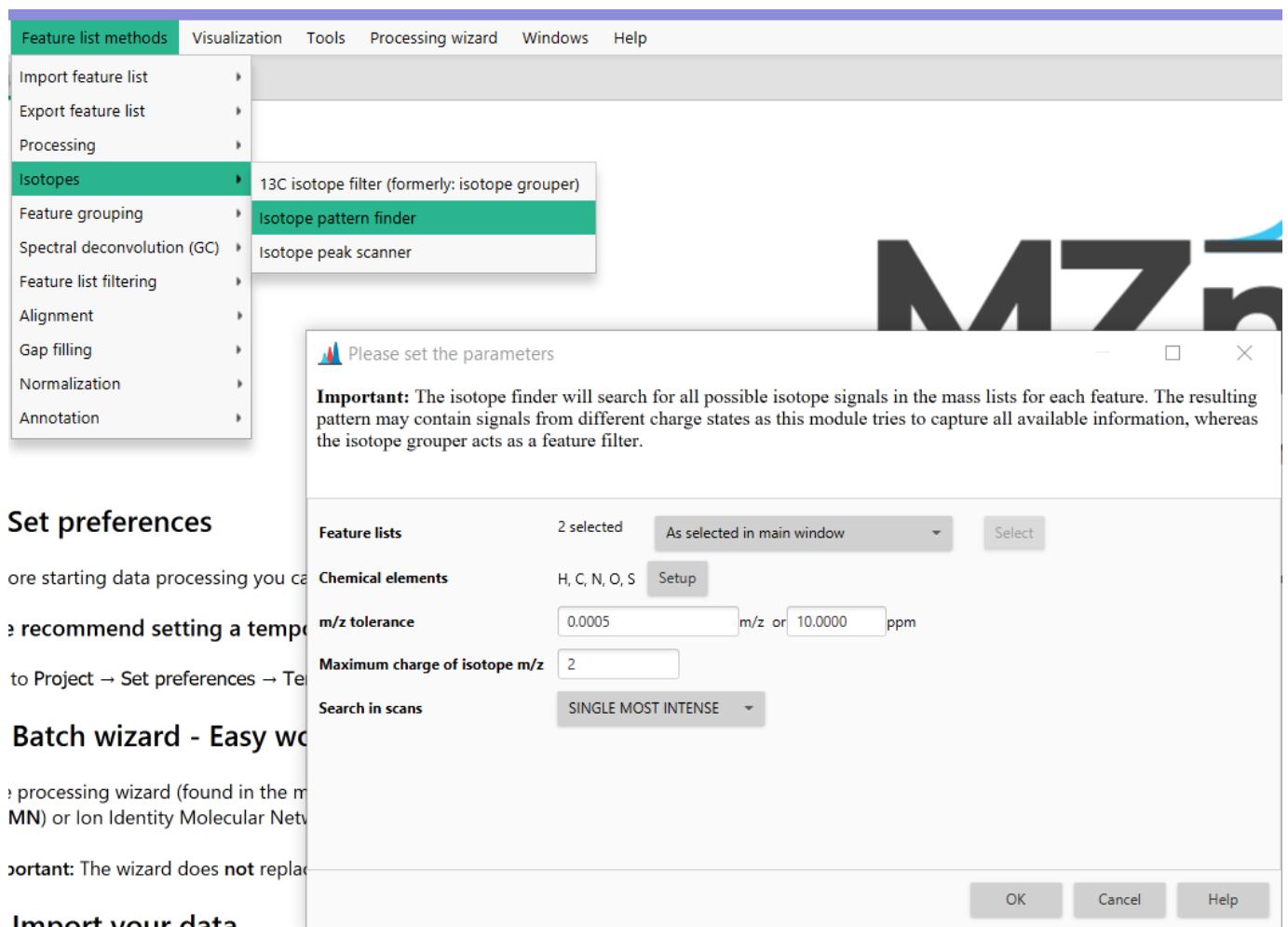
 [lalalana5, omokshyna, tdamiani](#)

11.15.2 Isotope pattern finder

Description

≡ Feature list methods → Isotopes → Isotope pattern finder.

The module searches isotope patterns for each feature in selected feature lists by going back to the mass spectra. Starting from the feature m/z the algorithm will first backtrack any possible preceding isotope signals using a list of delta masses created from elements, their stable isotopes, and an m/z tolerance. For example, a -2 signal might be detected when searching for Br isotopes. In a second step, all picked potential isotope m/z values are used to search next isotope (with higher m/z). This is done for each charge state.



Set preferences

ore starting data processing you can

⇒ recommend setting a temporary

to Project → Set preferences → Temp

Batch wizard - Easy workflow

processing wizard (found in the m/z

MN) or Ion Identity Molecular Network.

Important: The wizard does **not** replace

Import your data

PARAMETERS

Chemical elements

All stable isotopes of the chosen elements are used to create a list of mass differences to search. Signals with this mass difference (m/z difference with different charge states) are then considered as potential isotope signals.

m/z tolerance

Maximum allowed difference between two features' m/z values in order for them to be considered the same. The value is specified both as absolute tolerance (in m/z) and relative tolerance (in ppm). The tolerance range is calculated using maximum of the absolute and relative tolerances.

Maximum charge of isotope m/z

Maximum possible charge of isotope m/z distributions. All present m/z values obtained by dividing isotope masses with 1,2 ...,maxCharge values will be considered. The default value is 1, but insert an integer greater than 1 if you want to consider ions of higher charge states.

Search in scans

Currently, the supported option is "Single most intense", which means the search for isotopes will happen in the single most intense MS scan of each feature.

[NotReallyContributing](#), [Robin Schmid](#), [lalalana5](#), [omokshyna](#)

 November 24, 2022 21:13:07

 [NotReallyContributing](#), [Robin Schmid](#), [lalalana5](#), [omokshyna](#)

11.15.3 Isotope peak scanner

Description

≡ Feature list methods → Isotopes → Isotope peak scanner

This module can scan acquired MS-Data for an isotope pattern of a specified **element combination**. The element combination is given as a string parameter (e.g. "Cl3").

The isotope pattern of this element combination will be calculated and compared to your MS-Data. The result is a new feature list containing just the isotope features. The features will have an **isotope rating**, comparing the found features to the calculated isotope pattern giving the user the option to manually evaluate the results.

Additionally, the features will also be assigned to an isotope combination. Furthermore, the isotope features will be registered as an isotope pattern in MZmine so modules like Sum formula prediction can use the results to compare them to the calculated isotope pattern of predicted sum formulas.

In the resulting feature table, the monoisotopic mass (lowest) will be referred to as "Parent".

ISOTOPE RATING

Rating is calculated using the following formulas:

$$\begin{aligned} R_{m/z,k} &= \frac{m/z_{\text{child}}}{m/z_{\text{parent}} + \Delta M} \\ \frac{I_{\text{parent}}}{A_{\text{parent}}} &= \frac{I_k}{A_k} \\ I_{\text{exp.,k}} &= \frac{I_{\text{parent}}}{A_{\text{parent}}} \cdot A_k \\ R_{I,k} &= \frac{I_{\text{exp.,k}}}{I_k} = \frac{I_{\text{parent}}}{A_{\text{parent}}} \cdot \frac{A_k}{I_k} \end{aligned}$$

where R is the rating, k - the number of an isotope peak, exp. - the calculated intensities, parent - the isotope peak with the lowest mass, child describes all other isotope peaks.

 If any rating is bigger than 1, for example if $m/z_{\text{(parent)}} + \Delta M$ is smaller than $m/z_{\text{(child)}}$, then it will be adjusted by $1/R$ to be comparable.

 If intensity shall be checked as well, m/z and intensity rating will be multiplied resulting in the final rating.

Parameters

Feature lists

The feature list(s) that shall be analyzed.

m/z tolerance

m/z window around the expected isotope features to scan for isotope peaks.

Check RT

If chosen, compares RT of feature to that of a parent. Based on the following parameter of **Retention time tolerance** feature is either filtered out or preserved.

 Invoking this parameter might not make sense for imaging or direct infusion, but is critical for chromatographic data.

Retention time tolerance

Maximum allowed difference between two retention time values

Chemical formula

Element (combination) whose isotope pattern to be searched for. Please enter the two letter Symbol of the elements. (e.g. \"Gd\", \"Cl2Br\")

Auto carbon

If activated, Isotope peak scanner will calculate isotope patterns with variable numbers of carbon specified in Setup. The pattern with the best fitting number of carbon atoms will be chosen for every detected pattern.

 This will greatly increase computation time but helps with unknown-compound-identification.

 Please note that ^{13}C isotope peaks might overlap with hetero atom isotope peaks depending on the resolution of your MS. This influences intensity ratios and will yield **wrong results**, since this prediction is based on intensity ratios of isotope peaks.

 This option yeilds no exact results, but might give a good hint about data.

Charge

Amount and polarity(e.g.: $[\text{M}]^{++}=+1 / [\text{M}]^{--}=-1$).

 This is important for multiply charged molecules because the m/z offset between isotope peaks will halve for molecules with a charge of two.

Min. pattern intensity

The minimum normalized intensity of a peak in the final calculated isotope pattern. Depends on the sensitivity of your MS. This differs from minimum abundance. Min = 0.0, Max = 0.99999.

Merge width(m/z)

This will be used to merge peaks in the calculated isotope pattern if they overlap in the spectrum. Specify in m/z, this depends on the resolution of your mass spectrometer.

Minimum height

Minimum peak height to be considered as an isotope peak.

 Setting this parameter is crucial if you use the **Calculate accurate average** parameter. (see below)

Check intensity ratios

Compare intensity of peaks to the calculated abundance of the isotope pattern.

 It's recommended to check this parameter for more accurate results.

 However, when processing fragment data, and it's unknown how much of an isotope pattern remains charged it might be reasonable to uncheck this. (e.g.: Fragmenting a Cl₈-isotope-pattern-molecule -> If "Element pattern" = Cl₄ this module will recognize everything with Cl₄ or more Cl. However, this will lead to a messy result feature list)

Minimum rating

Minimum rating to be considered as an isotope peak. min = 0.0, max = 1.0

Rating type

Method to calculate the rating with.

- **Highest Intensity** is the standard method and faster.
- **Temporary average** is slower method but could be more accurate for more intense peaks.

Calculate accurate average

This method will use averaged intensities over all mass lists in which ALL relevant masses were detected in. This will only be done for features that match the defined rating-calculation with the given rating.

This will scan all mass lists for the peak closest to the identified isotope peak in the feature list and average the intensity.

If there are no scans that match all criteria avg rating will be -1.0.

 Make sure the mass list is contained in the feature list.

Name suffix

Suffix to be added to feature list name. If "auto" then the module will itself create a suffix.

Results

```
5300--IS PARENT-- BestPattern: C42 Intensity ratios: 1.0:0.45:0.1:0.01 pattern rating: 0.976
5300-Parent ID m/z-shift(ppm): 1.04 I(c)/I(p): 0.42 Identity: [12]C41[13]C Rating: 0.914
5300-Parent ID m/z-shift(ppm): 1.72 I(c)/I(p): 0.1 Identity: [12]C40[13]C2 Rating: 0.989
5300-Parent ID m/z-shift(ppm): -1.95 I(c)/I(p): 0.01 Identity: [12]C39[13]C3 Rating: 0.999
6777--IS PARENT-- BestPattern: C50 Intensity ratios: 1.0:0.54:0.14:0.02 pattern rating: 0.968
6777-Parent ID m/z-shift(ppm): -0.59 I(c)/I(p): 0.52 Identity: [12]C49[13]C Rating: 0.965
6777-Parent ID m/z-shift(ppm): -1.23 I(c)/I(p): 0.14 Identity: [12]C48[13]C2 Rating: 0.997
```

This figure shows an example of the result peak list produced by Isotope peak scanner. It features the detected isotope peaks, a detected m/z ppm-offset, expected (@monoisotopic mass) and detected intensity ratios, the isotope composition, the rating, and (if specified) the average rating.

Troubleshooting

Error: I'm using "Calculate accurate average" but the average rating is always -1.0!

Solution: All isotope features have been detected in the peak list. But they are not in the same mass lists at the same time which makes them incomparable since isotope features should be detected simultaneously, or they might me less intense than the specified "Minimum height".

Error: I'm not getting any results, although I'm sure a specific element is in the scan!

Solution 1: Are you sure every isotope has been detected? How sensitive is your MS? Try to increase values for minimum abundance or minimum pattern intensity. You might need low minimum abundance but high minimum intensity, because a peak of a specific isotope composition might not have been detected due to low relative intensity in the pattern. Check our preview function!

Solution 2: Another solution might be changing the merge width. Check how good the resolution of your MS data is and adjust the merge width to that. If (several) isotope compositions overlap, the intensities have to be merged. You can see a preview in the preview window!

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [omokshyna](#)

11.16 Feature grouping

11.16.1 metaCorrelate feature grouping

≡ Feature list methods → Feature grouping → Correlation grouping (metaCorrelate)

This module groups features based on various properties:

- Retention time
- Feature shape Pearson correlation (optional: Only apply if at least 5 data points)
- Feature height correlation

Recommended citations



When using this modules, please consider citing the corresponding publications:

Schmid, R., Petras, D., Nothias, LF. et al. Ion identity molecular networking for mass spectrometry-based metabolomics in the GNPS environment. Nat Commun 12, 3832 (2021). <https://doi.org/10.1038/s41467-021-23953-9>

Schmid R., Heuckerth S., Korf A., et al. Integrative analysis of multimodal mass spectrometry data in MZmine 3, In Review (2023)

Parameters

The dialog box contains the following parameters:

- Feature lists:** Set to "Those created by previous batch step".
- RT tolerance:** Set to 0.100 (absolute min).
- Sample set:** Set to "No groups".
- Min height:** Set to 0.0E0.
- Intensity correlation threshold:** Set to 0.0E0.
- Min samples filter:** Set to "Setup...".
- Correlation grouping:** Checked, with "Setup..." button.
- Feature height correlation:** Checked, with "Setup..." button.
- Suffix (or auto):** Set to an empty field.
- OK and Cancel buttons:** Standard dialog buttons.

On the right side, there are three stacked configuration panels:

- Min samples in all:** abs= 2, rel= 0.0 %
- Min samples in group:** abs= 0, rel= 0.0 %
- Min %-intensity overlap:** 60.0 %
- Exclude estimated features (gap-filled):** checked

- Min data points:** 5
- Min data points on edge:** 2
- Measure:** PEARSON
- Min feature shape correlation:** 85.0 %
- Min total correlation:** 50.0 %

- Min data points:** 2
- Measure:** PEARSON
- Min correlation:** 70.0 %

RT tolerance

First filter that only checks features that fall within the same retention time window. Should be strict ($\sim\text{FWHM} / 3$) when correlation grouping is disabled. With correlation grouping as a strict filter, the RT tolerance can be wider.

Min height

Minimum height of features to consider. Leave at 0 to use all features that passed the feature detection workflow criteria.

Intensity correlation threshold

Minimum intensity of data points in a feature to compare during feature shape Pearson correlation. All data points below this value are disregarded. Leave at 0 to use the noise levels set in the mass detection steps.

Sample set (optional)

Grouping of samples, only when using *Min samples filter*. Needs project metadata to be set.

Min samples filter

Only group features if they were detected in a minimum number of samples (absolute and relative minimum). Values are provided for - *Min samples in all* (all samples) and - *Min samples in group* (for groups from *Sample set* parameter) - *Min % intensity overlap* defines the percentage of intensity (sum of data point intensity) of the lower abundant feature needs to fall within the RT range of the larger feature - *Exclude estimated features (gap-filled)* excludes gap-filled features from the comparison

Correlation grouping (optional)

Applies a feature shape correlation filter in retention time dimension.

Warning

Only use when having enough data points, i.e., 5 data points total and 2 on each side of the apex. Otherwise, use feature height correlation and a more narrow RT tolerance

Parameters: - *Min data points*: Minimum number of correlated data points - *Min data points on edge*: Minimum number of points on each sides of the apex - *Measure*: Similarity measure (default: Pearson) - *Min feature shape correlation*: Minimum similarity of two features (within the same sample) to be grouped. Pearson $r=85\%$ is default. - *Min total correlation (optional)*: Minimum similarity when taking all the data points from all samples into account

Feature height correlation

Applies a correlation filter by taking all the feature heights across samples for feature pairs.

Warning

Only applicable if the heights are comparable across samples: Similar matrix and ionization conditions

Parameters: - *Min data points*: Minimum number of correlated data points (samples) - *Measure*: Similarity measure (default: Pearson) - *Min correlation*: Minimum similarity

Suffix (or auto) (optional)

Add a suffix to the feature list or just use an automatically generated suffix based on the parameters.

[Robin Schmid, Steffen Heuckeroth](#)

⌚ November 25, 2025 10:29:24

👤 [Robin Schmid, Steffen Heuckeroth](#)

11.16.2 Ion identity networking

≡ Feature list methods → Feature grouping → Ion identity networking

Annotates grouped features (same retention time + optionally, feature shape and height correlation) as ion adducts, in source fragments, and multimers. Also used as input to the [Ion Identity Molecular Networking workflow] (<https://ccms-ucsd.github.io/GNPSDocumentation/fbmn-iin/>) on GNPS. Searches all feature pairs against an ion library for each possible combination. Great in combination with the [Molecular Networking](#) module that calculates fragmentation pattern similarities in mzmine and allows for interactive visualization.



Apply [metaCorrelate](#) before running this module

Recommended citations



When using this modules, please consider citing the corresponding publication(s):

Schmid, R., Petras, D., Nothias, LF. et al. Ion identity molecular networking for mass spectrometry-based metabolomics in the GNPS environment. *Nat Commun* 12, 3832 (2021). <https://doi.org/10.1038/s41467-021-23953-9>

Schmid R., Heuckeroth S., Korf A., et al. Integrative analysis of multimodal mass spectrometry data in MZmine 3, In Review (2023)

Parameters

Feature lists: Those created by previous batch step

m/z tolerance: 0.0010 3.0000 ppm

Check: ONE FEATURE

Min height: 0.0E0

Ion identity library:

Annotation refinement:

m/z tolerance

Intra sample m/z tolerance describes the difference between two ions of the same molecule at the same retention time. This tolerance is usually very small and depends on the mass resolution. Orbitrap instruments for example 3 ppm.

Check

- **ONE FEATURE:** Only one feature needs to match (m/z difference within one sample)
- **ALL FEATURES:** All features need to match (m/z difference within ALL samples)
- **AVERAGE:** Matches the m/z difference of average values

Min height

Minimum height of features to consider. Leave at 0 to use all features that passed the feature detection workflow criteria.

Ion identity library

Defines the ions to search. When two features, applied with two ions, result in the same neutral mass - an annotation is made. New ions can be defined. Adducts (left) and neutral modifications (right) are combined to create the final ion library.



This step should only focus on the main ions that are typically detected in the MS method. Later the *Add more ion identities* module can add strange ions to existing networks.

MS mode POSITIVE

Maximum charge 2

Maximum molecules/cluster 2

Adducts

- [M-H]⁻ (-1.0073)
- [M-e]⁺ (-0.0005)
- ✓ [M+H]⁺ (1.0073)
- ✓ [M+Na]⁺ (22.9892)
- ✓ [M+K]⁺ (38.9632)
- ✓ [M+NH₄]⁺ (18.0338)
- ✓ [M+2H]²⁺ (2.0146)
- [M+Ca]²⁺ (39.9615)
- [M+Fe]²⁺ (55.9338)
- [M+Mg]²⁺ (47.9695)
- [M+H+Na]²⁺ (23.9965)
- [M+H+NH₄]²⁺ (19.0411)
- [M+H+K]²⁺ (39.9704)
- ✓ [M-H+2Na]⁺ (44.9712)
- [M+Ca-H]⁺ (38.9542)
- [M+Fe-H]⁺ (54.9266)

Modifications

- ✓ [M-H₂O] (-18.0106) Import...
- ✓ [M-2H₂O] (-36.0211) Export...
- [M-3H₂O] (-54.0317)
- [M-4H₂O] (-72.0423) (De)select
- [M-5H₂O] (-90.0528)
- [M-NH₃] (-17.0265) Clear
- [M+O] (15.9949) Add
- [M-CO] (-27.9949) Combine
- [M-CO₂] (-43.9898)
- [M-C₂H₄] (-28.0313) Reset
- [M+HFA] (46.0055)
- [M+HAc] (60.0211)
- [M+MeOH] (32.0262)
- [M+ACN] (41.0265)
- [M+IsoProp] (60.0581)

Annotation refinement (optional)

Annotation refinement is optional but should be applied to finalize ion identities, after all subsequent optional steps of *Add more ion identities* steps.

Parameters:

- *Minimum size (optional)*: Only retain ion networks with at least n ions (often 3). The more ions the higher the confidence in the annotation.
- *Delete small networks without major ion (optional)*: Major ions are defined as M+H, M+Na, M+NH₄, M-H₂O+H
- *Delete smaller networks: Link threshold (optional)*: Important parameter to only keep the best annotation, when this annotation is supported by n-1 ions (network size n).
- *Delete networks without monomer (optional)*: Only keep a network if at least one is M+... and not a multimer like 2M, 3M...
- *Delete rows without ion id*: Remove all rows from the feature list that have no ion identity annotation

Minimum size	<input checked="" type="checkbox"/> <input type="text" value="3"/>
Delete small networks without major ion	<input checked="" type="checkbox"/>
Delete smaller networks: Link threshold	<input checked="" type="checkbox"/> <input type="text" value="4"/>
Delete networks without monomer	<input checked="" type="checkbox"/>
Delete rows without ion id	<input checked="" type="checkbox"/>

Robin Schmid, Steffen Heuckeroth

⌚ November 25, 2025 10:29:24

👤 Robin Schmid, Steffen Heuckeroth

11.16.3 Spectral / Molecular Networking

≡ Feature list methods → Feature grouping → Spectral / Molecular Networking

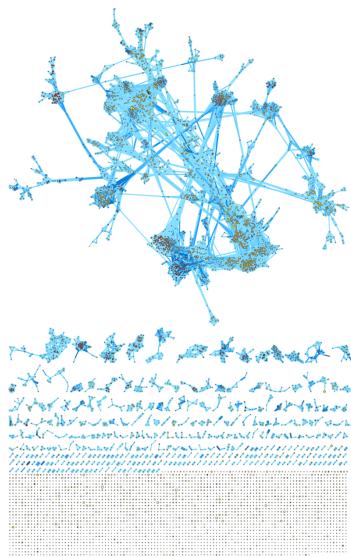


This module runs molecular networking in mzmine. There is no data uploaded to GNPS in this process. For GNPS interoperability, refer to the dedicated export module [export module](#).

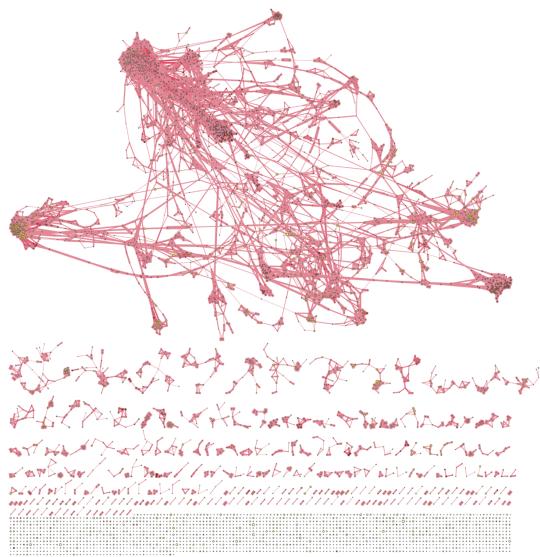
This module integrates multiple popular molecular networking algorithms on MS1 and MS2 spectral relationships like modified cosine, ion identity networking, and machine learning-based models including DreaMS and MS2Deepscore2.

Spectral/Molecular Networking

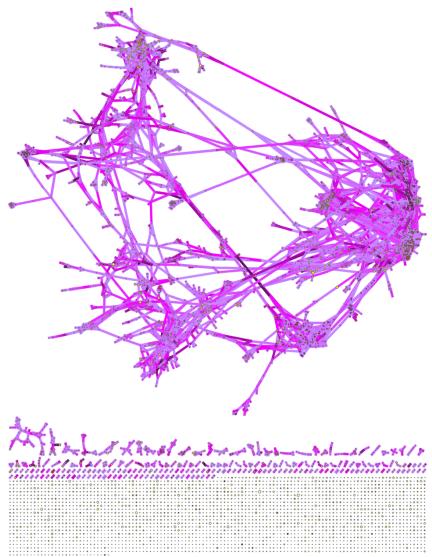
Modified Cosine



DreaMS



MS2Deepscore



PyTorch model, dreams-docs.readthedocs.io

PyTorch model, github.com/matchms/ms2deepscore

Feature-Based Molecular Networking (FBMN) and Ion Identity Molecular Networking (IIMN) group features based on their fragmentation pattern similarity. While similar to spectral library search, the algorithms used for molecular networking are open for modifications of the precursor m/zs. Choose between the modified cosine similarity that is known to connect fragmentation spectra of analog molecular structures, mostly with single modifications, or the MS2Deepscore similarity that aims to converge spectral and structural similarity.

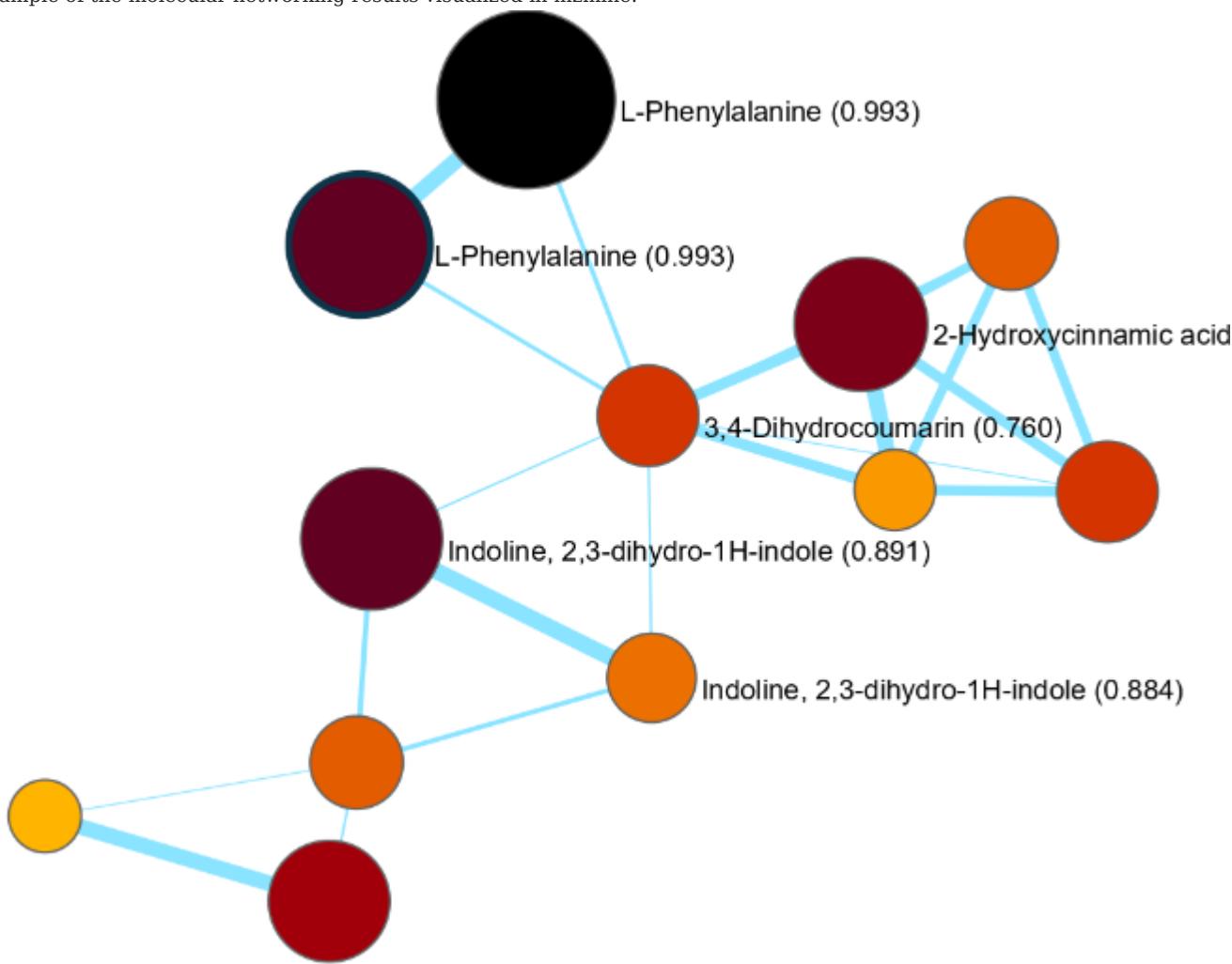
Molecular networking can also be applied to GC/EI-MS data. Cosine similarity is used to group features, based on pseudo MS1 spectra, which can be generated by the [GC-EI Spectral deconvolution module](#).



MS2 scans (LC-MS2) or pseudo spectra (GC/EI-MS) are required for molecular networking. For LC-MS2 make sure to run mass detection on MS2 level. For GC/EI-MS, GC-EI spectral deconvolution is required.

Finally, molecular networks can be visualized in mzmine's interactive network visualizer [interactive_ion_id_netw.md](#). Exports are available to .graphml format and to edges files (.csv).

Example of the molecular networking results visualized in mzmine.



Recommended citations

info

When applying Feature-Based Molecular Networking (FBMN), please consider citing:

When applying Feature-Based Molecular Networking (FBMN), please consider citing:

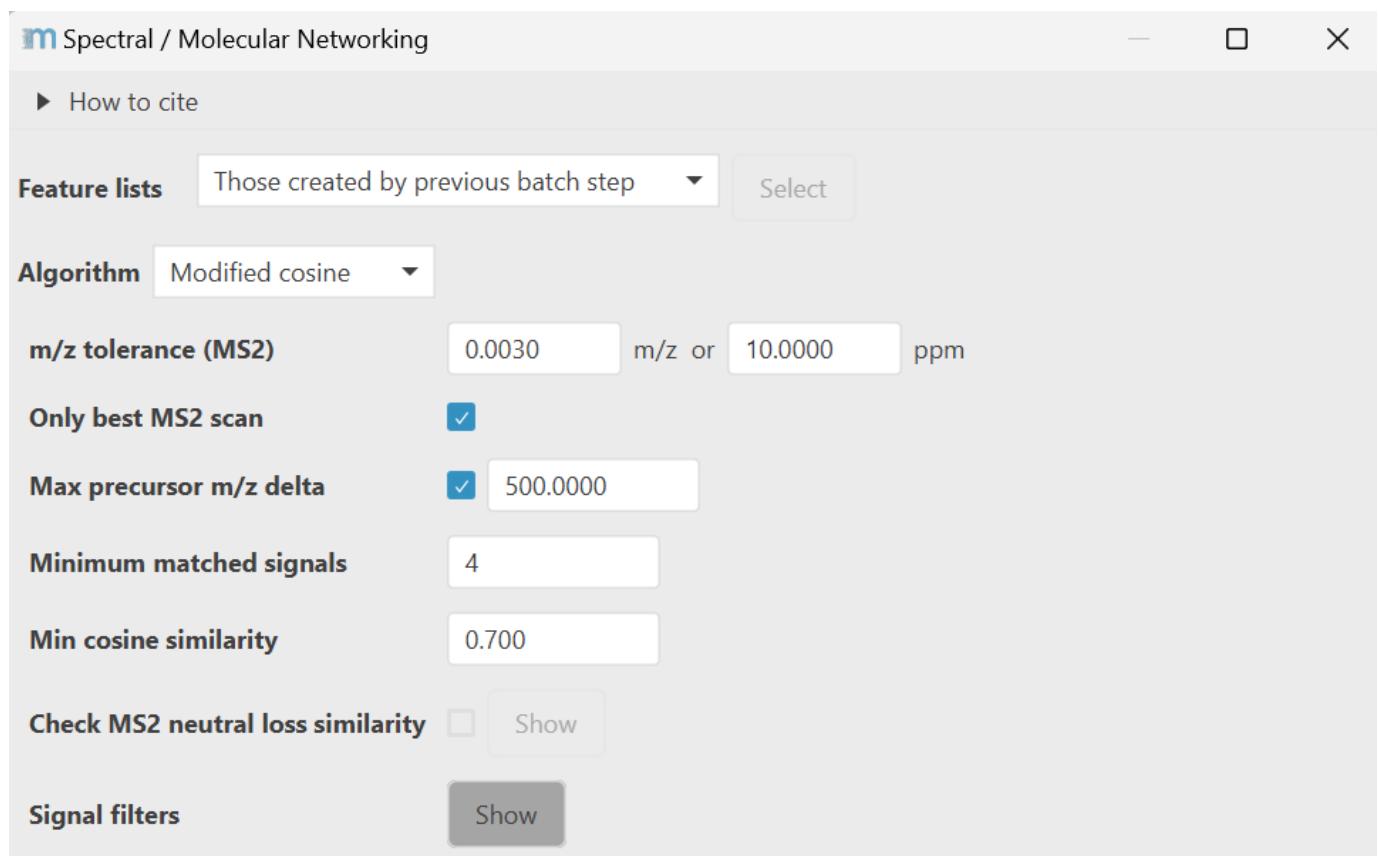
When applying Ion Identity Molecular Networking (IIMN), please consider citing:

When applying for Identity Microbial Networking (IMN), please consider citing:

When using MS?DeepScore, please consider citing:

When using MS2DeepScore, please consider citing:
de Jonge N, Joas D, Truong J, van der Hooft JH, Huber F. bioRxiv 2024; 03:25:586580

Parameters



The parameter dialog changes the parameters based on the selected algorithm. Here shown for the modified cosine algorithm.

ALGORITHM: (MODIFIED) COSINE

The modified cosine similarity aligns spectra by grouping signals within the same m/z tolerance. Compared to spectral library matching, which allows only direct matches of the same fragment signals - modified cosine also considers matching neutral loss mass differences to the precursor ions. Therefore, signal pairs may be derived from the same neutral loss or the same produced fragment ion, allowing for structural analogs to cluster in the same sub networks. Signals without matching partner will be scored against 0 intensity. All spectra are square root transformed to balance the contribution of medium and high intensity signals.

m/z tolerance

Tolerance to group fragment signals by m/z. Fragments are grouped by direct match of m/z or by having the same neutral loss mass difference to the precursor ion. Most intense fragment signals are considered first.

Merge & select fragment scans

This parameter controls how fragment spectra are filtered, merged, and selected for downstream analysis (see [detailed description](#)). Briefly, either choose preset based spectral merging, input scans without merging, or an advanced setup for more options.

Molecular networking checks all features against each other - this is an expensive task and therefore, mzmine recommends using the **Single scan: Merged across energies** preset. For more density and checks, consider using the preset **Representative scans** which also adds scans for each fragmentation energy. The more scans the longer the task.

Max precursor m/z delta

If checked only features with precursor m/z difference lower than the specified value are compared.

Minimum matched signals

Minimum required signals to match two fragmentation spectra. Default 4 signals for small molecules. Use higher number of signals for more confident matches and compound classes with richer fragmentation data.

Min cosine similarity

Minimum cosine for matches between fragmentation spectra.

Signal filters:

Additional signal filters that reduce the spectral complexity while retaining most of the spectral information. This speeds up the matching for spectra with many fragment signals.

Remove residual precursor m/z

Remove all signals +- the precursor m/z.

Crop to top N signals

First applies the intensity filter (see below) then crops the remaining signals to the top 250.

Signal threshold (intensity filter)

Above N signals, start using the intensity percentile filter (see below). This does **not** filter to 50 signals so the final number of signals may be higher.

Intensity filter at >N signals

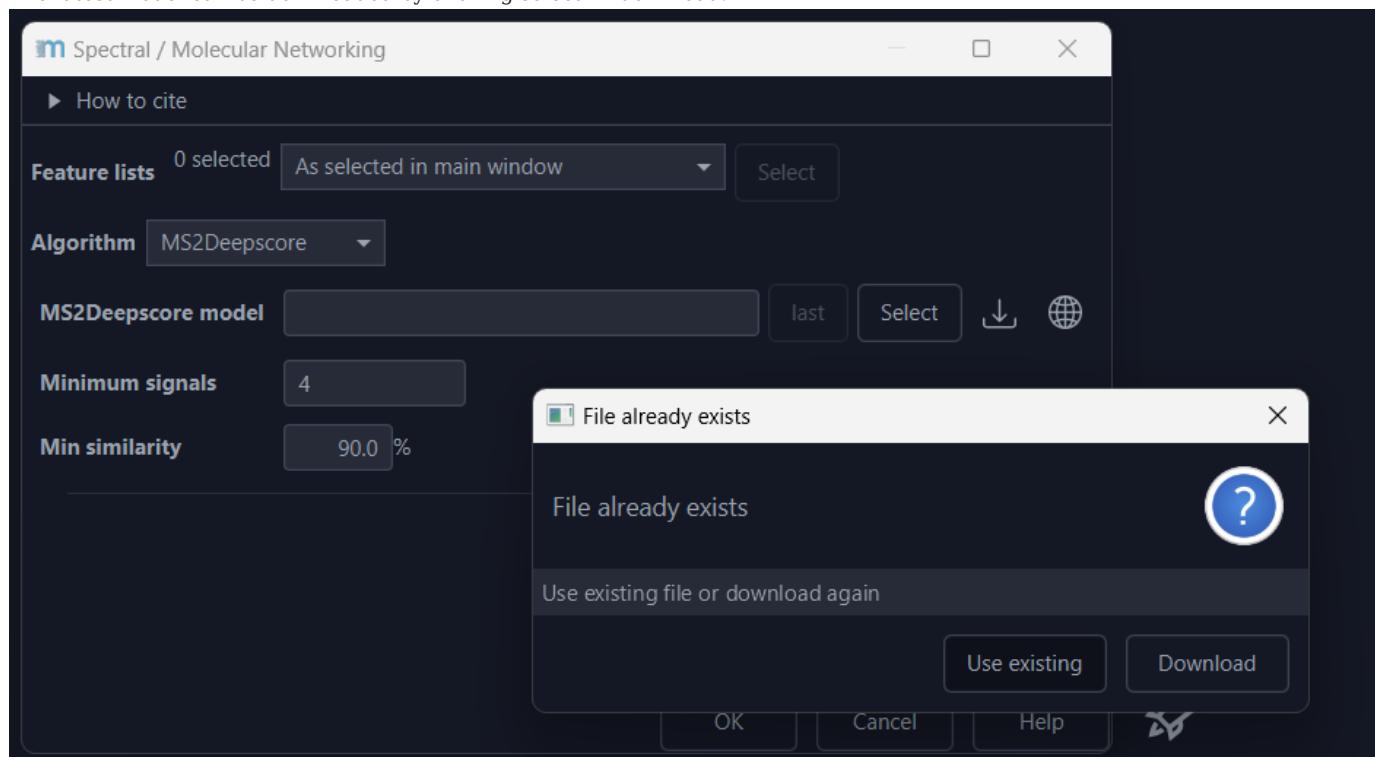
Remove signals so that X% of intensity is retained. This removes lower abundant signals that play a minor role in matching. Overall this can speed up matching.

ALGORITHM: MS2DEEPScore

This algorithm uses a deep neural network to predict embeddings that aim for spectral similarities that approach structural similarities. The module allows loading of pre-trained models in the PyTorch script format. More information about MS2DeepScore can be found on [github](#). If you use MS2DeepScore, please cite [de Jonge et al. 2024](#) and [Huber et al. 2021](#).

MS2DeepScore model

The latest model can be downloaded by clicking select -> download:



Alternatively the pre-trained MS2Deepscore model can be downloaded from <https://zenodo.org/records/12628369> and contains the PyTorch script model (.pt) which should be selected in this parameter. The same folder should contain the settings.json file also supplied on Zenodo. If you want to convert a different MS2Deepscore model to a java/mzmine compatible format, you can find the relevant scripts in [this repository](#).

Minimum signals

Minimum number of signals (fragments) to consider a fragment spectrum. This is **not** the number of matched signals because MS2Deepscore works by comparing its embeddings.

Min similarity

Minimum MS2Deepscore spectral similarity to consider matched. The score is scaled from 0-1 with 1 being the best match.

Ansgar Korf, Niek de Jonge, Robin Schmid, Steffen Heuckeroth

 November 24, 2025 21:59:03

 Ansgar Korf, Niek de Jonge, Robin Schmid, Steffen Heuckeroth

11.17 Feature list filtering

11.17.1 Duplicate feature filter

Description

≡ Feature list method → Feature list filtering → Duplicate feature filter.

This filter can help eliminate misaligned feature list rows after the gap-filling process.

It has three different **modes**:

- **Old average (the old filter):**

Keeps only the feature list row with the maximum average area. Compares rows with the average m/z and RT.

- **New average:**

Compares rows with the average m/z and RT and creates a consensus row. Two peaks are considered duplicates when their average m/z and retention time differences are lower than the tolerances set by the user.

When two (or more) duplicates are found, a **consensus row** is created with the lowest row ID of all duplicates. For this consensus row, all DETECTED features are favored over ESTIMATED (gap-filled) and ESTIMATED are favored over UNKNOWN. Furthermore, if there are only ESTIMATED features in a raw data file, the highest is chosen.

- **Single feature:**

Compares rows on a raw data file basis. Marks rows as duplicates if they share one feature within the RT and m/z tolerance in the same raw data file. Creates a consensus row.

Parameters

Name suffix

This is the suffix to identify the new aligned peak list.

Filter mode

User can choose one of three modes: old average, new average, and single feature.

m/z tolerance

Maximum m/z difference between duplicate peaks.

RT tolerance

Maximum retention time difference between duplicate peaks.

Require same identification

If the checkbox is selected duplicate peaks must have the same identification.

Original feature list

Can be either processed in place of, kept or replaced.

[lalalana5, omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [lalalana5, omokshyna](#)

11.17.2 Feature list rows filter

Description

Feature list methods → Feature list filtering → Feature list rows filter

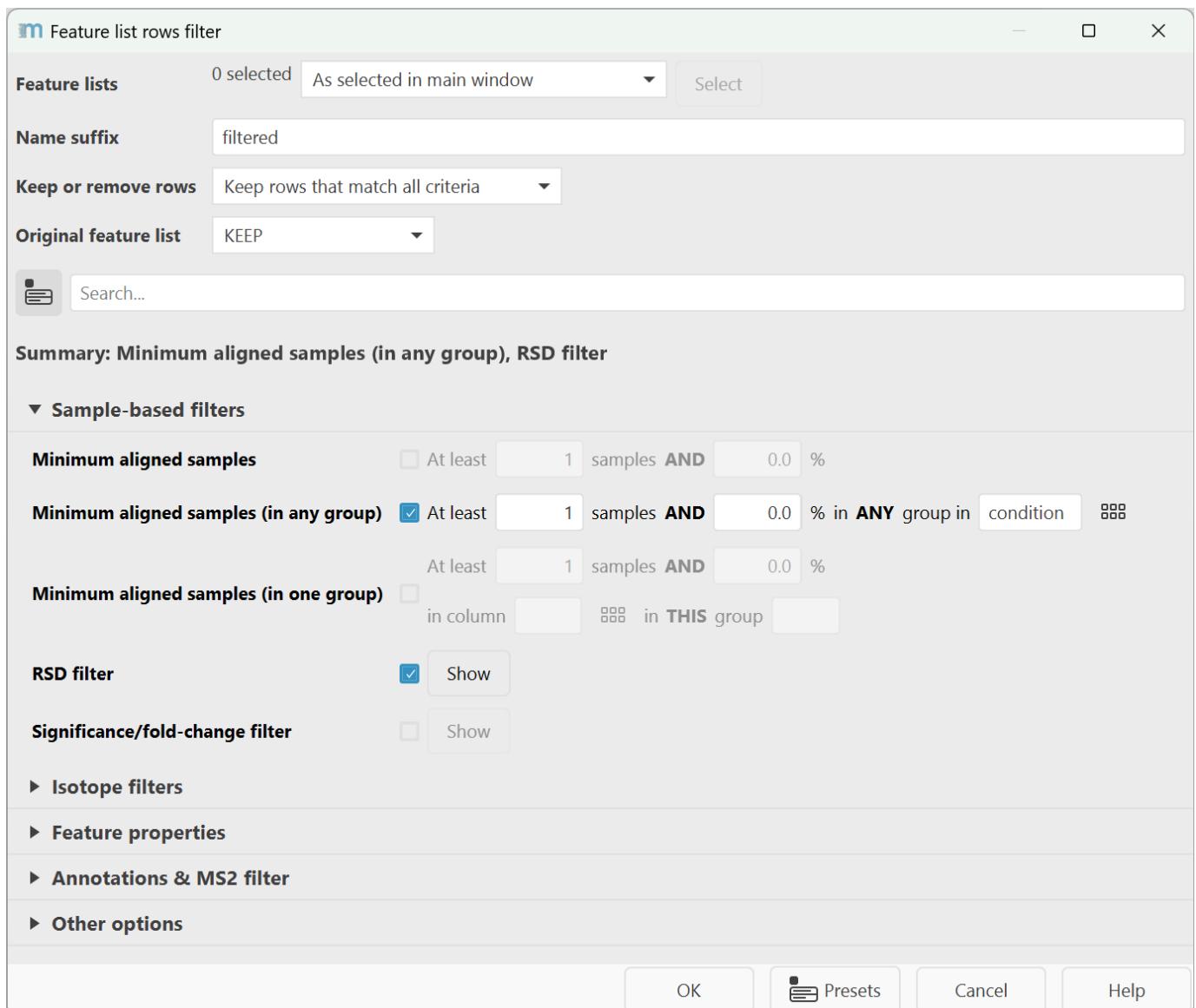
This module provides filters for many feature list row properties. The final resulting feature list contains rows that **match all criteria** or those that match none, depending on the **Keep or remove rows** parameter.

A range of different requirements can be set, such as the minimum number of features in the row, the minimum number of features in an isotope pattern, peak duration etc.

Feature properties are referred to the representative row values for aligned feature list with multiple samples. This means that values like the feature's retention time are averages. Other filters may use statistics across all samples.

Parameters

The dialog holds the most important general parameters on the top to define the behavior of these filters. All other parameters are grouped by topic and can be searched by the search bar. Selected parameters are then displayed in the summary as links that, on mouse click, provide a quick filter to find the parameters for setup.



Feature lists

Select the feature lists to apply the filters to.

Name suffix

Suffix to be added to the resulting feature list name.

Keep or remove rows

Select to either keep or remove the rows that match all defined criteria. Default is to keep rows that match criteria.

Original feature list

It can be either processed in place, kept, or removed.

SAMPLE-BASED FILTERS**Minimum aligned samples (previously Minimum features in a row)**

Minimum number of samples a feature needs to be detected in a row across all samples to match. The value can be set both as an absolute number and a percentage of samples that both need to match. The percentage can help to scale the filter with increasing study sizes.

Minimum aligned samples (in any group)

Minimum number of samples a feature needs to be detected in **any group** in a **metadata column**. The percentage is applied for each group separately. The value can be set both as an absolute number and a percentage of samples that both need to match.

Minimum aligned samples (in one group)

Minimum number of samples a feature needs to be detected in one **specific group** in a **metadata column**. The value can be set both as an absolute number and a percentage of samples that both need to match.

RSD filter

Filters by the relative standard deviation of the abundance (height or area) in one metadata group. This filter is often used with pooled QCs.

Define a sample group by selecting a metadata column and a group in that metadata column that may only show a maximum relative standard deviation (= coefficient of variation). Only rows that show a CV below the given value, will be retained. A good sample group for this application are pooled QC samples. If a feature is not detected in all QC samples, it will be removed, unless the "Keep undetected" option is enabled.

Significance/fold-change filter ("Volcano plot filter")

Filter that works similar to the volcano plot on both the significance and fold-change between samples from two groups. Requires internal setup of the abundance measure, missing value imputation, significance test, and thresholds for the p-value and log2(fold-change).

Groups are selected from the project metadata, two groups in the same column.

ISOTOPE FILTERS

Isotope filters require that the **13C isotope filter** and/or **isotope finder** modules were applied.

Minimum features in an isotope pattern

The required minimum number of signals in a row's isotope pattern. This is a basic filter and stems from mzmine 2 that only considers the number of signals. It is better to use the more sophisticated **Validate 13C isotope pattern** filter.

Validate 13C isotope pattern

Searches for a +1 13C signal (considering possible charge states) within an estimated range of carbon atoms. Optionally: Detect and filter rows that are 13C isotopes by searching for the preceding-1 signal.

Remove redundant isotope rows

Removes rows that are not the most intense or the monoisotopic peak in an isotope pattern.

FEATURE PROPERTIES**m/z**

Range of acceptable (average) m/z values in a row across all samples.

Retention time

Range of acceptable (average) retention times in minutes.

Chromatographic width (previously Features duration range)

Range of acceptable (average) feature durations as chromatographic widths in minutes.

Chromatographic FWHM

Range of acceptable chromatographic full-width at half maximum (FWHM) in minutes.

Charge

Range of row Charge

⚠ Please, run isotope finder to determine charge.

Mass defect

Mass defect as a feature filter can be used for selective detection of compounds of interest, and the values accepted are 0.314-0.5 or 0.90-0.15.

Kendrick mass defect

Filter features in a Kendrick mass defect (KMD) range. For more details see [Kendrick mass defect](#).

If KMD is used, the following parameters can be changed in the setup.

- **Kendrick mass defect** Permissible range of a Kendrick mass defect per row
- **Kendrick mass base** Enter a sum formula for a Kendrick mass base, e.g. "CH2"
- **Shift** Enter a shift for shift dependent KMD filtering
- **Charge** Enter a charge for charge-dependent KMD filtering
- **Divisor** Enter a divisor for fractional base unit dependent KMD filtering
- **Use Remainder of Kendrick mass** Use Remainder of Kendrick mass (RKM) instead of Kendrick mass defect (KMD)

ANNOTATIONS & MS2 FILTER**Never remove rows with MS2**

If checked, all rows with MS2 are retained without applying any further filters on them.

Require MS2 scan (previously Feature with MS2 scan)

If checked, only features that have MS2 scan will be kept.

Never remove annotated rows

If checked, keep all rows with annotation and skip further filtering for them.

Flexible filter (e.g., annotation)

This filter is defined by a search type, matching mode, and textual query. In general, annotation search types will match against any annotation like spectral library match, lipid match, or other feature annotations from other modules. All annotations are considered, this means that sometimes a row will match the filter, but the reason is not the primary annotation visible but a secondary one that may be hidden in the list of, e.g., library matches. The **comment (row)** search with modes ** any and all** are useful options for row tagging and searching. Just insert comments in the feature table comment column and use this filter.

[More details](#)

Only identified?

If the checkbox is selected, only identified compounds will be retained.

Text in identity

Only rows that contain this text in their annotations will be retained. Check out the **flexible filter** for better filtering options.

Text in comment

Only rows that contain this text in their comment field will be retained. Check out the **flexible filter** for better filtering options like substring matching against the comment (row) type.

OTHER OPTIONS

Require other detector correlation

If checked, the rows that do not have at least one feature that is correlated to a signal of another detector will be removed.

Reset the feature number ID

If checked, row IDs will be reset. This can be helpful for downstream tools that require consecutive row IDs. Otherwise, the original IDs may provide better traceability throughout the mzmine data processing.

[Robin Schmid](#), [Steffen Heuckeroth](#), [lalalana5](#), [omokshyna](#)

 October 21, 2025 15:45:16

 [Robin Schmid](#), [Steffen Heuckeroth](#), [lalalana5](#), [omokshyna](#)

11.17.3 Feature filter

Description

≡ Feature list methods → Feature list filtering → Feature filter

This module deletes features in a selected feature list that do not meet the requirements defined by the user. The filter is performed separately on each raw data file in the peak list.

Parameters

Name suffix

Suffix to be added to feature list name.

Duration

Peaks with a duration outside the entered range will be removed.

Area

Peaks with an area outside the entered range will be removed.

Height

Peaks with a height outside the entered range will be removed.

data points

Peaks with fewer data points than the entered range will be removed.

FWHM

Peaks with a FWHM outside the entered range will be removed.

Tailing factor

Peaks with a tailing factor outside the entered range will be removed.

Asymmetry factor

Peaks with an asymmetry factor outside the entered range will be removed.

Minimum RT shape score

(since mzmine 4.7)

Tries to fit a chromatographic peak with a Gaussian, a bi-Gaussian, or two Gaussian functions. The Gaussian function models an ideal case. The bi-Gaussian models each side with by a different gaussian function, creating a fronting or tailing gaussian. The model with two Gaussian functions describes a feature that has not been completely resolved during the [feature resolving step](#). Features with less than 5 points will be removed automatically.

Minimum mobilogram shape score

(since mzmine 4.7)

Tries to fit a mobilogram with a Gaussian, a bi-Gaussian, or two Gaussian functions. The Gaussian function models an ideal case. The bi-Gaussian models each side with by a different gaussian function, creating a fronting or tailing gaussian. The model with two Gaussian functions describes a feature that has not been completely resolved during the [feature resolving step](#). Features with less than 5 points will be removed automatically.

Top to edge ratio

(since mzmine 4.7)

Determines the top-to-edge ratio of the chromatographic peak by dividing the highest by the lowest point of the feature (see [feature resolving](#)).

Keep only features with MS/MS scans

Peaks without any MS/MS scans will be removed.

Remove source feature list after filtering

If the checkbox is selected, the source feature list will be removed, and the filtered version will remain.

[Steffen Heuckeroth, omokshyna](#)

 May 27, 2025 12:10:17

 [Steffen Heuckeroth, omokshyna](#)

11.17.4 Peak comparison rows filter

Description

≡ Feature list methods → Feature list filtering → Peak comparison rows filter

This method removes certain rows from an aligned feature list based on peak comparisons in two columns.

11.17.5 Parameters

Name suffix

Suffix to be added to feature list name.

1st peak column to compare (zero indexed)

Index of second column for comparison, e.g. "0".

2nd peak column to compare (zero indexed)

Index of second column for comparison, e.g. "1".

Fold change range : log2(peak1/peak2)

Return peaks with a fold change within this range.

m/z difference range : peak1 to peak2 (ppm)

Return peaks with a m/z difference within this range.

RT difference range : peak1 to peak2 (min)

Return peaks with an RT difference within this range.

Remove source feature list after filtering

If checked, the original feature list will be removed leaving only the filtered version.

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

11.17.6 Neutral loss filter

Description

≡ Feature list methods → Feature list filtering → Neutral loss filter

This module can scan acquired MS data for neutral losses.

The result is a new feature list containing only features and their neutral-loss equivalents.

The results will be displayed in the following manner:

- The feature with higher mass without the neutral loss will be named "**Parent**". The description will contain the ID of the corresponding feature with the neutral loss.
- The feature with lower mass will be named "**Child**". The description of the child feature will contain the ID of the parent feature and a ppm-shift relative to the calculated mass.

Parameters

m/z tolerance

m/z window size around the expected features.

Check RT

Specify whether the algorithm should compare RT to those of a parent in found peaks or not.

 We recommend to not use it for direct infusion.

 If evaluating chromatographic data we recommend to use this parameter.

Retention time tolerance

Tolerance range of the retention time.

Minimum height

Minimum height of a feature to be considered a parent or child.

Neutral loss (m/z)

m/z ratio of the neutral loss to be detected. If the text box of "Molecule" is not left blank, this parameter will be ignored.

Molecule

Element combination/Molecule of the neutral loss (e.g. HI) This module will calculate the mass of the given molecule and ignore the input in the "Neutral loss" text box.

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

11.17.7 m/z mobility region extraction

Description

≡ Feature list methods → Feature list filtering → mobility-m/z region filtering

Extracts subregions of interest from m/z-mobility regions.

Parameters

Region

Regions to extract.

Mobility/CCS

Defines if mobility or mz shall be used for extraction.

Suffix

The suffix of newly created feature lists.

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [omokshyna](#)

11.17.8 Chromatogram blank subtraction

Description

⚠ Warning

This module should be applied after chromatogram builder and before the feature resolving steps. Project metadata is used to define blank samples.

💡 Tip

This module works on the chromatogram level (quite early in the workflow) and can help reduce the processing time and false features detected. Also check out the [Feature list blank subtraction](#) that works on aligned feature lists and requires blank samples in the aligned list.

≡ Feature list methods → Feature list filtering → Chromatogram blank subtraction

Subtracts blank chromatograms from all other samples. Blank chromatograms are constructed by maximizing intensity over multiple blank files. Uses **sample metadata** to define **blank** samples in the **mzmine_sample_type** column.

By default, the blank samples will be not processed further because they will be empty after blank subtraction.

Parameters

m/z tolerance (sample-to-sample)

Matching tolerance to merge blank chromatograms and to then subtract from matching sample chromatograms.

Suffix

The suffix for the new feature list.

Original feature list

Defines how to handle the original feature list.

[Robin Schmid](#)

⌚ June 13, 2025 13:24:03

👤 [Robin Schmid](#)

11.17.9 Feature list blank subtraction

Description

⚠ This module needs aligned feature lists as an input. This aligned feature list should contain features from blank/control.

≡ Feature list methods → Feature list filtering → Feature list blank subtraction

Subtracts the features appearing in (procedural) blank measurements feature list from an aligned feature list.

Parameters

Minimum # of detection in blanks

Specifies in how many of the blank files a peak has to be detected.

Fold change increase

Specifies a percentage of increase of the intensity of a feature. If the intensity in the list to be filtered increases more than the given percentage to the blank, it will not be deleted from the feature list.

Suffix

The suffix for the new feature list.

omokshyna

⌚ November 24, 2022 21:13:07

👤 omokshyna

11.17.10 Split aligned feature list

This module can be used to split an aligned feature list into separate feature lists, either on a per-sample basis or on a metadata group basis.

The split feature tables will maintain the feature IDs of the aligned feature list. This makes this module useful for processing data where e.g. report generation or spectral library matching is required on a per-sample basis, without having to rely on post-alignment outside of mzmine, since all previously aligned features have the same ID.

Parameters

Feature lists

Select the aligned feature list that shall be split.

Split by metadata group

If disabled, each raw data file in the aligned feature list will be placed in an individual feature list.

If enabled, the aligned feature list will be split by the selected group in the metadata table. Each metadata group will review its own feature list.

 October 20, 2025 08:48:37

 Steffen Heuckeroth

11.18 Alignment

11.18.1 Join aligner

Description

≡ Feature list methods → Alignment → Join aligner

This method aligns detected peaks in different samples through a **match score**. This score is calculated based on the mass and retention time of each peak and ranges of tolerance specified in the parameter setup dialog.

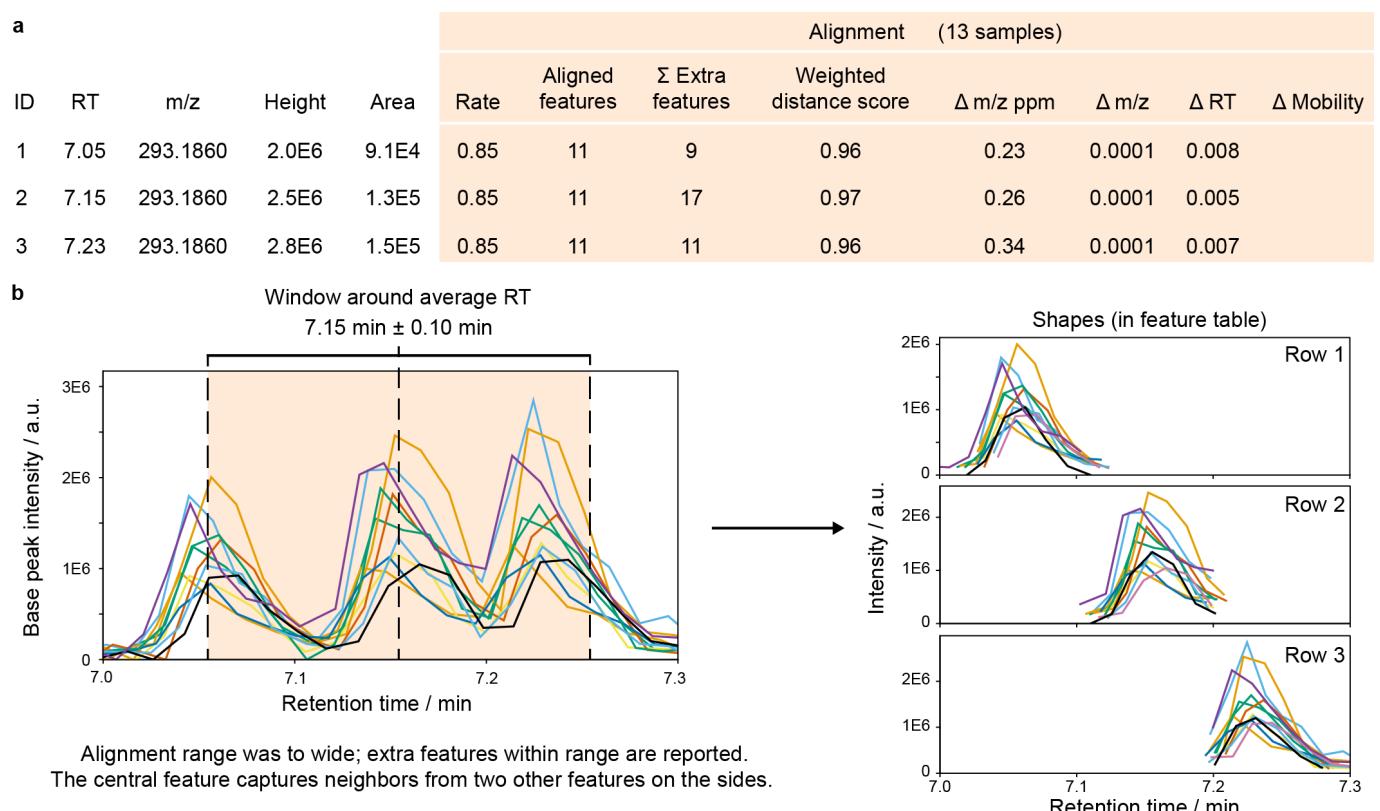


Fig. The alignment columns in feature tables (MZmine 3.3.0). **a**, After alignment, new columns provide results on the alignment, including the alignment rate (aligned/total samples) and mean differences between all samples to the mean m/z, RT, and mobility. The weighted distance score reflects the alignment score, calculated for the final aligned feature. The summed extra features counts all additional features that were found within the m/z, RT, and mobility tolerances. **b**, Show the LC-MS EICs of 13 samples extracted with the alignment m/z tolerance around feature ID2. With a wide RT tolerance, 17 additional LC-MS features were found that indicate closely eluting isomers or other ions that might interfere with the alignment.

Please set the parameters

Feature lists Aligned feature list As selected in main window Select

Feature list name Aligned feature list

m/z tolerance 0.0010 m/z or 5.0000 ppm

Weight for m/z 3

Retention time tolerance 0.100 absolute (min)

Weight for RT 1

Mobility tolerance

Mobility weight

Require same charge state

Require same ID

Compare isotope pattern Setup

Compare spectra similarity Setup

Original feature list KEEP ▾

OK Cancel Help

ALGORITHM**Input**

The peak alignment algorithm uses:

1. A **master list of peaks** (L) against which every new **sample** (S_j) will be matched.

When aligning peaks from multiple samples, the master list is initially set to the first sample. Subsequently, it becomes a combination of samples aligned this far.

Master list contains the samples as the columns and the matching peaks as the rows.

2. For every row i in L , a two-dimensional window (a window size is selected by the user), called **Alignment window (AW)**, defines the ranges of m/z and RT.

The window is centered around the average of m/z and RT of all the individual peaks in the row.

3. A score function is used to compute the similarity of peaks between L and the new sample S_j inside the alignment window.

The **score function** computes the similarity based on the similarities in m/z, retention time, and (optionally) on identification, and isotope patterns between the peaks to be matched.

The score is calculated as follows:

$$\text{score} = \frac{1 - \text{MZ}_{\{\text{difference}\}}}{\text{MZ}_{\{\text{tolerance}\}}} \times \text{MZ}_{\{\text{weight}\}} + \frac{1 - \text{RT}_{\{\text{difference}\}}}{\text{RT}_{\{\text{tolerance}\}}} \times \text{RT}_{\{\text{weight}\}}$$
Steps

The algorithm works as follows:

- It iterates through the rows of L .
- For each row, it looks for peaks within the alignment window in S_j that has to be aligned with L .
- A score is calculated for each possible match
- The pair getting the best score will be aligned.

Parameters**Feature list name**

Name of the new aligned peak list.

m/z tolerance

Maximum allowed difference between two m/z values in order for them to be considered the same and thus the peaks aligned.

The value is specified both as absolute tolerance (in m/z) and relative tolerance (in ppm).

The tolerance range is calculated using maximum of the absolute and relative tolerances for possible peaks to be aligned.

Weight for m/z

This is the assigned weight for m/z difference at the moment of match score calculation between peak rows, as can be seen in the aforementioned formula. Only in cases where there is a perfect match of m/z values, the score receives the complete m/z weight. Generally, higher weight is given to m/z values than to RT values.

Retention time tolerance

Maximum allowed difference between two retention times in order for them to be considered the same and thus peaks aligned. Maximum RT difference can be defined either using absolute or relative value.

Weight for RT

This is the assigned weight for RT difference at the moment of match score calculation between peak rows. Only in cases where there is a perfect match of RT values, the score receives the complete RT weight.

Mobility tolerance

In case of IM data, the user can determine the mobility tolerance. If checked, this parameter specifies the tolerance range for matching the mobility values.

Mobility weight

Score for perfectly matching mobility values. Only taken into account if "Mobility tolerance" is activated. Furthermore, score calculation that is mentioned in the **Algorithm** is then modified to account for the mobility as well. Mobility tolerance and weight are accounted for in the same manner as m/z and RT parameters.

Require same charge state*Optional parameter*

If checked, only rows having same charge can be aligned.

Require same ID*Optional parameter*

If checked, only rows having same compound identities (or no identities) can be aligned.

Compare isotope pattern*Optional parameter*

If both peaks represent an isotope pattern, checking this box will add isotope pattern score to the match score calculation. Additionally, the user can set up **isotope m/z tolerance** which defines what isotopes would be considered same when comparing two isotopic patterns, **minimum absolute intensity** below which isotopes will be ignored and **minimum score %** between isotope patterns that has to be satisfied in order for the match to not be discarded.

Compare spectra similarity*Optional parameter*

Compare MS1 or MS2 scans similarity. Select the m/z tolerance, MS level and spectra similarity algorithm. Only features meeting this criteria will be aligned. See [compare spectra similarity](#) for additional information.

Original feature list

The user can choose to either KEEP the original feature list and generate a new processed one, or REMOVE the original feature list with the processed one. Generally, you would keep the original feature list, but opting for REMOVE will save memory.

[Robin Schmid](#), [lalalana5](#), [lalalana5](#), [omokshyna](#)

 December 10, 2022 05:01:51

 [Robin Schmid](#), [lalalana5](#), [lalalana5](#), [omokshyna](#)

11.18.2 Merge lists

Description

≡ Feature list methods → Alignment → Merge lists

This method merges feature lists by appending all rows into a new list.

⚠ Perform alignment before to align all features from comparable samples and use this method to merge feature lists that should not be aligned: e.g., positive and negative mode data.

Parameters

Feature lists

Feature lists this module will take as an input

Feature list name

Resulting feature list name

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [omokshyna](#)

11.18.3 RANSAC peak list aligner

Description

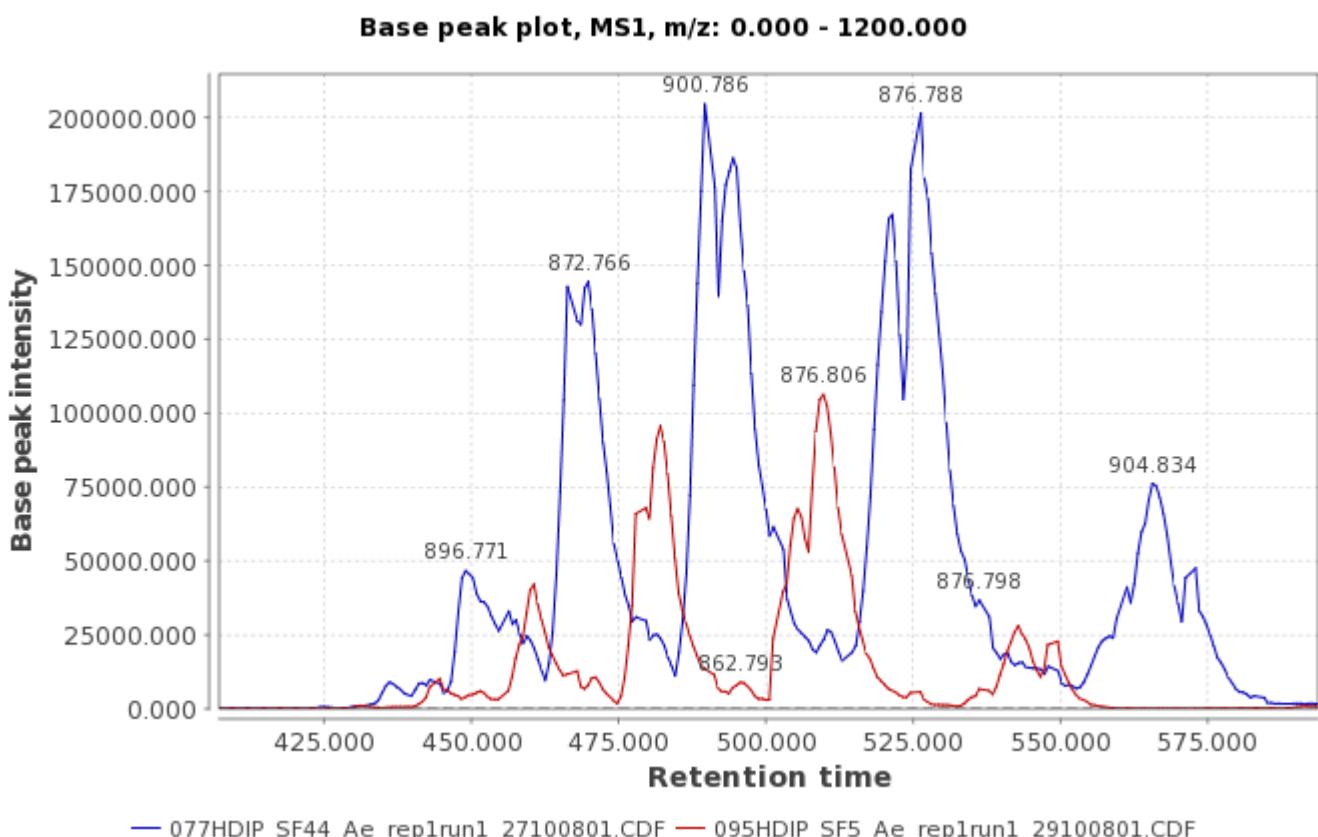
≡ Feature list methods → Alignment → RANSAC aligner

This method is an extension of the **Join aligner** method.

The alignment of each sample is done against a **master peak list**, which is taken from the first sample in the first round and from the average of all aligned peak lists in every round.

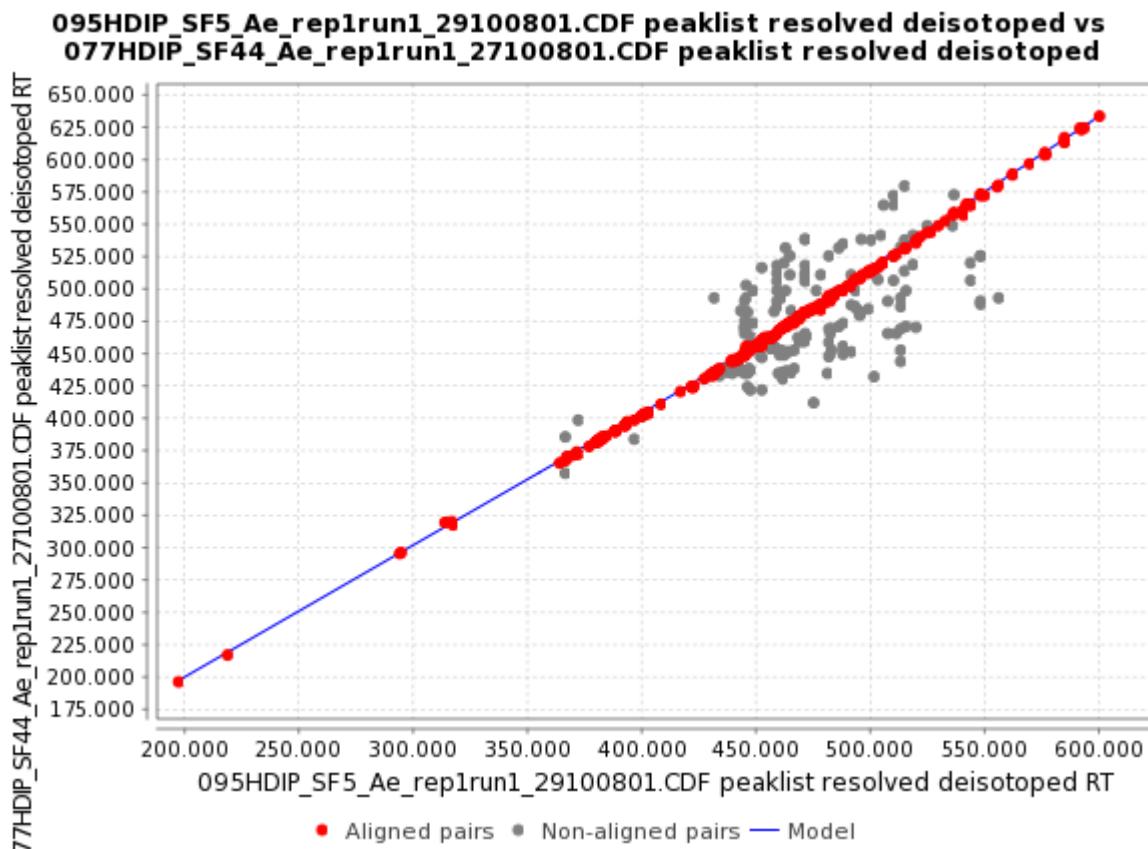
It corrects any linear or non-linear deviation in the retention time of the chromatograms by creating a model of this deviation.

This picture shows an example of two samples with a non-linear deviation in the retention time:



The "**deviation**" model for the retention time is created by taking some corresponding points from the peak list of two samples using the RANSAC algorithm (<http://en.wikipedia.org/wiki/RANSAC>) and using a non-linear regression method to fit the model.

This picture shows a preview of the model with the red dots representing the aligned peaks taken using RANSAC algorithm, and the blue line represents the fitted model using a non-linear regression.



Using this model, the algorithm can predict the shift in the retention time along all the peak list and use the match score function, used also in **Join Align** algorithm, to match the peaks.

This score is calculated based on the mass and retention time of each peak and ranges of tolerance stipulated in the parameter setup dialog.

Parameters

Feature list name

The name of the new aligned feature list.

m/z tolerance

This value sets the range, in terms of m/z, to verify for possible peak rows to be aligned. Maximum allowed m/z difference.

RT tolerance

This value sets the range, in terms of retention time, to create the model using RANSAC and non-linear regression algorithm. Maximum allowed retention time difference.

RT tolerance after correction

This value sets the range, in terms of retention time, to verify for possible peak rows to be aligned. Maximum allowed retention time difference.

RANSAC Iterations

Maximum number of iterations allowed in the algorithm to find the right model consistent in all the pairs of aligned peaks.

- When the value is 0, the number of iterations (k) will be estimate automatically.

Minimum Number of Points

% of points required to consider the model valid (d).

Threshold value

Threshold value (minutes) for determining when a data point fits a model (t).

Linear model

This option should be selected only if the model has to be linear.

! Please, remember that when the shift in the retention time between the peaks in the samples is not constant the model shape is nonlinear, and this parameter should not be selected.

Recommendations for setting optimal parameters

The three first parameters (m/z tolerance, RT tolerance after the correction and RT tolerance) define **2 bi-dimensional windows** with the same "altitude" (m/z tolerance) and different "longitude" (RT tolerances).

The first window (m/z tolerance - RT tolerance after the correction) sets the space where the matching peak should be present, and **the second window** (m/z tolerance - RT tolerance) sets the total space where RANSAC algorithm will be applied.

- So, "**RT tolerance**" should be as big as the maximum deviation in the retention time along all the chromatogram, and "**RT tolerance after the correction**" can be more flexible and depends on the complexity of the data.

If the data contains few peaks and the separation is good, the window can be bigger than "RT tolerance" window. It will improve the recall without including mistakes. This parameter should not change too much the final results.

- RANSAC is a non-deterministic algorithm, and the probability to find a good result increases with the **number of iterations**. If the user sets "0 iterations" into the parameter "RANSAC iterations" the algorithm will automatically set the optimum number of iterations depending on the number of data points.

! In the case that there is a big number of data points it is better to limit this parameter even though the result could be non-optimal. The preview module can help in setting this parameter.

- The parameter "**Minimum number of points**" should be an estimation of the proportion of the data points inside the model. It is important not to get models composed by few data points which do not correspond to the real model. All the models which contain less proportion of data points won't be taken into account by RANSAC algorithm.

- **Threshold** value represents the width of the model and depends on the nature of the data. If this parameter is too big, it can lead to deviation of the model.

The preview module can help to set the optimal value.

- The choice of **model** depends on whether the deviation in the retention time can be considered linear or not in the data.

If the deviation in the retention time is linear, a simple linear regression will be used to fit the model.

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

11.18.4 GC-EI Feature Alignment

Description

≡ Feature list methods → Alignment → GC aligner

This module aligns GC/EI-MS feature lists by grouping deconvoluted features based on their retention times and spectral similarities to ensure accurate alignment of features across different samples.



GC aligner requires mass spectra to be constructed prior to the alignment (using GC-EI spectral deconvolution).

A typical workflow can be constructed with the [mzwizard](#) by selecting GC-EI-MS. See [this page](#) for a more detailed description of the workflow.

Parameters

FEATURE LISTS

Select the feature lists to be aligned.

M/Z TOLERANCE (SCAN-TO-SCAN)

Tolerance for m/z values used for spectral similarity matching.

RETENTION TIME TOLERANCE

Maximum allowed difference between two retention time values for them to be considered for alignment.

WEIGHT FOR RT

Score for perfectly matching RT values. The RT delta is divided by the RT tolerance and scaled 0-1. The weight then multiplies this score so that higher RT delta gives a penalty. The RT weight shifts the focus on RT or spectral similarity.

