PLOS COMPUTATIONAL BIOLOGY

SOFTWARE

Common data models to streamline metabolomics processing and annotation, and implementation in a Python pipeline

Joshua M. Mitchell[1], **Yuanye Chi**[1], **Maheshwor Thapa**[1], **Zhiqiang Pang**[2], **Jianguo Xia**[2], **Shuzhao Li**[1,3]*

1 The Jackson Laboratory for Genomic Medicine, Farmington, Connecticut, United States of America, **2** Institute of Parasitology, McGill University, Montreal, Quebec, Canada, **3** University of Connecticut School of Medicine, Farmington, Connecticut, United States of America

* shuzhao.li@jax.org

OPEN ACCESS

Citation: Mitchell JM, Chi Y, Thapa M, Pang Z, Xia J, Li S (2024) Common data models to streamline metabolomics processing and annotation, and implementation in a Python pipeline. PLoS Comput Biol 20(6): e1011912. https://doi.org/10.1371/journal.pcbi.1011912

Editor: Sunil Laxman, Institute for Stem Cell Science and Regenerative Medicine, INDIA

Received: February 12, 2024

Accepted: May 20, 2024

Published: June 6, 2024

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.1011912

Data Availability Statement: The version of the pcpfm, notebooks, and previously unreleased datasets used to generate the results presented in this manuscript are available at https://doi.org/10.

Abstract

To standardize metabolomics data analysis and facilitate future computational developments, it is essential to have a set of well-defined templates for common data structures. Here we describe a collection of data structures involved in metabolomics data processing and illustrate how they are utilized in a full-featured Python-centric pipeline. We demonstrate the performance of the pipeline, and the details in annotation and quality control using large-scale LC-MS metabolomics and lipidomics data and LC-MS/MS data. Multiple previously published datasets are also reanalyzed to showcase its utility in biological data analysis. This pipeline allows users to streamline data processing, quality control, annotation, and standardization in an efficient and transparent manner. This work fills a major gap in the Python ecosystem for computational metabolomics.

Introduction

Metabolomics aims to comprehensively detect, identify, and quantify the diverse small molecules, i.e., metabolites, present in biological systems. This provides key information on biochemical phenotypes, often reflecting the function of genes and genomes. With the progress of technologies, metabolomics is becoming a regular component of many biomedical projects [1,2,3,4]. Thousands of metabolomics datasets are now available in major data repositories [5,6,7]and the annual citation of "metabolomics" in PubMed now exceeds ten thousand. Due to this increasing popularity, solutions for processing such data need to be better incorporated into the regular bioinformatics workflows [8,9,10]. This integration will require an ecosystem in both the R and Python programming languages, the two dominant languages for bioinformatics, each with unique strengths and a large user community.

The foundational tool of a software ecosystem in computational metabolomics is the preprocessing tool that, among other functions, converts raw data into feature tables representing signals of interest likely to represent metabolites. XCMS [11] has served this role for the R programming language, and various tools for further data processing, including annotation,

5281/zenodo.10629957. The HZV029 HILIC+, RP- are uploaded as study ID: ST003109, Two-Phase HILIC as study ID: ST003075 and HZV029 QC was already available as study ID: ST002233. The Checkmate dataset was retrieved from Metabolomics Workbench (https://www.metabolomicsworkbench.org) with study ID: ST00127. Bowen 2023 dataset was retrieved from Metabolights (https://www.ebi.ac.uk/metabolights/, accession code MTBLS2746). The Ansone 2021 dataset was retrieved from Metabolights (accession code MTBLS3852).

Funding: • This work was supported by the National Institutes of Health (U01CA235493, R01AI149746, R01AI149746S1, UM1HG012651 to SL). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

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

quality assurance and quality control (QA/QC), have been built utilizing its outputs [12,13,14,15]. Many optimization tools and pipelines have been built around XCMS [16,17,18,19,20]. Despite the popularity of Python in machine learning and bioinformatics in general, a robust ecosystem for metabolomics in Python remains lacking, primarily due to the lack of a preprocessing tool for metabolomics raw data. While a handful of Python tools have been developed over the past decade [21,22], they are either dated or not production-ready. With the recent release of Asari [23], a preprocessing tool implemented in Python, Python has become a viable option for data processing in computational metabolomics.

As computational metabolomics evolves, the community continues working to define operational terminology and best practices. These efforts have resulted in various workgroups and multiple publications [24,25,26,27]. Since metabolomics analysis is often part of larger biomedical projects, there is an urgent need to standardize terminologies that cover sample preparation, experimental protocols, steps of software processing and metadata. While Asari fills a key gap in the computational metabolomics ecosystem, the fundamental issue of interoperable data structures remains a challenge. To standardize the computational aspects of metabolomics analysis and empower future computational developments, a set of common, well-defined, and reusable data structures will be essential, regardless of the programming language. This paper, therefore, describes a collection of common data structures involved in metabolomics data processing and illustrates how they are utilized in a full-featured Python-centric pipeline.

Design and implementation

Semi-automated data analysis pipelines are essential for the mainstream adoption of metabolomics and its continued growth in the biomedical sciences. With pipelines, researchers of diverse backgrounds can process their data quickly and meaningfully, allowing for higher throughput and more extensive experiments. Furthermore, pipelines allow researchers to define highly reproducible workflows that are repeatable and reproducible by others. Our pipeline, named the Python-centric pipeline for metabolomics (pcpfm), enables start-to-finish metabolomics data processing based on Asari. The pipeline ingests centroided mzML data or Thermo raw files and returns a human-readable set of tables summarizing the detected features and their annotations and sample metadata. Annotation is a major step after preprocessing, utilizing multiple sources, such as authentic compound libraries and tandem mass spectral libraries. Annotation levels in pcpfm are described in accordance with Schmanski 2014 [28]. Additionally, the pipeline performs various processing steps, including normalization, feature interpolation, removal of rare features, quality assurance, quality control evaluations, and generates PDF reports to summarize results.

We designed a set of core data models, which are described in the MetDataModel package and summarized in Table 1. The goal of MetDataModel is to encourage reuse and extension, therefore the data models are kept minimal. Developers are free to extend them to more detailed and specific models. Such extensions and applications are exemplified here in the pipeline package, pcpfm.

A mass spectrum typically consists of a list of m/z (mass to charge ratio) values and corresponding intensities. It can be from a full scan (MS^1) or tandem mass spectrometry (MS^2 and beyond). The mass spectrum can be in profile mode or centroid mode. In profile mode, the term "mass peak" is still used by some applications to refer to a group of m/z values that belong to the same ion species. Data in profile mode can be converted to centroid mode (mass peak picking) by software from the manufacturers or from scientific community, and usually done

Table 1. Core concepts implemented in the MetDataModel package.

Name	Operational Definition
MS Spectrum	List of m/z values and associated intensity, typically from a scan on a mass spectrometer
Mass Track	An extracted ion chromatogram of consensus m/z, spanning the full retention time.
Elution Peak	Peak of intensity values along the axis of chromatography.
Feature	A set of peaks that are aligned across samples, specific to an experiment.
Empirical Compound	A group of associated features, typically isotopes and adducts, that belong to the same tentative compound and co-elute if there is chromatography.
Compound	A metabolite or a chemical of xenobiotic origin, including contaminants.
Reaction	Biochemical process that interconverts one or more compounds, often catalyzed by an enzyme.
Enzyme	A protein that catalyzes a biochemical reaction.
Gene	An inheritable sequence of nucleotides, some of which code for proteins.
Metabolic Pathway	A series of linked reactions that typically involve structurally related compounds, usually defined by human knowledge.
Metabolic Network	A set of reactions connected by shared compounds. Mathematically identical to pathway, but not limited by pathway definition.
Metabolic Model	A collection of metabolic reactions and their associated metabolites, enzymes, and genes. Additional parameters, e.g. reaction rates and flux rates, can be included.
Study	A collection of experiments on a set of related samples.
Experiment	A set of acquisitions collected on a set of samples using consistent methods.
Method	The approach and parameters used for data collection in an experiment, e.g., chromatography and ionization parameters.
Sample	A biological sample or a control sample that is analyzed in a study. A sample can be analyzed in multiple experiments, by a single or multiple methods. An instance of data file generated by analyzing a sample is referred to as an acquisition. Analytical replicates need to be modeled explicitly if used.

