Bioinformatics Advances, 2026, **6**(1), vbag138
https://doi.org/10.1093/bioadv/vbag138
Published: 15 May 2026
Application Note | Software

OXFORD

eMZed 3: flexible and interactive development of scalable LC-MS/MS data analysis workflows in python

Uwe Schmitt[1,‡], Jethro Hemmann[2,‡], Nicola Zamboni[3], Julia A. Vorholt[4], Patrick Kiefer[4,‡,*]

[1]Scientific IT Services, ETH Zurich, Binzmuehlestrasse 130, Zurich 8092, Switzerland
[2]Junior Research Group Metabolomics-Guided Natural Product Discovery, Leibniz Institute for Natural Product Research and Infection Biology (Leibniz-HKI), Beutenbergstraße 11a, Jena 07745, Germany
[3]Institute of Molecular Systems Biology, ETH Zurich, Otto-Stern-Weg 3, Zurich 8093, Switzerland
[4]Institute of Microbiology, ETH Zurich, Zurich, Vladimir-Prelog-Weg 4, Zurich 8093, Switzerland
[‡]These authors contributed equally to this work.
*Corresponding author. Institute of Microbiology, Department of Biology, ETH Zurich, Zurich 8093, Switzerland. E-mail: pkiefer@ethz.ch
Associate Editor: Guoqiang Yu

Abstract

Summary: **Liquid chromatography–mass spectrometry (LC-MS/MS) data analysis requires adaptable software solutions to meet diverse analytical needs. We present eMZed 3, a modern Python framework for flexible and interactive analysis of LC-MS/MS data. eMZed 3 enables users to develop scalable workflows tailored to their specific requirements while leveraging Python's extensive ecosystem of libraries. Building on its predecessor, eMZed 3 is now Python 3-based and includes substantial enhancements, including support for chromatogram-based LC-MS data, a new SQLite-based backend supporting optional out-of-memory processing, and rich interactive visualization tools. Compared to the previous version, eMZed 3 is now split into three packages: *emzed* (core functionalities), *emzed-gui* (interactive data visualization), and *emzed-spyder* (an integrated development environment). This modular architecture allows straightforward integration of the *emzed* core library into headless Python environments, including computational notebooks (such as Jupyter) or high-performance computing clusters. eMZed 3 incorporates well-established libraries such as OpenMS, and is suited for both targeted and untargeted metabolomics. Overall, eMZed 3 supports the efficient development of scalable and reproducible LC-MS data analysis and is accessible to both novice and advanced programmers.**

Availability and implementation: **eMZed 3 and its documentation are freely available at https://emzed.ethz.ch, the source code is hosted at https://gitlab.com/groups/emzed3.**
An online-executable example workflow is available on Binder at: https://mybinder.org/v2/gl/emzed3%2Femzed-example-workflow/HEAD?_labpath=example.ipynb.

1 Introduction

Liquid chromatography–mass spectrometry (LC-MS) is a core analytical technique in many fields of biology, environmental sciences, and chemistry. Advances in LC-MS instrumentation have enabled the generation of increasingly large and complex data sets, leading to computational challenges. Consequently, reliable and reproducible analysis of LC-MS data has become a critical step in the analytical workflow and is supported by a variety of commercial and open-source software tools. These tools often belong to one of two extremes: user-friendly software with limited flexibility, and libraries that require advanced programming skills but offer greater versatility. For example, MZmine 3 (Schmid *et al.* 2023) is a widely-used GUI-based application allowing the simple setup of data analysis workflows consisting of pre-defined processing steps. However, it offers limited flexibility and lacks easy programmatic customization. In contrast, the OpenMS project (Pfeuffer *et al.* 2024) provides a robust and performant C++ library with Python bindings that allows high flexibility in workflow programming, but lacks advanced interactive visualization capabilities. To bridge this gap and support the interactive prototyping and development of tailored workflows, we had introduced eMZed as a novel open-source Python framework (Kiefer *et al.* 2013). Since then, eMZed has been used within a number of research projects (Hartl *et al.* 2020, Martano *et al.* 2020, Nguyen *et al.* 2020, Reiter *et al.* 2020, Hemmann *et al.* 2021, Maier *et al.* 2021, Butin *et al.* 2022, Keller *et al.* 2022, Ryback *et al.* 2022, Reiter *et al.* 2024, Mori *et al.* 2025) and continued to be further developed.