SIMILARITY

Algorithm to calculate spectral similarity between samples. Pseudo spectra from GC-EI spectral deconvolution will be compared. Usually uses the **Composite cosine identity** which is similar to the algorithm in NIST search for GC-EI data.

FEATURE LIST NAME

Name for the aligned feature list.

ORIGINAL FEATURE LIST

Either keep or remove the original feature lists.



The alignment is performed independently of the representative feature. Only spectral similarity and retention time are taken into account. The final representative feature will be determined as during the alignment.

[Ansgar Korf, Robin Schmid](#)

⌚ July 30, 2025 10:41:01

👤 [Ansgar Korf, Robin Schmid](#)

11.18.5 LC-Image Aligner

Description

≡ Feature list methods → Alignment → LC-Image-Aligner

Aligns LC and imaging measurements based on m/z and mobility. Images are aligned to all LC features that match, only the best match is retained.

Parameters

Feature lists

Select at least two feature lists. The image feature list(s) are aligned to a single (pre-aligned) LC feature list.

m/z tolerance

The file-to-file tolerance for two features.

m/z weight

Maximum score for a perfectly matching m/z. Default value is 1.

Mobility tolerance

Optional parameter

The file-to-file mobility tolerance. If the files don't contain mobility information, this parameter will be ignored. Default value is 0.01.

Mobility weight

Maximum score for a perfectly matching mobility. Default value is 1.

Feature list name

The name of the new feature list. Use {lc} to use the name of the input (LC/DI) feature list.

omokshyna

⌚ November 24, 2022 21:13:07

👤 omokshyna

11.18.6 MRM aligner

Warning

To process multiple reaction monitoring (MRM) data, the MRM service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

The MRM aligner offers feature alignment that is specific to the MRM workflow. This aligner allows the option to only align features that only contain the same MRMs (see [Only allow matches to same MRM](#)).

In principle, the [Join aligner](#) can be used to align MRM files, but will not take the transitions into account. However, the Join aligner can be used to align an already aligned MRM feature list to an HRMS feature list, although this would require really stable chromatography, as scoring by m/z may not produce the desired results.

Parameters

Feature lists

Select the feature lists that shall be aligned. All feature lists must contain only MRM features.

[m/z tolerance \(sample-to-sample\)](#)

The m/z tolerance between two MRM features in different samples. As the m/z values in MRM acquisitions usually resemble the set m/z in the acquisition software, this parameter may be set smaller than for full-scan low resolution data.

[Retention time tolerance](#)

The RT tolerance for the same feature in two different samples/raw files.

[Only allow matches to same MRM](#)

If this parameter is selected, alignment of two rows is only allowed, if:

- a. they have the same number of transitions
- b. their transitions have the same Q1 and Q3 masses
- c. they have the same quantifier ion mass

Feature list name

The name of the created aligned feature list.

Original feature list

Specify if the feature lists that are aligned in this step shall be removed from the project or if they should be retained.

Algorithm

The MRM aligner searches for matches in all feature lists within the given m/z and RT tolerances. All matching features will then be checked for the same MRMs, if the [Only allow matches to same MRM](#) parameter is enabled. In case there are multiple rows, that match the criteria, only the RT will be used to find the best match.

After the alignment, all aligned rows will be set to the same quantifier ion m/z . By default, this is the quantifier of the most intense feature. If the [Only allow matches to same MRM](#) was disabled and the most intense feature has a quantifier transition set, that is not available in all aligned features, an error will be produced and the alignment will fail, because the results would not be valid. To circumvent that, only align features with the same MRM, or manually set a quantifier ion that appears in all samples using the [Set MRM quantifiers](#) module prior to the alignment step.

Robin Schmid, Steffen Heuckeroth

⌚ March 6, 2025 09:17:43

👤 Robin Schmid, Steffen Heuckeroth

11.19 Gap filling

11.19.1 Peak finder

Description

 It is a recommended gap-filling algorithm.

 Feature list methods → Gap filling → Peak finder.

[//] # (TODO ADD FIGURE ALIGNED TABLE)

When a feature cannot be identified/quantified in certain sample will be assigned a zero-intensity value during the feature alignment (see [here](#) for more details). This produces **gaps** in the aligned feature table, commonly referred to as **missing values**.

There is a number of occasions where, due to suboptimal feature detection, a *missing value* is assigned even though the peak is actually present. Some chromatographic features in an aligned feature list may not be detected in every sample for several reasons, such as:

1. Peak intensity below the minimum abundance threshold. The peak will be not detected even though there is a peak, and it can be gap-filled for the alignment process if the peak having the same RT and m/z is detected in other samples. [//] # (TODO Explain minimum abundance threshold)
2. Peak discarded in a previous processing step. Peaks filtered out due to low peak intensity
3. Misalignment due to shifts in m/z, retention time, or ion mobility within feature lists from different samples (or batches). Might originate from inaccurate mass calibration, etc.
4. Inaccurate peak detection and deconvolution of co-eluting compounds. Co-eluting features that are not baseline separated might be resolved in one sample but kept unsplit in another
5. Feature shape constraints in the resolver or later feature filters

All of these reasons can result in undesirable gaps (missing values) in the aligned feature table. Those gaps are not limited to smaller signals but can also affect abundant features.

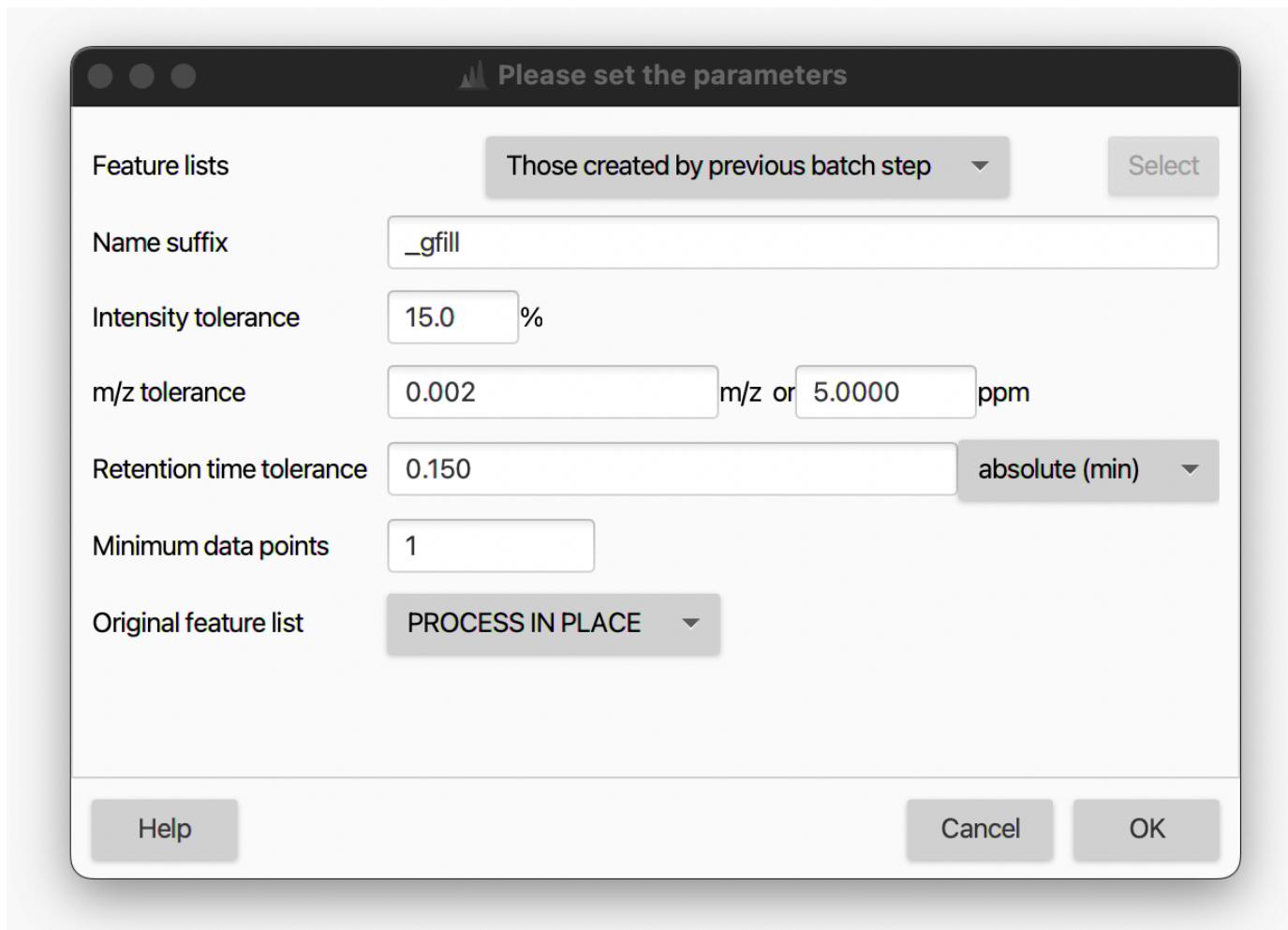
To account for this problem, the user can use the Peak finder module as a secondary, informed feature finding step. The gap-filling module (*i.e. 'Peak finder' algorithm*) aims at reducing false missing values and 'fill the gaps' by going back to the original raw data and re-integrating the peak area where the peak is expected. This approach is sometimes referred to as 'back-filling'.

The algorithm searches for signals within the original centroided mass spectra. It fills the gaps in the feature list according to the user parameters, with the most crucial being **m/z tolerance** and **RT tolerance**. These two tolerances define the window where the algorithm should find the new feature.

Each feature in the aligned table is examined individually within the RT window associated to the examined features. Algorithm searches for appropriate features in that window. If the user-defined requirements are met, the peak is integrated, and the retrieved peak area is used to reconstruct omitted features and fill the gap.

[//] # (TODO ORIGINAL of TEXT above: If requirements are met (e.g. REQUIREMENTS) the peak integrated and the retrieved peak area used to XXX, this filling the gap. and reconstructs omitted chromatographic peaks in "empty" samples by scanning the m/z and retention time region of LC-MS spectra corresponding to the detected peaks in other samples.)

In the feature table, gap-filled features are marked with a grey color as the feature state (see [feature table description](#)). There may still be some gaps in which no evidence for the peak was found, marked by orange.



⚠ DANGER OF DUPLICATED FEATURES. Some features in the dataset can show duplicates, which appear during feature recognition and alignment of samples. The duplicates falsify the exploratory analysis of data and are removed in MZmine.

The gap-filled feature table can be further used in downstream data analysis

Parameters

Name suffix

Suffix to be added to the peak list name.

Intensity tolerance

Maximum allowed deviation from the expected peak shape in chromatographic direction.

m/z tolerance

m/z range which will be applied when searching for the possible feature in the raw data.

The tolerance can be specified as absolute tolerance (in m/z), relative tolerance (in ppm), or both. When both are specified, the tolerance range is calculated using the maximum between the absolute and relative tolerances.

Retention time tolerance

Retention time range when searching for the possible feature in the raw data.

Minimum data points

Feature will be used for gap filling only if it satisfies the set minimum number of data points.

💡 Usually a lower number of data points is used compared to the primary feature finding workflow with the resolvers.

Original feature list

User can either keep, remove, or process in place of the original feature list. The latter two increase memory efficiency and throughput while users might want to keep the original feature list as a reference.

[Robin Schmid](#), [lalalana5](#), [omokshyna](#), [tdamiani](#)

⌚ November 24, 2022 21:13:07

👤 Robin Schmid, lalalana5, omokshyna, tdamiani

11.19.2 Same m/z and RT range gap filler

Description

≡ Feature list methods → Gap filling → Same m/z and RT range gap filler.

This method fills in gaps in each peak list row by using the same m/z and retention time range as other peaks in the row. The m/z and retention time defines where the new peaks will be sought based on the ranges of the rest of the peaks in the same row. The minimum value of these ranges is the minimum value in the range of all the peaks in the row and the same happens with the maximum value. User-specified tolerance is added to the m/z range. The new peak is constructed using the highest data point of each scan within the determined m/z and retention time ranges.

Parameters

Name suffix

Suffix to be added to the peak list name.

m/z tolerance

Tolerance, which is added to the m/z range of other peaks in the peak list row.

Original feature list

The user can either select to keep or remove the original feature list.

[lalalana5, omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [lalalana5, omokshyna](#)

11.20 Normalization

11.20.1 Retention time correction

Description

≡ Feature list methods → Normalization → Retention time correction

The retention time correction module attempts to reduce the deviation of retention times between feature lists, by searching for common features in these feature lists and using them as RT correction standards.

This process involves identifying standard features, calculating their average RTs, and adjusting the RTs of other features based on these standards. The RT correction value will be added to each feature.

Identifying RT correction standards

RT correction standards are features that:

- Meet a minimum intensity threshold.
- Are present in all feature lists.
- Fall within specified m/z and RT tolerances.

Averaging RTs of correction standards

RTs of identified RT correction standards are averaged. The averaged RT is set as RT for the calibration standard features across all samples.

Correcting retention times

The correction of RTs is performed based on the relationship between the original RTs of features and the RTs of the calibration standards. The following cases describe how the corrected RTs are calculated:

CASE 1: FEATURE IS A RT CORRECTION STANDARD

The RT for this feature is set to its averaged RT calculated from all samples.

CASE 2: FEATURE BETWEEN TWO RT CORRECTION STANDARDS

For features that are not calibration standards and fall between the RTs of two identified calibration standards:

The corrected RT is calculated by linear interpolation between the closest previous and next standards.

For a feature (F) with an original retention time (RT_F) , situated between two standards (S_{prev}) and (S_{next}) with original retention times (RT_{prev}) and (RT_{next}) , and corrected retention times $(RT_{\text{prev, cor}})$ and $(RT_{\text{next, cor}})$:

1. **Calculate the weight:** $[\text{weight} = \frac{RT_F - RT_{\text{prev}}}{RT_{\text{next}} - RT_{\text{prev}}}]$
2. **Calculate the corrected RT:** $[RT_F = RT_{\text{prev, cor}} + \text{weight} \times (RT_{\text{next, cor}} - RT_{\text{prev, cor}})]$

Explanation

- **Weight:** Represents the relative position of feature (F) between the two standards (S_{prev}) and (S_{next}) .
- **Corrected RT:** Computed by linearly interpolating between the corrected RTs of the two surrounding RT correction standards, weighted by the relative position of (F) .

CASE 3: ONLY ONE STANDARD

If only one RT correction standard is identified:

The corrected RT is adjusted by adding the difference between the original RT of the standard and its averaged RT to the original RT of the feature.

! The method requires at least two resolved (for example, Local minimum resolver) feature lists. This step needs to be performed before alignment.

Parameters**Name suffix**

Suffix to be added to a processed feature list name

m/z tolerance

Maximum allowed m/z difference for two values to be considered the same

Retention time tolerance

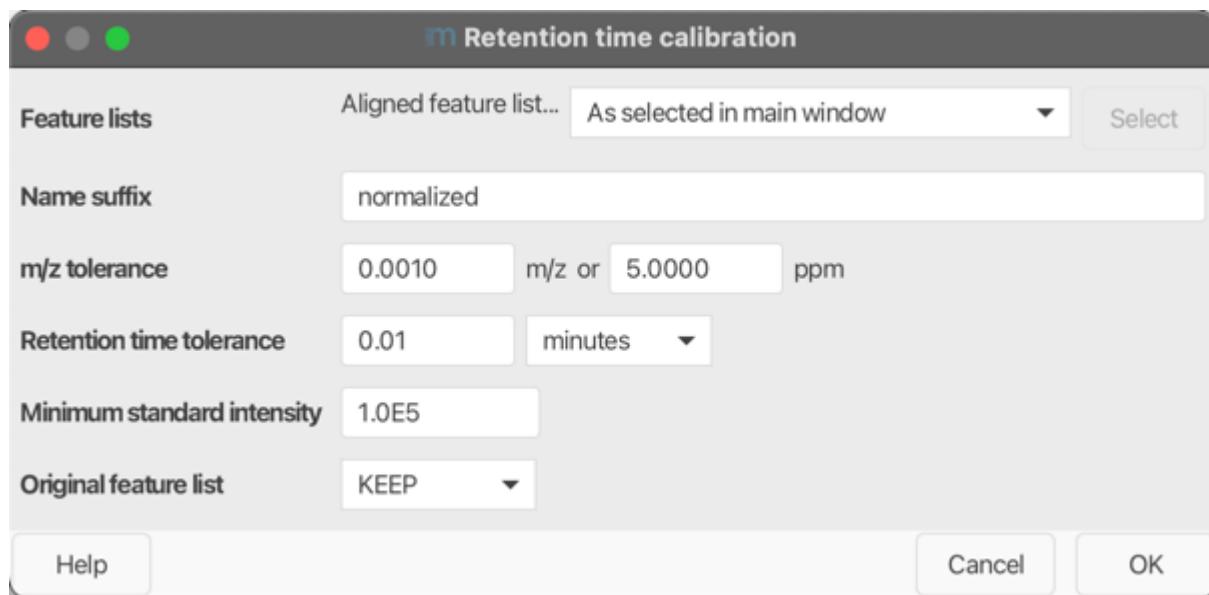
Maximum allowed difference between two retention time values

Minimum standard intensity

Minimum height of a feature to be selected as RT correction standard

Original feature list

If REMOVE option is selected, the original feature list is removed, allowing to save memory.



Ansgar Korf, omokshyna

⌚ July 9, 2024 08:06:21

👤 Ansgar Korf, omokshyna

11.20.2 Retention time correction of scans



This module is currently in beta phase.

Corrects retention time drifts by searching for standards across selected samples and corrects retention time drifts between samples by changing the scan retention times.

IMPLICATIONS AND RESTRICTIONS

Correcting retention times of scans instead of features has certain implications and up- and downsides.

Pros:

- All visualizations in mzmine will use the calibrated retention times to visualize.
 - this means that e.g. the [Raw data overview](#) or the [feature table](#) will visualize corrected chromatograms
 - Gap filling will use the corrected retention times, which may lead to less misalignments during gap filling of complex samples

Cons:

- Corrected scan retention times must still be in strongly monotonous order (cannot swap scans)
 - This means that large RT shifts cannot be mitigated
 - Irregular shifts cannot be corrected: If feature A at RT 5.4 is shifted to higher RT and feature B at RT 5.4 is shifted to lower RT the effects may cancel each other out
 - Such shifts may happen if the stationary phase is damaged/overloaded or the pH of the eluent changes and thus affects compounds with different pKs differently
 - All files must start at 0.0 min, no negative RTs possible.



The retention time correction may be applied multiple times to correct larger shifts.

EXEMPLARY RESULTS

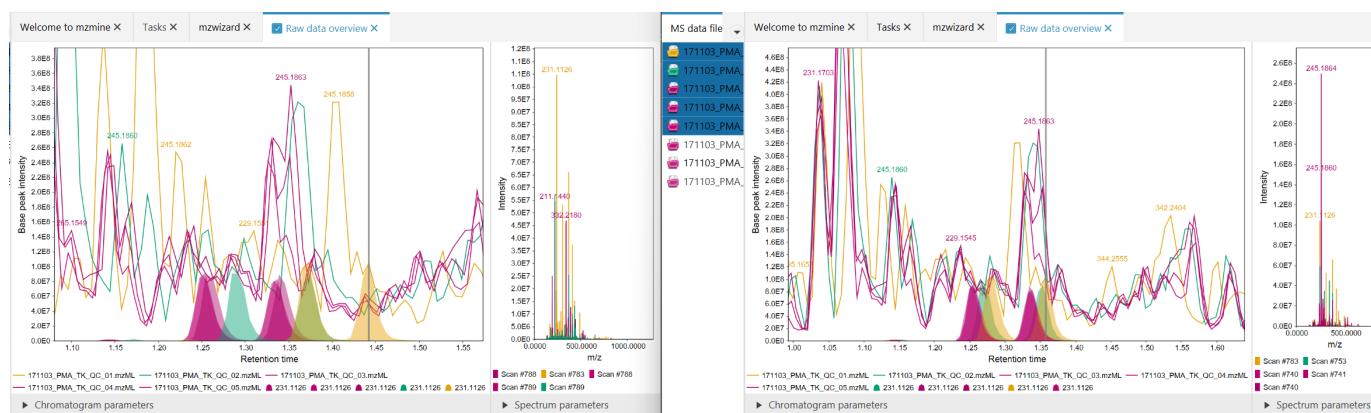


Figure 1: Before (left), After (right). Significantly better alignment was achieved for the orange data file.

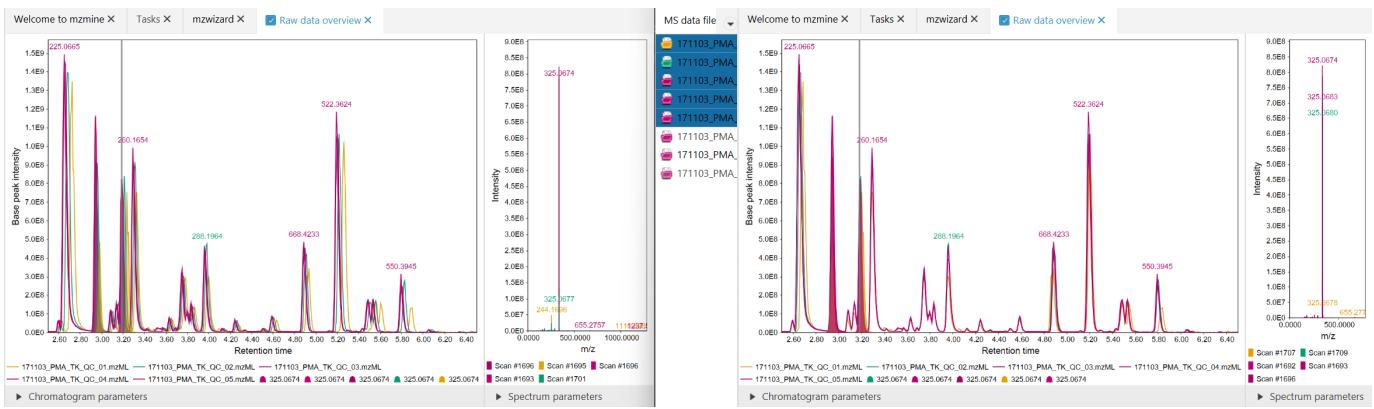


Figure 2: Before (left), After (right). Significantly better alignment was achieved for the orange data file.

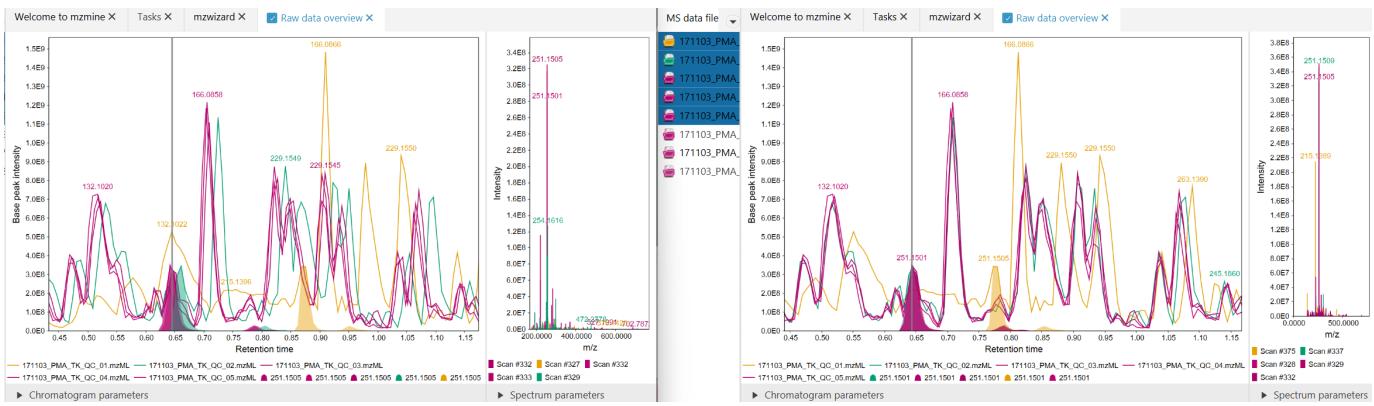


Figure 3: Before (left), After (right). Large retention time shift of the orange signal. After correction, the signals move closer together, but the large shift cannot be corrected completely due to the mentioned restrictions.

STANDARD SELECTION

The calibration module will look for features that appear in all feature lists that are selected as [reference samples](#). Ambiguity will be avoided during standard selection by also examining the same [mz tolerance](#) but three times the [RT tolerance](#) window around a potential standard. If the extended window contains more matching features in [any sample](#), a high MS2 similarity is required or the feature will not be used as a standard.

Parameters

Feature lists

The feature lists whose data files shall be corrected. Each feature list may only contain a single raw data file. Visualizers will need to be reopened to show the result.

It is recommended to use [resolved feature lists](#) for this step. Since the scans are corrected, all other (e.g. prior) feature lists using the same data file will be affected as well, once a new processing step is applied (RTs are not recalculated automatically for old feature lists).

Reference samples

Select the sample types (from the metadata table) that shall be used as reference to search standards. The created calibration will be interpolated onto the other data files by weighting the nearest two standard files by their acquisition date.

mz tolerance

Maximum mz deviation for potential standards.

Retention time tolerance

Maximum RT tolerance for potential standards.

Minimum standard intensity

The minimum required intensity of a feature to be considered a standard for RT calibration.

RT standard calculation

Select if the corrected standard value to which the samples shall be calibrated to shall be chosen as average or median.

Correction method

Select the method which shall be used for correction.

Multi linear correction

- Calculate difference of **standard** and specific RT in a sample.
- Smooth difference using a moving average of num_standards * bandwidth points.
- Apply asymmetric least squares regression (ALS) on the smoothed differences
- Apply loess interpolation on current retention times (domain values) vs the ALS result (range values)
- calculate potential new differences (= correction values) from the loess fit
- ensure monotonicity of the new fit
- create a final loess fit of the corrected differences
- apply correction to original scan RTs

Interpolation bandwith

Define a percentage of standards for moving average calculation.

⌚ October 20, 2025 11:21:41

👤 Steffen Heuckeroth

11.20.3 Linear normalizer

Description

≡ Feature list methods → Normalization → Linear normalizer

Linear normalizer divides the height (or area) of each feature in the feature list by a normalization factor, chosen according to the "Normalization type" parameter.

Each column of the feature list is normalized separately. In other words, normalization factor is determined independently for each raw data file.

NORMALIZATION FACTORS

Different normalization factors can be applied:

1. Average intensity

Average height (or area) of all peaks in the column is calculated and used as the normalization factor

2. Average squared intensity

Same as Average intensity, but values are squared before calculating the average

3. Maximum peak intensity

Maximum height (or area) in the peak list column is used as the normalization factor

4. Total raw signal

Sum of the height (or area) of all peaks in the peak list column is used as the normalization factor

Parameters

Name suffix

Suffix to be added to a processed feature list name

Normalization type

Selection of the normalization factor. Available options:

- Average intensity
- Average squared intensity
- Maximum peak intensity
- Total raw signal

Feature measurement type

Selection of either feature height or feature area, which will be used to calculate the normalization factors

Original feature list

If REMOVE option is selected, the original feature list is removed, allowing to save memory.

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 [omokshyna](#)

11.20.4 Standard compound normalizer

Description

≡ Feature list methods → Normalization → Standard compound normalizer

The purpose of this module is to reduce the deviation between samples caused by different detection efficiency.

Internal standard peaks must be present in the detected samples. User can select one or multiple internal standard peaks, which must be present in all raw data files. Then peak height (or area) of each peak is normalized by either the **nearest standard** or a **weighted contribution** of all standards.

In case a weighted contribution is used, the contributions of all standards are weighted by distance. The distance of the standard peak to the peak being normalized is calculated as

```
\[distance = MZvsRT_{Balance} * MZ_{difference} + RT_{difference}\]
```

where `\(MZvsRT_{Balance}\)` is a multiplier of m/z difference set by **m/z vs RT balance** parameter

 Feature list must be aligned prior to normalization.

Parameters

Name suffix

Suffix to be added to a processed feature list name

Normalization type

Normalize intensities using either only one (nearest) standard or using a weighted contribution of all selected standards, weighted by distance.

Feature measurement type

Selection of either feature height or feature area, which will be used to calculate the normalization factors

m/z vs RT balance

Used in distance measuring as a multiplier of m/z difference.

Standard compounds

List of features for choosing the normalization standards

Original peak list

If selected, the original peak list is automatically removed

omokshyna

 November 24, 2022 21:13:07

 **omokshyna**

11.20.5 Retention index calculation

Description

≡ Feature list methods → Normalization → Retention index calculation

Retention index calculation assigns a retention index to each feature in the feature list. These are calculated by linear interpolation through the retention times of the straight-chain alkanes, which are listed in CSV files.

Each raw data file has its features' retention indices calculated separately. In other words, the alkane list is selected independently for each raw data file.

To select the appropriate alkane list, a date is extracted from the name of each CSV file, and each raw data file is assigned the file with the most recent date, up to the raw data file's run date.

Parameters

Feature lists

Selects the feature lists for which retention indices will be calculated

Alkane scale files

A list of CSV files, each with a list of straight-chain alkanes and their respective retention times. For example:

Carbon #	RT
5	4.374
6	5.423
7	7.934
8	11.99
9	16.793
10	21.615
11	26.181

Each file's name should contain the date that the measurements were taken. Accepted formats:

- yyyy-MM-dd like 2025-12-25
- yyyyMMdd like 20251225
- with time: yyyy-MM-ddTHH-mm-ss like 2025-12-25T20-30-15
- with time: yyyy-MM-dd_HH-mm-ss like 2025-12-25_20-30-15

Extrapolation

This will enable calculation of retention indices for features with retention times not between those of the first and last straight-chain alkanes, performing linear extrapolation past the endpoints of the scale. Not recommended.

Add row summaries

Enabling this will cause each feature to show the CSV that was used to calculate its retention index.

Additionally, each feature list row will display the lowest and highest retention index among its features, alongside the difference between those values.

Original feature list

If REMOVE option is selected, the original feature list is removed, allowing to save memory.

Name suffix

Suffix to be added to each processed feature list's name

[Robin Schmid, William Chu](#)

 April 24, 2025 11:25:39

 [Robin Schmid, William Chu](#)

11.21 Precursor mass search

11.21.1 Local compound database search

Description

≡ Feature list methods → Annotation → Search precursor mass → Local compound database (CSV) search

This method assigns identity to features according to their m/z. Additionally, retention time, mobility, and CCS values can be provided to restrict the annotation. Otherwise, only the m/z is used for the matching, which can be used for the spectral library building. For compound annotation in complex samples, an additional identifier should be available.

 If you want to filter your feature list for a list of target compounds afterwards, you can use the [Feature list rows filter](#) with the only identified option.

The user has to provide a database of m/z values (or neutral masses, formulae, smiles strings to calculate m/zs from) and retention times in ***.csv format** (see below).

DATABASE FILE

Database file has to be provided in **.csv*** or **.tsv format*** (Comma-Separated or tab). Such files can be exported from a spreadsheet software such as MS Excel, or edited manually using a text editor.

The following examples shows the structure of the database file:

```
ID,monoisotopic_mass,rt,compound_name,formula
1,175.121,24.5,Arginine,C6H14N4O2
2,133.063,11.9,Asparagine,C4H8N2O3
3,134.047,11.7,Aspartate,C4H7NO4
```

 If the m/z value or Retention time value in the CSV file is 0, then the value is considered as a wild card. E.g, the following item will match all peaks of 174.121 m/z without considering the retention time:

```
1,175.121,0,Arginine,C6H14N4O2
```

The available fields in a library file include:

Field name	Field description
monoisotopic_mass	Neutral mass
mz	Precursor m/z
rt	Retention time
formula	Formula
smiles	SMILES
inchi	InChI
compound_name	Compound name
CCS	CCS/ Å ²
mobility	Ion mobility
comment	Text comment
adduct	Information on adduct
pubchem_cid	Compound ID in PubChem database

Parameters

Database file

Path and Name of the file that contains information for peak identification.

Field separator

Character(s) used to separate fields in the database file. - .csv = , - .tsv = \t

Columns

Columns that should be imported from the library file are selected. The choice of columns depends on the availability of retention time, mobility data, information about adducts, and the presence of the PubChemID in your database file. If the exact mass should be calculated, the neutral mass (or structure information) needs to be provided and enabled together with the adduct settings. When only a structure is provided, MZmine calculates the monoisotopic mass based on the given information. If the table contains already the adduct m/z value, the precursor m/z can be selected. This might be useful, when the ionization differs within one dataset. The column names can be adapted to your table by double click and pressing enter after changing.

m/z tolerance

Maximum allowed m/z difference to set an annotation to a peak (max. difference between exact and accurate mass).

Retention time tolerance

Maximum allowed retention time difference to set an identification to a peak.

Mobility time tolerance

Maximum allowed tolerance to set an identification to a peak.

CCS tolerance (%)

Maximum allowed difference (in percents) to set an identification to a peak.

Use isotope matcher

This option matches the predicted isotope pattern against the detected ones. It is only possible to use if the isotope finder was run before.

Use adducts

This can be used, if the precursor should be calculated. If chosen, m/z values for all set adducts will be calculated and matched against the feature list. For this option, you have to provide a neutral mass, a formula or a SMILES code to calculate the m/z from. Be aware, if multiple adducts are set, more spurious hits will be created. This is also important for the spectral library creation, when one sample contains multiple compounds. Therefore the modifications can be limit to specific adduct combinations. This can be done by adding and combining them. For example M-H₂O+H, by adding M-H₂O and combining it with the M+H.

Filter filename header

This option is important for the library building workflow. This column contains a unique sample id to link the compound to the corresponding acquisition file. The unique sample id can be either part of the filename or the complete filename. If data are acquired multiple times with differnet method, a part of the filename enables the option of one metadata file or one column for the identification. The unique sampel id can be a combination of library_id, the plate or rack location together with the well or rack position. It is important to notice, that the unique sample id should not end on a number, otherwise libid_Plate1_A2 is also contained in libid_Plate1_A20 ect.

Append comment fields

Multiple fields, that are appended to the comment. They are separated by comma. For example: Pathway, Synonyms,...

Steffen Heuckeroth, corinnabrunungs, omokshyna

⌚ September 18, 2023 17:03:03

👤 Steffen Heuckeroth, corinnabrungs, omokshyna

11.21.2 Precursor search in local spectral MS/MS library

Description

≡ Feature list methods → Annotation → Search precursor mass → Precursor search in spectral libraries

This module uses a **local spectral MS/MS library** to search for putative precursor ions in a feature list.

Supported formats:

- MoNA *.json,
- NIST *.msp,
- GNPS *.json (internal library submission format),
- and JCAMP-DX *.jdx.

Parameters

Spectral libraries file (MS/MS)

Name of the library file of the supported format.

Precursor m/z tolerance

Matches the average row m/z against the precursor m/z of the spectral library entry

Retention time tolerance

Optional parameter, should only be used if the DB entry has a retention time

omokshyna

⌚ November 24, 2022 21:13:07

👤 omokshyna

11.22 Spectra search

11.22.1 Spectral library search

Description

≡ Feature list methods → Annotation → Search spectra → Spectral library search or ≡ with a right click on one or multiple selected feature rows Search → Spectral library search

The spectral library search module can be performed on feature lists, individual features (contained in feature list rows), or single scans.

Depending on the **MS level** (MS1 or MS2), all corresponding query scans (e.g., extracted from the rows) will be matched against selected spectral libraries that were previously imported.

 Preferred ways to import libraries are in this order:

- together with spectral data files in the advanced data import:

Raw data methods → Raw data import → MS data (advanced)

- drag and drop into the main window

- with a dedicated import module:

Raw data methods → Raw data import → Spectral library import or from **Feature list methods → Annotation → Search spectra → Import spectral libraries**.

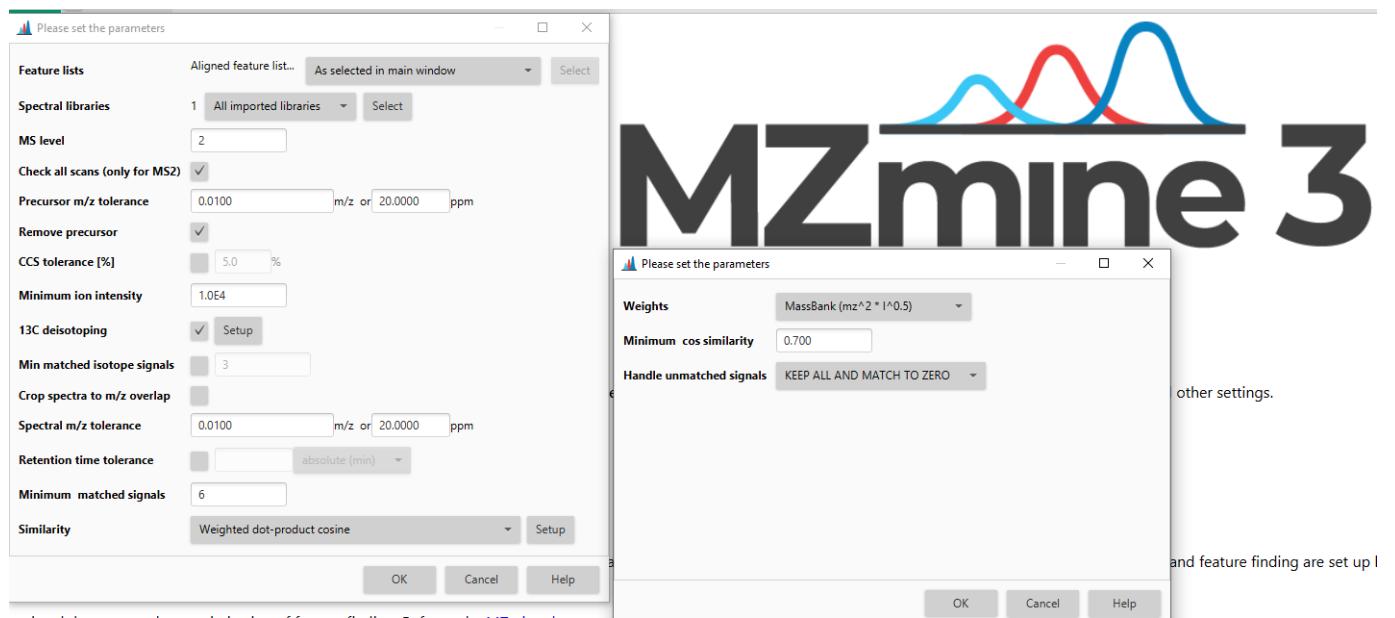
DOWNLOADS FOR OPEN SPECTRAL LIBRARIES

- MassBank of North America: [download](#) (download .json)
- MassBankEU: [download](#) (download *_NIST.msp)
- GNPS (ALL_GNPS_NO_PROPOGATED): [download](#) (download .json)
- MSnLib: [download](#) (download .mgf)
- GC/EI-HRMS for lipids: [download](#) (download .json)

SUPPORTED LIBRARY FORMATS

- *.json: MassBank of North America (MoNA)
- *.json: The Global Natural Product Social Molecular Networking (GNPS) (format from the spectral DB submission module)
- *.mgf: GNPS
- *.msp: MoNA, MassBankEU (NIST compatible), National Institute of Standards and Technology (NIST)
- *.jdx: JCAMP-DX

Parameters



Spectral libraries

The spectral libraries of interest need to be imported before applying spectral library search. Either uses all imported spectral libraries or only the selected libraries.

Merge & select fragment scans

This parameter controls how fragment spectra are filtered, merged, and selected for downstream analysis (see [detailed description](#)). Briefly, either choose preset based spectral merging, input scans without merging, or an advanced setup for more options.

For spectral library matching mzmine recommends using the representative scans = one for each fragmentation energy and one merged across all.

MS level filter

Select MS2 for fragment spectra with precursor m/z.

Select MS1 for GC-EI-MS spectra.

! In case of issues with the scan selection, check how the actual scan numbers are reported in the data files and in mzmine's raw data overview.

Precursor m/z tolerance

This option is only used for MS level > 1. Here, the library entries are filtered by their precursor m/z reducing the number of spectral-pairs to match.

The absolute (in m/z) and relative (in ppm) m/z tolerance can be set. The **maximum tolerance** for each precursor is applied.

Considering that the precursor isolation window is often far greater than the resolution or accuracy of the MS scan, this parameter is often set to higher m/z tolerances.

Another aspect is the used library, which might contain uncalibrated reference spectra from lower resolution instruments.

Spectral m/z tolerance

This m/z tolerance is used to pair signals in the query and library scans. It can be set in absolute (in m/z) and relative (in ppm) m/z tolerance, whereas the maximum tolerance for each m/z value is applied. It must be kept in mind, which mass resolutions are achieved within the experimental spectra and within the spectral library.

Remove precursor

 Can be selected only for MS level > 1.

Depending on the fragmentation method, e.g., collision induced dissociation (CID) or higher-energy collisional dissociation (HCD), the precursor can be detected with different intensities resulting in varying cosine similarities during the library matching.

Therefore, this option enables the removal of the precursor signal (\pm 4 Da) prior to the matching.

Min matched signals

The query mass list and spectral library entry must contain at least this number of matched (paired) m/z values (+- m/z tolerance).

Common parameters include 4 signals for smaller molecules and 6 for more confident matches.

 This parameter must be set carefully to not exclude compounds that show less fragmentation, when using a higher number of matched signals.

 Choosing a lower number of matched signals can result in spurious library hits.

Similarity

Several algorithms can be applied to calculate the similarity of the query and library scans and to filter the resulting library matches. The available algorithms are:

- Weighted cosine similarity,
- Composite cosine identity (e.g., GC-EI-MS; similar to NIST search).

More details are available [here](#).

 The **weighted cosine similarity** is used for comparing MS^2 data, whereas the **composite cosine identity (e.g., GC-EI-MS; similar to NIST search)** considers the relative intensity of neighboring signals and is, therefore, applied to MS^1 spectra from GC-EI-MS.

Retention time tolerance

These option can be used to include these values as further spectral matching identifiers. The maximum allowed retention time difference when comparing the query and library scan. It can be set in absolute (min or sec) or relative (%) values.

 This option is intended for in-house libraries or standardized libraries that follow the same acquisition protocol with the same set-up, e.g., column, instrument, and method).

 However, this method will boost false matches and needs strict manual interpretation.

CCS tolerance

The **collision cross-section (CCS)** tolerance can be used in a similar way as the retention time tolerance.

Accordingly, the CCS value of a query will be compared with the library entries and the maximum tolerance can be set in %.

 If the query or library entry was analyzed without ion mobility (no CCS values), no spectrum will be matched.

Retention index tolerance

This option can be used to exclude library entries that are not sufficiently close in retention index.

Skip library entries without RIs

This option can be used to exclude library entries that do not have a defined retention index for the selected column type. Otherwise, those library entries can be used to match any row.

[13C deisotoping](#)

This option allows to remove potential ^{13}C isotope signals in the experimental spectrum before the library search. It can be enabled when co-isolation of precursor's ^{13}C isotopes is expected during MS2 data acquisition (e.g., isolation window > 1 Da). Following parameters can be set:

- **m/z tolerance:** Maximum allowed difference between the measured and predicted isotope m/z values. The absolute (in m/z) and relative (in ppm) m/z tolerance can be set, whereas the maximum tolerance for each m/z value is applied.
- **Monotonic shape:** If enabled, the monotonically decreasing height of isotope pattern is required.
- **Maximum charge:** The maximum charge that will be considered for detecting the isotope pattern. For singly charged ions, the ^{13}C isotope will be expected +1 whereas for doubly charged ions it will be +0.5 (+1 m/z divided by the charge 2).

[Min matched isotope signals](#)

This option can be used to set a minimum number of matched 13 isotopes signals. It is only useful if the query AND library entries contain isotope patterns (e.g., in MS¹ or with wider precursor isolation windows).

The minimum number of matched signals of ^{13}C isotopes.

 It cannot be applied when ^{13}C deisotoping is enabled.

[Crop spectra to m/z overlap](#)

This option can be used to consider only the mass range where both the experimental and library spectra exhibit m/z signals. All other signals are ignored during the matching. For example, if query and library scans were acquired with different methods, e.g., mass range, fragmentation energy or mode, it can be helpful to crop the spectra to their overlapping m/z range (+ m/z tolerance). This is done by using the maximum m/z range where both spectra contain signals. However, this tends to return more false positive matches. Therefore, we recommend extra curation of the matching results when this option is enabled.

[Ansgar Korf](#), [Robin Schmid](#), [Steffen Heuckeroth](#), [William Chu](#), [corinnabrungs](#), [corinnabrungs](#), [omokshyna](#)

 June 3, 2025 11:54:41

 [Ansgar Korf](#), [Robin Schmid](#), [Steffen Heuckeroth](#), [William Chu](#), [corinnabrungs](#), [corinnabrungs](#), [omokshyna](#)

11.22.2 NIST MS search

Description

≡ Feature list methods → Annotation → Search spectra → NIST MS Search

or, for an individual row in a feature table

≡ highlight the row, right-click on the selection and choose ** Search → NIST MS Search** from the pop-up menu.

This module allows searching spectra against spectral libraries using the **NIST MS Search program**, which accepts spectra as input to its searches.

The MS level may be specified to limit a search to MS/MS fragment spectra, clustered spectra produced from any Spectral Deconvolution module, or MS1 precursors ions. The spectra will be searched using the default library search parameters in the NIST MS Search program.

To adjust these parameters: open the NIST MS Search software and adjust the library search parameters: **Options or Icon → Library Search Options → Search: enable automation → Libraries: choose the preferred libraries**. More information about the libraries and their abbreviations: <https://www.nist.gov/srd/nist-standard-reference-database-1a>

Save the configuration to a *.ini file.

💡 Automation must be enabled in the library search options to enable automatic searching. Be sure all appropriate libraries are included before starting a search.

Repeated MS/MS spectra may be merged between multiple data files using the Merge MS/MS (experimental) module. The input Mass list filters the fragment ions by intensity, and repeatable signals are assessed via cosine dot product.

REQUIREMENTS

This module relies on the installed NIST MS Search software, which is currently **only available for Microsoft Windows**.

Parameters

NIST MS Search directory

Full path to the directory containing the NIST MS Search executable (**nistms\$.exe**).

MS level

MS spectra level for searching.

Use MS level = 1 to search for MS1 spectra or ADAP-GC clustered spectra produced from Spectral Deconvolution modules.

Min cosine similarity

The minimum cosine similarity score (dot product) for identification.

Merge MS/MS (experimental)

Optional parameter.

Merge multiple high-quality MS/MS spectra into consensus feature instead of using the most intense one.

Integer m/z

Optional parameter.

Merging mode for fractional m/z to unit mass. Converts accurate mass m/z measurements to low-resolution integer values.

Available options are Sum or Maximum.

Spectrum Import

Options for import, can be by either **Overwrite** (overwriting previous spectra) or **Append** (appending new ones).

corinnabrungs, omokshyna

⌚ January 28, 2025 10:54:27

👤 corinnabrungs, omokshyna

11.22.3 Chemical formula prediction

Description

≡ Feature list methods → Annotation → Search spectra → Chemical formula prediction

This module attempts to calculate all possible molecular formulas for every peak in the peak list, using given elemental and heuristic constraints.

For a detailed description of the functionality and the embedded algorithms, please see the publication [1].

References

1. Pluskal T. et al, Highly Accurate Chemical Formula Prediction Tool Utilizing High-Resolution Mass Spectra, MS/MS Fragmentation, Heuristic Rules, and Isotope Pattern Matching. *Anal Chem* (2012), 84(10):4396-403. DOI: [10.1021/ac3000418](https://doi.org/10.1021/ac3000418).
2. Kind and Fiehn, Seven Golden Rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. *BMC Bioinformatics* (2007). DOI: [10.1186/1471-2105-8-105](https://doi.org/10.1186/1471-2105-8-105).

Parameters

Charge

The neutral mass is calculated from the peak m/z value, its charge and type of ionization adduct.

Ionization type

The neutral mass is calculated from the peak m/z value, its charge and type of ionization adduct.

Sorting

Optional parameter

If this option is chosen, the user-defined sorting is applied to all selected feature lists.



Max best formulas per feature

This parameter defines the maximum number of formulas to be added to a single feature.

m/z tolerance

Tolerance of the neutral mass for searching the formula.

Elements

Elements allowed in the formula and their minimum and maximum counts.

Element count heuristics*Optional parameter*

Selection of heuristic restrictions on element counts. These heuristics check the formula's element counts as defined in [2].

Available parameters

These include:

- **H/C ratio**

Ratio of hydrogen to carbon atoms, with available value range $0.1 \leq \text{H/C ratio} \leq 6$.

- **NOPS/C ratios**

Ratio of nitrogen to carbon atoms, with available value $\text{N/C ratio} \leq 4$, Ratio of oxygen to carbon atoms $\text{O/C ratio} \leq 3$, Ratio of phosphorus to carbon atoms $\text{P/C ratio} \leq 2$, Ratio of sulphur to carbon atoms $\text{S/C ratio} \leq 3$

- **Multiple element counts**

If this parameter is chosen, then several following rules apply:

- if number of N/O/P/S atoms all > 1 then $\text{N} < 10$, $\text{O} < 20$, $\text{P} < 4$, $\text{S} < 3$
- if number of N/O/P atoms all > 3 then $\text{N} < 11$, $\text{O} < 22$, $\text{P} < 6$
- if number of O/P/S atoms all > 1 then $\text{O} < 14$, $\text{P} < 3$, $\text{S} < 3$
- if number of P/S/N atoms all > 1 then $\text{P} < 3$, $\text{S} < 3$, $\text{N} < 4$
- if number of N/O/S atoms all > 6 then $\text{N} < 19$, $\text{O} < 14$, $\text{S} < 8$

RDBE restrictions*Optional parameter*

Selection of restrictions on RDBE (rings double bonds) values. The **Ring Double Bond Equivalents (RDBE)** value estimates the number of rings and unsaturated bonds in a molecule. It can be calculated from a chemical formula using the following general equation:

$$\text{RDBE} = 1 + \frac{1}{2} (\sum_i n_i (\nu_i - 2))$$

where n_i is the number of atoms and ν_i the formal valence of the element i.

Theoretically, each ring or a double bond increases the RDBE value by 1, while each triple bond increases the value by 2.

⚠ This equation can only be used for formulas composed of elements with a well-defined formal valence.

A number of exceptions to the RDBE rule are known, however, the RDBE value still provides a useful indicator regarding the validity of a molecular formula.

Available parameters

These include:

- **RDBE range**

Range of allowed RDBE (Range or Double Bonds Equivalents) value.

[2] recommended the RDBE upper limit of 40 for common chemical compounds. The authors also stated that RDBE should not be negative, although certain exceptions may occur when formal valence states are exceeded.

- **RDBE must be an integer**

Only integer values are allowed for RDBE. This condition is a natural implication of the principle of valence balance, which states that the number of atoms with odd valence must be even. Such assumption is valid for all neutral, non-radical molecules.

Isotope pattern score*Optional parameter*

If selected, only results that fit the required isotope pattern similarity score will be returned.

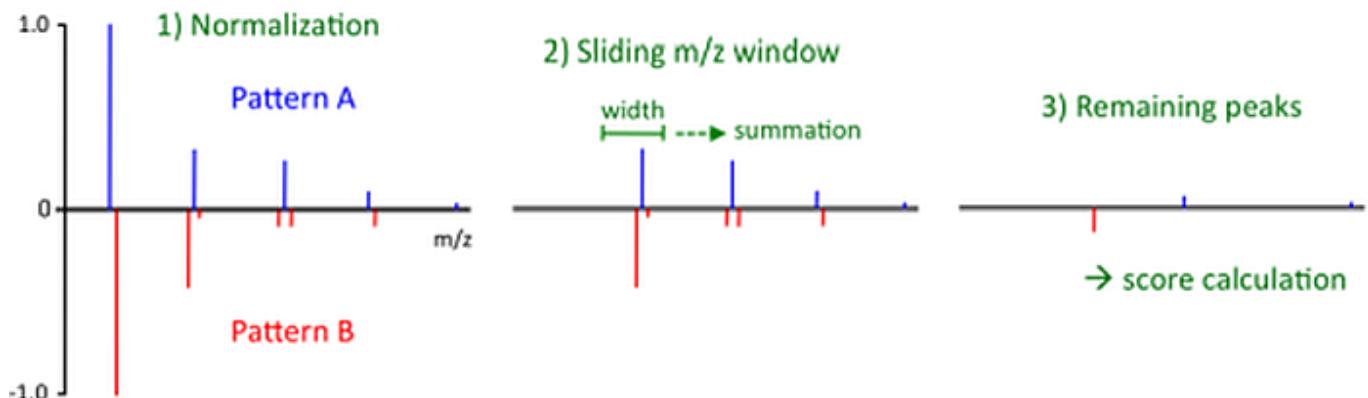
Several modules in MZmine offer the option to compare the isotope patterns of peaks and assign a score (percentage) of similarity to them.

Until MZmine version 2.2, the CDK (Chemistry Development Kit) library was used to perform this operation. An improved algorithm, introduced in MZmine 3, is described below. The **similarity of two isotope patterns** is determined as follows:

- Both isotope patterns are normalized (such that highest isotope in each pattern has the intensity of 1.0) and merged into a single spectrum. In this single spectrum all isotopes from the first pattern have a positive intensity, while the isotopes of the second pattern have negative intensity.
- A sliding window of user-defined width ("Isotope m/z tolerance" parameter) is moved over the whole m/z range, from the lowest m/z to the highest. Each pair of isotope peaks fitting within the window is added together, forming a single peak with the m/z value of the higher m/z of the pair.
- The final similarity score is calculated from the remaining peaks as

$\text{score}_{\{\text{isotopes}\}} = \prod_{\{i\}} (1 - |I_i|)$

where $|I_i|$ is the intensity of remaining peak $|i|$.



A trivial observation is that for two identical isotope patterns the similarity score will be 100%, while for two completely different patterns 0% score is returned.

Only a single parameter is required for the evaluation of the algorithm, defining the width of the sliding window.

It should be noted, though, that the optimal value of width of the sliding window parameter might be different from the commonly perceived "mass accuracy" of the instrument as mass resolving power and preprocessing of the data must be considered. For example, even if the mass accuracy of the major isotopes may be less than 0.001 m/z, the mass difference between minor isotopes may be significantly higher.

Additional parameters

These include:

- **Isotope m/z tolerance**

m/z tolerance which defines what isotopes would be considered same when comparing two isotopic patterns.

This tolerance needs to be higher than general m/z precision of the data, because some small isotopes may overlap with the sides of bigger isotopic peaks.

- **Minimum absolute intensity**

Minimum absolute intensity of the isotopes to be compared. Isotopes below this intensity will be ignored.

- **Minimum score**

If the score between isotope pattern is lower, the match will be discarded.

MS/MS filter*Optional parameter*

Use MS/MS pattern for candidate formula evaluation.

In tandem mass spectrometry (MS/MS), during fragmentation, part of the original ion is detached, and the mass of the detached part is called the **neutral loss**.

The neutral loss represents a fragment of the original molecule, so the chemical formula of such fragment must be a subset of the chemical formula of the precursor.

When searching for the ion's chemical formula, each candidate formula may therefore be evaluated using the ion's MS/MS spectrum using the **algorithm** described below.

1. Mass list must be provided for the MS/MS spectrum of the ion of interest (see the [Mass detection modules](#)). It is assumed that all items in the mass list represent true fragment ions and noise has been removed.
2. If the mass list contains any isotopes, remove them from the list. Isotopes are defined as ions with mass approximately 1 Da higher than another ion on the list, which has higher intensity.
3. Calculate neutral losses for all the ions on the list by subtracting the fragment ion mass from the precursor mass.
4. Try to generate a chemical formula for each neutral loss using the elements and maximum counts of formula F, within the user-defined mass tolerance. Small neutral losses (less than 5 Da) are ignored.
5. If at least one formula can be found, the neutral loss is considered as interpreted.
6. The evaluation score is calculated for each candidate formula F as described below:

$\text{[score}_{\text{MS/MS}}\text{]=}\frac{\text{n}_{\text{found}}}{\text{n}_{\text{total}}}$

where $\text{(n}_{\text{found}}\text{)}$ is the number of ions for which the neutral loss could be interpreted, and $\text{(n}_{\text{total}}\text{)}$ is the total number of considered fragment ions.

The dialog box contains the following settings:

- MS/MS m/z tolerance:** 0.0010
- MS/MS score threshold:** 80 %
- Use only top N signals:** checked, 20

Buttons at the bottom: OK, Cancel, Help.

Available parameters**• MS/MS m/z tolerance**

Tolerance of the m/z value to search (+/- range).

• MS/MS score threshold

If the score is lower than the threshold, a match is discarded.

• Use only top N signals

Use only N most abundant signals for scoring. ⓘ This option speeds up the search.

[omokshyna](#)

⌚ November 24, 2022 21:13:07

👤 omokshyna

11.22.4 Lipid Annotation

Description

≡ Feature list methods → Annotation → Search spectra → Lipid annotation

This module uses lipid class and ion notation specific fragmentation rules to annotate lipids in the feature lists. Lipids will be annotated based on accurate m/z on MS1 level and matched lipid fragments in MS2 spectra. This rule-based annotation approach follows Lipidomics Standards Initiative guidelines and uses the latest shorthand notation for lipid structure from mass spectrometry to avoid over-annotation. Annotations on species and molecular species level are currently supported.

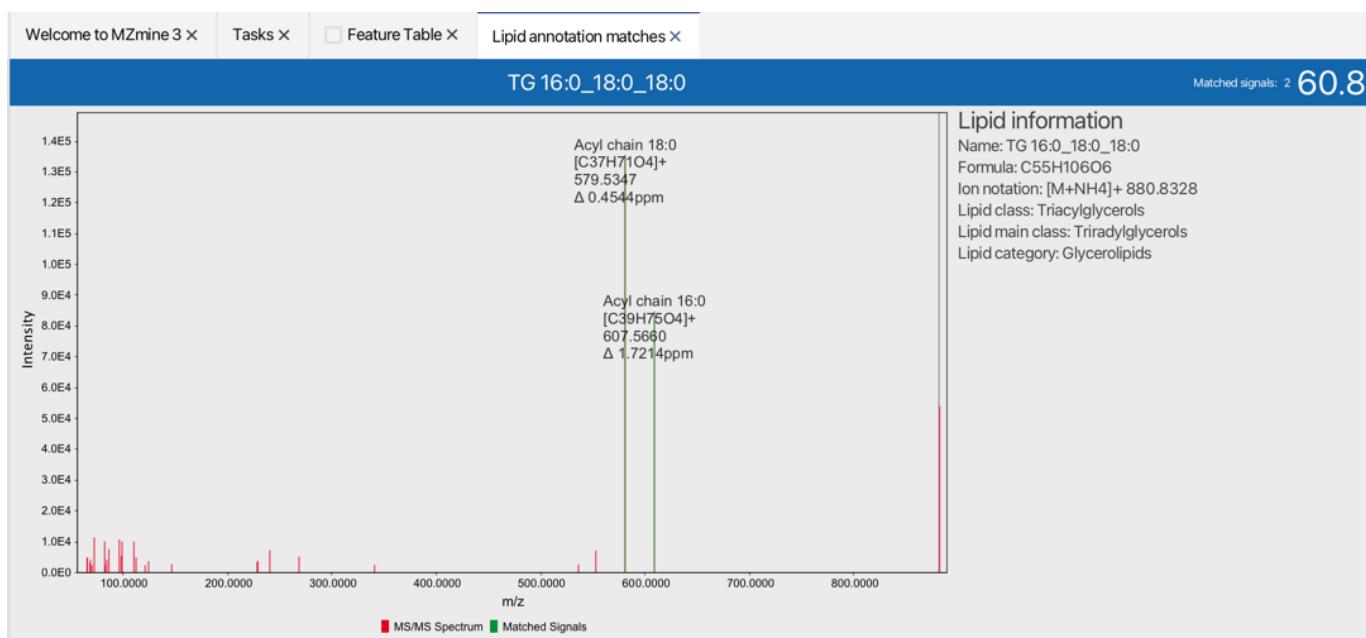


Lipid annotations are directly added to the feature list. Matched fragments can be quickly evaluated. To validate the annotation quality right click on the annotated feature list row and select Matched lipid signals.

≡ Show → Matched lipid signals

Aligned feature list peak gaps dup corr PEARSON r greq 0.85 dp greq 5

Lipid Annotation						
Lipid Annotation	Adduct	Formula	Comment	Δ m/z ppm	MS/MS Score	Matched Lipid Signals
TG 16:0_18:0_18:0	[M+NH4]+	C55H10...	XIC (quick) XIC (dialog) Ion mobility trace Show ▾ Search ▾ Identification ▾ Export ▾ Define manually Delete row(s)			<p>MS/MS Spectrum Matched Signals</p>
TG 14:0_16:0_18:0	[M+NH]		Image Co-located images Feature overview (network) Feature in 2D Feature in 3D Plot using Intensity plot module Show m/z ranges in IMS raw data overview Plot mobility/CCS vs. m/z Mass spectrum Accumulated mass spectrum (FWHM) Best mobility scan Extract spectrum from mobility FWHM Most intense MS/MS Mirror MS/MS (2 rows) All MS/MS Show Pseudo Spectrum DIA spectral mirror: Correlated-to-all signals Isotope pattern Compound DB search results Spectral DB search results Matched lipid signals			<p>MS/MS Spectrum Matched Signals</p>
SM 44:2;20	[M+H]+	C49H9...				<p>MS/MS Spectrum Matched Signals</p>
SM 44:1;20	[M+H]+	C49H9...				<p>MS/MS Spectrum Matched Signals</p>
SM 43:3;20	[M+H]+	C48H9...				<p>MS/MS Spectrum Matched Signals</p>
<input type="text" value="m/z: 380.1692 - 1199.7883"/> RT: <input type="text"/>			Row(s) summary			



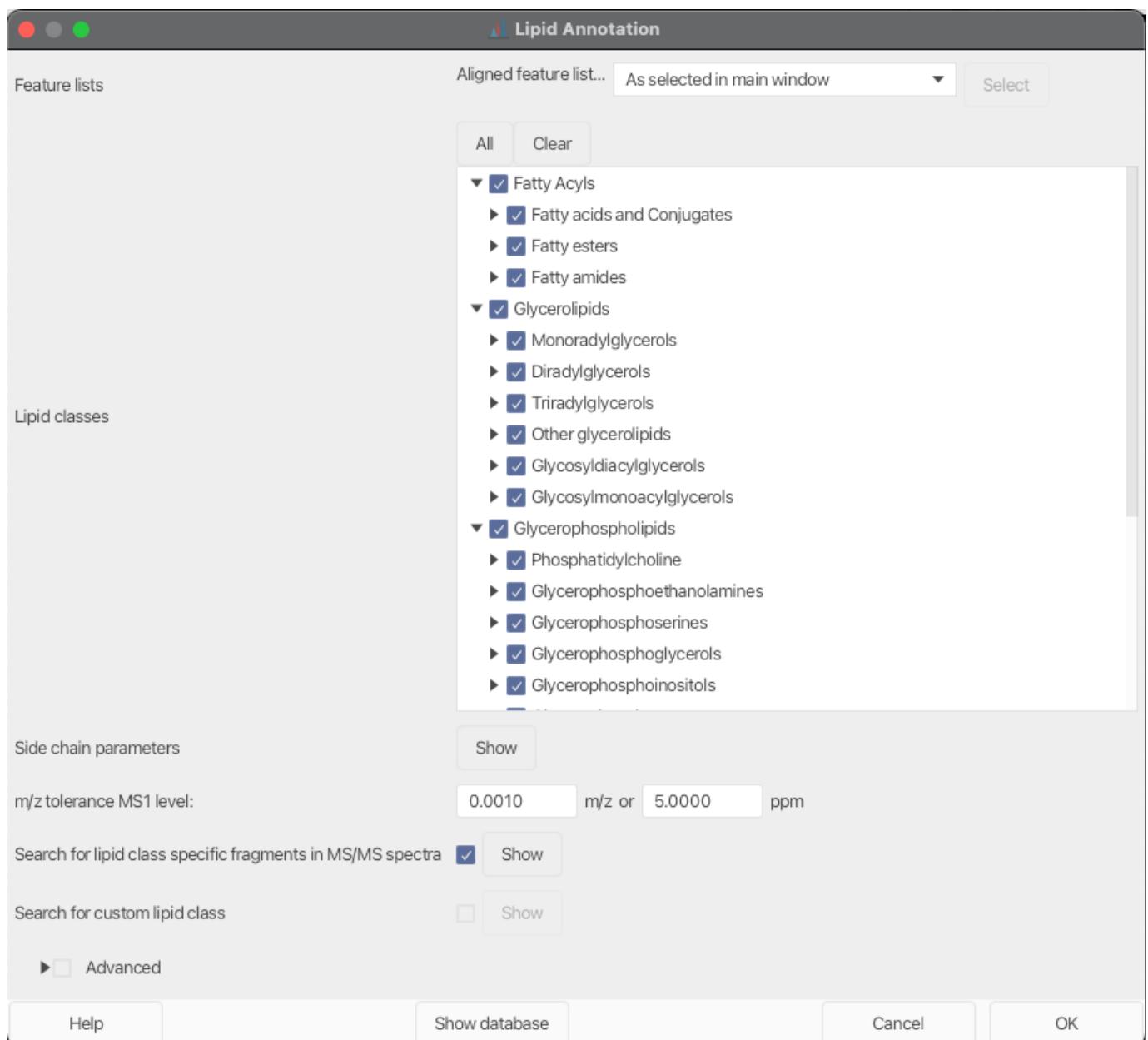
Recommended citations



When using the Lipid annotation module consider reading and citing the original publication and the MZmine 3 publication.

1. Korf, A., Jeck, V., Schmid, R., Helmer, P. O., & Hayen, H. (2019). Lipid Species Annotation at Double Bond Position Level with Custom Databases by Extension of the MZmine Open-Source Software Package. *Analytical chemistry*, 91(8), 5098-5105. doi:10.1021/acs.analchem.8b05493
2. Schmid R., Heuckeroth S., Korf A., et al. Integrative analysis of multimodal mass spectrometry data in MZmine 3, *Nature Biotechnology* (2023), doi:10.1038/s41587-023-01690-2.

Parameters



Lipid classes

This parameter lists all pre-configured lipid classes in MZmine. Each selected lipid class will be added to the search scope.

Side chain parameters

Hide

Side chain parameters	Minimum chain length	12
	Maximum chain length	26
	Minimum number of DBEs	0
	Maximum number of DBEs	6
	Only search for even chain length	<input type="checkbox"/>

Set the allowed min and max for chain length (number of carbons) and double bond equivalents (DBE) in a lipid chain. The number of chains is defined in the lipid class. 12 to 26 carbons and 0 to 6 DBEs is recommended as a starting point.

⚠ Warning

Lipid chain variation highly depends on the analyzed sample.

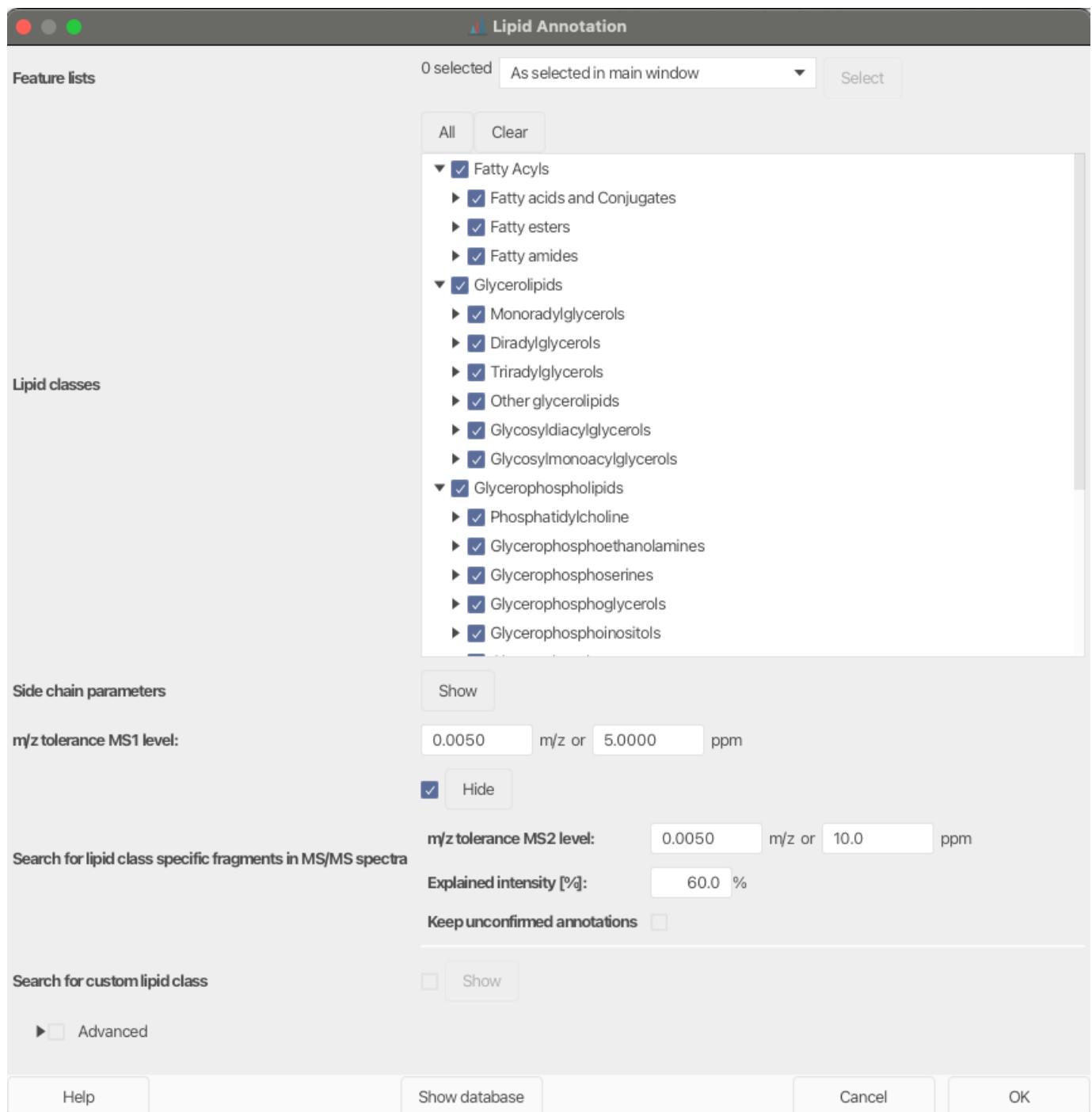
💡 Tip

Odd chain lipid chains can be skipped by "Only search for even chain length to avoid false positive annotations.

m/z tolerance MS1 level

Enter m/z tolerance for exact mass matching on MS1 level. Lipids outside the m/z tolerance will not be matched.

Search for lipid class-specific fragments in MS/MS spectra



Activate this parameter to highly increase lipid annotation confidence.



Mass detection on MS2 level needs to be performed first.

Set a m/z tolerance for matching signals in MS2 spectra. Set a Minimum Explained intensity to accept a lipid annotation. Matches below the set value will be removed, except the optional parameter Keep unconfirmed annotations is selected. Unconfirmed annotations will be marked with a comment in the feature list.

Search for custom lipid class

The screenshot shows the 'Custom lipid classes' search panel. At the top right are buttons for 'Add...', 'Import...', 'Export...', 'Remove selected', and 'Clear'. Below these buttons is a list containing 'MyClass My Custom Lipid Class'. On the left side, there is a search bar labeled 'Search for custom lipid class' and a section titled 'Custom lipid classes'.

If selected, the user can add their own custom lipid class that will be added to the search scope. Click the Add... button or double-click on an already defined custom lipid class to start with the configuration. Configured custom classes can be exported and re-imported for further analysis.



You want your lipid class preset in MZmine? Contribute to the code or contact the developers.

The dialog box has several sections: 'Lipid category' set to 'Glycerolipids', 'Lipid main class' set to 'Triradylglycerols', 'Custom lipid class name' set to 'My Custom Lipid Class', 'Custom lipid class abbreviation' set to 'MyClass', 'Lipid backbone molecular formula' set to 'C3H8O3', and 'Add lipid chains' with options for 'Acyl chain', 'Acyl chain', and 'Alkyl chain'. On the right, the 'Lipid class summary' section provides detailed information about the lipid class, including its species level (MyClass O-60:5), molecular formula (C63H114O5), neutral mass (950.8666), and number of chains (3). Two mass spectra plots are shown: one for [M+Na]+ and one for [M+NH4]+. The [M+Na]+ plot shows peaks for acyl chain 24:0 [6703]+ at m/z 620.0000, acyl chain 20:2 [C43H78NaO3]+ at m/z 665.5843, and acyl chain 22:0 [C47H88NaO3]+ at m/z 723.6626. The [M+NH4]+ plot shows peaks for acyl chain 24:0 [6703]+ at m/z 600.0000, acyl chain 20:2 [C43H79O3]+ at m/z 643.6024, acyl chain 16:3 [C47H89O3]+ at m/z 701.6806, and headgroup fragment [C63H114O3]+ at m/z 933.8633. At the bottom, there are buttons for 'Add...', 'Import...', 'Export...', 'Remove selected', 'Clear', and 'OK'.

Advanced

Optional parameters

Ions to ignore

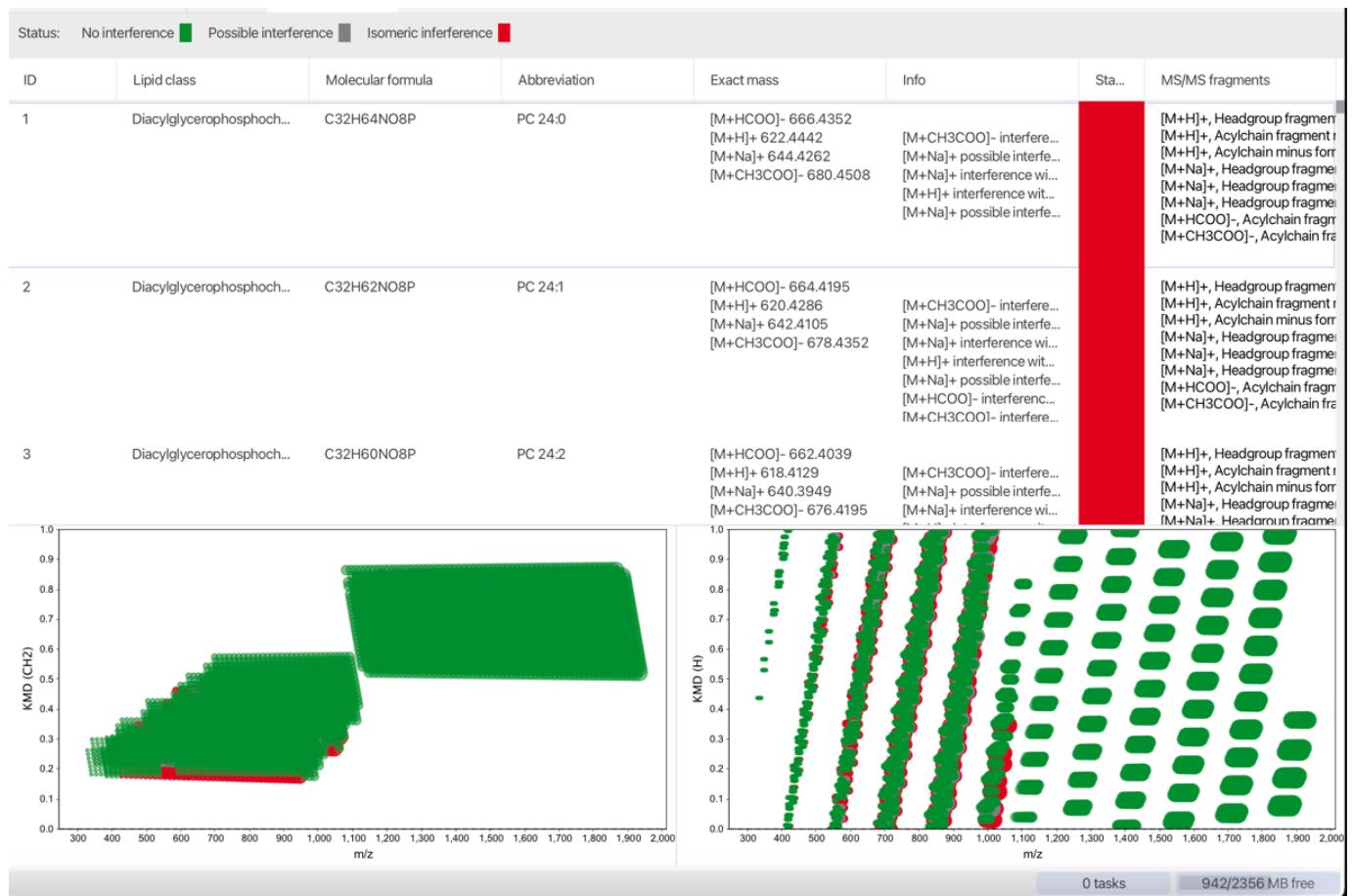
Selected ion notations can be ignored for annotations. This is very important in negative mode when using acetate or formate buffer solutions as LC eluent. Lipids that form adducts with formate in negative mode usually also form adducts with acetate.

Therefore, most preset lipid classes have implemented fragmentation rules for both ion notations. Since MZmine does not know the experimental LC setup, it is highly recommended to manually rule out ions that are unlikely to be formed.

Danger

The delta between formate and acetate is CH₂, exactly the difference of a lipid chain length. Including both ions when analyzing negative mode data will result false positive annotations.

Show database



By clicking the button "Show database" at the bottom of the window, a tab holding information on the selected lipid classes is opened. All lipids are displayed in a table and in two Kendrick mass plots (KMD CH₂ left, KMD H right). Lipids that interfere in the selected m/z window are marked grey, isobaric lipids are marked red. The others are displayed in green.

Tip

As a starting point select a preset lipid class from the combo box at the bottom left corner.

Custom lipid classes and fragmentation rules

As described in the parameters section, custom lipid classes can be defined, stored and reused. The following parameters need be set up:

Lipid Category

Select a lipid category. The selected lipid category influences the lipid annotation algorithm since different rules are applied for species and molecular species level annotation construction. E.g. when using a sphingolipid chain type and a sphingolipid related

fragmentation rule, make sure to set the lipid category to Sphingolipids. Currently featured lipid categories: - Fatty Acyls - Glycerolipids - Glycerophospholipids - Sphingolipids - Sterol Lipids

Lipid Main Class

Select a lipid main class. The selection is for reference only and has no impact on the annotation process.