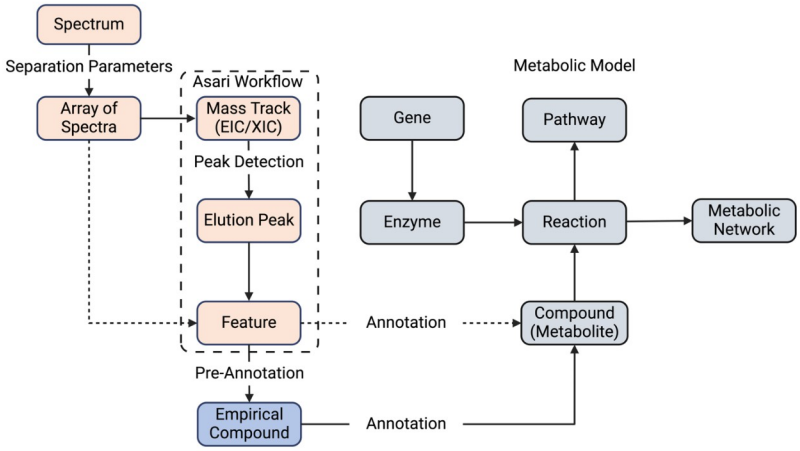
https://doi.org/10.1371/journal.pcbi.1011912.t001

by default in format conversion to the common mzML format. Centroided data is much reduced in size and there is little reason to use profile mode.

A mass spectrometer is often connected to chromatography (typically liquid phase or gas phase); therefore, such an experiment acquires many mass spectra at different chromatographic retention times. Thus, data processing requires the detection of signals across spectra, i.e., scans. Such signals are typically presented as an extracted ion chromatogram (EIC or XIC). In the Asari software, this concept of EIC is extended to a "mass track" [23], which is a vector of intensity values spanning the full scan range under one consensus m/z value. The use of mass tracks leads to new algorithms for alignment and feature detection [23]. Because "peak picking" or "peak detection" could refer to either mass peaks or elution peaks, we recommend the explicit term of elution peak detection. An elution peak is defined by ion intensity along the axis of retention time in the 2-dimensional representation. A mass peak is defined by ion intensity along the axis of m/z, usually in profile data. We define an elution peak at the level of a sample and as a feature at the level of an experiment (Table 1). The definition of "feature" here is consistent with its use in XCMS [11] and MZmine [29], but different from OpenMS [30]. OpenMS refers to a feature as a group of ions, likely due to its root in proteomics. The relationships between these concepts are illustrated in Fig 1A.

The relationship between metabolite, reaction, enzyme, gene, pathway, and network is described on right side of Fig 1A, which are collectively considered as a "metabolic model". Metabolic reactions are central to connect these entities, and the links to enzymes (proteins) and genes (measured in transcriptomics, genomics and epigenomics) are the most important basis for analyzing multi-omics data [31,32]. These concepts mirror the extensive development

A



B


```
"kp4_166.0488": {
   "interim_id": "kp4_166.0488",
   "neutral_formula_mass": 166.04884896677,
   "Database_referred": [
      "MoNA-export-LC-MS-MS_Negative_Mode.msp", "HMDBv5"],
   "identity": [["3-Methylxanthine", 0.98]],
   "MS1_pseudo_Spectra": [
      {
         "id_number": "F21",
         "mz": 165.0418,
         "rtime": 73.72,
         "isotope": "M0",
         "modification": "M-H-",
         "ion_relation": "M0,M-H-",
      },...],
   "MS2_Spectra": [
         {"ms_level": 2,
         "precursor_ion_mz": 165.041976928711,
         "list_mz": [ 55.02987289428711, ...],
         "list_intensity": [0.010449324526009295,...],
         "rtime": 69.007309953,
         "precursor_ion_id": "165.041976928711_69.007309953_plasma_ID_01.mzML",
         ...],
   "list_matches": [
      ["C6H6N4O2_166.049075",...],...
   ],
   "Level_4": [{
         "accession": "HMDB0001886",
         "name": "3-Methylxanthine",
         "chemical_formula": "C6H6N4O2",
         "primary_db": "HMDBv5"
      },...],
},
```


Fig 1. Design of core concepts and data models in computational metabolomics. A) The core concepts in MetDataModel with the metabolomics data processing in salmon and metabolic modeling in grey. We introduce "empirical compound" as a key bridge in between. The dashed lines indicate alternative workflows. Created with Biorender.com. **B)** Abridged empirical compound example including the listing of MS^1 features, annotation from MS^2 and other sources. This JSON format enables chaining of multiple annotation tools.

https://doi.org/10.1371/journal.pcbi.1011912.g001

from the field of genome-scale metabolic models (GEMs) in the previous two decades. Connecting GEMs with the experimental measurement by mass spectrometry is not trivial, because a) the identifiers of metabolites need to be consistent; b) charge states of molecules and experimental measurements need to be consistent; c) a significant knowledge gap exists

between the GEMs and experimental metabolomics; and d) metabolite identification is limited in experimental metabolomics.

The reality of metabolomics is that many features are not definitively identified. We have introduced the concept of empirical compound to describe the measurement of a tentative metabolite (Fig 1B). For example, in LC-MS (liquid chromatography coupled mass spectrometry) metabolomics, some isomers (molecules of identical mass) may not be resolved, limiting the annotation level. That is, the isotopologues and adducts clearly belong to the group, but the group may be isomer A, isomer B, or a mixture of both. Empirical compounds model this property and serve as an operational unit to link computational steps. It has been part of the software implementation since version 2 of mummichog and version 4 of MetaboAnalyst [33]. This design enables an organized presentation of degenerate MS^1 features, and chaining annotation from MS^n and multiple methods. The isotopes and adducts from pre-annotation are modeled as a grid structure, made computable by the khipu package [34], which is also incorporated in the pcpfm pipeline. Annotation remains the most critical step in the meaningful interpretation of metabolomics data and the field faces the challenge of handling annotation uncertainty and probability. Empirical compounds provide an operational data structure as a path forward.

The abstract concepts in Table 1 and Fig 1 are intrinsically agnostic to programming languages. We demonstrate their implementation in Python 3 and JSON in MetDataModel. The pcpfm package is written in Python 3 and JSON is used extensively for intermediary data. Many pipeline data structures inherit from, and expand upon, objects provided by the MetDataModel library. Specific extension of empirical compound is exemplified in Fig 1B.

The inputs to our pipeline minimally consist of .mzML or .raw files and a metadata CSV file, that minimally maps sample names to acquisition file paths. While Asari was initially developed for orbitrap data, pcpfm is expected to be compatible with the data from major manufacturers that can be converted into mzML format [35]. The final output consists of a feature table detailing the observed m/z and retention time values for observed features mapped to unique identifiers, an annotation table mapping these identifiers to annotations and metadata for those annotations, and a third table summarizing the acquisition and experiment-level metadata. This three-table format handles multiple annotations gracefully and will be supported in future versions of MetaboAnalyst and Mummichog for downstream analysis and interpretation.

Each step in an analysis corresponds to one command in the CLI and one function in the main pipeline process (S1 Table). In brief, every analysis starts with assembling an experiment object from the metadata and acquisition data. This experiment object records the location of intermediates on disk for reuse in later steps. Optionally, any.raw files are converted to centroided .mzML files using the ThermoRawFileParser [36] before preprocessing with Asari which yields a "preferred" and "full" feature table.

Quality control is necessary in every project but depends on the experimental design. Multiple QA/QC operations are available including PCA, t-SNE, correlation cluster maps, Z-scores that quantify the median pearson correlation to all other samples, the number of features or missing features per-sample are implemented as well as scatter plots that summarize median and mean feature intensities plus bar plots of TICs (total ion counts) at each step in the workflow are implemented. PCA and t-SNE are implemented as wrappers around scikit-learn functions [37], cluster maps use a combination of seaborn [38] for hierarchical clustering and either scipy [39] or numpy [40] for the calculation of the input correlation matrix. Z-scores and TICs are calculated using custom routines implemented using a mixture of pandas [41] and numpy. All plotting is based on matplotlib [42] except clustermaps which are using seaborn [38].

Operations to correct common data quality issues are provided including normalization, blank masking, batch correction, removal of uncommon features, and missing value imputation. Blank masking, missing value imputation, and feature removal is implemented using custom routines in Pandas and numpy, batch correction using a wrapper around pycombat [43], while normalization is implemented using numpy and can be performed in a one-pass or two-pass approach. In the two-pass approach, normalization is done within a batch of samples and then the batches are scaled to one another using their median TIC values based on the conserved features. Missing value imputation is implemented using a scalar multiple of the minimum observed intensity for that feature in the dataframe. Reasonable defaults for the user-provided parameters for these operations are implemented and described in S1 File while the ordering of steps is modifiable. For example, batch correction is sensitive to missing features and can be performed after removing frequently missing features or after missing value imputation; however the default workflow consisting of blank masking, outlier sample removal, normalization, outlier feature removal, imputation and filtering before annotation is recommended in that order to remove bias in each subsequent step. Outlier samples can be removed using any of the Z-scores or other 1-D metrics described above via a user-provided filter; however, by default, samples that have an absolute value for their number of features Z-score greater than 2.5 are dropped.

Empirical compounds are constructed from a feature table using Khipu [44] and most methods for empirical compounds concern annotation. Using MatchMS [45,46], MS^2 based annotations can be generated using data from DDA or deep scan workflows such as AcquireX [47,48] and MS^2 spectral databases such as MoNA [49] or authentic standards libraries. MS^1-based annotations are generated using our JSON metabolite services library and appropriately formatted inputs or m/z and retention time similarity to authentic standards. These annotations can be mapped back to any feature table to generate the previously mentioned tabular output. PDF reports can be created using the fpdf library [50]. The contents of the report can be defined by the end-user via a JSON template but by default include PCAs, log TICs, pearson correlation clustermaps, missing feature Z-score plots for each feature table, an accounting of all annotations and features explained for each set of empirical compounds created, a timestamp for the report generation, and a timeline of all commands used in the analysis. Example reports are provided in S2 and S3 Files.

Most operations in the pipeline are chainable meaning they can be performed in a user-specified order with outputs from previous iterations being used as inputs. This flexibility allows users to build custom workflows; however, example workflows are provided as .sh and nextflow scripts [51]. Nearly all parameters are user-configurable, but reasonable defaults are provided and documented, allowing the pipeline to be as hands-off or hands-on as the end user desires.

Results

The pcpfm is designed to prepare data for downstream data analysis, which can be performed by bioinformaticians or data scientists without a background in mass spectrometry. The major steps are shown in Fig 2A and a comparison of the provided functionality to other metabolomics data processing tools [19,20,52,53,54,55,56] is shown in S2 Table. Additionally, we demonstrate first the results on data processing, annotation, and quality control, then on biological applications. Seven metabolomics and one lipidomics datasets from four studies, three fully public [57,58,59] and one in-house, are used in these examples (details in S1 File).

A distinct advantage of pcpfm and Asari is the computational efficiency to process large datasets. The computational times are summarized on two high-resolution LC-MS datasets of

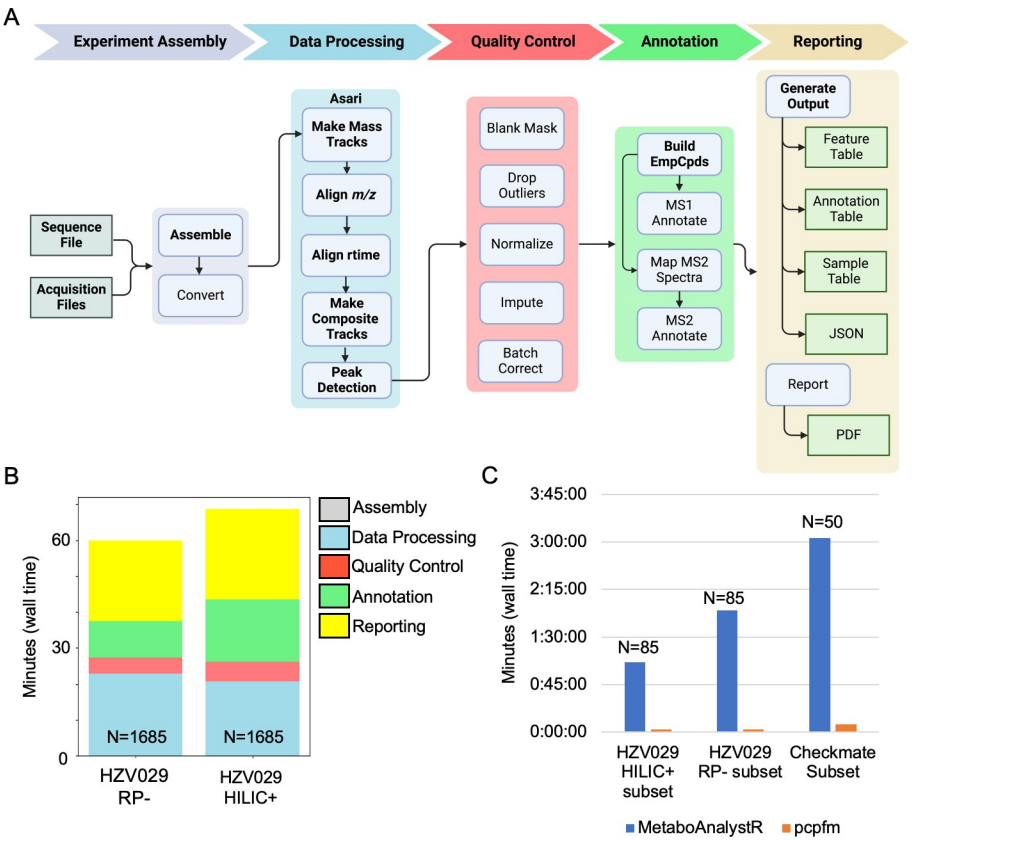


Fig 2. Design and computational performance of the pcpfm pipeline. A) The pipeline has five major sections: assembly, data processing, quality control, annotation and reporting. assembly creates the on-disk data structures needed for pcpfm analysis and optionally performs conversion to mzML. Data processing encapsulates everything from the start of a processing job to the creation of a feature table using Asari. Quality control consists of multiple chainable commands that allows for a raw feature table to be curated into a table suitable for downstream analysis. Annotation concerns the mapping of empirical compounds to metabolites using formula or MS^2 similarity to databases, m/z and retention time mapping to authentic standards and optionally, MS^2 similarity. Finally, reporting handles the creation of the three-table format for downstream analysis, PDF report generation, and JSON outputs for advanced users. Squares represent inputs and outputs, arrows represent dependencies between any steps, while bolded sections collectively represent a minimal workflow. Created with BioRender.com. **B)** Using the two largest datasets (N is the number of MS^1-only acquisitions), the high computational performance of our pipeline is demonstrated. Most of the wall time is spent during reporting. All steps are single threaded by default except Asari which uses 4 processes. In the HILIC+ and RP- datasets, 40008 and 32086 features are detected (full asari table including non-study samples) corresponding to 27851 and 23400 empirical compounds of which 16431 and 11962 received a level 4 annotation and 614 and 267 received a level 2 annotation. **C)** A comparison of the wall time required for a minimal pcpfm workflow (Asari+Khipu) compared to its MetaboAnalystR v4.0.0 equivalent on subsets of three studies where N is the number of MS^1-only acquisitions included in each subset. For the CheckMate subset, 3902 and 8907 features were detected by the MetaboAnalystR and PCPFM minimal workflows respectively while in HZV029 HILIC+ and RP- MetaboAnalystR workflow detects 2835 and 5966 features while the pcpfm workflow detects 12142 and 9939 respectively. All pcpfm counts are for the preferred feature table.

https://doi.org/10.1371/journal.pcbi.1011912.g002

1685 samples. Processing and QC use less than half an hour on a laptop computer for each datasets, while the annotation step depends on the databases involved while report generation depends on the number of samples and intermediate tables selected for figure generation (Fig 2B). The computational performance of the pre-processing and pre-annotation was assessed by comparing a minimal pcpfm workflow consisting of pre-processing and pre-annotation only using asari and khipu respectively versus a MetaboAnalystR v4.0.0-based pipeline (using OptiLCMS v1.1.0's implementation of XCMS, and CAMERA [13], details in S1 File) on

a subset of three datasets (Fig 2C), showing a clear improvement in the performance of our pipeline.

The metabolomics community have a consensus that metabolite annotation should be reported according to its confidence level. We have incorporated empirical compounds into both MS^1 and MS^2 annotations. By building empirical compounds first, i.e. pre-annotation via the khipu package, MS^1 annotation is improved because the search of databases does not query many degenerate features (Fig 3A). The MS^2 annotation utilizes MatchMS but with an optimization using an interval tree algorithm [60]. Because there are many implementations of MS^2 annotation under similar principles, it is important to be explicit on the algorithm in pcpfm (Fig 3B). The MS^2 annotation in pcpfm is efficient enough to run large experiments on consumer-grade hardware, as shown in Fig 2B. When authentic compounds are used to annotate metabolites, it is straight forward to match their m/z and retention time to biological samples (Fig 3C). Multiple annotations of different sources are chained in the empirical compound data structure (Fig 1B), which is amendable to future enhancements, e.g., context specific databases.

We compared the MS^2 annotations generated by the pcpfm to those from vendor's software, Compound Discoverer (CD) [48]. Full details for the annotation procedures in both softwares are provided in S1 File; however, CD annotations did require an additional step to map the generated annotations to the Asari feature tables which used an m/z tolerance of 10 ppm and a retention time tolerance of 30 seconds. For both pcpfm and CD annotations sets of annotated features were constructed by concatenating the annotated compound name with the asari feature (e.g., Caffeine_F1345) and these annotation sets then compared using set operations in Python. Considerable overlap is seen between CD and pcpfm annotations (Fig 3D). Because the algorithm in CD is closed source, it is not feasible to trace the differences between the tools, which highlights the importance of open-source tools for continued improvement.

The applications of pcpfm to quality control are demonstrated on a dataset consisting of 17 batches and 1685 samples (Fig 4). This analysis was performed using a batch-correction variation of our default workflow (S1 File). First, the QC metrics generated by Asari are summarized using kernel density plots to illustrate the high quality of features yielded by asari as evidenced by their high cSelectivity, peak shape (i.e., goodness of fit to a gaussian), their peak areas and high signal-to-noise ratio (Fig 4A). Next, hierarchical clustering of the inter-sample pearson correlation across all features was performed revealing two clusters of samples, representing a clear batch effect that was traced back to a recalibration of the instrument after batch 8 (Fig 4B). The log TICs of a random subset of samples and PCA plots were generated to further investigate these batches and identify abnormal samples (Fig 4C and 4D). The standard two-pass normalization does not adaquetly correct the batch effect; however, after normalization and batch correction via pycombat, both the log TICs and PCA show more consistency and no sub-clustering, suggesting the batch effect was largely mitigated.

Another common data quality issue addressed by the pcpfm are failed injections. Using the per-sample number of features Z-score, we demonstrate the ability to detect failed injections automatically in two datasets (Fig 4E). Failed injections are readily identified by their anomalously low Z-score (red) compared to successful injections (black). When the failed injections are compared to the preceeding successful injection, the absence of clear signal is appreciated in their TICs, confirming they were failed injections (Fig 4E). These results motiviated the inclusion of this metric and a default cutoff of $|Z| > 2.5$ for the removal of outliers by default in the pcpfm.

Multiple previously published datasets were reanalyzed using pcpfm to evaluate the pipeline's general suitability. These analyses were performed using either a modified default

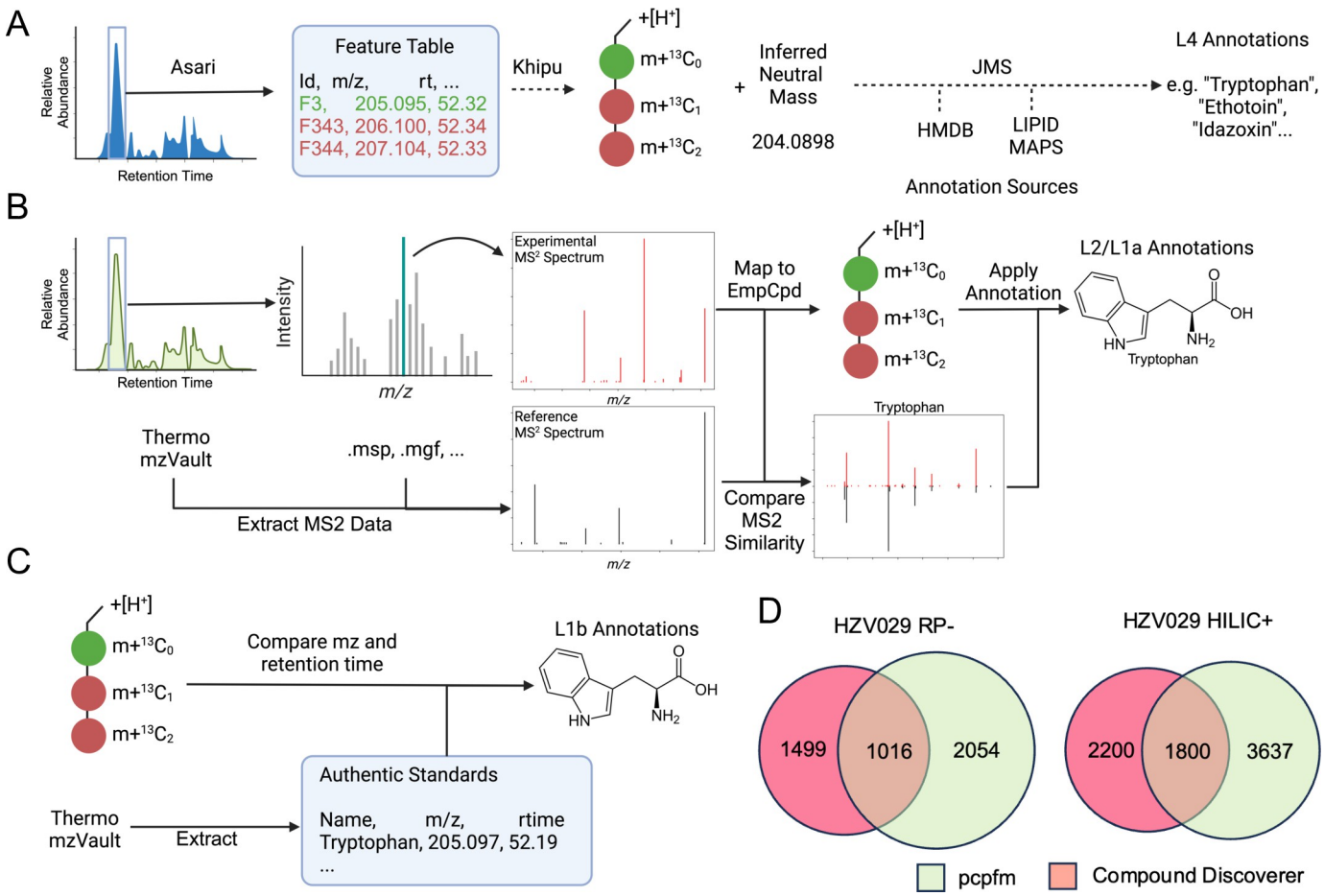


Fig 3. Annotation methods in pcpfm. A) Empirical compounds are constructed from Asari feature tables using khipu, which groups degenerate features such as isotopologues and adducts. The inferred neutral mass of an empirical compound is compared to known metabolites to generate level 4 annotations (via JMS, https://github.com/shuzhao-li-lab/JMS). Panels A, B, and C created with BioRender.com. **B)** Level 2 and 1a annotations are generated using MS^2 similarity. Experimental MS^2 spectra are mapped to empirical compounds and then compared to reference spectra, to annotate metabolite structures. **C)** Level 1b annotations are generated based on m/z and retention time match to authentic chemical standards. The use of empirical compound improves search efficiency and reduces false positives, while annotations at all levels can also be mapped to the feature level. **D)** Overlap of MS^2 annotations by pcpfm and CD in the two HZV029 plasma datasets. Detailed dissection of the differences is difficult since CD is closed-source.

https://doi.org/10.1371/journal.pcbi.1011912.g003

workflow or, in the case of the CheckMate analysis, a minimal workflow as described in S1 File. Bowen et al (2023 [57]) designed a specialized xenobiotic-focused workflows to detect metabolites of the drug sunitinib. Our pipeline with default parameters detects all but one of the previously reported sunitinib-related metabolites in cardiomyocyte cell pellets and all features in culture media (Fig 5A) based on a 5 ppm m/z tolerance and 10 second retention time tolerance to the features reported in Bowen et al 2023. The sole missing feature is due to low signal-to-noise ratio, not passing Asari quality threshold (Fig 5B). Using ANOVA followed by Benjamini-Hochberg correction [61], sunitinib-treated and control cell pellets were compared and hierarchical clustering performed using the significant features (Fig 5C). This yields two distinct clusters corresponding to the treated and control samples consistent with an induced metabolic response resulting from sunitnib exposure as reported in the original analysis [57]. These results indicate the potential of pcpfm as a simplified yet broadly applicable workflow.

To compare pcpfm feature detection against a state-of-art R-based pipeline (MetaboAnalystR v4.0.0), we reprocessed a subset of published metabolomics data on the CheckMate

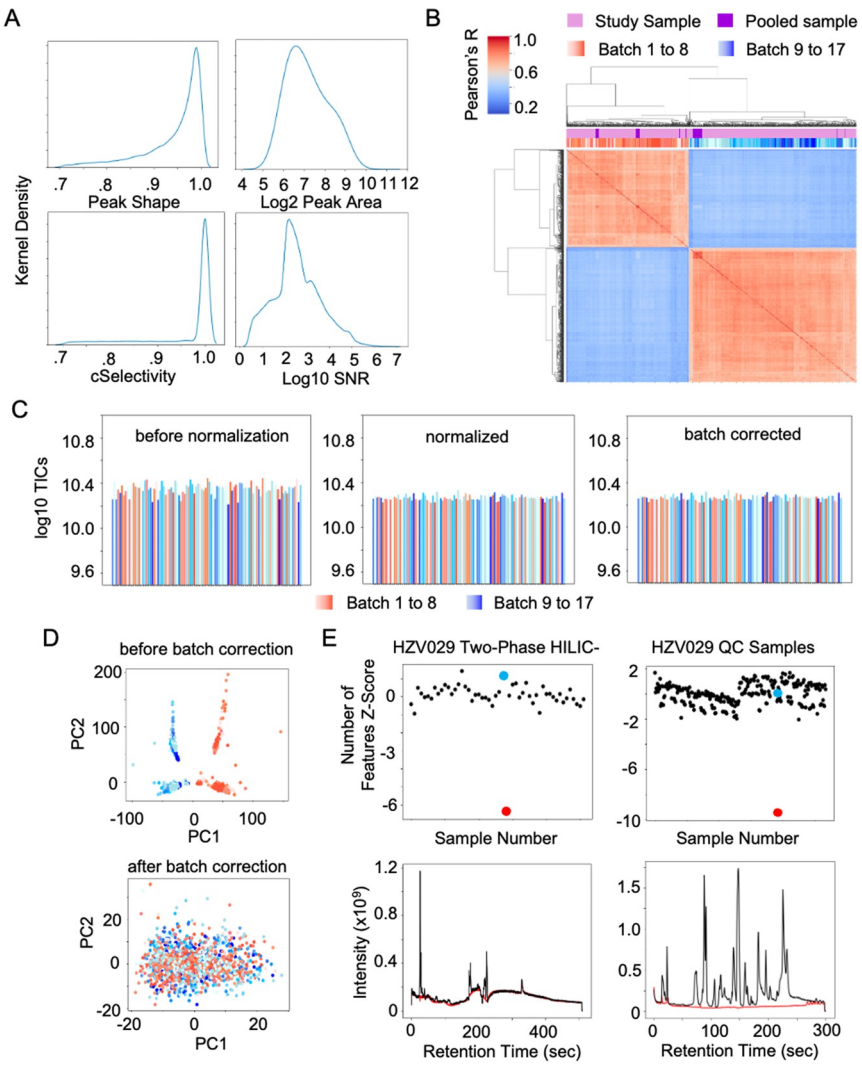


Fig 4. Examples of quality control in the pcpfm pipeline. A) A collection of QA/QC metrics generated by Asari on an example dataset ("HZV029 Plasma RP-"). **B)** The correlation clustermap of all study samples and pooled samples from the HZV029 Plasma RP- dataset (preferred feature table) illustrating the batch effect induced by instrument calibration. **C)** Log10 TICs of a random subset of samples before normalization, after normalization, and after batch correction. **D)** PCA demonstrating the presence of a batch effect (top) and its removal (bottom). **E)** Detection of failed acquisition by the number of feature Z-scores. The failed injection is highlighted in red and a representative "good" injection in blue for both the plasma HZV029 Two-Phase HILIC- and HZV029 QC dataset (left and right, top). The two-phase failed injection is simulated by replacing a missing sample with an empty vial while the other was identified post-hoc. The TICs of the failed and good injections are shown in red and black respectively (bottom).

https://doi.org/10.1371/journal.pcbi.1011912.g004

immunotherapy cohort [58] using comparable minimal workflows as previously described (additional details in S1 File). The authors' in-house metabolite library serves as a proxy of ground truth here. An m/z tolerance of 5 ppm and an RT tolerance of 5 seconds was used to identify library features in the MetaboanalystR and pcpfm results.The sets of identified features were then compared to ground truth using the set logic operations in Python. The pcpfm consistently detects more features representing more true metabolites than the MetaboanalystR workflow, however, considerable overlap is observed (Fig 5D).

Lastly, as an example for generating biologically meaningful results, we reanalyzed the metabolomics data from a COVID-19 exposure and recovery cohort (Ansone 2021, [59]).

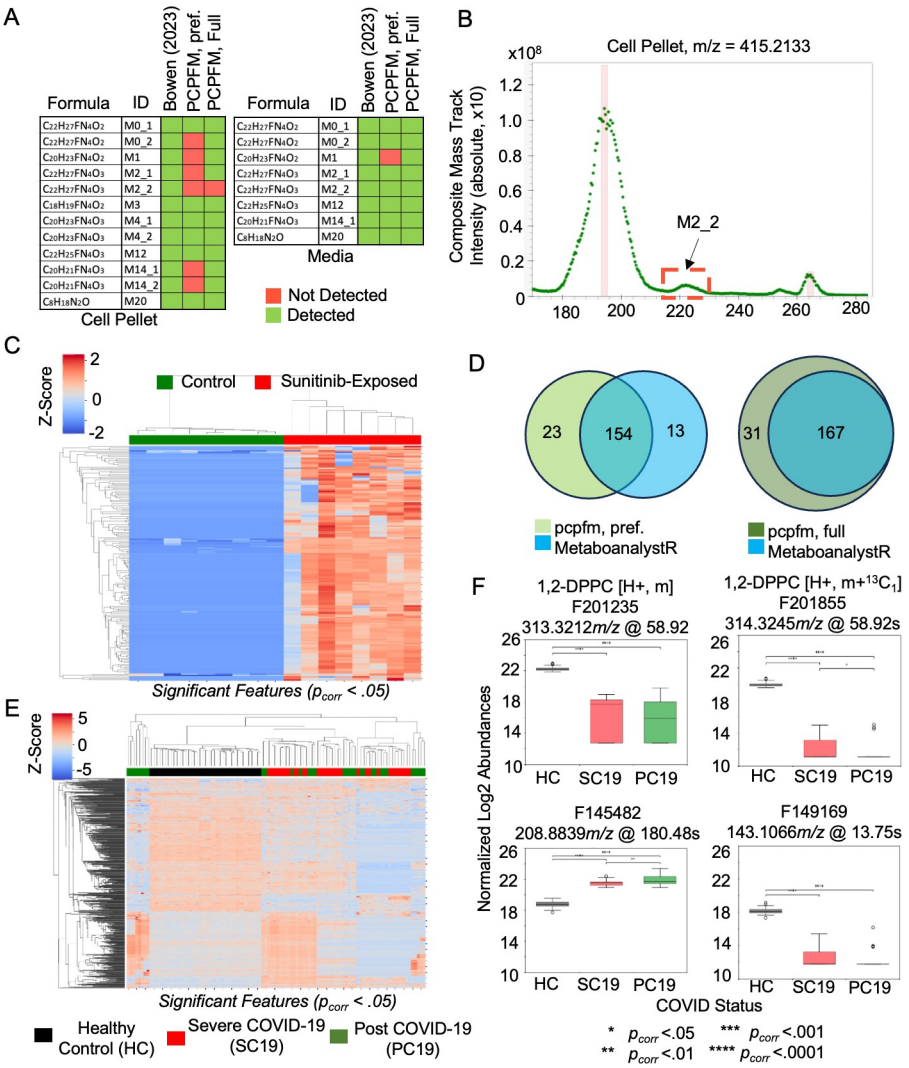


Fig 5. Applications of pcpfm to analyzing biological datasets. A) In the Bowen 2023 cardiomyocyte dataset, the pcpfm identifies most of the reported sunitinib-related features in both cell pellets and media using a standard workflow. Asari and pcpfm output both a preferred feature table and a full feature table, the former of higher feature quality and the latter more inclusive. **B)** The mass track for the sole feature undetected in the Bowen 2023 cell dataset is shown and the suspected undetected peak is in red box (M2_2), which fails to pass Asari's quality requirement. **C)** Significant differential metabolite features between sunitinib exposure groups in cell pellets. ANOVA p-values are corrected for multiple testing by Benjamini-Hochberg method. **D)** Both the pcpfm and MetaboAnalystR were used to extract features from a subset of the CheckMate study. Of 202 compounds in their authentic standard library, MetaboAnalystR identified 167, while the full table from the pcpfm identified 198 of the confirmed features. **E)** Clustering pattern of the Ansone 2021 cohort using features differentially abundant between treatment groups. **F)** Example boxplots of differentially abundant features in the Ansone 2021 cohort. F201235 and F201855 (top) were mapped to the same empirical compound that was tentatively annotated as 1,2-DPPC, a pulmonary surfactant by its sole level 4 annotation. Significance was evaluated using ANOVA and post-hoc Tukey's HSD test in E and F.

https://doi.org/10.1371/journal.pcbi.1011912.g005

Following pcpfm, the significant features tested by ANOVA followed by a Tukey's HSD [62] were subjected to hierarchical clustering (Fig 5E), which recapitulated the original observation that metabolic profiles cluster by COVID infection and recovery vs. control in the Anosne 2021 paper. The box plots of selected features confirm the patterns of abundance changes in participant groups (Fig 5F). Interestingly, two features (Fig 5F, top) are found to belong to an empirical compound with a single level 4 annotation to 1,2-dipalmitoylphosphatidylcholine

(1,2-DPPT), a pulmonary surfactant known to be less abundant in COVID patients than healthy controls [63]. This result demonstrates that novel biology can be gained with the pcpfm. The Jupyter notebooks and workflows underlying these examples are included in the pcpfm code repository, so that users can easily perform their own data analysis based on the templates.

Availability and Future Directions

The MetDataModel and pcpfm are available through GitHub (https://github.com/shuzhao-li-lab/metDataModel and https://github.com/shuzhao-li-lab/PythonCentricPipelineForMetabolomics), and both are installable by pip via PyPi or from source. All dependencies are open source and downloadable via pip, except for the ThermoRawFileConverter and mono framework, both of which are optional. Example workflows are provided in bash and as nextflow; however, users can implement their own using the CLI or the pipeline internals available using standard Python conventions for APIs. API usage will be officially supported in an upcoming release.

Future development of pcpfm will implement additional options and methods for data processing, including normalization, interpolation, and batch correction. Improving support for non-orbitrap instruments is another priority for the pipeline and the underlying Asari algorithm. A cloud-based application is planned to allow users to process data in a friendly web interface.

Supporting information

S1 Table. List of commands in the pcpfm pipeline, their inputs and outputs, and if they are chainable.
(XLSX)

S2 Table. Comparison of pcpfm features to other metabolomics data processing tools.
(XLSX)

S1 File. Description of datasets, methods for generating previously unpublished datasets, and compound discoverer annotation workflow.
(PDF)

S2 File. HZV029 Plasma HILIC+ example PDF report.
(PDF)

S3 File. HZV029 Plasma RP- example PDF report.
(PDF)

S4 File. zip of pcpfm v1.0.13.
(ZIP)

S5 File. zip of MetDataModel v0.6.1.
(ZIP)

Acknowledgments

We would like to thank Paul Robson, Arti Taggar, Julianna Alcoforado Diniz, Zukai Liu, and Lucas Chang who graciously provided data for the initial development and testing of the pipeline.

Author Contributions

Conceptualization: Shuzhao Li.

Data curation: Maheshwor Thapa.

Formal analysis: Joshua M. Mitchell.

Funding acquisition: Shuzhao Li.

Investigation: Joshua M. Mitchell, Yuanye Chi, Shuzhao Li.

Methodology: Joshua M. Mitchell, Maheshwor Thapa, Zhiqiang Pang, Shuzhao Li.

Software: Joshua M. Mitchell, Yuanye Chi, Shuzhao Li.

Supervision: Jianguo Xia, Shuzhao Li.

Validation: Yuanye Chi, Zhiqiang Pang.

Visualization: Joshua M. Mitchell.

Writing – original draft: Joshua M. Mitchell.

Writing – review & editing: Joshua M. Mitchell, Zhiqiang Pang, Jianguo Xia, Shuzhao Li.

References

1. Barnes S. Overview of Experimental Methods and Study Design in Metabolomics, and Statistical and Pathway Considerations. Methods Mol Biol. 2020; 2104:1–10. Epub 2020/01/19. https://doi.org/10.1007/978-1-0716-0239-3_1 PMID: 31953809.
2. McGarrah RW, Crown SB, Zhang GF, Shah SH, Newgard CB. Cardiovascular Metabolomics. Circ Res. 2018; 122(9):1238–58. Epub 2018/04/28. https://doi.org/10.1161/CIRCRESAHA.117.311002 PMID: 29700070; PubMed Central PMCID: PMC6029726.
3. Rinschen MM, Ivanisevic J, Giera M, Siuzdak G. Identification of bioactive metabolites using activity metabolomics. Nature Reviews Molecular Cell Biology. 2019; 20(6):353–67. https://doi.org/10.1038/s41580-019-0108-4 PMID: 30814649
4. Fuller H, Zhu Y, Nicholas J, Chatelaine HA, Drzymalla EM, Sarvestani AK, et al. Metabolomic epidemiology offers insights into disease aetiology. Nat Metab. 2023; 5(10):1656–72. Epub 2023/10/24. https://doi.org/10.1038/s42255-023-00903-x PMID: 37872285.
5. Sud M, Fahy E, Cotter D, Azam K, Vadivelu I, Burant C, et al. Metabolomics Workbench: An international repository for metabolomics data and metadata, metabolite standards, protocols, tutorials and training, and analysis tools. Nucleic Acids Res. 2016; 44(D1):D463–70. Epub 2015/10/16. https://doi.org/10.1093/nar/gkv1042 PMID: 26467476; PubMed Central PMCID: PMC4702780.
6. Haug K, Cochrane K, Nainala VC, Williams M, Chang J, Jayaseelan KV, et al. MetaboLights: a resource evolving in response to the needs of its scientific community. Nucleic Acids Research. 2019; 48(D1): D440–D4. https://doi.org/10.1093/nar/gkz1019 PMID: 31691833
7. Choi M, Carver J, Chiva C, Tzouros M, Huang T, Tsai TH, et al. MassIVE.quant: a community resource of quantitative mass spectrometry-based proteomics datasets. Nat Methods. 2020; 17(10):981–4. Epub 2020/09/16. https://doi.org/10.1038/s41592-020-0955-0 PMID: 32929271; PubMed Central PMCID: PMC7541731.
8. Ebbels TMD, van der Hooft JJJ, Chatelaine H, Broeckling C, Zamboni N, Hassoun S, et al. Recent advances in mass spectrometry-based computational metabolomics. Curr Opin Chem Biol. 2023; 74:102288. Epub 2023/03/27. https://doi.org/10.1016/j.cbpa.2023.102288 PMID: 36966702.
9. Gardinassi LG, Xia J, Safo SE, Li S. Bioinformatics tools for the interpretation of metabolomics data. Current Pharmacology Reports. 2017; 3:374–83.
10. Pittard WS, Villaveces CK, Li S. A Bioinformatics Primer to Data Science, with Examples for Metabolomics. Computational Methods and Data Analysis for Metabolomics. 2020:245–63. https://doi.org/10.1007/978-1-0716-0239-3_14 PMID: 31953822
11. 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; 78(3):779–87. https://doi.org/10.1021/ac051437y PMID: 16448051

12. Mahieu NG, Genenbacher JL, Patti GJ. A roadmap for the XCMS family of software solutions in metabolomics. Curr Opin Chem Biol. 2016; 30:87–93. Epub 2015/12/18. https://doi.org/10.1016/j.cbpa.2015.11.009 PMID: 26673825; PubMed Central PMCID: PMC4831061.

13. Kuhl C, Tautenhahn R, Böttcher C, Larson TR, Neumann S. CAMERA: an integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets. Anal Chem. 2012; 84(1):283–9. Epub 2011/11/25. https://doi.org/10.1021/ac202450g PMID: 22111785; PubMed Central PMCID: PMC3658281.

14. Tautenhahn R, Patti GJ, Kalisiak E, Miyamoto T, Schmidt M, Lo FY, et al. metaXCMS: Second-Order Analysis of Untargeted Metabolomics Data. Analytical Chemistry. 2011; 83(3):696–700. https://doi.org/10.1021/ac102980g PMID: 21174458

15. Tautenhahn R, Patti GJ, Rinehart D, Siuzdak G. XCMS Online: a web-based platform to process untargeted metabolomic data. Anal Chem. 2012; 84(11):5035–9. Epub 2012/04/27. https://doi.org/10.1021/ac300698c PMID: 22533540; PubMed Central PMCID: PMC3703953.

16. Delabriere A, Warmer P, Brennsteiner V, Zamboni N. SLAW: A Scalable and Self-Optimizing Processing Workflow for Untargeted LC-MS. Analytical Chemistry. 2021; 93(45):15024–32. https://doi.org/10.1021/acs.analchem.1c02687 PMID: 34735114

17. Manier SK, Keller A, Meyer MR. Automated optimization of XCMS parameters for improved peak picking of liquid chromatography-mass spectrometry data using the coefficient of variation and parameter sweeping for untargeted metabolomics. Drug Test Anal. 2019; 11(6):752–61. Epub 2018/11/28. https://doi.org/10.1002/dta.2552 PMID: 30479047.

18. McLean C, Kujawinski EB. AutoTuner: High Fidelity and Robust Parameter Selection for Metabolomics Data Processing. Analytical Chemistry. 2020; 92(8):5724–32. https://doi.org/10.1021/acs.analchem.9b04804 PMID: 32212641

19. Pang Z, Chong J, Li S, Xia J. MetaboAnalystR 3.0: Toward an Optimized Workflow for Global Metabolomics. Metabolites. 2020; 10(5). Epub 2020/05/13. https://doi.org/10.3390/metabo10050186 PMID: 32392884; PubMed Central PMCID: PMC7281575.

20. Shen X, Yan H, Wang C, Gao P, Johnson CH, Snyder MP. TidyMass an object-oriented reproducible analysis framework for LC–MS data. Nature Communications. 2022; 13(1):4365. https://doi.org/10.1038/s41467-022-32155-w PMID: 35902589

21. Kiefer P, Schmitt U, Müller JE, Hartl J, Meyer F, Ryffel F, et al. DynaMet: a fully automated pipeline for dynamic LC-MS data. Anal Chem. 2015; 87(19):9679–86. Epub 2015/09/15. https://doi.org/10.1021/acs.analchem.5b01660 PMID: 26366644.

22. Riquelme G, Zabalegui N, Marchi P, Jones CM, Monge ME. A Python-Based Pipeline for Preprocessing LC-MS Data for Untargeted Metabolomics Workflows. Metabolites. 2020; 10(10). Epub 2020/10/22. https://doi.org/10.3390/metabo10100416 PMID: 33081373; PubMed Central PMCID: PMC7602939.

23. Li S, Siddiqa A, Thapa M, Chi Y, Zheng S. Trackable and scalable LC-MS metabolomics data processing using asari. Nature Communications. 2023; 14(1):4113. https://doi.org/10.1038/s41467-023-39889-1 PMID: 37433854

24. Villas-Bôas SG, Rasmussen S, Lane GA. Metabolomics or metabolite profiles? Trends in Biotechnology. 2005; 23(8):385–6. https://doi.org/10.1016/j.tibtech.2005.05.009 PMID: 15939497

25. Kirwan JA, Gika H, Beger RD, Bearden D, Dunn WB, Goodacre R, et al. Quality assurance and quality control reporting in untargeted metabolic phenotyping: mQACC recommendations for analytical quality management. Metabolomics. 2022; 18(9):70. Epub 2022/08/28. https://doi.org/10.1007/s11306-022-01926-3 PMID: 36029375; PubMed Central PMCID: PMC9420093.

26. Dunn WB, Kuligowski J, Lewis M, Mosley JD, Schock T, Ulmer Holland C, et al. Metabolomics 2022 workshop report: state of QA/QC best practices in LC–MS-based untargeted metabolomics, informed through mQACC community engagement initiatives. Metabolomics. 2023; 19(11):93. https://doi.org/10.1007/s11306-023-02060-4 PMID: 37940740

27. de Jonge NF, Mildau K, Meijer D, Louwen JJR, Bueschl C, Huber F, et al. Good practices and recommendations for using and benchmarking computational metabolomics metabolite annotation tools. Metabolomics. 2022; 18(12):103. Epub 2022/12/06. https://doi.org/10.1007/s11306-022-01963-y PMID: 36469190; PubMed Central PMCID: PMC9722809.

28. Schymanski EL, Jeon J, Gulde R, Fenner K, Ruff M, Singer HP, et al. Identifying small molecules via high resolution mass spectrometry: communicating confidence. Environ Sci Technol. 2014; 48(4):2097–8. Epub 2014/01/31. https://doi.org/10.1021/es5002105 PMID: 24476540.

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. Nature Biotechnology. 2023; 41(4):447–9. https://doi.org/10.1038/s41587-023-01690-2 PMID: 36859716

30. Röst HL, Sachsenberg T, Aiche S, Bielow C, Weisser H, Aicheler F, et al. OpenMS: a flexible open-source software platform for mass spectrometry data analysis. Nature Methods. 2016; 13(9):741–8. https://doi.org/10.1038/nmeth.3959 PMID: 27575624

31. Baysoy A, Bai Z, Satija R, Fan R. The technological landscape and applications of single-cell multi-omics. Nature Reviews Molecular Cell Biology. 2023; 24(10):695–713. https://doi.org/10.1038/s41580-023-00615-w PMID: 37280296

32. Subramanian I, Verma S, Kumar S, Jere A, Anamika K. Multi-omics Data Integration, Interpretation, and Its Application. Bioinform Biol Insights. 2020; 14:1177932219899051. Epub 2020/02/23. https://doi.org/10.1177/1177932219899051 PMID: 32076369; PubMed Central PMCID: PMC7003173.

33. Chong J, Soufan O, Li C, Caraus I, Li S, Bourque G, et al. MetaboAnalyst 4.0: towards more transparent and integrative metabolomics analysis. Nucleic Acids Res. 2018; 46(W1):W486–w94. Epub 2018/05/16. https://doi.org/10.1093/nar/gky310 PMID: 29762782; PubMed Central PMCID: PMC6030889.

34. Li S, Zheng S. Generalized Tree Structure to Annotate Untargeted Metabolomics and Stable Isotope Tracing Data. Analytical Chemistry. 2023; 95(15):6212–7. https://doi.org/10.1021/acs.analchem.2c05810 PMID: 37018697

35. Martens L, Chambers M, Sturm M, Kessner D, Levander F, Shofstahl J, et al. mzML—a community standard for mass spectrometry data. Molecular & Cellular Proteomics. 2011; 10(1). https://doi.org/10.1074/mcp.R110.000133 PMID: 20716697

36. Hulstaert N, Shofstahl J, Sachsenberg T, Walzer M, Barsnes H, Martens L, et al. ThermoRawFileParser: Modular, Scalable, and Cross-Platform RAW File Conversion. J Proteome Res. 2020; 19(1):537–42. Epub 2019/11/23. https://doi.org/10.1021/acs.jproteome.9b00328 PMID: 31755270; PubMed Central PMCID: PMC7116465.

37. Kramer O. Scikit-Learn. In: Kramer O, editor. Machine Learning for Evolution Strategies. Cham: Springer International Publishing; 2016. p. 45–53.

38. Waskom M, Botvinnik O, O'Kane D, Hobson P, Lukauskas S, Gemperline DC, et al. mwaskom/seaborn: v0.8.1 (September 2017). v0.8.1 ed: Zenodo; 2017.

39. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature methods. 2020; 17(3):261–72. https://doi.org/10.1038/s41592-019-0686-2 PMID: 32015543

40. Oliphant TE. Guide to numpy: Trelgol Publishing USA; 2006.

41. McKinney W. pandas: a foundational Python library for data analysis and statistics. Python for high performance and scientific computing. 2011; 14(9):1–9.

42. Hunter JD. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering. 2007; 9(3):90–5. https://doi.org/10.1109/MCSE.2007.55

43. Behdenna A, Colange M, Haziza J, Gema A, Appé G, Azencott C-A, et al. pyComBat, a Python tool for batch effects correction in high-throughput molecular data using empirical Bayes methods. BMC Bioinformatics. 2023; 24(1):459. https://doi.org/10.1186/s12859-023-05578-5 PMID: 38057718

44. Li S, Zheng S. Generalized Tree Structure to Annotate Untargeted Metabolomics and Stable Isotope Tracing Data. Anal Chem. 2023; 95(15):6212–7. Epub 2023/04/06. https://doi.org/10.1021/acs.analchem.2c05810 PMID: 37018697; PubMed Central PMCID: PMC10117393.

45. de Jonge NF, Hecht H, van der Hooft JJ, Huber F. Reproducible MS/MS library cleaning pipeline in matchms. 2023.

46. Florian Huber SV, Christiaan Meijer, Hanno Spreeuw, Efrain Manuel Villanueva Castilla, Culiang Geng, Justin J. J. van der Hooft, Simon Rogers, Adam Belloum, Faruk Diblen, and Jurriaan H. Spaaks. matchms - processing and similarity evaluation of mass spectrometry data. The Journal of Open Source Software. 2020; 5(52):2411. https://doi.org/10.21105/joss.02411

47. AcquireX Intelligent Data Acquisition Technology for Orbitrap Tribrid mass spectrometers. 2020.

48. Cooper B, Yang R. An assessment of AcquireX and Compound Discoverer software 3.3 for non-targeted metabolomics. Scientific Reports. 2024; 14(1):4841. https://doi.org/10.1038/s41598-024-55356-3 PMID: 38418855

49. MassBank of North America (MoNA) 2023 [Feb 8, 2024]. Available from: https://mona.fiehnlab.ucdavis.edu/.

50. Slabon J. FPDF. v1.86 ed: github; 2023. p. FPDF is a PHP class which allows to generate PDF files with pure PHP. F from FPDF stands for Free: you may use it for any kind of usage and modify it to suit your needs.

51. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nature Biotechnology. 2017; 35(4):316–9. https://doi.org/10.1038/nbt.3820 PMID: 28398311

52. Helmus R, ter Laak TL, van Wezel AP, de Voogt P, Schymanski EL. patRoon: open source software platform for environmental mass spectrometry based non-target screening. Journal of Cheminformatics. 2021; 13(1):1. https://doi.org/10.1186/s13321-020-00477-w PMID: 33407901

53. Yu M, Dolios G, Petrick L. Reproducible untargeted metabolomics workflow for exhaustive MS2 data acquisition of MS1 features. Journal of Cheminformatics. 2022; 14(1):6. https://doi.org/10.1186/s13321-022-00586-8 PMID: 35172886

54. Plyushchenko IV, Fedorova ES, Potoldykova NV, Polyakovskiy KA, Glukhov AI, Rodin IA. Omics Untargeted Key Script: R-Based Software Toolbox for Untargeted Metabolomics with Bladder Cancer Biomarkers Discovery Case Study. J Proteome Res. 2022; 21(3):833–47. Epub 2021/06/24. https://doi.org/10.1021/acs.jproteome.1c00392 PMID: 34161108.

55. Giacomoni F, Le Corguillé G, Monsoor M, Landi M, Pericard P, Pétéra M, et al. Workflow4Metabolomics: a collaborative research infrastructure for computational metabolomics. Bioinformatics. 2015; 31 (9):1493–5. https://doi.org/10.1093/bioinformatics/btu813 PMID: 25527831

56. Liang D, Liu Q, Zhou K, Jia W, Xie G, Chen T. IP4M: an integrated platform for mass spectrometry-based metabolomics data mining. BMC Bioinformatics. 2020; 21(1):444. https://doi.org/10.1186/s12859-020-03786-x PMID: 33028191

57. Bowen TJ, Southam AD, Hall AR, Weber RJM, Lloyd GR, Macdonald R, et al. