PLOS Computational Biology

SOFTWARE

LCMSpector: A simple open-source viewer for targeted hyphenated mass spectrometry analysis

Mateusz Fido[1], **Etienne Hoesli**[2], **Elisa Cappio Barazzone**[3], **Renato Zenobi**[1], **Emma Slack**[3,4,5]*

1 Department of Chemistry and Applied Biosciences, Swiss Federal Institute of Technology, Zürich, Switzerland, **2** Department of Biology, Swiss Federal Institute of Technology, Zürich, Switzerland, **3** Department of Health Sciences and Technology, Swiss Federal Institute of Technology, Zürich, Switzerland, **4** Sir William Dunn School of Pathology, University of Oxford, Oxford, United Kingdom, **5** Basel Research Center for Child Health, Basel, Switzerland

* emma.slack@hest.ethz.ch

Abstract

The ubiquitousness of hyphenated mass spectrometry techniques across life sciences has made researchers around the world aware of their analytical power. However, the analysis of multidimensional high-complexity mass spectrometry data remains virtually impossible for non-specialists. LCMSpector is a standalone graphical user interface application for straightforward analysis of targeted mass spectrometry data, distributed under the MIT license and available free of charge for Windows, MacOS, and Linux. The user can upload and process the data entirely locally on their machine by specifying the desired mass-to-charge ratios of the targeted compound ions and simply clicking “Process”. It allows for the processing of multiple files simultaneously, freely modifying and exporting graphs in real time, and calculating compound concentrations based on calibration standards. We show the versatility and applicability of LCMSpector to different kinds of mass spectrometry data by analyzing a series of publicly available datasets from various samples recorded on instruments from multiple vendors and by different research teams.

Author summary

Mass spectrometry coupled with chromatography is among the most powerful analytical techniques used in the life sciences today. However, the generated datasets are large and complex, requiring specialized software or custom scripts for analysis. Therefore, data viewing and treatment in mass spectrometry create a barrier to entry for non-specialists within the field. Open-source, user-friendly tools for inspecting raw data, enabling evaluation of both chromatographic and mass spectrometric information, are lacking. Here we present LCMSpector, an easy-to-use graphical application for visualizing and analyzing hyphenated mass

Citation: Fido M, Hoesli E, Barazzone EC, Zenobi R, Slack E (2025) LCMSpector: A simple open-source viewer for targeted hyphenated mass spectrometry analysis. PLoS Comput Biol 21(12): e1013095. https://doi.org/10.1371/journal.pcbi.1013095

Editor: Mohammad Sadegh Taghizadeh, Shiraz University, IRAN, ISLAMIC REPUBLIC OF

Received: April 24, 2025

Accepted: November 28, 2025

Published: December 15, 2025

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pcbi.1013095

spectrometry data, available on Windows, MacOS, and Linux. Using several public datasets, we demonstrate how LCMSpector's comprehensive raw data viewing features aid targeted analysis of metabolic products. We benchmarked our application against software with similar functionality and conducted a small-scale usability study to evaluate adoption by non-specialists. The results indicate that LCMSpector lowers the technical barrier of entry and improves efficiency for researchers analyzing hyphenated mass spectrometry data. We hope this open-source software benefits the life science community and grows into a robust tool across subfields, such as metabolomics, proteomics, glycomics, and other "omics" employing hyphenated mass spectrometry.

Data availability statement: LCMSpector is distributed under the MIT license. The source code and executables for Windows, MacOS, and Linux, as well as detailed guides and installation instructions can be found free of charge on GitHub (https://github.com/MateuszFido/LCMSpector). As of the date of writing, LCMSpector is under constant development: the contents of this manuscript relate to the release version 0.9.10 (commit hash 95ad0e1, https://github.com/MateuszFido/LCMSpector/releases/tag/v0.9.10). Version 0.3 (script bundle) can be downloaded from Zenodo (DOI: https://doi.org/10.5281/zenodo.13990448). The overall test coverage of the codebase is 82% (source to test lines of code), with a 2:1 logic to UI testing ratio. The dependencies necessary for the installation and running of the non-executable versions are listed in the README and requirements text files. Documentation, learning resources and short articles describing the functionality are available on the project's wiki on GitHub: https://github.com/MateuszFido/LCMSpector/wiki. Links to the publicly available datasets reused in this article:—NIST mds2-2601 https://doi.org/10.18434/mds2-2601—license: public domain - MassIVE MSV000088442 https://doi.org/10.25345/C5M58V—license: CC0 1.0 - Public repository of ETH Zürich https://doi.org/10.3929/ethz-b-000663653—license (as stated by the repository host): https://rights-statements.org/page/InC-NC/1.0/—MassIVE MSV000092750 https://doi.org/10.25345/C5JD4Q057—license: CC0 1.0—MassIVE MSV000092751 https://doi.org/10.25345/C5DN40600—license: CC0 1.0.

Funding: ES, EH, and ECB were funded by the Basel Research Center for Child Health as part of the Multi-Investigator Project: Microbiota engineering for child health. ES acknowledges the support of the Swiss National Science Foundation (40B2-0_180953, 310030_185128), and European Research Council Consolidator Grant (865730). This work was supported as a part of NCCR Microbiomes, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 180575) and by the LOOP Zurich mTORUS project (Helmut Horten Foundation). ECB was paid by the Basel Research Center for Child Health. MF, RZ, and ES received salaries from the ETH Zurich. EH did not receive any specific funding for this project. The funders had no role in study

Introduction

Hyphenated mass spectrometry (MS) techniques have revolutionized the way analytical sciences shape today's progress in science, industry, and the clinic [1–7]. Owing to the rapid development in instrumentation, spectrometers are now proficient at analyzing thousands of samples in record times and are capable of resolving many previously challenging isomeric and isobaric interferences. Liquid chromatography–mass spectrometry (LC–MS) has been at the forefront of analytical chemistry, becoming a leading technique and the analytical method of choice in drug discovery [1,8,9], pharmacology [10], the omics sciences [5,11,12] clinical sciences [7,13], toxicology [3,14,15] and many others [16–18]. Other branches of MS have made strides in the fields of environmental sciences [19], structural biology [20,21], imaging [22,23], forensics [24,25], and virtually any other area where the sample's structure, identity, composition, or amount are of value.

Nevertheless, one of the consequences of the high throughput, precision, and accuracy of high-resolution MS measurements is that they tend to generate enormous datasets in the process. This "big data" problem requires mass spectrometrists or their collaborators to become skilled at conversion, processing, interpretation, and integration of this complex data or to employ additional bioinformaticians and data analysts for this purpose. The closed-source nature of the MS vendors' application programming interfaces exacerbates this problem by making raw data inaccessible for processing without the usage of their proprietary tools. This leads to customer lock-in and promotes unhealthy dependency on the vendors' products and workflow monopoly, a regrettably widespread practice [26].

There have been several attempts at bridging the gap between generating highly complex MS data and their processing and interpretation. Initiatives such as ProteoWizard [27], XCMS [28], or mzio's MZmine [29] have brought together large communities of applied MS users and are now widespread among research groups and companies alike. Since their inception, many related open-source chemometric tools [30,31], programming language libraries [32,33], frameworks and applications [34,35] have been developed, further increasing the diversity of applied conversion, processing, and analysis methods.

design, data collection and analysis, decision to publish, or preparation of the manuscript. The authors declare no financial conflict of interest.

Competing interests: The authors have declared that no competing interests exist.

Despite this, many challenges remain in the field of MS data analysis. The multitude of instruments, possible data acquisition methods, workflows, and statistical methods used in life sciences make it unrealistic to apply a cut-and-dried analysis pipeline offered by most currently available software. Moreover, as MS data generally requires significant computational power, many analytical platforms require the user to upload their data onto an external server, further limiting their independence and shifting the trust to a third party, as the data is no longer processed locally on their workstation or personal computer.

Here, we present LCMSpector, which is a completely free-of-charge, fully local, and open-source program that provides a graphical user interface developed for viewing and analysis of MS data. It integrates several Python libraries, enabling data visualization processing of different MS and LC data formats, and simple quantitative analysis. In this work, we present the main features of LCMSpector and demonstrate its application to the preprocessing and analysis of several hyphenated MS data sets.

Design and implementation

Application structure

The main graphical user interface is implemented via the model-view-controller architecture (MVC) using the PySide6 framework and Python bindings for Qt (LGPL license) [36]. The main thread runs the user interface, while most of the resource-heavy computation is relegated to separate processes and threads using Python’s built-in multiprocessing and concurrent.futures modules. This minimizes UI-thread blocking, maintains user interactivity, and makes maximum use of the available system resources. Files can be uploaded into LCMSpector via the “Upload” tab of the main tab view Qt widget, with a custom list view supporting both drag-and-drop functionality and manual file browsing by displaying the OS-native file browsing dialog. The absolute paths are stored as a list of strings and converted into derivatives of the Measurement class upon starting the processing. The subclasses of Measurement define several methods for preprocessing the raw data, storing the processed data as properties, and enabling additional plotting modules.

Supported input formats

The application accepts two types of data: text files (TXT and CSV file formats) containing chromatography data in a column-wise format, and either MS files (mzML) or peaklists with pre-assigned retention times, also in text format. For text-based formats, LCMSpector automatically determines the delimiter and applies a custom algorithm, skipping any optional file header and looking for two columns containing floating-point numbers. The default behavior considers the first found column as retention time as the last column as the detector signal. mzML is an XML-based, vendor-agnostic file format that can be translated from any major manufacturer’s propriety MS raw data file via tools such as MS Convert (ProteoWizard package) [27]. This allows for unified data sharing and analysis irrespective of the instrument and has been known for many years in the MS community alongside other formats such as mzXML, mzMLb, or HDF5 [37,38].

Data processing

After loading the data, LCMSpector performs the initial preprocessing. If operating in the LC/GC–MS mode, the chromatography and MS data are processed concurrently. The chromatography data is parsed from any text file format containing columns of data convertible to Python floating-point numbers. Subsequently, baseline interpolation and correction are performed (described in the Results section). The program simultaneously reads the mzML files into memory, discarding any scans other than MS [1]. It then looks for the closest *m*/*z* values to the user-specified ion lists within a specific mass-to-charge range (3 ppm by default). It then rebuilds the time-dependent ion traces (extracted ion chromatograms, XICs) for those *m*/*z* ranges, optionally connecting them to their chromatography retention times and automatically annotating the run with the desired MS-identified compounds.

Plotting

The plotting takes place in a separate module and is based on pyqtgraph, a Python scientific graphics and GUI Library for Python [39]. Pyqtgraph, as opposed to more popular and established libraries like matplotlib, maintains high user interactivity, enables real-time dynamic data processing such as log transformation, averaging, and fast Fourier transformation, and allows for exporting graphs in various formats. While not as mature as matplotlib (as of writing, 0.14.0 is the latest developer version of pyqtgraph), the library is comparatively lightweight, capable of performant plotting of millions of data points at the same time, dynamic panel docking and undocking, and embedding user-interactive and three-dimensional scientific graphs. It also provides limited capability for data export directly from the graph itself.

Data export and further analysis

The export from LCMSpector is a CSV file containing column-grouped data consisting of the filename, the traced ion's name, its m/z value and an optional description, the ion's retention time, integrated MS and LC intensities, and optional concentration. The format facilitates further processing or graphing via table-oriented software (e.g., R, pandas, GraphPad Prism, OriginLab Origin, Microsoft Excel) by saving data in vertical (long) format. The output of every analysis was subsequently processed using R.

Results

Working principle

The easy-to-use interface of LCMSpector provides very simple access to the processing of raw MS data. The graphical layout of the application consists of a tab view, which is split into three sections: the Upload tab, where the user enters their data and provides initial parameters for the analysis. The Results tab, where they can view, modify, and export the graphs plotted for each provided file, and the Quantitation tab is used to create calibration curves and calculate concentrations based on relative ion counts (Fig 1).

Upon launching the application, the only available tab is the Upload tab, where the user can input their data. The main window area holds tables responsible for storing the absolute paths to MS data saved on the user's filesystem, including shared and network disks. Each table widget supports drag-and-drop functionality for easy access, on top of being able to browse the operating system's file manager with the "Browse" button. Every widget enforces proper input format by checking the extension of imported files—this means that the user can drag-and-drop entire folders into a widget and only the files compatible with it are going to be uploaded. There are several safeguards in place to ensure proper application state before starting processing – the user gets prompted with a warning in case an input file is corrupt or unreachable, or no ion list is selected for processing. The software also has several stages of logging implemented for debugging purposes, so that whenever an exception is caught, it bubbles up the callers to one of the main MVC modules (Fig A in S1 Text).

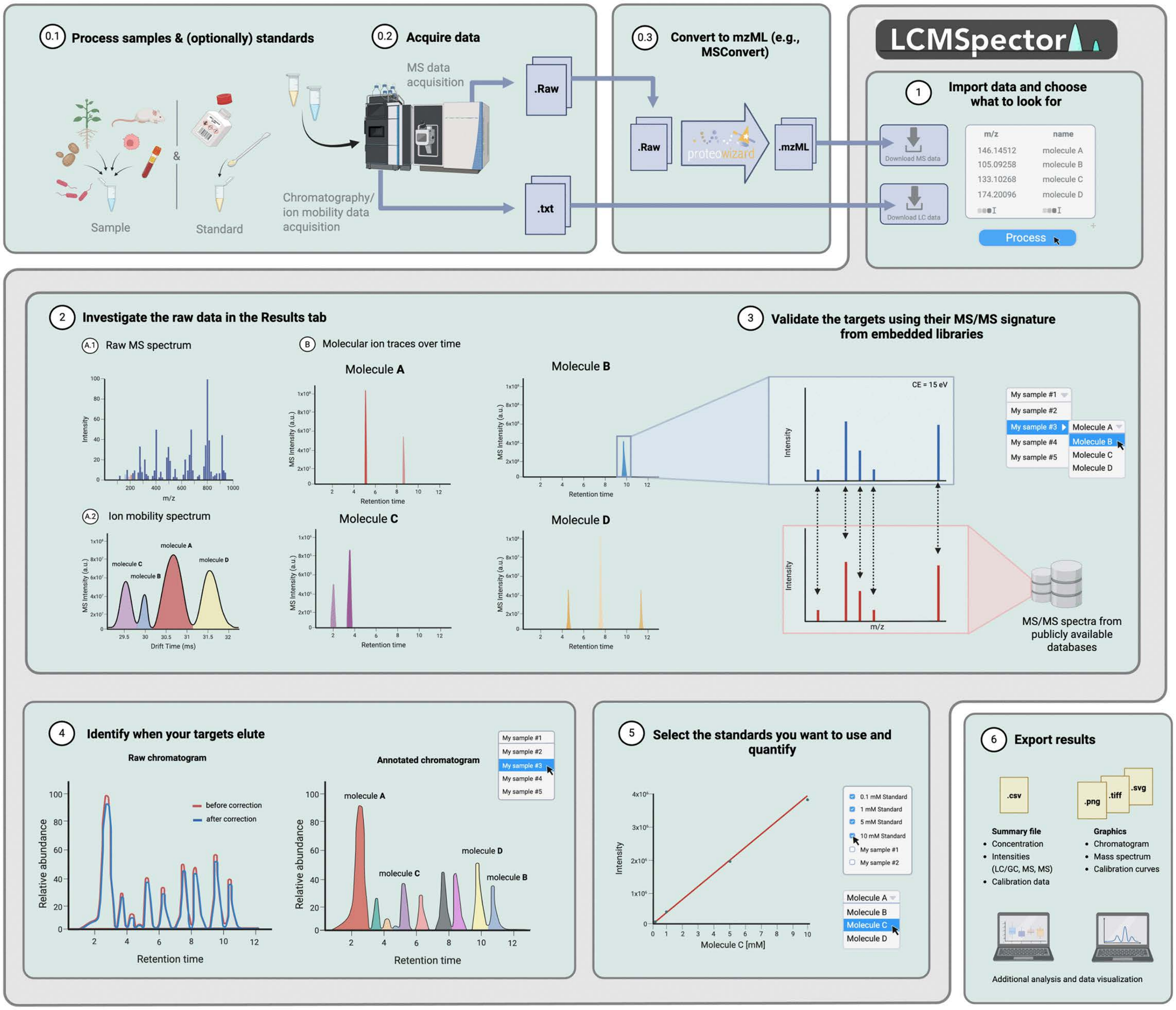

Fig 1. Flowchart illustrating the data flow between components of LCMSpector's user interface and data processing architecture. Created in BioRender with a publication license [40].

https://doi.org/10.1371/journal.pcbi.1013095.g001

LCMSpector can work in three separate modes: "LC/GC–MS", "MS Only", and "Chromatography Only". The user can choose the desired mode by using the combo box in the top left-hand corner of the Upload tab. The application then switches the input type, depending on the mode. The LC/GC–MS mode requires data in text file formats for the chromatography data, and the open mzML file format for the MS data. It then uses those data to display the chromatogram, the corresponding mass spectra at all retention times within the chromatogram, the chromatogram annotated with the respective *m*/*z* values and extracted ion chromatograms on which the annotation is based. The MS Only mode only requires mzML files for processing, treating the total ion current as a chromatogram-like representation of the time-resolved signal.

The Chromatography Only mode requires the input of chromatography data, as well as retention time-based manual peak labels to display annotated data.

Processing algorithms

The core features of LCMSpector, aside from the graphical interface, lie in its preprocessing techniques. One of the main examples is its chromatography baseline correction, which utilizes the Statistical Non-linear Iterative Peak clipping algorithm [41], similarly to the hplc-py package [42], where we saw good results for even highly variable LC run backgrounds, including negative absorbance values. Briefly, the algorithm works in a series of steps:

1. Take the original chromatography signal (detector response over time), S
2. Compress the signal through approximation of the baseline, applying the compressing LLS operator [43]:

$$S_{LLS} = \ln\left[\ln\left(\sqrt{S+1}+1\right)+1\right],$$

where S is the original chromatography signal. This step helps to reduce the impact of extreme values in the signal.

3. Perform iterative filtering to remove noise: The algorithm filters the compressed signal by selecting the minimum value between the current signal intensity and the average of the signal intensities at neighboring points:

$$S_{\text{iter}_i} = \min\left[S_{LLS_{i-1}}(t),\quad \frac{S_{LLS_{i-1}}(t-i)+S_{LLS_{i-1}}(t+i)}{2}\right],$$

where i is the current iteration. This step helps to remove noise from the signal. LCMSpector uses a fixed default number of 20 iterations for all chromatography data, as we noticed this to be sufficient for most applications while keeping the computational preprocessing load relatively low.