Custom lipid class name

Define a name for your lipid class. The name is for reference only.



Must be unique when using multiple custom lipid classes.

Custom lipid class abbreviation

Define an abbreviation of the lipid class. The abbreviation will be used for the shorthand annotation.

Lipid backbone molecular formula

Define the molecular formula of a lipid class without lipid chains. E.g. for Diacylglycerophosphocholines (PC) use the formula of glycerol-choline C₈H₂₀O₆PN.



Compared to other lipid categories, sphingolipids share atoms in the backbone chain and the polar headgroup. Therefore, the determination of the backbone formula is more complicated. E.g. for Sphingosine use C₃H₉N as the lipid backbone formula. Missing oxygens will be added by the selected Sphingolipid backbone chain.

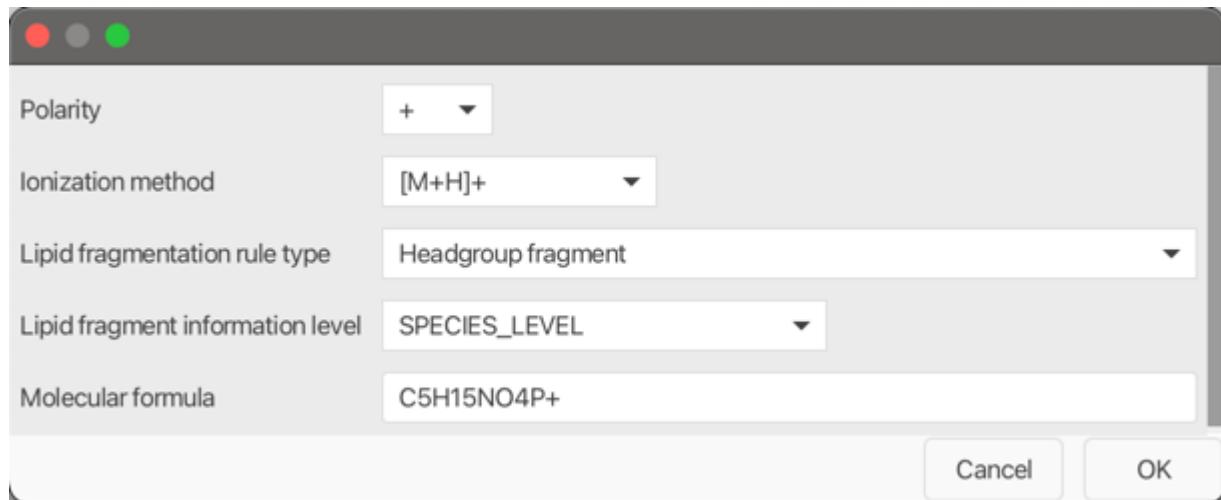
Add lipid chains

Add lipid side chains by selecting the Add... button. The following lipid chains are currently supported: - Acyl chain - Acyl mono hydroxy chain - Alkyl chain - Amid chain - Amid mono hydroxy chain - Sphingolipid mono hydroxy backbone chain - Sphingolipid di hydroxy backbone chain - Sphingolipid tri hydroxy backbone chain

Add lipid chains



When using Sphingolipid backbone chains make sure to use Sphingolipids as Lipid category.

ADD FRAGMENTATION RULES**Polarity**

Select polarity (+ or -).

Ionization method

Select an ion notation from the combo box. The ion notation will also be considered when calculating the intact lipid ion.

Lipid fragmentation rule type

Use the pseudo spectra on the right containing in-silico fragments based on defined rules, to validate your custom fragmentation rules

Following lipid fragmentation rule types are currently supported:

- Headgroup fragment - Fix ion formula: C5H15NO4P⁺ for choline head group fragment
- Headgroup fragment nl - Fix neutral loss: C2H8NO4P for PE head group loss
- Acylchain fragment - [M-H]⁻ form of any acyl chain
- Acylchain fragment nl - Neutral loss of an acyl chain
- Acylchain plus formula fragment - An acyl chain fragment plus a defined formula
- Acylchain plus formula fragment nl - A neutral loss of an acyl chain plus a defined formula
- Acylchain minus formula fragment - An acyl chain fragment minus a defined formula
- Acylchain minus formula fragment nl - A neutral loss of an acyl chain minus a defined formula
- Two acylchains plus formula fragment - Two acyl chains combined plus a defined formula
- Select CL lipid class to see an example
- Alkylchain plus formula fragment - An alkyl chain fragment plus a defined formula

⚠ Warning

Use the following Sphingolipid related rule types only when selecting Sphingolipids as Lipid Category.

- Sphingolipid mono hydroxy backbone chain fragment
- Fragment of the sphingolipid backbone chain
- Sphingolipid di hydroxy backbone chain fragment
- Fragment of the sphingolipid backbone chain
- Sphingolipid tri hydroxy backbone chain fragment
- Fragment of the sphingolipid backbone chain
- Sphingolipid mono hydroxy backbone chain minus formula fragment
- Fragment of the sphingolipid backbone chain minus a defined formula
- Sphingolipid di hydroxy backbone chain minus formula fragment
- Fragment of the sphingolipid backbone chain minus a defined formula
- Sphingolipid tri hydroxy backbone chain minus formula fragment
- Fragment of the sphingolipid backbone chain minus a defined formula
- Amid chain fragment
- Fragment of an amid chain
- Amid chain fragment nl
- Neutral loss of an amid chain
- Amid chain plus formula fragment
- Amid chain plus formula fragment
- Amid chain plus formula fragment nl
- A neutral loss of an amid chain plus a defined formula
- Amid chain minus formula fragment
- Amid chain minus formula fragment
- Amid chain minus formula fragment nl
- A neutral loss of an amid chain minus a defined formula
- Amid mono hydroxy chain fragment
- Fragment of an oxidized amid chain
- Amid mono hydroxy chain plus formula fragment
- Oxidized amid chain plus formula fragment
- Amid mono hydroxy chain minus formula fragment
- Oxidized amid chain minus formula fragment
- Precursor
- Special case for fatty acids, which checks if ONLY the precursor is present.

[Robin Schmid, kyrasolaris01, omokshyna](#)

⌚ November 25, 2025 10:29:24

👤 [Robin Schmid, kyrasolaris01, omokshyna](#)

11.22.5 MS2 Similarity Search

Description

≡ Feature list methods → Annotation → Search spectra → MS2 similarity search

Ions arising from compounds with similar chemical structures often give similar fragmentation patterns (MS2 spectra). Therefore, calculating the similarity between MS2 spectra is a useful approach for the discovery of structurally similar compounds.

This module calculates the similarity between **centroided MS2 spectra** associated with two feature lists. These two feature lists can be the same feature list, or different feature lists.

The module outputs the result of the search, the MS2 similarity comparisons of feature list (1) with feature list (2), into the "Identity" column of feature list (1).

 In practice, you can consider the features in feature lists (1) as "bait", which is used to "fish" for MS2 similarity from feature list (2). Feature list (1) or feature list (2) can be a single peak + MS2 spectra, or an entire experiment.

MS2 SIMILARITY SCORE

The similarity metric used is as follows:

- For all MS2 spectra in feature list (1) "**MS2 spectra A**", and feature list (2) "**MS2 spectra B**", iterate over all ions in MS2 spectra A (**ion "i"**), and over all ions in MS2 spectra B (**ion "j"**).
- For a given ion, only include this ion in following calculations if its intensity is greater than the **minimum ion intensity parameter**.
- Compare the m/z values of ions "i" and "j". If these m/z values are within the range specified by the **m/z tolerance** parameter, consider these ions identical, and therefore "matched".
- If two ions match, roughly score the match by multiplying the intensity of ions "i" and "j". Save that as the "**ion match subscore**".
- Repeat this for every ion i and j in MS2 spectra A and MS2 spectra B, and report the sum of the ion match subscores as the total "**spectral match score**".
- If this spectral match score is greater than the **minimum spectral match score** parameter, annotate the "Identity" column of feature list (1) with the matched ions, and the total spectral match score of the MS2 similarity calculation.

INTERPRETING THE RESULTS

The score from of a MS2 similarity match should not be taken as an absolute measure, as it depends on the instrument reported intensity value (which is an uncalibrated and relative measure).

That being said, as the reported score increases when the intensity of the matched ion is higher, it is useful as a quick metric to find the matches between the most intense MS2 spectra, and potentially the most reliable compounds.

 It is worth mentioning that the link between the MS2 spectra, and the presumed precursor ion in the feature list is somewhat tentative. As the isolation window of the quadrupoles typically used for selection of the precursor ion in MS2 fragmentation analysis is typically around ~1 m/z unit, ions from the MS2 fragmentation spectra from an abundant compound with a long chromatographic tail will often show up in the MS2 values of unrelated compounds, but whose precursor isolation window picks up ions from the original compound.

An experimental Python script which converts the MS2 similarity relationships exported from MZmine2 in CSV format into GraphML format suitable for viewing in the freely available **graph manipulation software Cytoscape** is available on Github (https://github.com/photocyte/ms2_graph).

Parameters

Feature list (1)

A single feature list with features that have associated centroided MS2 spectra. Results from the module are output into the identifications column of Feature list (1)

Feature list (2)

A single feature list, which has centroided MS2 spectra (in the masslist). This can be the same as Feature list (1), or a different peaklist / experiment.

For best results Feature list (2) should have MS data with the same polarity as Feature list (1), and close range of m/z values close (e.g. LC/MS data obtained on the same day, or m/z calibrated between the two feature lists).

m/z tolerance

Maximum allowed m/z difference between two ions to be considered identical, and therefore "matched".

Minimum MS2 ion intensity

Minimum ion intensity to consider in MS2 comparison. Ions of the MS2 spectra below this threshold will be ignored.

 This parameter depends on your instrument, but 1e5 is a reasonable value.

 Set to 0 to use all ions.

Minimum ion(s) matched per MS2 comparison

Minimum number of matched ions needed in a given MS2 similarity comparison. Otherwise, that spectral match will not be reported.

 This depends on the compounds being compared, but roughly speaking 2-5 is a reasonable number. For complex spectra values of 10-20+ may be reasonable.

 Set to 0 to report all matches.

Minimum spectral match score to report

Minimum spectral match score threshold, below which spectral matches will not be reported.

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

11.23 Other parameters

11.23.1 Merge & select fragment scans

This parameter is used in many modules like spectral library search, molecular networking, scan export, etc. to control which fragment spectra are filtered, merged, and selected. The final list of scans used in those modules may be input scans, e.g., a single MS2 scan or a GC-EI-MS pseudo MS2 scan, or they grouped and merged on various levels. To make this complex process simple and to provide options, there are multiple ways to configure this parameter.

Merging is generally done in this order:

- split scans by samples
- merge for each energy and across energies (for each sample)
- merge across all samples

For MSⁿ data:

- merge MSⁿ scans for each MSⁿ tree node (precursor ion) for each energy and across energies (for each sample)
- merge all MSⁿ scans to a pseudo MS² scan

THE FOUR CONFIGURATION OPTIONS ARE:

1. **Merged (simple):** Presets with few settings
2. **Merged (preset):** Presets with more control
3. **Input scans:** Input scans without merging
4. **Advanced:** All options

1. MERGED (SIMPLE)

Simple preset-based scan selection that will always merge scans across all samples. For more control, choose **Merged (preset)**. The final selection of scans is based on presets defined below. In general, all presets include a scan merged across all energies and for MSⁿ data a pseudo MS² scan that merges the whole MSⁿ tree.

Presets

1. **Single scan: Merged across energies** - Merge all input MS² scans. For MSⁿ data also merge all MSⁿ to an additional pseudo MS². A great option for speed by limiting the number of fragment scans.
2. **Representative scans: Each energy & merged across energies** - Like 1 and adds scans for each fragmentation energy. This is a great option for algorithms where the intensity differences in various energies is important - like spectral library matching by cosine similarity.
3. **MSⁿ tree: Merged across energies & pseudo MS²** - Like 1 when used on MS² data - but for MSⁿ one spectrum per tree node (precursor ion) will be added. This option can be used with MS² data as well.
4. **Representative scans or MSⁿ tree: Each energy, merged across energies & pseudo MS²** - Like * 2 and 3* this module will add spectra for each MSⁿ tree node and fragmentation energy for each. This option can be used with MS² data as well.

Merging m/z tolerance

Used to match signals across scans and merge them. This tolerance is typically higher than the advertised mass accuracy. As it is from scan-to-scan and across samples.

2. MERGED (ADVANCED)

Similar to the simple preset-based setup. The same presets but with more additional parameters, as described below:

Merge

Select one or multiple options how to handle multiple samples:

- Across samples (like the simple setup)
- Each sample

Intensity merging mode

- **Summed** - suppresses low intensity signals and spurious noise but changes the amplitude
- **Maximum value** - retains the general amplitude
- **Average** - average intensity of signals across all merged scans

3. INPUT SCANS

Options to disable merging and choose the input scans instead. This is either:

1. **Most intense scan across samples**: One scan per fragmentation energy across all samples with the highest summed intensity
2. **Most intense scan per sample**: One scan per fragmentation energy for each sample with the highest summed intensity
3. **All scans**: All input scans will be selected

4. ADVANCED

This is the full setup with the highest flexibility. The advanced option allows to combine multiple different scan selections.

Merging options

Select multiple options that will be combined with each other. Options include handling of multiple samples, fragmentation energies, and MSⁿ data.

1. **Across samples** - Merging happens across all samples .
2. **Each sample** - Merging happens for each sample separately and also across samples if selected.
3. **Across energies** - Merging happens across energies to combine all into one.
4. **Each energy** - Merging happens for each fragmentation energy and also across energies if selected.
5. **MSⁿ tree** - Merging happens for each MSⁿ tree node (precursor) - this option will be combined with sample and energy handling. This has no effect on MS² data.
6. **MSⁿ to pseudo MS²** - Merges all MSⁿ spectra into a single 'MS²' spectrum. This has no effect on MS² data.

Also include input scans

Also include unmerged input scans into the final scan selection. This is from the "raw" or "source" scans that are input to the merging.

1. **Most intense scan across samples**: One scan per fragmentation energy across all samples with the highest summed intensity
2. **Most intense scan per sample**: One scan per fragmentation energy for each sample with the highest summed intensity
3. **All scans**: All input scans will be selected
4. **None**: Select **no** input scan but rely only on merged scans. This requires merging to be active, by selecting valid **merging options**

Examples

Example 1: Across samples + across energies: This will produce a single scan that is merged across all scans of all energies and samples. Adding **each sample** will then also export one more scan for each sample (if this sample has fragmentation spectra). Adding **each energy** will then add scans for each energy state. If across samples and each sample are selected, this means there will be spectra added for each sample and then merged across all samples.

Example 2: Each sample + Single most intense scan + across energies: This combination will export the single best scan in each sample and then one merged scan per sample - both only if there are fragment scans in this sample.

Robin Schmid

⌚ January 9, 2025 20:58:14

👤 Robin Schmid

11.23.2 Spectra similarity

Spectral m/z tolerance

Spectral m/z tolerance is used to match all signals between spectra of two compared raw files.

MS level

MS level of scans that should be compared. It can be 1 for MS1 or 2 for MS2 level.

Compare spectra similarity

1. **Weighted dot-product cosine** - used to determine the similarity between two spectra (usually library and query spectra). This option is used for MS2 level.
2. **Composite dot-product identity** - used to determine the similarity between two spectra (usually library and query spectra). Especially useful for very reproducible generation of spectra (GC-EI-MS). Takes into account the relative intensities of neighbouring signals in the two spectra. This option is used for MS1 level.

Additional setup of spectra similarity comparison enables modification of the following parameters:

Weights

Weights for the m/z and intensity values. Usually, MassBank is used, in which higher m/z values contribute more to the cosine similarity calculation.

Minimum cos similarity

Minimum cosine similarity for a match between compared spectra.

Handle unmatched signals

Usually, **keep all and match to zero** is used, which will take all signals into account, and the unmatched ones will decrease the cosine similarity.

[lalalana5, omokshyna](#)

 November 24, 2022 21:13:07

 [lalalana5, omokshyna](#)

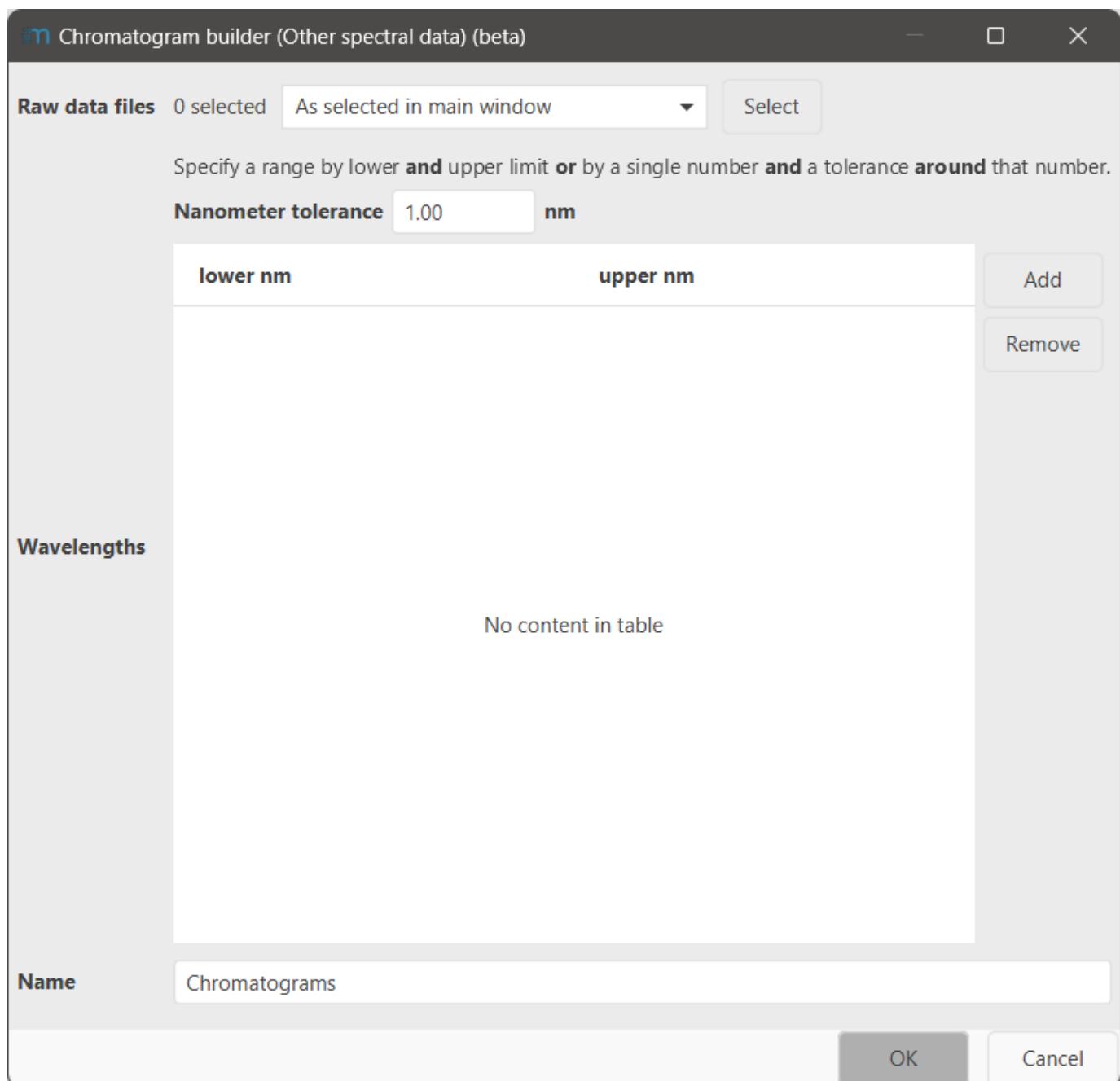
11.24 Other detectors (UV)

11.24.1 Chromatogram builder (Other spectral data)



To process data from other detectors, the Other detector service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

The **Chromatogram builder (Other spectral data)** constructs chromatograms for data of other detectors that were acquired alongside an MS detector. Spectral raw data without MS data cannot be analysed by mzmine.

Parameters**Raw data files**

Select the MS raw data files with spectra from other detectors to build chromatograms from.

Wavelengths

Specify the wavelengths in **nm** that the chromatograms shall be built for.

You can specify either a range by giving a **lower and upper** limit, or by specifying only a single wavelength (in the lower **or** upper cell) and a **Nanometer tolerance** around the given single value. The tolerance is **not** applied to ranges, only single wavelengths.

The intensity inside a wavelength range is averaged when building the chromatograms.

Name

A name for the chromatograms. If that name is already in use, the existing chromatograms will be replaced.

⌚ July 28, 2025 13:48:29

👤 Steffen Heuckeroth

11.24.2 Shift, trim, and bin traces (Other detectors)



To process data from other detectors, the Other detector service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

Commonly, a shift in the observed retention time is observed between the MS detector and other detectors due to different lengths in flow path.

Sudden changes in background intensity, e.g., by changes in solvent composition, may lead to subpar performance of the baseline correction. In such cases, it may be useful to trim the traces of other detectors.

As other detectors typically have a higher acquisition rate than mass spectrometers, it may be useful to bin the detected intensities/absorptions together to achieve a similar acquisition rate for better [cross-correlation](#) and processing performance.

Parameters

m Shift, trim, and bin traces (Other detectors) (beta)

Raw data files	0 selected	As selected in main window	▼	Select
Chromatogram type: ABSORPTION				
Range unit filter: range unit filter				
Trace selection	Range label filter: range label filter			
Description filter: description filter				
Trace type: raw				
RT shift (min)	0.00			
Trim RT range	<input checked="" type="checkbox"/>	0.00	-	100.00 min. Auto range
Bin width (manual/auto)	<input checked="" type="checkbox"/>	auto	▼	5
<i>Show preview</i> <input type="checkbox"/>				
OK Cancel Help				

Raw data files

Select the MS data files to shift the selected traces of.

Trace selection

Select the specific traces you want to shift. Typically, it is not necessary to shift all existing traces, but only the ones you want to correlate with the MS, e.g., the UV (=Absorption) traces

For a detailed description of all sub-parameters see [trace selection parameter](#).

RT shift

Specify the RT shift that shall be applied to the other detector. The numeric value will be added to the retention time (minutes) of the other detector trace. Positive values will shift the trace further back, negative values will shift the trace to the front.

Trim RT range

Specify the RT range you want to keep. Trimming will be applied **after** shifting.

Bin width (manual/auto)

Set how many points shall be binned. "Auto" will adjust the acquisition rate of the other detector to **four** times the MS acquisition rate. "Manual" will bin the given number of points manually.

Binning will be applied **after** shifting and trimming.

⌚ July 28, 2025 13:48:29

👤 Steffen Heuckeroth

11.24.3 Baseline correction (Other detectors)

Info

To process data from other detectors, the Other detector service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

Baseline correction is a useful tool to correct for large drifts in baseline during chromatographic separation. However, it can be tricky to optimize the baseline correction.

Info

While baseline correction can correct some drifts in the baseline, the better approach is usually to acquire a reference wavelength and reference the acquired data on that during the data acquisition.

Parameters

m Baseline correction (Other detectors) (beta)

Raw data files	2000ng_mL_Gradient8...	As selected in main window	<input type="button" value="Select"/>
Chromatogram type:	ABSORPTION		
Range unit filter:	range unit filter		
Trace selection	Range label filter:	range label filter	
	Description filter:	description filter	
	Trace type:	pre-processed	
Suffix	bl		
Baseline corrector	LOESS		
Exclude peaks	<input checked="" type="checkbox"/>		
Percentage of baseline samples	5.0 %		
Bandwidth	10 %		
Iterations	5		
Show preview	<input type="checkbox"/>		
<input type="button" value="OK"/> <input type="button" value="Cancel"/>			

Raw data files

Select the MS data files to baseline correct the selected traces of.

Trace selection

Select the specific traces you want to baseline correct. Typically, it is not necessary to baseline correct all existing traces, but only the ones you want to correlate with the MS, e.g., the UV (=Absorption) traces

For a detailed description of all sub-parameters see [trace selection parameter](#).

Suffix

A suffix that will be appended to the name of the individual trace.

Baseline correctors

Exclude peaks (all correctors)

If selected, two iterations per baseline correction will be performed internally. Initially the baseline is corrected for the full chromatogram. Then, the baseline is subtracted and a peak detection is performed to exclude peaks from the baseline calculation. In a second iteration, the baseline correction is reapplied to the chromatogram, without including the peaks in the baseline calculation.

Percentage of samples

Specify how many of the data points of the chromatogram shall be included in the baseline calculation. This values depends on the acquisition rate of the detector and if previous [binning](#) has been applied. The higher the acquisition rate, the lower the **percentage of samples**.

Loess

The LOESS (locally estimated scatterplot smoothing) baseline corrector fits the data using the LOESS algorithm. It is the recommended baseline corrector in most scenarios, due to its robustness. The LOESS baseline corrector smoothes the baseline data.

Bandwidth

Specify how wide the window for the baseline calculation should be at any point of the chromatogram. E.g., 10% means that 10% of the total data points are used at any specific retention time. For a chromatographic run of 10 min, a total window of 1 min around each point will be taken into account.

Iterations

Specify how many iterations the LOESS algorithm shall perform. This is independent of the iterations specified in the [exclude peaks](#) parameter. This parameter may increase robustness against sudden changes in the baseline.

Polynomial

The polynomial baseline corrector fits the data using an n^{th} degree polynomial.

Degree of Polynomial

Specify the degree of the polynomial. The higher the degree, the more the baseline will resemble the original data (which may be a disadvantage).

Spline

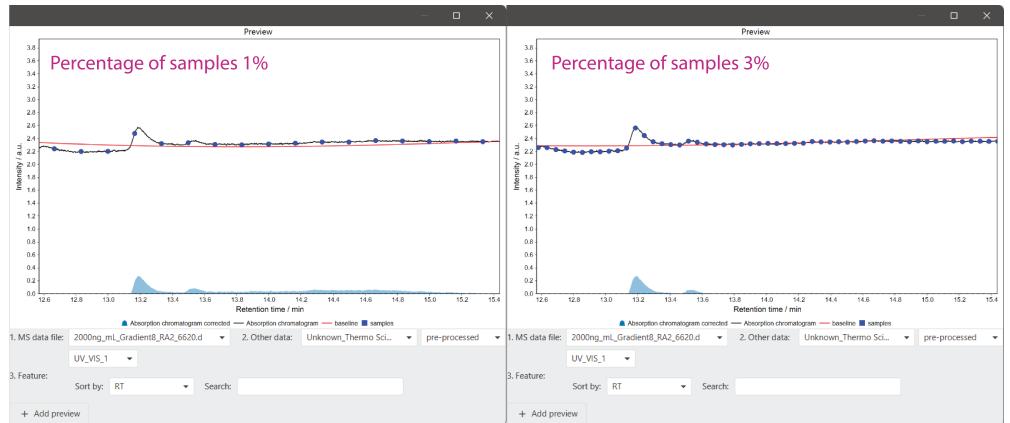
The Spline baseline corrector uses a 3rd degree polynomial to fit 3 points of the raw data. This means, that every input point will appear 1:1 in the corrected data, without any smoothing or fitting (see [LOESS](#), [Polynomial](#)).

Optimization

The **show preview** option is recommended to optimize the baseline correction.

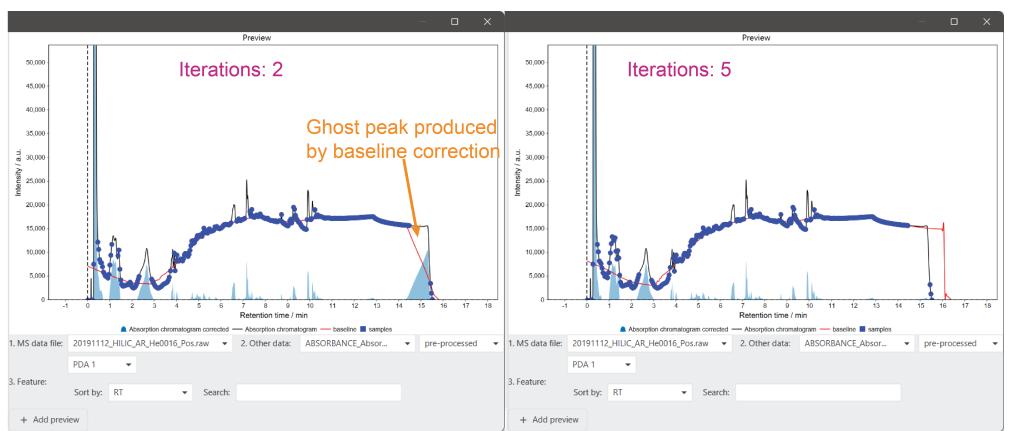
Percentage of samples

The percentage of samples specifies how many data points of the baseline will be included in the baseline correction. This parameter depends vastly on the acquisition rate. The influence of this parameter is shown here:



Iterations (LOESS)

Increasing the number of iterations may reduce the occurrence of ghost peaks due to baseline correction. In case this does not help, [trimming traces](#) may be the better option.



⌚ July 29, 2025 10:34:57

👤 Steffen Heuckeroth

11.24.4 Local minimum resolver (Other data)



To process data from other detectors, the Other detector service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

The resolver detects individual chromatographic peaks in a trace and resolves them into individual features (see [trace types](#)).

At the same time, the area percentage is calculated in reference to the [preprocessed trace](#).

Parameters

Trace selection

Select the specific traces you want to resolve. Typically, it is not necessary to resolve all existing traces, but only the ones you want to correlate with the MS, e.g., the UV (=Absorption) traces

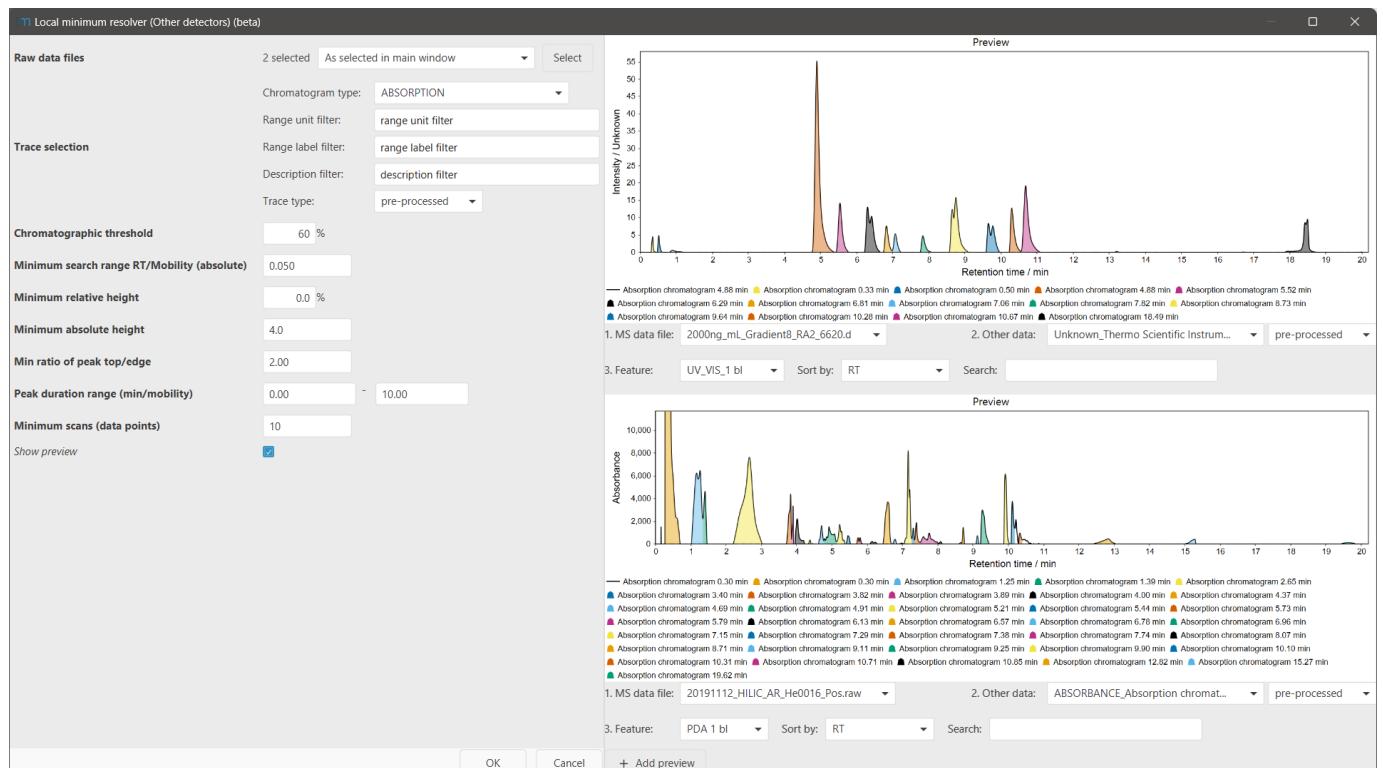
For a detailed description of all sub-parameters see [trace selection parameter](#).

Chromatographic threshold

This filter assumes that most of the data points contain no signals to remove unnecessary noise.

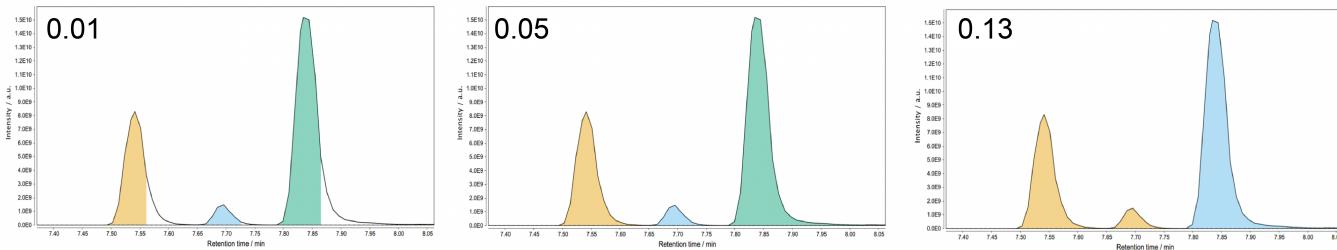
Specify the percentage of data points in the trace to be removed before local minima search. This represents an important filter for noisy chromatogram and significantly reduces the processing time. The algorithm finds the intensity value (threshold) that leaves the specified percentage of data points in the trace below the given value. All such data points are removed.

The typical values for data from other detectors may be significantly different from from MS data. E.g., MS data may require 85%, whereas other data may require lower values such as 50%.



Minimum search range

Size of the RT window examined for local minimum search. An overly narrow search range can cause peak edges to be cut off, whereas a too wide search ranges might lead to an incomplete resolution of narrowly eluting peaks (see example below).



Minimum absolute height

Minimum absolute intensity a peak needs to reach to be retained as a feature.

Minimum relative height

Minimum relative intensity (with respect to the highest data point in the trace) a peak needs to reach to be retained as a feature. This parameter can be used in combination with the **Minimum absolute height** setting as to filter the resolved features to be retained.

Min ratio of peak top/edge

Minimum ratio between the intensity of the highest (apex) and side (left and right 'edges') points of a peak, to retain it as a feature. The peak edges have to be X times less intense than the peak apex for the feature to be retained.

💡 This parameter can best be optimized using the *Show preview* option. We recommend values between 1.7 (not baseline separated) and 2 to start the optimisation.

Peak duration range

Range of acceptable peak length expressed in minutes (RT dimension). This parameter can be used to filter out noisy features based on their overly short (or long) duration.

Minimum scans (data points)

Minimum number of data points a resolved peak needs to have to be considered valid and retained as a feature. This parameter can be used along with the **Peak duration range** setting as peak duration constraint to filter out noisy features.

⌚ July 28, 2025 13:48:29

👤 Steffen Heuckeroth

11.24.5 Correlate MS features with other detectors



To process data from other detectors, the Other detector service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

This module correlates features of an MS workflow, e.g., [LC-MS](#), to features from other detector traces, allowing you to find which MS features are behind a certain peak in the UV, CAD, or radioactivity chromatograms.

Parameters

m Correlate MS features with other detectors (beta)

Feature lists	Aligned feature list... <input type="button" value="As selected in main window"/> <input type="button" value="Select"/>
Trace selection	Chromatogram type: <input type="button" value="ABSORPTION"/> <input type="button" value="range unit filter"/> Range unit filter: <input type="button" value="range label filter"/> Range label filter: <input type="button" value="description filter"/> Description filter: <input type="button" value="features"/>
RT tolerance	4.00 <input type="button" value="seconds"/>
Minimum correlation score	0.800

Feature lists

Select the feature list you want to correlate the other traces for. Each MS data file in the selected feature list(s) will only be correlated to other detector features of it's own file. The selected feature list may be an aligned feature list.

Trace selection

Select the specific traces you want to correlate to the MS features. Typically, it is not necessary to correlate all existing traces.

For a detailed description of all sub-parameters see [trace selection parameter](#).

RT tolerance

Allowed RT difference of the peak maxima in the MS and other detector features.

⚠ Warning

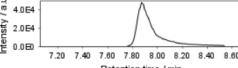
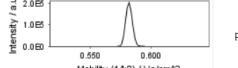
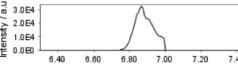
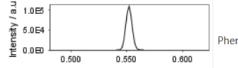
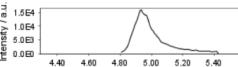
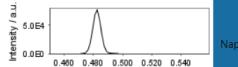
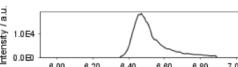
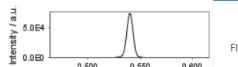
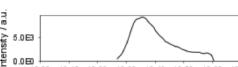
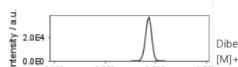
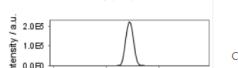
This allowed RT tolerance does not cause an alignment of the other detector feature and the MS feature. If the traces are affected by an retention time shift, use the [Shift traces](#) module to correct for that shift. Not correcting an existing retention time shift will cause poor correlation between the two detectors.

Minimum correlation score

Set a minimum (Pearson) correlation factor between the chromatographic shape of the two features.

Results

If correlated traces are found, the "Correlated traces" column will be added to the feature table. (Not visible by default) You can enable it by clicking the "+" in the upper right corner of the feature table and selecting the "Features:Correlated traces" type and the sub columns you want to see.

ID	+ m/z	+ RT	Area	Height	+ Mobility	Shapes	Mobilograms	Compound DB	Correlated traces				
									Compound DB	Formula	Height	Correlated traces	Area
41	202.0783	7.87	7.2E3	4.7E4	0.582			Pyrene: [M]+: 0.938	C16H10	4.7E4	Absorption, Calculated (0.956)	5.2E-1	1.7 %
33	178.0782	6.87	3.8E3	3.3E4	0.552			Phenanthrene: [M]+: 0.915	C14H10	3.3E4	Absorption, Calculated (0.930)	9.9E-1	3.2 %
20	128.0626	4.94	2.7E3	1.6E4	0.483			Naphthalene: [M]+: 0.919	C10H8	1.6E4	Absorption, Calculated (0.980)	8.1E0	26.3 %
30	166.0778	6.47	2.8E3	1.9E4	0.539			Fluorene: [M]+: 0.984	C13H10	1.9E4			
56	278.1100	10.35	1.4E3	9.4E3	0.693			Dibenzo[ah]anthracene: [M]+: 0.894	C22H14	9.4E3	Absorption, Calculated (0.886)	1.9E0	6.1 %
42	228.0940	8.78	1.0E4	4.7E4	0.621			Chrysene: [M]+: 0.927	C18H12	4.7E4	Absorption, Calculated (0.982)	3.2E0	10.4 %

Additionally, the [Correlation dashboard](#) allows a deeper investigation and also manually assigning or removing the correlated traces.

Clear correlated traces

This module removes all correlated other detector traces from the selected MS feature list.

PARAMETERS

Feature lists

The feature list to remove the correlated traces from.

⌚ July 29, 2025 10:09:01

👤 Steffen Heuckeroth

12. Visualization modules

12.1 Visualization modules

MZmine provides a range of interactive visualization tools for analysis of raw and processed data. JFreeChart library is used for majority of plots. Most of the generated plots are interactive.

The following **functions** are available:

- Use the + or - key on the keyboard to zoom in or out.
- Scroll with the mouse to zoom in or out.
- Drag the mouse from left to right to select the area to zoom in.
- Drag the mouse from right to left to zoom out to the default view.
- Single click on the y-axis to auto set the intensity to auto height.
- Double click on the y- or x-axis to reset the zoom to default.

In the right part of the plot there is a toolbar. Its functionality is also included in a pop-up menu, which appears when you make right click on the plot area.

[omokshyna](#)

 November 24, 2022 21:13:07

 [omokshyna](#)

12.2 MS data visualisation

12.2.1 Raw data overview (LC-MS)

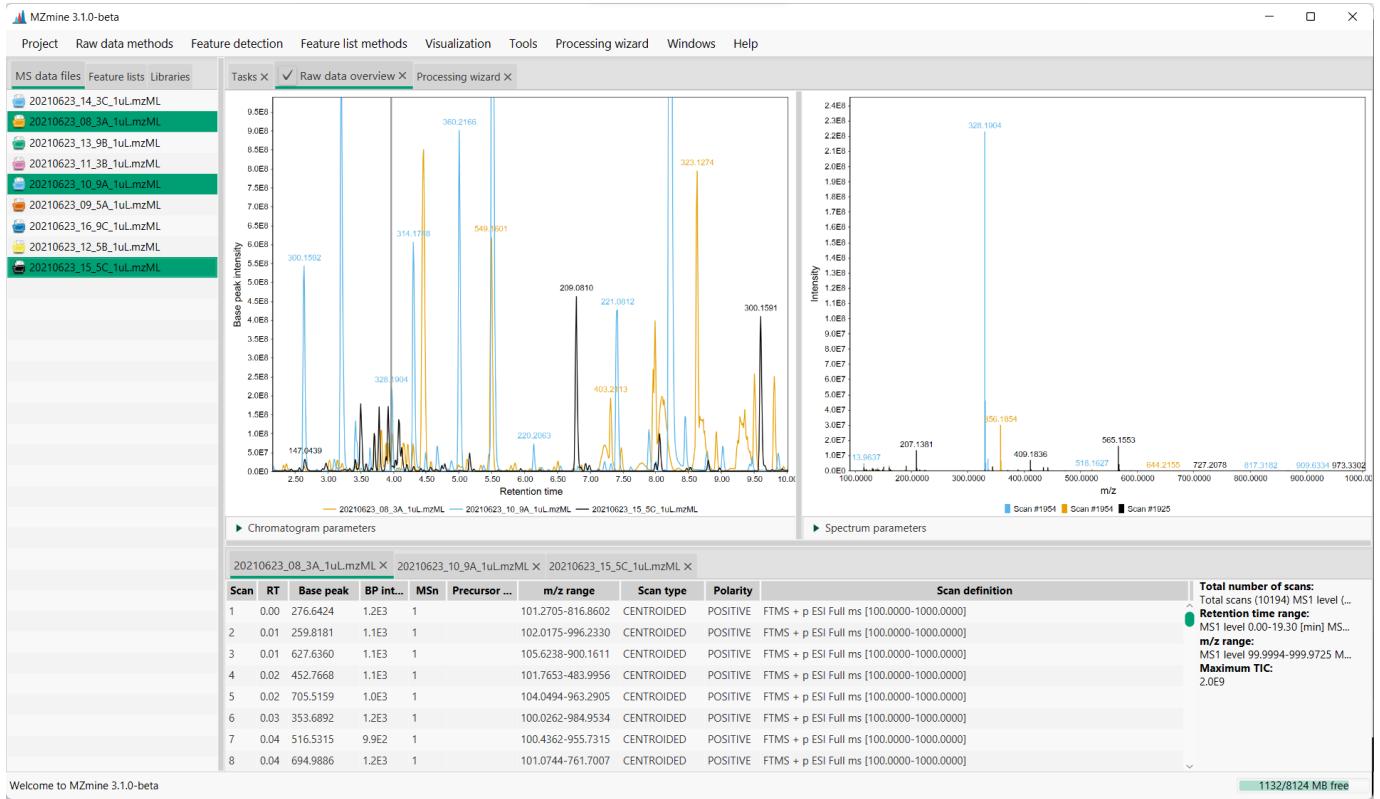
Description

≡ **Visualization → Raw data overview**, or it can be accessed by **double-click of left mouse button**, or by clicking the right mouse button on **MS data files** table and choosing **≡ Show raw data overview**

Raw data overview allows user to explore both chromatogram and MS views across all the selected files. By double-clicking on the raw file of interest in the "MS data files" tab, **Raw data overview** tab will open in the main content pane. Raw data overview can either display single or multiple overlaid chromatograms, depending on how many raw files are selected.

💡 To allow selection of multiple raw files, click the checkbox near the tab name "Raw data overview".

The view consists of three panes - chromatogram representation (on the left), mass spectrum (on the right), and table with tabs containing additional information about raw data files.



Chromatogram parameters

Show XIC

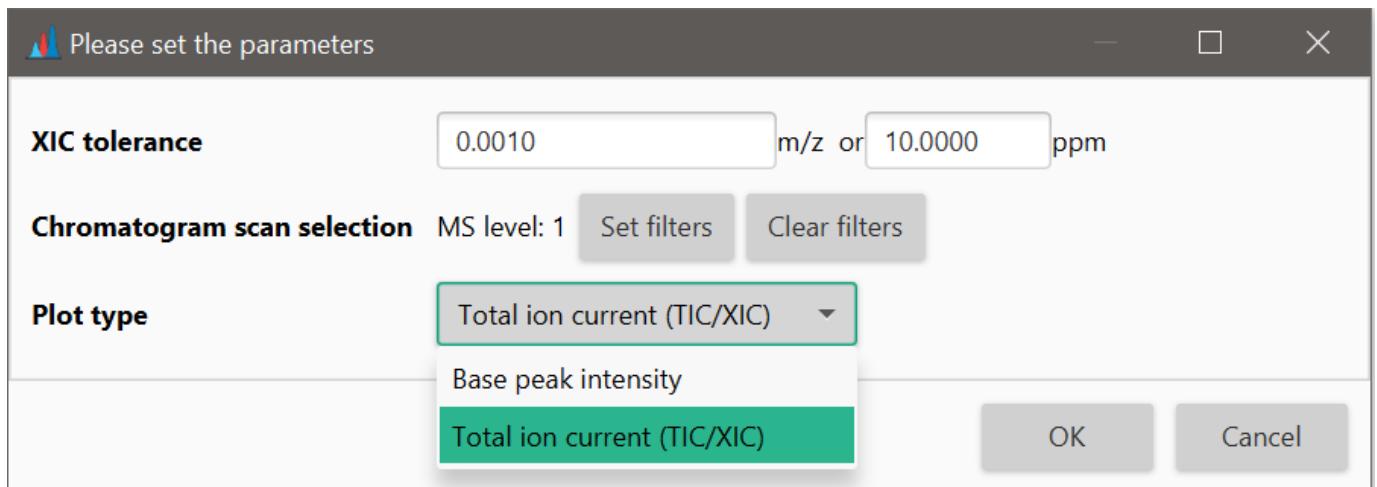
To display an extracted ion chromatogram (XIC), the user can tick the "Show XIC" box. There are several options for defining the m/z range - from mass, from formula and auto range.

▼ Chromatogram parameters

Show XIC Update chromatogram(s)

149.9981 - 1499.9678 Auto range From mass From formula

"Setup" button allows to choose the appropriate plot type. **Base peak intensity** plot only shows the signal of the most intense mass peak in each MS spectrum, while the **Total ion current** plot shows the summed signal intensity of all masses at any one retention time point.



Spectrum parameters

When masses are detected, it is possible to display them on the spectrum by ticking an option "Show mass list".

Page Contributors

[lalalana5, omokshyna](#)

⌚ October 3, 2022 08:08:05

👤 [lalalana5, omokshyna](#)

12.2.2 Additional tools

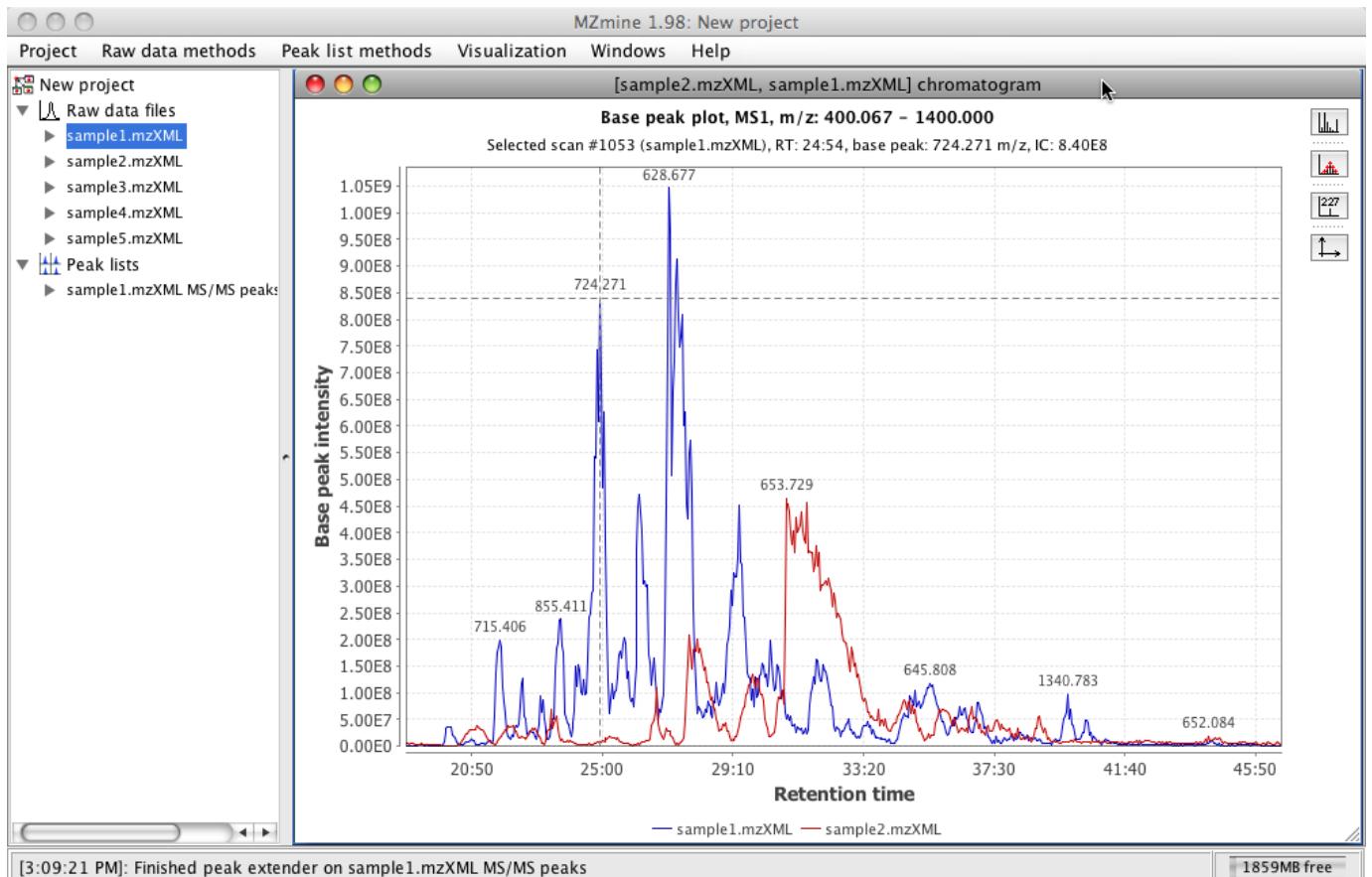
Chromatogram plot

DESCRIPTION

☰ Visualization → Chromatogram plot

There is a possibility to display chromatographic peaks outside of raw data overview. This standalone two-dimensional plot visualizes TIC/XIC data. All the chromatograms are displayed in the same plot.

The x-axis corresponds to retention time and the y-axis is the intensity level of the signal.



PARAMETERS

Raw data files

List of raw data files to display in the TIC visualizer.

MS level

Scan level (MS1,MS2,... ,MSn) to display in the plot.

Plot type

TIC or base peak

Retention time

Retention time (x-axis) range.

m/z range

Range of m/z values. If this range does not include the entire scan m/z range, the resulting visualizer is XIC type.

Selected peaks

List of chromatographic peaks to display in the TIC visualizer. This option is available only if a peak list related to the selected raw data file exists in the current project.

MS spectrum**≡ Visualization → MS spectrum**

Displays all the ions from a selected scan. Can be used to explore mass spectrum outside of raw data overview. **Only one** raw file can be chosen.

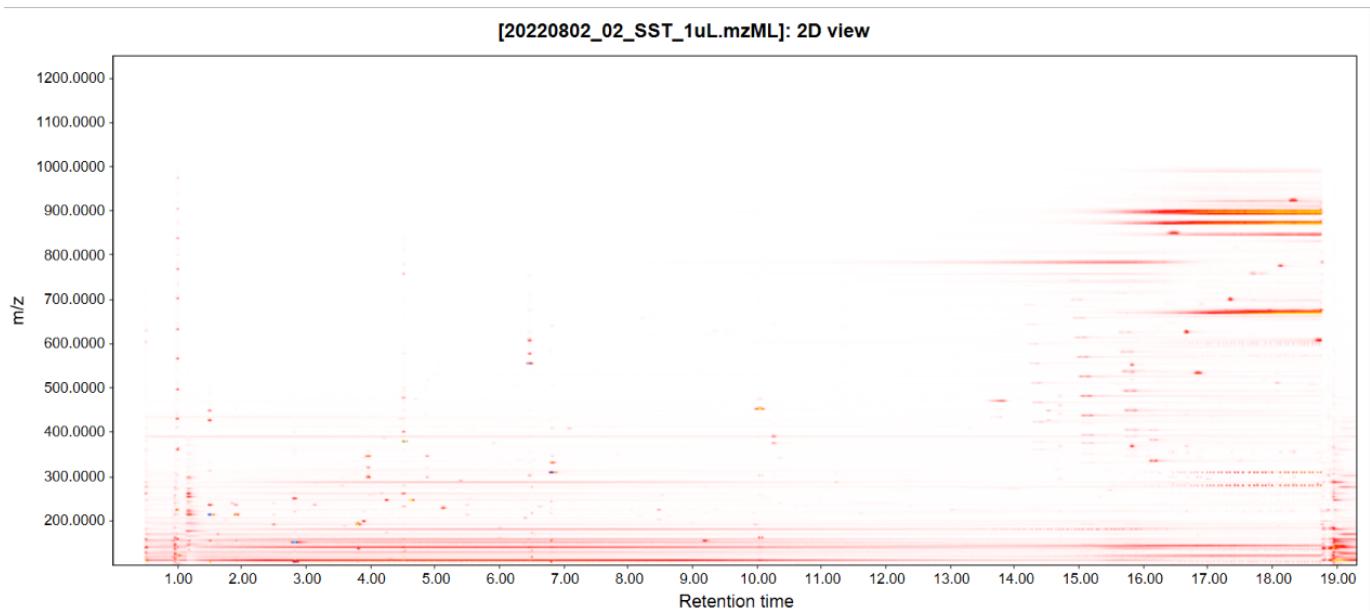
It shows a plot of two dimensions, where X axis corresponds to m/z value and Y axis is the intensity of the ion signal.

PARAMETERS**Scan number**

Choose the scan to visualize

2D visualizer**≡ Visualization → 2D plot**

This tool displays a plot of two dimensions, where X axis corresponds to retention time and Y axis is the m/z value. This visualization of spots in the plot corresponds with the intensity of the data in that region.



User can define features from the feature list to be displayed on the plot.

PARAMETERS**Type of data**

This plot can use either resampled data as input (faster), or raw data (slower).

Scans

Here user can set up the level (MS1,MS2,... ,MSn), polarity, retention time, and the other parameters of the scans to be used.

m/z

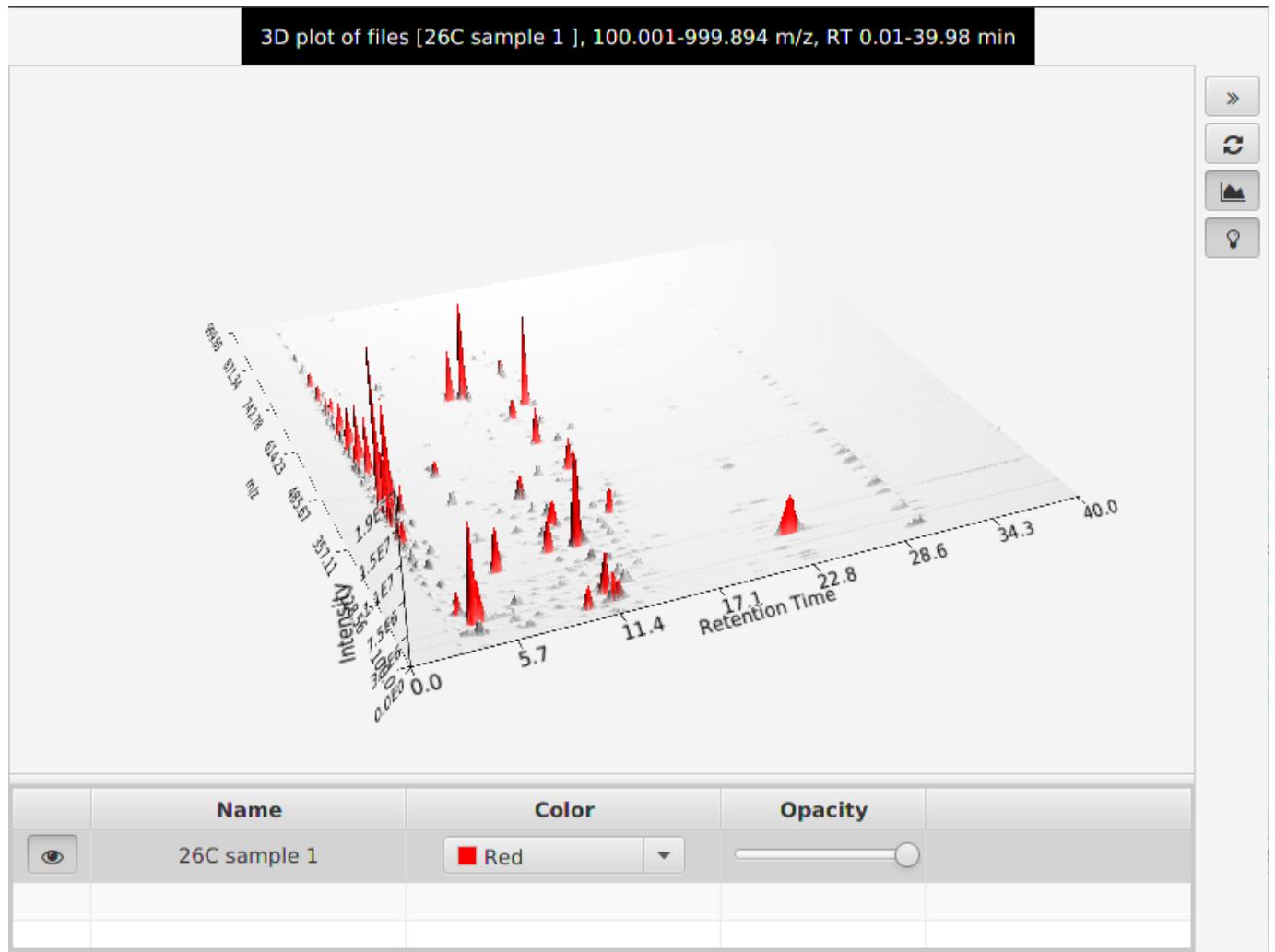
Defines range of m/z values.

3D visualizer

DESCRIPTION

☰ Visualization → 3D plot

This tool presents a three dimensional plot where X axis represents the retention time, Y axis the m/z value and Z axis the intensity of the signal. This plot is the collection of all the information from the raw data in a graphical representation.



PARAMETERS

Scans

Here user can set up the level (MS1,MS2,... ,MSn), polarity, retention time, and the other parameters of the scans to be used.

m/z range

Range of m/z values.

Features

Features that will be taken as an input.

Retention time resolution

Number of data points on retention time axis

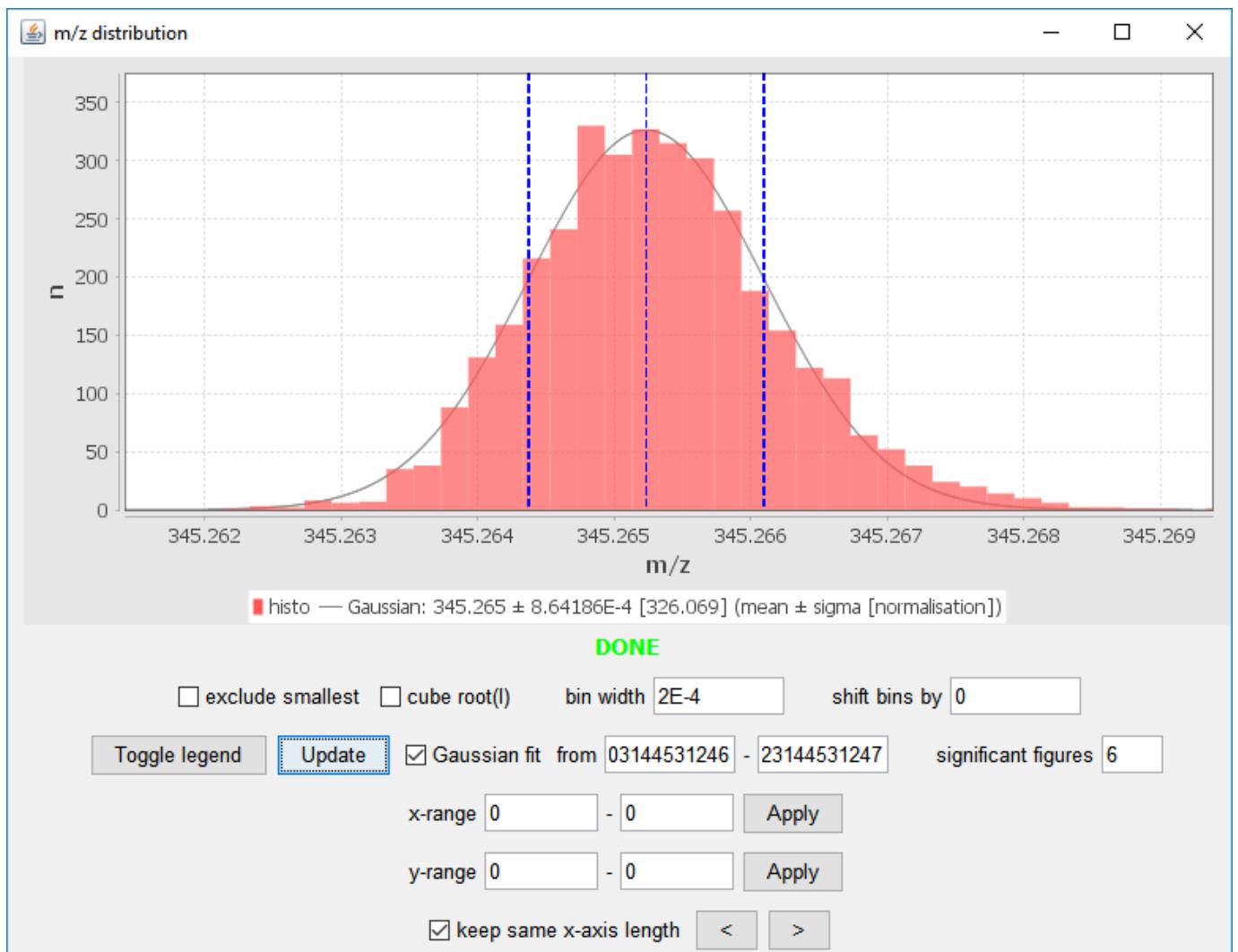
Number of data points on m/z axis

Scan histogram

DESCRIPTION

≡ Visualization → Scan histogram

This module creates m/z distribution histograms of all m/z values in mass lists across specified scans. The binning width, in which the m/z values are counted, can be changed dynamically. The number of scans that contain a specific m/z value (bin) are plotted.



<https://youtu.be/31hwc74vUjA>

PARAMETERS

Scans

Here user can set up the level (MS1,MS2,... ,MSn), polarity, retention time, and the other parameters of the scans to be used.

m/z

Limit the range of the histogram (can improve performance).

Signal intensity range

Optional parameter

Allows to limit signal intensities (can improve performance).

Mass defect

Optional parameter

Filters for mass defects in the signals.

Type

Type of the histogram to be created. Available options:

- m/z,
- Intensity,
- Intensity (noise recalibrated),
- Mass defect.

Bin width

The binning width to count m/z value occurrence in scans.

Use mobility scans

If the input data has ion mobility dimension, this data can be used instead of the data from the summed frames.

Scan inject time analysis**MS(n) spectra tree**

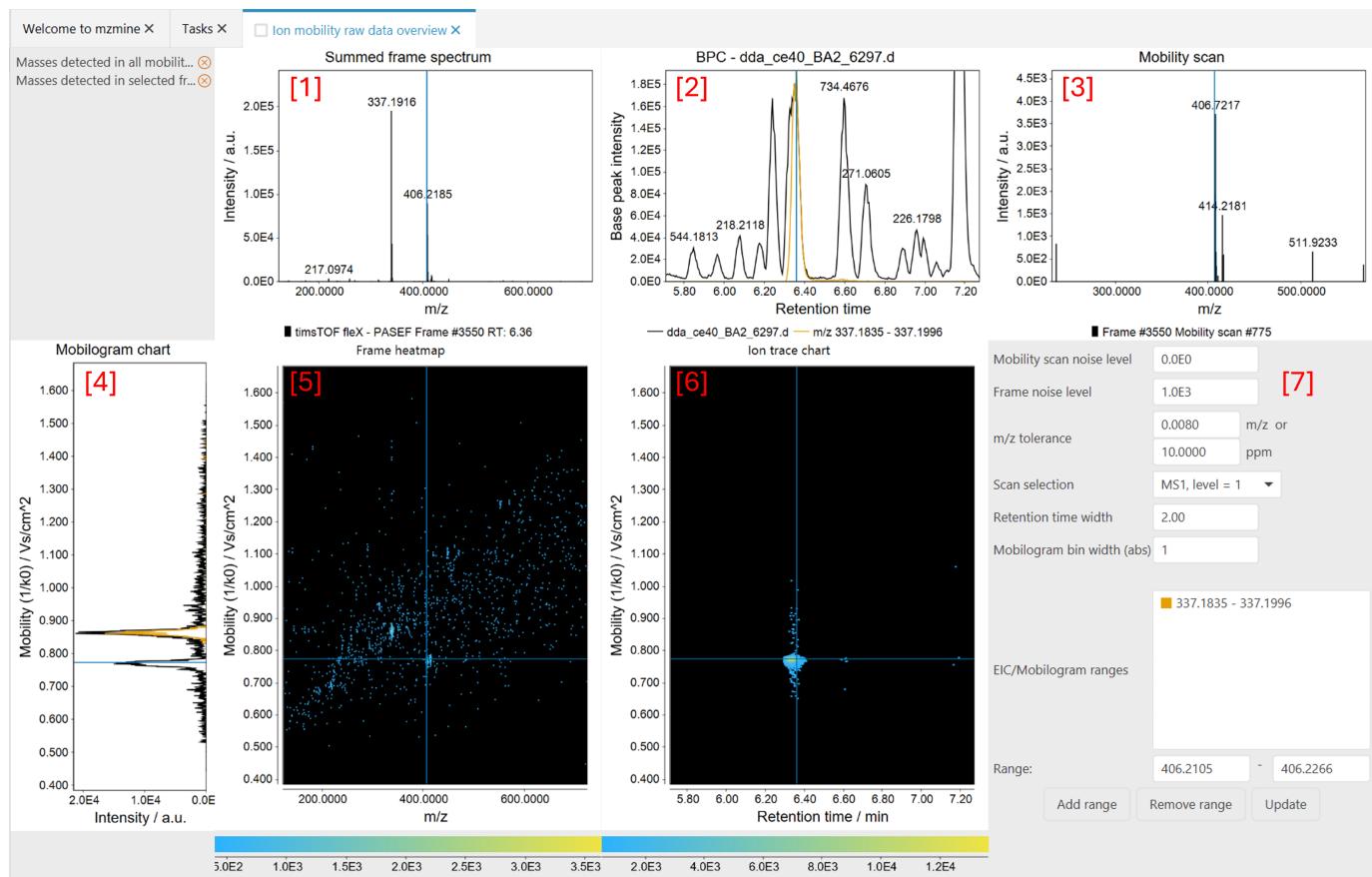
omokshyna

⌚ November 24, 2022 21:13:07

👤 omokshyna

12.3 Ion mobility raw data overview (LC-IMS-MS)

The "Ion mobility raw data visualization" module allow a comprehensive navigation of the complex LC-IM-MS raw data. The screenshot below shows an example of LC-IM-MS data acquired with a Bruker timsTOF instrument:



⚠ Warning

If you don't see any data in the Summed frame [1] or BPC panel [2], the IMS raw data was probably imported from an mzml file. IMS-mzmls do not contain accumulated spectra. Therefore, two processing steps are required, before the data can be displayed: [Mass detection for all mobility scans](#) and [Mobility scan merging](#). Both steps are done automatically, if the file is imported through the mzwizard.

The main window consists of 5 panels and a set of displaying parameters. All the panels are interconnected, which means that moving the cursor in one panel, automatically updates the others. Cursors are displayed as light-blue solid lines in the panels.

12.3.1 Summed frame spectrum panel [1]

The MS spectrum corresponding to each [frame](#) is shown in this panel. The displayed MS spectrum is the sum of all the [mobility scans](#) acquired over that frame (see [Ion mobility spectrometry terminology](#)).

12.3.2 BPC panel [2]

In this panel, the [base peak chromatogram](#) is displayed. Each data point corresponds to an individual [frame](#). Moving the cursor frame-by-frame automatically updates the 'frame heatmap' and 'summed frame spectrum' panels. Moving the cursor frame-by-frame automatically updates the 'summed frame spectrum' panels as changing data point in regular LC-MS data would display a

different MS scan. Since each frame is made of several [mobility scans](#), the 'mobilogram chart' and 'frame heatmap' panels automatically updates too. *Note*. It is currently not possible to display the [TIC chromatogram](#))

Clicking in the chromatogram updates the selected frame.

12.3.3 Mobility scan [3]

Displays the scan at the current selected mobility value. A scan can be selected by clicking in the mobilogram plot, the frame heatmap or the ion trace chart.

Clicking in the spectrum updates the selected m/z value.

12.3.4 Mobilogram chart [4]

Displays the total ion mobilogram at the selected retention time/frame (see frame spectrum) in the colour of the selected raw data file. Additional ranges for mobilograms can be added using the displaying parameters.

Clicking in the mobilogram updates the selected mobility value.

12.3.5 Frame heatmap [5]

Displays the selected frame as a 2D-heatmap. The signal intensity is displayed as a continuous colour scale.

Clicking in the heatmap updates the selected mobility and m/z values.

12.3.6 Ion trace chart [6]

Plots the intensity distribution of the selected m/z value along the retention time (domain) and mobility (range) axes within the currently set **Retention time width** (see [displaying parameters](#))

12.3.7 Displaying parameters [7]

Parameters to adjust the visualisation.

Mobility scan noise level: This parameter controls the signals shown in the [Mobility scan panel \(3\)](#) and the [Frame heatmap \(5\)](#). For example, a noise level of 5.0E1 will remove signals below this intensity.

Frame noise level: This parameter sets a threshold for the signals shown in the [Summed frame spectrum panel](#).

m/z tolerance: A tolerance range for displaying chromatograms, mobilograms, and ion traces of the selected m/z value.

Scan selection: Use this to switch between the selected scans, e.g., MS1 and MS2.

Retention time width: Boundary around the currently selected retention time to build ion traces for the selected m/z.

Mobilogram bin width (abs): Bins the given number of mobility scans together. Achieves a smoother visualisation. May be used to manually investigate the effects of altering the bin width in the [IMS expander](#) to also achieve smoother mobilograms during feature detection.

EIC/mobilogram ranges: Manually add m/z ranges that are visualised constantly. Either define a range, or just define the center of the range and click add. In the latter case, the **m/z tolerance** is used around the given m/z.

⌚ October 31, 2024 12:12:02

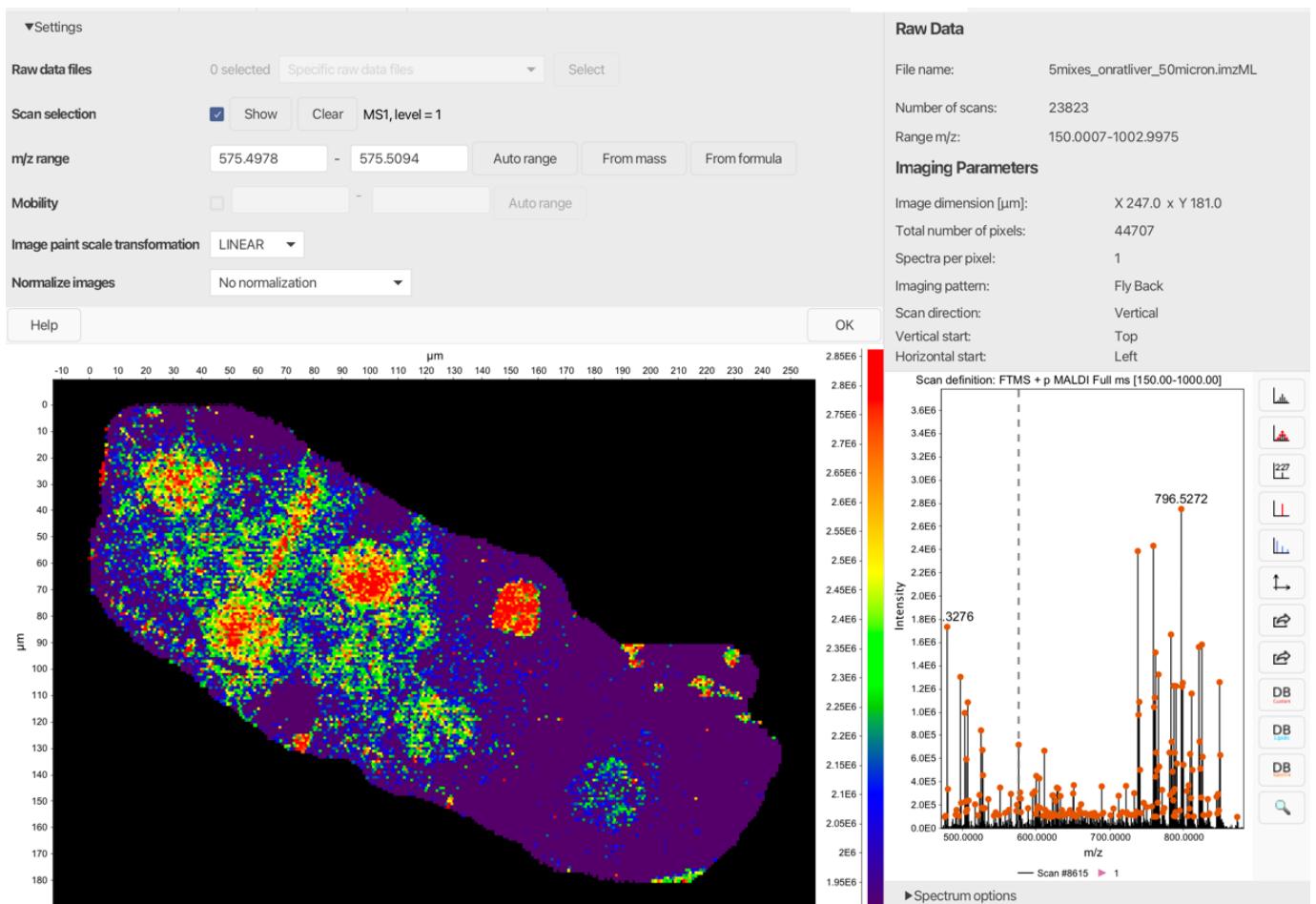
👤 Steffen Heuckeroth, omokshyna, tdamiani

12.4 Image viewer

12.4.1 Description

☰ Visualization → Image viewer

This visualization module provides an overview of imaging data.



The interactive imaging plot (left) allows to choose any pixel and explore the underlying spectrum (bottom right). When selecting a signal from the spectrum, the corresponding image will be generated. In *Spectrum options* menu, the user can choose a feature from a feature list to be depicted on a scan (prior feature detection is required).

The top right panel gives information about the raw file and imaging parameters used for the acquisition.



The Settings menu on top can be used to change the imaging parameters.

12.4.2 Parameters and Settings

▼Settings

Raw data files	5mixes_onratliver_50...	Specific raw data files	▼	Select
Scan selection	<input checked="" type="checkbox"/> Show	Clear	MS1, level = 1	
m/z range	616.1711	-	616.1835	Auto range From mass From form...
Mobility	<input type="checkbox"/>	-		Auto range
Transform paint scale	LINEAR	▼		
Normalize images	No normalization	▼		

The

following parameters have to be set when running the module. They can be adjusted in the image viewer settings menu.

Raw data files

Select raw data files

Scan selection

Select which scans should be shown in the image. Typical filter for imaging experiments is MS level filter, level = 1.

m/z range

Define m/z extraction window.

Mobility

Define mobility extraction window if available.

Transform pain scale

Apply a paint scale transformation for the image plot.

Normalize images

Apply image normalization.

[kyrasolaris01, omokshyna](#)

⌚ January 11, 2024 08:17:33

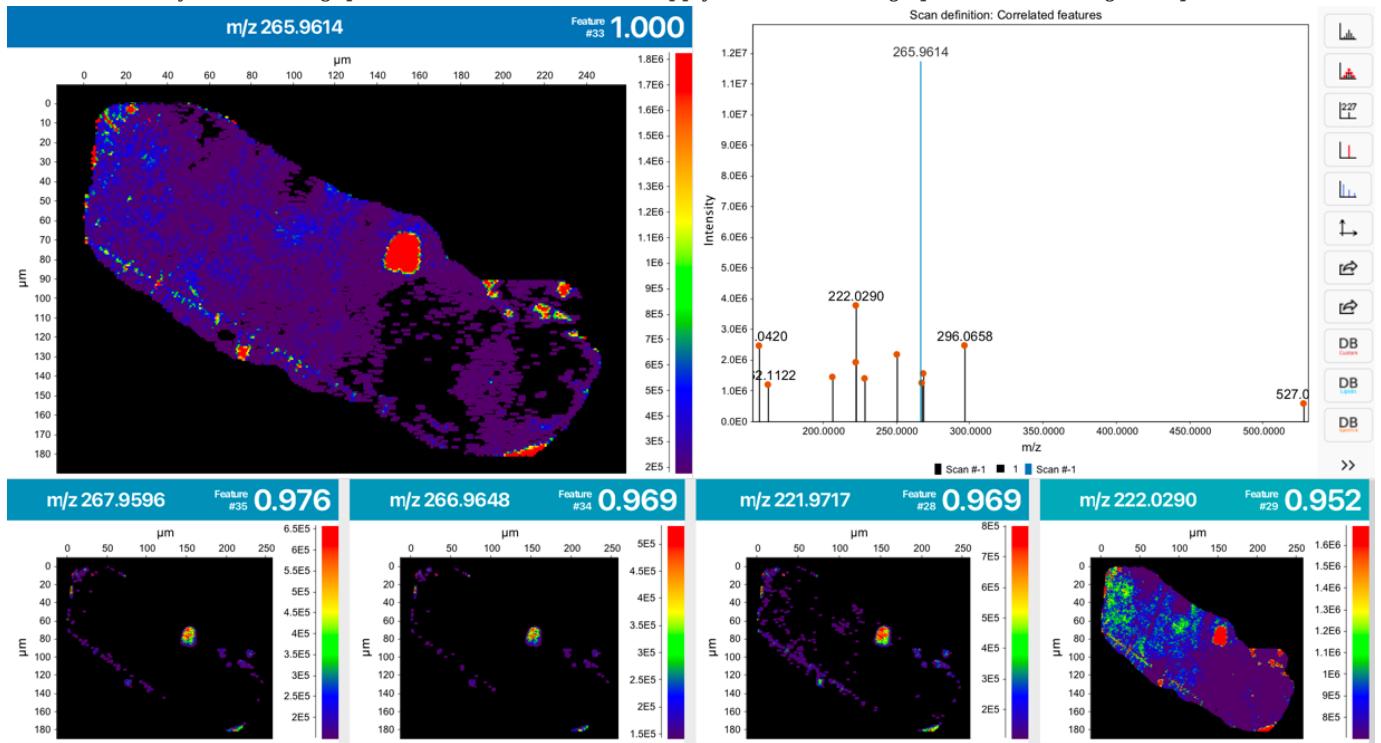
👤 [kyrasolaris01, omokshyna](#)

12.5 Image co-localization

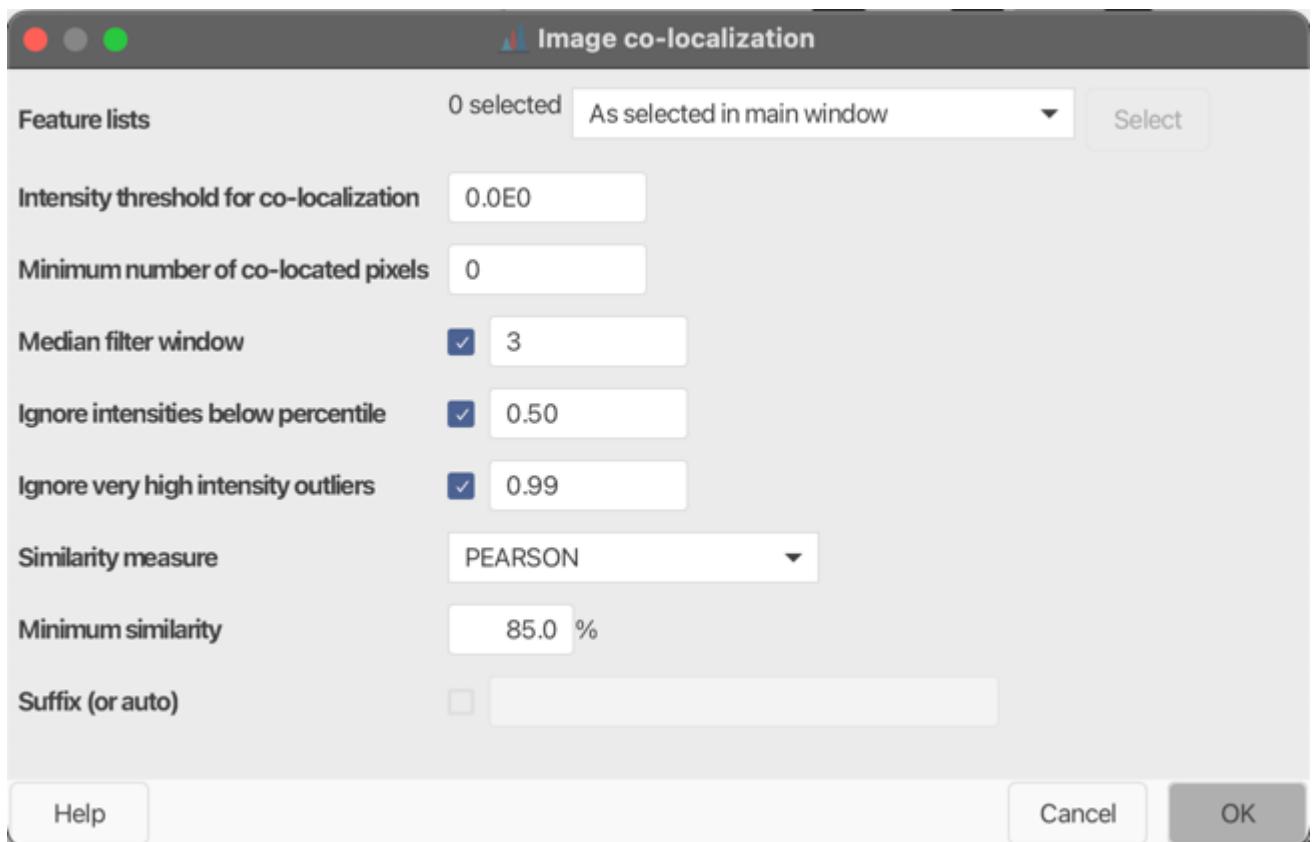
12.5.1 Description

≡ Feature list methods → Feature grouping → Image co-localization.

This module systematically identifies and groups images sharing similar spatial ion distributions. The user can choose between various similarity measure (e.g. pearson correlation) and can apply different filtering options before image comparison.



12.5.2 Parameters



FEATURE LISTS

Select feature list

INTENSITY THRESHOLD FOR CO-LOCALIZATION

This intensity threshold is used to filter data points (pixel) before image co-localization.

MINIMUM NUMBER OF CO-LOCATED PIXELS

Minimum number of locations (pixels) that must be co-located.

MEDIAN FILTER WINDOW

Optional: Smooth over pixels to reduce noise and remove outliers, 3 ist recommended.

IGNORE INTENSITIES BELOW PERCENTILE

Optional: Only consider intensities above the selected percentile, 0.5 is recommended.

IGNORE VERY HIGH INTENSITY OUTLIERS

Optional: Only consider values below the selected percentile, 0.99 is recommended.



Using the optional parameters Median filter window, Ignore intensities below percentile, and Ignore very high intensity outliers may result in better co-localization results.

SIMILARITY MEASURE

Select similarity measure.

MINIMUM SIMILARITY

"Minimum percentage for image similarity to be considered co-located."

SUFFIX (OR AUTO)

Prefix to be added to the feature list name.

[kyrasolaris01](#)

 January 11, 2024 08:37:02

 [kyrasolaris01](#)

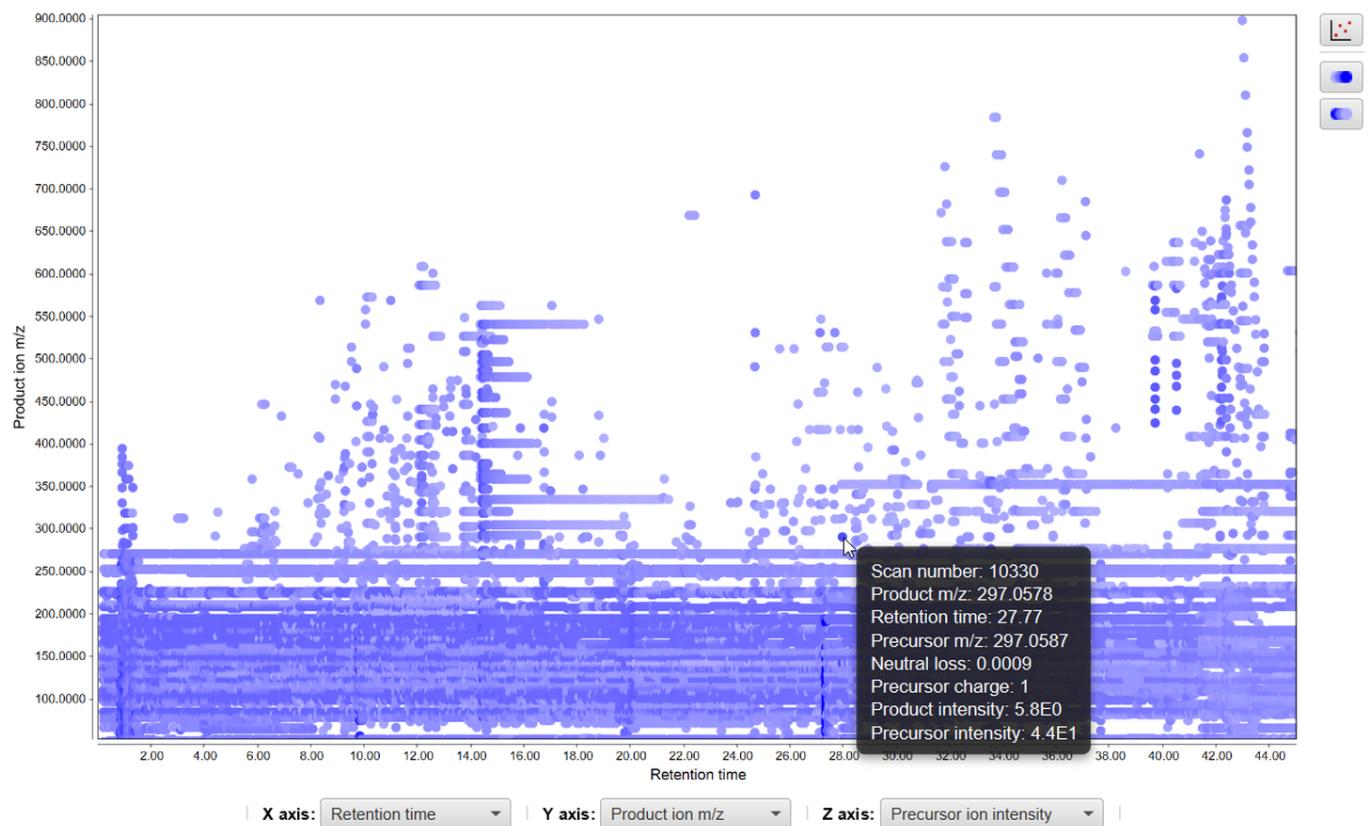
12.6 Processed data visualization

12.6.1 MS/MS plot

Description

☰ Visualization → MS/MS scatter plot

This module provides a colored scatter plot of the MS/MS data. There are 4 options for X and Y axes: retention time, precursor ion m/z, product ion m/z, neutral loss and 3 options for Z axis (color): precursor ion intensity, product ion intensity, retention time. The module additionally allows you to filter ions by their intensities and to perform diagnostic fragmentation filtering. In order to focus on the values of interest you can highlight specific data points and sort them by color axis. This tool can be very useful to get an overview of large amounts of MS/MS data by tuning parameters and filters.



Parameters

Raw data file

Selection of the raw data file to visualize. Only one file can be selected.

X axis

Selection of the values for X axis. There are 4 options available: Retention time, Precursor ion m/z, Product ion m/z, Neutral loss.

Y axis

Selection of the values for Y axis. Options are the same as for X axis.

Z axis

Selection of the values for Z axis. There are 3 options available: Precursor ion intensity, Product ion intensity, Retention time.

MS level

MS level of the scans to be plotted.

Retention time

Retention time range.

m/z range

Range of m/z values for precursor ions in MS_n scans.

m/z tolerance

Maximum allowed difference between two m/z values to be considered same.

Intensities filtering

Optional parameter to filter ions by intensity. There are 3 different ways of filtering:

- Number of best fragments - Number of ions with highest intensities from each scan to be visualized.
For example 5(for each scan 5 ions with highest intensities will be plotted).
- Base peak percent, % - Ions with intensity values lower than the given percent of base peak intensity will be plotted.
For example 95(ions with intensity values lower than 0.95 multiplied by base peak intensity will not be plotted).
- Intensity threshold - Ions having intensities lower than the given value will not be plotted.
For example 6.0E6(ions with intensity values lower than 6.0E6 will not be plotted).

Diagnostic fragmentation filtering

Optional parameter for diagnostic fragmentation filtering described below. It has 2 subparameters: diagnostic product ions and diagnostic neutral loss values. Scans not containing any ion satisfying each input criterion will not be considered for the visualization.

Diagnostic fragmentation filtering

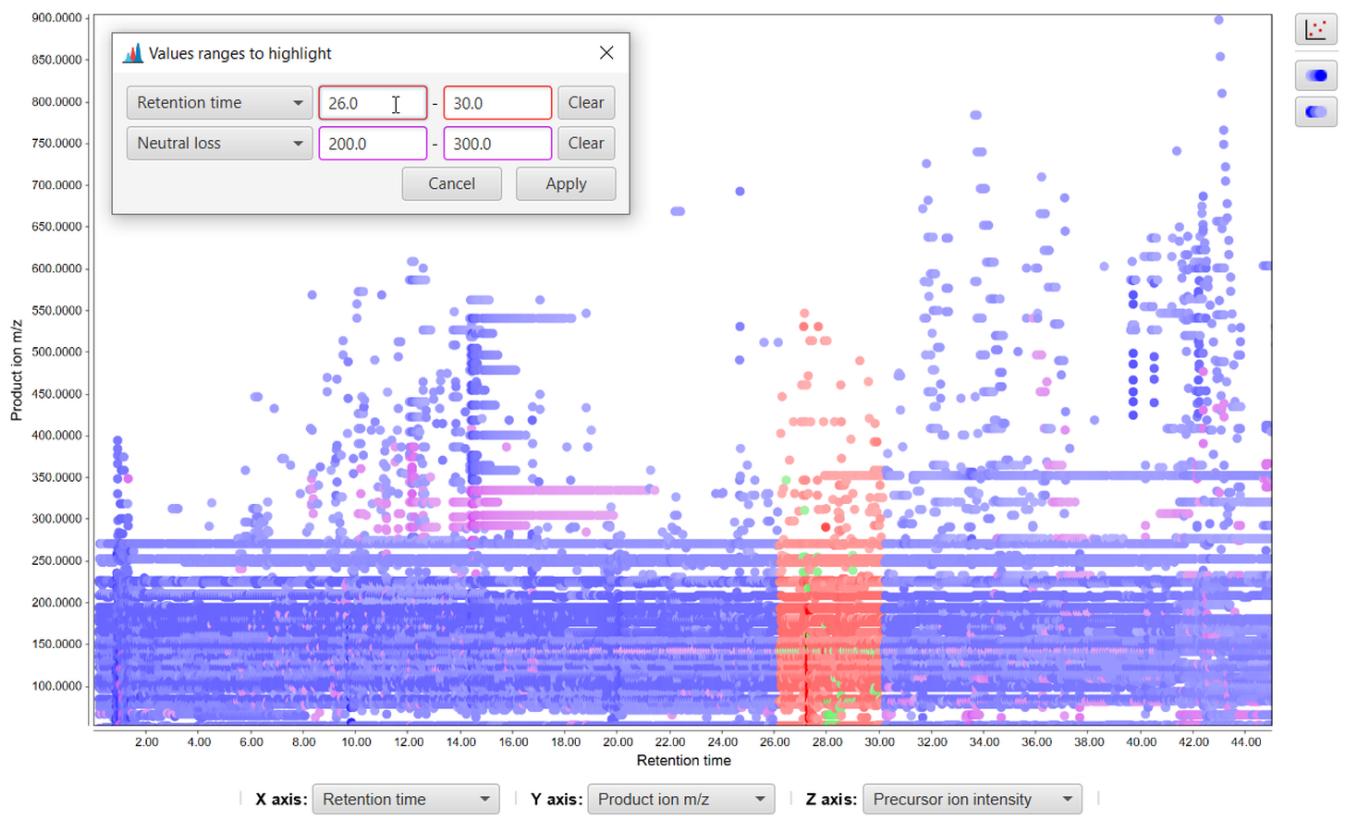
Due to common structural features, compounds within the same class undergo similar MS/MS fragmentation and as a result of many identical product ions and/or neutral losses. Diagnostic fragmentation filter (product ion filter) is a post-acquisition approach to screen LC-MS/MS datasets for entire classes of both known and unknown natural products. This tool searches all MS/MS spectra for product ions and/or neutral losses that has defined as being diagnostic for the entire class of compounds. In other words it screens LC-MS/MS datasets for MS/MS spectra containing production ions and/or neutral losses that are specific to that class of compounds. The user defines the diagnostic product ions and/or the diagnostic neutral loss values (Da) to use in the filtering.

The user can also define the minimum diagnostic ion intensity (% base peak) to use in the filtering. If a recurrent neutral loss occurs, a line pattern in the plot can be observed. If compounds carrying those diagnostic product ions and/or the neutral loss values are detected the resulting plot will show their product ion m/z and precursor ion m/z. Additionally, an output file may be specified that will output the results of the filtering. For a detailed view of diagnostic fragmentation filtering: [Walsh, Jacob P., et al. "Diagnostic Fragmentation Filtering for the Discovery of New Chaetoglobosins and Cytochalasins." Rapid Communications in Mass Spectrometry \(2018\).](#)

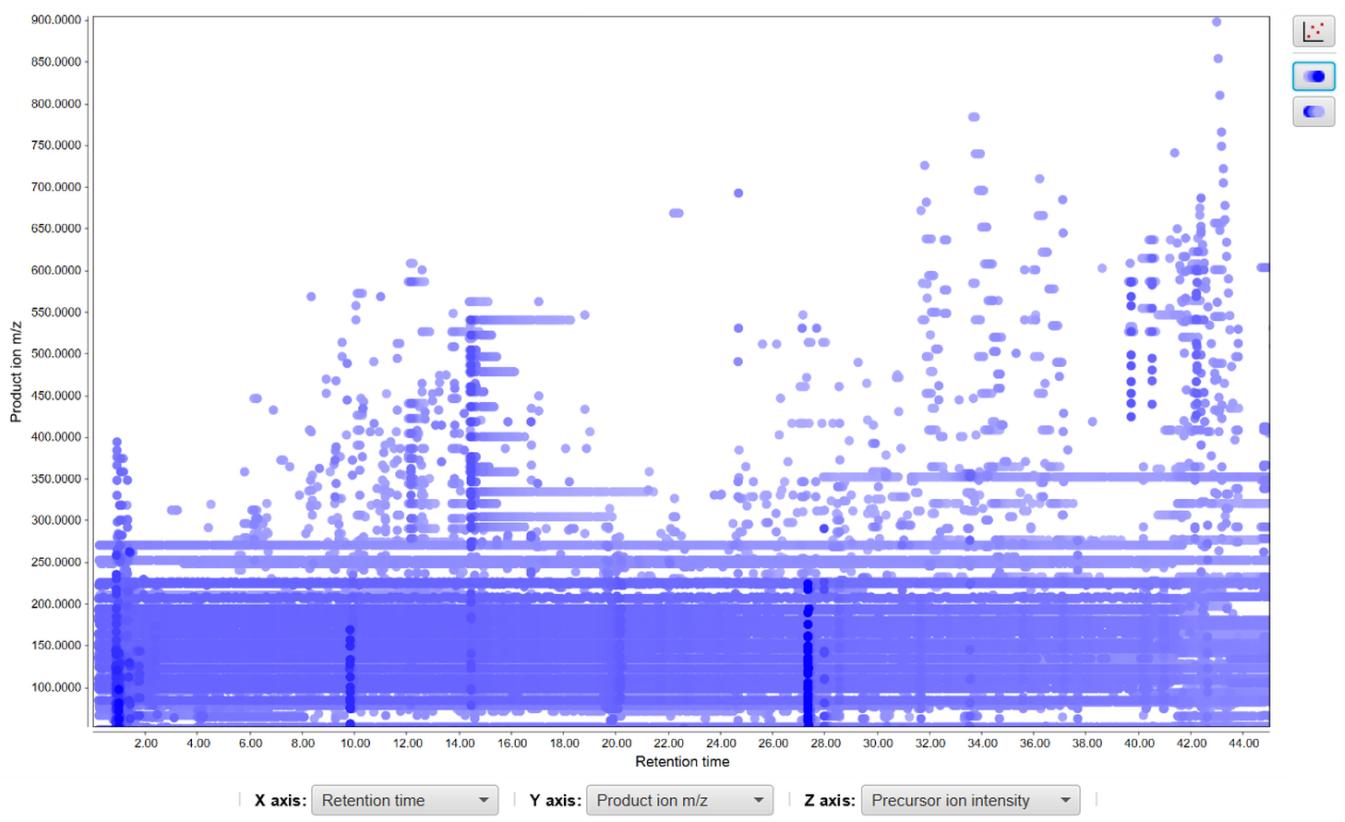
Functionality

This plot is using the third part library JfreeChart for its basic functionality.

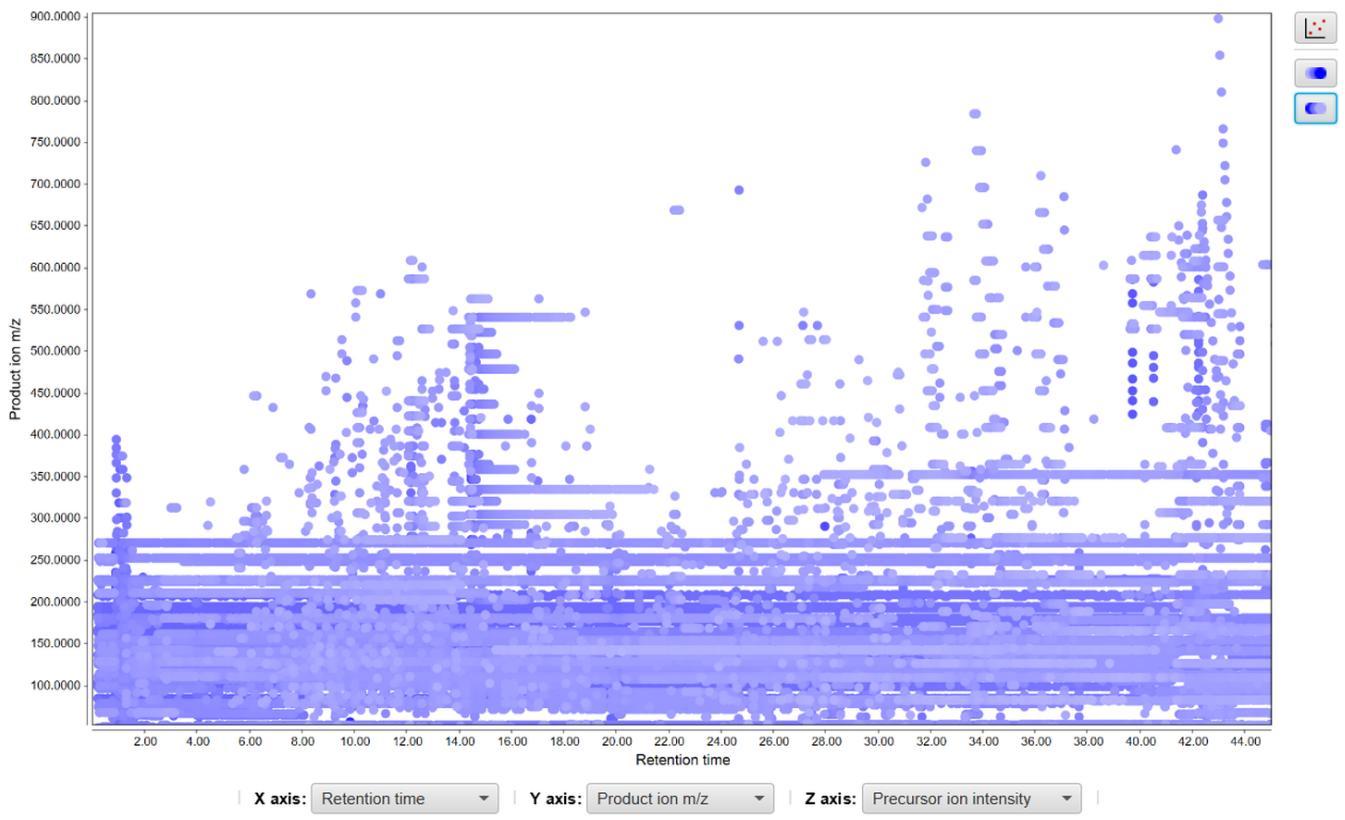
- Drag the mouse from left to right - selecting the area to zoom
- Drag the mouse from right to left - zoom out
- Select combo boxes below - change axes types
- Hold the mouse on data point - show detailed information in a tooltip
- Double click on data point - show spectrum plot
- - highlight points representing ions with specific values given by input ranges (Note: colors of range input boxes determine the highlighting color, green color denotes ions satisfying both ranges)



- - show intense points in front



- show pale points in front



Robin Schmid, Roman Bushuiev, omokshyna

⌚ November 24, 2022 21:13:07

✿✿ Robin Schmid, Roman Bushuiev, omokshyna

12.6.2 Interactive Feature-Based Molecular Networks (FBMN) & Ion Identity Molecular Networks (IIMN)

Description

☰ Visualization → Interactive ion identity molecular networks

Molecular Networking is available in mzmine or through other tools (GNPS, matchms) and connects mass spectra of molecules based on the similarity of their fragmentation patterns. However, during ionization, molecules commonly form multiple ion species with different fragmentation behavior. As a result, the fragmentation spectra of these ion species often remain unconnected in tandem mass spectrometry-based molecular networks, leading to redundant and disconnected sub-networks of the same compound classes.

To overcome this bottleneck, MZmine employs **Ion Identity Molecular Networking (IIMN)** module. This module unites chromatographic peak shape correlation analysis with molecular networks, which is able to connect and collapse different ion species of the same molecule. This feature relationships improve network connectivity for structurally related molecules.

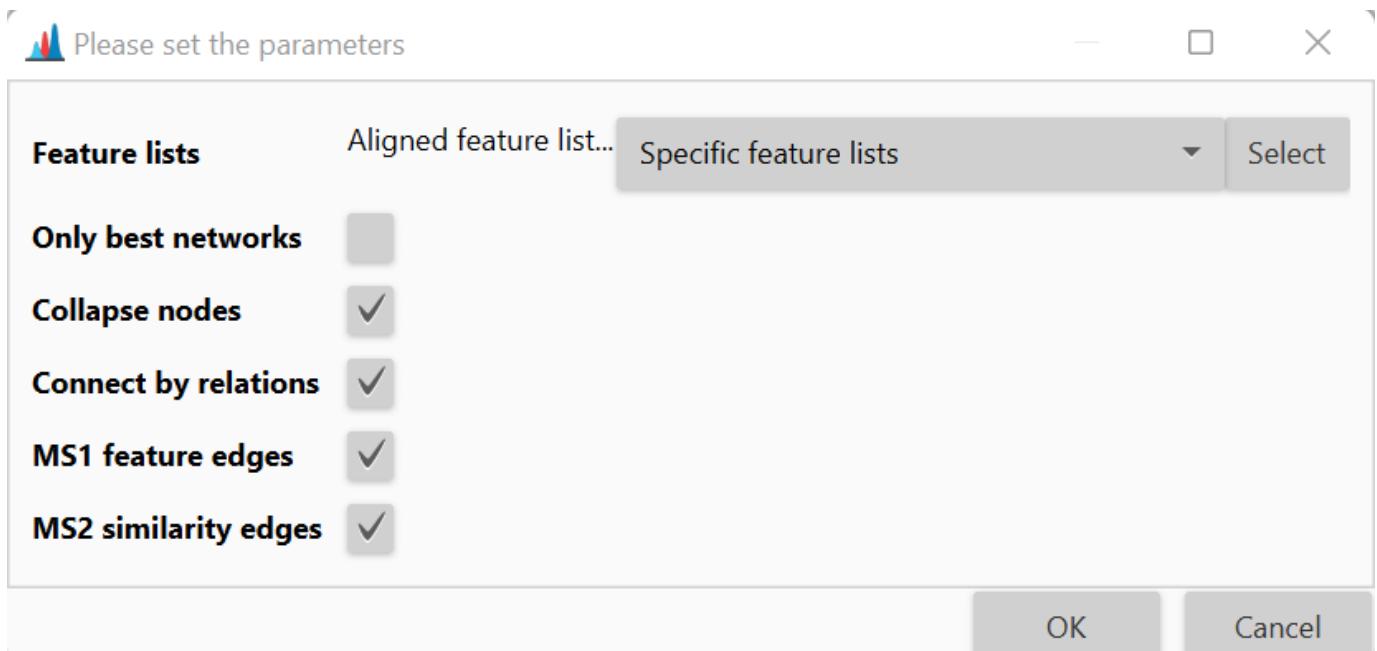
This module can be used to reveal unknown ion-ligand complexes, enhance annotation within molecular networks, and facilitate the expansion of spectral reference libraries.

💡 Using IIMN it is possible to query [GNPS](#) and other spectral libraries for matches.

More detailed description of theory behind IIMN can be found in the following video:



Parameters



Only best networks

Only the networks that only contain first ranked ion identities for all rows.

Collapse nodes

Collapse all nodes into neutral molecule nodes.

Connect by relations

Connect neutral molecule nodes by network relations.

MS1 feature edges

Include feature correlation edges.

MS2 similarity edges

Show MS2 similarity edges.

[Robin Schmid, omokshyna](#)

 April 22, 2025 14:31:25

 [Robin Schmid, omokshyna](#)

12.6.3 Histogram plot

Description

≡ Visualization → Histogram plot

This plot displays a graphic representation of frequencies. Each rectangle represents an interval of frequency. The height is also equal to the frequency density in that interval. The total area of the histogram is equal to the number of data. This tool can use the m/z value, height, area or retention time as frequency value (X axis) and number of peaks in each interval (Y axis).

Parameters

Raw data files

Column of peaks to be plotted.

Plotted data type

Peak's data to be plotted (m/z value, height, area or retention time)

Plotted data range

Range of data to be plotted. This range is automatically loaded with max values from the raw data.

Number of bins

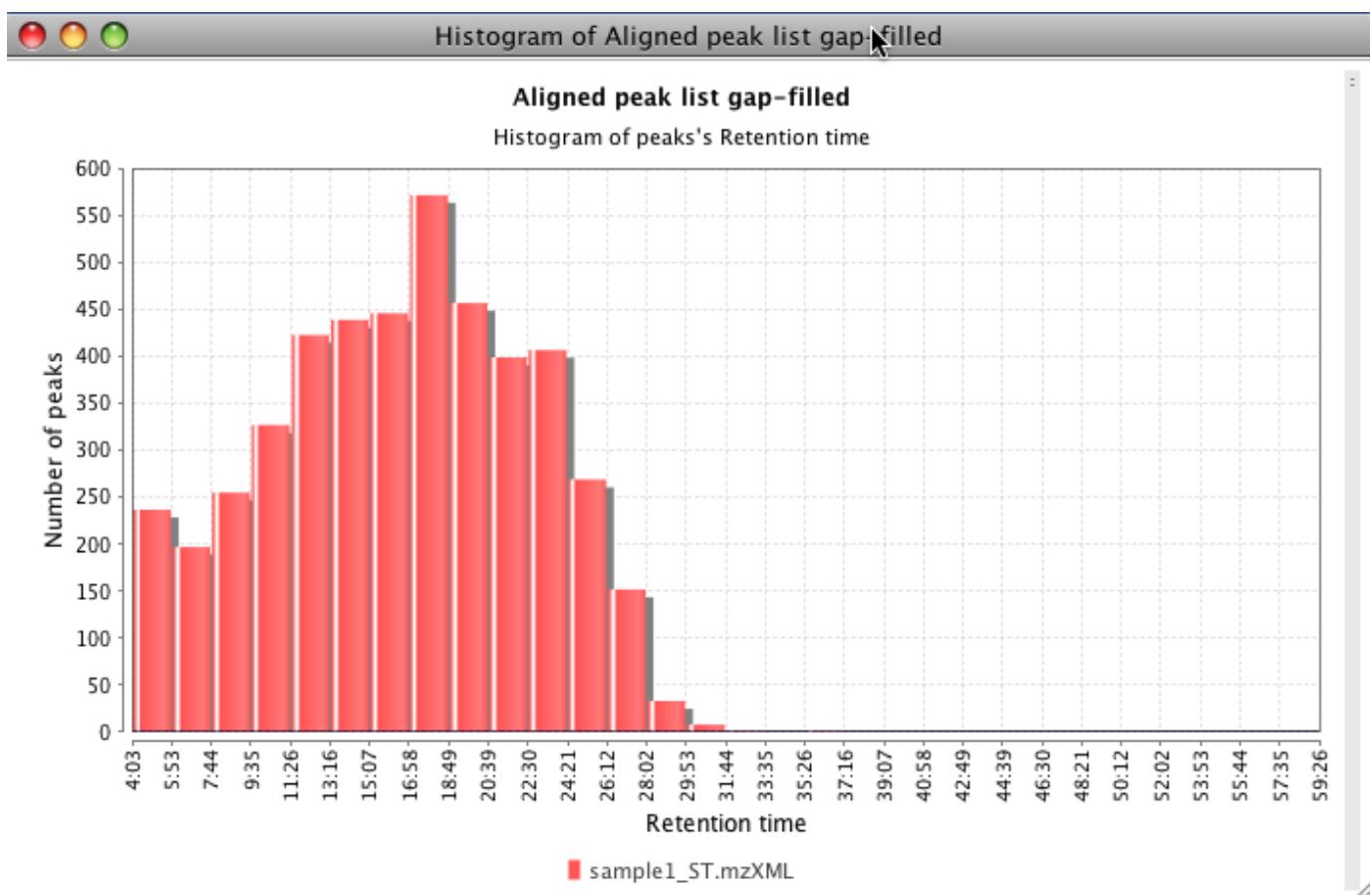
The plot is divides into this number of bins.

Functionality

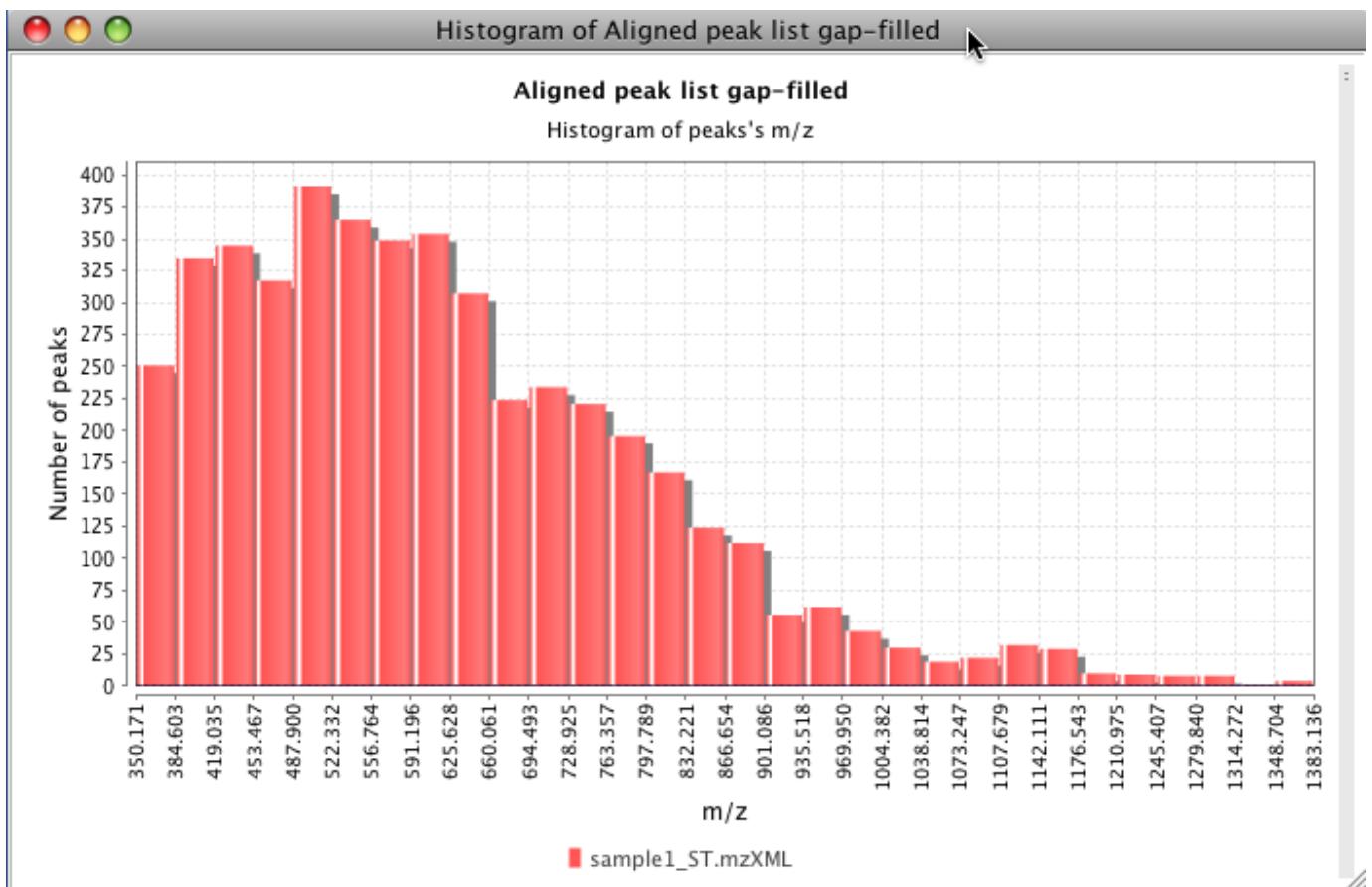
This plot is using the third part library JfreeChart for its basic functionality.

To zoom in, drag the mouse from left to right, selecting the area to zoom. To zoom out drag the zoom from right to left.

The next figure shows a histogram using the retention time value. This data is coming from a raw data with a duration of 60 min. Most of the peaks appears around 17 min.



The next figure shows a histogram using the m/z value. This data is coming from a raw data with a range from 350 to 1400 m/z.



omokshyna

⌚ November 24, 2022 21:13:07

👤 omokshyna

12.6.4 4D feature plot (Kendrick mass plot)

Description

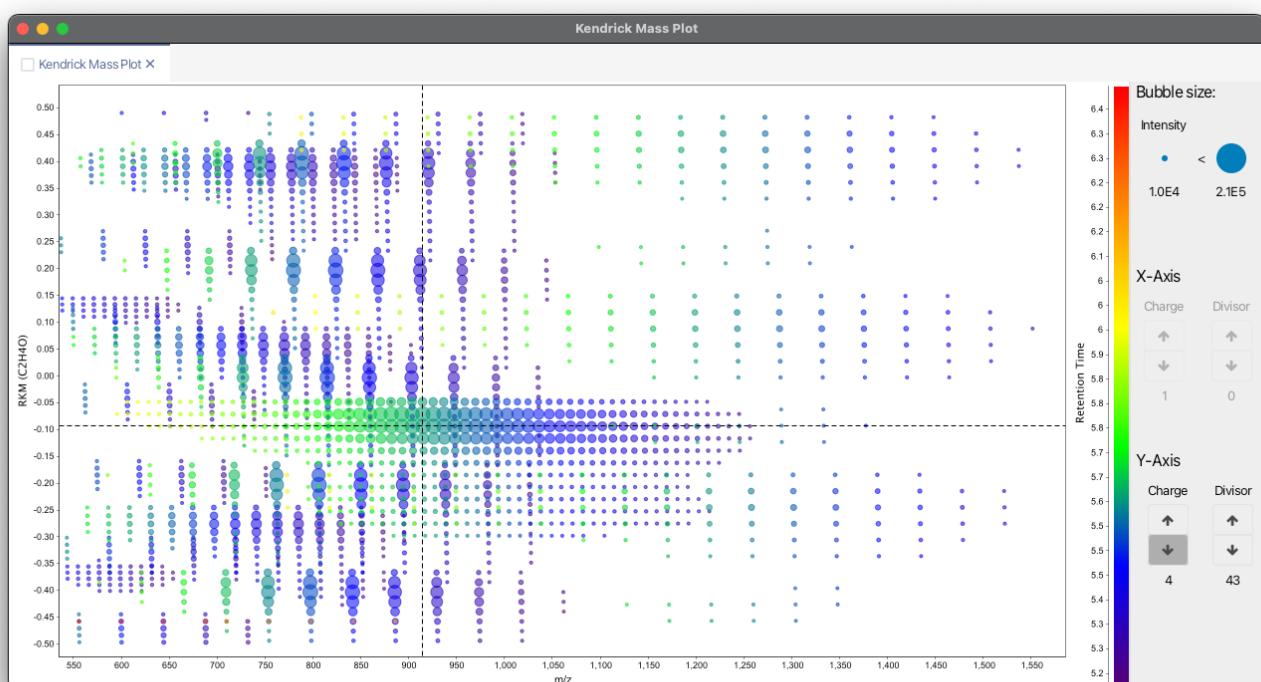
☰ Visualization → 4D feature plot (Kendrick)

In 1963 Kendrick published his idea of a mass scale, the so-called Kendrick mass scale, which is based on defining the mass of CH₂ as 14.0000 u.

The Kendrick mass scale is calculated by multiplying the IUPAC mass scale with the factor 14.0000 u/14.01565 u = 0.9988834. This results in the same mass defect for homologous components, the so-called **Kendrick mass defect (KMD)**. The KMD is defined as the $\langle(\Delta)\rangle$ of a nominal Kendrick mass and its associated Kendrick mass. Using the Kendrick mass scale has the purpose of data reduction.

$$\begin{aligned} \text{KM}(R) &= m/z \cdot \text{cdot} \cdot \text{frac}\{\text{round}(R)\}{R}\} \\ \text{KMD}(R) &= \text{round}(\text{KM}(R)) - \text{KM}(R) \end{aligned}$$

where $\langle KM \rangle$ is Kendrick mass, $\langle KMD \rangle$ - Kendrick mass defect, $\langle R \rangle$ - exact mass of selected base unit.



If you use this module for your analysis or visualization, please cite:

Three-dimensional Kendrick mass plots as a tool for graphical lipid identification. A. Korf, C. Vosse, R. Schmid, P. O. Helmer, V. Jeck, H. Hayen, *Rapid Communications in Mass Spectrometry* 32.12 (2018): 981-991.

Expanding the Kendrick Mass Plot Toolbox in MZmine 2 to Enable Rapid Polymer Characterization in Liquid Chromatography–Mass Spectrometry Data Sets A. Korf, T. Fouquet, R. Schmid, H. Hayen, and S. Hagenhoff, *Analytical Chemistry* 2020 92 (1), 628-633.

Detailed functionality

This module allows to create 4 dimensional Kendrick mass plots. Feature characteristics, such as retention time or feature area, can be plotted as color scale or bubble size.

CHARGE DEPENDENT KENDRICK MASS PLOTS

Multiply charged ions can cause splits in Kendrick mass plot. Fouquet et al. have shown considering the charge for the calculation of the KM can help to overcome this problem through clustering of features.

$$[KM(R,Z)=Z\cdot KM(R)=Z\cdot m/z\cdot \frac{1}{\text{round}(R)}\{R\}]$$

where $\{Z\}$ is charge.

Fouquet, Thierry NJ, et al. "On the Kendrick Mass Defect Plots of Multiply Charged Polymer Ions: Splits, Misalignments, and How to Correct Them." Journal of The American Society for Mass Spectrometry 29.8 (2018): 1611-1626.

RESOLUTION ENHANCED KENDRICK MASS DEFECT PLOTS

Fouquet and Sato have shown that a fractional base unit (visor) can enhance the resolution of Kendrick mass plots.

$$[X>1, KM(R,X)=m/z\cdot \frac{1}{\text{round}(R/X)}\{R/X\}]$$

where $\{X\}$ is a fractional base unit.

Fouquet, Thierry, and Hiroaki Sato. "Extension of the Kendrick mass defect analysis of homopolymers to low resolution and high mass range mass spectra using fractional base units." Analytical chemistry 89.5 (2017): 2682-2686.

COMBINING CHARGE AND FRACTIONAL BASE UNIT (DIVISOR)

If both charge and fractional base unit are changed, the following equation is used:

$$[KM(R,Z,X)=Z\cdot KM(RX)=Z\cdot m/z\cdot \frac{1}{\text{round}(R/X)}\{R/X\}]$$

Fouquet, Thierry NJ, et al. "On the Kendrick Mass Defect Plots of Multiply Charged Polymer Ions: Splits, Misalignments, and How to Correct Them." Journal of The American Society for Mass Spectrometry 29.8 (2018): 1611-1626.

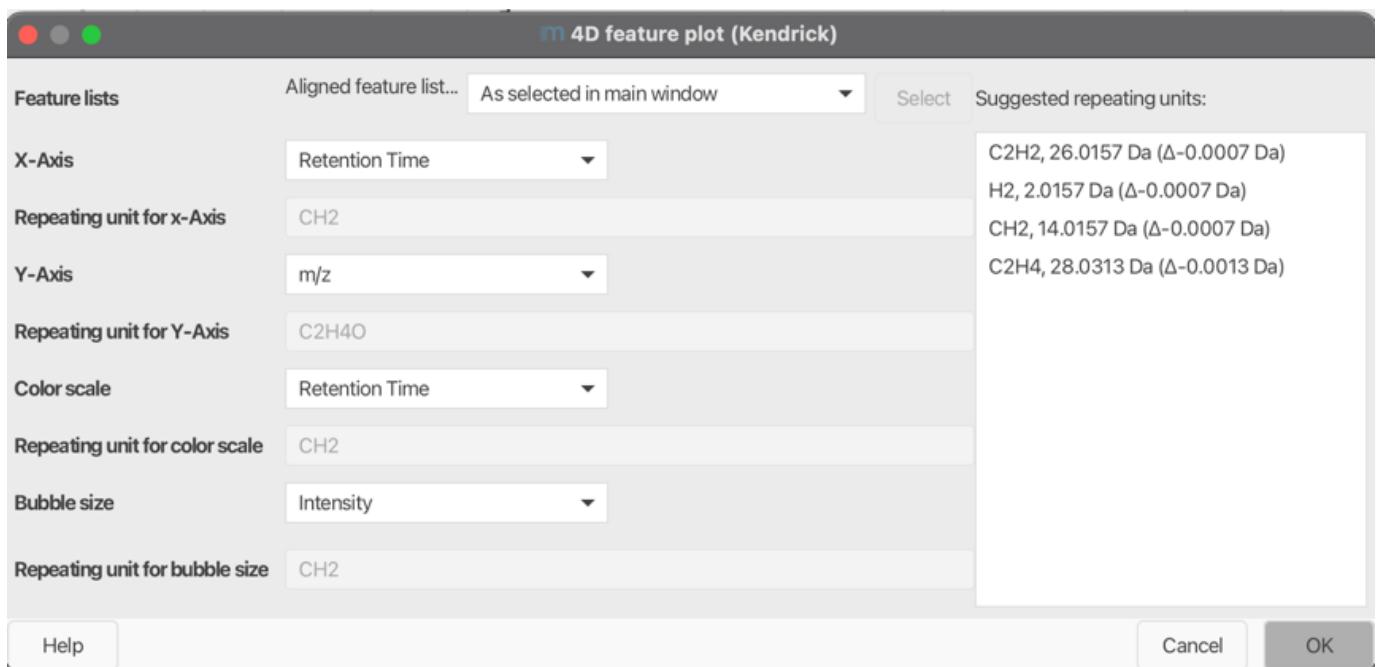
Fouquet, Thierry, Takaya Satoh, and Hiroaki Sato. "First gut instincts are always right: the resolution required for a mass defect analysis of polymer ions can be as low as oligomeric." Analytical chemistry 90.4 (2018): 2404-2408.

REMAINDERS OF KENDRICK MASSES (RKM)

Another option to increase the resolution of Kendrick mass plots is the by Fouquet et al. proposed concept of RKM (remainders of Kendrick masses). $[RKM(R)=\{\text{frac}\{\text{KM}(R)\}\{\text{round}(R)\}\}]$

with $\{\cdot\}$ being the fractional part function defined as $\{(x=x-\text{floor}(x))\}$

Parameters



Feature List

Select one feature or aligned feature list.

X-axis

Select which metric you want to display on the X-Axis. Possible metrics are m/z, Kendrick Mass, Kendrick Mass Defect, Remainder of Kendrick Mass, Retention Time, Mobility, Intensity, Area, Tailing Factor, and Asymmetry Factor.

Repeating unit for X-axis

Enter a molecular formula which will be used for Kendrick Mass calculation. The field is only active if Kendrick, Kendrick Mass Defect, or Remainder of Kendrick Mass is selected.

Y-axis

Select which metric you want to display on the Y-Axis. Possible metrics are m/z, Kendrick Mass, Kendrick Mass Defect, Remainder of Kendrick Mass, Retention Time, Mobility, Intensity, Area, Tailing Factor, and Asymmetry Factor.

Repeating unit for Y-axis

Enter a molecular formula which will be used for Kendrick Mass calculation. The field is only active if Kendrick, Kendrick Mass Defect, or Remainder of Kendrick Mass is selected.

Color scale

Select which metric you want to display as a color scale. Possible metrics are m/z, Kendrick Mass, Kendrick Mass Defect, Remainder of Kendrick Mass, Retention Time, Mobility, Intensity, Area, Tailing Factor, and Asymmetry Factor.

Repeating unit for color scale

Enter a molecular formula which will be used for Kendrick Mass calculation. The field is only active if Kendrick, Kendrick Mass Defect, or Remainder of Kendrick Mass is selected.

Bubble size

Select which metric you want to display as bubble size. Possible metrics are m/z, Kendrick Mass, Kendrick Mass Defect, Remainder of Kendrick Mass, Retention Time, Mobility, Intensity, Area, Tailing Factor, and Asymmetry Factor.

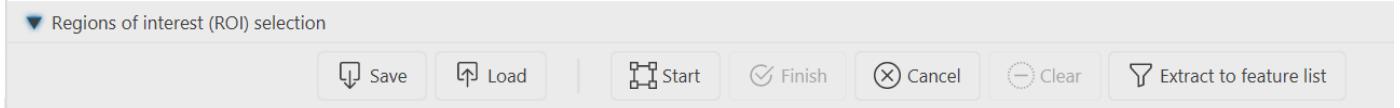
Repeating unit for bubble size

Enter a molecular formula which will be used for Kendrick Mass calculation. The field is only active if Kendrick, Kendrick Mass Defect, or Remainder of Kendrick Mass is selected.

Region of interest ROI extraction

The Kendrick/4D plot can be used to extract regions of interest from a plot. E.g., you can use this to extract polymers, PFAS, Lipids and more depending on their localization in a m/z vs. RT plot or in a Kendrick plot.

1. Find the "Region" panel below the kendrick plot

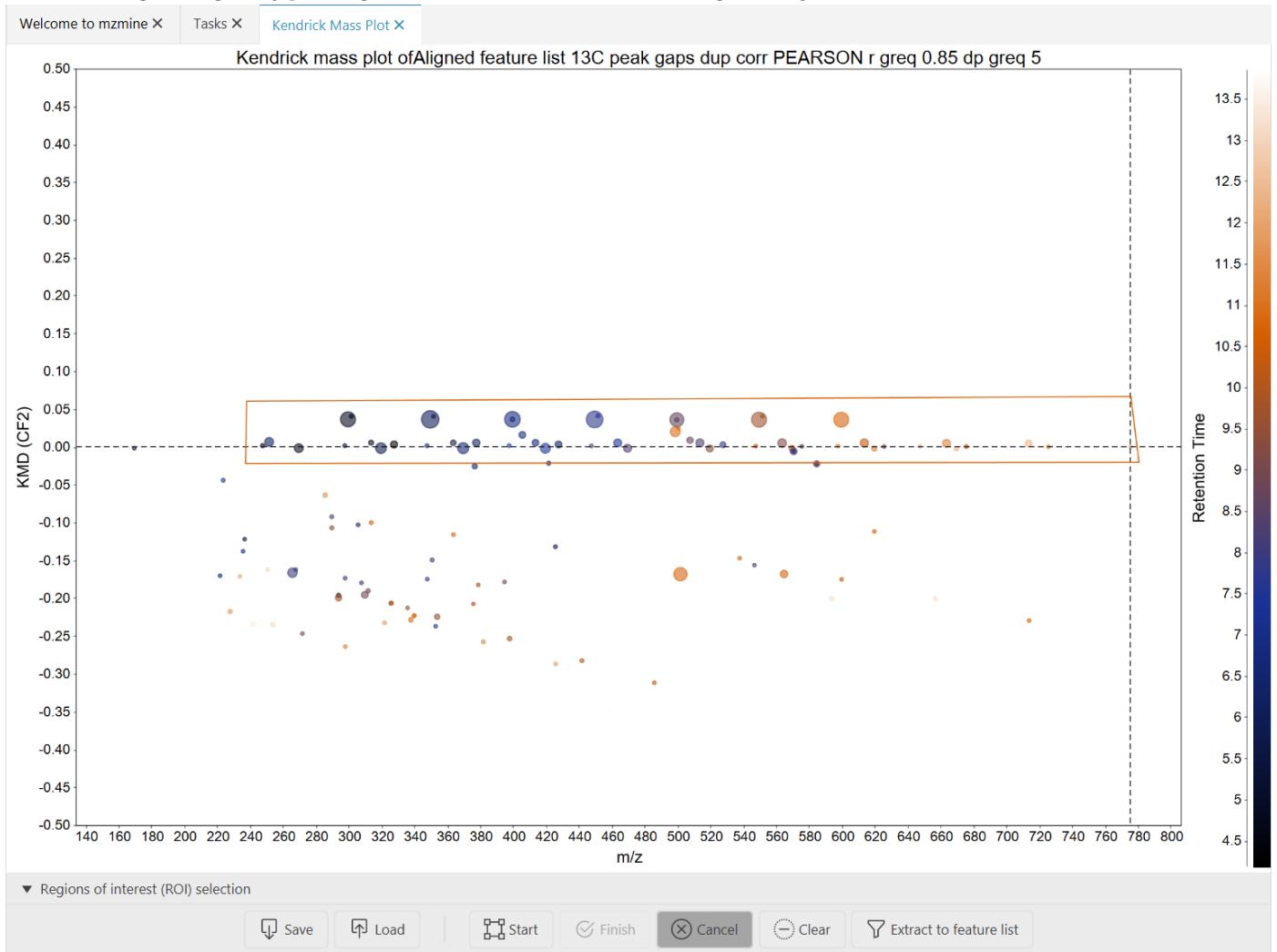


2. Start drawing by pressing the "Start" button.

3. Draw your region of interest in the plot. Unfinished ROIs will be filled in gray.



4. Finish drawing the region by pressing the "Finish" button. The finished region will just be outlined.



5. Click the "Extract to feature list" button to create a new feature list of features in this region.

!!! Note: The chart supports multiple regions. Region drawing can be canceled, regions can be removed or loaded using the respective buttons.

!!! Batch mode: To use the ROI extraction in the batch mode, set up the desired ROI using the 4D chart and save the region or extract it. You can either save the whole batch from the feature list summary or load the regions into the 'Region extraction' module.

Automatic Repeating Unit Suggestion

DESCRIPTION

The Repeating Unit Suggester in mzmine is designed to automatically predict repeating units based on the m/z values of features in a feature list. This functionality is crucial for simplifying the identification of repeating units in polymer analysis, lipid identification, and other applications requiring pattern recognition in mass spectrometry data.

FUNCTIONALITY

The Repeating Unit Suggester performs the following steps to suggest repeating units. The results are displayed on the right side of the [parameter setup dialog](#):

1. **Extract m/z Values:** The m/z values are extracted from the provided feature list, taking into account the detected charge states of the features, if available (default charge = 1).
2. **Calculate Delta Frequencies:** The frequency of all m/z deltas is calculated to identify common mass differences between features.
3. **Identify Top Deltas:** The top N delta medians are identified based on their frequency, which helps in pinpointing the most common repeating units.
4. **Filter Multimers:** The deltas are filtered to remove multimers, ensuring that only the most relevant repeating units are considered.
5. **Predict Formulas:** Potential molecular formulas corresponding to the identified deltas are predicted. This involves:
 - Generating molecular formulas.
 - Applying heuristic checks (RDBE, elemental, and nitrogen rule) to validate the predicted formulas.

 October 15, 2024 19:11:13

 Ansgar Korf, Steffen Heuckeroth

12.6.5 Van Krevelen diagram

DESCRIPTION

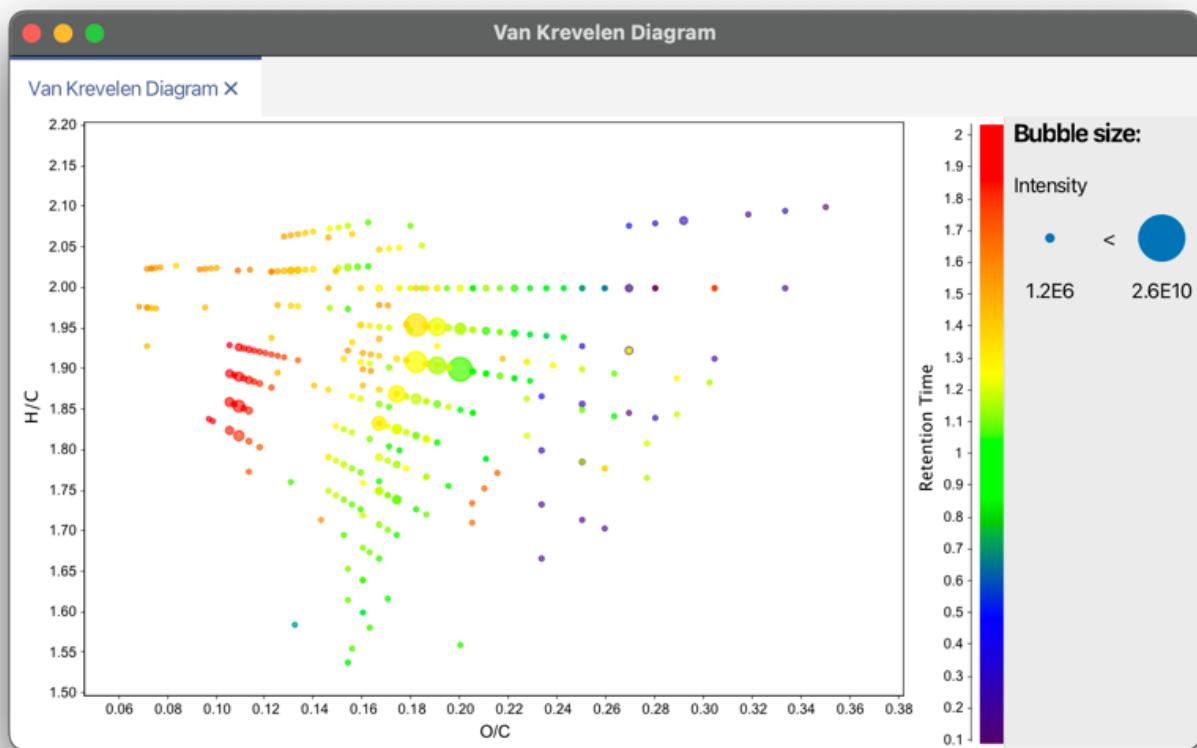
Van Krevelen diagrams are graphical plots developed by Dirk Willem van Krevelen (chemist and professor of fuel technology at the TU Delft) that are used to assess the origin and maturity of kerogen and petroleum.

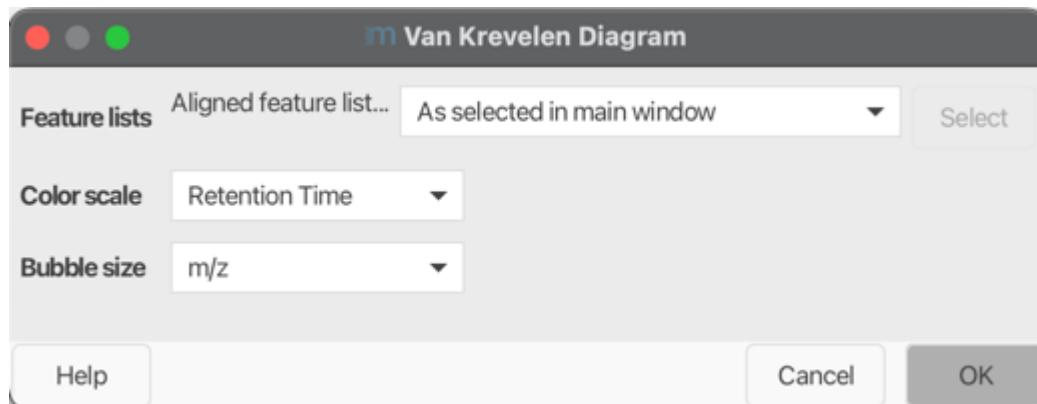
The diagram cross-plots the hydrogen:carbon (hydrogen index) as a function of the oxygen:carbon (oxygen index) atomic ratios of carbon compounds.

1. Van Krevelen, D.W. (1950). "Graphical-statistical method for the study of structure and reaction processes of coal", Fuel, 29, 269-84
2. https://en.wikipedia.org/wiki/Van_Krevelen_diagram



Since element ratios are plotted only annotated features are considered.



PARAMETERS**Feature lists**

Select feature list to display as Van Krevelen diagram.

Color scale

Select which parameters you want to display as a color scale.

Bubble size

Select which parameters you want to display as bubble size.

[Ansgar Korf, Steffen Heuckeroth](#)

⌚ October 15, 2024 19:17:27

👤 Ansgar Korf, Steffen Heuckeroth

12.6.6 ECN (Equivalent Carbon Number) models

⚠ Warning

This module requires lipid annotations from the Lipid Annotation module.

Description

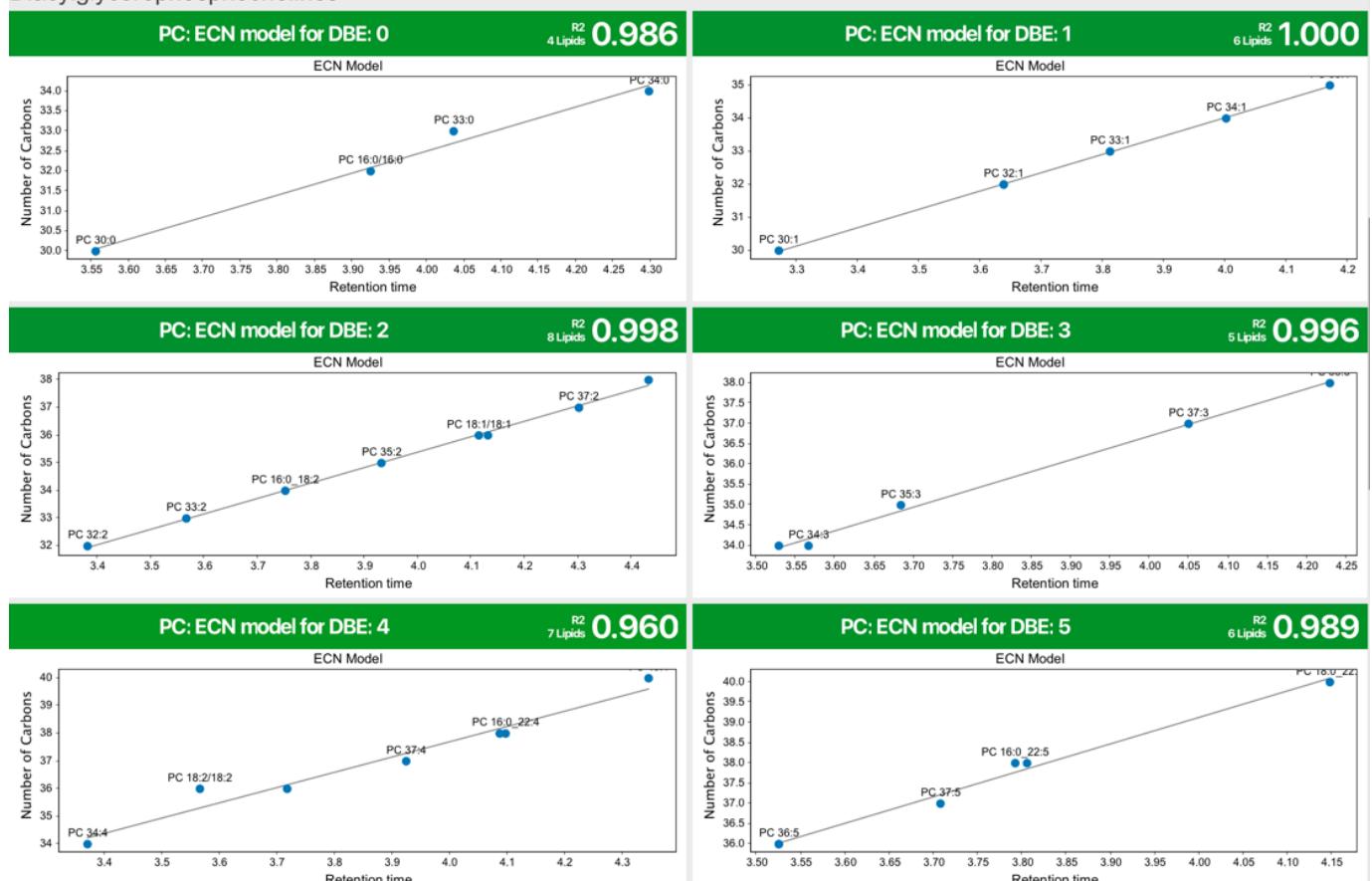
☰ Visualization → (Equivalent Carbon Number) plot

This module groups lipids of the same lipid subclass with the same number of double bond equivalents (DBEs). Retention time against the number of carbons is plotted for each group to evaluate the annotation quality in context of the chromatographic behaviour. Trend line and score are added to assist the evaluation.

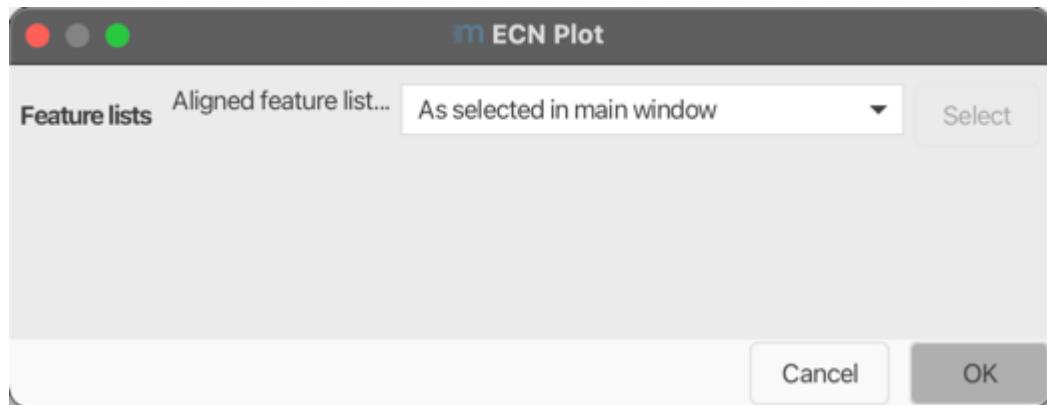
ⓘ Info

ECN models are most practical in combination with reversed phase chromatography.

Diacylglycerophosphocholines



Parameters



Feature List

Select one feature or aligned feature list.

⌚ October 31, 2024 09:02:47

👤 Ansgar Korf

12.6.7 Lipid Annotation summary

⚠ Warning

This module requires lipid annotations from the Lipid Annotation module.

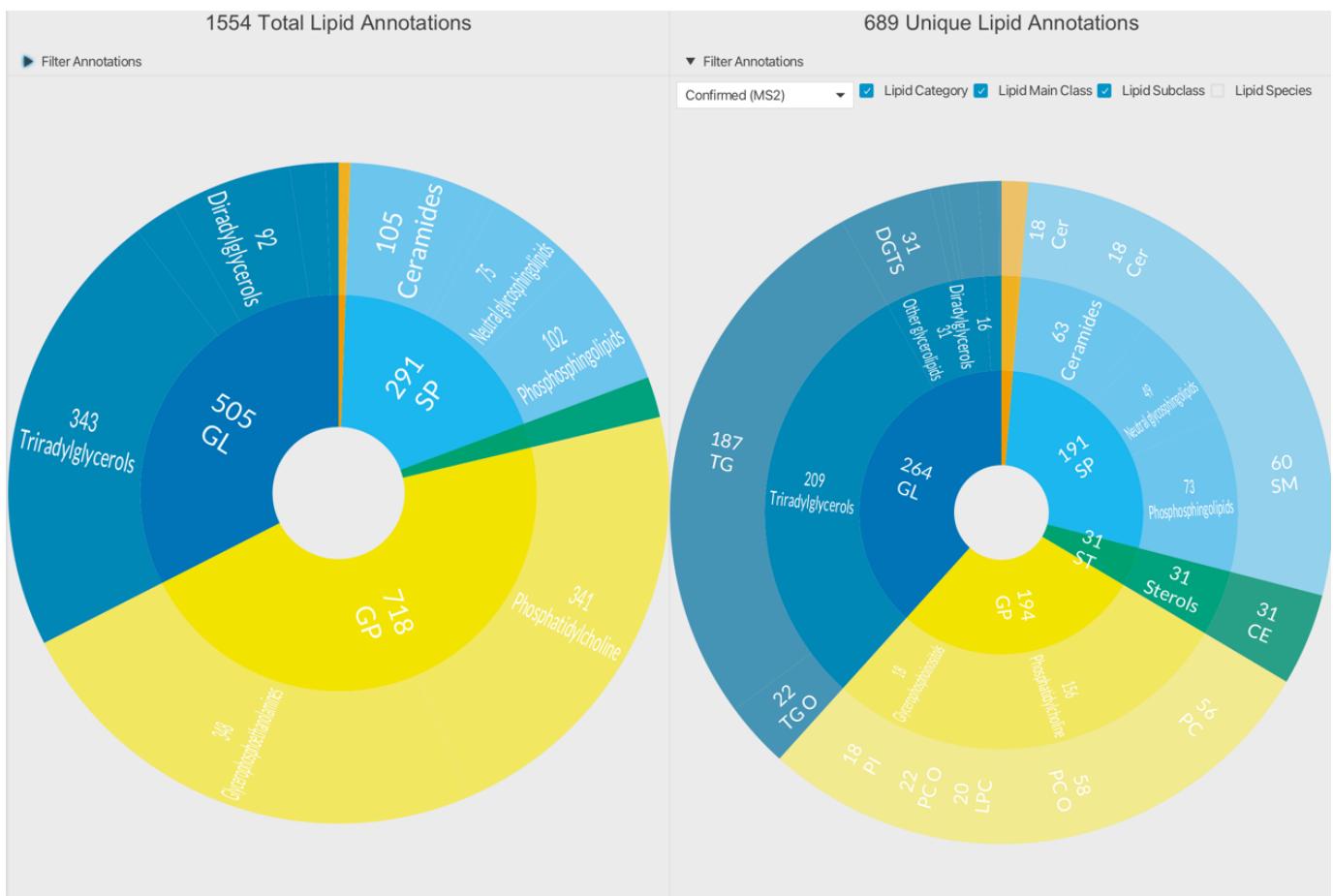
Description

☰ Visualization → Lipid annotation summary

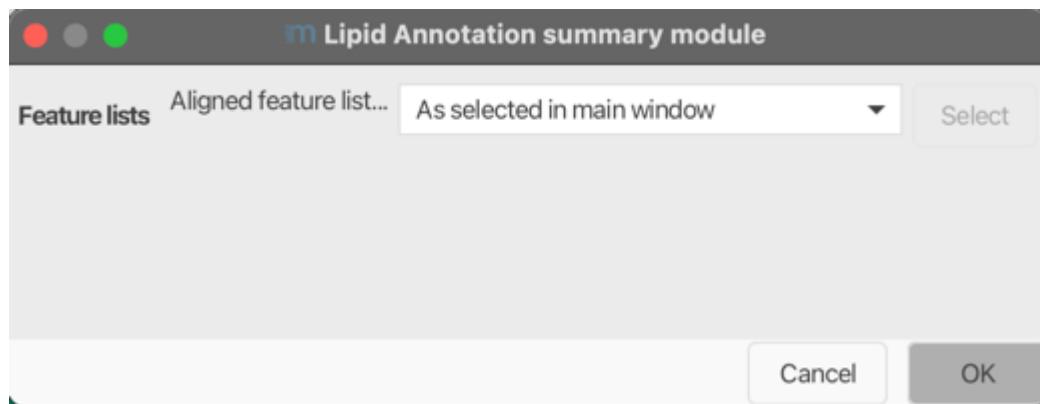
This module summaries all lipid annotations of a selected feature list in form of a sunburst chart. The layers from inside to outside follow the LipidMaps hierarchy: - Lipid Category - Lipid Main Class - Lipid Sub Class - Lipid Species

Unfolding the accordion "Filter Annotations" allows to control which hierarchy level should be displayed.

The left plot shows the total of all lipid annotations, including multiple annotations per feature. Multiple annotation levels (species and molecular species level) are not counted. The right plot shows the unique number of lipids, meaning the number of annotated features.



Both plots can be filtered individually. You can display: - "All", including all lipid annotations regardless of annotation quality - "Confirmed (MS2)", only annotations confirmed by MS2 fragments - "Unconfirmed (only MS1)", only annotations not validated by MS2, solely based on MS1 accuracy

Parameters**Feature List**

Select one feature or aligned feature list.

⌚ November 15, 2024 08:23:46

👤 Ansgar Korf

12.6.8 Additional tools

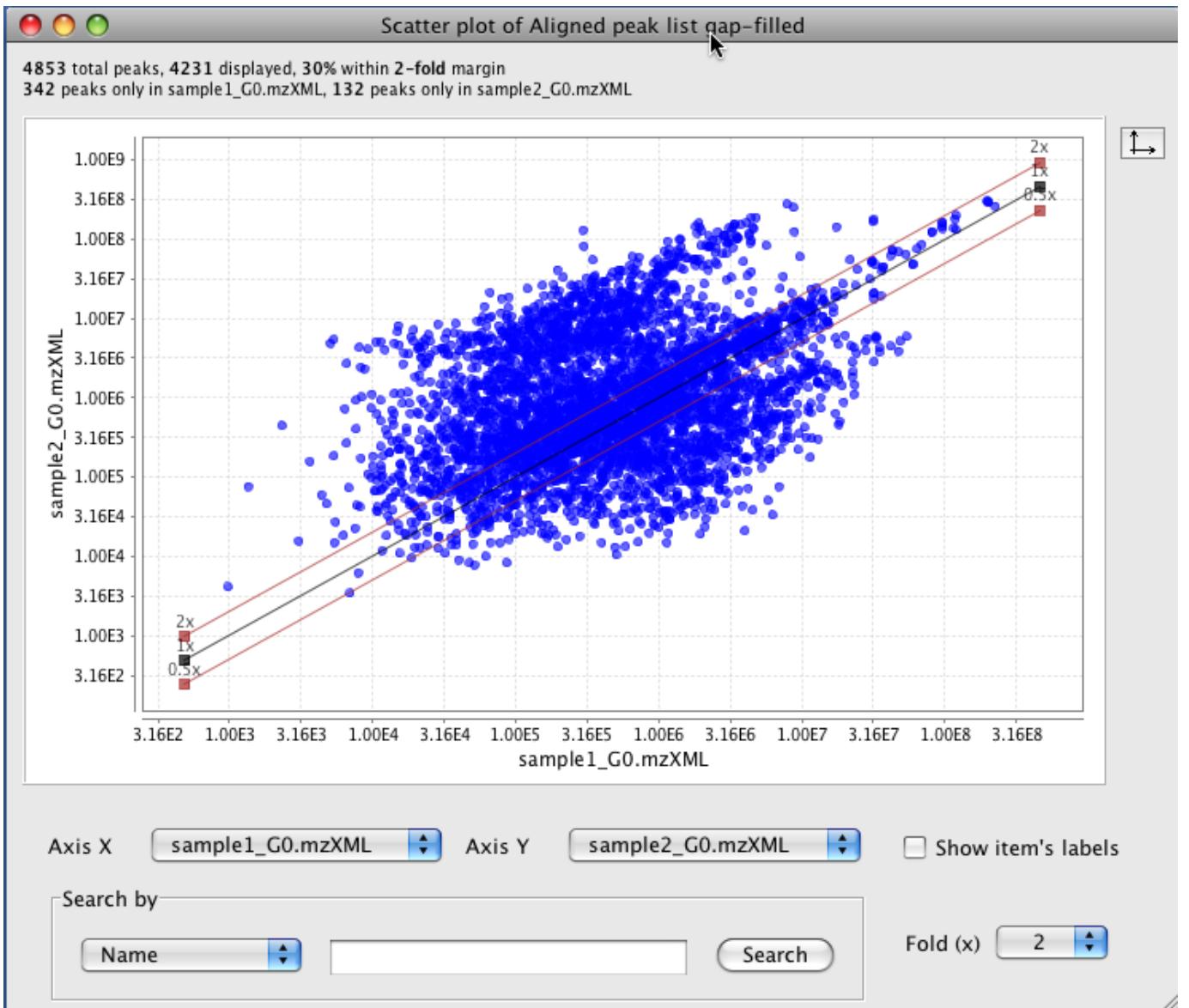
Scatter plot

DESCRIPTION

Visualization → Scatter plot

This tool shows a scatter plot with data from identified peaks in aligned feature list.

A search for a peak can be done using three options (name, retention time and m/z value).



Correlated features \(\Delta m/z histogram

DESCRIPTION

Visualization → Correlated features \(\Delta m/z histogram

This module plots all m/z deltas between correlated features in a histogram and offers a Gaussian fit.

PARAMETERS**Minimum Pearson correlation**

Minimum Pearson correlation of feature shapes.

Limit delta to m/z

Maximum m/z delta is the m/z of the smaller ion (feature list row).

m/z bin width

Binning of m/z values for feature picking

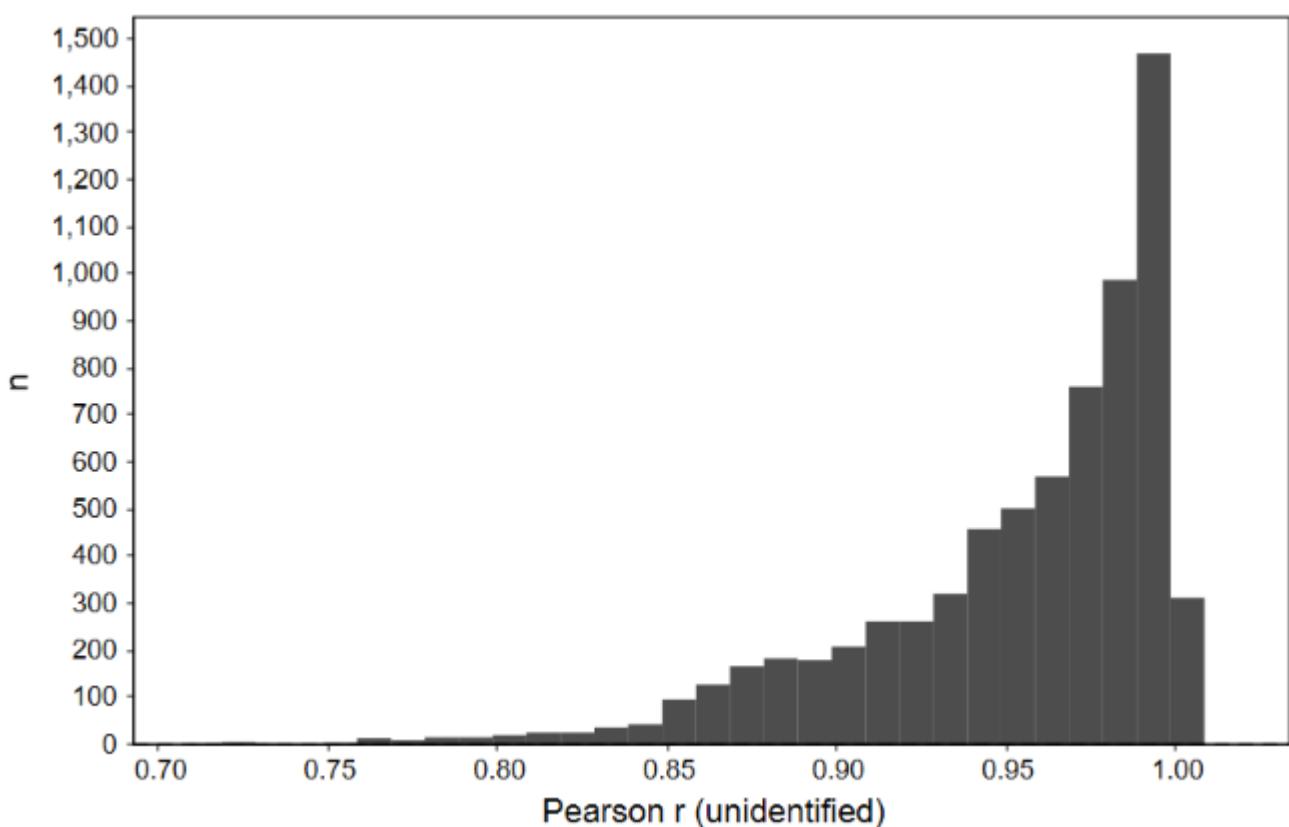
Append to file

Append the correlated features delta m/z to a csv file.

Correlation coefficient histogram**DESCRIPTION**

 This module is being updated. Some newer functionality might not be documented.

This module allows to plot all correlations between feature shapes.

**PARAMETERS****m/z bin width**

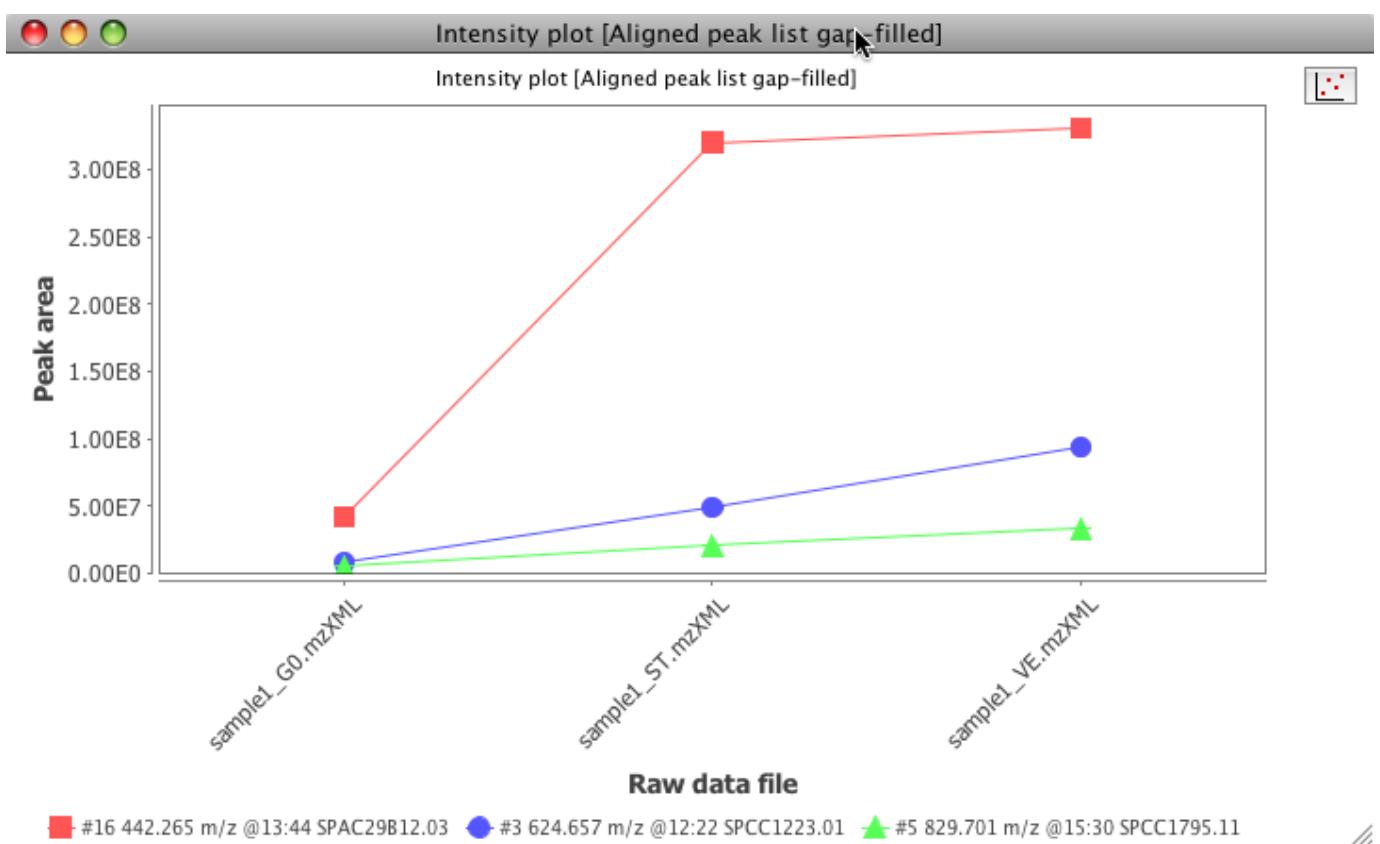
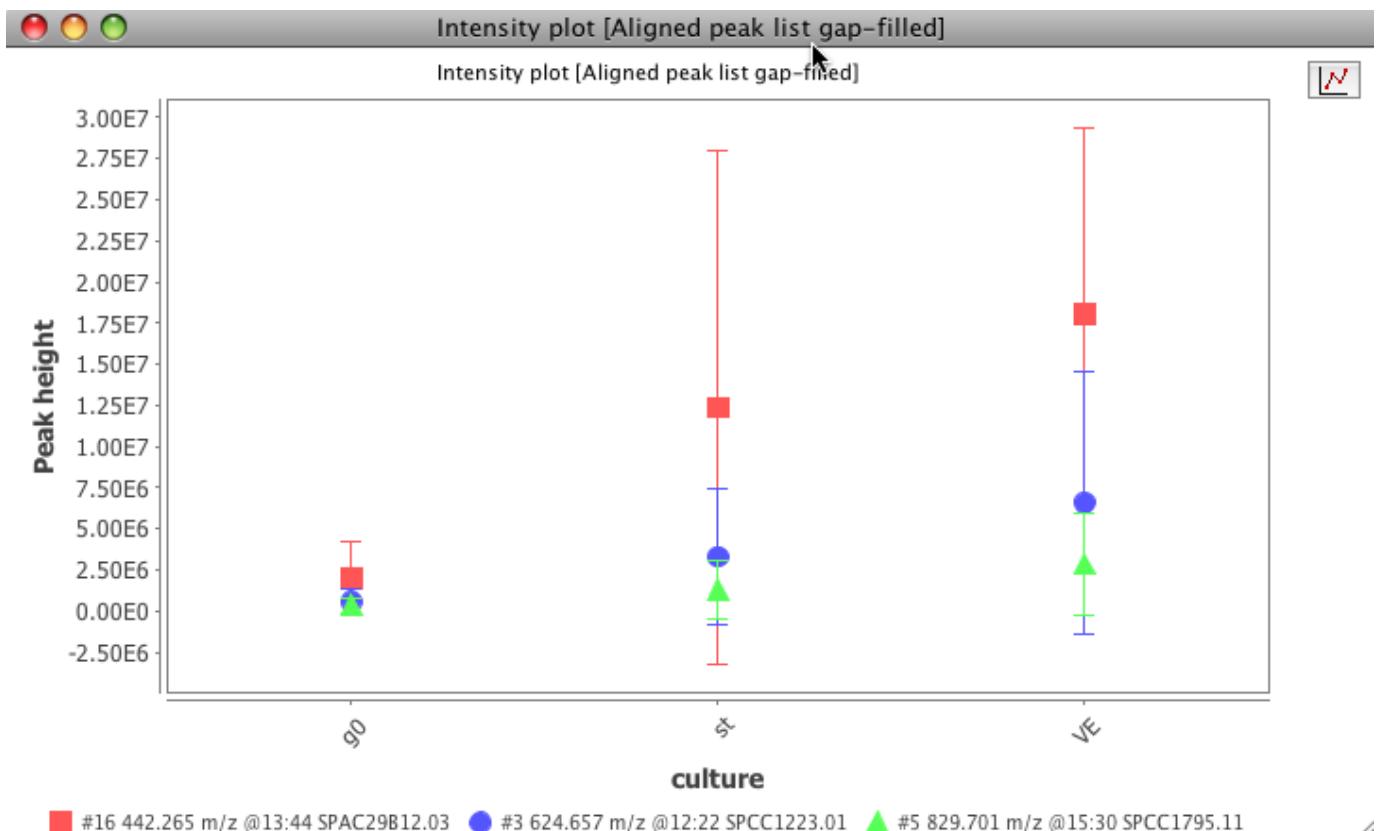
Binning of m/z values for feature picking

Feature intensity plot**DESCRIPTION**

 This module has a known bug and is being updated, which might affect its functionality.

≡ Visualization → Feature intensity plot

This plot allows to explore specific features against the raw data.



PARAMETERS**Data files**

Selects the raw data files from where the peaks were detected

X axis value

X axis display the raw data file name or the parameter defined in the "set sample parameters" window

X axis value

The user can choose from peak's height, area or retention time value to display in this axis.

Peaks

The user can select the peaks to use in this plot.

[omokshyna](#)

⌚ October 15, 2024 19:10:53

👤 [omokshyna](#)

12.7 Massvoltammogram

12.7.1 Electrochemistry/Mass Spectrometry

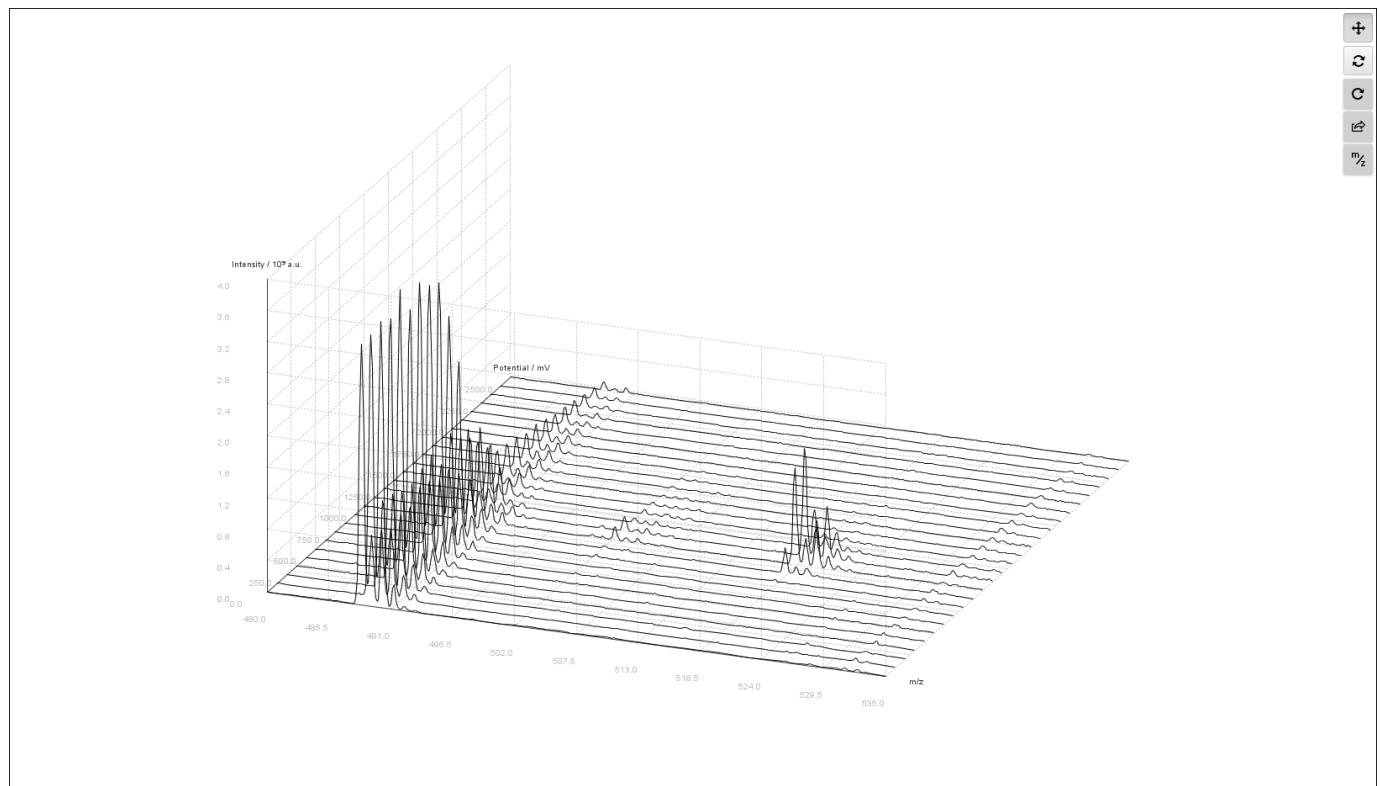
The hyphenation of electrochemistry (EC) and mass spectrometry (MS) is an analytical technique for the instrumental simulation of the human metabolism. By applying an electric potential to an electrochemical cell, that contains the analyte solution, it is possible to generate transformation products. These transformation products often closely resemble *in vivo*-metabolites. [1,2]

The applied potential is ramped in a given range to detect transformation products formed at different potentials. For an easy identification of the generated transformation products the collected data is often shown in so-called mass voltammograms. These mass voltammograms are three-dimensional plots in which mass spectra are shown in dependence of the corresponding applied potential. Using a mass voltammogram it is very easy to observe the different transformation products generated at different applied potentials. [3,4]

12.7.2 Description

« Visualization → Massvoltammogram

The "massvoltammogram" module allows the visualisation of EC/MS-data in the form of mass voltammograms. The screenshot below shows a mass voltammogram generated with MZmine3. Mass voltammograms can be generated directly from the raw data files without the need for any other processing steps. Otherwise ist is also possible to create mass voltammograms from processed feature lists.



The mass voltammogram can be moved and rotated via the first two buttons in the toolbar and scaled by using the mouse wheel. The third button in the toolbar can be used to reset the view. By right-clicking on any given point in the mass voltammogram the points exact coordinates are shown. Furthermore the range of shown m/z -values can be readjusted by pressing the " m/z "-button in the toolbar .

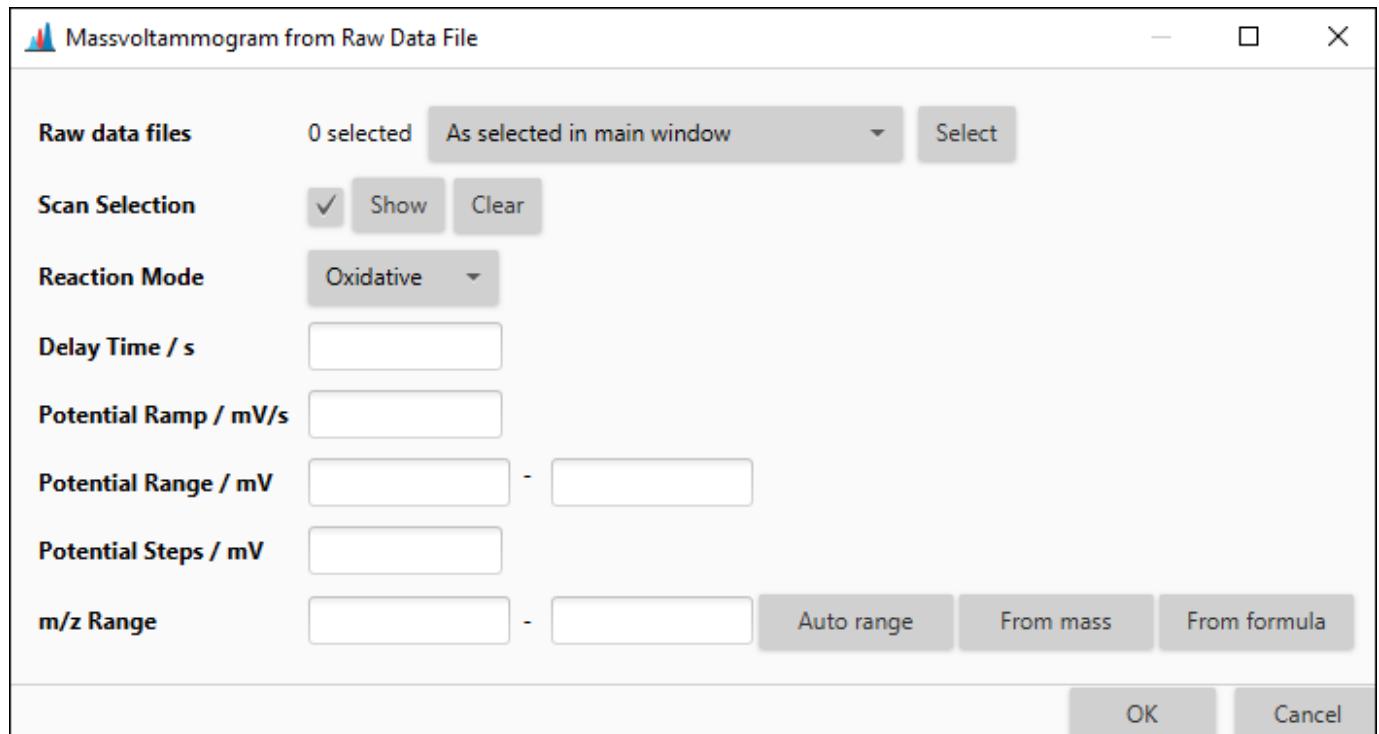
To export the mass voltammogram the second-last button in the toolbar needs to be pressed. There are two file formats available. The mass voltammogram can be exported as an image to a png-file or all mass spectra in the mass voltammogram can be exported to single csv-files.

12.7.3 Parameter

From raw data file

Visualization → Massvoltammogram → From Raw Data File

This module generates the mass voltammogram from a raw data file.



Raw data files

The raw data file of the EC/MS-experiment the mass voltammogram will be generated from.

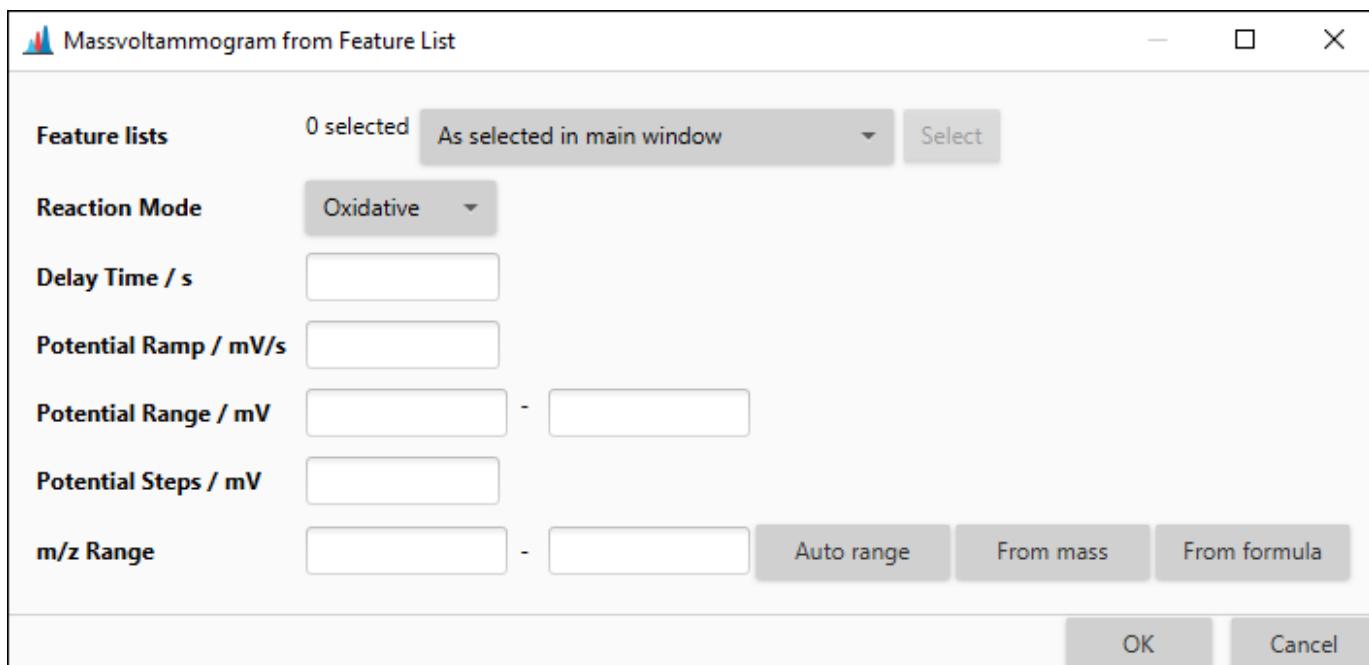
Scan Selection

The scan selection filter used to define the scan types to be used for the generation of the mass voltammogram. For example is it possible to select the MS level or the ion polarity.

From feature list

Visualization → Massvoltammogram → From Feature List

This module generates the mass voltammogram from a feature list.



Feature lists

The feature list the mass voltammogram will be generated from.

General Parameter

The following parameter are the same whether the mass voltammogram will be created from a raw data file or a feature list.

Reaction Mode

The mode the electrochemical cell was used in, whether oxidative or reductive.

Delay Time

The delay time in s between the generation of the transformation products and their detection in the mass spectrometer.

Potential Ramp

The speed in mV s^{-1} the applied potential is ramped at.

Potential Range

The potential range of the applied potential ramp in mV.

Potential Steps

The potential step size in mV between the mass spectra that are shown in the mass voltammogram.

m/z Range

The range of m/z -values to be shown in the mass voltammogram.

12.7.4 References

- [1] A. Baumann, U. Karst, *Expert. Opin. Drug. Metab. Toxicol.* **2010**, *6*, 715–731.
- [2] W. Lohmann, U. Karst, *Anal. Bioanal. Chem.* **2008**, *391*, 79–96
- [3] H. Faber, M. Vogel, U. Karst, *Anal. Chim. Acta* **2014**, *834*, 9–21.
- [4] T. Herl, F.-M. Matysik, *ChemElectroChem.* **2020**, *7*, 2498–2512.

⌚ March 13, 2023 12:42:51

👤 m_wesn01

12.8 Statistics dashboard

The statistics dashboard gives you an overview of your dataset. The statistical insight is directly linked to the raw features, giving you direct access to the significant differences in your samples. Nevertheless, the dashboard is not meant to substitute downstream statistical analysis.

Warning

The description here will be held in simple terms. For a more in-depth description refer to other sources.

Info

Both plots allow ROI selection, allowing you to extract the area of the most significant features into a new feature list.

12.8.1 Principal component analysis (PCA)

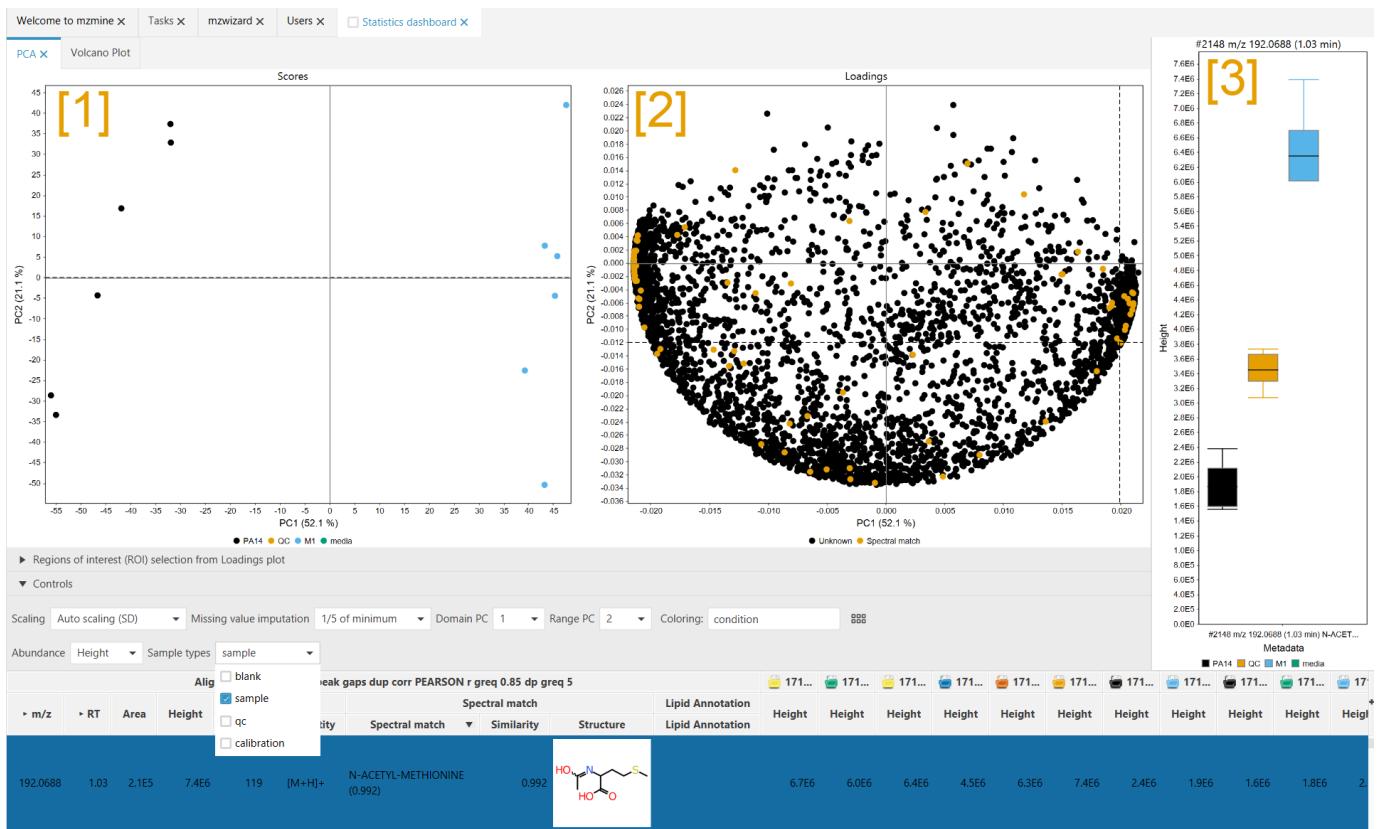
Plot description

Principal component analysis is a type of unsupervised, multivariate statistics, meaning multiple variables (here features) are analysed for each observation (sample/data file). The PCA will find features that show the most variance in your sample set and then cluster your samples based on the abundance of the features.

The PCA consists of two plot, the Scores plot [1] and the Loadings plot [2]. The scores plot shows how your samples (= data files) have been clustered by the PCA. Each dot represents a single data file. The loadings plot shows how "*responsible*" the features are for the clustering. Each dot represents a single row (= multiple features across all your samples).

Notice the identical axis titles in the Scores and Loadings plot. In general this means that features with a high x-axis value in the Loadings plot are responsible for shifting the data files in the scores plot towards high x-axis values. (The same counts for the y-axis)

As seen in the screenshot, a feature on the right side of the loadings plot is selected. The scores plot shows that the "M1" group samples (blue) are located on the right. The bar chart [3] shows that the selected feature is upregulated on the in the "M1" group (blue) and downregulated in the "PA14" group (black), which is shifted to the left in the Scores plot [1].



Controls

Scaling Defines how the intensities of features are scaled before the PCA. All scaling methods perform mean-centering prior to scaling. **Auto scaling** scales by the standard deviation in a row and is the most common scaling method. **Range scaling** scales all features to values between -1 and 1, meaning every feature is equally important for the grouping. **Pareto scaling** scales by the square root of the standard deviation of the intensity in a row. This is common in metabolomics, but puts more emphasis on the absolute intensity than auto scaling.

Missing values imputations: Defines how missing values (= no feature detected) are handled. Since missing values are not allowed in PCAs, they must be replaced by something. **1/5 of minimum** assumes that the detector was not sensitive enough to detect something, but that there may be something below the LOD. **Zero** imputes missing values with 0 (less common).

Domain PC and **Range PC** define the axis in the plots. The dimensionality reduction by PCA creates more than one possible projections. In your dataset, more than principle component (PC)1 and PC2 may be relevant. The PCs are sorted by how much of the variance they explain in your dataset.

Coloring Selects a metadata column to color your samples in the Scores plot. This information has no influence on the grouping, since PCA is performed without any metadata.

Abundance Select which abundance measure shall be used to perform the PCA. Height or area.

Sample types Defines which sample types are used for the PCA. mzmine automagically detects the type of your sample by its naming. If the sampel name contains *blank* or *QC* mzmine will pick that up. The sample types can be set in the metadata table (CTRL + M).

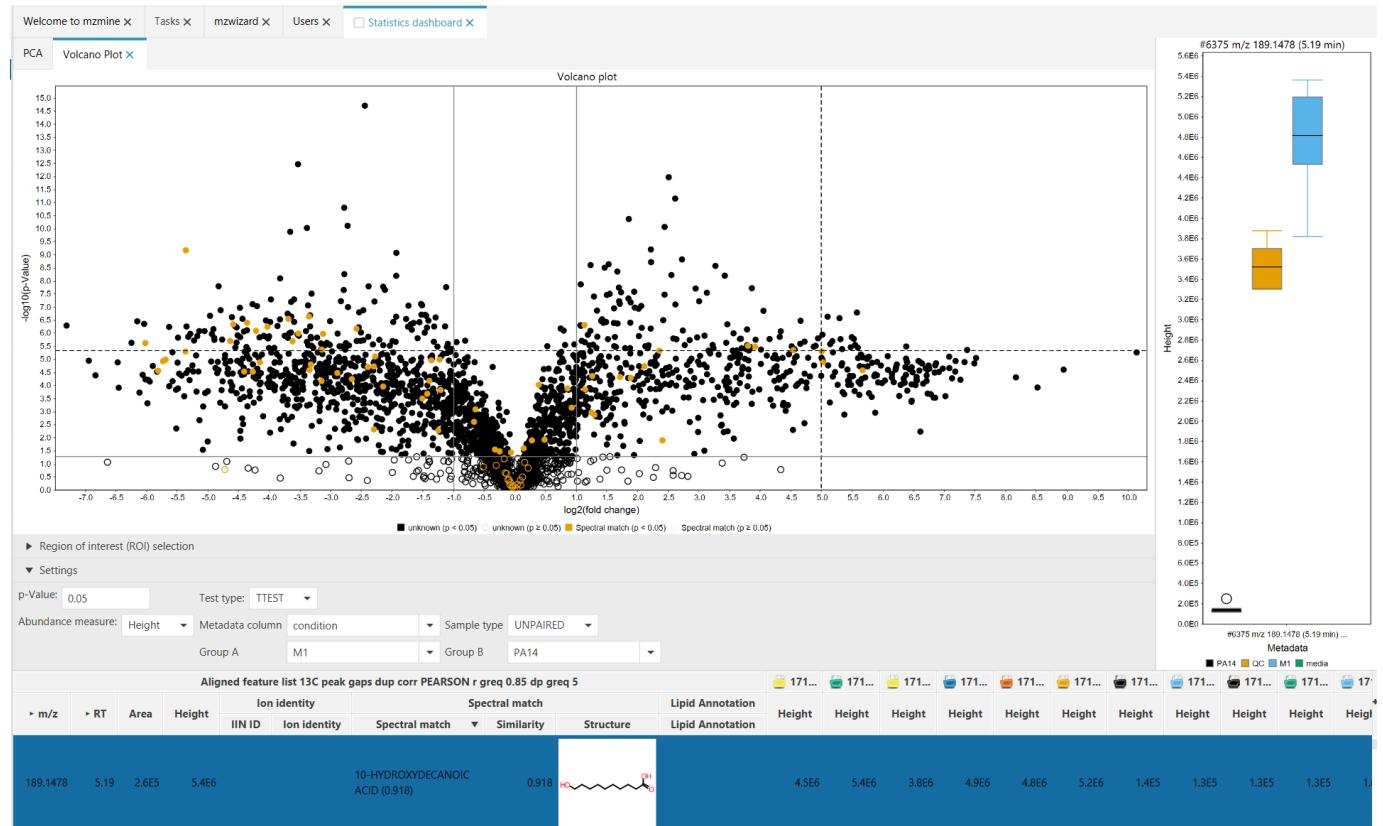
12.8.2 Volcano Plot

Plot description

The Volcano plot is a univariate statistical method to compare two sample groups. **Each group must have at least 2 samples.** To achieve this, the significance of the fold change difference ($-\log_{10}(p)$) is plotted against the \log_2 of the fold change between two groups (= $\text{avg}(\text{grp1})/\text{avg}(\text{grp2})$).

To be classified as a significant change, the p-Value is usually set at 0.05. This means that the possibility for this change to happen randomly is lower than 5 percent. This "border" is shown in the plot by a horizontal marker. In simple terms, this means that a feature has different intensities in the two groups, while the standard deviation within a group is small.

The two vertical markers at -1 ($=\log_2(1/2)$) and 1 ($=\log_2(2/1)$) indicate a down- or upregulation by a factor of two.



CONTROLS

p-Value Controls the marker of the p value for features to be significant. All features below this p-Value (= above this marker, meaning significantly changed) are drawn as filled dots. All features below this marker (= higher p value, not significant) are drawn as outlined circles.

Abundance measure Define if height or area shall be used for the statistics.

Test type Define which test shall be used to calculate the p-Values. Only T-Test is available for now.

Metadata column Select which metadata column shall be used to read the sample groups.

Test type Define if the samples originate from a paired or unpaired experiment. Unpaired is the default and means that the, e.g., two sets of bacterial strains were taken for this analysis, in which one set was treated and the other one was not treated. Paired means that the test originates from a single set of, e.g., bacterial strains, which was untreated before and treated afterward (with samples being taken before and after the treatment). Paired testing requires equal group size.

Group A and **Group B** define the two groups in the **Metadata column** that are being compared in the volcano plot.

⌚ April 9, 2025 13:30:39

👤 Steffen Heuckeroth

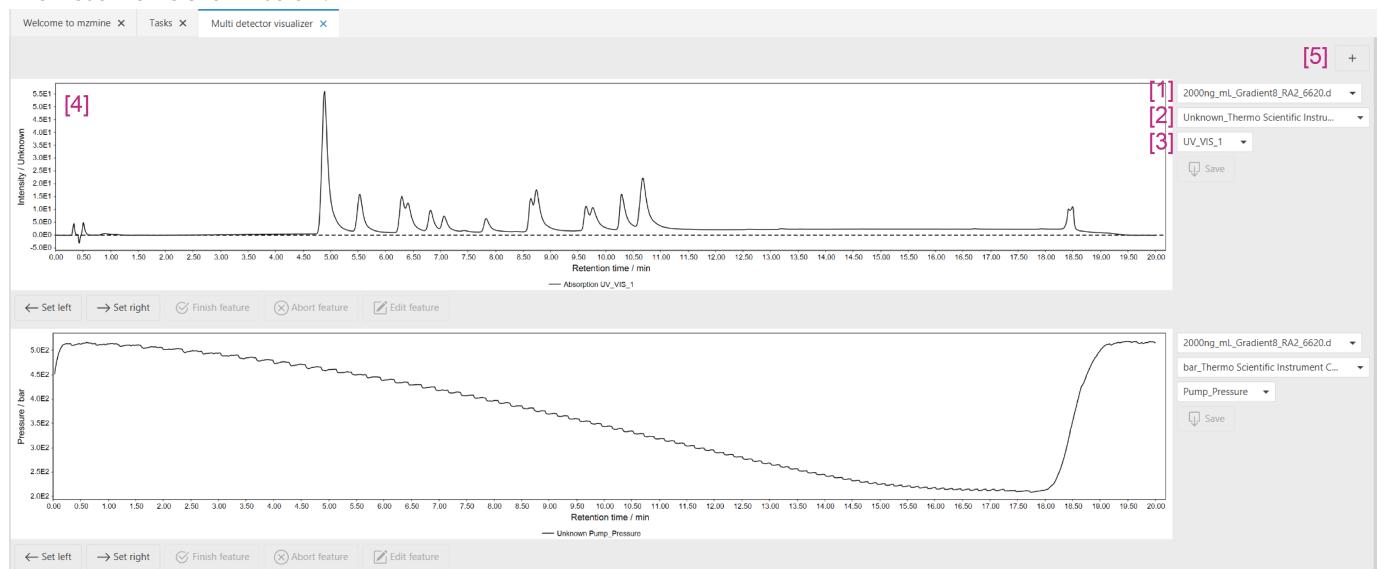
12.9 Multi detector visualizer

The multi detector visualizer allows visualization and manual integration of **traces** from other detectors.

To open the **Multi detector visualizer**, at least one data file in the project must contain data from other detectors. Only files with such traces will be eligible for the visualizer.

12.9.1 Visualizer layout

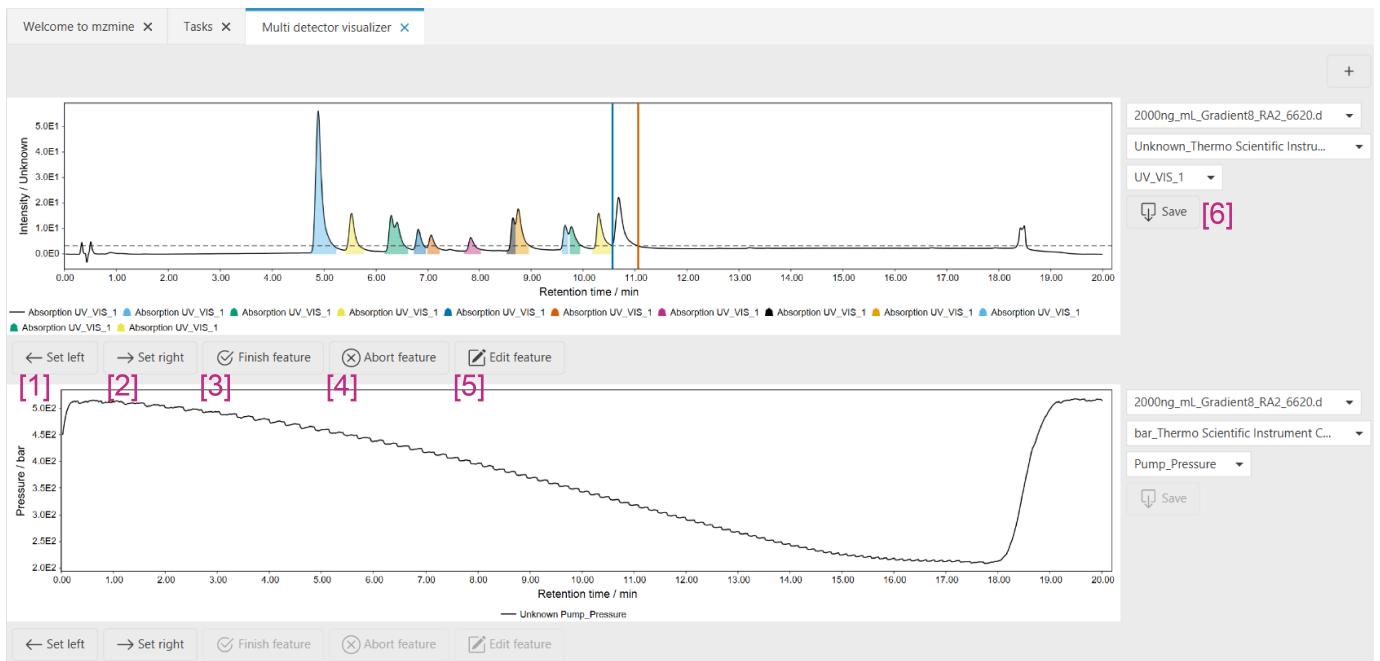
The visualizer is shown below:



1. Select the data file you want to visualize. After importing a new data file, the visualizer must be reopened.
2. This box lists the available detector traces for the selected files. The traces imported from the raw data will be named as such: <unit>_<detector name> and grouped by the unit of the detector. If they are created from other spectral data by the [Chromatogram builder \(Other spectral data\)](#), the [Name parameter](#) will be used.
3. Select the specific trace you want to visualize. In case there are [preprocessed traces](#), these will be used for visualization and [manual integration](#).
4. The chromatogram chart shows the selected trace.
5. In case you want to visualize multiple files/traces, you can add another chart by the + button.

12.9.2 Manual integration

Manual integration is performed by first selecting the trace you want to integrate (see above).



1. The left boundary of a peak is set by first clicking the "Set left" button and then clicking **on** the chromatogram in the chart or by holding **CTRL/⌘ + CLICK**
2. The right boundary of a peak is set by first clicking the "Set right" button and then clicking **on** the chromatogram in the chart or by holding **CTRL/⌘ + SHIFT + CLICK**
3. After setting the left and right boundary, the integration is accepted by clicking "Finish feature" or by **ALT + CLICK** in the chart.
4. In case a feature shall not be accepted and the boundaries shall be removed, you can use the " Abort feature" button.
5. After a feature has been accepted, you can re-edit it, by clicking the "Edit feature" button. If you want to delete a feature from a trace, start the editing mode and then press "Abort feature".
6. To save the manual edits, press the "Save" button. Unless "Save" is pressed, the edits will not be saved.

⌚ July 28, 2025 09:10:04

👤 Steffen Heuckeroth

12.10 Correlation dashboard



To process data from other detectors, the Other detector service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

The Correlation dashboard allows investigation of the results created by the [Correlate Ms features with other detectors](#).

12.10.1 Interface

The interface is shown below:



- Shows the full [preprocessed trace](#) (line) and detected [features](#) (filled area) of the other detector trace. If an MS feature with an associated correlated trace is selected in the feature table, this plot updates automatically and re-positions the crosshair to the correlated feature. Selecting a different feature in this plot, will update the other detector feature in plot **3**.
- Shows the full chromatogram (re-extracted from the raw data, line) and the selected feature from the feature table (area). Selecting a new feature in the feature table will update this plot. The RT axis is synchronized to plot **1**.
- Shows an overlay of the selected MS feature (from the feature table) and the selected other detector feature. By default, if a new MS feature is selected, this plot automatically shows the correlated feature. If there is no correlated feature or the selection in plot **1** is updated manually, this plot shows the selected other detector feature from plot **1**.
- Shows a feature table with the correlated other detector traces.

Control panel

The additional controls are described below:



1. The trace selection allows selecting a specific other detector trace in case no correlation was found or the correlation was not executed yet.
2. Set an additional RT shift to the other detector trace. This is useful during method optimisation to determine the shift between the other detector and the MS. This shift can then be set in the wizard or the [Shift traces](#) module.
3. **Filter MS features** sets the RT filter (8) of the feature table to the RT range of the currently selected other detector feature. This allows you to investigate all MS features that have their peak maximum in the RT range of the other detector feature.

Define as correlated allows you to manually specify which feature should be marked as correlated.

Clear selected correlation removes the currently selected correlated feature from the selected MS feature. Note that there may be more than one correlated feature. This button only removes a single one.

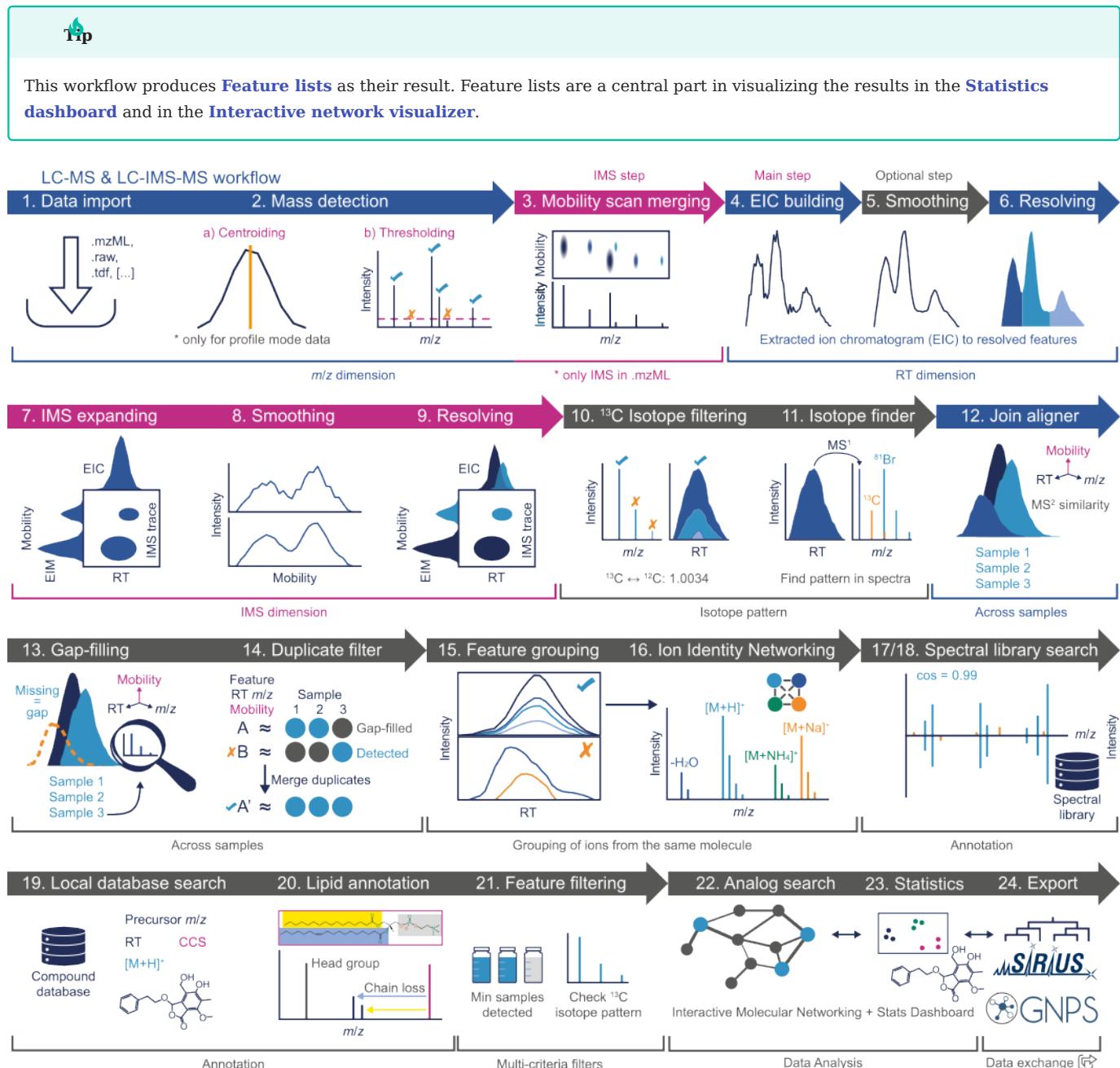
⌚ July 29, 2025 10:09:01

👤 Steffen Heuckeroth

13. Workflows

13.1 Untargeted LC-MS Workflow

The workflow proposed herein is intended as a general pipeline for untargeted LC-MS (or LC-MS/MS) data preprocessing. The main goal is essentially to turn the highly-complex LC-MS raw data into a list of features, and corresponding signal intensity, detected across the analysed samples. Such feature lists can then be exported for further downstream analysis (e.g., identification, search against spectral libraries, statistical analysis, etc.). A schematic representation of the workflow is shown below:



13.1.1 Raw data processing

The raw data processing consists of essentially two steps: [Data import](#) and [Mass detection](#)

Raw data import

Either open (e.g. mzML) and native vendor (e.g. Thermo, Bruker) data formats can be imported in mzmne. All the supported formats can be found [here](#).

Mass detection

This step produces a list (referred to as "mass list") of the m/z values found in each MS scan across the LC run that exceed a user-defined threshold (i.e. noise level). For more details see the [Mass detection](#) module.

13.1.2 Feature processing

The goal of the "Feature processing" is to obtain a list of all the detected features (characterized by a RT and m/z value) from the raw LC-MS data.

Chromatogram building

The first step in the "Feature processing" is to build the so-called extracted ion chromatograms (EICs) for each detected mass (see "Mass detection"). For this, use the [Chromatogram builder](#) module.

The "detected" features in each file are listed in the so-called "feature lists", which are then further processed and aligned to connect corresponding features across all samples.

Smoothing in retention time dimension (optional)

Depending on the LC peak shape (i.e. data noisiness), the user can perform smoothing in retention time dimension. For more details see the [Mass detection](#) and [Smoothing](#) modules.

Feature resolving

Feature resolving step enables separation of co-eluting and overlapping chromatography peaks and as such is one of the pivotal steps in data preprocessing. For more details on the algorithm used and parameters settings, see the [Local minimum resolver](#) module.

¹³C isotope filter (isotope grouper)

In order to remove redundant features, such as the ones generated due to the presence of isotopologues, isotope filter should be applied. ¹³C isotope filter (isotope grouper) removes ¹³C isotope features from the feature list. Use the isotope finder for more sensitive detection of possible isotope signals.

Isotope pattern finder

Isotope pattern finder searches for the isotope signals of selected chemical elements in the mass list of each feature. The isotope pattern detected by the **isotope finder** module has priority over the one detected by the **isotope filter (grouper)** module, if both are available. For more details, see the [Isotope pattern finder](#) module.

13.1.3 Feature alignment

Feature alignment enables alignment of corresponding features across all samples.

Join aligner

This module aligns detected peaks in different samples through a match score. The score is calculated based on the mass and retention time of each peak and ranges of tolerance stipulated in the parameter setup dialog. For more information, see the [join aligner](#) module.

13.1.4 Gap-filling

Absence of features in some samples can either reflect the truth - the metabolite is absent in the given sample, or it can be due to data preprocessing. To account for this, gap filling is applied as the next step.

Gap-filling (peak finder)

Gap-filling can be performed on the aligned feature lists to cope with missing features that might be artifacts of the feature-detection process. For more details see the [Gap-filling \(peak finder\)](#) module.

13.1.5 Export

Depending on the downstream analyses, there are several export options which are accessible through **Feature list methods → Export feature list**.

For GNPS-Feature based molecular networking, see [GNPS-FBMN](#) or apply Interactive Molecular Networking directly in mzmine [molecular_networking.md](#)

13.1.6 References

Karaman, I.; Climaco Pinto, R.; Graça, G. Chapter Eight - Metabolomics Data Preprocessing: From Raw Data to Features for Statistical Analysis. In *Comprehensive Analytical Chemistry*; Jaumot, J., Bedia, C., Tauler, R., Eds.; Elsevier, 2018; Vol. 82, pp 197-225.

Pluskal, T.; Korf, A.; Smirnov, A.; Schmid, R.; Fallon, T. R.; Du, X.; Weng, J.-K. CHAPTER 7:Metabolomics Data Analysis Using MZmine. In *Processing Metabolomics and Proteomics Data with Open Software*; 2020; pp 232-254.

Du, X.; Smirnov, A.; Pluskal, T.; Jia, W.; Sumner, S. Metabolomics Data Preprocessing Using ADAP and MZmine 2. In *Computational Methods and Data Analysis for Metabolomics*; Li, S., Ed.; Springer US: New York, NY, 2020; pp 25-48.

13.1.7 Page Contributors

Ansgar Korf, Robin Schmid, Steffen Heuckeroth, lalalana5, omokshyna, tdamiani

⌚ August 15, 2025 19:03:26

👤 Ansgar Korf, Robin Schmid, Steffen Heuckeroth, lalalana5, omokshyna, tdamiani

13.2 Untargeted LC-IMS-MS Workflow Overview

Compared to regular LC-MS, LC-IM-MS data is more complex due to the additional separation dimension. Since some terms might not be straightforward for new users, a basic explanation of IM separation principles and the terminology used within this documentation is provided [here](#).



This workflow produces **Feature lists** as their result. Feature lists are a central part in visualizing the results in the **Statistics dashboard** and in the **Interactive network visualizer**.

13.2.1 Supported formats

- Vendor formats:
- .tdf (Native Bruker LC-IMS-MS and MALDI-IMS-MSI format)
- .tsf (Native Bruker MALDI-IMS-MS (single shot) format)
- .mzML
- Created via [MSConvert](#) from native Bruker data
- Created via [MSConvert](#) from native Waters/Agilent data

13.2.2 Feature detection workflows

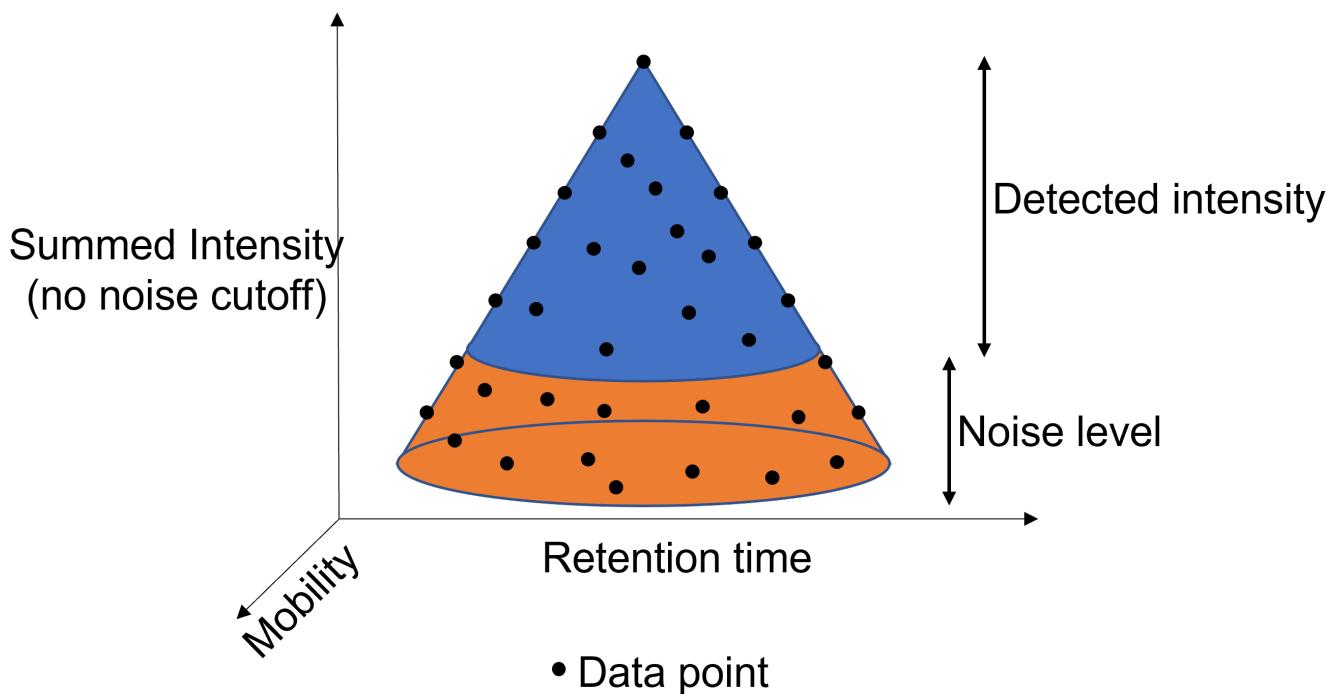
Ion mobility data can be processed in mzmine in two ways. The first few steps are different for the two workflows (see below).

1. [LC-IMS-MS workflow via Chromatogram builder and IMS expander \(recommended\)](#)
2. [LC-IMS-MS workflow via Ion mobility trace builder / Recursive IMS builder](#)

While these lists might seem fairly similar, there are some differences in the processing approach. The LC-IMS-MS workflow builds ion mobility traces from the data in the mobility scans, whilst the LC-MS workflow builds EICs from the summed frames. For ion mobility data imported from .mzML files, accumulated frame spectra have to be built from the individual mobility scans after [mass detection](#). Since the mass detection impacts the computation of accumulated frame spectra in the same way it would impact the [ion mobility trace builder](#), the differences from this workflow and the [ADAP workflow](#) will be negligible. However, frame spectra for native Bruker .tdf raw data are summed by the vendor library during file import. Here, the frame spectra are generated from the raw data and thus result in higher intensities, since the low abundant data points on the edges of

the mobility and retention time peaks are not cut-off by the mass detection step. (see below)

Ion mobility trace builder noise cut-off



Therefore, the more low abundant compounds might be detected, if the [LC-MS workflow](#) is recommended.

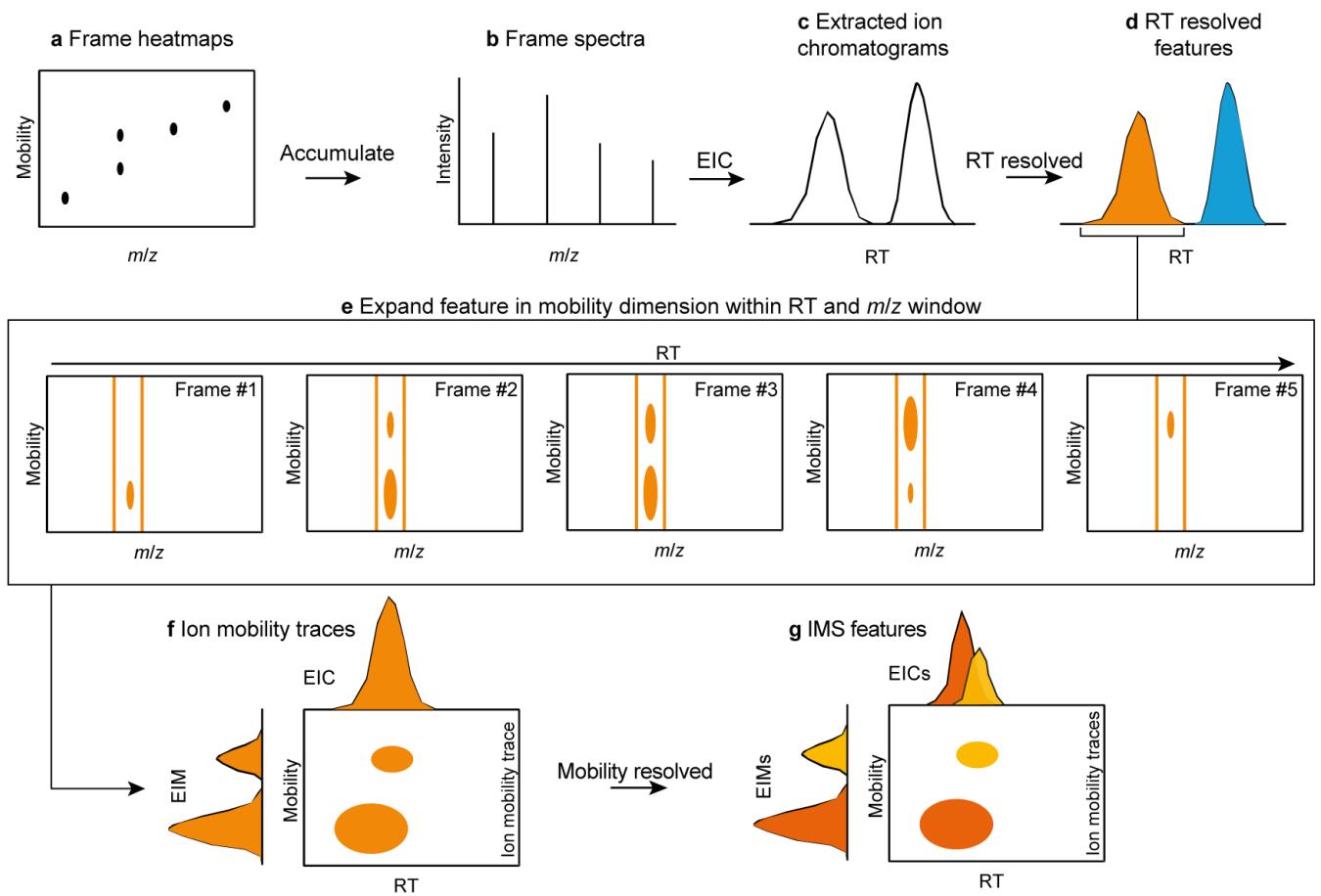
[LC-MS workflow \(recommended\)](#)

LC-IMS-MS data can also be processed via the regular LC-MS modules. If necessary, detected features can be expanded into the mobility dimension.

For this workflow, generation of summed frame spectra via the [Mobility scan merging](#) module is a mandatory step, if the data was imported from an .mzML file (automatically generated via native Bruker import).

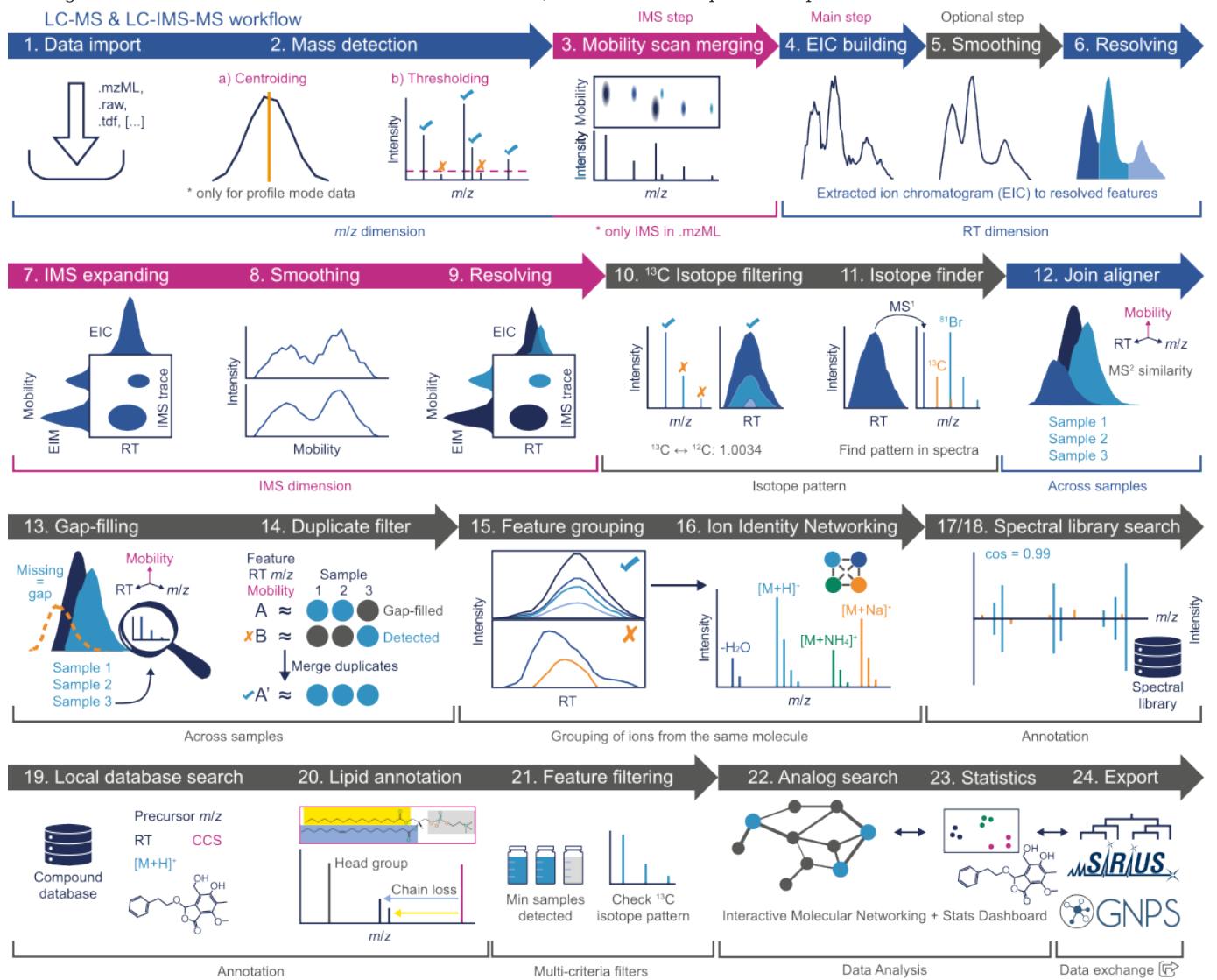
- [Data import](#)
- [Mass detection](#)
- [Mobility scan merging \(mzML data only\)](#)
- [Chromatogram builder](#)
- [Smoothing in retention time dimension \(optional\)](#)
- [Resolving in retention time dimension](#)
- [Expanding EICs in mobility dimension](#)
- [Smoothing in mobility dimension \(optional\)](#)
- [Resolving in mobility dimension](#)
- [Smoothing in rt and mobility dimension \(optional\)](#)
- Some recognised features might have rather noisy signals (in rt and mobility dimension) after the mobility resolving step. If smoother shapes are required, the smoothing can be reapplied afterwards. In that case, smoothing can be applied to both dimensions at once.

The basic principle of the workflow is illustrated below:



a, IMS heatmaps are accumulated to frame spectra (**b**). From the accumulated frame spectra, EICs are built (**c**). These EICs are resolved to individual features by a resolver (**d**). The resolved features are now defined by RT and m/z windows, which can be expanded in mobility dimension by the Ims expander (**e**). The resulting ion mobility traces (**f**), have to be resolved in mobility dimension afterwards, to create individual IMS features.

This figure shows the workflow in a more detailed manner, with additional optional steps.



LC-IMS-MS workflow

The LC-IMS-MS workflow will directly build ion mobility traces from the raw data in the mobility scans. This workflow does not necessarily require summed frame spectra. However, if extracted ion chromatograms shall be visualized via the [Chromatogram visualizer](#), the frame intensities are used. In case these are not present, the chromatograms will be blank. Note that feature

intensities from the LC-IMS-MS workflow might not exactly match the frame chromatograms due to summing being executed prior to thresholding (for native Bruker data). Furthermore, multiple isomers might hide behind a single chromatographic peak.

- Data import
- Mass detection
- Ion mobility trace builder
- Smoothing in retention time dimension (optional)
- Resolving in retention time dimension
- Smoothing in mobility dimension (optional)
- Resolving in mobility dimension
- Smoothing in rt and mobility dimension (optional)
- Some recognised features might have rather noisy signals (in rt and mobility dimension) after the mobility resolving step. If smoother shapes are required, the smoothing can be reapplied afterwards. In that case, smoothing can be applied to both dimensions at once.

13.2.3 Graphical comparison of LC-MS and LC-IMS-MS data

Data comparison

13.2.4 Page Contributors

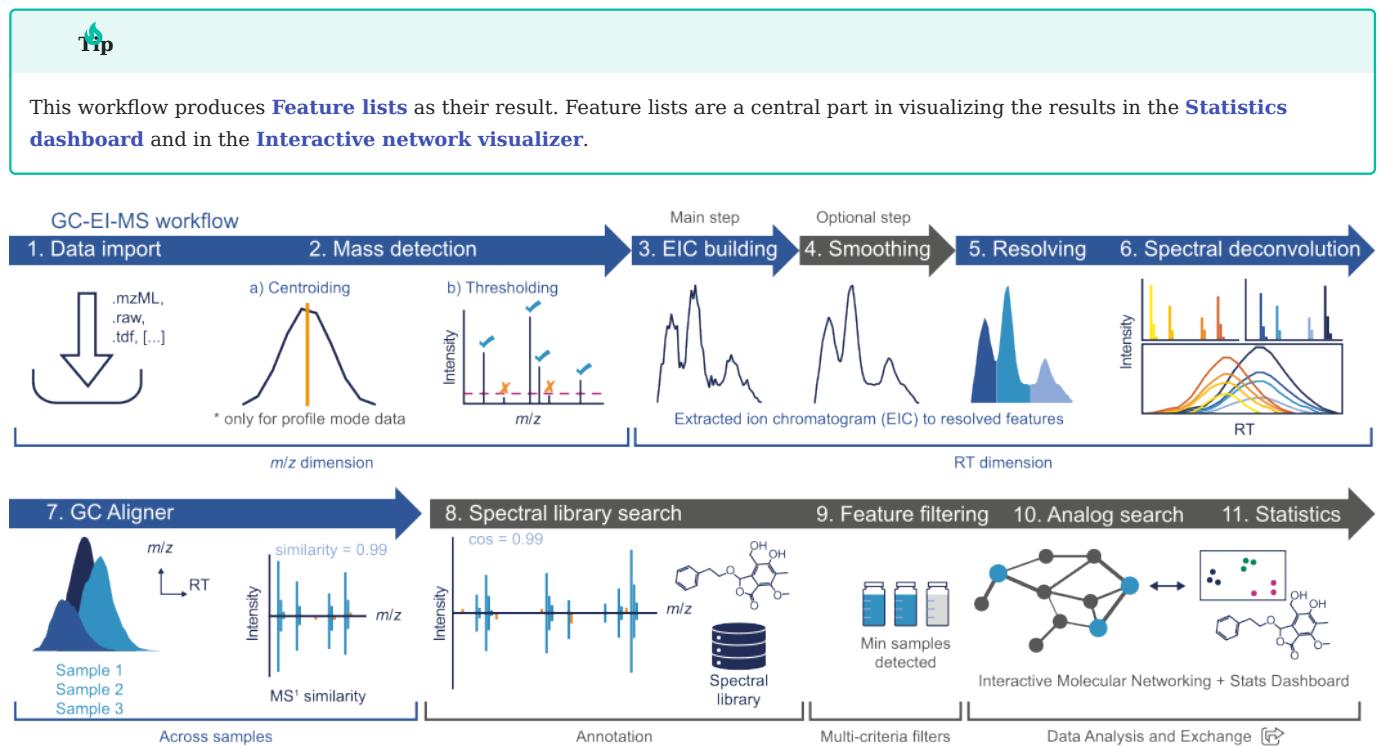
Ansgar Korf, Robin Schmid, Steffen Heuckeroth, omokshyna, tdamiani

⌚ August 15, 2025 19:03:26

👤 Ansgar Korf, Robin Schmid, Steffen Heuckeroth, omokshyna, tdamiani

13.3 Untargeted GC-MS Workflow

The workflow proposed herein is intended as a general pipeline for untargeted GC/EI-MS data preprocessing. GC/CI-MS can be processed analog to [LC-MS](#). The main goal is essentially to turn the highly-complex GC-MS raw data into a list of features, and corresponding signal intensity, detected across the analysed samples. Such feature lists can then be annotated and/or exported for further downstream analysis (e.g., identification, search against spectral libraries, statistical analysis, etc.). A schematic representation of the workflow is shown below:



13.3.1 Raw data processing

The raw data processing consists of essentially two steps: [Data import](#) and [Mass detection](#)

Raw data import

Either open (e.g. mzML) and native vendor (e.g. Thermo, Bruker) data formats can be imported in mzmine. All the supported formats can be found [here](#).

Mass detection

This step produces a list (referred to as "mass list") of the m/z values found in each MS scan across the LC run that exceed a user-defined threshold (i.e. noise level). For more details see the [Mass detection](#) module.

13.3.2 Feature detection

The goal of the "Feature detection" is to obtain a list of all the detected features (characterized by a RT and m/z value) from the raw GC-MS data.

Chromatogram building

The first step in the Feature detection is to build the extracted ion chromatograms (EICs) for each detected m/z (see [Mass detection](#)). For this, use the [Chromatogram builder](#) module.

The "detected" features in each file are listed in the so-called "feature lists", which are then further processed and aligned to connect corresponding features across all samples.

Smoothing in retention time dimension (optional)

Depending on the GC peak shape (i.e. data noisiness), the user can perform smoothing in retention time dimension. For more details see the [Mass detection](#) and [Smoothing](#) modules.

Feature resolving

Feature resolving step enables separation of co-eluting and overlapping chromatography peaks. It is one of the pivotal steps in data preprocessing. For more details on the algorithm used and parameters settings, see the [Local minimum resolver](#) module.

Spectral deconvolution

When using a hard ionization technique such as electron ionization (EI), multiple m/z values belong to the same compound. These m/z fragments can be grouped together based on their chromatographic behaviour (peak shape correlation). The grouping results in a cleaned up feature list as well as high quality deconvoluted GC/EI-MS spectra, perfect for spectral library matching. Find more info on [spectral deconvolution](#) here.

13.3.3 Feature alignment

Feature alignment enables alignment of corresponding features across multiple samples.

GC aligner

This module aligns detected features in different samples through a match score. The score is calculated based on the retention time and spectral similarity of each feature. For more information, see the [GC aligner](#) module.

13.3.4 Annotation, Filtering, Statistics and Export

Depending on the downstream analyses, there are several options which are accessible through the **Feature list methods** menu. Annotate compounds using [spectral library search](#), apply various filtering criteria, explore the results using the statistics dashboard, or export the results.

13.3.5 Page Contributors

Ansgar Korf, Robin Schmid

 August 15, 2025 19:03:26

 Ansgar Korf, Robin Schmid

13.4 Imaging Workflow

This page describes the mzmine workflow for non targeted feature detection in imaging datasets. The workflow is designed for MS imaging data and tested with matrix-assisted laser desorption/ionisation (MALDI)-MS data.



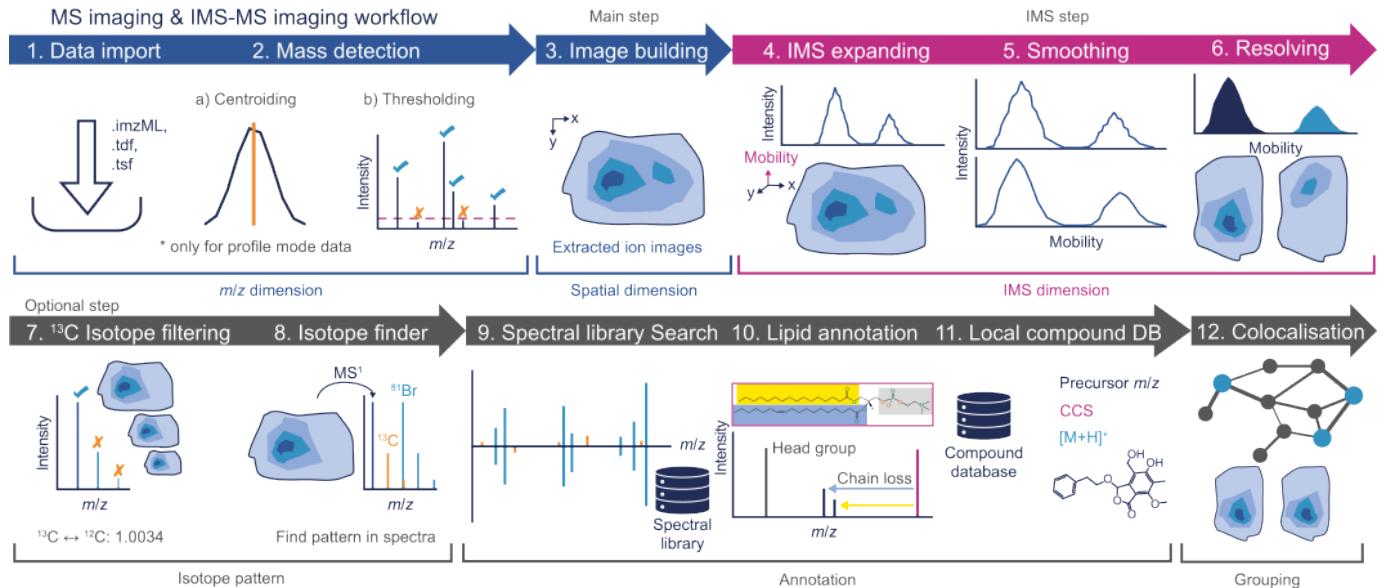
This workflow produces **Feature lists** as their result. Feature lists are a central part in visualizing the results in the **Statistics dashboard** and in the **Interactive network visualizer**.

13.4.1 Supported data formats

This workflow requires the spatial coordinates to be provided in the raw data file, which is the case in for the open-source imzML format. The native Bruker .d folders (with .tsf and .tdf data files) are also supported. Please note that ion mobility spectrometry (IMS) only supported in the native Bruker (tdf) format and not supported in imzML data.

13.4.2 Procedure overview

The general procedure is similar to the [LC-MS workflow](#). The raw data is imported and noise is removed ([raw data processing](#)) prior to [feature detection](#). Afterwards, multiple [filters](#) are available to refine the data.



13.4.3 Raw data import and processing

Raw data is imported by a simple drag-and-drop gesture to the MS data files tab in the main window (see [Raw data import](#)). After the data import, noise must be filtered from the raw data via the [Mass detection](#) module. Please note that this heavily impacts the performance of the whole workflow, since imaging spectra are usually richer in information than LC-MS spectra. If you experience performance (e.g., RAM issues), consider using a higher cutoff. IMS-MS imaging data sets require mass detection on the [Frame](#) and [mobility scan](#) level.

13.4.4 Feature detection

After mass detection, the feature detection is performed by the [Image builder](#).

When working with IMS-MS imaging datasets, the ion mobility dimension should be added by the [IMS expander](#) after the image detection. After expanding, the IMS dimension must be [resolved](#).

13.4.5 Isotope pattern and annotation

Isotope filtering, pattern finding, and feature annotation can be performed analog as described in the [LC-MS workflow](#).

⚠️ Warning

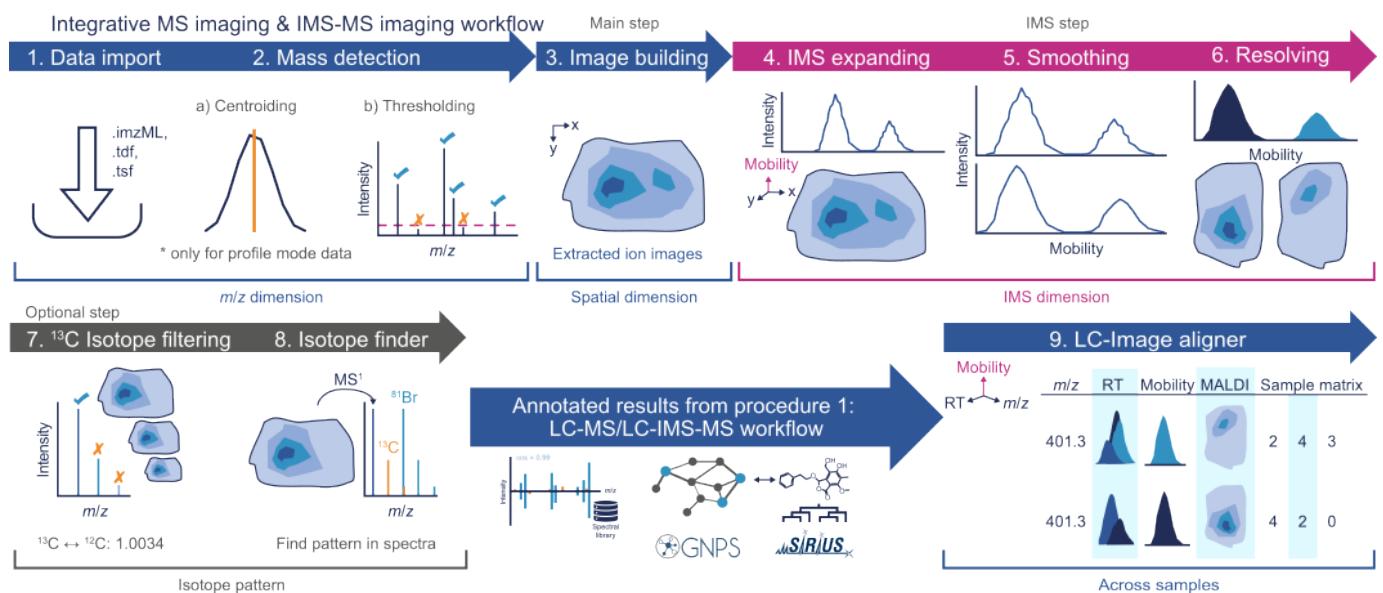
Tools, such as spectral library matching require MS2 spectra. Make sure your imaging data was acquired with MS/MS experiments.

13.4.6 Co-localization

Find co-located molecules using the [Image co-localization module](#).

13.4.7 LC-Image aligner

If an LC-MS dataset was acquired for the imaging sample, the results can be aligned using the [LC-Image Aligner](#). This allows integration of the two datasets and can be used for more confident identifications in imaging experiments. (see <https://www.nature.com/articles/s41587-023-01690-2>)



13.4.8 Feature filtering

After feature detection, the ion image features can be filtered to refine the results, for example by the [Feature filter](#) or the [Rows filter](#). Additional filters are found in the [Feature list methods → Feature filtering](#) menu. When using the deisotoping modules, consider that there is no chromatographic separation.

Ansgar Korf, Robin Schmid, Steffen Heuckeroth, kyrasolaris01

⌚ August 15, 2025 19:03:26

👤 Ansgar Korf, Robin Schmid, Steffen Heuckeroth, kyrasolaris01

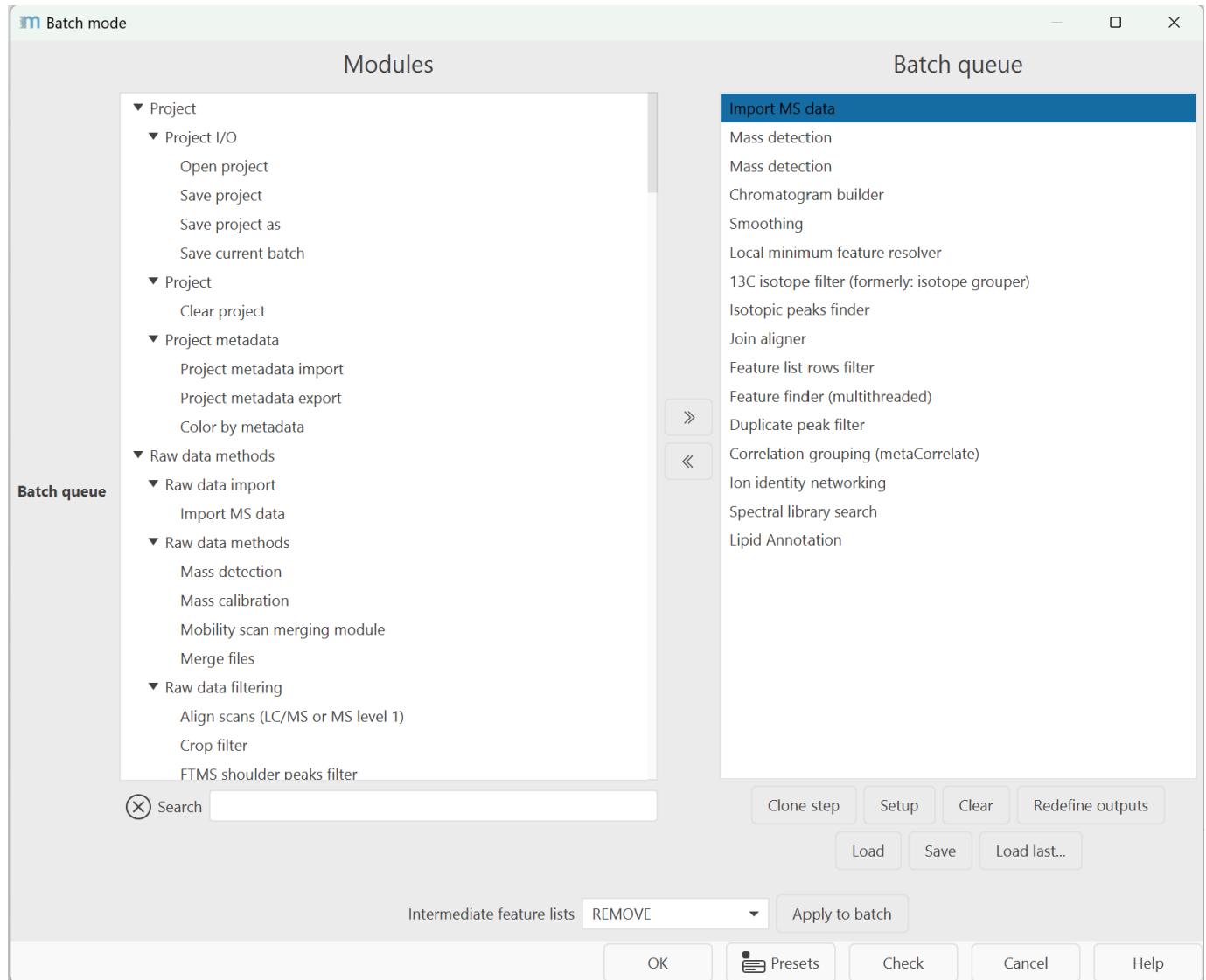
13.5 Batch mode processing

Besides the interactive GUI, mzmine allows the user to run processing workflows in an automated manner using the "batch mode". Entire processing pipelines (including data import/export) can be run with few clicks, or even through the command-line application. This makes MZmine suitable to be integrated into automated data analysis pipelines (e.g., QC systems).

Batch files (XML format) are essentially lists of tasks run by mzmine one after another. Any of the methods available in mzmine can be included in the batch file.

13.5.1 How to run batch processing

Project → Batch mode



When a new step is added to the queue its parameter setup dialog is shown. The "Set parameters" button allows the user to modify a step's parameter settings. The "Clear" button removes all steps. The "Load" and "Save" buttons make it possible to read and write batch steps to XML files.

Tip

Redefine outputs is a quick aid to set a new file path and base filename to all export modules (e.g., the feature list CSV export). Each step will add a suffix for the user to identify the output files. If one module exists multiple times, the suffix will also contain a numbering.

The first step of a batch queue is performed on those raw data files and/or feature lists selected by the user. The remaining steps are performed on the results produced by each preceding step (File/Feature list selection must be set to *Those created by previous batch step*). For example, if the first step of the batch queue is the [Chromatogram builder](#), it will produce feature lists as a result. If the following step were Peak list deconvolution then it will be performed on the peak lists produced by the preceding Chromatogram builder step.

⚠ Tip mzmine "remembers" the last settings used.

Tip

Intermediate feature lists is an option to apply the same handling of intermediate results to all batch steps. For example set all to keep during batch optimization or to remove/process in place, to run a final batch with maximum performance.

13.6 mzwizard

The **mzwizard** facilitates quick set up of general workflows for various sample introduction systems, (ion mobility) mass spectrometers, and workflows. This is the recommended way to configure workflows and create batch files. Batch files can then be modified further to improve results or to add more export steps.

mzwizard → Processing wizard



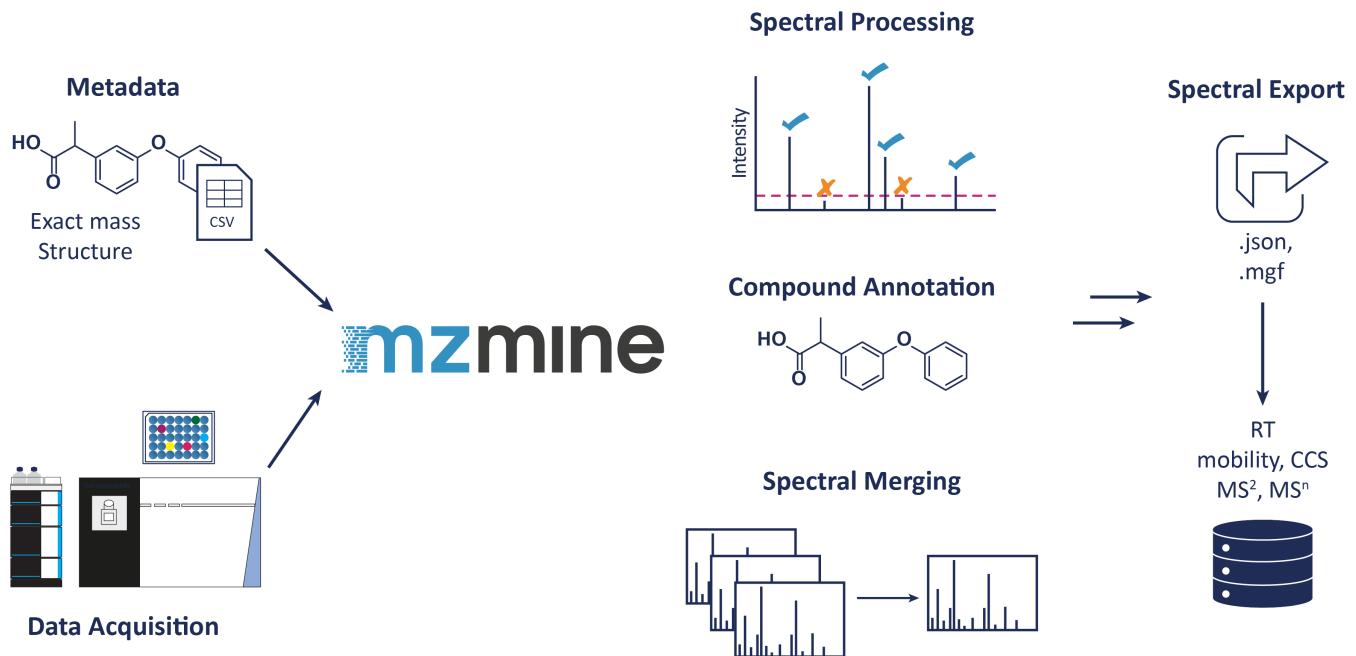
Robin Schmid, omokshyna, tdamiani

⌚ August 15, 2025 19:21:05

👤 Robin Schmid, omokshyna, tdamiani

13.7 Library generation workflow

This workflow is intended to extract library spectra automatically from your mass spectrometry data. The general idea is, to link the metadata to your mass spectrometry data. This facilitates the data processing at any time and enables the data extraction for each compound in the corresponding injection. Therefore, a unique sample id in the metadata table that is also used as part of the filename links the mass spectrometry data with the analyzed compounds, which is used during the mzmine processing.



13.7.1 Overview

The following steps are included:

- Metadata cleanup (Python script)
- Sequence creation (Python script for Thermo)
- Data acquisition
- Data conversion to mzML and centroiding (MSConvert, ThermoRawFileParser,...)
- Automatic library building workflow in mzmine by using the converted data files + cleaned Metadata
- Uploading of the raw data files, the library file(s) and the cleaned metadata

13.7.2 Related resources

- [Metadata Template](#)
- [Metadata cleanup](#)
- [Metadata upload](#)
- [Sequence creation](#)
- [mzmine batch example](#)

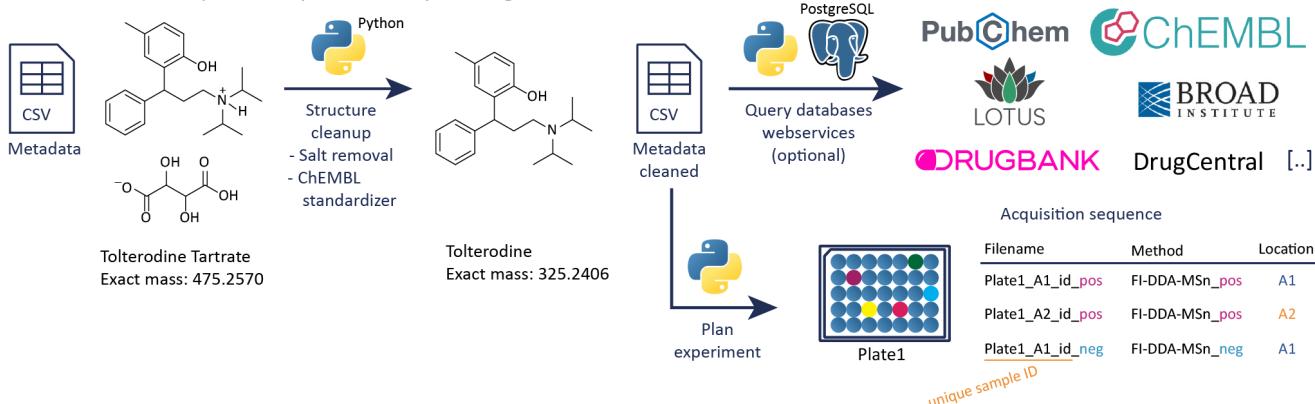
13.7.3 Providing of metadata

The metadata are used for further processing including a cleanup step, which means cleaning a given structure or searching for missing structures by a unique identifier, e.g., PubChem CID, or name, and adding further information by public database search. Therefore, we provide this [Template](#) to harmonize column names. The best would be to provide a compound name and the SMILES or InChI together with the rack or plate and well location. The library id can be used to link the dataset to a specific standard library or subset that you used. The unique sample id will be build during the metadata clean up to link the dataset with the metadata. This can be also the whole filename or part of it if data were already acquired. If a full filename is used, this

identifier needs to be changed if data are newly acquired and the filename changes. All other columns are optional and can be filled during the cleanup step.

Cleaning of metadata

a Metadata cleanup and acquisition sequence generation



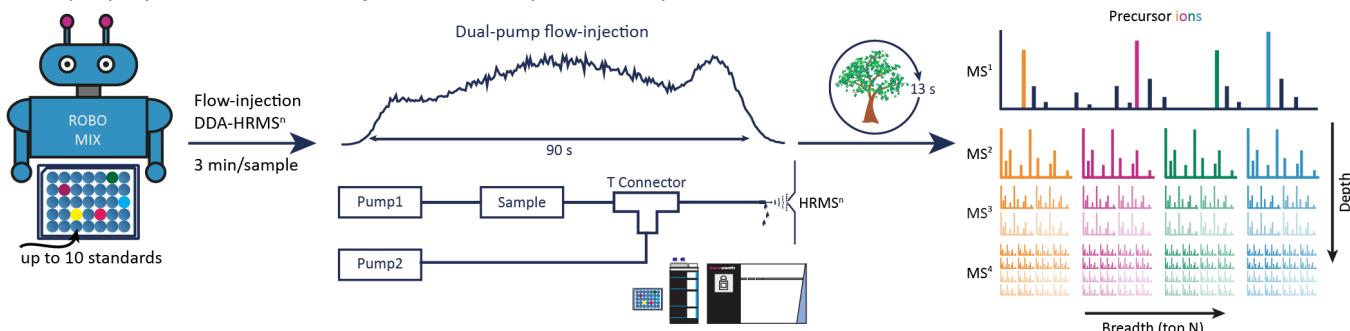
We provide a Python written metadata cleanup script to remove salts from SMILES structures and to search for structure information in PubChem if only a name or PubChem CID is given. Based on the cleaned SMILES, different structure information are computed, including the canonical and isomeric SMILES, SMARTS, the monoisotopic mass, InChI and InChIKey. Especially the monoisotopic mass is important for the automatic library generation workflow to match the measured mass against possible masses of the compound's adduct. Additionally, we need a unique sample id to search compounds only in the corresponding data file. This unique sample id is build by the library id, e.g., group+library, and the well/vial location including the plate/rack and ending with `_id`. Furthermore, we included searches in public databases to gather additional information about each compound, especially drug related information, e.g., in any clinical phase or indication. The script can be found on [GitHub](#).

Uploading of metadata and raw data

The library building is a non-static workflow. Accordingly, it is important to link raw data to their metadata to optimize the spectra extraction based on new knowledge, quality parameters and contamination in the future. Normally, library spectra are exported, maybe manually curated and stay as they are within the spectral database, without further knowledge about the processing behind. A repository to upload the metadata and raw data can be found on [GitHub](#).

13.7.4 Sequence creation for data acquisition

b Sample preparation and flow-injection-data-dependent acquisition MSⁿ

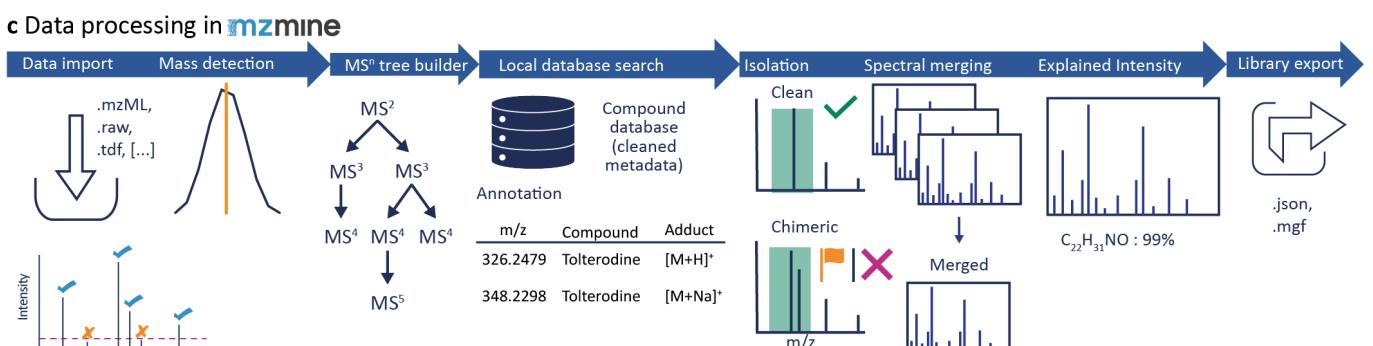


The shape of the data acquisition sequence table is vendor specific. We provide a python script for XCalibur Thermo and we will update it for further instruments. The columns should be named as in the [Metadata Template](#). For Thermo following information are needed:

- Data acquisition storing path
- Instrument method path (multiple polarities are possible)
- sequence will be first for one rack or well in one polarity and switch to the other in the second run
- Autosampler location for each well or rack
- Injection volume μL
- library identifier, for example group and used library
- Option:
 - Adding a method suffix, if different methods will be used for comparison, (e.g., changing AGC target, column,...)
 - Running blanks and QC after a specific number of injections (position of blank and QC needed)

The sequence is exported individually for each plate or rack and as a combined sequence. The combined sequence will have the Blank and QC in the beginning and end of the sequence and after the set number of injections. The data acquisition file names will be a combination of the date, unique sample id, the method suffix, and polarity. [GitHub](#)

13.7.5 Data processing in mzmine



Data should be converted to open data formats, e.g., mzML before importing in mzmine. This step guarantees the import at any time, since the vendor formats can change. For more information and possibilities go to [Data conversion](#).

In mzmine, open the Processing wizard to create your own batch configuration. Change the instrument settings to your instrument set up, e.g., Flow inject, and Orbitrap, and to **Library generation**. For more information, go to the [Processing wizard](#) page. The default settings are only suggestions and need to be updated and optimized for each dataset.

Under the **Annotation** tab you need to upload your metadata sheet (Local CSV database): - Tick the checkbox and Show - Set the path to your metadata file - Set if the precursor should be calculated by neutral mass (or structure)+adduct or the precursor m/z is stored in the table for each compound individually (important, if the adduct formation is very different in one dataset) - If you want to use the adduct calculation, the adducts can be selected after batch creation - Tick the filename column and type in the column name header - You need a unique sample id in the metadata table that is included in the filename. It can be the whole filename or just a part of it, so you can use the same table if only the date in your filename changes. If you use the metadata cleanup (creates the unique sample id) and the sequence creation, the filename is created by the date, unique sample id, method suffix and polarity. - The unique sample id should not end on a number, as this will enable false matches, e.g., A2 is also in A20, A200,... - Select all column you want to import - Enable the Neutral mass type in mzmine, if you want to calculate the precursors, otherwise use the Precursor m/z type - you can change the name by double click, changing the name, and pressing enter - To press enter is important as the name will not be saved otherwise

Import	Column name (csv)	Data type (MZmine)
<input checked="" type="checkbox"/>	monoisotopic_mass	Neutral mass
<input type="checkbox"/>	mz	Precursor m/z
<input type="checkbox"/>	rt	RT
<input checked="" type="checkbox"/>	formula	Formula
<input type="checkbox"/>	smiles	SMILES
<input checked="" type="checkbox"/>	inchi	InChI
<input checked="" type="checkbox"/>	inchikey	InChI key
<input checked="" type="checkbox"/>	compound_name	Compound
<input type="checkbox"/>	CCS	CCS / Å
<input type="checkbox"/>	mobility	Mobility
<input type="checkbox"/>	comment	Comment
<input type="checkbox"/>	adduct	Adduct
<input type="checkbox"/>	PubChemCID	PubChemCID

Go to **Library generation** tab: - You can add specific information to your library, e.g., what was the library, which instrument, who acquired the data, or who is the PI,...) - Set the **Export path** to store the extracted library

After you set everything, you can save the presets. You can save the library method separately from the sample introduction and instrument, to stitch the library settings together with any other instrument setup. Click the **Create batch** button, which opens a new window. Every algorithm is prefilled by the wizard settings.

Go to the **Custom database search** to change more library settings, especially possible adducts. Go to the **Batch spectral library generation** to set the format and the MS level for the export. Furthermore you can set additional quality settings, including the handling of chimeric spectra, minimum number of signals and minimum number of explained signals in each scan based on all subformula combinations. For more information about each parameter go to [Custom database search](#) and [Batch spectral library generation](#)

During the spectral library export, all set quality parameter are checked and only spectra that fulfill all set criteria are exported. For chimeric spectra, instead of discarding those, they can be flagged as chimeric.

⌚ June 12, 2024 08:49:50

👤 corinnabrungs

13.8 SIMSEF Workflow

13.8.1 Workflow description

The spatial ion mobility-scheduled exhaustive fragmentation (SIMSEF) module can be used in trapped ion mobility (TIMS) imaging workflows to schedule fragmentation spectra acquisitions for every detected feature across the tissue. This achieves high MS2 coverage and allows more confident compound annotations. To use the SIMSEF tool, the prmMALDI prototype instrument control software is required, which is distributed by Bruker Daltonics to cooperation partners. You can contact your Bruker representative to ask for the prmMALDI prototype and refer to SIMSEF.

13.8.2 Requirements

- access to timsTOF fleX instrument
- timsControl 6.1
- [Download](#) SIMSEF acquisition tool for the respective timsControl version
- Note: while SIMSEF versions for timsControl 4.1, 5.0, and 5.1 exist, we recommend using timsControl 6.1 as earlier versions require access to prototypic timsControl software, which is not supported anymore

13.8.3 Procedure



Note:  acquire MS1 data with a smaller laser spot size than raster size (= stage movement/pixel size)

 **Warning**

Note:  Due to the size of IMS-MS imaging data, we recommend to use 10,000-40,000 pixels in your MS1 run for SIMSEF experiments

- this leaves some tissue for MS2 acquisition
- Prepare the sample following your default procedure.
- Setup the imaging run in flexImaging. Set the raster size bigger than the laser spot size in your imaging method (e.g. laser spot size = 20 µm, raster size = 50 µm).
- Save the imaging run via "Save run file as..." ("Save imaging run as..." prior to tC 6.1). This will store the geometry files for later use.
- Restart timsControl (prm-MALDI prototype) to make the instrument recognize the geometry files. Recalibrate the instrument (recommended).
- Start the MS1 imaging run.
- Copy the raw data to a powerful data analysis machine (Windows and Linux support Bruker raw data)
- Analyse the MS1 data and filter it to your liking (see [imaging worflow](#)). If your IMS separation allows, filter for [isotopic peaks](#).
- Run the [SIMSEF scheduler](#). Using the preview, evaluate the fragmentation schedules and optimise the parameters. Check the task manager for running tasks. The scheduling may need some time.
- In SIMSEF for timsControl versions 4.1, 5.0, and 5.1 the path of the exported schedule must not contain any spaces and must be the same on the analysis and the instrument computer
- In SIMSEF for timsControl 6.1 this limitation does not exist
- Copy the schedule to the instrument computer. (**Must be the same path as selected on the analysis computer, not including spaces or special characters**, unless you are using timsControl 6.1)
- Load the instrument method, you want to acquire the MS2 spectra with. Make sure the method is appropriate for your expected fragment ion m/z range.
- Disable MALDI mode, enable PASEF mode, ensure that "advanced" is disabled for the collision energy settings.
- Enable MALDI mode, save the method.
- Set MS mode to MS/MS in the MALDI settings. (prmPASEF in timsControl 6.1 and higher)
- Increase the laser size to the raster size
- Set instrument to operate and recalibrate if needed (recommended).
- Select the correct geometry file on the instrument computer (saved in step 3)
- If the geometry selection is disabled by the target geometry (due to barcode reading), disable this feature at: Preferences - System Settings - MALDI Settings
- Run "simsef_pewpew.exe" on the instrument computer, select the acquisition.txt and click "Run acquisition."
- After the acquisition, copy the folder to the analysis computer.
- [Import all MS2 files](#)
- Run [Mass detection](#) with the centroid mass detector and noise level 50-100 (starting point, for MS2 only!)
- Pair the MS2 spectra to the image features using the [Assign MALDI MS2s to features](#) module
- Annotate using any annotation module in mzmine

[Steffen Heuckeroth](#), Steffen Heuckeroth

 November 6, 2025 14:31:20

 Steffen Heuckeroth, Steffen Heuckeroth

13.9 MRM workflow

Warning

To process multiple reaction monitoring (MRM) data, the MRM service of mzmine is required. This feature is included in the mzmine PRO subscription. Academic program and other users can reach out to inquire [access options](#).

In contrast to HRMS data, MRM acquisitions are usually stored as chromatograms of specific transitions, e.g., m/z 413 -> 96 instead of full spectra. This is a raw data format, that mzmine is not used to handling, since workflows usually start with a set of centroided spectra along the retention time.

Therefore, a specific module is required to build scans and chromatograms, which is the [MRM to scans](#) module.

13.9.1 Workflow

Import raw data

Raw data is import in the usual fashion see [data import](#).

MRM to scans

The imported raw data is converted using the [MRM to scans](#) module. This creates pseudo MS1 and MS2 spectra as well as a feature list, which groups the MRM transitions by their retention time and precursor ion mass.

Set quantifier masses

mzmine automatically selects the quantifier mass based on the intensity of the MRM transitions and sets the most intense ion as a quantifier. If other ions shall be used as quantifiers, this module allows the selection. [mrm-set-quant-mass.md](#)

Smoothing in retention time dimension (optional)

Depending on the LC peak shape (i.e. data noisiness), the user can perform smoothing in retention time dimension.

Feature resolving

Feature resolving step enables separation of co-eluting and overlapping chromatography peaks and as such is one of the pivotal steps in data preprocessing. For more details on the algorithm used and parameters settings, see the [Local minimum resolver](#) module.

Feature alignment

mzmine offers a specific alignment module for MRM data. The [MRM aligner](#) takes the transitions of a feature into account and only allows alignment of features with the same transitions (in default settings). In principle, MRM feature lists can be aligned to HRMS feature lists using the [Join aligner](#). For downsides of this procedure, see the [the MRM aligner documentation](#)

[Robin Schmid, Steffen Heuckeroth](#)

 March 6, 2025 09:17:43

 [Robin Schmid, Steffen Heuckeroth](#)

13.10 Workflow optimization

13.10.1 General remarks for mzmine workflow optimization

The mzwizard builds modular workflows, so-called "batches", made up of individual steps. Each step applies to the results of the previous step. If your results do not contain e.g. specific features you were expecting, you have to pinpoint which step omits the expected feature(s).



We recommend to apply the optimization to a representative subset of your dataset. A good starting point are usually pooled QC or other representative samples.

Making your workflow "optimization ready"

To pinpoint which steps need to be optimized, you must first keep the **intermediate results** of each step.

Do either:

1. In the wizard: in the [mzwizard](#) tab, set the [Original feature lists](#) option to "Keep".
2. In the [batch window](#): Set the "Intermediate feature lists" to "Keep" and press "Apply to batch".

Is my benchmark feature in the feature list?

To quickly assess if a specific feature is in your feature list, you can use the [Feature list filters](#) in the feature table. Enter a specific m/z, or m/z range and/or RT or RT range to search.

Determine which step to optimize

To determine which step you need to optimize, check the name of the first feature list, that does not contain your benchmark feature anymore. For example, we may have two feature lists called "*171103_PMA_TK_QC_03.mzML eics sm*" and "*171103_PMA_TK_QC_03.mzML eics sm r*". The former still contains our benchmark feature, while the latter does not.

Most processing steps create a new feature list and append a suffix to the name of the previous feature list. These suffixes may be user-defined, but the wizard uses suffixes that refer to the name of the individual steps. (Set in the "Suffix" or "Name suffix" parameters in each module)

In our example, the latest suffix is "r", which refers to the [Local minimum resolver](#). This means that our benchmark feature is lost in this step.

Where do I optimize the workflow?

Since we have now pinpointed the problematic step, we can now optimize it using either the wizard or the batch mode.

It must be noted that you can optimize every step in the batch mode, but not in the wizard, since the wizard gives access to a subset of parameters and provides reasonable defaults for others. Where possible, this document will detail approaches to optimize the step using the wizard and the individual steps.

While the wizard does not offer insight into the effect of adapted parameters without running it, the optimization of individual steps in the batch mode does offer previews.

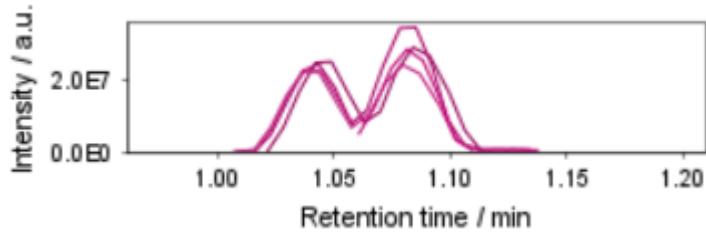
Danger

Optimizing the created batch only affects the specific batch you are currently working on. If you close the batch dialog and create a new batch using "Create batch" in the wizard, previous batch optimizations are overridden. We recommend saving the optimized batch file, or saving presets (since mzmine 4.7.29) for the optimized modules, so you can apply the module preset to new batches.

13.10.2 Common optimizations

Coeluting features not split

Problem: Some features have show coelution with isomers and/or isobars.



Note: This may have different causes. Either the expected peak width was too high, or the wizard's default resolution setting is too high.

Solution 1: Reducing the expected peak width.

- In the wizard: Reduce the "Approximate feature FWHM" in the HPLC/UHPLC/HILIC tab.
- In the batch: Reduce the [Minimum search range](#) in the [Local minimum resolver](#)

Solution 2: Reducing the expected peak resolution. (if Solution 1 does not achieve the expected result)

- In the wizard: Not possible
- In the batch: Reduce the [Min ratio of peak top/edge](#) in the [Local minimum resolver](#)

Note: Optimization of peak resolution is a crucial step. If parameters are chosen too loosely, the quality of the peak detection may be reduced severely.

Feature not detected

Problem: A specific feature is not detected in any feature list, even in the most basic one.

Solution 1: Reducing the minimum intensity requirements, if the feature is of low intensity.

- In the wizard: Lower the [Minimum feature height](#) in the mass spectrometer tab.
- In the batch: As the wizard parameter influences multiple steps, all these need to be updated:
 - Minimum absolute height in the [Chromatogram builder](#)
 - Minimum absolute height in the [Local minimum feature resolver \(RT and IMS dimension\)](#)

Solution 2: Reducing the minimum number of data points per feature, if the feature is detected in few consecutive scans.

- In the wizard: Lower the [Minimum consecutive scans](#)
- In the batch: As the wizard parameter influences multiple steps, all these need to be updated:
 - Minimum consecutive scans in the [Chromatogram builder](#)
 - Minimum scans (data points) in the [Local minimum feature resolver \(RT and IMS dimension\)](#)

⌚ October 16, 2025 12:47:54

👤 Steffen Heukeroth

14. Tools

14.1 Spatial ion mobility-scheduled exhaustive fragmentation (SIMSEF)

14.1.1 Description

The spatial ion mobility-scheduled exhaustive fragmentation (SIMSEF) module can be used in trapped ion mobility (TIMS) imaging workflows to schedule fragmentation spectra acquisitions for every detected feature across the tissue. This achieves high MS2 coverage and allows more confident compound annotations. To use the SIMSEF tool, the prmMALDI prototype instrument control software is required, which is distributed by Bruker Daltonics to cooperation partners.

Note: ! Due to the size of IMS-MS imaging data, we recommend to use 10,000-40,000 pixels in your MS1 run for SIMSEF experiments

14.1.2 Parameters

Feature lists

Select the feature list to create the MS2 events for.

Data location

File path to store the schedule in. This directory must also exist on the data acquisition PC and not contain spaces or special characters.

Path to msmsmaldi.exe

Path to the simsef_pewpew.exe

NUMBER OF MS2 SPECTRA

Sets the desired number of MS2 events to be scheduled per precursor and collision energy.

COLLISION ENERGIES

Lists the collision energies to acquire MS2 spectra for every precursor.

MINIMUM DISTANCE OF MS2 PIXELS

Defines the spatial pixel-based distance between MS2 events of the same precursor ion and with the same collision energy.

MINIMUM MS1 INTENSITY

Describes a relative and absolute intensity threshold for the precursor in an MS1 pixel to be selected for MS2 experiments. Specified as absolute and relative parameter, which will use the larger of the two values with regard to the feature.

MINIMUM PURITY SCORE

Defines an exclusion criterion for scheduling MS2 events in a pixel, in case the quadrupole isolation would lead to chimeric MS2 spectra composed of multiple precursor ions.

EXPORT MSMS LISTS ONLY

Exports only the MS2 lists and does not start an acquisition.

Advanced parameters

QUADRUPOLE SWITCH TIME (MS)

Defines the assumed switch time of the quadrupole. This influences how close multiple precursors will be scheduled. Low values may lead to co-isolation and overlapping of precursor values.

ISOLATION WIDTH

Sets the quadrupole isolation window in the instrument parameters and influences the purity score calculation.

MINIMUM MOBILITY WINDOW AND MAXIMUM MOBILITY WINDOW

Describe the minimum and maximum length of a precursor isolation width in the mobility dimension. The features detected mobility range will be cropped or extended to be within the minimum and maximum mobility window.

14.1.3 Output

The scheduling module creates a folder including an "acquisition.txt" including the acquisition table, collision energy table(s), precursor lists for every spot, and "precursors.csv" and "spots.csv" files. The "precursors.csv" and "spots.csv" files allow statistical evaluation of the expected MS2 coverage prior to the analysis.

To acquire the MS2 spectra, the whole folder must be copied to the acquisition computer to the location specified as the "data location" parameter.

14.1.4 References

[1] Wolf, C. et al. Mobility-Resolved Broadband Dissociation and Parallel Reaction Monitoring for Laser Desorption/ionization-Mass Spectrometry - Tattoo Pigment Identification Supported by Trapped Ion Mobility Spectrometry. *Anal. Chim. Acta* 2023, 1242, 340796.

doi: <https://doi.org/10.1016/j.aca.2023.340796>

[2] Heuckeroth, S. et al. On-tissue dataset-dependent MALDI-TIMS-MS2 bioimaging. *Nat Commun* 14, 7495 (2023).

doi: <https://doi.org/10.1038/s41467-023-43298-9>

Steffen Heuckeroth

⌚ November 30, 2023 12:41:35

👤 Steffen Heuckeroth

15. Additional resources

15.1 Installation

15.1.1 Installation on Microsoft Windows

1. Download the latest mzmine Windows installer (.msi)
2. Double-click the installer to run the installation
3. Windows might ask if you trust mzmine, select yes.
4. Run mzmine by double-clicking the mzmine.exe (can be found by pressing the Windows button and searching for mzmine)

Updating mzmine

There is no need to deinstall mzmine, a new version can be installed over an older version. Only in rare cases mzmine might ask to remove an old version before installing a new version.

The mzmine configuration, presets, and other files are kept and shared between versions.

 November 24, 2025 20:48:11

 Robin Schmid

15.1.2 Installation on Linux

Download the latest version, install mzmine, login, and run mzmine. See mzmine [command-line interface](#) as a reference.

```
# with gh (github) installed, download of latest .deb installer is quite easy
# gh auth login
# sudo apt install gh
# gh release download --repo mzmine/mzmine --pattern "mzmine*.deb"

# or find installer at https://github.com/mzmine/mzmine/releases/latest
wget https://github.com/mzmine/mzmine/releases/download/text-action-release/mzmine_4.3.1_amd64.deb

# create required dir and install mzmine
sudo mkdir -p /usr/share/desktop-directories/
sudo apt install mzmine*.deb

# potential dependencies that may be required
# sudo apt-get install xdg-utils
# sudo apt-get install libglib1
# sudo apt-get install libgtk-3-0
# sudo apt-get install libxtst

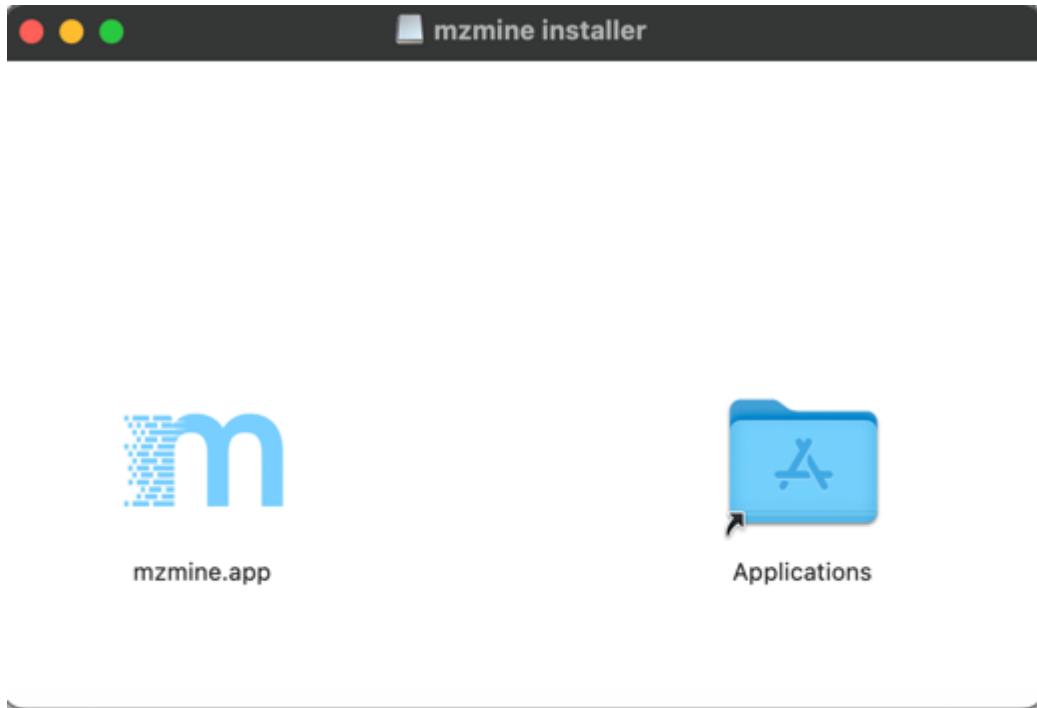
# run mzmine and print help. also check -login-console -batch
/opt/mzmine/bin/mzmine -help
```

⌚ November 24, 2025 20:48:11

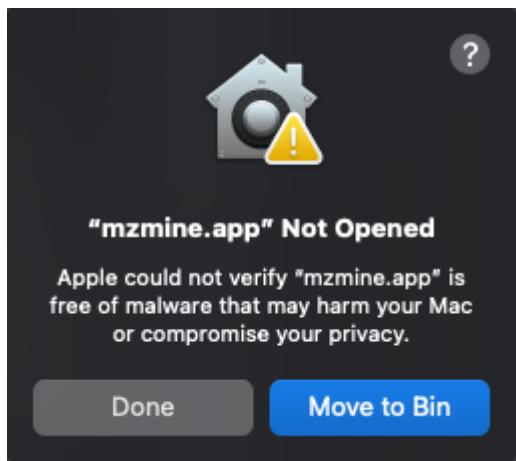
👤 Robin Schmid

15.1.3 Installation on macOS

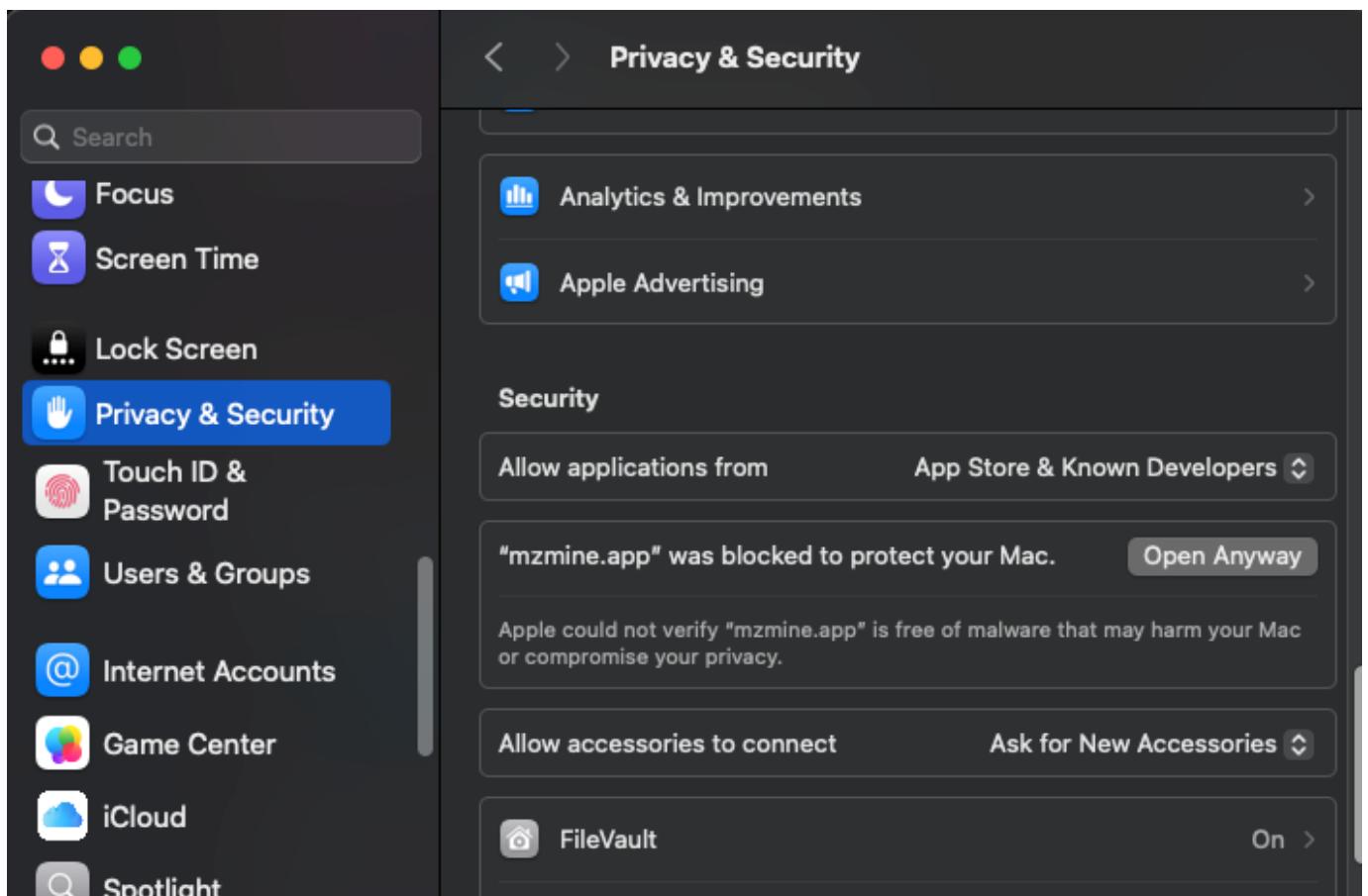
Download the latest version of the mzmine_macOS_installer_academia.dmg. Double click to start installation and drag and drop mzmine.app into the Applications folder.



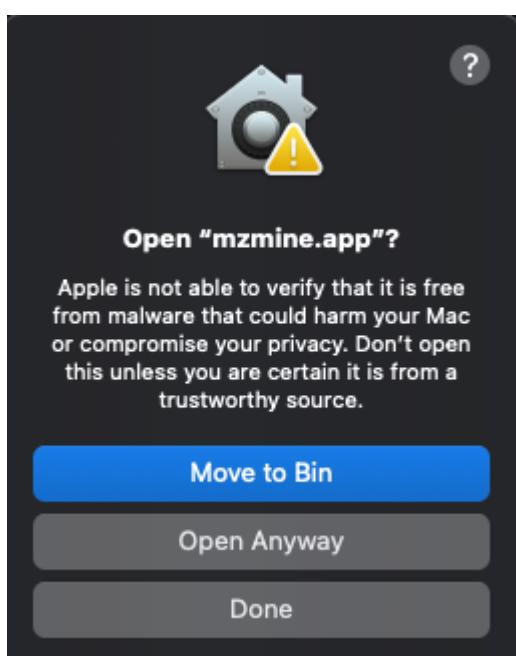
Select "replace" to override older versions. First time starting mzmine may trigger a warning "mzmine.app" Not Opened on macOS 15 or later.



To open mzmine, go to System Settings, scroll down to Privacy & Security, and click on "Open Anyway" next to the message "mzmine.app" was blocked to protect your Mac.



A new message pops up. Click again on "Open Anyway" and enter your macOS password.



This procedure only has to be done once. Now, mzmine can be started like other applications.

⌚ November 24, 2025 20:48:11

👤 Robin Schmid

15.2 System requirements

15.2.1 Installation

Installation of mzmine is described on the [getting started](#) page. mzmine is available as an installable package or a portable version. The portable version does not require administrator rights to be run, making it useful for university students without elevated permissions.

15.2.2 Hardware requirements

Minimum (small datasets < 30 LC-MS files):

- 64 bit CPU, 4 Cores (2.5 GHz, Intel Core i5 or AMD Ryzen 5), 16 GB RAM, secondary SATA SSD drive (512 GB), integrated graphics
- Display 1920 x 1080 pixel (Full HD)
- Internet connection for login and spectral library/machine learning model download
- Keyboard and mouse

Recommended (medium to large data sets ≥ 100 LC-MS files, or IMS-MS in general):

- 64 bit CPU, ≥ 16 Cores (≥3 GHz, Hyper threading), ≥ 64 GB RAM (scales with data files), secondary NVMe SSD drive (≥ 1 TB), integrated graphics
- Display 2560 x 1440 pixel or higher
- Internet connection for login and spectral library/machine learning model download
- Keyboard and mouse
- dedicated GPU (CUDA enabled) for accelerated machine learning models (not required)



Offline login is possible, see [Offline use](#).



Processing speed scales with CPU cores/threads and speed of the SSD for temporary files. We recommend setting the temporary files directory to a fast, secondary SSD in the mzmine preferences (CTRL+P, **Project -> Preferences**). (see)

15.2.3 Software requirements

- Up-to-date operating system, e.g., Windows 10 or newer, recent Linux or MacOS (academic only) versions
- mzmine does not require a dedicated Java installation, even though it is a Java software. All requirements are shipped with mzmine
- Microsoft Visual Studio C++ Redist for Bruker raw data import [download page](#)
- MSConvert (on Windows) for native Sciex, Waters, Shimadzu, MOBILion, Thermo data support [download page](#)
- Thermo alternative: ThermoRawFileParser for native Thermo support on Windows, Mac, and Linux [download page](#)
- ThermoRawFileParser does not need to be installed but only downloaded and imported via the mzmine preferences

15.2.4 Internet connection

- An internet connection is recommended, but not strictly required for core processing

- Offline user login is possible, see [Offline use](#).
- To allow mzmine to download spectral libraries and the recent versions of machine learning models for spectral networking using MS2DeepScore and DReaMS, an internet connection is required
- A proxy may need to be set in the **Preferences** if your University/Company uses one (**Project -> Preferences**, CTRL+P)
- Required URLs (in case University/Company blocks unknown):
 - <https://auth.mzio.io/> must be accessible user login
 - <https://zenodo.org/> machine learning models
 - <https://djl.ai/> machine learning models
 - <https://zenodo.org/> spectral libraries
 - <https://external.gnps2.org/gnpslibrary> spectral libraries

 November 25, 2025 07:48:04

 Robin Schmid, Steffen Heuckeroth

15.3 General terminology

15.3.1 MS

Precursor and fragment ions

The **precursor ion** (or "parent ion") is the selected ion used for MS/MS experiments.

A **fragment ion** ("daughter ion" or "product ion") is the charged product of an ion dissociation. A fragment ion may be stable or may dissociate further to form other charged fragment ions and neutral species of successively lower mass.

Accurate mass, exact mass and mass accuracy

The **accurate mass** is the experimentally-determined mass of an ion measured with a high-resolution mass spectrometer.

The **exact mass** is the calculated mass of an ion based on its elemental formula, isotopic composition and charge state. While the accurate mass is an experimentally-measured quantity, the exact mass is a theoretically-calculated quantity.

The **mass accuracy** is defined as the difference between the measured value (accurate mass) and the true value (exact mass). It can be expressed either in **absolute (mDa)** or **relative (ppm)** units.

Monoisotopic mass

Exact mass of an ion calculated by using the mass of the naturally most abundant isotope of each element.

Isotopic pattern

Isotopic (or isotope) pattern describes a set of peaks related to the ions with the same chemical formula but containing different isotopes; e.g. the 16 and 17 mass/charge peaks in a CH₄ sample arising from ¹²CH₄⁺ and ¹³CH₄⁺ ions.

Mass resolution

Resolution describes an ability of mass spectrometers to distinguish two peaks of different mass-to-charge ratios. Can be interchangeably used with [mass resolving power](#)

[Wikipedia article on MS resolution](#)

Mass resolving power

In a mass spectrum, the observed mass divided by the difference between two masses that can be separated, m/Δm.

Data acquisition mode

Process of sampling to capture the signals. Different modes have been introduced to better capture signals after LC separation, especially in metabolomics. In MS data can be acquired using three main modes:

- [Full scan acquisition mode](#)
- [Data-dependent acquisition mode](#)
- [Data-independent acquisition mode](#)

Read more: [Comparison of data-dependent and data-independent modes](#)

Full scan acquisition mode

In full-scan mode, the mass spectrometer runs on MS1-only mode, and measures m/z values and abundances of all the metabolic features.

Widely used as it allows to capture most of the relevant ions.

Further confirmation of statistically significant features is typically carried out by a separate LC-MS/MS run in a targeted manner.

Data-dependent acquisition mode (DDA)

Mode of the data collection in **tandem mass spectrometry**. In data-dependent acquisition (**DDA**) schemes, the mass spectrometer detects 'suitable' precursor ions in each MS scan and selects them for fragmentation in consecutive MS2 scans.

! DDA can redundantly identify high-abundant features, while neglecting low-abundant ones.

TopN acquisition scheme

In TopN scheme, the set of N ions is selected for fragmentation by their intensity in the latest MS survey scan.

Data-independent acquisition mode (DIA)

DIA can be conducted either by fragmenting all ions that enter the instrument at a given time (called broadband DIA) or by sequentially focusing on a m/z window of precursors and fragmenting all precursors detected within that window.

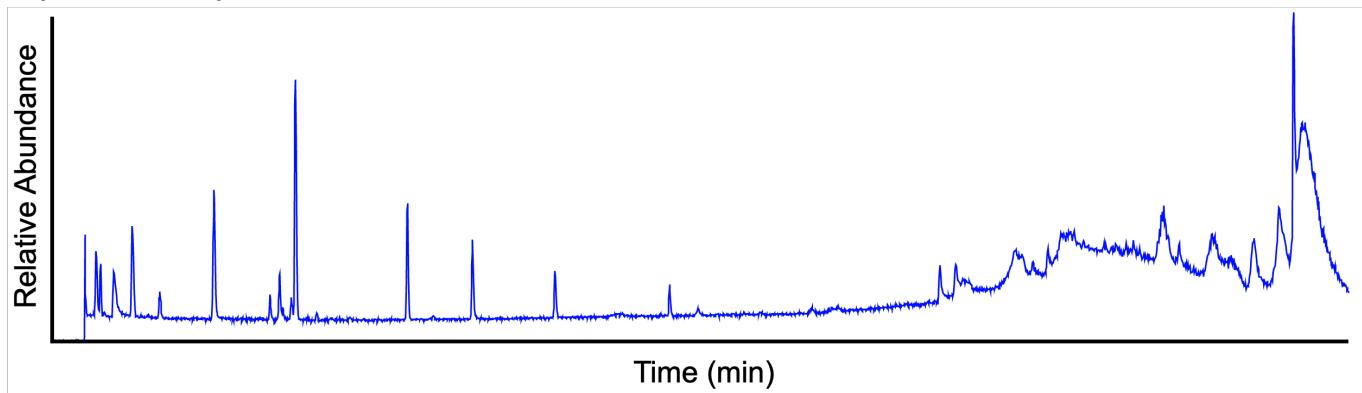
Cycle time acquisition scheme

In cycle time acquisition, a set of precursor ions is selected using m/z values (usually 1.0 to 2.0 m/z range). A full MS/MS fragment ion spectrum is collected for each ion. Cycle time is determined by scan times of all scans in the set.

15.3.2 LC-MS

Total ion current chromatogram

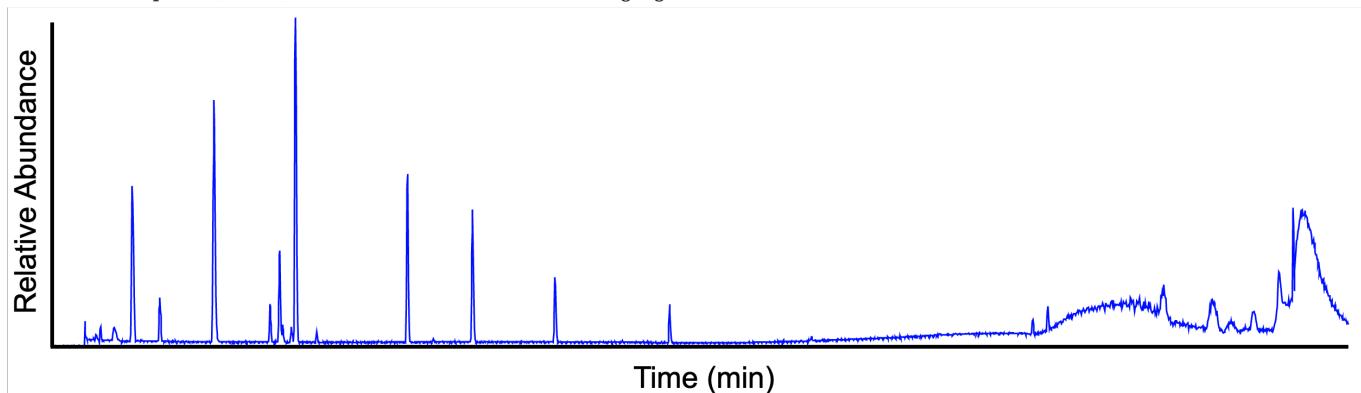
The total ion current (TIC) chromatogram displays the summed signal intensity (y-axis) over the entire m/z range at any one retention time point (x-axis) in the LC-MS run. The following figure shows a TIC chromatogram of a 9-compounds mixture analysed on LC-MS system.



! In complex samples, the TIC chromatogram often provides limited information as multiple analytes elute simultaneously, obscuring individual species.

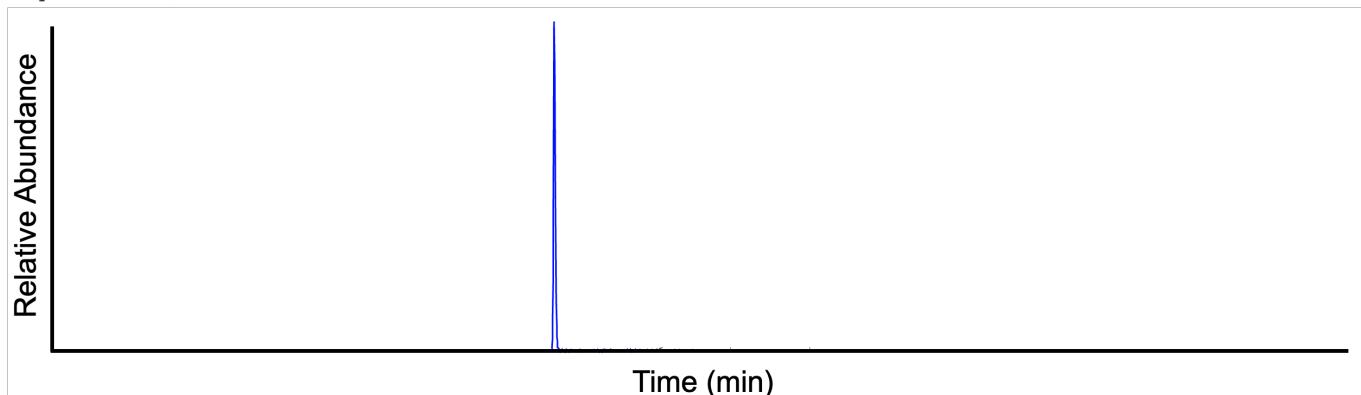
Base peak chromatogram

The base peak chromatogram (BPC) displays the signal intensity of the most intense mass peak in the MS spectra at any one retention time point (x-axis) in the LC-MS run. The following figure shows the same data as above, visualized in BPI mode.



Extracted ion chromatogram

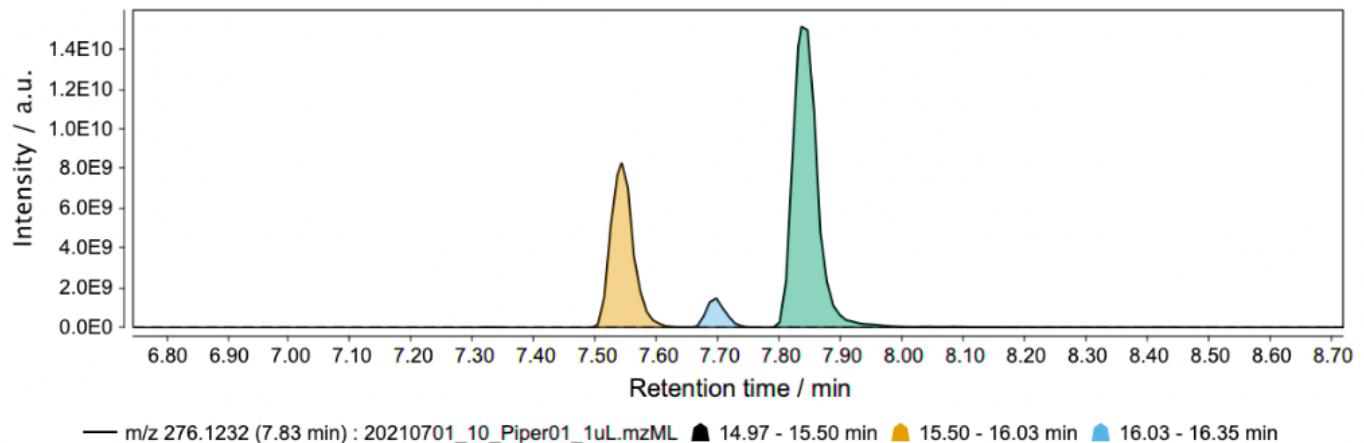
The extracted ion chromatogram (EIC) displays the signal intensity of a specific m/z value, within a defined tolerance (e.g. ± 5 ppm), at any one retention time point in the LC-MS run. The following figure shows the EIC of m/z 455.2945 ± 5 ppm (same sample as above).



Chromatographic resolving

Peak overlapping, or co-elution, is a common problem in any chromatographic separation technique. In the case of LC-MS (especially untargeted *omics* analysis), it is virtually impossible to obtain a full baseline separation for the hundreds (or thousands) of analytes eluted through the column. The split of partially-overlapping and shoulder peaks into individual features is generally referred to as *chromatographic resolving* and is one of the most crucial steps of data processing. (see [Local minimum resolver](#))

Resolved features (= EIC split into features):



Missing values

MISSING VALUE DEFINITION (see <https://www.nature.com/articles/s41598-017-19120-0>).

Missing values can be caused by biological and/or technical reasons. A gap (match-to-nothing) is defined as a missing value in an alignment. Peaks that cannot be aligned (corresponding to a gap) are designated with a value of "NA" (missing value).

Missing value imputation

Missing values are generally undesired as they might introduce bias and significantly affect downstream data analysis, especially when statistics is involved. Therefore, it is important to REDUCE THEM.

To tackle this issue, a value for the peak needs to be imputed. The [peak finder](#) gap-filling module.

A simple gap-filling approach is to integrate the area where the peak is expected but not detected. These areas usually correspond to spectral noise. By doing so, no bias is introduced. Therefore, we recommend to always run the gap filling module before the missing value imputataion.

The other alternative for gap filling is the [Same RT and m/z range gap filler](#), which limits the gap fill to features within the original detected peak window. This leads to much cleaner results. ⚠ However, this module causes MISSING VALUE IMPUTATION. For purposes of further data processing, noise should be always preferred to zero values.

💡 mzmine does not support missing-value imputation.

15.4 mzmine-specific terminology

Masses and Features

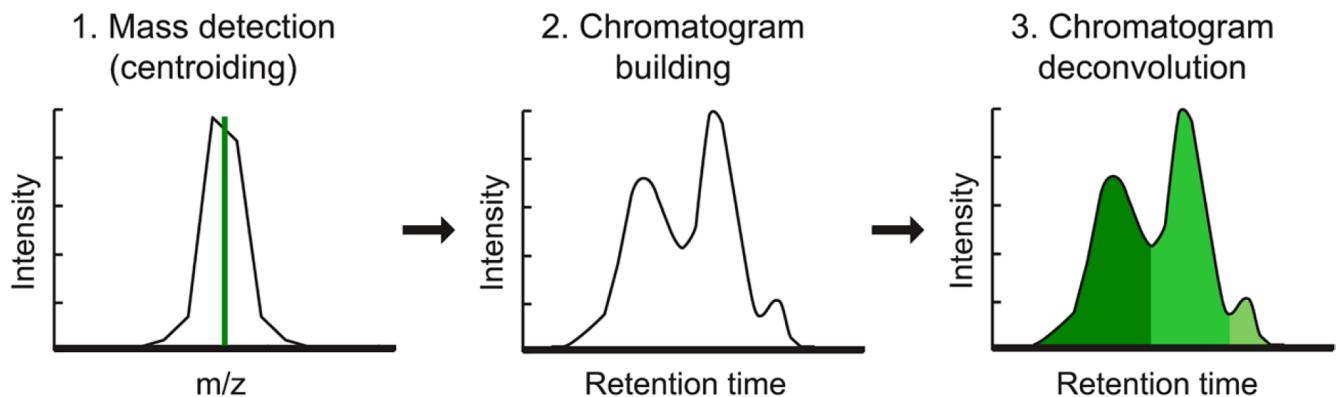
In MS data processing, the term **mass** is normally used to refer to an individual signal in a mass spectrum, which corresponds to an ion detected by the mass spectrometer (see [Mass detection](#)).

In LC-MS, a **feature** is defined as a bounded, two-dimensional (*m/z* and RT dimensions) signal characterized by a pair of *m/z* and RT values and associated with the detected signal intensity.

In LC-IM-MS, a **feature** is also characterized by the ion mobility value recorded for the ion (see [LC-MS and LC-IMS-MS data comparison](#)).

mzmine provides a selection of different algorithms for LC-(IM)-MS feature detection, depending on the nature of the MS data (*e.g.* mass accuracy and resolution). All the algorithms follow the same logic:

- EICs are constructed starting from each m/z value in the mass lists
- Then, EICs are subsequently deconvoluted into individual features (see figure).
- Finally, additional information, such as isotope pattern, adduct type, *etc.* can be assigned to the individual features.



Mass list

In mzmine, we call **mass list** the output of the [mass detection](#) module.

A **mass list** is a list of m/z values and corresponding signal intensities, found in each mass spectrum (MS or MS_n) of each processed raw data file.

Every mass spectrum contained in the raw file is processed individually. The signals exceeding the set noise threshold are included in the mass list. See [Mass detection](#) module.

Feature list

In mzmine, **feature lists** are the output of the **feature detection** process (see [Masses and features](#)).

The set of detected features in each LC-MS run is stored as a list, hence the name "feature list" (see, for example, [Chromatogram builder](#) and [Local minimum resolver](#) for more details). Multiple feature lists can undergo further processing (*e.g.* feature alignment) which results in a table (often referred to as **feature table**) where samples are arranged in columns, features in rows and each entry contains the signal intensity detected for the corresponding feature in the corresponding sample.

m/z tolerances

- scan-to-scan
- intra-sample
- sample to sample

The m/z tolerance is defined as maximum allowed difference between m/z values in order to be considered the same or belonging to each other (isotopes or different adducts). The **scan-to-scan m/z tolerance** is used for values between different scans (in, *e.g.*, [Chromatogram Builder](#)). The **m/z tolerance (intra-sample)** is used for features in the same scan that belong together (used, *e.g.*, in ^{13}C isotope filter). The **m/z tolerance (sample-sample)** takes the fluctuation of the same feature in different samples into account. This tolerance is used in the [Join aligner](#)).

Chromatogram resolving

Was referred to as **Deconvolution** in MZmine 2. Process of splitting "imperfect" - overlapping and partially co-eluting - peaks, which are retained as single features, into the separate features.

15.4.1 References

- Pluskal, T., Castillo, S., Villar-Briones, A. & Oresic, M. MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics* (2010). DOI: [10.1186/1471-2105-11-395](https://doi.org/10.1186/1471-2105-11-395)
- Pluskal, T. et al. Processing Metabolomics and Proteomics Data with Open Software: A Practical Guide, Chapter 7: Metabolomics Data Analysis Using MZmine (2020). DOI: [10.1039/9781788019880-00232](https://doi.org/10.1039/9781788019880-00232)
- Smoluch M., Piechura K. Mass Spectrometry: An Applied Approach, Chapter 3: Basic Definitions (2019). DOI: [10.1002/9781119377368.ch3](https://doi.org/10.1002/9781119377368.ch3)
- IUPAC. Compendium of Chemical Terminology, 2nd ed. (the "Gold Book"). Compiled by A. D. McNaught and A. Wilkinson. Blackwell Scientific Publications, Oxford (1997). Online version (2019-) created by S. J. Chalk. ISBN 0-9678550-9-8. [10.1351/goldbook](https://doi.org/10.1351/goldbook)
- Guo, J., Huan T. Evaluation of significant features discovered from different data acquisition modes in mass spectrometry-based untargeted metabolomics. *Analytica Chimica Acta* (2020). DOI: [10.1016/j.aca.2020.08.065](https://doi.org/10.1016/j.aca.2020.08.065)

Ansgar Korf, Robin Schmid, Steffen Heuckeroth, Steffen Heuckeroth, corinnabrungs, omokshyna, tdamiani

⌚ October 20, 2025 10:14:12

👤 Ansgar Korf, Robin Schmid, Steffen Heuckeroth, Steffen Heuckeroth, corinnabrungs, omokshyna, tdamiani

15.5 Ion mobility spectrometry terminology

15.5.1 Background

Ion-mobility mass-spectrometry, here simply referred to as **ion-mobility (IM)**, is an analytical technique where ions are separated through a gas-filled mobility cell prior to the MS acquisition.

In classic **drift tube ion mobility (DTIM)**, ions migrate through an inert buffer gas under the influence of a weak electric field. Ions drift with different **velocity** based on their interaction with the buffer gas, which allows for the separation of different shaped molecules. Modern devices are able to perform IM separation on a millisecond timescale, typically within 10 to 100 ms.

As larger ions have more collisions with the gas, they are more strongly retarded than their smaller counterparts. Thus, smaller ions, having a smaller cross section, arrive earlier at the detector than ions with a larger collisional cross section (**CCS**).

The **ion mobility** $\langle K \rangle$ is then defined as the ratio of the analyte's steady-state net drift velocity to the applied electric field, and it is convention to calculate the reduced ion mobility $\langle K_0 \rangle$ at standard pressure and standard temperature. This value is often reported as the inverse reduced ion mobility $\langle 1/K_0 \rangle$.

From practical point of view, IM nicely fits in-between LC separation (~seconds timescale) and MS detection of TOF instruments (~microseconds timescale). This allows LC-IM-MS instruments to acquire several MS spectra during each **accumulation**, without incurring sensitivity loss.

For example, assuming a typical 100 μs MS-acquisition time of TOF analyzers, around 1000 spectra can be recorded within 100 ms of IM separation. Therefore, as opposed to LC-MS, multiple MS (or MS₂) spectra are associated to each RT in LC-IM-MS data.

A more detailed explanation of LC-MS and LC-IMS-MS raw data structure is provided [here](#).

[Visual explanation of IMS by Waters](#)

Trapped ion mobility spectrometry (TIMS)

Trapped ion mobility spectrometry (TIMS) reverses the concept of traditional drift tube IM. Rather than moving ions through a stationary gas, TIMS holds ions stationary against a **moving gas** and then releases them according to their mobility.

TIMS has the advantage that the physical dimension of the analyzer can be smaller, whereas the analytical column of gas - the column that flows past during the course of an analysis - can be large and user defined. This can lead to increased method sensitivity.

[Bruker's TIMS TOF Video](#)

Time-dispersive ion mobility spectrometry (DTIMS and TWIMS)

Time-dispersive IM devices include "traditional" **drift tube (DTIMS)** and **travelling-wave (TWIMS)** devices.

In TWIMS, ions are propelled through a gas-filled stacked ring ion guide with the help of travelling voltage waves.

Higher mobility ions undergo less 'roll over' events on the waves than the lower mobility ions. As the waves pass along the device, ions can 'surf' on the wave front for a period of time before being overtaken by the wave. Usage of travelling waves makes possible to increase sensitivity, selectivity, and speed of the method.

For more information, see *Fundamentals of Traveling Wave Ion Mobility Spectrometry* DOI: [10.1021/ac8016295](https://doi.org/10.1021/ac8016295)

15.5.2 Terminology

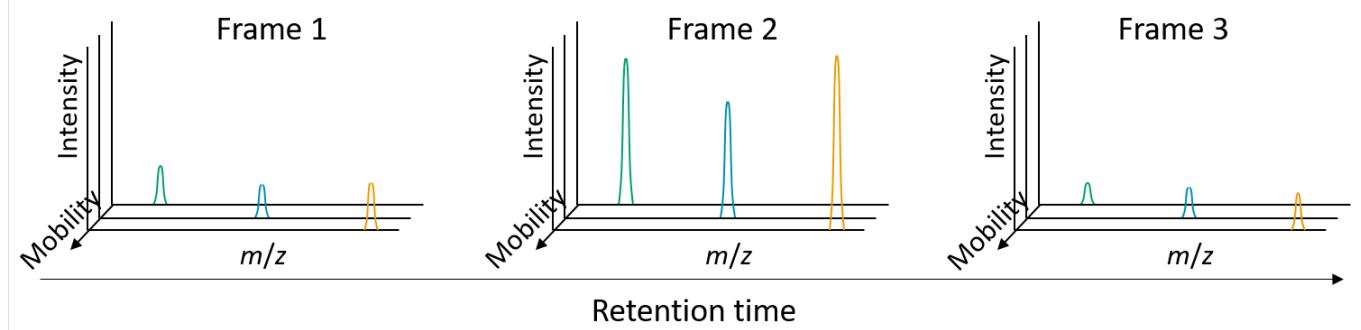
Accumulations, Mobility Scans and Frames

Altough mainly used for TIMS, the term "**accumulation**" refers to the pack of ions gathered at the head of the IM device prior to the release and separation in the IM cell.

As explained [above](#), since the accumulation-separation cycle typically last ~100 ms, multiple MS spectra (referred to as "**mobility scans**" in MZmine) are acquired during each cycle.

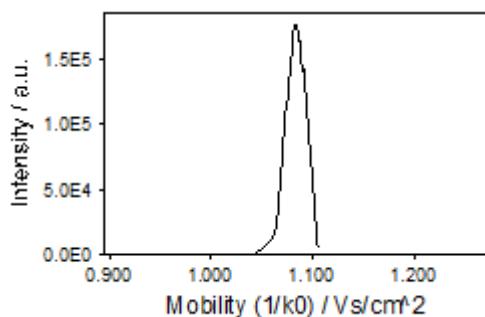
Frame is the set of **mobility scans** collected during each IM separation. A frame can be seen as the IM separation of a single accumulation, along which multiple MS spectra are collected. Several frames are contained within one LC peak. Thus, the **frame number** are a natural unit to measure chromatographic RT.

See [here](#) for more details.



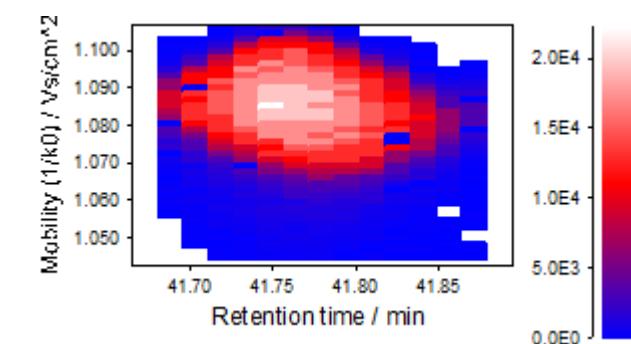
Mobilograms

A "mobilogram" represents the intensity of an m/z or m/z range along the mobility axis. A *mobilogram* may be build from multiple frames and summed or built from a single frame.



Ion mobility trace

An "ion mobility trace" basically represents a mobility resolved extracted ion chromatogram (EIC).



Collisional Cross Section

Collision cross section (CCS) can be defined as area of interaction between an individual ion and gas molecule. CCS depends on ion's size, shape, and charge. IM-derived CCS values can be used as an additional molecular descriptor to support the compound unknown identification process.

15.5.3 References

- Meier, F., Brunner, A.D., Koch, S., Cox, J., Räther, O., Mann, M. Online Parallel Accumulation-Serial Fragmentation (PASEF) with a Novel Trapped Ion Mobility Mass Spectrometer. *Molecular & Cellular Proteomics* (2018). DOI: [10.1074/mcp.TIR118.000900](https://doi.org/10.1074/mcp.TIR118.000900)
- Paglia, G. et al. Ion Mobility Derived Collision Cross Sections to Support Metabolomics Applications. *Anal. Chem.* (2014). DOI: [10.1021/ac500405x](https://doi.org/10.1021/ac500405x)

[omokshyna](#), [tdamiani](#)

 November 24, 2022 21:13:07

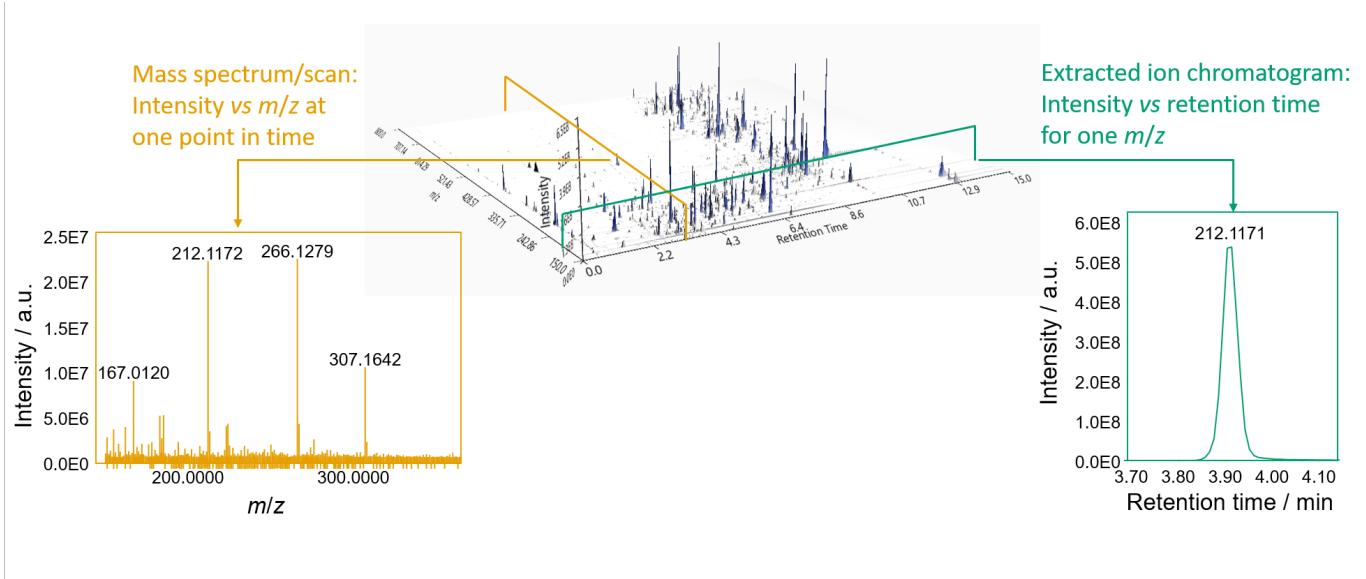
 [omokshyna](#), [tdamiani](#)

15.6 Graphical comparison of LC-MS and LC-IMS-MS data

Classic LC-MS data consists of three dimensions:

- m/z,
- intensity,
- and retention time.

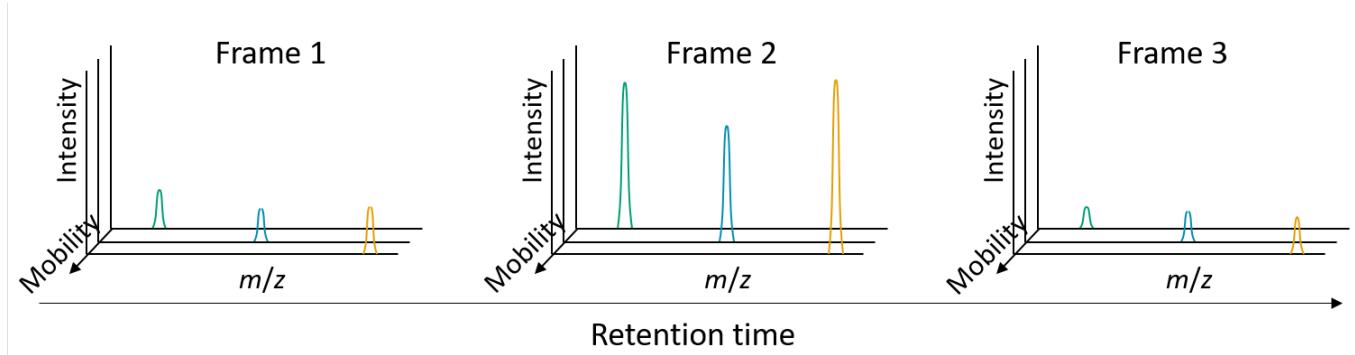
At every retention time, a whole mass spectrum is acquired (yellow). Putting all scans together creates a three-dimensional plane. By slicing the three-dimensional data at a single m/z value (+- a value of tolerance), **EICs** can be created (green).



Ion mobility resolved data, on the other hand, consists of a three-dimensional data plane at each retention time. The three dimensions being:

- m/z,
- intensity,
- and mobility (as drift time (ms) or inverse reduced mobility $\langle(1/K_0 [Vs/(cm^2)])\rangle$).

The 3D projection of regular LC-MS data can be created by summing all mobility scans of a frame to create a frame spectrum. (see [Mobility scan merging](#))



Steffen Heuckeroth, omokshyna

⌚ November 24, 2022 21:13:07

👤 Steffen Heuckeroth, omokshyna

15.7 Spectral similarity measures

15.7.1 Weighted cosine spectral similarity

The most common spectral similarity measure for library search is the **weighted cosine similarity**. Generally, the cosine similarity is calculated as following:

$$\text{similarity} = \cos\theta = (u \cdot v) / (\sqrt{\sum(u^2)} * \sqrt{\sum(v^2)})$$

In case of weighted cosine similarity, the previous formula is modified according to the weighting function. In case of weighting function in form $(m/z^a * \text{Intensity}^b)$ (where a, b are weights) the weighted cosine similarity is calculated as follows:

$$\begin{aligned} \text{weighted similarity} &= (u' \cdot v') / (\sqrt{\sum(u'^2)} * \sqrt{\sum(v'^2)}) \\ u' &= \sum(u_i) = \sum(m_i \cdot I_i^a \cdot I_i^b) \\ v' &= \sum(v_i) = \sum(m_i \cdot V_i^a \cdot V_i^b) \end{aligned}$$

where u and v are the aligned vectors of the two spectra.

It is used to determine the similarity between two spectra (usually library and query spectra). Both spectra are turned into vectors and cosine similarity is calculated by division of vectors dot product over cosine value of the angle between them.

15.7.2 Composite weighted cosine spectral similarity (identity)

This similarity measure can be especially useful for very reproducible generation of spectra (GC-EI-MS). This measure is modified by a ratio based on the relative intensities of adjacent m/z signals in the two spectra.

Composite weighted cosine similarity is calculated as follows:

$$\begin{aligned} \text{composite similarity} &= \frac{N * \text{cosine similarity} + overlap * \text{ratio factor}}{N + overlap} \\ \text{ratio factor} &= \frac{\min(r_{lib}, r_{query})}{\max(r_{lib}, r_{query})} \\ r_{lib} &= I_{i-1}/I_i \quad (lib) \\ r_{query} &= I_{i-1}/I_i \quad (query) \end{aligned}$$

where (N) - number of signals in a query spectrum, cosine similarity is calculated as described [previously](#), overlap - number of matching signals in query and library spectra, $(ratio\ factor)$ - relative intensities ratio, (r_{lib}) - relative ratio of adjacent signals in a library spectrum, (r_{query}) - relative ratio of adjacent signals in a query spectrum.

It is used to determine the similarity between two spectra (usually library and query spectra).

15.7.3 Parameters

WEIGHTS

For calculating the cosine similarity, different weighting strategies for m/z and signal intensities can be applied.

Several weighting schemes are available:

- None $(m/z^0 * I^1)$ (weighting only by intensities)
- SQRT $(m/z^0 * I^{0.5})$ (weighting only by intensities)
- MassBank $(m/z^2 * I^{0.5})$
- NIST11 (LC) $(m/z^{1.3} * I^{0.53})$
- NIST (GC) $(m/z^3 * I^{0.6})$

 Choice of the similarity measure depends on your data, and intensity-based schemes might work better on homogenous datasets. However, recommended approach for choosing the weighting scheme would be trial-and-error.

MINIMUM COSINE SIMILARITY

This option defines the minimum accepted similarity score that is taken into account for annotation. The similarity score depends on the data handling of unmatched signals.

HANDLE UNMATCHED SIGNALS

Signals that only occur within one scan (query OR library entry) can be handled in different ways to affect the cosine similarity and controlling the quality of matching results.

- KEEP ALL AND MATCH TO ZERO (default)

This is the standard conservative approach where all unmatched signals weigh negatively on the score. It is used for both GC-EI-MS and MS² spectra.

- REMOVE ALL

The opposite option that discards all unaligned signals, which increases the similarity score artificially.

 This option is only feasible if both the library and query spectrum are considered to be complex mixtures. Therefore, the next two options are more conservative.

- KEEP LIBRARY SIGNALS

Results in discarding all unaligned signals of the query scan, whereas all unaligned library signals are matched to zero, setting the library spectrum as the ground truth.

 Here, the negative impact of contaminating signals in the query scans are reduced. This might be helpful for mixed spectra of multiple compounds, especially during imaging techniques without any further separation or all ion fragmentation/data independent fragmentation workflows.

- KEEP EXPERIMENTAL SIGNALS

Results in discarding all unaligned signals of the library scan, whereas all unaligned query signals are matched to zero.

Has reversed effects compared to the previous option.

[Robin Schmid, omokshyna](#)

 January 11, 2023 16:29:45

 Robin Schmid, omokshyna

15.8 Other Detector Workflow Glossary

15.8.1 Trace

A **trace** or **chromatogram** refers to paired 2D (x-y) data acquired from another detector, e.g. the intensity/absorption vs. retention time from an UV or DAD detector.

15.8.2 Trace type

There are three types of traces in mzmine's **Other detector** workflow. **Raw traces**, **Preprocessed traces**, and **Features**.

Raw trace

The raw trace as acquired from the detector, without any alterations such as binning, smoothing, baseline correction, or shifting.

Preprocessed trace

A preprocessed state of the data from a non-MS detector. In this state, a varying number of preprocessing steps has been applied (binning, smoothing, baseline correction, or shifting...). There is only a single preprocessed trace per raw trace at a time. Most *other detector processing modules* work on the preprocessed trace by default. In case there is no preprocessed trace, mzmine will fall back to the raw trace.

Features

A feature is a specific peak in a trace/chromatogram.

15.8.3 Trace selection parameter

This parameter is used throughout the workflow for processing data from other detectors to select the traces that are being processed.

Chromatogram type:	ABSORPTION
Range unit filter:	range unit filter
Trace selection	range label filter
Description filter:	description filter
Trace type:	pre-processed



Typically, it is not necessary to select all existing traces here, but only the ones you want to correlate with the MS, e.g., the UV (=Absorption) traces.

Chromatogram type Specify the type of chromatogram you want to process. Usually, specifying this is enough.

Range unit filter (Uncommon) filter the traces of the selected chromatogram type to only have a specific unit on the range axis to be affected by this filter. Use `*` as a wildcard (match any character)

Range label filter (Uncommon) filter the traces of the selected chromatogram type to only have a specific label on the range axis to be affected by this filter. Use `*` as a wildcard (match any character)

Description filter (Uncommon) filter the traces of the selected chromatogram type to only have a specific description (=name) to be affected by this filter. Use * as a wildcard (match any character)

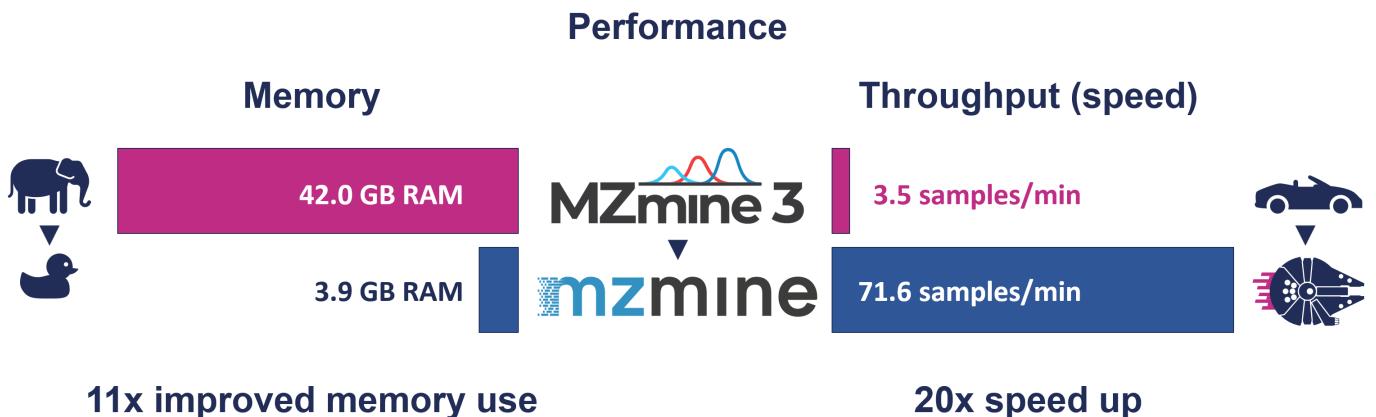
Trace type Specify the [trace type](#) to be used as input for this module.

 July 28, 2025 13:30:35

 Steffen Heuckeroth

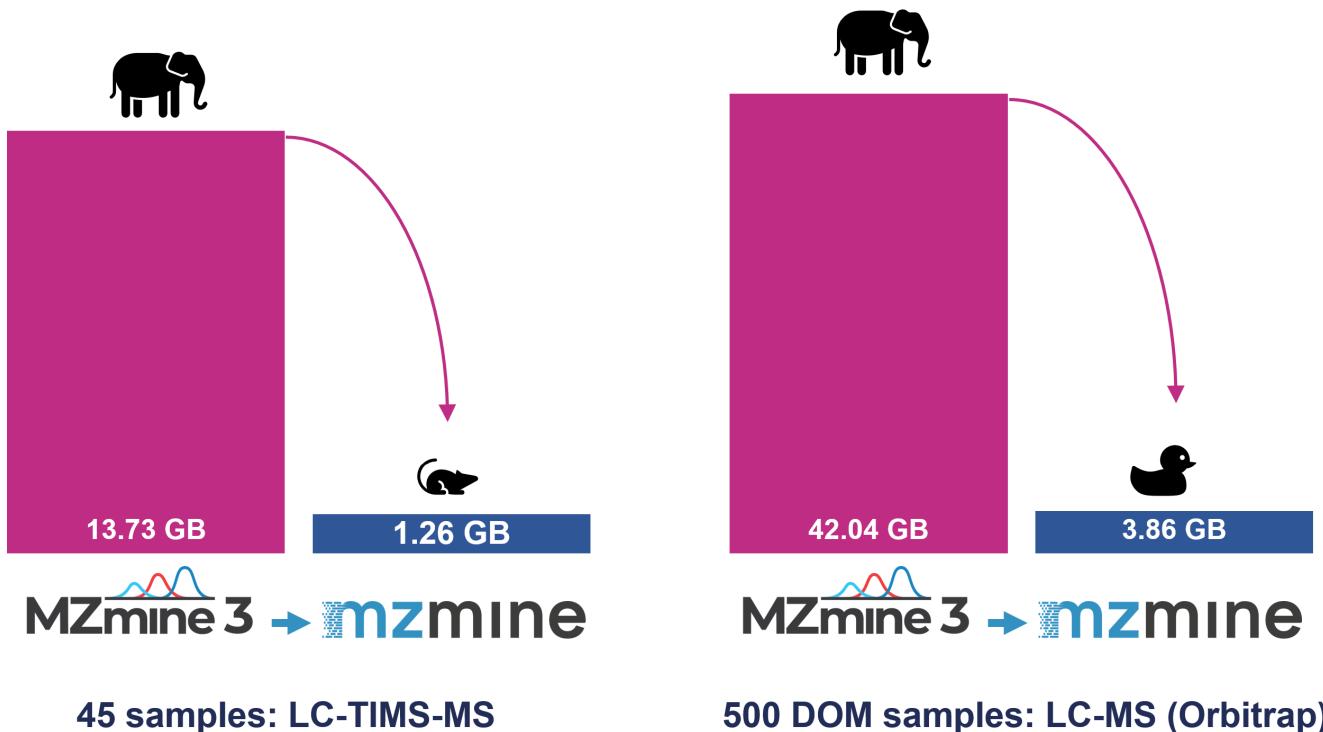
15.9 Additional figures

15.9.1 Performance

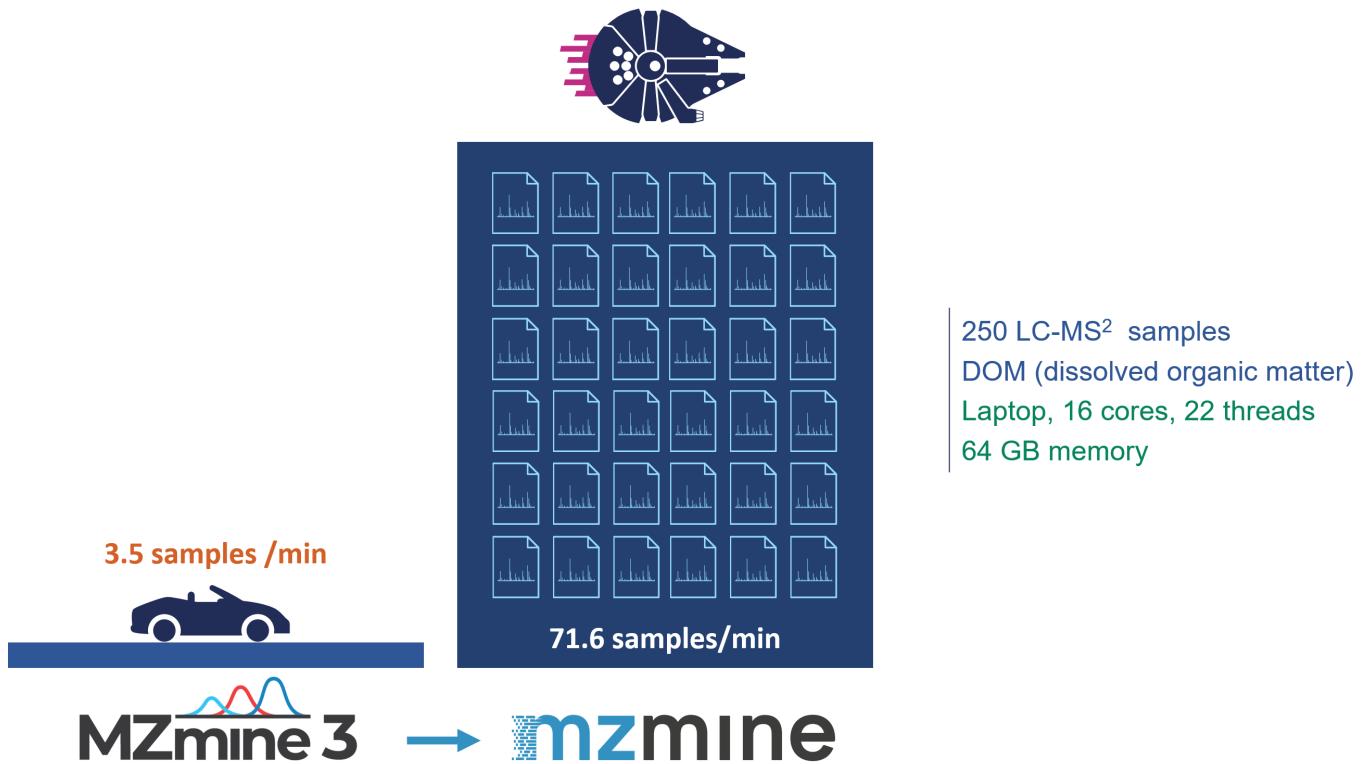


500 LC-MS² samples DOM (dissolved organic matter)
Intel Core Ultra 9 185H, 16 cores, 22 threads, 64 GB RAM, laptop

Scaling MS Data Processing – RAM Memory



Scaling MS Data Processing & Molecular Networking



250 LC-MS² samples
DOM (dissolved organic matter)
Laptop, 16 cores, 22 threads
64 GB memory

15.9.2 Feature table

Aligned feature list 13C peak gaps dup (1) corr PEARSON r greeq 0.85 dp greeq 5																
ID	+ m/z	> RT	Area	Shapes		Height box plot	Ion identity		Spectral match				Height	Height	Height	Height
				condition	Ion identity		Spectral match	Similarity ▾	Adduct	Formula	Structure					
976	182.0811	0.631	7.9E5		L-tyrosine (0.840)		0.840	[M+H] ⁺	C9H11NO3	<chem>O=C(Oc1ccc(O)c(Oc2ccccc2N))N</chem>		7.6E6	6.5E6	6.4E6	5.2E6	
1406	281.1493	0.768	2.6E7		[M+H] ⁺		0.836	[M+H] ⁺	C14H20N2O4	<chem>O=C(Oc1ccc(O)c(Oc2ccccc2N))N</chem>		5.2E6	5.2E6	4.3E6	2.6E6	
3646	342.2386	1.566	3.1E7		[M+H] ⁺		0.832	[M+H] ⁺	C17H31N3O4	<chem>O=C(Oc1ccc(O)c(Oc2ccccc2N))N3CCN(C)CC3</chem>		7.3E7	6.9E7	7.2E7	4.9E7	
5181	224.0376	2.532	1.2E6		Dihydroeruginic acid (0.822)		0.822	[M+H] ⁺	C10H9NO3S	<chem>O=C(Oc1ccc(O)c(Oc2ccccc2N))N3CCN(C)CC3</chem>		2.6E7	2.2E7	2.1E7	1.7E7	
783	200.1280	0.573	6.7E5		1103336-91-9 (0.805)		0.805	[M+H] ⁺	C10H17NO3	<chem>O=C(Oc1ccc(O)c(Oc2ccccc2N))N3CCN(C)CC3</chem>		2.4E7	3.0E7	2.1E7	1.8E7	
4568	342.2386	1.948	2.5E6		[M+H] ⁺		0.803	[M+H] ⁺	C17H31N3O4	<chem>O=C(Oc1ccc(O)c(Oc2ccccc2N))N3CCN(C)CC3</chem>		1.9E7	1.5E7	1.9E7	9.0E6	
3777	245.1284	1.621	5.3E6		14705-60-3 (0.801)		0.801	[M+H] ⁺	C14H16N2O2	<chem>O=C(Oc1ccc(O)c(Oc2ccccc2N))N3CCN(C)CC3</chem>		1.4E8	1.1E8	1.1E8	1.0E8	
1170	217.1546	0.650	3.3E7		[M+H] ⁺		0.800	[M+H] ⁺	C10H20N2O3	<chem>O=C(Oc1ccc(O)c(Oc2ccccc2N))N3CCN(C)CC3</chem>		1.6E7	1.2E7	1.0E7	1.2E7	

 November 25, 2025 09:21:47

 Robin Schmid

16. Contribute

16.1 Coding

16.1.1 mzmine architecture & development process

Technology

mzmine is developed in modern Java, and the distributed installers and portable versions ship with a matching Java Runtime. This makes mzmine standalone platform independent. There is no need to install Java on the target machine. Updates can be installed over older releases and follow semantic versioning, i.e., major.minor.patch like 4.8.0. In rare cases, the mzmine installer may ask for manual removal of a previous installation. Run mzmine through its graphical user interface (GUI), built with JavaFX, or via [command line interface \(CLI\)](#) to run batch processing.

Contributions to the mzmine [source code](#) and [documentation](#) are possible through pull requests (code) or issues to the respective repositories. Feel free to create a fork of each repository to work on feature branches. We recommend that you reach out to our team early in your idea or development phase to boost your productivity. We can provide help to new developers to get started with mzmine development and provide pointers to already existing resources.

- How to contribute to the [documentation](#)?
- How to contribute to the [mzmine code base](#)?



The practical instructions for mzmine coding can be found [here](#).

GitHub issues as a pool of ideas and feedback:

- Providing **feedback** on specific modules or functions
- Outlining **feature requests**: This can be small improvements or full module ideas. Useful information includes: Links to articles, reference implementations, and descriptions of user inputs, expected results, and example use cases.
- Reporting **bugs**: Provide all details on how to reproduce bugs, the log file, batch configuration, example data

One of the main design goals is to keep a strict separation between the application core and modules for data processing and visualization. The general architecture of the modules is presented at the following scheme.

Roadmap

The development process of mzmine is agile and driven by multiple factors:

- **User Feature Requests**: Community needs and feedback shape our development priorities
- **Strategic Partnerships**: Collaborations with academic and industry partners guide development directions
- **Contracted Development**: Specific features can be developed through funded projects
- **Community Contributors**: Open-source contributions from developers worldwide
- **Internal Planning**: Strategic development guided by the mzio team's expertise and vision

Our roadmap remains flexible to accommodate emerging needs while maintaining focus on core functionality improvements.

16.1.2 Development

Core architecture

mzmine's core architecture is built on several key components that enable efficient and modular data processing:

TASK CONTROLLER

The TaskController manages the execution of processing tasks, allowing multiple operations to run in parallel. It handles task scheduling, priority management, and resource allocation to optimize performance on multi-core systems. Tasks can be monitored through progress tracking and canceled if needed.

MODULE SYSTEM

mzmine uses a modular architecture where each processing step is implemented as an independent module. This design:

- Ensures loose coupling between different processing steps
- Makes modules reusable across different workflows
- Simplifies maintenance and testing
- Allows easy addition of new functionality without modifying existing code

GUI FRAMEWORK

The graphical user interface is built with JavaFX following the Model-View-Controller (MVC) pattern:

- Interactive data visualization dashboards
- Real-time updates of processing results
- Flexible window management for multiple views
- Custom controls for mass spectrometry data visualization
- Interactive linking between dashboards through reactive programming (binding of JavaFx properties)

PARAMETER SYSTEM

User parameters are easy to design and extend in their functionality and offer:

- Consistent, auto generated user interfaces across all modules
- Type-safe parameter handling
- Automatic validation of user inputs
- Persistent storage of parameter values
- Versioning of parameters, offering handling of simple changes in parameter behavior or to notify about impactful changes

Module development

Each module in mzmine is built from three classes:

- [Name]Module.java
- [Name]Parameters.java
- [Name]Task.java



You can find examples for all these classes in the `java/io/github/mzmine/modules/example` folder

MODULE CLASS

The module holds metadata like the name, description, and ParameterSet class.

Each Module class implements one of three interfaces or its subclasses:

- **MZMineModule** interface - the most basic interface, aimed at any module that has to store some kind of parameters,
- **MZMineRunnableModule** interface - an interface that extends mzmine module and presents a `runModule()` method, which provides an opportunity to run task (for example, from the dialog),
- **MZMineProcessingModule** interface - an interface that extends MZMineRunnableModule and gives an opportunity to execute a module in a batch.



There are special implementations of Module to make the definition and scheduling easier, like [TaskPerFeatureListModule](#) that splits the work in one task per feature list, and [SingleTaskFeatureListsModule](#) that processes all feature lists in a single task.

PARAMETERSET CLASS

The Parameters class is defined by the **ParameterSet interface**, which declares functions necessary to get, load, and save module parameters. This class provides versioning with mapping functions to load parameters that were saved by a previous version of mzmine. This includes simple mapping of parameter values, e.g., if a parameter changed its data type, or more complex handling of version changes to ensure compatibility with newer versions. The parameters are displayed in a **ParameterSetupDialog** and interfaces are auto-generated to allow user optimization. Custom * *ParameterSetupDialog**s may provide charts and other visual feedback to facilitate parameter tuning. The ParameterSet class also points to this documentation by providing a URL. If you develop a module, please provide a documentation and link as well. information must

ParameterSets are made up of **Parameters** that are accessed through static instances of these parameters. Each **Parameter** defines the GUI controls, saving and loading, and value validation.



Typically, new Parameters classes can extend the [SimpleParameterSet class](#).

Latest module parameters

MZmineCore contains a map of the latest module parameters. This ParameterSet is often copied as the starting point for creating the parameter setup dialog. Once the user runs a module with different parameters, those latest parameters are updated.

TASK

This class implements the logic of a module. The main specifications are defined by the **Task interface** and its abstract child classes. It defines methods that control and monitor the flow of a task execution (such as `cancel()`, `getTaskPriority()`, `getFinishedPercentage()`, `getStatus()`).

A new task is usually created by its module, added to the TaskController with a specified TaskPriority (HIGH, NORMAL), and the task controller will call its **run** method. If successful, the task should add an applied method to the processed RawDataFiles and FeatureLists. These applied methods hold a trace of all operations applied to data files and feature lists and will enable later generation of batch configurations from the processing history.

The best starting points are newer abstract classes like **AbstractSimpleTask**, **AbstractFeatureListTask**, or **AbstractRawDataFileTask** that already define much of the control flow logic and facilitate the creation of applied methods.

Robin Schmid, omokshyna

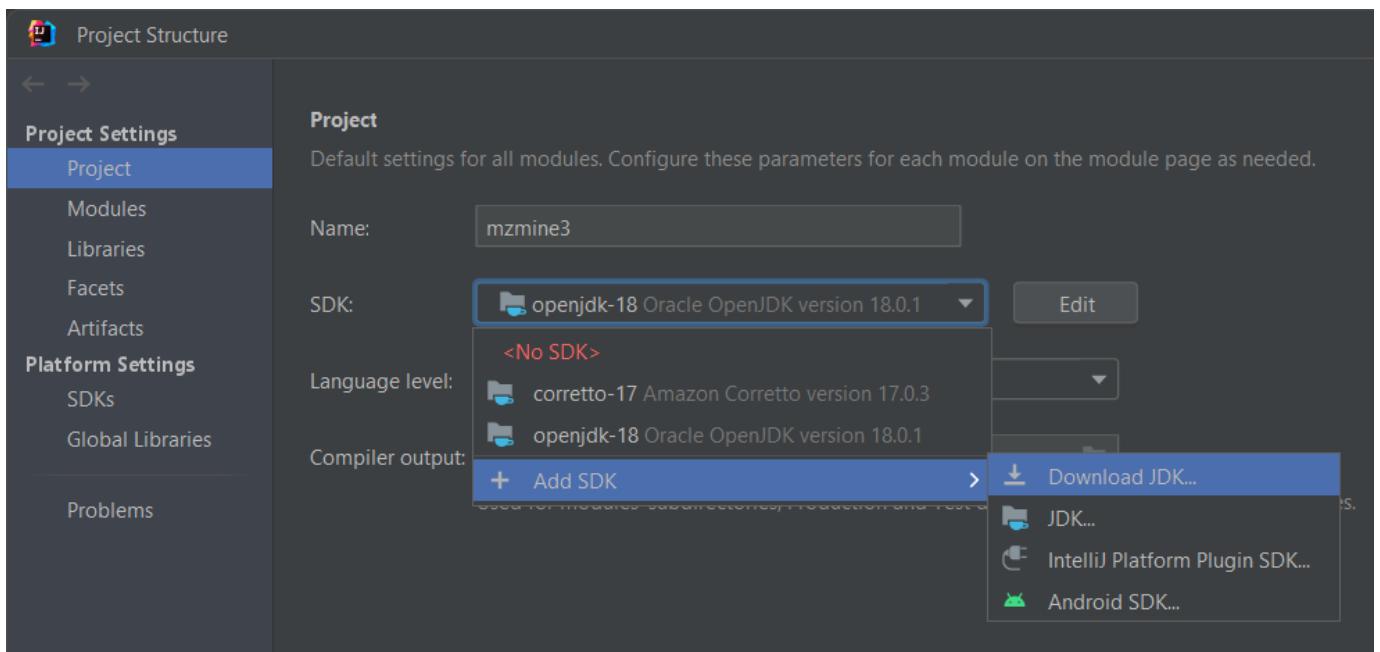
⌚ November 25, 2025 10:29:24

👤 Robin Schmid, omokshyna

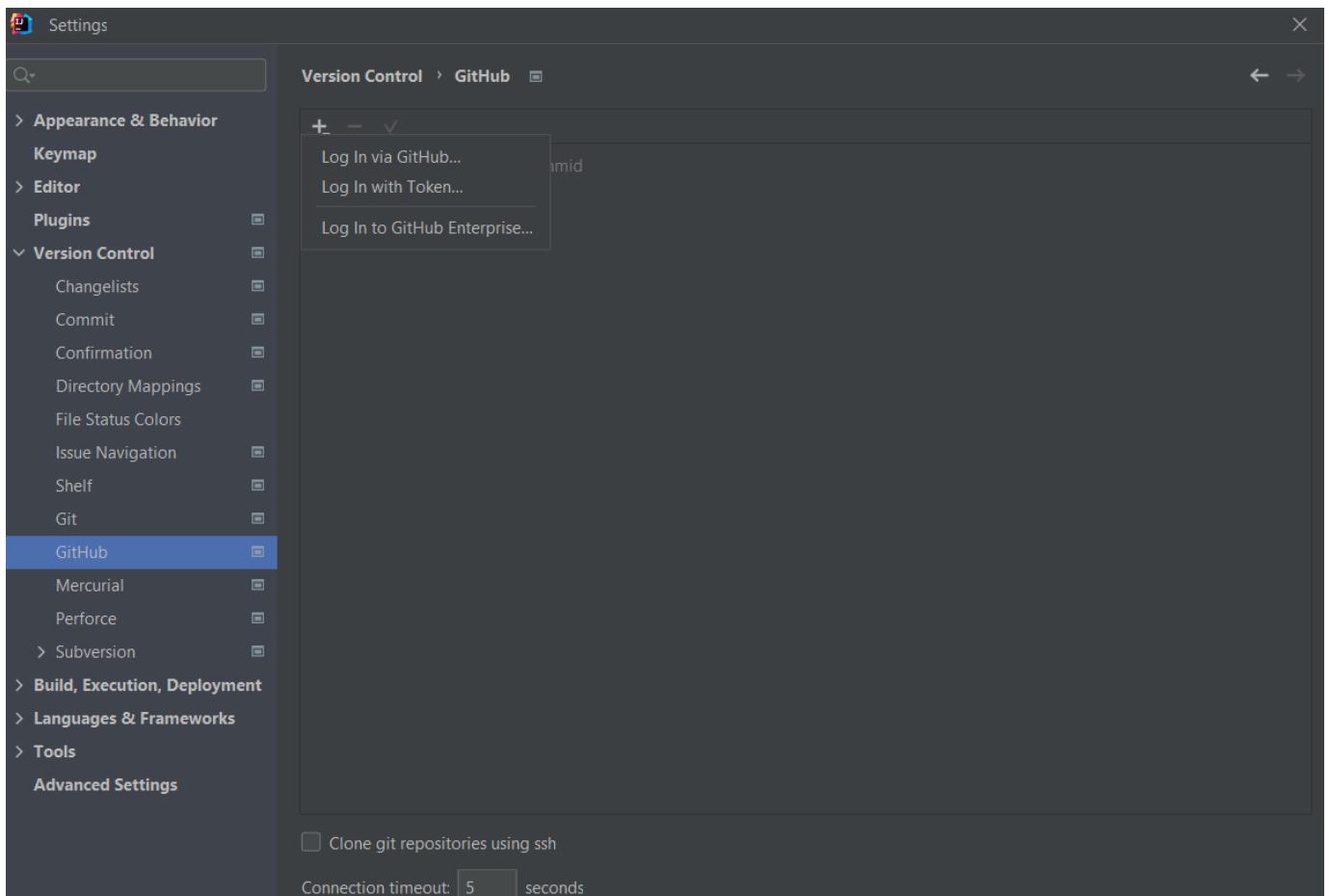
16.1.3 Development in IntelliJ

Set up

1. Fork the mzmine (<https://github.com/mzmine/mzmine3>) GitHub repository (needs free GitHub account) (See <https://help.github.com/en/github/getting-started-with-github/fork-a-repo>)
2. Download & Install IntelliJ IDEA (there is a free educational license for students and teachers) from <https://www.jetbrains.com/idea/download/>
3. Download & Install the current JDK. We recommend the OpenJDK. However, you can also use any other distribution, e.g., the Oracle JDK. This can be done from within IntelliJ. Open *File/Project Structure* (CTRL+ALT+SHIFT+S) and select SDKs and add the latest JDK with the +button:



1. Add your GitHub account via **Settings/Version Control/GitHub** +button. Below exemplified with the Log in with Token... option:
 - Log in with Token... **Generate** - redirects to GitHub
 - Make sure to select the **Workflow** scope to avoid conflicts that arise from changing GitHub actions



The screenshot shows the IntelliJ IDEA settings interface. The left sidebar is open, showing various configuration sections like Appearance & Behavior, Editor, Plugins, Version Control, Build, Execution, Deployment, Languages & Frameworks, Tools, and Advanced Settings. The 'Version Control' section is expanded, and 'GitHub' is selected, highlighted with a blue background. On the right, there's a 'GitHub' configuration panel with options for logging in via GitHub, token, or GitHub Enterprise, and a checkbox for cloning repositories using SSH.

Settings / Developer settings

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

IntelliJ IDEA GitHub integration plugin

What's this token for?

Expiration *

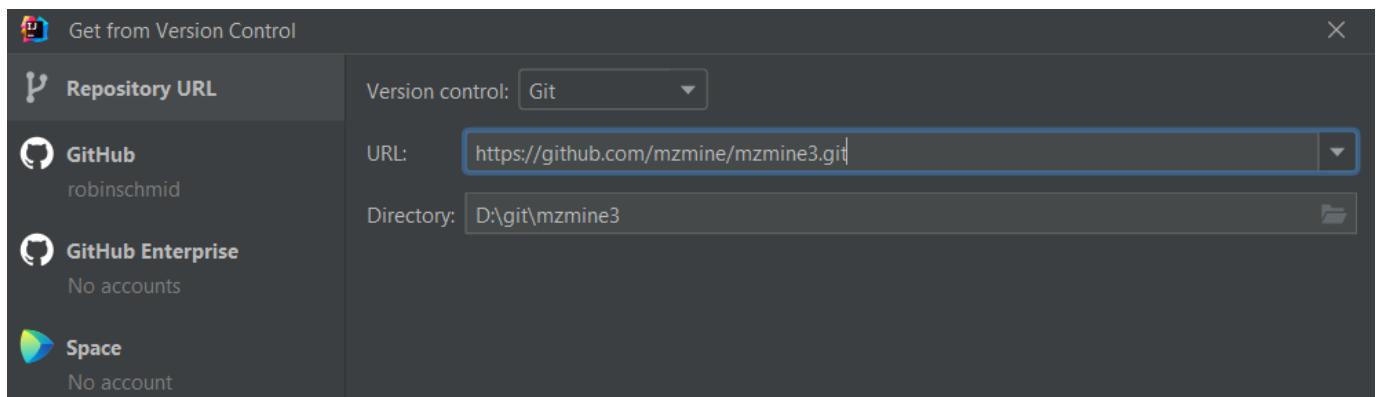
30 days The token will expire on Fri, Jun 10 2022

Select scopes

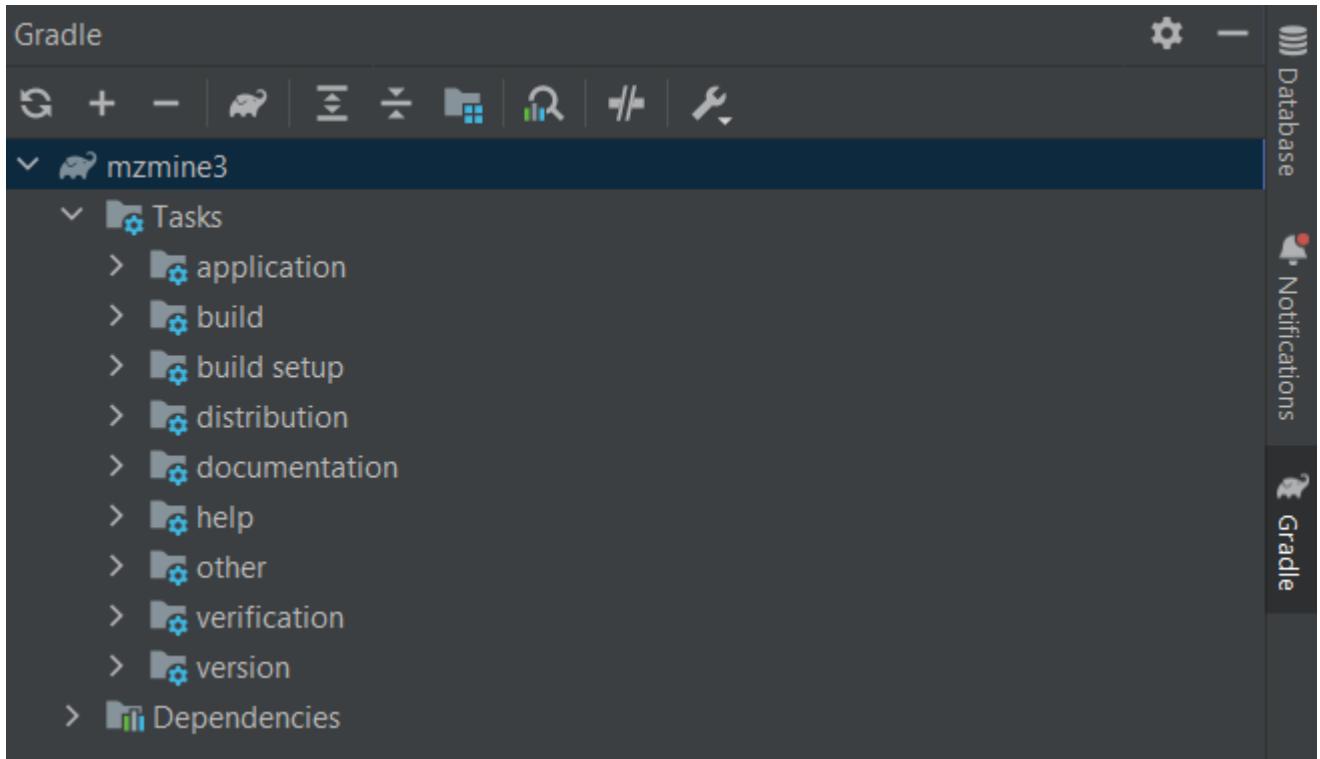
Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repostatus	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry

1. Clone GitHub project via version control: **File/New/Project from version control** use your user name to get your fork: `https://github.com/YOUR_USERNAME/mzmine3.git`



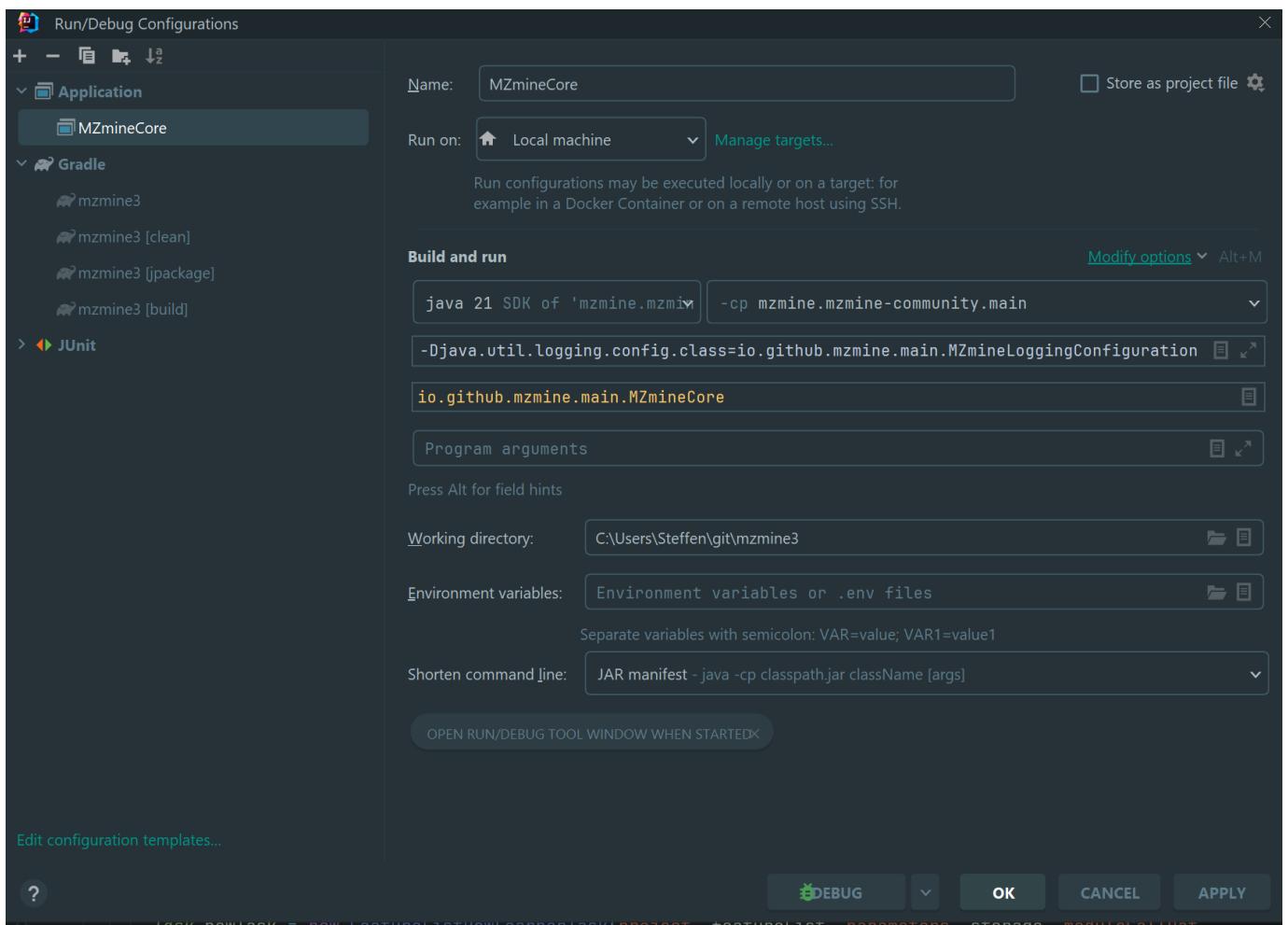
1. Make sure that gradle runs and updates the project. Otherwise, click on **Reload Gradle Project**. Now the project structure (CTRL+SHIFT+ALT+S) should show the source, test, and resource folders which are described in the build.gradle.



1. Click on Add a Configuration. Select “Application” from the template list). via the + button (don’t just edit the template):

- Main class: `io.github.mzmine.main.MZmineCore`
- VM options (add via "Modify options"): `-Djava.util.logging.config.class=io.github.mzmine.main.MZmineLoggingConfiguration --add-opens java.logging/java.util.logging=ALL-UNNAMED -ea`

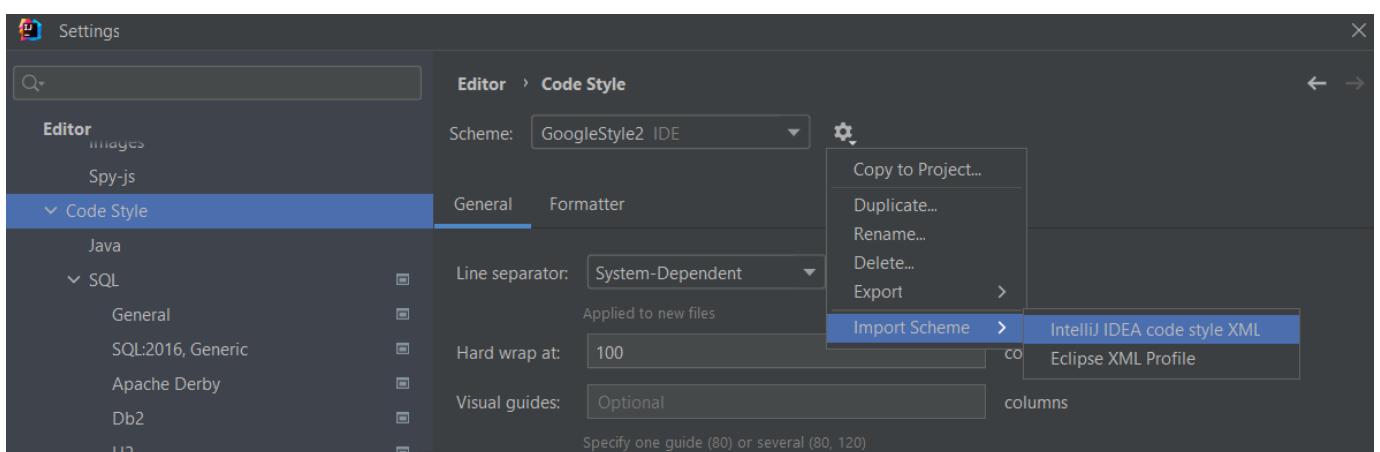
2. Run or debug with this configuration

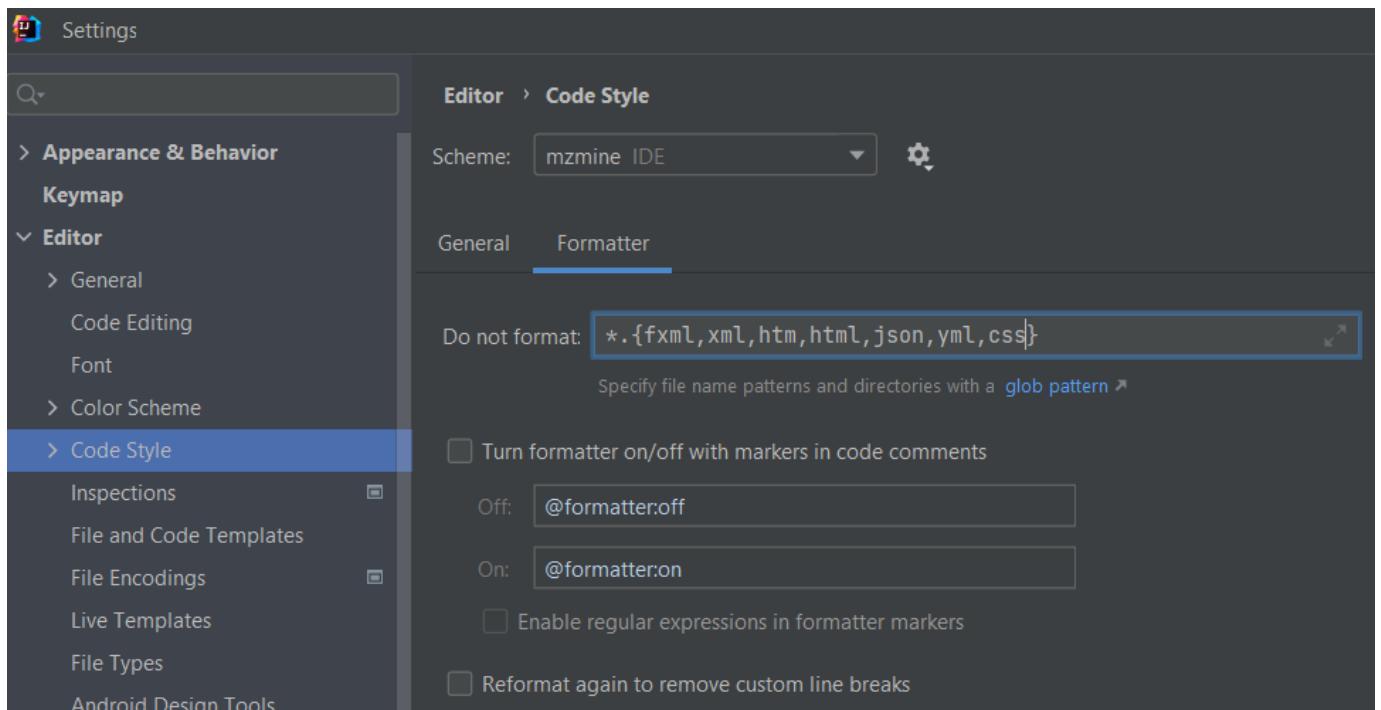


Code formatter

The code formatter is important for contributions to the main version of MZmine and each java file should contain the license header.

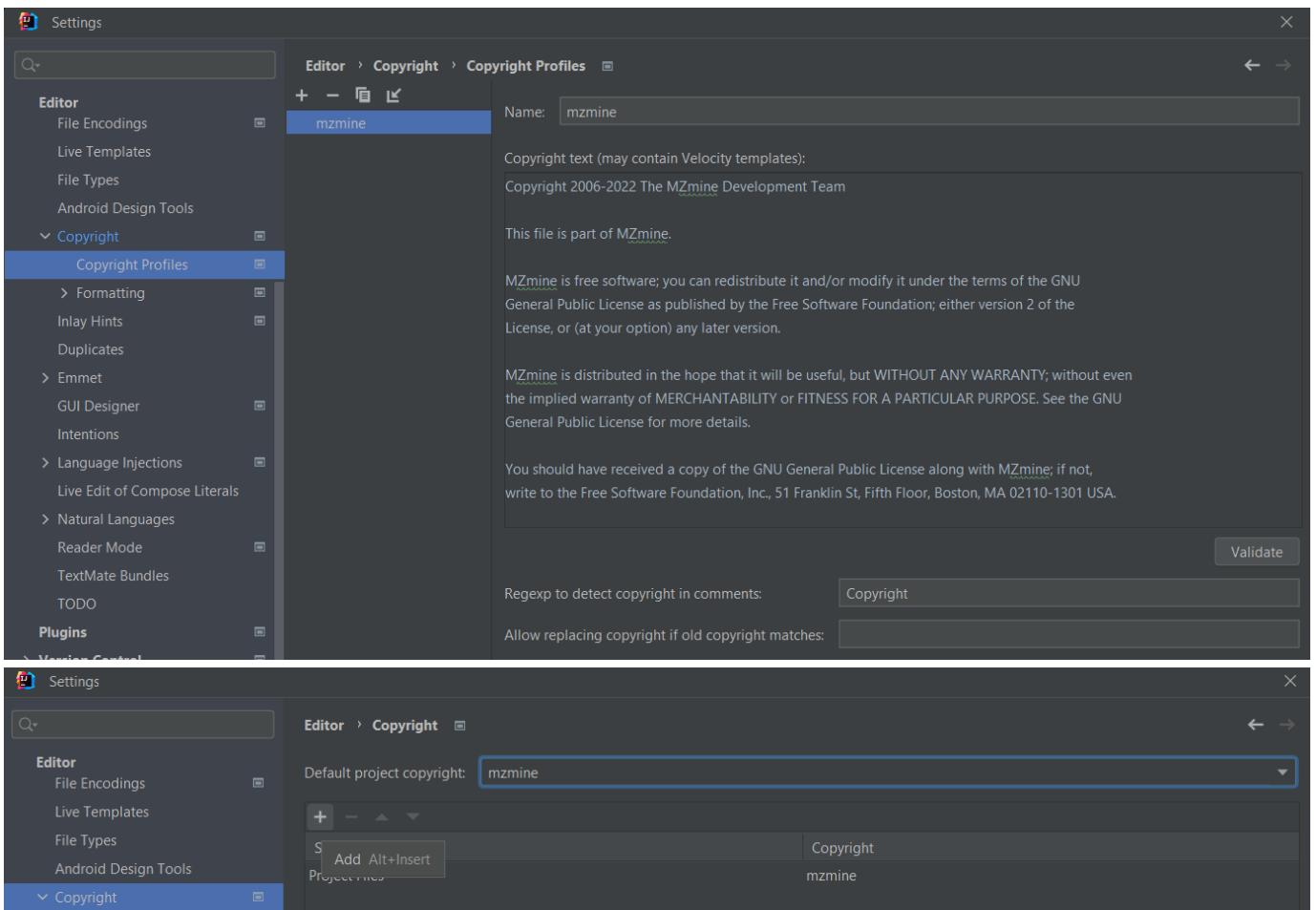
1. Import the style format from `mzmine-intellij-code-formater.xml`
2. Apply format to any file with **CTRL+ALT+L**
3. Can also be applied on each file saved or each committed change (see below)
4. Better exclude specific file formats from being formatted (see below screenshot 2) (`*.{fxml,xml,htm,html,json,yml,css}`)





Copyright header

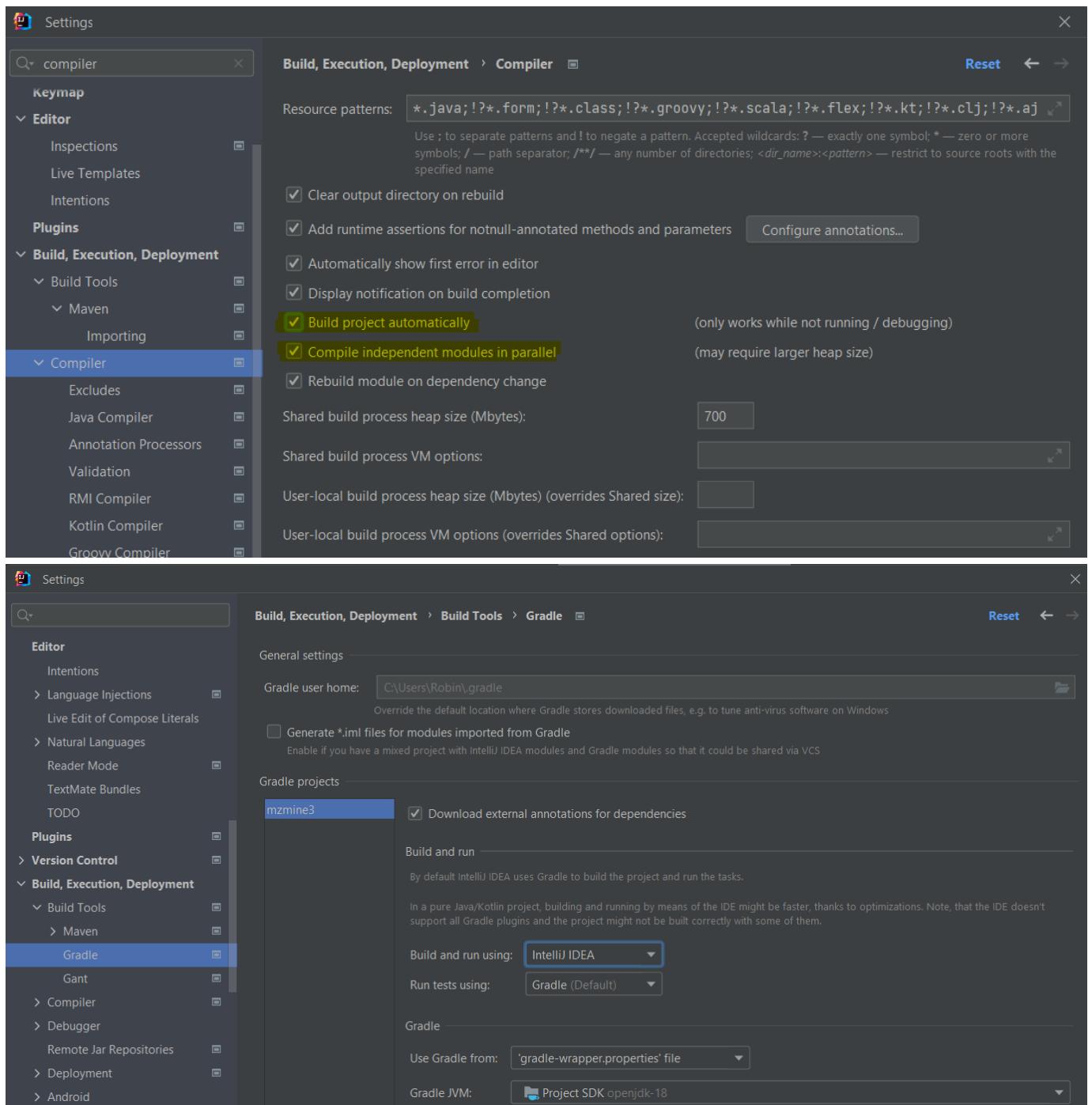
1. Add the license header to each file - the easiest way is to add the copyright profile to intelliJ
2. Import the copyright from `mzmine_intellij_licence_header.xml` or create a new one with the exact text specified in `license_header.txt`
3. Add a new scope for all project files
4. Apply after file save or commit operation or run the **Update copyright...** action



Useful settings

FASTER BUILDING

1. Activate auto building (consumes more resources)
2. Set **Settings/Gradle/build and run** to IntelliJ
3. Activate HotSwap to automatically load changed classes during debugging



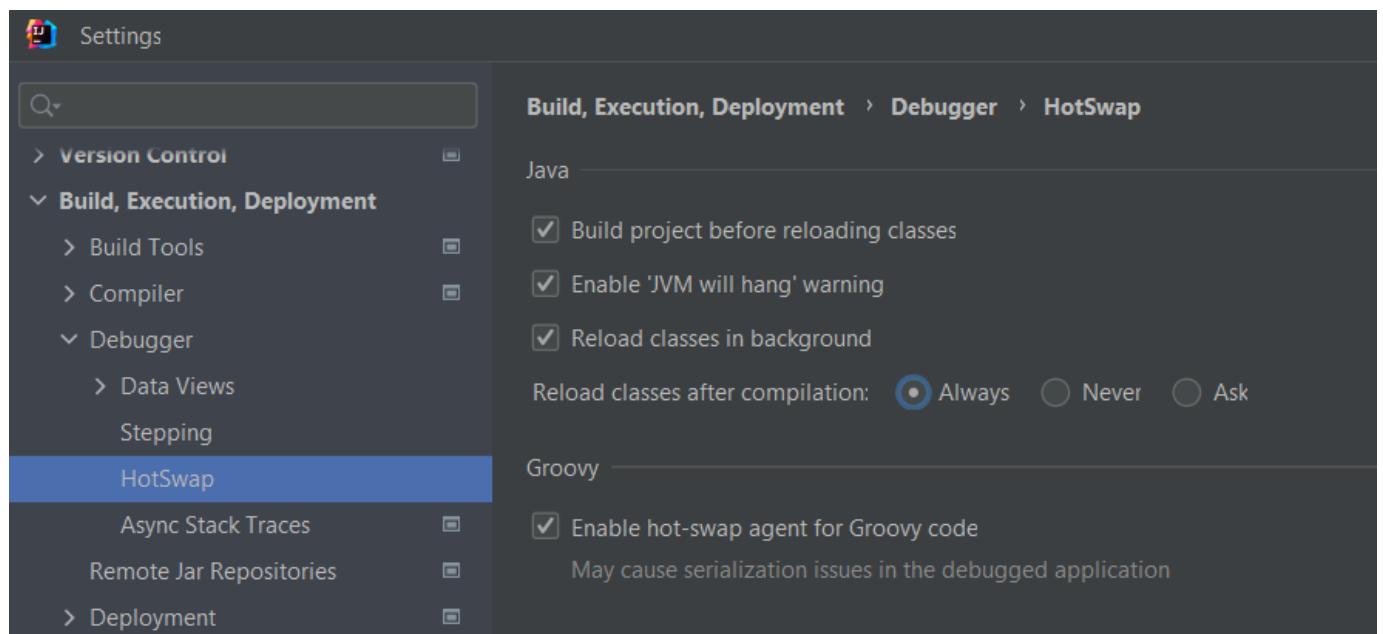
The screenshot shows two panels of the IntelliJ IDEA settings interface.

Compiler Settings (Top Panel):

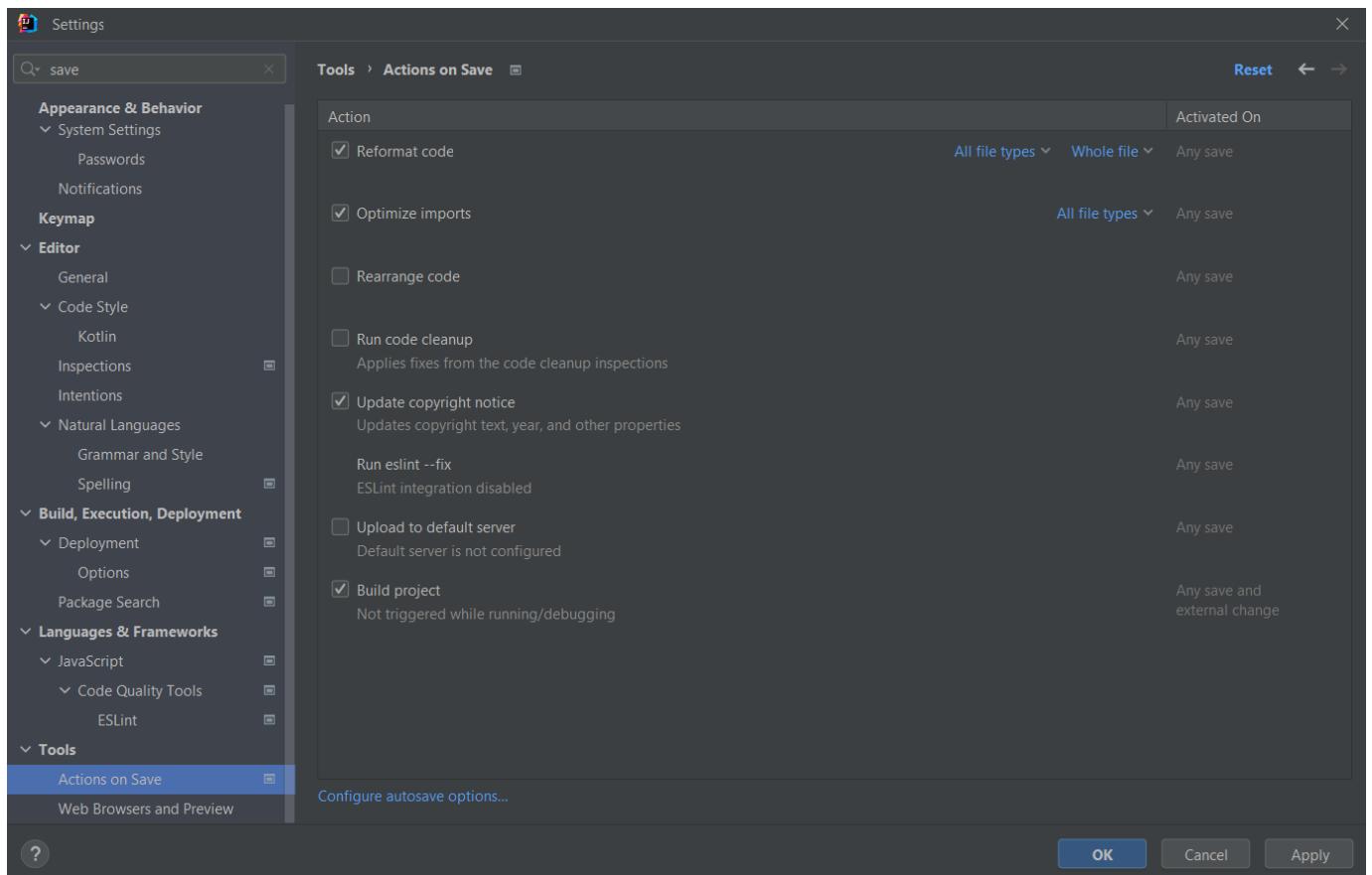
- Resource patterns:** *.java;!?*.form;!?*.class;!?*.groovy;!?*.scala;!?*.flex;!?*.kt;!?*.clj;!?*.aj
- Clear output directory on rebuild:**
- Add runtime assertions for notnull-annotated methods and parameters:** [Configure annotations...](#)
- Automatically show first error in editor:**
- Display notification on build completion:**
- Build project automatically:** (only works while not running / debugging)
- Compile independent modules in parallel:** (may require larger heap size)
- Rebuild module on dependency change:**
- Shared build process heap size (Mbytes):** 700
- Shared build process VM options:** [empty field]
- User-local build process heap size (Mbytes) (overrides Shared size):** [empty field]
- User-local build process VM options (overrides Shared options):** [empty field]

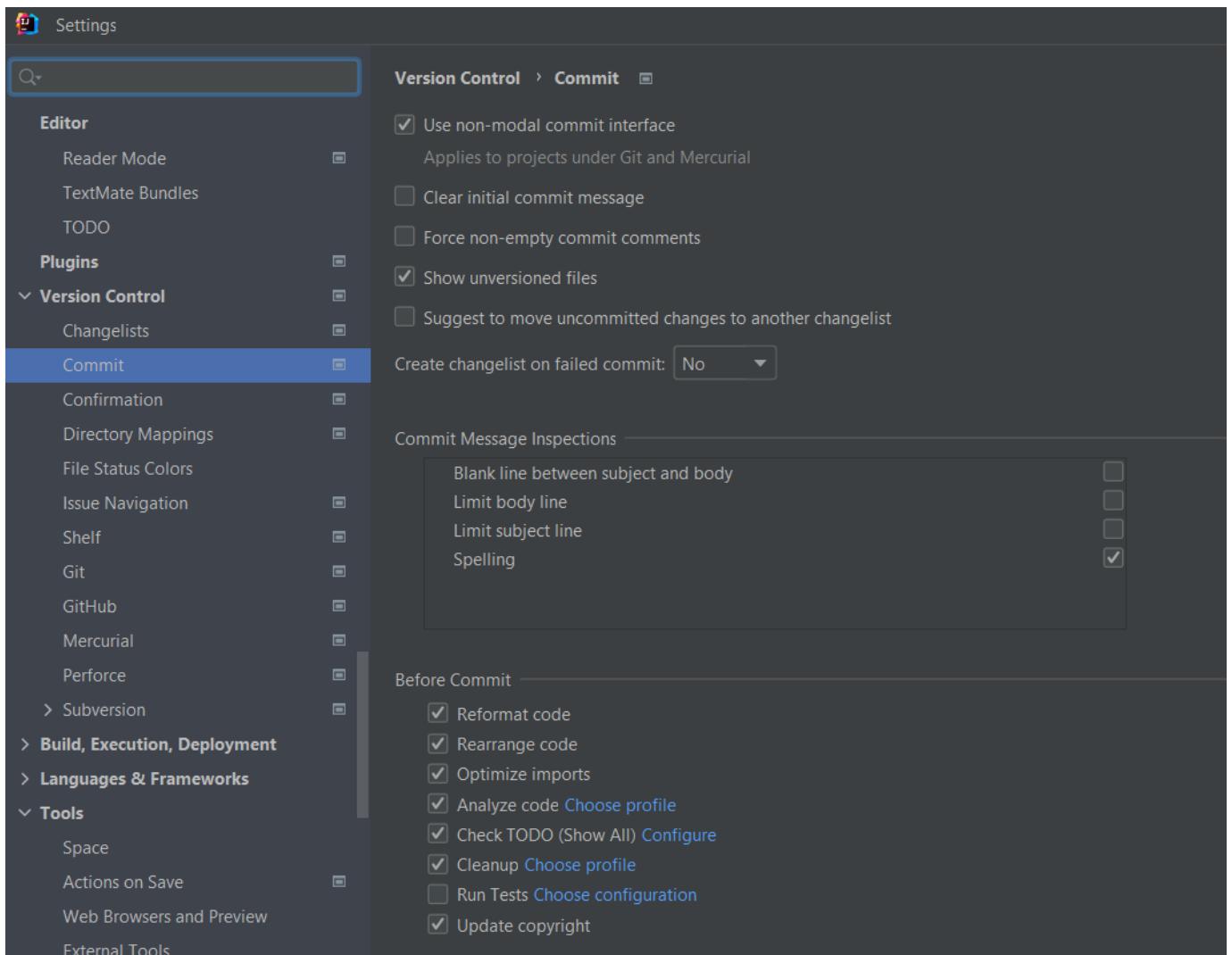
Build Tools > Gradle Settings (Bottom Panel):

- General settings:**
 - Gradle user home:** C:\Users\Robin\gradle
 - Override the default location where Gradle stores downloaded files, e.g. to tune anti-virus software on Windows:**
 - Generate *.iml files for modules imported from Gradle:** Enable if you have a mixed project with IntelliJ IDEA modules and Gradle modules so that it could be shared via VCS
- Gradle projects:** mzmine3
 - Download external annotations for dependencies:**
- Build and run:**
 - In a pure Java/Kotlin project, building and running by means of the IDE might be faster, thanks to optimizations. Note, that the IDE doesn't support all Gradle plugins and the project might not be built correctly with some of them.**
 - Build and run using:** IntelliJ IDEA
 - Run tests using:** Gradle (Default)
- Gradle:**
 - Use Gradle from:** 'gradle-wrapper.properties' file
 - Gradle JVM:** Project SDK openjdk-18

**SAVE AND COMMIT ACTIONS: APPLY COPYRIGHT, FORMAT, ETC**

Select any actions to perform when a file is saved or committed.





LIVE TEMPLATES

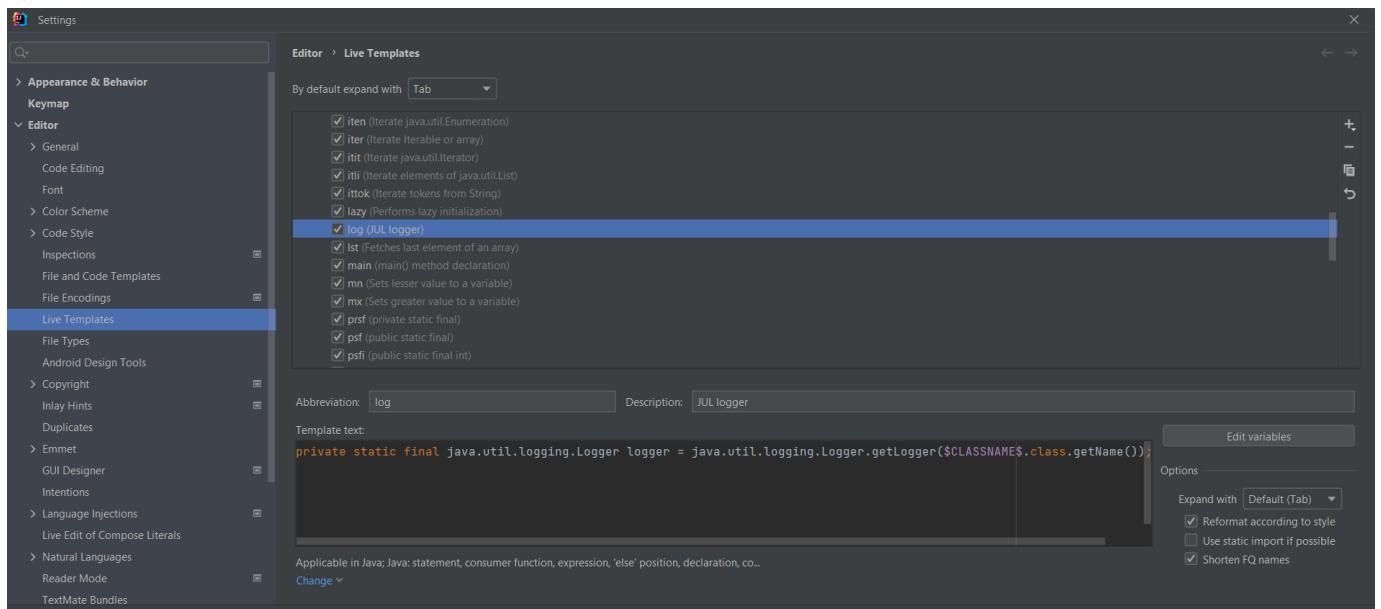
Live templates are used to add custom entries to the code completion, e.g., to quickly create a class-specific logger. Access **Settings/Editor/Live Templates** (CTRL+ALT+S on Windows) and add a new template (+ button). Define the abbreviation to trigger autocompletion at this statement, define the template text, and change the target context ("Java") at the bottom. The template below generates a logger after typing log and pressing CTRL+SPACE. The variable `$CLASS_NAME$` was set to represent the current `className()` under **Edit variable**.

The template:

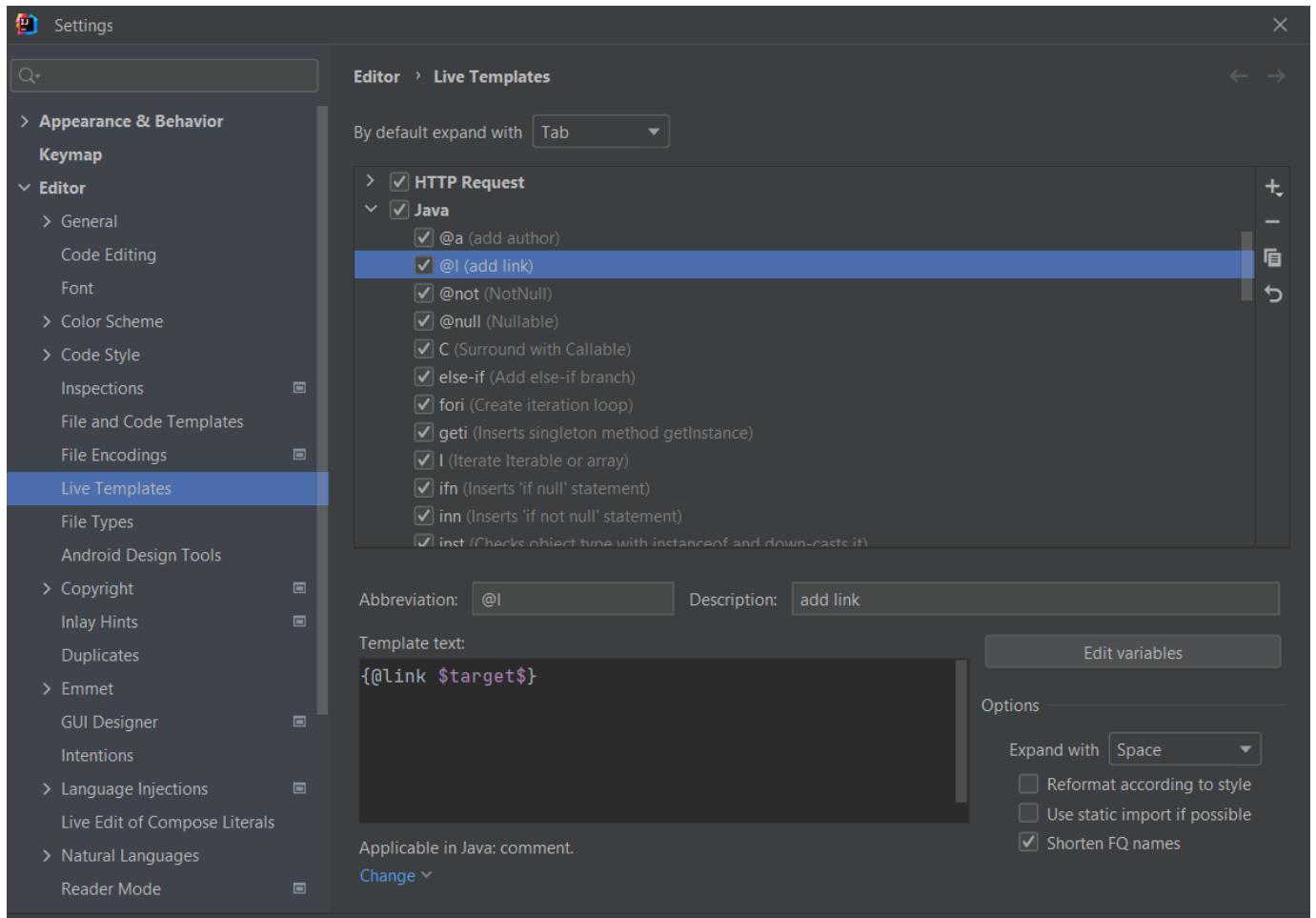
```
private static final java.util.logging.Logger logger = java.util.logging.Logger.getLogger(
$CLASSNAME$.class.getName());
```

Generates the output in class Scan:

```
private static final Logger logger = Logger.getLogger(Scan.class.getName());
```



Another example to create Javadoc links for `@l`. The variable (here `$target$`) places the cursor.



POSTFIX TEMPLATES

Postfix templates allow to surround a preceding variable or expression with code. Similar to live templates but after the expression. Go to the settings (CTRL+ALT S on Windows) and search for postfix templates. Click on the + button to add a new

Java template. Add a key, specify the minimum java version, check "Use static import if possible" and enter the following expression text:

- **key:** nne
- **expression:** Objects.requireNonNullElse(\$EXPR\$, \$END\$)

This will enable the following:

```
// input as variable.nne ...
spectrum.nne
// ... auto complete will produce with _ being the cursor position
Objects.

requireNonNullElse(spectrum, _)

// ... or this when "use static import" is activated
requireNonNullElse(spectrum, _)
```

Troubleshooting

CORRECT JDK SELECTION

Make sure the correct JDK is set in these places:

1. File/Project Structure/SDKs
2. File/Settings/Build, Execution, Deployment/Build Tools → Gradle → Gradle JVM → "Project SDK" this will update automatically if the project SDK changes.
3. File/Settings/Build, Execution, Deployment/Java Compiler → Project Bytecode version → 21 (for JDK 21) Correct run configuration: Select Default JRE (this will update with the project sdk) or select the correct one manually.

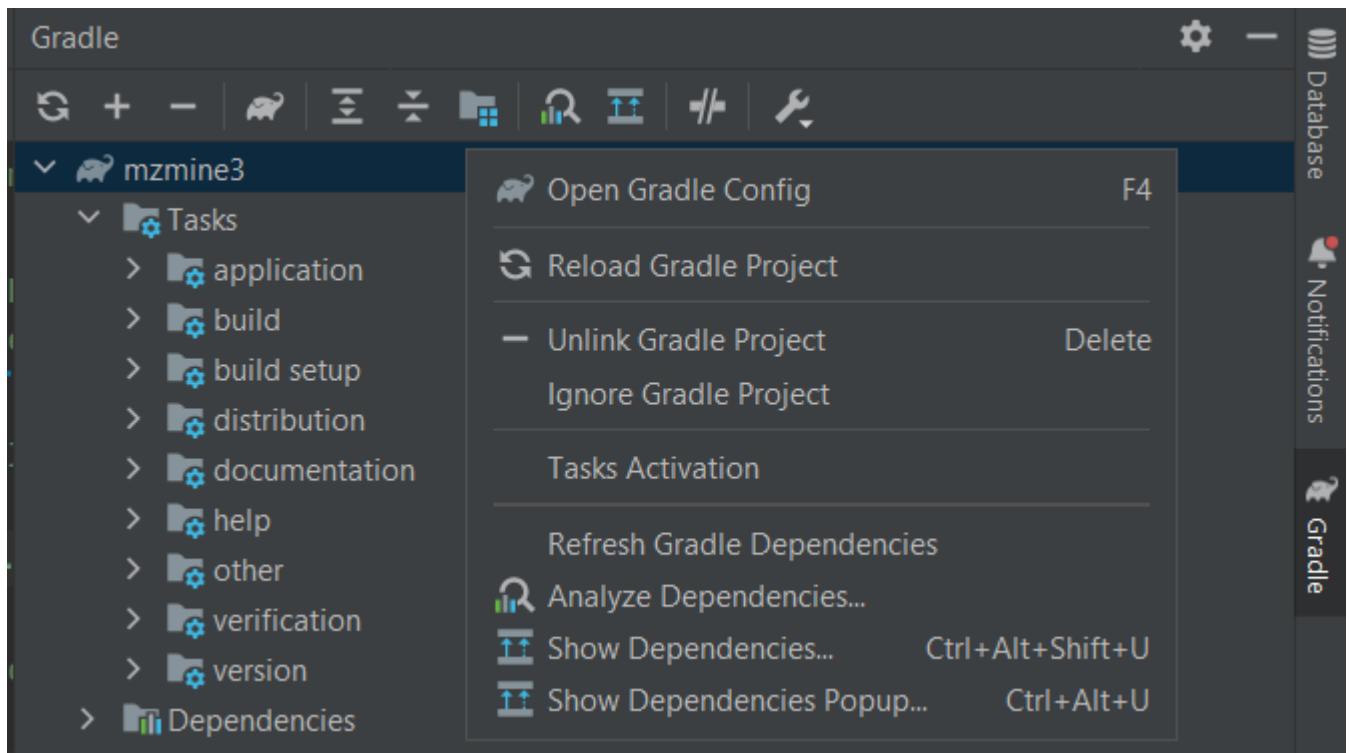
COULD NOT TARGET PLATFORM

Error: When building via gradlew: "Could not target platform: 'Java SE 21' using tool chain: 'JDK 13 (13)'". Solution (Windows): Set the JAVA_HOME environment variable to the JDK 21 root directory. See https://docs.oracle.com/cd/E19182-01/821-0917/inst_jdk_javahome_t/index.html

GRADLE PROJECT NOT IMPORTED

If gradle tool window is not shown:

1. To import the Gradle project navigate to the build.gradle in the project tool window right click → import gradle project. The gradle tool window should now be visible.
2. To update the imports click the update gradle project button in the gradle tool window



Robin Schmid, Steffen Heuckeroth, Steffen Heuckeroth, omokshyna

⌚ September 30, 2025 06:37:19

👤 Robin Schmid, Steffen Heuckeroth, Steffen Heuckeroth, omokshyna

16.1.4 Upgrading JDK version in development builds

1. Update your IDE

IntelliJ and other IDEs need to be updated to handle new JDK versions.

2. Update gradle version in the wrapper

Check the compatibility of gradle and JDKs. Sometimes gradle has to run on an earlier JDK but can compile with the latest JDK as target.

Try upgrading the gradle wrapper:

Or change the gradle version manually in the **/gradle/wrapper/gradle-wrapper.properties** file:

```
distributionUrl=https://services.gradle.org/distributions/gradle-8.1.1-bin.zip
```

3. Change target JDK in build.gradle

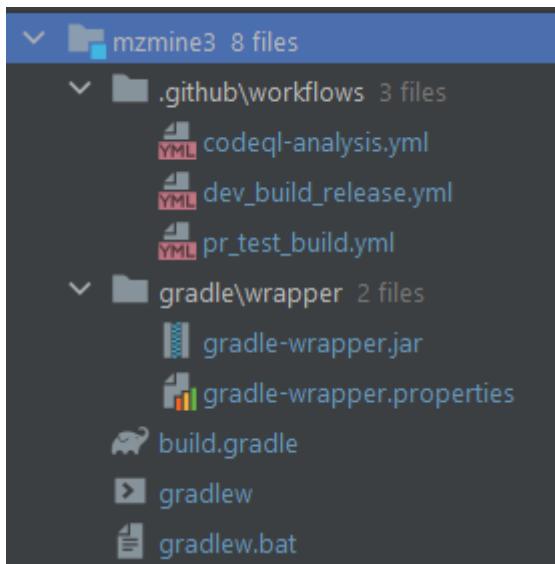
```
java {
    toolchain {
        languageVersion = JavaLanguageVersion.of(20)
        vendor =JvmVendorSpec.ADOPTIUM // force use ADOPTIUM TEMURIN JDK
    }
    // not needed with toolchain command
    // sourceCompatibility = "20"
    // targetCompatibility = "20"
}
```

4. Update gradle project

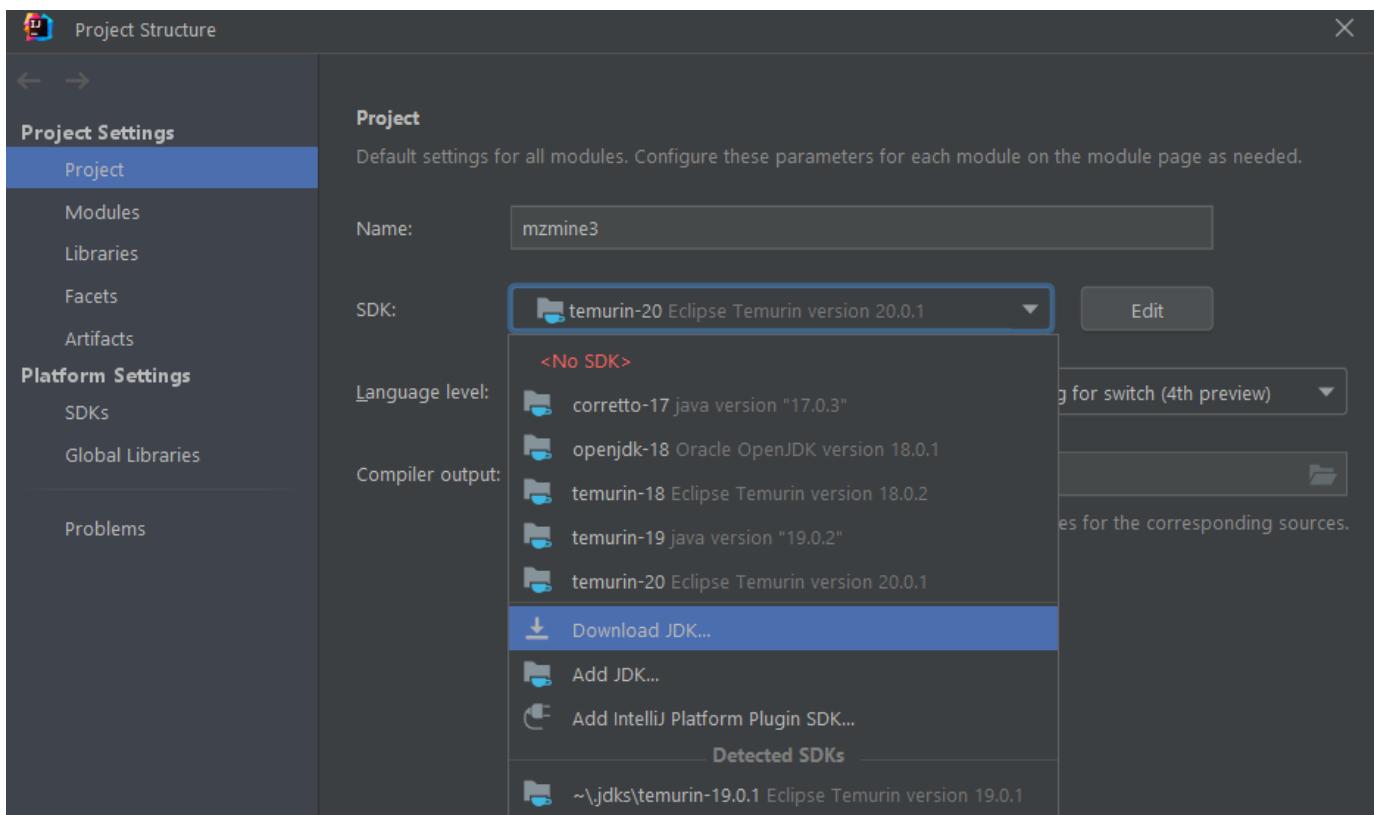
Run update all gradle projects to update all dependencies.

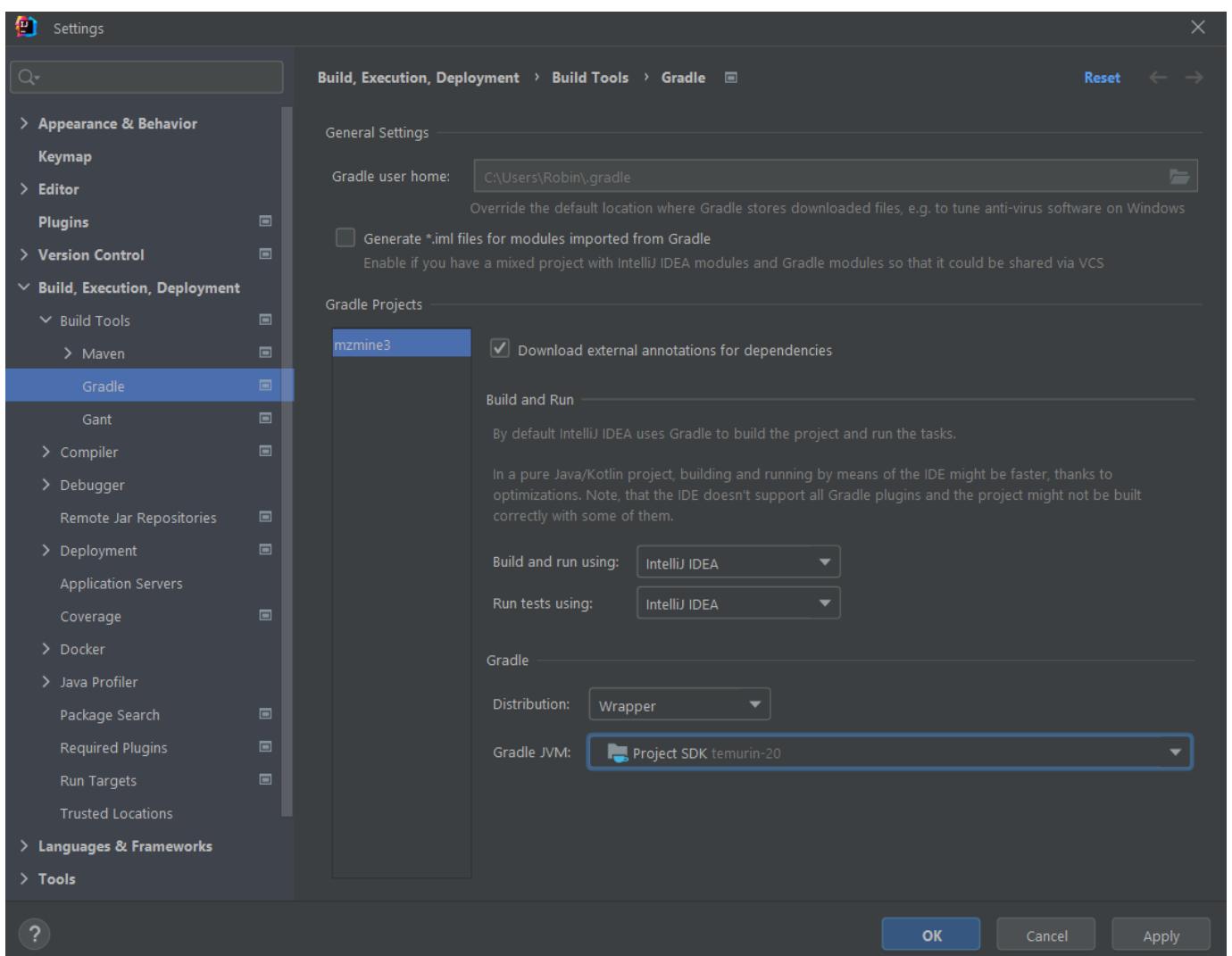
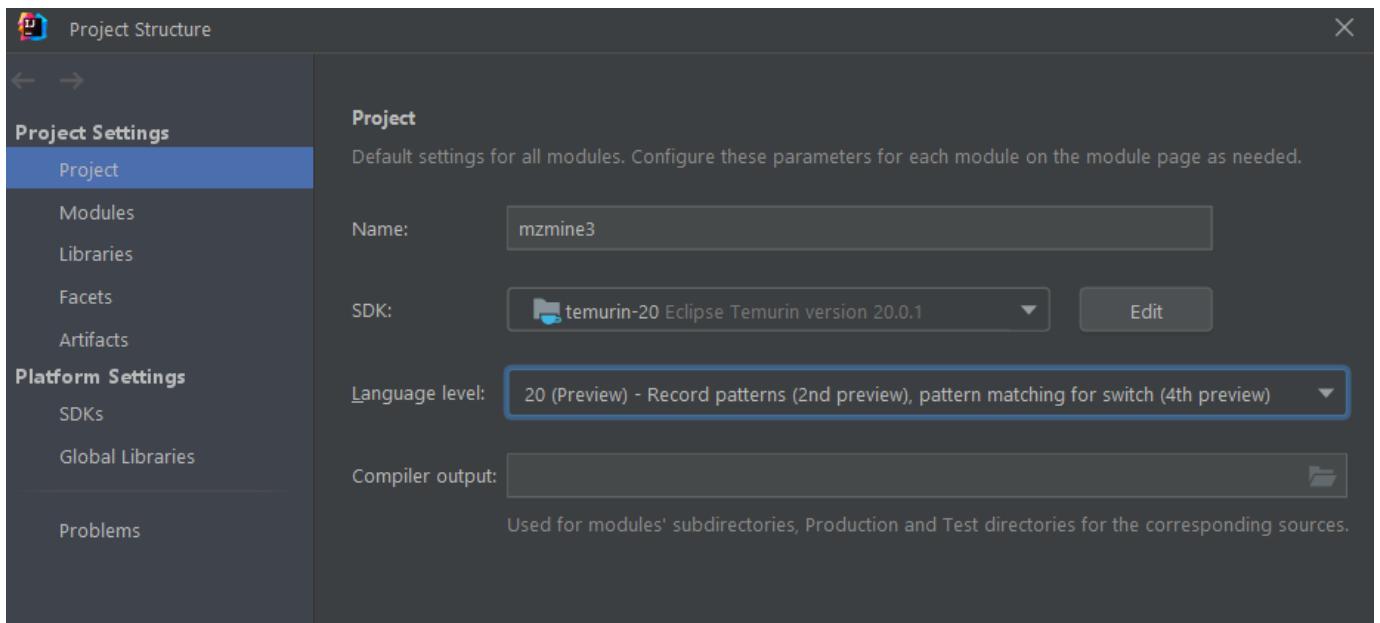
5. Update JDK in GitHub actions

```
steps:
  - uses: actions/checkout@v3
  - uses: actions/setup-java@v3
  with:
    distribution: 'temurin'
    java-version: '20.0.1'
    cache: 'gradle'
```

PR might look like:**IntelliJ specifics**

1. Download a new JDK
2. Set the language Level (preview?)
3. Set the project jdk gradle should use and run tests on intellij (faster)





 August 11, 2023 07:08:55

 Robin Schmid, corinnabrungs

16.1.5 Key steps to mzmine module development

Coding tutorials

Detailed coding tutorials by Robin Schmid:

01. First module and new columns
02. Git, GitHub, contribute code to MZmine

1. Define the role and functionality of your module

Each module in mzmine belongs to some module category (see [MZmine architecture](#)). Before starting, try to define category, to which your module will belong - raw data processing, feature detection, visualization, etc.

A good idea is to browse already existing modules. Currently, MZmine contains more than 100 modules, so there are high chances that one of the existing modules may carry similar functionality to the one you want to code. This can help you to create your own module. Assess the scale of the added functionality and how the new module can be integrated with the existing ones. For example, if you'd like to add a new filter for the feature list rows, the best option might be not to create a new module but to add the new option as a parameter to the already existing Feature list rows filter.

2. Create a new branch locally for your changes

You can either do it from command line using git command `git checkout -b new-branch-name` or directly from your IDE (detailed explanation can be found in the video tutorial).

3. Create a package for your module

Go to the folder that corresponds to the functionality of your module (e.g., dataprocessing for the module that will be process features in the feature list). Create a new package in this folder.



Make sure that a package name starts with a meaningful prefix, such as "filter_" for a module that will filter rows in a feature table.

4. Create necessary classes

Following the module structure described [here](#), create Module, Parameters, and Task classes.



You can directly copy examples for all these classes from the `java/io/github/mzmine/modules/example` folder and **rename them** accordingly. These example files also contain some useful tips.

5. Define module name, description, and category

Start by defining MODULE_NAME and MODULE_DESCRIPTION variables. Assign your module a category by modifying `getModuleCategory()` method. Available categories are listed in MZmineModuleCategory file.

6. Work on your Parameters class

You can search for available parameters in the Parameters package or copy them from similar modules. After you have described all parameters, add parameters to the constructor of your Parameters class

```
super(new Parameter[] {parameters go here})
```

Warning

A parameter in the Parameters class should be public static final.

7. Define `runModule()` method [For the runnable modules]

Here you are creating a task that will be launched after user has chosen parameters. You do not need to worry about task scheduling here, the task is only created and passed here.

8. Work on your Task class

As you extend the AbstractTask class, some functionality will already be defined. To get the values of the parameters from ParameterSet, use the following approach `parameters.getValue(YourClassParameters.parameterName)`. You can modify the message by the logger, check for cancelled state, and update progress. Before progress update, implement the logic of your calculations.

Warning

Do not forget to modify `addAppliedMethodsAndResultsToProject`, so that the method info will be accessible in the feature list information.

Tip

If you have to implement some method to process scans, check ScanUtils first, it might already have been implemented.

9. Integrate a newly created module into mzmine GUI

As the last step, for your module to be accessible, add it to the `AbstractWorkspace.java` or the `AcademicWorkspace.java` file. If you wish your module to be accessible in the batch mode, add it in the `BatchModeModulesList.java`

10. Test and debug

Before pushing your final changes to the remote branch, try to thoroughly test your code. Think about corner cases and possible bugs. Using IDE tools can be extremely useful in this step.

11. Create documentation

Prepare a description for your module that will go into the documentation. Creating a thorough documentation will help more people to use and understand your module. The documentation template can be found [here](#)

12. Push to the remote branch and create a pull request

After you coded and tested your module, you can push your changes to the remote branch and create a pull request. After your pull request will be reviewed by one of the main contributors. There might be some comments that you'll have to address. After that, your module will be merged into the master branch and become a part of MZmine!

Jonas L., Robin Schmid, Steffen Heuckeroth, omokshyna

 November 25, 2025 10:29:24

 Jonas L., Robin Schmid, Steffen Heuckeroth, omokshyna

16.1.6 MZmine module overview

List of MZmine modules

List of MZmine modules extracted on: 2022/12/01

Please note that some contributors might not be correctly reflected due to reformatting and refactoring. Use git-blame at the [MZmine code repository](#) for a more detailed insight.

name	description	main class	sub class	contributors	path
13C isotope filter (formerly: isotope grouper)	This module detects isotopic features and removes them from the feature list. Its isotope patterns are limited to detected features. For a more comprehensive isotope pattern coverage and isotopes other than 13C, use the Isotope finder.	Feature filtering	Isotopes	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/IsotopeGrouper
2D visualizer	Two dimensional visualization of feature lists.	Visualization	Visualization	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/twod/TwoDVisu
3D visualizer	3D visualization of raw data.	Visualization	Visualization	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/fx3d/Fx3DVisua
GC aligner	This module calculates pairwise convolution integral for each pair of unaligned peaks in order to find the best alignment	Feature detection	Alignment	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; tnguy271	src/main/java/io/dataprocessing/ADAP3Aligner
Chromatogram Builder	This module connects data points from mass lists and builds chromatograms.	Feature detection	Chromatogram detection	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/featdet_adapch ModularADAPC
ADAP3D feature detection (experimental)	This module detect peaks from profile spectra using the ADAP3D algorithm developed by the Xiuxia Du lab. The module requires no parameter settings. Note that it will not work with centroid spectra.	Feature detection	Chromatogram detection	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/ADAP3DModule

name	description	main class	sub class	contributors	path
Adap Resolver	Resolves chromatograms to features.	Feature processing	Feature resolving	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/featdet_chrom/ADAPpeakpicker
Add ion identities to networks	This module adds ion identities to existing networks	Feature processing	Ion identity networking	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/addionannotation/AddIonNetwork
Aird file import	This module imports aird raw data into the project.	Spectral data	Raw data import	Robin Schmid; Tomas Pluskal; lms19880914	src/main/java/io/import_rawdata
Align scans (LC/MS or MS level 1)	This module align scans for small fluctuations correlating consecutive scans.	Spectral data	Raw data filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/AlignScansModule
Annotate possible isomers	Annotates features with the same (or similar) m/z but different mobilities within a specified retention time window.	Feature processing	Feature grouping	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/AnnotateIsomers
Average formulas for annotation networks	Creates a list of average molecular formulas from all formula lists of all ions in this network (e.g., M+H and M+Na)	Feature processing	Ion identity networking	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/formula/createavg/CreateAvgNetwork
Baseline correction	This module performs a baseline correction on raw data files.	Spectral data	Raw data filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/BaselineCorrection
Baseline resolver	Resolves chromatograms to features.	Feature processing	Feature resolving	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/featdet_chrom/BaselineFeature
Batch mode		Project	Project		

name	description	main class	sub class	contributors	path
	This module allows execution of multiple processing tasks in a batch.			Kirill Ponomarov; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/BatchModeMod
Batch spectral library generation	Uses metadata sheet to generate spectral library	Feature IO	Feature list export	Steffen Heuckeroth; Robin Schmid	src/main/java/io/spectraldbsubn/LibraryBatchGe
Bruker TDF file import	This module imports raw data into the project.	Spectral data	Raw data import	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/import_rawdata/TDFImportMod
CAMERA search	This method searches for pseudo-spectra using the CAMERA algorithm.	Feature processing	Annotation	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/CameraSearch
CCS calculation module	Calculates CCS values for features.	Feature processing	Annotation	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; tarush.singh	src/main/java/io/dataprocessing
Calculate EC-MS potentials	Calculates metabolite formation potentials in EC-MS experiments.	Feature processing	Annotation	Tomas Pluskal; m_wesn01	src/main/java/io/dataprocessing/CalcEcmsPoten
CentWave resolver	Resolves chromatograms to features.	Feature processing	Feature resolving	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/featdet_chrom/ CentWaveResol
Check all ion identities by MS/MS	Checks ion identities (in-source fragments and multimers) by MS/MS	Feature processing	Ion identity networking	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/checkmsms/IonNetworkMS
Chromatogram and spectra visualizer	Inspect LC-MS data by navigating through raw data.	Visualization	Visualization	Annexhc; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; annexhc; knakul853; Robin Schmid	src/main/java/io/chromatogramma/Chromatogram

name	description	main class	sub class	contributors	path
Clear feature list annotations	This method clears the annotations from the feature list	Feature filtering	Feature list filtering	Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/idae/dataprocessing/ClearFeatureAnnotations
Clear ion identities	Deletes all ion identities and networks of all rows	Feature processing	Ion identity networking	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/idae/dataprocessing/clearionids/ClearIonIdentites
Clear project	Clearing all project files and feature lists	Project	Project	Robin Schmid	src/main/java/idae/clear_project/ClearProject
CliqueMS annotation	This method groups features and annotates the groups/cliques using CliqueMS algorithm.	Feature processing	Annotation	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; SurajV; Tomas Pluskal	src/main/java/idae/dataprocessing/CliqueMSModule
Correlated features $\Delta m/z$ histogram	This module plots all m/z deltas between correlated features in a histogram and offers a Gaussian fit.	Visualization	Visualization feature list	Olena Mokshyna; Tomas Pluskal; Robin Schmid	src/main/java/idae/scan_histogram/CorrelatedFeatures
Correlation coefficient histogram (grouped feature shapes)	This module plots all correlations between feature shapes.	Visualization	Visualization feature list	Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/idae/histo_feature_c/FeatureCorrelat
Crop filter	This module performs cropping of raw data files.	Spectral data	Raw data filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/idae/dataprocessing/CropFilterModule
Custom DB search	Search spectrum against local csv database	Feature processing	Annotation	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/idae/spectra/simplest/customdatabase/CustomDBSpec
Custom database search	This method searches a custom database (CSV file) using m/z and retention time values.	Feature processing	Annotation	Olena Mokshyna; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/idae/dataprocessing/LocalCSVDatabase

name	description	main class	sub class	contributors	path
DIA pseudo MS2 builder (experimental)	Builds pseudo MS2 scans based on feature shape correlation.	Feature filtering	Feature list filtering	Steffen Heuckeroth; Robin Schmid	src/main/java/ida/processing/DiaMs2CorrMo
Data point processing	Processes individual spectra with multiple steps like mass detection, deisotoping, annotation	Raw data	Annotation	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/spectra/simple/DataPointProce
Duplicate peak filter	This method removes duplicate peaks (peaks with same retention times and m/z) from the feature list.	Feature filtering	Feature list filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/DuplicateFilter
Export CSV (legacy MZmine)	Legacy MZmine 2 method: This method exports the feature list contents into a CSV (comma-separated values) file.	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/export_features/LegacyCSVExp
Export MS/MS spectra	Export MS/MS spectra of detected features.	Spectral data	Export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/featurelisttable/MSMSExportM
Export MSn trees	This method exports the feature list contents into a CSV (comma-separated values) file.	Spectral data	Raw data export	Tomas Pluskal; Robin Schmid	src/main/java/ida/export_msn_trees
Export feature networks to csv	Export results from correlation/MS annotation as networks to csv	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; sanbit876	src/main/java/ida/processing/ExportCorrAnn
Export for SIRIUS	This method exports a MGF file that contains for each feature, (1) the deconvoluted MS1 isotopic pattern, and (2) the MS/MS spectrum (highest precursor ion intensity). This file can be open and processed with Sirius, https://bio.informatik.uni-jena.de/software/sirius/ .	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/ida/export_features/Sirius
Export images to csv	Exports spatially resolved data of features to csv files.	Feature IO	Export	Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/export_image_csv/ImageToCsvExp

name	description	main class	sub class	contributors	path
Export isotope pattern	Export isotope pattern of detected features.	Spectral data	Export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/featurelisttable/IsotopePattern
Export library analysis csv	Cosine similarity of library to csv	Feature IO	Feature list export	Robin Schmid; Tomas Pluskal; Robin Schmid	src/main/java/io/export_library/LibraryAnalysis
Export scans	Export scans from raw data.	Spectral data	Export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_scans/ExportScans
Export scans into one file	Export scans or mass lists into one file	Spectral data	Raw data export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_scans/ExportScansFr
Export to CSV file (modular)	This method exports the feature list contents into a CSV (comma-separated values) file.	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_features/CSVExportMod
Export to GNPS-GC-MS (with ADAP)	GNPS GC-EI-MS workflow. Export the mgf, quant table and Kovats reference file (optionally).	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_features/GnpsGcExportA
Export to MGF file (ADAP)	This method exports all ADAP clustered spectra in a feature lists to an MGF file. Used in the GNPS GC-MS workflow.	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_features/AdapMgfExport
Export to MSP file (ADAP)	This method exports all ADAP clustered spectra in a feature lists to an MSP file.	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_features/AdapMspExport
Export to MetaboAnalyst file	This method exports the feature list contents into a CSV (comma-separated values) file that is formatted for easy processing by the MetaboAnalyst service (http://www.metaboanalyst.ca).	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_features/MetaboAnalyst
Export to SQL database	This method exports the feature list contents into an SQL database using JDBC.	Feature IO	Feature list export	Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/export_features
Export to mzTab file	This method exports the feature list contents into a mzTab file.	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_features/MzTabExportM

name	description	main class	sub class	contributors	path
Export to mzTab-m file.	This method exports the feature list contents into a mzTab-m file.	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_features/MZTabmExport
Export/Submit to GNPS-FBMN	GNPS feature-based molecular networking export and submit module. Exports the MGF file for GNPS (only for MS/MS), the quant table (CSV export) and additional edges (ion identity networks and correlation)	Feature IO	Feature list export	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_features/GnpsFbmnExport
External CCS calibration	Uses an externally calculated CCS calibration to calculate CCS values.	Feature processing	Annotation	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/ExternalCCSCalibration
Extract scans	Extract a defined number of scans around a center time	Spectral data	Raw data export	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/ExtractScansModule
FTMS shoulder peaks filter	This method filters mass lists and removes residual signals known as 'shoulder peaks' and commonly observed in FTMS data.	Spectral data	Raw data filtering	Robin Schmid; Roman Bushuiev; Sonja-analytik; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/ShoulderPeaksFilter
Feature intensity plot	Feature intensity plot.	Visualization	Visualization feature list	Olena Mokshyna; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/intensityplot/IntensityPlot
Feature list blank subtraction	Subtracts features appearing in (procedural) blank measurements feature list from an aligned feature list.	Feature filtering	Feature list filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/FeatureListBlankSubtraction
Feature list info visualizer	Generates a batch from the list of applied methods to reproduce this feature list.	Visualization	Visualization	Roman Bushuiev; Tomas Pluskal; m_wesn01	src/main/java/io/infovisualizer/InfoVisualizer

name	description	main class	sub class	contributors	path
Feature list rows filter	This method removes certain rows from a feature list based on given restrictions.	Feature filtering	Feature list filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/dataprocessing/RowsFilterModule
Feature table	MZmine 3 feature table.	Visualization	Visualization	Annexhc; Kirill Ponomarov; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/featurelisttable/FeatureTableFactory
Formula prediction	This method gets the predicted formula for each unknown compound	Feature processing	Annotation	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/FormulaPrediction
Formula prediction	Predict molecular formulae for features.	Feature processing	Annotation	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Weiching Chen; knakul853	src/main/java/io/dataprocessing/FormulaPrediction
Formula prediction (on annotation networks)	This method predicts formulae for each ion identity in all annotation networks (generated by MS annotations module or similar)	Feature processing	Ion identity networking	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/formula/prediction/FormulaPrediction
Formula prediction on a single spectrum	Predict formulae in a single spectrum.	Raw data	Annotation	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/spectra/simplest/sumformula/SumFormulaSp
Frame visualizer	Visualizes frames from ion mobility raw data files.	Visualization	Visualization	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/frames/FrameV

name	description	main class	sub class	contributors	path
Graphical manual picker	Manually pick or reintegrate features in EICs by point and click operations.	Feature processing	Feature resolving	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/XICManualPicker
Graphics export	Export charts to vector or pixel graphics	Visualization		Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/graphicsexport
GridMass - 2D feature detection	This module detect peaks based on a 2D peak recognition procedure.	Feature detection	Chromatogram detection	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/dataprocessing/GridMassModule
Group MS2 scans with features	This method assings all MS2 scans within range to all features in this feature list	Feature filtering	Feature list filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/dataprocessing/GroupMS2Module
Hierarchical aligner (GC)	This method aligns detected peaks using a match score. This score is calculated based on the MZ profile and RT of each peak using preset tolerances.	Feature detection	Alignment	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/HierarAlignerGC
Hierarchical Clustering	This method combines peaks into analytes and constructs fragmentation spectrum for each analyte	Feature filtering	Spectral deconvolution	Aleksandr Smirnov; Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Varun Suresh; tnguy271	src/main/java/io/dataprocessing/ADAPHierarchical
Histogram plot	Histogram plot	Visualization	Visualization feature list	Olena Mokshyna; Roman	src/main/java/io/histogram/Histo

name	description	main class	sub class	contributors	path
				Bushuiev; Steffen Heuckeroth; Tomas Pluskal	
ICP MS CSV file import	This module imports raw data into the project.	Spectral data	Raw data import	Steffen Heuckeroth; Tomas Pluskal	src/main/java/import_rawdata/IcpMsCVSImport
IMS Raw data visualizer	Multidimensional raw data inspection of IMS raw data.	Visualization	Visualization	Annexhc; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; aiden; knakul853	src/main/java/ims/ImsVisualizer
IMS mobility mz plot	Heatmap view for detected features.	Visualization	Visualization	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ims_mobilitymz/IMSMobilityMz
Image builder	This module connects data points from mass lists and builds images.	Feature detection	Chromatogram detection	Annexhc; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/dataprocessing/ImageBuilderM
Image visualizer	Image visualizer	Visualization	Visualization	Annexhc; Olena Mokshyna; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/image/ImageVi
Import GNPS results	Imports GNPS feature based molecular networking results into the selected feature list (library matches)	Feature processing	Annotation	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/dataprocessing/GNPSResultsIn
Import MS data	This module combines the import of different MS data formats and provides advanced options	Spectral data	Raw data import	Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; lms19880914; Robin Schmid	src/main/java/import_rawdata/AllSpectralData
Import from csv file.	This method imports the feature list contents from a csv file.	Feature IO	Feature list import	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/import_feature/

name	description	main class	sub class	contributors	path
Import mzTab file	This method imports a feature list from a mzTab file.	Feature IO	Feature list import	Nils Hoffmann; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/import_features/MzTabImportM
Import mzTab-m file	This method imports a feature list from a mzTab-m file.	Feature IO	Feature list import	Nils Hoffmann; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/import_features/MZTabmImport
Import spectral library	This method imports spectral libraries to speed up spectral library matching	Spectral data	Raw data import	Olena Mokshyna; Robin Schmid; Tomas Pluskal; Robin Schmid	src/main/java/io/import_spectra/SpectralLibrary
Ims expander	Expands rt chromatograms in mobility dimension.	Feature detection	Chromatogram detection	Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/ImsExpanderM
Injection time analysis	Plots the minimum signal intensity against the injection time to see trends in trap based mass spectrometers	Visualization	Visualization	Olena Mokshyna; Tomas Pluskal; Robin Schmid	src/main/java/io/injection_time/I
Internal standard CCS calculation	Calculates CCS values based on an reference compound list and it's detected features in a feature list.	Feature processing	Annotation	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/ReferenceCCS
Ion Identity Molecular Networks	Visualize the results of the MS annotation module	Visualization	Visualization feature list	Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/networking/Ann
Ion identity network refinement	Refinement of ion identity networks	Feature processing	Ion identity networking	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/refinement/Ioni
Ion identity networking	This method searches for adducts, in-source fragments, and clusters	Feature processing	Ion identity networking	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/ionidnetworkin
Ion mobility raw data overview	Visualizes ion mobility raw data files.	Visualization	Visualization	Olena Mokshyna; Roman Bushuiev; Steffen	src/main/java/io/rawdataoverview/IMSRawDataO

name	description	main class	sub class	contributors	path
				Heuckeroth; Tomas Pluskal	
Ion mobility trace builder	Builds ion mobility traces for a raw data file	Feature detection	Chromatogram detection	Annexhc; Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/IonMobilityTra
Isotope pattern preview	Calculate and view isotope patterns.	Other	Tools	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/ida/isotopatternn/IsotopePatternPr
Isotope peak scanner	Searches for m/z and intensity distribution of a calculated isotope pattern within a feature list.	Feature filtering	Isotopes	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/IsotopePeakScan
Isotopic peaks finder	Find isotope signals in the FWHM of all features in the list	Feature filtering	Isotopes	Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; tarush.singh	src/main/java/ida/processing/IsotopeFinderM
Join aligner	This method aligns detected peaks using a match score. This score is calculated based on the mass and retention time of each peak using preset tolerance.	Feature detection	Alignment	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/ida/processing/JoinAlignerMod
Kendrick mass plot	Kendrick mass plot.	Visualization	Visualization feature list	Annexhc; Olena Mokshyna; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; annexhc; knakul853	src/main/java/ida/kendrickmassp
Kovats index extraction	Extract Kovats indexes from a raw data file and save to csv.	Other	Tools	Robin Schmid; Roman Bushuiev;	src/main/java/ida/KovatsIndexExt

name	description	main class	sub class	contributors	path
				Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	
LC-Image Aligner	Aligns LC and imaging measurements based on m/z and mobility. Images are aligned to all LC features that match, only the best match is retained.	Feature detection	Alignment	Olena Mokshyna; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/mzmine/dataprocessing/LcImageAligner
Learner module	This module is for learners only	Feature filtering	Feature list filtering	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/mzmine/LearnerModule
Library search results window	Visualize spectral database search results.	Visualization	Visualization	Kevin-Murray; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; murra668; Robin Schmid	src/main/java/io/mzmine/spectra/spectra/SpectralIdentification
Library submit	Submit MS2 spectra for spectral libraries.	Spectral data	Export	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/mzmine/spectraldbsubmit
Linear normalizer	Linear normalizer divides the height (or area) of each feature in the feature list by a normalization factor, determined according to the given normalization type.	Feature processing	Normalization	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; tnguy271	src/main/java/io/mzmine/dataprocessing/LinearNormalizer
Lipid annotation	This method searches and annotates for features which m/z value matches a predicted mass of selected lipids.	Feature processing	Annotation	Annexhc; Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; SurajV; Tomas	src/main/java/io/mzmine/dataprocessing/LipidSearchModule

name	description	main class	sub class	contributors	path
				Pluskal; Robin Schmid	
Lipid annotation	Lipid annotation on individual spectra	Raw data	Annotation	Annexhc; Robin Schmid; Steffen Heuckeroth; SurajV; Tomas Pluskal	src/main/java/ide/spectra/simplest/lipidsearch/Lipid
Local minimum resolver	Resolves chromatograms to features.	Feature processing	Feature resolving	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/dataprocessing/featdet_chromaminimumsearch/MinimumSearch
MS/MS data visualizer	Scatter plot for MS/MS data visualization, fouraxes types supported: retention time, precursor m/z, product m/z, neutral loss	Visualization	Visualization	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Sandeep; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/msms/MsMsVis
MS/MS quality export	Exports metrics for MS/MS spectra	Other	Tools	Steffen Heuckeroth; SteffenHeu	src/main/java/ide/export_msmsquality/MsMsQualityEx
MS/MS spectral networking (Molecular networking)	Checks MS2 similarity of all rows within the groups or on all networks and between networks	Feature processing	Feature grouping	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/ide/dataprocessing/similarity/MS2
MS2 similarity search	This method searches for similar MS2 fragmentation spectra between two feature lists	Feature processing	Annotation	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/dataprocessing/Ms2SearchModule
MSn Tree Visualizer	Visualizes MSn spectra in a tree structure and all spectra of the same level in one spectra chart	Visualization	Visualization	Olena Mokshyna; Robin Schmid; Tomas Pluskal; Robin Schmid	src/main/java/ide/spectra/msn_tree/MSnTreeVisual
MSn feature list builder	This module looks through the whole raw data for MSn scans and makes a list of chromatographic	Feature detection	Chromatogram detection	Kevin-Murray; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/dataprocessing/MsnFeatureDetector

name	description	main class	sub class	contributors	path
	features using the precursor mass.				
MSn tree feature list builder	Builds all MSn trees and extracts chromatograms for the precursor m/z	Feature detection	Chromatogram detection	Robin Schmid	src/main/java/ida/processing/MsnTreeFeature
MZ range from formula calculator	Calculate mz range from formula and mz tolerance	Other	Tools	Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/mzrangecalculator/MzRangeFormu
MZ range from tolerance calculator	Calculate mz range from mz tolerance and mass	Other	Tools	Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/mzrangecalculator/MzRangeMassTol
Main resolver	Resolves chromatograms to features. Provides additional parameters and orchestrates the algorithms	Feature processing	Feature resolving	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/featdet_chrom/FeatureResolve
Manual picker	Manually pick or reintegrate features in EICs.	Feature processing	Feature resolving	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/ManualFeature
Mass calibration	This module estimates systematic error of measurement by analysing differences between actual mass peaks and masses of assigned compounds, it then shifts the mass peaks to account against that bias.	Spectral data	Raw data methods	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/ida/processing/MassCalibration
Mass detection	This module detects individual ions in each scan and builds a mass list for each scan.	Spectral data	Raw data methods	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/ida/processing/MassDetection
Massvoltammogram chart	A chart that combines a specific instrumental setup (redox cell) with mass spectrometry. Different timepoints are converted into the potentials applied to	Visualization	Visualization	Tomas Pluskal; m_wesn01	src/main/java/ida/massvoltammogram/Massvoltammogram

name	description	main class	sub class	contributors	path
	the sample, this is plotted as 3D spectra against this potential.				
Merge files	Merge all scans of multiple files to one raw data file	Spectral data	Raw data methods	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/dataprocessing/RawFileMerge
Merge lists	This method merges feature lists by appending all rows into a new list. Perform alignment before to align all features from comparable samples and use this method to merge feature lists that should not be aligned: e.g., positive and negative mode data.	Feature detection	Alignment	Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/MergeAlignerM
Mobility scan merging module	Merges mobility scans at the same retention time to a summed frame spectrum.	Spectral data	Raw data methods	Olena Mokshyna; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/MobilityScanM
Mobilogram summing	Bins intensities within given mobility ranges to correct for noise in mobilograms.	Feature detection	Chromatogram detection	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/MobilogramBin
MsMs Spectra merge	Merge multiple MS2 spectra to a single spectrum.	Spectral data		Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/msmsspectram/MsMsSpectraM
Multivariate Curve Resolution	This method combines peaks into analytes and constructs fragmentation spectrum for each analyte	Feature filtering	Spectral deconvolution	Aleksandr Smirnov; Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Varun Suresh; tnguy271	src/main/java/io/dataprocessing/ADAPMultivariate
NIST MS Search		Feature processing	Annotation	Kevin-Murray; Olena	

name	description	main class	sub class	contributors	path
	This method searches spectra against the NIST library.			Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; murra668; Robin Schmid	src/main/java/io/dataprocessing/NistMsSearchM
Neutral loss filter	Searches for neutral losses within a feature list.	Feature filtering	Feature list filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/NeutralLossFilter
Noise amplitude resolver	Resolves chromatograms to features.	Feature processing	Feature resolving	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/featdet_chromnoiseamplitude/NoiseAmplitude
Online DB search	Online DB search on individual spectra	Raw data	Annotation	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/spectra/simplesonlinedatabase/OnlineDBSpect
Online database search	This module attempts to find those peaks in a feature list, which form an isotope pattern.	Feature processing	Annotation	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; SurajV; Tomas Pluskal; knakul853	src/main/java/io/dataprocessing/OnlineDBSearch
Open project	This module opens an existing MZmine project. The current workspace will be discarded.	Project	Project I/O	Annexhc; Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; knakul853; Robin Schmid	src/main/java/io/projectload/Proj
Path aligner			Alignment	Robin Schmid; Roman	

name	description	main class	sub class	contributors	path
	Description not yet available.	Feature detection		Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/PathAlignerModule
Peak comparison rows filter	This method removes certain rows from a feature list based on peak comparisons.	Feature filtering	Feature list filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/PeakComparisonModule
Peak filter	This method removes certain peak entries based on given restrictions.	Feature filtering	Feature list filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/FeatureFilterModule
Peak finder	Fill gaps in aligned feature lists of LC-MS measurements.	Feature processing	Feature resolving	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/PeakFinderModule
Peak finder (multithreaded)	This method fills the missing peaks (gaps) in the feature list by searching for a peak in the raw data.	Feature detection	Gap filling	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ida/processing/multithreaded/MultiThreadPeakFiller
Precursor search in spectral libraries	This method searches all peaklist rows against a local spectral libraries to identify possible precursor ions	Feature processing	Annotation	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/ida/processing/PrecursorDBSearcher
Processing wizard	Creates processing batches with a reduced set of parameters.	Other	Tools	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/ida/batchwizard/BatchWizard
Project metadata import	This module imports metadata into the project from .tsv-format files.	Project	Project metadata	Kirill Ponomarov; Tomas Pluskal; Robin Schmid	src/main/java/ida/projectmetadata/ProjectMetadataImport

name	description	main class	sub class	contributors	path
Quality parameters	Calculates quality parameters such as FWHM, asymmetry factor, tailing factor, S/N ratio.	Other	Tools	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/qualityparameters/QualityParamet
RANSAC aligner	Feature list aligner based on the RANSAC algorithm.	Feature detection	Alignment	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/RansacAligner
Raw data export	This module exports raw data files from your MZmine project into various formats	Spectral data	Raw data export	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_rawdata
Raw data export to netCDF	This module exports raw data files from your MZmine project into various formats	Spectral data	Raw data export	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_rawdata/NetCDFExport
Raw data file rename	Renames raw data files.	Other	Tools	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/rawfilerename/
Raw data overview	Raw data overview	Visualization	Visualization	Annexhc; Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; knakul853; Robin Schmid	src/main/java/io/rawdataoverview
Recursive IMS Builder	Builds m/z traces for ion mobility spectrometry data	Feature detection	Chromatogram detection	Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/RecursiveIMSB
Relations between ion networks	This method searches for relations between ion networks (e.g., chemical modifications, heterodimers, covalently bound multimers	Feature processing	Ion identity networking	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/relations/IonNe
Retention time calibration	The retention time calibration attempts to	Feature processing	Normalization	Olena Mokshyna;	

name	description	main class	sub class	contributors	path
	reduce the deviation of retention times between feature lists, by searching for common features in these lists and using them as calibration standards.			Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/RTCalibrationM
Same RT and m/z range gap filler	This method fills the missing peaks (gaps) in the feature list by looking at the whole m/z and retention time range of the feature list row and adding all raw data points in the same range.	Feature detection	Gap filling	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/SameRangeGap
Save project	This module saves the current MZmine project.	Project	Project I/O	Annexhc; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; knakul853; Robin Schmid	src/main/java/io/projectsave/Pro
Save project as	This module saves the current MZmine project.	Project	Project I/O	Annexhc; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; knakul853; Robin Schmid	src/main/java/io/projectsave/Pro
Savitzky Golay resolver	Resolves chromatograms to features.	Feature processing	Feature resolving	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/featdet_chromsavitzkygolay/SavitzkyGolayR
Scan by scan filtering	This module performs filtering algorithms on each scan individually.	Spectral data	Raw data filtering	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; knakul853	src/main/java/io/dataprocessing/ScanFiltersMod
		Spectral data		Olena Mokshyna;	

name	description	main class	sub class	contributors	path
Scan smoothing (LC/MS or MS level 1)	This module averages intensity values within a time-scan frame.		Raw data filtering	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/dataprocessing/ScanSmoothing
Scatter plot	Scatter plot.	Visualization	Visualization feature list	Olena Mokshyna; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/scatterplot/Scat
Set CCS calibration	Sets a CCS calibration to a raw data file.	Feature processing	Annotation	Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/dataprocessing/SetCCSCalibrat
Single Spectrum library search	Search single spectrum in a spectral library.	Raw data	Annotation	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/ide/spectra/simplespectraldatabase/SingleSpectrum
Smoothing	Smooths intensity along the retention time and/or mobility dimension.	Feature detection	Resolving	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/dataprocessing/SmoothingMod
Sort feature lists	Sort selected feature lists alphabetically	Feature detection	Feature list methods	Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/sortfeaturelists
Sort formulas	This module sorts all formula lists of rows in a feature list	Feature processing	Annotation	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/dataprocessing/FormulaSortMod
Sort raw data files	Sort selected raw data files alphabetically	Spectral data	Raw data methods	Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/sortdatafiles/SortDatafil
Sort results of spectra library search	This method sorts all results of spectral library search.	Feature processing	Annotation	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/ide/dataprocessing/SortSpectralMa

name	description	main class	sub class	contributors	path
Spectra stack visualizer	Visualize multiple spectra.	Visualization	Visualization	Tomas Pluskal; Robin Schmid	src/main/java/io/spectra/spectra/SpectraStackVi
Spectra visualizer	Spectra visualizer.	Visualization	Visualization	Annexhc; Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; SurajV; Tomas Pluskal; aiden; Robin Schmid; ? ukasz Fiszer	src/main/java/io/spectra/simple/SpectraVisualiz
Spectral library search	This method searches all feature list rows (from all feature lists) against a local spectral libraries (needs to be loaded first).	Feature processing	Annotation	Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/dataprocessing/SpectralLibrary
Spectral mirror (USI enabled)	Compare spectra	Other	Tools	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/spectra/simple/MirrorScanMod
Standard compound normalizer	This module normalizes the feature heights and areas according to selected internal standards.	Feature processing	Normalization	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/StandardComp
Submit GNPS MASST search	Submit an MS2 spectrum to GNPS MASST search, searching against public data.	Feature IO	Feature list export	Robin Schmid; Tomas Pluskal; Robin Schmid	src/main/java/io/export_features/GnpsMasstSub
TIC/XIC visualizer	TIC/XIC visualizer.	Visualization	Visualization	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/chromatogram/Chromatogram

name	description	main class	sub class	contributors	path
TSF import module	Imports Bruker .d directories with tsf data files.	Spectral data	Raw data import	Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/io/import_rawdata/TSFImportModule.java
Targeted feature detection	Targeted feature detection	Feature detection	Chromatogram detection	Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/TargetedFeatureDetection.java
Thermo RAW file import	This module imports raw data into the project.	Spectral data	Raw data import	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/import_rawdata/ThermoRawImport.java
Van Krevelen Diagram	Van Krevelen Diagram.	Visualization	Visualization feature list	Annexhc; Olena Mokshyna; Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/vankrevelendata/VanKrevelenDiagram.java
Venn export module	Exports detected/undetected features in a feature list so they can be plotted as a venn diagramm.	Feature IO	Feature list export	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/export_features/VennExportModule.java
Waters RAW file import	This module imports raw data into the project.	Spectral data	Raw data import	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/import_rawdata/WatersRawImport.java
ZIP/gZIP file import	This module imports raw data into the project.	Spectral data	Raw data import	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/import_rawdata/ZipImport.java
iMZML file import	This module imports raw data into the project.	Spectral data	Raw data import	Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/import_rawdata/ImzMLImport.java
m/z mobility region extraction	Extracts regions of m/z-mobility regions	Feature filtering	Feature list filtering	Olena Mokshyna; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/filter_mobility/MobilityMzRegionExtraction.java
metaCorrelate	This method correlates and groups feature list rows based on multiple criteria (feature shape, intensity across samples, ...)	Feature processing	Feature grouping	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal	src/main/java/io/dataprocessing/corrgrouping/Correlation.java

name	description	main class	sub class	contributors	path
module	describe	Feature filtering	Feature list filtering	Robin Schmid; Roman Bushuiev; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/EmptyFeatureL
mzData file import	This module imports raw data into the project.	Spectral data	Raw data import	Steffen Heuckeroth; Tomas Pluskal	src/main/java/import_rawdata/MzDataImport
mzML file import via MSdk	This module imports raw data into the project.	Spectral data	Raw data import	Kirill Ponomarov; Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/import_rawdata/MSDKmzMLIm
mzML file import via jmzml	This module imports raw data into the project.	Spectral data	Raw data import	Steffen Heuckeroth; Tomas Pluskal	src/main/java/deprecated_jmzml
mzXML file import	This module imports raw data into the project.	Spectral data	Raw data import	Robin Schmid; Steffen Heuckeroth; Tomas Pluskal; Robin Schmid	src/main/java/import_rawdata/MzXMLImport
netCDF file import	This module imports raw data into the project.	Spectral data	Raw data import	David Mařek; Steffen Heuckeroth; Tomas Pluskal	src/main/java/import_rawdata/NetCDFImport
scan histogram	This module plots all values of all selected scans into one histogram and offers a Gaussian fit.	Visualization	Visualization	Olena Mokshyna; Tomas Pluskal; Robin Schmid	src/main/java/scan_histogram

⌚ July 30, 2025 10:41:01

👤 Robin Schmid, Steffen Heuckeroth

16.2 How to contribute to the MZmine documentation

16.2.1 About the documentation

Supporting and updating the documentation is especially important for MZmine due to the variety and complexity of the modules. If you wish to contribute by creating a module, please, create a corresponding documentation. In this way, the MZmine community will be able to easier adapt the novel modules in their workflows and everyday practice.

MZmine documentation uses [MkDocs site generator](#). Documentation source is created in Markdown and configured using a single YAML file. If you are new to MkDocs or Markdown, the detailed User guide can be found [here](#).

16.2.2 Pages structure

Structure of the pages is described in `mkdocs.yml` file, which represents the tree-like structure:

```
nav:
  - Home page: index.md
  - Getting started: getting_started.md
  - Main window overview: main-window-overview.md
  - Data conversion: data_conversion.md
  - Citations: citations.md

  - Coding:
    - General structure: coding/general_structure.md
    - Module development: coding/module_development.md
    - Raw data structure: coding/raw_data_structure.md
    - Feature data structure: coding/feature_data_structure.md
    - Example module - Spectral entropy: coding/example_module.md
```

When a new documentation page or subsection is added, yaml structure changes correspondingly. For example:

```
nav:
  - Home page: index.md
  - Getting started: getting_started.md
  - Main window overview: main-window-overview.md
  - Data conversion: data_conversion.md
  - Citations: citations.md

  - Coding:
    - General structure: coding/general_structure.md
    - Module development: coding/module_development.md
    - Raw data structure: coding/raw_data_structure.md
    - Feature data structure: coding/feature_data_structure.md
    - Example module - Spectral entropy: coding/example_module.md
    - NEW SECTION:
      - NEW PAGE: new_section/new_page.md
```

16.2.3 How to edit the documentation pages

1. Make a GitHub Account

You'll need to make a [GitHub Account](#).

2. Click Edit Button on Page You Want to Edit

MZmine 3 Documentation

[Home page](#)
[Main window overview](#)
[LC-MS workflow](#)
[LC-IMS-MS workflow](#)
[Raw data visualisation](#)

LC-IMS-MS Workflow

Supported formats

- Vendor formats: *

 - .tdf (Native Bruker LC-IMS-MS and MALDI-IMS-MSI format) *
 - .tsf (Native Bruker MALDI-IMS-MS (single shot) format)

- .mzML *

 - Created via [MSConvert](#) from native Bruker data *
 - Created via [MSConvert](#) from native Waters data

	Table
	Supplementary
	Edit this page
	Background information
	Motif form
	Français
	Motif form
	Ion intensity
	Raw data
	Raw data
	Mass spectra
	See also

3. Fork the Repository When Prompted (only the first time)



You need to fork this repository to propose changes.

Sorry, you're not able to edit this repository directly—you need to fork it and propose your changes from there instead.

 [Fork this repository](#)

[Learn more about forks](#)

4. Make the Edits in MarkDown

mzmine_documentation / docs / Ion-mobility-data-proc

Cancel changes

<> Edit file Preview Spaces 3 Soft wrap

```

1 # LC-IMS-MS Workflow
2 ## Supported formats
3
4 * Vendor formats:
5 *
6     * .tdf (Native Bruker LC-IMS-MS and MALDI-IMS-MSI format)
7 *
8     * .tsf (Native Bruker MALDI-IMS-MS (single shot) format)
9 * .mzML
10 *
11     * Created via [MSConvert](https://proteowizard.sourceforge.io/download.html) from native Bruker
12     data
13 *
14     * Created via [MSConvert](https://proteowizard.sourceforge.io/download.html) from native Waters
15     data
16
17 **Note**: mzML via MSConvert from Agilent raw data can be imported, but certain restrictions might
18 hinder processing workflows due to the nature of the raw data format.
19
20
21 ***
22
23 ## Background information and terminology
24
25 Since ion mobility spectrometry (IMS) resolved data is more complex due to the additional dimension
26 when compared to regular LC-MS data, some terms shall be clarified before going into details of the
27 processing steps.
28
29 ### Mobility separation and data format
30
31 Ion mobility separation usually occurs on the millisecond timescale, fitting nicely in-between
32 liquid chromatography (LC) (few seconds per chromatographic peak) and mass spectra acquisition of
33 TOF instruments (several micro seconds). Therefore, the mobility dimension can be resolved by
34 acquiring multiple spectra during a mobility separation (e.g. 1000 spectra per 100 ms). This leads
35 to multiple mass spectra acquired at one IMS accumulation. Thus, at one retention time, multiple
36 spectra are acquired. A detailed comparison of LC-MS and LC-IMS-MS raw data can be

```

Attach files by dragging & dropping, selecting or pasting them.

5. Propose Changes

Please describe the change you are making.

Commit changes

update mobility resolving step

add msms pairing description in mobility resolving step

steffen.heuckeroth@gmx.de

Choose which email address to associate with this commit

Commit directly to the `master` branch.

Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

 SteffenHeu-patch-1

Propose changes

Cancel

6. Create Pull Request

The screenshot shows the GitHub repository interface for 'mzmine / mzmine_documentation'. The 'Pull requests' tab is selected. A yellow banner at the top indicates 'SteffenHeu-patch-1 had recent pushes 1 minute ago'. A green button labeled 'Compare & pull request' is visible.

7. Finalize Pull Request with Description

The screenshot shows the pull request description editor. The base branch is set to 'master'. The compare branch is 'SteffenHeu-patch-1'. A green checkmark indicates 'Able to merge'. The description text area contains 'update mobility resolving step' and 'add msms pairing description in mobility resolving step'. A note says 'Attach files by dragging & dropping, selecting or pasting them.' A green 'Create pull request' button is at the bottom right.

i Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

16.2.4 How to create a new page

Follow steps 1 - 3.

Navigate to `mzmine_documentation/docs` in your fork and create a new file

The screenshot shows a GitHub repository interface for 'mzmine_documentation / docs /'. The top navigation bar includes 'master', 'mzmine_documentation / docs /', 'Go to file', 'Add file', and a three-dot menu. A blue button labeled 'Create new file' is highlighted. The main area displays a list of files with their descriptions and last modified times:

File	Description	Last Modified
try png logo	try png logo	20 hours ago
module_docs	add all documents to nav	yesterday
workflows	Update Ion-mobility-data-processing-workflow.md	22 hours ago
Contribute.md	Add credit	3 months ago
Main-window-overview.md	typos, fix links	2 days ago
Raw-data-visualisation.md	typos, fix links	2 days ago
index.md	fix remaining dead links	yesterday
performance.md	fixed layout	last month
wikiacknowledgements.md	add page contributors, add gnps acknowledgements	3 months ago

Follow steps 4 - 7.

16.2.5 Additional useful information

Using icons

To highlight some points in the prepared documentation, it is advised to use icons. **Material for MkDocs** is integrated into MZmine documentation and suggests a wide range of icons, logos, and more. Icons can be found [here](#).

Using mathematical formulas

To add mathematical formulas, it is recommended to use LaTex notation, which can be directly incorporated into MkDocs pages.

To insert inline formula, one can use single "\$" signs, like this `$x^n + y^n = z^n$`, resulting in $(x^n + y^n = z^n)$. To insert formula on a new line, use double "\$" signs, for example `$$E=mc^2$$`, resulting in

$E=mc^2$

The excellent reference guide to different mathematical expressions can be found [here](#).

16.2.6 Page Contributors

[Steffen Heuckeroth, omokshyna](#)

This page was adapted from the [GNPS documentation](#).

⌚ November 3, 2022 11:38:29

👤 Steffen Heuckeroth, omokshyna

16.3 Markdown template to create online documentation for a processing module

Download immediately usable template from [here](#)

To facilitate a consistent contribution to the MZmine online documentation from new users, please refer to the present Markdown template. Changes in the structure are of course allowed, but will have to be justified in the pull request. Remove this top part from the new page and start at **Module name**.

In order not to interrupt the document flow, we encourage the use of admonition boxes insight any point of the document to provide warnings, tips, and side contents in general. A few examples are provided below, while a variety of other options is available [here](#).



This module is currently under maintenance / deprecated / etc.



Since mass lists are taken as input by EIC building algorithms, the Mass detection module must be run first



A good starting point for 'Parameter #1' is between 0.1 and 0.5.

REMOVE ALL TEXT ABOVE THIS LINE TO START YOUR OWN MODULE. CHANGE THE RECOMMENDED CITATIONS.

16.4 Module name

≡ Menu1 → Menu2 → Module

Provide here a concise description of the module functioning principles and usage. If possible, briefly describe the algorithm(s) employed by the module. If a detailed description of the algorithm functioning is needed, provide it in the [Algorithm](#) section). Please cite relevant references in this section using the following box:

16.4.1 Recommended citations



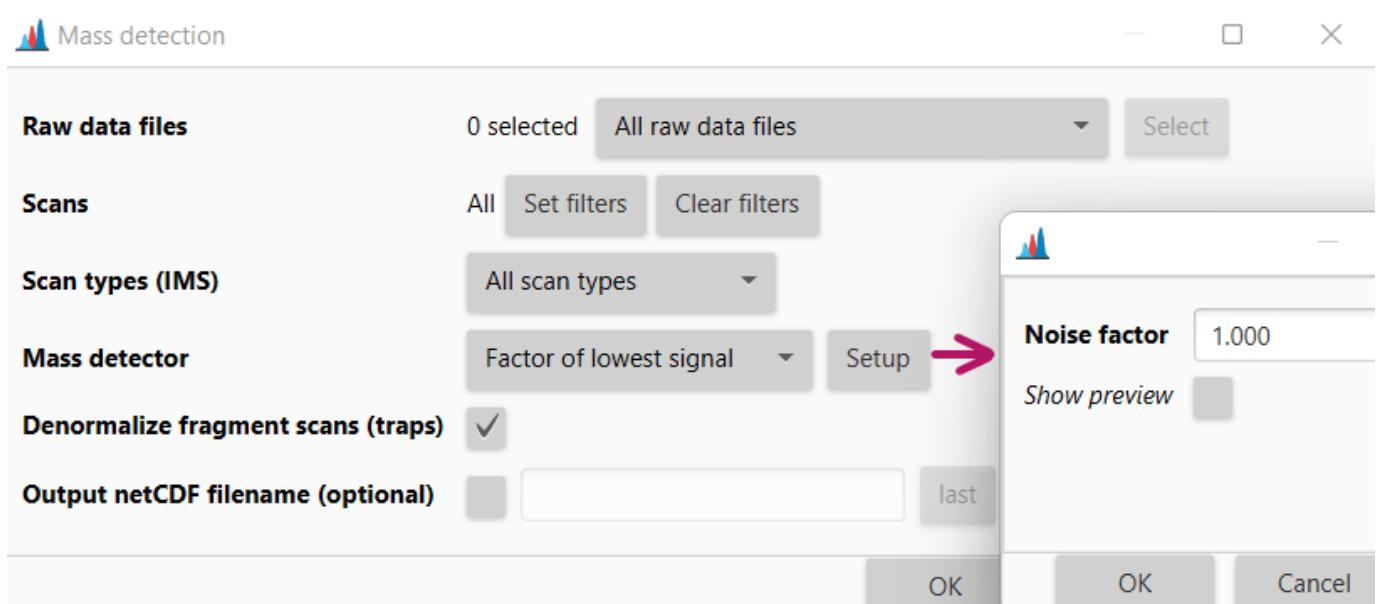
When using mzmine for your work, please consider citing:

Schmid R., Heuckeroth S., Korf A., et al. Integrative analysis of multimodal mass spectrometry data in MZmine 3, *Nature Biotechnology* (2023), doi:10.1038/s41587-023-01690-2.

When using this modules, please consider citing the corresponding publication(s):

Schmid, R., Petras, D., Nothias, LF. et al. Ion identity molecular networking for mass spectrometry-based metabolomics in the GNPS environment. *Nat Commun* 12, 3832 (2021). <https://doi.org/10.1038/s41467-021-23953-9>

16.4.2 Parameters



Screenshot of the module dialog should contain default values. If multiple sub dialogs are available from parameters, either create one or multiple dialog images.

Provide here a concise description of each processing parameter. Where possible, provide default values and/or recommended values to start the tuning from.

PARAMETER #1

Provide here a brief description for Parameter #1. Default values, tuning points, hints...

PARAMETER #2

Provide here a brief description for Parameter #2.

PARAMETER #3 (OPTIONAL)

Provide here a brief description for Parameter #3. Optional parameters should be labelled as *(Optional)* following the parameter's name.

16.4.3 Algorithm

Optional: A more detailed description of the algorithm(s) employed by the module can be provided in this section.

Ansgar Korf, Robin Schmid, omokshyna, tdamiani

⌚ October 21, 2025 15:45:16

✍ Ansgar Korf, Robin Schmid, omokshyna, tdamiani

17. Acknowledgements

We would like to point out that this wiki was set up in tight collaboration with the [GNPS](#) staff. We highly appreciate your help!

17.1 Related projects

- GNPS
- SIRIUS

17.2 Libraries we use in MZmine

- [Apache XML Graphics](#) - EPS image export
- [Chemistry Development Kit](#) - Isotope pattern and molecular calculations
- [Freehep](#) - EMF image export
- [Google Guava](#) - Utility classes
- [JDK Documentation](#)
- [JChemPaint](#) - 2D molecule visualization
- [JFreeChart](#) - TIC, Spectra and 2D visualizers
- [J Mol](#) - 3D molecule visualization
- [jmzml](#) - mzML file import
- [jmzTab](#) - mzTab file import and export
- [NetCDF-Java](#) - NetCDF file import
- [VisAD](#) - 3D visualizer
- [WEKA](#) - Clustering and other machine learning algorithms
- [Bruker TDF SDK](#) - Native tdf/tdf file import (requires VC++ 2017 redist.)
- [Thermo raw file parser](#) - Native Thermo raw import

[Robin Schmid](#), [Steffen Heuckeroth](#), [omokshyna](#)

 June 2, 2023 12:19:27

 [Robin Schmid](#), [Steffen Heuckeroth](#), [omokshyna](#)

18. Troubleshooting

18.1 GUI freezes

The graphical user interface freezes sometimes on remote desktops if the user logs out and logs in again. This is a known issue and might be fixed in future JavaFX versions.

18.2 mzmine does not start

Maybe the MZmine configuration got corrupted (broke). Delete it and reinstall MZmine. The * *.mzmine.conf** file can be found in the user folder and might be hidden. On Windows, activate " Show hidden items" in the Explorer/View menu.

18.3 mzML files not importing

An illegal character might exist in the mzML file. When converting the raw data files to mzML, the path of the file is added to the resulting mzML file. The path may contain characters that are not parsed correctly by MZmine.

Solution: Reconvert the file and place the input and output directory to a path with only Roman characters.

18.4 Thermo raw files not importing

Error message usually contains "No scans found".

Solution: Search for the "mzmine_thermo_raw_parser" folder in the mzmine temp folder and the system temp folder and delete it.

18.5 Scans not sorted in retention time dimension

This issue often occurs with raw data from Waters instruments. We recommend re-converting the files using the [Waters data connect tool](#).

18.6 Scans were recognised as empty

This is displayed as a warning and means that some scans did not contain any data. This is **not** considered as an error. The warning **may** indicate that a threshold was too high, if conversion was applied. You may consider lowering said threshold. If no thresholding was applied, consider this a false positive. When importing a vendor data format into mzmine, this warming is very likely to be a false positive.

18.7 Zero intensity values in scans

0 intensities in scans.

This warning may occur when converting files using an old MSConvert version and the "Peak picking" filter being used after the "title maker". This caused the wrong peak picking algorithm to be used and not just adds zero values but also causes falsely centroided m/z values. In case you converted the data manually without mzmine, we recommend re-converting. In case the issue still appears, keep in mind that this is just a warning.

18.8 Reporting issues

Report issues on the [MZmine GitHub](#), providing all information to reproduce the issue.

This might include

- **Input data?** provide example file
- **** Batch file**** defining processing steps and settings
- attach the log file located in the user folder: **mzmine_0_0.log**
- **Expected behavior**
- **Issue**

 July 28, 2025 13:48:29 Robin Schmid, Steffen Heuckeroth

19. Changelog

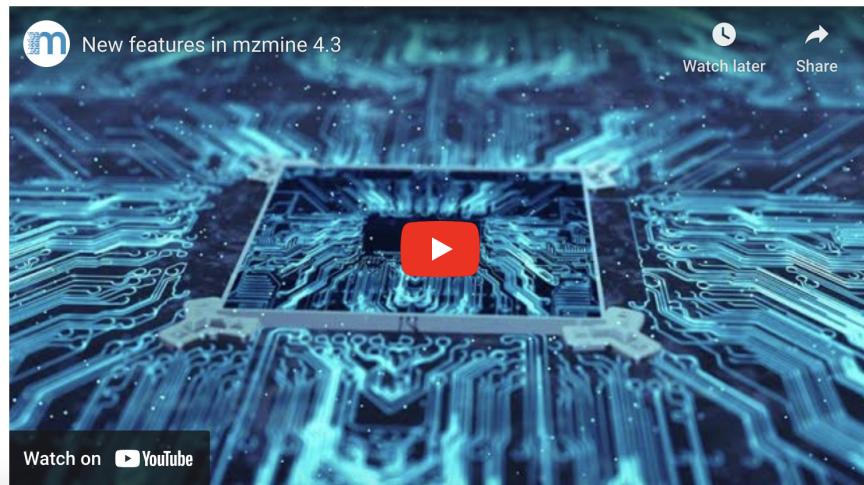
19.1 Latest mzmine changes

Find a detailed [changelog](#) on our GitHub release page.

Check out latest [mzmine news](#).

WHAT'S NEW IN

mzmine



Version 4.3 is out now!

- Improved memory management for better resource usage
- Interactive molecular networking with MS2Deepscore
- Fragmentation dashboard: A new tool for molecular formula prediction
- Improved vendor data support – Seamlessly import additional vendor data formats if MSConvert is installed

⌚ October 15, 2024 08:28:07

👤 Ansgar Korf, Steffen Heuckeroth

20. Workshops

Venue	Description	Dates	Speakers/ Instructors	Material	Estimated Attendance
Webinar, hosted by VMOL	mzmine Series: Automatic library generation: LC-MS2, MSn trees	2024-12-17	Corinna Brungs	Video	
Webinar, hosted by GA	Natural products research in the digital era: mzmine	2024-12-06	Corinna Brungs, Robin Schmid, Jean- Luc Wolfender		280
Webinar, hosted by VMOL	mzmine Series: LC-MS and LC-IMS- MS: PFAS, contaminants, and pesticides	2024-12-03	Steffen Heuckeroth	Video	250+
Webinar, hosted by VMOL	mzmine Series: Multimodal data processing using mzwizard: LC- MS2, GC/EI- MS, MALDI- Imaging, and lipid annotation	2024-11-19	Ansgar Korf	Video	250+
Webinar, hosted by VMOL	mzmine Series: Non- target LC-MS2 Feature Detection, Statistical Analysis, Molecular Networking, Compound Annotation	2024-11-05	Robin Schmid	Video	350+
Seoul, South Korea	Non-Target CompMS mzmine Workshop	2024-07-09	Corinna Brungs, Robin Schmid		250
OIST, Okinawa, Japan	CompMS Workshop joint with mzmine, SIRIUS, and AI for metabolomics	2024-06-24	Sebastian Böcker, Corinna Brungs, Robin Schmid, Tomas Pluskal, Roman		40

Venue	Description	Dates	Speakers/ Instructors	Material	Estimated Attendance
			Bushuiev, Louis-Felix Nothias, Tito Damiani		
Osaka, Japan	Metabolomics conference: mzmine CompMS workshop	2024-06-16	Corinna Brungs, Robin Schmid, Tomas Pluskal, Roman Bushuiev, Louis-Felix Nothias, Tito Damiani		90+
IOCB Prague, Czechia	Prague Computational Mass Spectrometry Workshop: 20 years mzmine	2024-04-15 to 17	Corinna Brungs, Robin Schmid, Steffen Heuckeroth, Ansgar Korf, Tomas Pluskal, Sebastian Böcker, Louis-Felix Nothias, Adriano Rutz, Niek de Jonge, Matej Orešić, Yasin El Abiead, Marcus Ludwig		100 +online
University of Vienna, Austria	Vienna MassSpec-Forum	2024-02-21	Corinna Brungs, Robin Schmid		50
Helmholtz, Munich, Germany	Munich Metabolomics Meeting: MZmine workshop	2023-10-12	Robin Schmid, Corinna Brungs		150+
SSI, Copenhagen, Denmark	International Summer School on Non-Targeted Metabolomics 2023	2023-08	Steffen Heuckeroth, Corinna Brungs, Robin Schmid, Daniel Petras, and many more	Playlist	75 (+150 online)
DTU, Copenhagen, Denmark	Computational mass spectrometry and spectral library generation in MZmine	2023-08	Corinna Brungs, Robin Schmid		50

Venue	Description	Dates	Speakers/ Instructors	Material	Estimated Attendance
EMBL, Heidelberg, Germany	Computational mass spectrometry and spectral library generation in MZmine	2023-07-24	Robin Schmid, Corinna Brungs		50
Niagara Falls, Canada	Metabolomics Conference: IMS Workshop (with MZmine 3)	2023-06-18	Tomas Pluskal		
Uni Geneva, Switzerland	Computational mass spectrometry and spectral library generation in MZmine	2023-04-24	Robin Schmid, Corinna Brungs		50
ETH, Zürich, Switzerland	Non-target MS analysis and spectral library generation in MZmine	2023-04-19	Robin Schmid, Corinna Brungs		50
Wageningen, The Netherlands	NPLinker eScience Workshop on Genome and Metabolome Mining	2023-03-21 to 24	Robin Schmid, Mitja Zdouc, Lindsay Caesar and more	Soon	30+
Wageningen, The Netherlands	Computational mass spectrometry in MZmine	2023-03-17	Robin Schmid, Corinna Brungs		40
Tuebingen, Germany	International Summer School on Non-Targeted Metabolomics 2022	2022-08-22 to 26	Organizers of the Summer School	Video	75+online
Valencia, Spain	Metabolomics conference: MZmine 3 workshop	2022-06-19	Steffen Heuckeroth, Robin Schmid, Tomas Pluskal, Tito Damiani, Daniel Petras		100+
CMFI Mass Spec Seminar	Zoom Video Series	2022-02	Daniel Petras organized series	Video	20-100+

Venue	Description	Dates	Speakers/ Instructors	Material	Estimated Attendance
IOCB, Prague, Chechia	Computational metabolomics workshop on MZmine, GNPS, and SIRIUS	2021-07-20	Steffen Heuckeroth, Robin Schmid, Tomas Pluskal, Ansgar Korf, Daniel Petras, Marcus Ludwig	Twitter	70
Zoom!	MZmine/GNPS Feature Based Molecular Networking in Portuguese - Part 1	2020-12-09	Anelize Bauermeister, Carlismari Grundmann, Helena Mannochio, Juliano Amaral, Tiago Leão	Material	200
Zoom!	MZmine/GNPS Feature Based Molecular Networking in Portuguese - Part 2	2020-12-10	Anelize Bauermeister, Carlismari Grundmann, Helena Mannochio, Juliano Amaral, Tiago Leão	Material	200
Zoom!	MZmine/GNPS Feature Based Molecular Networking in Espanol - Part 1	2020-07-08	Mauricio Caraballo, Daniel Petras	Material Video	40
Zoom!	MZmine/GNPS Feature Based Molecular Networking in Espanol - Part 2	2020-07-09	Mauricio Caraballo, Daniel Petras	Material Video	50
Zoom!	MZmine/GNPS Ion-Identity Molecular Networking, Native Metabolomics	2020-05-20	Robin Schmid, Daniel Petras, Allegra Aron	Material Video	65
Zoom!	MZmine/GNPS Feature Based Molecular Networking (Job Submissions and Exploration)	2020-05-14	Louis-Felix Nothias, Daniel Petras, Allegra Aron	Material	103

Venue	Description	Dates	Speakers/ Instructors	Material	Estimated Attendance
Zoom!	MZmine/GNPS Feature Based Molecular Networking	2020-05-13	Daniel Petras, Allegra Aron	Material Video	147
MIT	Molecular Networking with MZmine and GNPS	20219-12-05	Tomas Pluska, Daniel Petras, Robin Schmid	Twitter	30
Prague	MZmine/GNPS Workshop (4h)	2019-06-28	Mingxun Wang, Louis- Felix Nothias, Robin Schmid, Justin van der Hooft	TBA	20

⌚ December 6, 2024 16:32:20

👤 Robin Schmid, Steffen Heuckeroth