Here, we present a completely revised and modernized version 3 of eMZed. eMZed 3 has been rewritten for compatibility

Received: 14 January 2026. **Revised:** 6 April 2026. **Accepted:** 7 May 2026

with Python 3 (Van Rossum and Drake 2009) and features numerous new functions and performance enhancements, including support for chromatogram-based LC-MS/MS data (e.g. multiple reaction monitoring data), out-of-memory processing, and improved interactive data inspection and visualization. The framework retains its core strength of offering a comprehensive set of elementary functions and data types for reading, processing, and analyzing LC-MS data. With an intuitive Python API, eMZed 3 allows users to create custom workflows that seamlessly integrate with popular Python libraries, such as pandas (McKinney 2010), NumPy (Harris *et al.* 2020), or scikit-learn (Pedregosa *et al.* 2011). It also enables interactive visualization of chromatograms, spectra, and peakmaps at any stage of a workflow. The optional Spyder-based integrated development environment (IDE) bundles the eMZed modules, helping programming beginners to quickly develop their own LC-MS data analysis workflows. eMZed supports all aspects of reproducible and tailored data analysis for both targeted and untargeted metabolomics, as well as other applications.

2 Results

2.1 Functionalities

eMZed was designed as an easy-to-use coding framework that allows users to develop highly customizable workflows for LC-MS data analysis in Python (Fig. 1). To that end, eMZed builds upon existing algorithms from e.g. PyOpenMS (Röst *et al.* 2014) while incorporating additional features, including a versatile table data structure to store and access LC-MS peak lists and GUI-tools for the interactive inspection and re-integration of peaks. While eMZed includes many features found in widely used tools such as MZmine (Schmid *et al.* 2023) and XCMS (Louail *et al.* 2025) strength lies in enabling users to program custom workflows beyond the scope of existing tools and to address specific

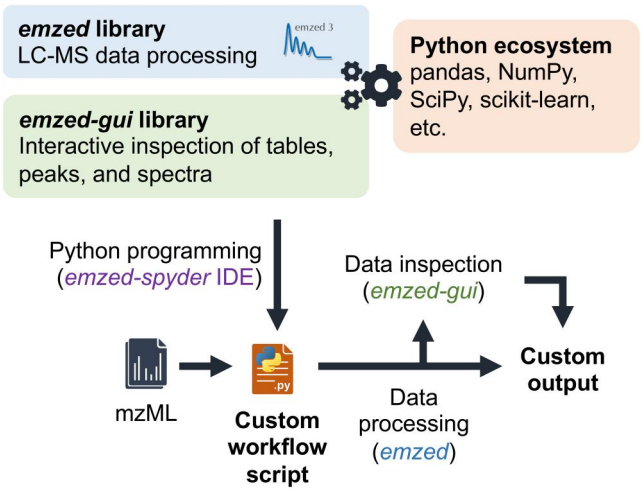

Figure 1 The core concept of eMZed. eMZed is designed as a modular framework (*emzed*, *emzed-gui*, *emzed-spyder*) to enable the Python-based development of highly customizable processing workflows for LC-MS/MS data. It incorporates key functions to access and process peaks, spectra, and chromatograms and connects with commonly used libraries from the Python ecosystem.

problems requiring tailored solutions (see also example workflow below).

The new version 3 of eMZed builds on the architecture and core features of the previous version of eMZed (Kiefer *et al.* 2013), but is now compatible with Python 3 and fully separated into three individual packages (all available at PyPI): *emzed*, *emzed-gui*, and *emzed-spyder*. The package *emzed* provides core functionalities of eMZed, while the optional *emzed-gui* allows interactive data inspection and visualization (see Fig. 2A for example code). To facilitate the interactive development of workflows with eMZed, *emzed-spyder* bundles a customized version of the Spyder IDE with the eMZed packages. For Windows, an installer is available at the eMZed website that allows an easy installation of eMZed 3 (*emzed-spyder* with all other eMZed packages).