4. Transform back: Finally, the filtered signal S_{iter} is transformed back to its original scale with the inverse of the LLS operator and subtracted from the original signal (Fig 2).

Fig 2A demonstrates the application of the SNIP algorithm on a very noisy spectrum obtained from the GNPS-MassIVE repository (MSV000088442, Agilent 1260 HPLC and Bruker micrOTOF-Q MS) [44]. The interpolated background is visibly contributing to the signal as the run progresses. Upon subtraction, the resulting corrected UV spectrum contains much sharper peaks and an effectively null baseline. Fig 2B presents the performance of the algorithm on a GC spectrum recorded with an Agilent 8890/5977B GC–MS system (NIST Public Data Repository, https://doi.org/10.18434/mds2-2601) [45]. The plots are directly exported from LCMSpector (and recolored), facilitated by the underlying graphing library, which also allows extensive editing, removing, and transforming the plotted data in real-time.

Another very important algorithm in LCMSpector's processing pipeline is the construction of XICs, which is critical for proper MS-based compound quantification. It is also the most computationally expensive part of the processing step, needing to iterate over all the MS scans S found in the input files. For each scan, it needs to process its m/z array M, the corresponding intensities, and retrieve the relevant metadata, such as the injection time T. Then, for each compound C targeted by the user, the ion current needs to be processed within the m/z boundaries for the given ion I expected for this compound, yielding MS peak area as a function of scan number (or time), otherwise known as the ion chromatogram.

Computationally, a naïve implementation would have a polynomial time complexity:

$$O(C \times I \times S \times (M+T))$$

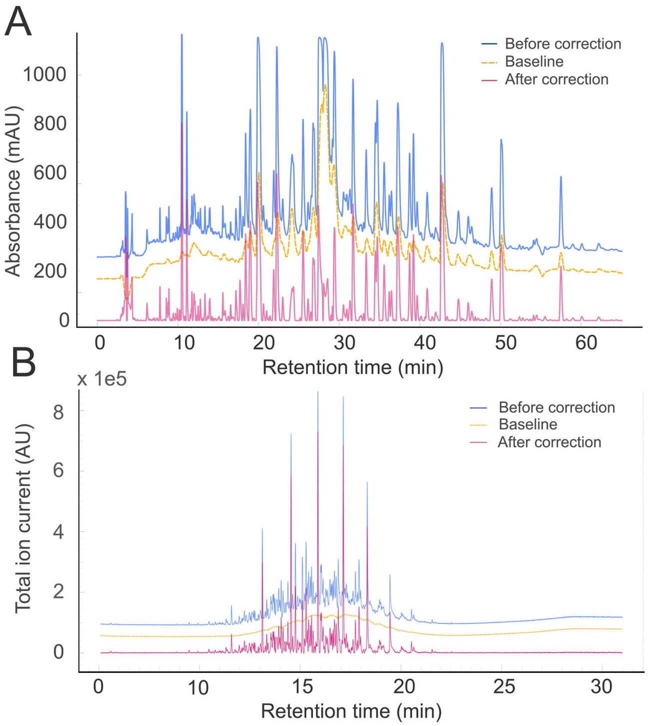

Fig 2. A) An LC chromatogram and B) a GC chromatogram before (blue, top) and after (red, bottom) applying the SNIP algorithm (20 iterations). The interpolated baseline is highlighted in yellow dashed lines (in between). Data taken from public data repositories A (GNPS-MassIVE repository, MSV000088442) and B (NIST Public Data Repository, https://doi.org/10.18434/mds2-2601).

https://doi.org/10.1371/journal.pcbi.1013095.g002

where T is a nested dictionary lookup, which in Python has a complexity of O(1). This therefore reduces to:

$$O(C \times I \times S \ \times \ M)$$

The current design of LCMSpector precomputes the scan metadata to avoid repeated lookups, groups ions avoiding re-scanning data for identical ion *m*/*z* across compounds, and uses a binary search for *m*/*z* array traversal, reducing per-scan work from linear $O(M)$ to logarithmic $O(\log M)$. The algorithm's complexity, while still mostly influenced by C, I, and S, is substantially faster in practice, especially as M gets large for high-resolution instruments:

$$O(U \times S \ \times \ \log(M))$$

where U is the unique ions grouped from all compounds $(U \ \leq C \ \times \ I)$.

Quantification

Targeted MS workflows often involve some kind of quantification, particularly within the omics sciences [46–48]. To support this, LCMSpector enables the user to select a subset of their input data as calibration data with known concentrations. Only one concentration value per file is supported, i.e., a single file represents a single calibrant data point, although an unlimited number of compounds is supported. Since this approach presupposes a large number of files, concentrations are imputed from the filename, though they can also be entered and modified manually. The user can choose to use either chromatography data or MS ion counts for the calibration procedure, depending on the mode. Upon selecting the calibration files and clicking the "Calculate" button, a list of calibration curves is presented for every compound that was initially declared for tracing.

Application to a biologically relevant dataset

The main purpose of LCMSpector is facilitating chromatographic and mass spectrometric investigation of biological samples, leveraging their analytical power without requiring an analytical background or programming skills. To illustrate this, we attempted to showcase the application's utility by examining publicly available LC–MS and GC–MS datasets acquired in a study of hospitalized patients with chronic liver disease, treated with the non-digestible disaccharide lactulose (GNPS-MassIVE repository IDs MSV000092750 and MSV000092751) [49]. The authors collected metagenomic and metabolomic data from fecal samples, providing insight into the expansion of commensal *Bifidobacteria* following lactulose treatment, reducing infections caused by antibiotic-resistant pathobionts, and resulting in improved patient outcomes. The main analytical focus of the work was bile acids, short-chain fatty acids, as well as tryptophan-pathway metabolites significant of microbial activity, such as indole derivatives. To verify and present the applicability of LCMSpector's processing pipeline to this type of a cross-sectional study, we performed quantitative analysis summarizing all the differentially abundant compounds from the cohorts presenting the expansion of *Bifidobacteria* (Fig 3A) and then performed additional exploratory work using LCMSpector's built-in and easy-to-use raw data viewing capabilities (Fig 3C–3E).

Fig 3B shows raw exports from LCMSpector's Quantitation tab. We recreated calibration curves for all the compounds for which the authors [49] recorded serial dilutions (bile acids and short-chain fatty acids). We subsequently used the calibration values to recalculate the concentrations of the relevant compounds in the fecal samples coming from healthy donors and liver disease patients (Fig 3C). A major advantage of LCMSpector is the easy graphical extraction of ion chromatograms alongside raw view data, allowing the user to recognize other features of the analysis that may increase quantitative accuracy. The data shown in Fig 3D shows an abundant potential dimer peak coeluting with the primary pseudomolecular ion for all the bile acids used for calibration (Fig 3D). The relative proportion of these species varied from sample to sample, in some cases having a higher abundance than the primary ion (Fig 3E). Gas-phase dimerization is a common non-covalent phenomenon at high analyte concentrations in electrospray ionization mass spectrometry [50,51]. LCMSpector easily allows for accurate hyphenated MS data analysis by enabling such untargeted discovery thanks to its raw data viewing capabilities.

Non-standard mass spectrometry

Considering the continuous rapid evolution and development of MS, new hyphenated techniques emerge in the field on a regular basis. As a means of standardizing the input, LCMSpector accepts mzML as its file format of choice, as it is ubiquitous in the field, and its conversion is supported by all the major vendors [38]. This means that any MS data that can be written in mzML can be analyzed, including flow injection and direct infusion techniques.

One example of such a method is secondary electrospray ionization mass spectrometry (SESI-MS), wherein a gaseous analyte collides with a charged electrospray, producing secondary ions [52]. The method has gained popularity in the field of breath analysis and general analysis of trace vapors. Fig 4 shows the application of LCMSpector to SESI-MS data (ThermoFisher Scientific Q Exactive Plus) by using the MS Only mode. As the technique does not feature any chromatographic separation, the chromatogram is replaced by the total ion current, which serves as a global overview of the evolution of the signal across the time dimension.