The Python package *emzed* offers the core functionalities of eMZed 3, and does not depend on any GUI components. Thus, *emzed* can be used with any type of Python installation, including high-performance computing clusters or Jupyter notebooks (Kluyver *et al.* 2016). Among many features, *emzed* provides an easy-to-use API for:

i) Direct access to chromatograms, peaks, and spectra (MS^n) and their underlying data
ii) Targeted peak extraction
iii) Untargeted feature finding, isotope, and adduct grouping
iv) Alignment of peaks (m/z and retention time) and feature lists
v) Powerful table-based operations on peak lists
vi) Handling of chemical data such as molecular formulas
vii) Accessing R functionalities using integrated bridge to R

Several integration algorithms are provided by *emzed* and allow fast, parallel integration of chromatographic peaks. For untargeted analysis, eMZed integrates the feature finding algorithms FeatureFinderMetabo from OpenMS (Röst *et al.* 2016) and centWave (Tautenhahn *et al.* 2008), which is accessible from *emzed* via a builtin bridge to R (R Core Team 2021). Additionally, the ADAP feature finder from MZmine 2 (Myers *et al.* 2017) is available as an eMZed extension. eMZed 3 now also supports chromatogram-based LC-MS data, e.g. MRM data from triple quadrupole instruments. The powerful *Table* data structure plays a key role in eMZed, allowing operations such as filtering, grouping, joining, and aggregating and it now uses an SQLite database (Gaffney *et al.* 2022) as backend for improved performance and optional out-of-memory operations for processing large data sets.

For interactive visual inspection of *emzed* data structures, the *emzed-gui* package provides several tools that have been improved in eMZed 3. The *Peakmap Explorer* allows quick exploration of entire LC-MS/MS peakmaps using a 2D heatmap for visualization and can display chromatograms as well as mass spectra (Fig. 2B). The *Table Explorer* is a central element of eMZed and assists in inspecting and modifying data stored in *emzed* tables (Fig. 2C). A prominent use case for a *Table* is the management of peak lists, and in this case the *Table Explorer* provides interactive visualizations and modifications of extracted ion chromatograms and mass spectra (e.g. for manual re-integration of peaks). In eMZed 3, *Table Explorer* now offers improved speed and robustness and more features, e.g. the coloring of individual rows or cells. *emzed-gui* also comes with

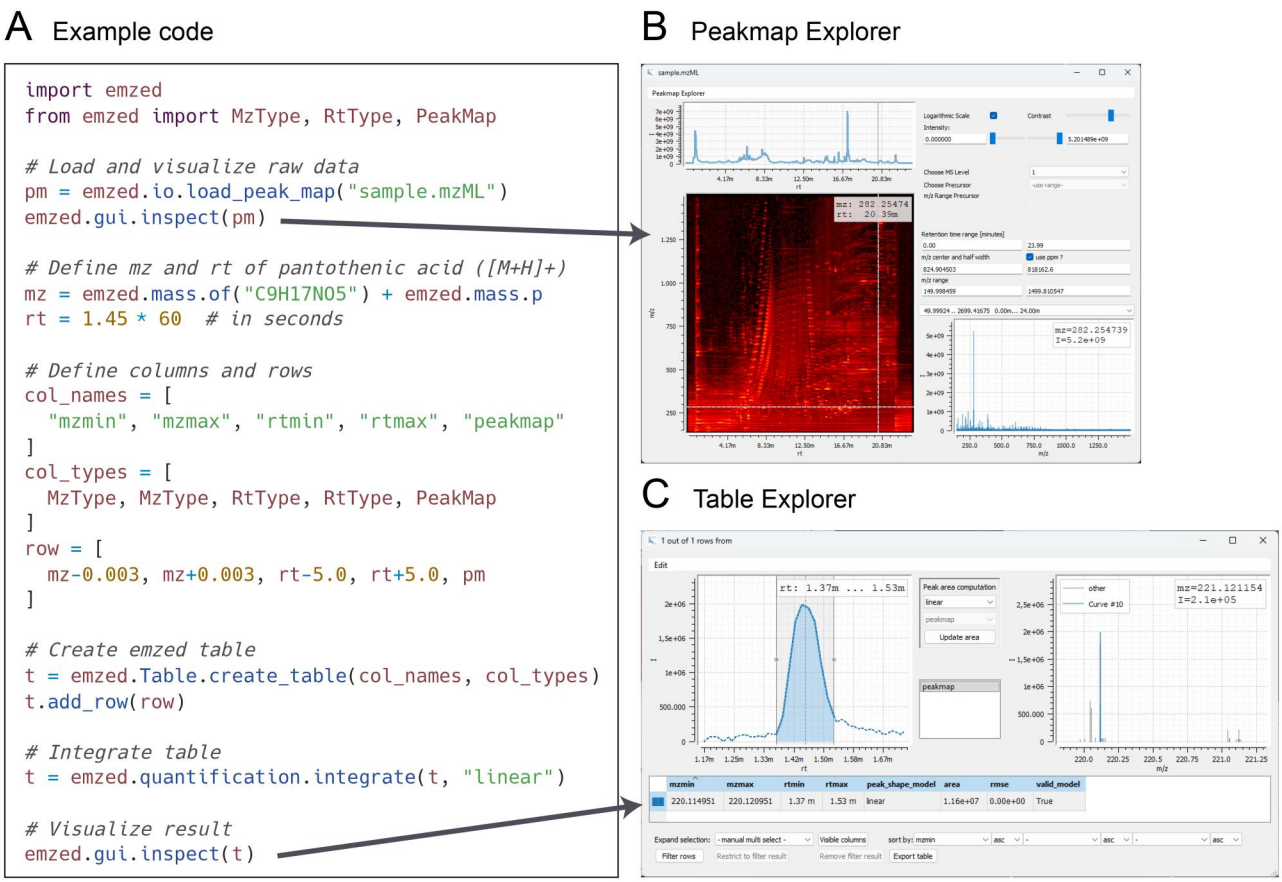

Figure 2 Visualization of LC-MS data using eMZed 3. *A*, Example code demonstrating a typical workflow: loading of LC-MS data, inspection of the peakmap, extraction and integration of LC-MS peaks corresponding to specific metabolites, and peak visualization. *B*, Screenshot of *Peakmap Explorer*, which allows the interactive inspection of LC-MS/MS peakmaps. *C*, Screenshot of *Table Explorer*, used to interactively inspect and (re-) integrate LC-MS peaks which are defined in an eMZed *Table* by their *m/z* and retention time values.

tools to create simple graphical input forms that can be used in a workflow for user input.

The optional *emzed-spyder* package is a customized IDE to facilitate interactive workflow development and prototyping. *emzed-spyder* is based on Spyder 5 and integrates all other eMZed packages. It supports direct inspection of *PeakMap* and *Table* data structures in the variable explorer and includes functionalities for the development of eMZed-based Python projects (also called *eMZed extensions*) based on separate virtual environments created for that purpose.

2.2 Implementation details

For the development of eMZed 3, the previous Python 2-based code (Kiefer *et al.* 2013) was revised and modernized to be compatible with Python 3 (tested with versions 3.11–3.14). While most of the architecture from the previous version of eMZed was kept, the new implementation includes several notable technical improvements. To store and process information within the eMZed data structures Table and PeakMap, an SQLite database is now used internally, which results in a major performance enhancement and significantly reduced memory footprint. The SQLite database is per default kept in memory but optionally allows out-of-memory operations on disk, thus circumventing working memory limitations that might occur for large data. To allow access to functionalities from OpenMS (e.g. I/O operations, feature finding, spectral processing, retention time alignment), PyOpenMS (version 3.3.0) is used, which is run in its own process to prevent version conflicts between shared libraries, such as the Qt framework. The ADAP peak picker from MZmine 2 (Myers *et al.* 2017) is offered as a separate extension module. The GUI tools of eMZed are now migrated to PyQt5. The customized IDE emzed-spyder is based on Spyder (version 5.5.6) and enhances Spyder to support eMZed-specific features and offers functions for the development of eMZed-based projects.

eMZed 3 runs on all modern operating systems (tested on Windows, Ubuntu, macOS), and the core library emzed is compatible with (cloud-based) notebook environments (e.g. Jupyter, Google Colab).

2.3 Example workflow

To assist new users, we provide a comprehensive tutorial for beginners on the eMZed website (https://emzed.ethz.ch). In addition, we created a more advanced example workflow to demonstrate the capabilities of eMZed 3. The example workflow can

be executed as a cloud-based notebook on Binder[1] and highlights how eMZed can be used to correct for retention time (RT) shifts using a problem-specific approach. In this example, we consider the analysis of amino acids in a previously published data set comprising 200 samples (Maier *et al.* 2021). Raw data visualization reveals RT instability within a smaller sample subset. However, simply enlarging the RT windows used for peak integration is not a suitable approach to cope with the observed RT shifts, since the RT difference of the two mass isomers leucine and isoleucine is smaller than the observed maximal inter-sample RT shift. The example workflow thus implements a simple, case-specific solution that fixes the issue by using RT shifts of other amino acids as estimates to shift the RT windows for leucine and isoleucine. In the example notebook, we further evaluate whether the strategy was successful and visualize treatment-specific differences of the amino acid profiles.

3 Discussion

eMZed seeks to fill a gap between other commonly used open-source tools for LC-MS data analysis, such as MZmine (Heuckeroth *et al.* 2024), OpenMS (Pfeuffer *et al.* 2024), MetaboAnalyst (Pang *et al.* 2024), or XCMS (Louail *et al.* 2025) by providing a powerful set of low-level functions to access, process, and interactively visualize LC-MS data that seamlessly integrates into the Python ecosystem of libraries for data analysis, machine learning, and plotting. By combining these functionalities, users can quickly develop tailored LC-MS data analysis workflows, while staying close to the raw data (chromatograms and spectra) with the help of the interactive GUI tools. We envision that eMZed 3 will be of particular benefit to users who require customized, yet robust and scalable data analysis pipelines, e.g. for analytical core facilities. We aim to continuously develop eMZed and provide additional eMZed-based tools and workflows in the future. As an open-source software, we particularly welcome community contributions (in the Gitlab repository).

Acknowledgements

We acknowledge Andrea Zamuner for testing eMZed 3 and providing input for improvements.

Supplementary material

Supplementary data are available at *Bioinformatics Advances* online.

Conflicts of interest

None declared.

Funding

This work was supported by ETH Zurich, Department of Biology, within the frame of an IT-strategy initiative; the Personalized Health and Related Technologies (PHRT) initiative of the ETH Domain; and the NCCR Microbiomes by the Swiss National Science Foundation [51NF40_180575, 51NF40_225148]. J.L.H. was supported by the Free State of Thuringia and the European Social Fund Plus [2024 FGR 0017].

[1] https://mybinder.org/v2/gl/emzed3%2Femzed-example-workflow/HEAD?labpath=example.ipynb

Data availability statement

No new data were generated or analysed in support of this research.

References

Butin N, Bergès C, Portais JC *et al.* An optimization method for untargeted MS-based isotopic tracing investigations of metabolism. *Metabolomics* 2022;**18**:41. https://doi.org/10.1007/s11306-022-01897-5.

Gaffney KP, Prammer M, Brasfield L *et al.* SQLite: past, present, and future. *Proc VLDB Endow* 2022;**15**:3535–47. https://doi.org/10.14778/3554821.3554842.

Harris CR, Millman KJ, van der Walt SJ *et al.* Array programming with NumPy. *Nature* 2020;**585**:357–62. https://doi.org/10.1038/s41586-020-2649-2.

Hartl J, Kiefer P, Kaczmarczyk A *et al.* Untargeted metabolomics links glutathione to bacterial cell cycle progression. *Nat Metab* 2020;**2**:153–66. https://doi.org/10.1038/s42255-019-0166-0.

Hemmann JL, Brühwiler MR, Bortfeld-Miller M *et al.* Structural diversity of the coenzyme methylofuran and identification of enzymes for the biosynthesis of its polyglutamate side chain. *J Biol Chem* 2021;**296**:100682. https://doi.org/10.1016/j.jbc.2021.100682.

Heuckeroth S, Damiani T, Smirnov A *et al.* Reproducible mass spectrometry data processing and compound annotation in MZmine 3. *Nat Protoc* 2024;**19**:2597–641. https://doi.org/10.1038/s41596-024-00996-y.

Keller P, Reiter MA, Kiefer P *et al.* Generation of an *Escherichia coli* strain growing on methanol via the ribulose monophosphate cycle. *Nat Commun* 2022;**13**:5243. https://doi.org/10.1038/s41467-022-32744-9.

Kiefer P, Schmitt U, Vorholt JA. eMZed: an open source framework in python for rapid and interactive development of LC/MS data analysis workflows. *Bioinformatics* 2013;**29**:963–4. https://doi.org/10.1093/bioinformatics/btt080.

Kluyver T, Ragan-Kelley B, Pérez F *et al.* Jupyter Notebooks—a publishing format for reproducible computational workflows. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Amsterdam: IOS Press, 2016, 87–90. https://doi.org/10.3233/978-1-61499-649-1-87.

Louail P, Brunius C, Garcia-Aloy M *et al.* Xcms in peak form: now anchoring a complete metabolomics data preprocessing and analysis software ecosystem. *Anal Chem* 2025;**97**:27639–45. https://doi.org/10.1021/acs.analchem.5c04338.

Maier BA, Kiefer P, Field CM *et al.* A general non-self response as part of plant immunity. *Nat Plants* 2021;**7**:696–705. https://doi.org/10.1038/s41477-021-00913-1.

Martano G, Leone M, D'Oro P *et al.* SMfinder: small molecules finder for metabolomics and lipidomics analysis. *Anal Chem* 2020;**92**: 8874–82. https://doi.org/10.1021/acs.analchem.0c00585.

McKinney W. Data structures for statistical computing in: *Python. Python in Science Conferences 2010*, 2010, 56–61. https://doi.org/10.25080/Majora-92bf1922-00a.

Mori MP, Lozoya OA, Brooks AM *et al.* Mitochondrial membrane hyperpolarization modulates nuclear DNA methylation and gene expression through phospholipid remodeling. *Nat Commun* 2025;**16**:4029. https://doi.org/10.1038/s41467-025-59427-5.

Myers OD, Sumner SJ, Li S *et al.* One step forward for reducing false positive and false negative compound identifications from mass spectrometry metabolomics data: new algorithms for constructing extracted ion chromatograms and detecting chromatographic peaks. *Anal Chem* 2017;**89**:8696–703. https://doi.org/10.1021/acs.analchem.7b00947.

Nguyen BD, V MC, Hartl J *et al.* Import of aspartate and malate by DcuABC drives H2/fumarate respiration to promote initial Salmonella Gut-Lumen colonization in mice. *Cell Host Microbe* 2020;**27**:922–36.e6. https://doi.org/10.1016/j.chom.2020.04.013.

Pang Z, Lu Y, Zhou G *et al.* MetaboAnalyst 6.0: towards a unified platform for metabolomics data processing, analysis and interpretation. *Nucleic Acids Res* 2024;**52**:W398–406. https://doi.org/10.1093/nar/gkae253.

Pedregosa F, Varoquaux G, Gramfort A *et al.* Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011;**12**:2825–30.

Pfeuffer J, Bielow C, Wein S *et al.* OpenMS 3 enables reproducible analysis of large-scale mass spectrometry data. *Nat Methods* 2024;**21**:365–7. https://doi.org/10.1038/s41592-024-02197-7.

R Core Team. *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, 2021. https://www.R-project.org/.

Reiter MA, Bradley T, Büchel LA *et al.* A synthetic methylotrophic *Escherichia coli* as a chassis for bioproduction from methanol. *Nat Catal* 2024;**7**:560–73. https://doi.org/10.1038/s41929-024-01137-0.

Reiter S, Cahn JKB, Wiebach V *et al.* Characterization of an orphan type III polyketide synthase conserved in uncultivated "Entotheonella" sponge symbionts. *Chembiochem* 2020;**21**: 564–71. https://doi.org/10.1002/cbic.201900352.

Röst HL, Sachsenberg T, Aiche S *et al.* OpenMS: a flexible open-source software platform for mass spectrometry data analysis. *Nat Methods* 2016;**13**:741–8. https://doi.org/10.1038/nmeth.3959.

Röst HL, Schmitt U, Aebersold R *et al.* pyOpenMS: a python-based interface to the OpenMS mass-spectrometry algorithm library. *Proteomics* 2014;**14**:74–7. https://doi.org/10.1002/pmic.201300246.

Ryback B, Bortfeld-Miller M, Vorholt JA. Metabolic adaptation to vitamin auxotrophy by leaf-associated bacteria. *ISME J* 2022; **16**:2712–24. https://doi.org/10.1038/s41396-022-01303-x.

Schmid R, Heuckeroth S, Korf A *et al.* Integrative analysis of multimodal mass spectrometry data in MZmine 3. *Nat Biotechnol* 2023;**41**:447–9. https://doi.org/10.1038/s41587-023-01690-2.

Tautenhahn R, Böttcher C, Neumann S. Highly sensitive feature detection for high resolution LC/MS. *BMC Bioinformatics* 2008; **9**:504. https://doi.org/10.1186/1471-2105-9-504.

Van Rossum G, Drake FL. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

Bioinformatics Advances, 2026, 6, 1–5
https://doi.org/10.1093/bioadv/vbag138
Application Note