As of today, there is no agreed-upon general analysis pipeline for the preprocessing and presentation of SESI-MS data. Most researchers default to analyzing and presenting data in the form of untargeted metabolomics, rarely investigating individual putative metabolites. LCMSpector was created with targeted approaches in mind, facilitating the exploration of small molecules underlying the examined experimental conditions, preferably confirmed with further measurements (e.g., with tandem MS, derivatization, standard spiking, etc.). Our hope is that LCMSpector allows diverse fields of researchers

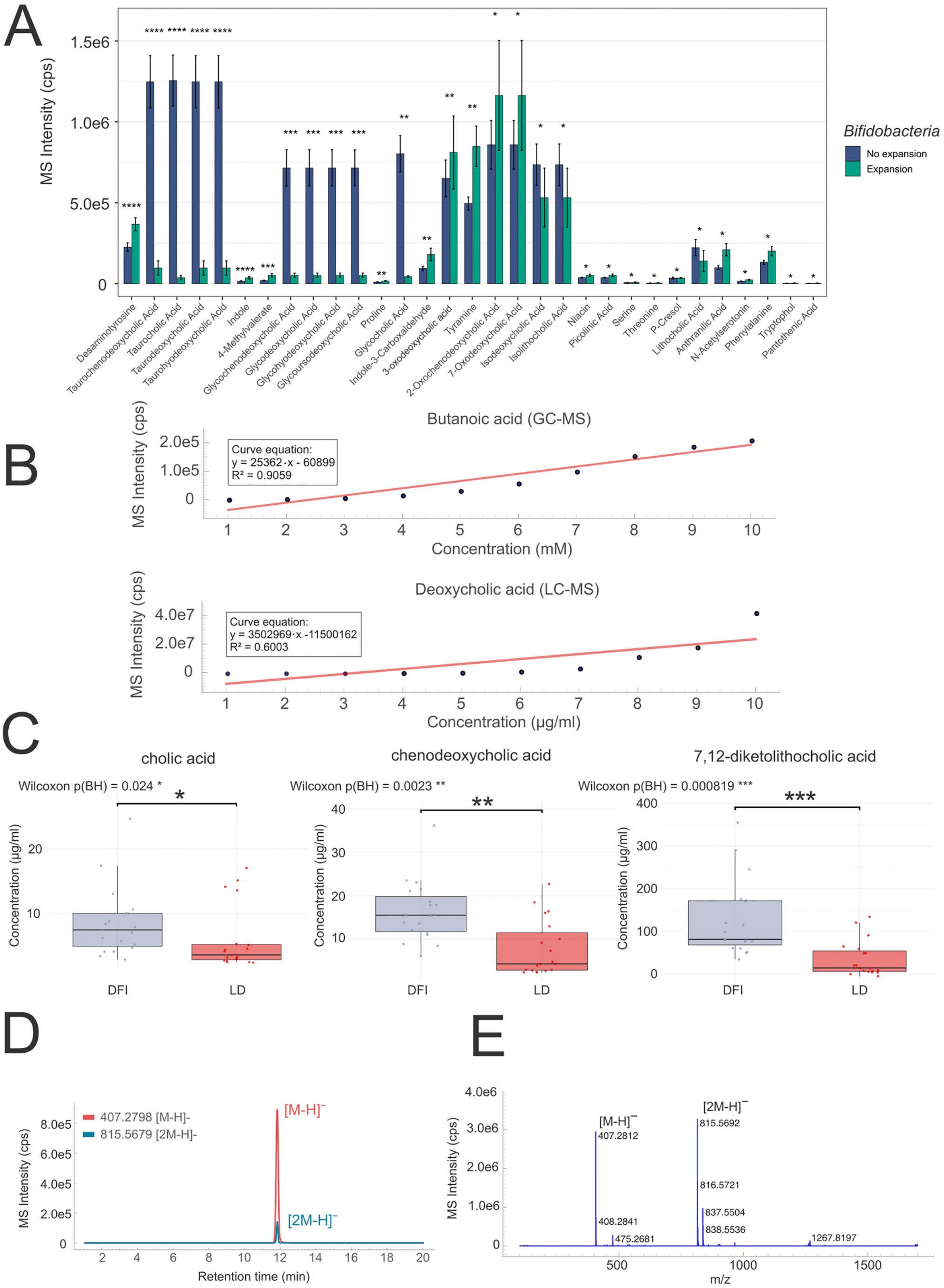
A
MS Intensity (cps)
1.5e6
1.0e6
5.0e5
0
Bifidobacteria
No expansion
Expansion
Desaminotyrosine
Taurochenodeoxycholic Acid
Taurocholic Acid
Taurodeoxycholic Acid
Taurohyodeoxycholic Acid
Indole
4-Methylvalerate
Glycochenodeoxycholic Acid
Glycodeoxycholic Acid
Glycohyodeoxycholic Acid
Glycoursodeoxycholic Acid
Proline
Glycocholic Acid
Indole-3-Carboxaldehyde
3-oxodeoxycholic acid
Tyramine
2-Oxochenodeoxycholic Acid
7-Oxodeoxycholic Acid
Isodeoxycholic Acid
Isolithocholic Acid
Niacin
Picolinic Acid
Serine
Threonine
P-Cresol
Lithocholic Acid
Anthranilic Acid
N-Acetylserotonin
Phenylalanine
Tryptophol
Pantothenic Acid
B
Butanoic acid (GC-MS)
MS Intensity (cps)
2.0e5
1.0e5
0
Curve equation:
y = 25362·x - 60899
R² = 0.9059
Concentration (mM)
Deoxycholic acid (LC-MS)
4.0e7
2.0e7
Curve equation:
y = 3502969·x -11500162
R² = 0.6003
Concentration (µg/ml)
C
cholic acid
Wilcoxon p(BH) = 0.024 *
chenodeoxycholic acid
Wilcoxon p(BH) = 0.0023 **
7,12-diketolithocholic acid
Wilcoxon p(BH) = 0.000819 ***
Concentration (µg/ml)
DFI
LD
D
407.2798 [M-H]-
815.5679 [2M-H]-
[M-H]⁻
[2M-H]⁻
MS Intensity (cps)
8.0e5
6.0e5
4.0e5
2.0e5
Retention time (min)
E
[M-H]⁻
407.2812
408.2841
475.2681
[2M-H]⁻
815.5692
816.5721
837.5504
838.5536
1267.8197
MS Intensity (cps)
4.0e6
3.0e6
2.0e6
1.0e6
m/z

Fig 3. Overview of the reanalyzed dataset from the study by Odenwald and colleagues (*Nat. Microbiol.* 2023, 8 (11), 2033–2049, MassIVE repositories MSV000092750 and MSV000092751) [49]. A) Bar plots summarizing the most significant differentially abundant compounds in patients with greater than or equal to (green) and lower than (blue) 10% of total abundance of *Bifidobacteria* in fecal samples. **B)** Example calibration curves exported from LCMSpector based on the GC–MS (top) data for butanoic acid and LC–MS (bottom) data for deoxycholic acid. **C)** Example box plots of differentially abundant bile acids in the healthy (DFI, blue) and liver disease (LD, red) patient populations. **D)** Extracted ion chromatogram from a 10 μg/ml concentration standard solution of allocholic acid and **E)** mass spectrum showing the presence of a pseudomolecular dimer peak [2M − H]$^-$ which could be included to further improve the accuracy of quantification of total intensity of allocholic acid.

https://doi.org/10.1371/journal.pcbi.1013095.g003

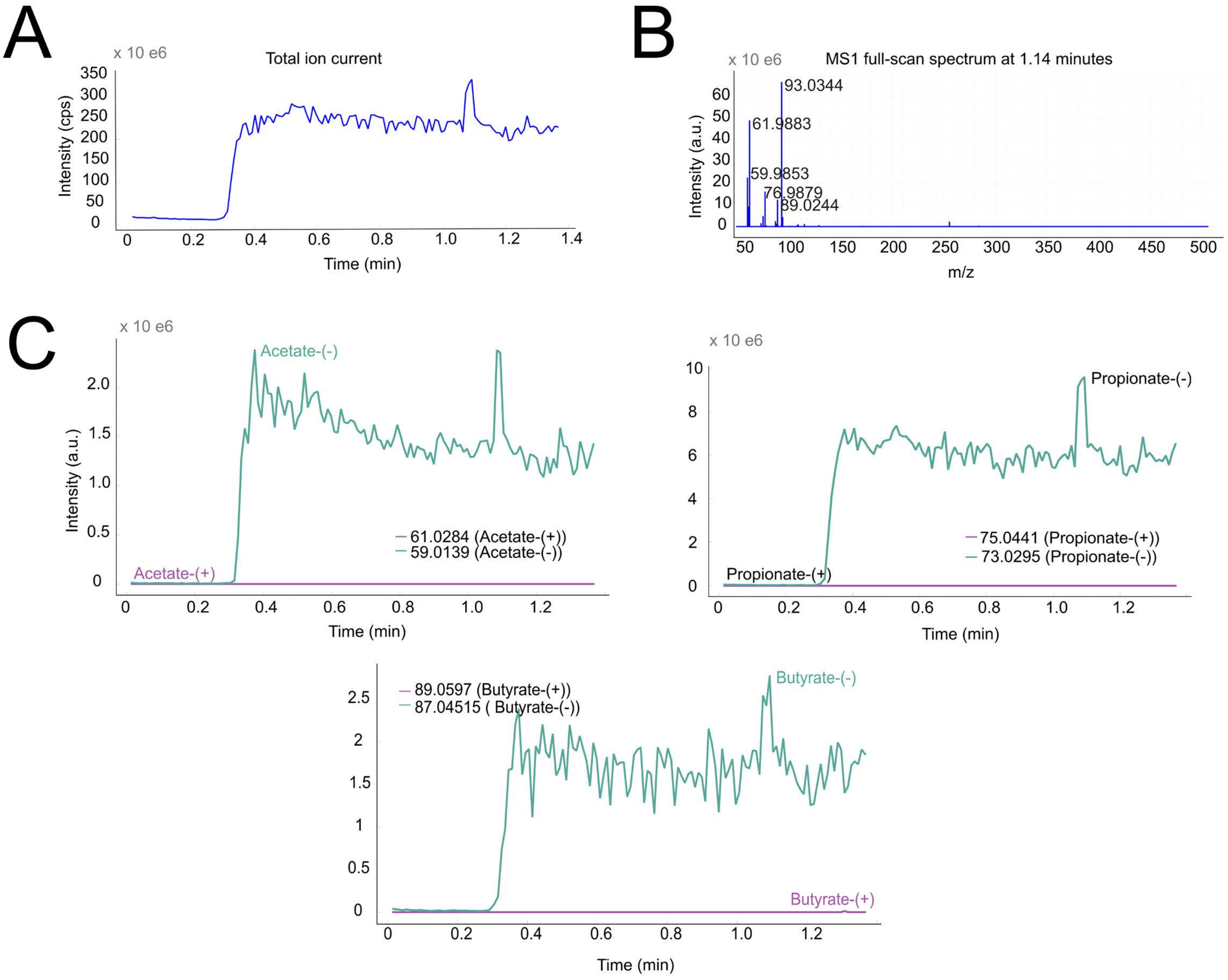

Fig 4. SESI-HRMS samples of human breath gas, which were stored in an ethylene vinyl alcohol copolymer sampling bag produced in-house, direct graphical output of LCMSpector, part of a published dataset [53,54]. A) Interactive plot of the total ion current, which displays the underlying mass spectrum **B)** when the user clicks on a corresponding scan time of the TIC. **C)** Extracted ion chromatograms of short-chain fatty acids present in the exhaled breath samples (acetic acid, propionic acid, butyric acid) can be seen correlating with the total ion current.

https://doi.org/10.1371/journal.pcbi.1013095.g004

expand into targeted workflows and pursue confirmation of the traced metabolites, which our software aims to assist and expedite.

Performance benchmarking

MS output is often classified as "big data" because of its high dimensionality, complexity, and sampling density. For example, LC–MS workflows produce a time-series signal at regular intervals for every ion they measure. With standard injection times in the order of tens to hundreds of milliseconds, an average MS run yields thousands of scans (e.g., 4,500 for a 15-min injection assuming 5 scans per second). Every scan carries multidimensional information consisting of the m/z values of detected ions and their intensities, and usually additional metadata such as charge states, polarity, retention time, etc. For high-resolution instruments (mass accuracy of 0.001 Da and greater) measuring between 100 and 1,000 *m/z*, the uncompressed intensity and m/z arrays would sum up to:

$$2 \cdot \frac{m/z_{\text{high}} - m/z_{\text{low}}}{\text{mass accuracy}} = 2 \cdot \frac{1000 - 100}{0.001} = 1.8 \times 10^6 \text{ data points}$$

per each scan, resulting in file size in the order of gigabytes ($1\ \text{GB} = 8 \times 10^9$ bits).

Therefore, proper loading and in-memory handling of data can be considered the most important bottleneck for any MS software. To assess the performance of LCMSpector in these tasks, we conducted thorough scalability tests on all the major platforms (Windows, MacOS, and Linux), consisting of loading and processing a series of MS files converted to the mzML format. We benchmarked against the latest versions of two established open-source alternatives with very similar functionality: mzio's MZMine and OpenMS TOPPView (Fig 5).

Both LCMSpector and MZMine provide verbose logging functionality which allowed for exact measurement of the loading and processing time, whereas TOPPView's loading time was measured manually: from when files were dropped into the user interface until the UI became available for interaction, indicating loading was complete. Additionally, processing time for MZMine and LCMSpector was measured by performing targeted feature extraction of 40 *m/z* values—pseudomolecular ions for the 20 essential amino acids in both positive and negative ion modes. In TOPPView this operation is only possible on one layer (file) at a time and would otherwise necessitate the use of the FeatureFinder tool orchestrated via the TOPPAS workflow design GUI, and so extends beyond TOPPView's functionality. After completing the loading and processing steps (or just loading in the case of TOPPView), peak memory usage was assessed using the respective platform's resource monitor.

We found LCMSpector and MZMine to perform roughly on par in terms of loading and processing speed, both using parallelized multithreaded loading pipelines, with MZMine additionally incorporating memory-mapping for even faster file access. TOPPView was much slower on every platform, caused primarily by sequential I/O operations. In terms of resource usage, LCMSpector performed the best out of all three programs, using about half the memory of TOPPView, and around 2–4 times less than MZMine, depending on the platform. This is likely because only the essential scan data (no extra metadata) is loaded and smaller C data structures are used by compiling Python to C code with a source-to-source compiler, Nuitka [55].

We also assessed LCMSpector's peak integration and quantification by measuring precision and accuracy of its concentration estimates against known compound concentrations and comparing results to the vendor's algorithm (Thermo Scientific's Chromeleon) and another open-source alternative, MZMine. Five amino acids (aspartic acid, glutamic acid, histidine, lysine, serine) were combined into a 10 mM calibration mix and serially diluted to 1 mM, 100 µM, 10 µM, and 1 µM. Each dilution was directly infused into a high-resolution ThermoScientific Exploris 240 mass spectrometer and full-MS data were recorded to construct calibration curves. For all programs, we extracted XICs using the same m/z window: $[M+H]^+ \pm 0.0003$ Da, integrated the peak intensities, and built 5-point linear calibration curves. We then interpolated

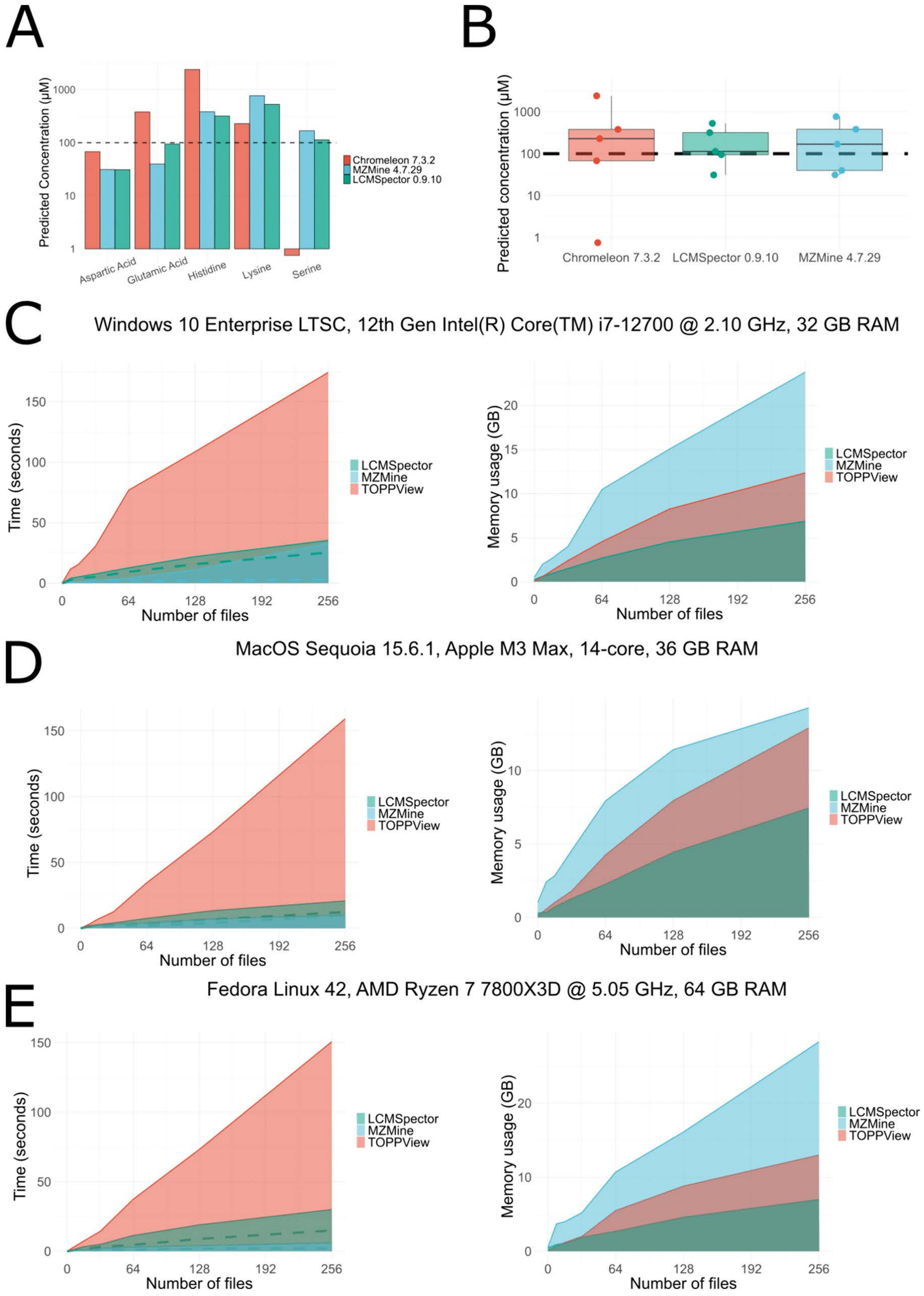

Fig 5. Performance and quantitation capability benchmarking of LCMSpector. Benchmarking data was generated on three different platforms, as denoted in each respective Fig's title. **A)** Barplots showing concentrations (logarithmic scale) of five different amino acids predicted by Chromeleon 7.3.2 (red, left), MZMine 4.7.29 (blue, middle), and LCMSpector 0.9.10 (green, right). The dashed horizontal line marks the real concentration (100 µM). **B)**

Residual boxplots showing the precision and accuracy of each software's prediction of a 100 μM concentration of each of the five tested amino acids. The dashed horizontal line marks the real concentration (100 μM). The solid horizontal line on each boxplot signifies the mean predicted concentration for all five amino acids. **C** Windows, **D)** MacOS, **E)** Linux performance comparisons of LCMSpector 0.9.10, MZMine 4.7.29 and OpenMS TOPPView 3.4.1 on. Left: Time taken to load (solid line) and process (dashed line) as a function of the number of input mzML files. Right: Peak memory usage as a function of the number of input mzML files. Red: TOPPView, blue: MZMine, green: LCMSpector.

https://doi.org/10.1371/journal.pcbi.1013095.g005

each curve to estimate the concentration at the 100 μM point and compared those predicted values across Chromeleon, MZMine, and LCMSpector.

Somewhat surprisingly, we found the vendor algorithm (Chromeleon) to perform the poorest out of all three, whereas both LCMSpector and MZMine generally produced very accurate estimates of the real concentration for all 5 compounds. Based on our experience, this is most likely because Chromeleon's integration seems to be designed to work on well-defined, chromatographically-resolved peaks—its algorithms require a clear baseline and peak contours to work properly. LCMSpector also uses retention time-based peak integration when available, automatically defining peak boundaries and baseline-correcting the MS signal. However, if this strategy fails, there is a built-in fallback mechanism to sum the intensities for the given ion over time using trapezoidal integration, similarly to MZMine [56]. Nonetheless, LCMSpector's approach was the most precise and the most accurate among all three when tested on this data (Fig 4B).

User-friendliness

To assess the approachability of LCMSpector to newcomers, we conducted a small usability study among ten users of different platforms that the application is distributed on (Windows, MacOS, and Linux). The users were given a set of instructions (Appendix A in S1 Text), consisting of downloading and installing the suitable executable for their platform, and then testing its various features on a set of sample LC-MS data. They were then asked to complete a short questionnaire on ease of use and functionality, rating installation difficulty, diversity of predefined compound lists, plotting-window operability, and reporting any bugs or confusing points (Appendix B in S1 Text). None of the users had previous experience in using software for MS data analysis.

The most represented platform was Windows (Fig 6). The median score for the easiness-of-use was 5 out of 6, indicating that no user found the application to be difficult or confusing. Most users described the installation process as straightforward.

Importantly, most basic functions, such as loading a file, starting the processing step, exporting the results and where the export is saved, were evident to all users. This indicates that the user-interface of LCMSpector was designed properly, and the core functionality is easily accessible and understandable.

With regards to potential improvements, the users cited stability, with some experiencing crashes when executing multiple functionalities at the same time, such as changing viewed files in the Results tab and reprocessing the data. This is a valuable indicator that additional safeguards against race conditions and potential premature garbage collection should be implemented, as well as expanding the codebase's integration testing as it matures and new features and functions are introduced.

Limitations

Despite its performance benefits and user-friendliness, LCMSpector is not without limitations. Currently, the main barrier to widespread adoption is the lack of functionality supporting many features useful in mass spectrometry—customizable peak-picking algorithms for discovery (untargeted) workflows, better support for data-independent and single/multiple-reaction-monitoring experiments or more sophisticated ion mobility features (e.g., loading and viewing are currently possible, but 2D analysis is not). Likewise, one important pain point in community adoption is the handling of

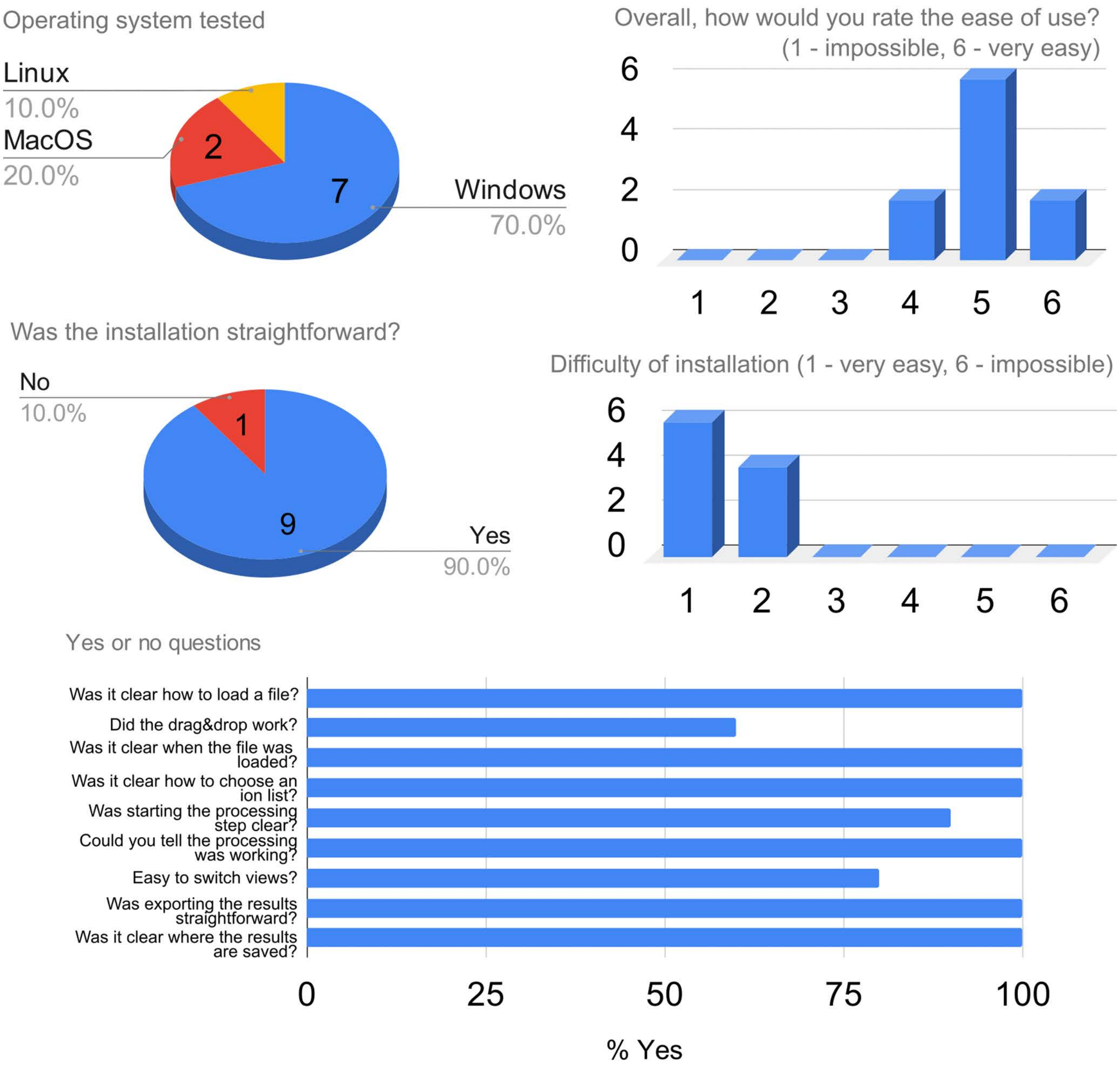

Fig 6. Results of the usability study.

https://doi.org/10.1371/journal.pcbi.1013095.g006

proprietary file formats and other open file formats, not only mzML. Loading performance could also be improved upon by creating Cythonized versions of the pyteomics library's reader objects and incorporating memory-mapping, as in this regard LCMSpector is currently still outperformed by MZMine (Fig 5C–5E). Additionally, stability issues experienced by users of some platforms need to be carefully inspected to provide a fluid experience regardless of operating system and processor architecture. Nonetheless, all these points are included in the roadmap for LCMSpector's future development (Fig B in S1 Text) and we hope that with further development and community contributions, these barriers can soon be overcome.

Availability and future directions

We presented the free, local, and open-source graphical interface software LCMSpector, enabling quick and simple viewing and investigation of MS data and discussed it in the scope of currently available technology. As MS becomes more widespread and common in laboratories throughout the world, parsing and evaluating the information it produces need to keep up with the technique's prominence. In our view, this means removing barriers of entry wherever possible and making the analysis streamlined and available also for non-experts, which is why we hope this software helps adopt targeted workflows with minimal experience in MS and data analysis. This could have broad appeal in the related fields, such as proteomics, lipidomics, glycomics, and metabolomics.

LCMSpector is distributed under the MIT license. The source code and executables for Windows, MacOS, and Linux, as well as detailed guides and installation instructions can be found free of charge on GitHub (https://github.com/MateuszFido/LCMSpector). As of the date of writing, LCMSpector is under constant development: the contents of this manuscript relate to the release version 0.9.10 (commit hash 95ad0e1, https://github.com/MateuszFido/LCMSpector/releases/tag/v0.9.10). Version 0.3 (script bundle) can be downloaded from Zenodo (DOI: https://doi.org/10.5281/zenodo.13990448). The overall test coverage of the codebase is 82% (source to test lines of code), with a 2:1 logic to UI testing ratio. The dependencies necessary for the installation and running of the non-executable versions are listed in the README and requirements text files. Documentation, learning resources and short articles describing the functionality are available on the project's wiki on GitHub: https://github.com/MateuszFido/LCMSpector/wiki.

Supporting information

S1 Text. Fig A. **(A)** The MVC architectural design of LCMSpector, created in BioRender with a publication license. Wetter, E. (2025) https://BioRender.com/4e4i8uvand. **(B)** LCMSpector data flow diagram. **Fig B.** Roadmap showing the plans for future development of LCMSpector. **Table A.** Windows—results of performance benchmarking between LCMSpector, MZMine and Sirius loading the same sets of mzML files. **Table B.** MacOS—results of performance benchmarking between LCMSpector, MZMine and Sirius loading the same sets of mzML files. **Table C.** Linux—results of performance benchmarking between LCMSpector, MZMine and Sirius loading the same sets of mzML files. **Fig D.** Calibration curve of aspartic acid from the calibration panel in Chromeleon Chromatography Studio (ver. 7.3.2) and calibration curve created based on the integrated intensities created in R (log–log). **Fig E.** Calibration curve of glutamic acid from the calibration panel in Chromeleon Chromatography Studio (ver. 7.3.2) and calibration curve created based on the integrated intensities created in R (log–log). **Fig F.** Calibration curve of histidine from the calibration panel in Chromeleon Chromatography Studio (ver. 7.3.2) and calibration curve created based on the integrated intensities created in R (log–log). **Fig G.** Calibration curve of lysine from the calibration panel in Chromeleon Chromatography Studio (ver. 7.3.2) and calibration curve created based on the integrated intensities created in R (log–log). **Fig H.** Calibration curve of serine from the calibration panel in Chromeleon Chromatography Studio (ver. 7.3.2) and calibration curve created based on the integrated intensities created in R (log–log). **Fig I** Calibration curve (log–log) of aspartic acid produced in R based on the integrated peak areas from MZmine version 4.7.28. **Fig J.** Calibration curve (log–log) of glutamic acid produced in R based on the integrated peak areas from MZmine version 4.7.28. **Fig K.** Calibration curve (log–log) of histidine produced in R based on the integrated peak areas from MZmine version 4.7.28. **Fig L.** Calibration curve (log–log) of lysine produced in R based on the integrated peak areas from MZmine version 4.7.28. **Fig M.** Calibration curve (log–log) of serine produced in R based on the integrated peak areas from MZmine version 4.7.28. **Fig N.** Calibration curve (log–log) of aspartic acid produced in R based on the integrated peak areas from LCMSpector 0.9.10. **Fig O.** Calibration curve (log–log) of glutamic acid produced in R based on the integrated peak areas from LCMSpector 0.9.10. **Fig P.** Calibration curve (log–log) of histidine produced in R based on the integrated peak areas from LCMSpector 0.9.10. **Fig Q.** Calibration curve (log–log) of lysine produced in R based on the integrated peak areas from LCMSpector 0.9.10. **Fig R.** Calibration curve (log–log) of serine produced in R

based on the integrated peak areas from LCMSpector 0.9.10. **Appendix A.** Instructions for the usability study of LCMSpector. **Appendix B.** Contents of the usability study questionnaire.
(DOCX)

Acknowledgments

The authors would like to thank the early-stage users of LCMSpector for helpful discussions and remarks when using the first versions of the software.

Author contributions

Conceptualization: Mateusz Fido.

Data curation: Mateusz Fido, Etienne Hoesli.

Formal analysis: Mateusz Fido.

Funding acquisition: Renato Zenobi, Emma Slack.

Investigation: Mateusz Fido, Etienne Hoesli.

Methodology: Mateusz Fido.

Project administration: Mateusz Fido, Emma Slack.

Resources: Elisa Cappio Barazzone.

Software: Mateusz Fido.

Supervision: Renato Zenobi, Emma Slack.

Visualization: Mateusz Fido, Etienne Hoesli.

Writing – original draft: Mateusz Fido, Etienne Hoesli, Renato Zenobi, Emma Slack.

Writing – review & editing: Mateusz Fido.

References

1. Dueñas ME, Peltier-Heap RE, Leveridge M, Annan RS, Büttner FH, Trost M. Advances in high-throughput mass spectrometry in drug discovery. EMBO Mol Med. 2023;15(1):e14850. https://doi.org/10.15252/emmm.202114850 PMID: 36515561
2. Metz TO, Zhang Q, Page JS, Shen Y, Callister SJ, Jacobs JM, et al. The future of liquid chromatography-mass spectrometry (LC-MS) in metabolic profiling and metabolomic studies for biomarker discovery. Biomark Med. 2007;1(1):159–85. https://doi.org/10.2217/17520363.1.1.159 PMID: 19177179
3. Maurer HH. Hyphenated high-resolution mass spectrometry-the “all-in-one” device in analytical toxicology?. Anal Bioanal Chem. 2021;413(9):2303–9. https://doi.org/10.1007/s00216-020-03064-y PMID: 33247339
4. Tanna S, Ogwu J, Lawson G. Hyphenated mass spectrometry techniques for assessing medication adherence: advantages, challenges, clinical applications and future perspectives. Clin Chem Lab Med. 2020;58(5):643–63. https://doi.org/10.1515/cclm-2019-0820 PMID: 31926065
5. Karpievitch YV, Polpitiya AD, Anderson GA, Smith RD, Dabney AR. Liquid chromatography mass spectrometry-based proteomics: biological and technological aspects. Ann Appl Stat. 2010;4(4):1797–823. https://doi.org/10.1214/10-AOAS341 PMID: 21593992
6. Zhou B, Xiao JF, Tuli L, Ressom HW. LC-MS-based metabolomics. Mol Biosyst. 2012;8(2):470–81. https://doi.org/10.1039/c1mb05350g PMID: 22041788
7. Grebe SK, Singh RJ. LC-MS/MS in the clinical laboratory—where to from here?. Clin Biochem Rev. 2011;32(1):5–31.
8. Chen G, Pramanik BN, Liu Y-H, Mirza UA. Applications of LC/MS in structure identifications of small molecules and proteins in drug discovery. J Mass Spectrom. 2007;42(3):279–87. https://doi.org/10.1002/jms.1184 PMID: 17295416
9. Apostol I, Bondarenko PV, Ren D, Semin DJ, Wu C-H, Zhang Z, et al. Enabling development, manufacturing, and regulatory approval of biotherapeutics through advances in mass spectrometry. Curr Opin Biotechnol. 2021;71:206–15. https://doi.org/10.1016/j.copbio.2021.08.001 PMID: 34508981
10. Beccaria M, Cabooter D. Current developments in LC-MS for pharmaceutical analysis. Analyst. 2020;145(4):1129–57. https://doi.org/10.1039/c9an02145k PMID: 31971527

11. Bian Y, Zheng R, Bayer FP, Wong C, Chang Y-C, Meng C, et al. Robust, reproducible and quantitative analysis of thousands of proteomes by micro-flow LC-MS/MS. Nat Commun. 2020;11(1):157. https://doi.org/10.1038/s41467-019-13973-x PMID: 31919466

12. Goli M, Yu A, Cho BG, Gautam S, Wang J, Gutierrez-Reyes CD, et al. LC-MS/MS in glycomics and glycoproteomics analyses. Carbohydrate Analysis by Modern Liquid Phase Separation Techniques. Elsevier. 2021. p. 391–441. https://doi.org/10.1016/b978-0-12-821447-3.00005-6

13. Seger C, Salzmann L. After another decade: LC-MS/MS became routine in clinical diagnostics. Clin Biochem. 2020;82:2–11. https://doi.org/10.1016/j.clinbiochem.2020.03.004 PMID: 32188572

14. Panda D, Dash BP, Manickam S, Boczkaj G. Recent advancements in LC-MS based analysis of biotoxins: Present and future challenges. Mass Spectrom Rev. 2022;41(5):766–803. https://doi.org/10.1002/mas.21689 PMID: 33624883

15. Viette V, Hochstrasser D, Fathi M. LC-MS (/MS) in clinical toxicology screening methods. Chimia (Aarau). 2012;66(5):339–42. https://doi.org/10.2533/chimia.2012.339 PMID: 22867548

16. Thevis M, Thomas A, Pop V, Schänzer W. Ultrahigh pressure liquid chromatography-(tandem) mass spectrometry in human sports drug testing: possibilities and limitations. J Chromatogr A. 2013;1292:38–50. https://doi.org/10.1016/j.chroma.2012.12.048 PMID: 23332783

17. Van Bocxlaer JF, Clauwaert KM, Lambert WE, Deforce DL, Van den Eeckhout EG, De Leenheer AP. Liquid chromatography—mass spectrometry in forensic toxicology. Mass Spectrom Rev. 2000;19(4):165–214. https://doi.org/10.1002/1098-2787(200007)19:4%253C165::AID-MAS1%253E3.0.CO;2-Y

18. Zhao Y, Gericke O, Li T, Kjaerulff L, Kongstad KT, Heskes AM, et al. Polypharmacology-labeled molecular networking: an analytical technology workflow for accelerated identification of multiple bioactive constituents in complex extracts. Anal Chem. 2023;95(9):4381–9. https://doi.org/10.1021/acs.analchem.2c04859 PMID: 36802535

19. Richardson SD. Mass spectrometry in environmental sciences. Chem Rev. 2001;101(2):211–54. https://doi.org/10.1021/cr990090u PMID: 11712246

20. Jones LM. Mass spectrometry-based methods for structural biology on a proteome-wide scale. Biochem Soc Trans. 2020;48(3):945–54. https://doi.org/10.1042/BST20190794 PMID: 32412054

21. Wang CR, Bubner ER, Jovcevski B, Mittal P, Pukala TL. Interrogating the higher order structures of snake venom proteins using an integrated mass spectrometric approach. J Proteomics. 2020;216:103680. https://doi.org/10.1016/j.jprot.2020.103680 PMID: 32028038

22. Li D, Ouyang Z, Ma X. Mass spectrometry imaging for single-cell or subcellular lipidomics: a review of recent advancements and future development. Molecules. 2023;28(6):2712. https://doi.org/10.3390/molecules28062712 PMID: 36985684

23. Buchberger AR, DeLaney K, Johnson J, Li L. Mass spectrometry imaging: a review of emerging advancements and future insights. Anal Chem. 2018;90(1):240–65. https://doi.org/10.1021/acs.analchem.7b04733 PMID: 29155564

24. Schänzer W, Thevis M. Human sports drug testing by mass spectrometry. Mass Spectrom Rev. 2017;36(1):16–46. https://doi.org/10.1002/mas.21479 PMID: 26213263

25. Thevis M, Schänzer W. Mass spectrometry in sports drug testing: structure characterization and analytical assays. Mass Spectrom Rev. 2007;26(1):79–107. https://doi.org/10.1002/mas.20107 PMID: 16888758

26. Brembs B, Huneman P, Schönbrodt F, Nilsonne G, Susi T, Siems R, et al. Replacing academic journals. R Soc Open Sci. 2023;10(7):230206. https://doi.org/10.1098/rsos.230206 PMID: 38107166

27. Chambers MC, Maclean B, Burke R, Amodei D, Ruderman DL, Neumann S, et al. A cross-platform toolkit for mass spectrometry and proteomics. Nat Biotechnol. 2012;30(10):918–20. https://doi.org/10.1038/nbt.2377 PMID: 23051804

28. 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. Anal Chem. 2006;78(3):779–87. https://doi.org/10.1021/ac051437y PMID: 16448051

29. Schmid R, Heuckeroth S, Korf A, Smirnov A, Myers O, Dyrlund TS, et al. Integrative analysis of multimodal mass spectrometry data in MZmine 3. Nat Biotechnol. 2023;41(4):447–9. https://doi.org/10.1038/s41587-023-01690-2 PMID: 36859716

30. Dührkop K, Fleischauer M, Ludwig M, Aksenov AA, Melnik AV, Meusel M, et al. SIRIUS 4: a rapid tool for turning tandem mass spectra into metabolite structure information. Nat Methods. 2019;16(4):299–302. https://doi.org/10.1038/s41592-019-0344-8 PMID: 30886413

31. Yi L, Dong N, Yun Y, Deng B, Ren D, Liu S, et al. Chemometric methods in data processing of mass spectrometry-based metabolomics: a review. Anal Chim Acta. 2016;914:17–34. https://doi.org/10.1016/j.aca.2016.02.001 PMID: 26965324

32. Kösters M, Leufken J, Schulze S, Sugimoto K, Klein J, Zahedi RP, et al. pymzML v2.0: introducing a highly compressed and seekable gzip format. Bioinformatics. 2018;34(14):2513–4. https://doi.org/10.1093/bioinformatics/bty046 PMID: 29394323

33. Goloborodko AA, Levitsky LI, Ivanov MV, Gorshkov MV. Pyteomics—a Python framework for exploratory data analysis and rapid software prototyping in proteomics. J Am Soc Mass Spectrom. 2013;24(2):301–4. https://doi.org/10.1007/s13361-012-0516-6 PMID: 23292976

34. Tsugawa H, Cajka T, Kind T, Ma Y, Higgins B, Ikeda K, et al. MS-DIAL: data-independent MS/MS deconvolution for comprehensive metabolome analysis. Nat Methods. 2015;12(6):523–6. https://doi.org/10.1038/nmeth.3393 PMID: 25938372

35. Sturm M, Bertsch A, Gröpl C, Hildebrandt A, Hussong R, Lange E, et al. OpenMS—an open-source software framework for mass spectrometry. BMC Bioinformatics. 2008;9:163. https://doi.org/10.1186/1471-2105-9-163 PMID: 18366760

36. Qt for Python. [cited 2024 Dec 22]. Available from: https://doc.qt.io/qtforpython-6.7/index.html

37. Deutsch EW. File formats commonly used in mass spectrometry proteomics. Mol Cell Proteomics. 2012;11(12):1612–21. https://doi.org/10.1074/mcp.R112.019695 PMID: 22956731

38. Deutsch EW. Mass spectrometer output file format mzML. Methods Mol Biol. 2010;604:319–31. https://doi.org/10.1007/978-1-60761-444-9_22 PMID: 20013381

39. PyQtGraph—pyqtgraph 0.14.0dev0 documentation. [cited 2024 Dec 22]. Available from: https://pyqtgraph.readthedocs.io/en/latest/

40. Wetter E. 2025. Available from: Https://biorender.com/90ag4vn

41. Ryan CG, Clayton E, Griffin WL, Sie SH, Cousens DR. SNIP, a statistics-sensitive background treatment for the quantitative analysis of PIXE spectra in geoscience applications. Nucl Instrum Methods Phys Res B. 1988;34(3):396–402. https://doi.org/10.1016/0168-583x(88)90063-8

42. Chure G, Cremer J. hplc-py: a python utility for rapid quantification of complex chemical chromatograms. JOSS. 2024;9(94):6270. https://doi.org/10.21105/joss.06270

43. Hampton CV, Lian B, McHarris WmC. Fast-Fourier-transform spectral enhancement techniques for γ-ray spectroscopy. Nucl Instrum Methods Phys Res A. 1994;353(1–3):280–4. https://doi.org/10.1016/0168-9002(94)91657-8

44. Staerk D. Spectral data and metadata for the manuscript: “Polypharmacology-labelled molecular networking for discovery of hypoglycemic and antibacterial diterpenoids in the Australian desert plant Eremophila rugosa”. 2021. https://doi.org/10.25345/C5M58V

45. Capistran BA, Sisco E. Data to support “rapid GC-MS as a screening tool for forensic fire debris analysis”. 2022. https://doi.org/10.18434/MDS2-2601

46. Liebler DC, Zimmerman LJ. Targeted quantitation of proteins by mass spectrometry. Biochemistry. 2013;52(22):3797–806. https://doi.org/10.1021/bi400110b PMID: 23517332

47. Alseekh S, Aharoni A, Brotman Y, Contrepois K, D’Auria J, Ewald J, et al. Mass spectrometry-based metabolomics: a guide for annotation, quantification and best reporting practices. Nat Methods. 2021;18(7):747–56. https://doi.org/10.1038/s41592-021-01197-1 PMID: 34239102

48. Pan S, Aebersold R, Chen R, Rush J, Goodlett DR, McIntosh MW, et al. Mass spectrometry based targeted protein quantification: methods and applications. J Proteome Res. 2009;8(2):787–97. https://doi.org/10.1021/pr800538n PMID: 19105742

49. Odenwald MA, Lin H, Lehmann C, Dylla NP, Cole CG, Mostad JD, et al. Bifidobacteria metabolize lactulose to optimize gut metabolites and prevent systemic infection in patients with liver disease. Nat Microbiol. 2023;8(11):2033–49. https://doi.org/10.1038/s41564-023-01493-w PMID: 37845315

50. Konermann L, Ahadi E, Rodriguez AD, Vahidi S. Unraveling the mechanism of electrospray ionization. Anal Chem. 2013;85(1):2–9. https://doi.org/10.1021/ac302789c PMID: 23134552

51. Kebarle P, Verkerk UH. Electrospray: from ions in solution to ions in the gas phase, what we know now. Mass Spectrom Rev. 2009;28(6):898–917. https://doi.org/10.1002/mas.20247 PMID: 19551695

52. Wu C, Siems WF, Hill HH Jr. Secondary electrospray ionization ion mobility spectrometry/mass spectrometry of illicit drugs. Anal Chem. 2000;72(2):396–403. https://doi.org/10.1021/ac9907235 PMID: 10658336

53. Fido M, Hersberger S, Güntner AT, Zenobi R, Giannoukos S. Systematic study of polymer gas sampling bags for offline analysis of exhaled breath. J Breath Res. 2024;18(4):10.1088/1752-7163/ad6a31. https://doi.org/10.1088/1752-7163/ad6a31 PMID: 39089291

54. Systematic study of polymer gas sampling bags for offline analysis of exhaled breath. [cited 2025 Sep 11]. Available from: https://www.research-collection.ethz.ch/entities/researchdata/6d805a9d-d928-481b-8095-6a34cc686f2b

55. Nuitka Compiler for Python. 2025 [cited 2025 Aug 29]. Available from: https://github.com/Nuitka/Nuitka

56. Myers OD, Sumner SJ, Li S, Barnes S, Du X. Detailed investigation and comparison of the XCMS and MZmine 2 chromatogram construction and chromatographic peak detection methods for preprocessing mass spectrometry metabolomics data. Anal Chem. 2017;89(17):8689–95. https://doi.org/10.1021/acs.analchem.7b01069 PMID: 28752757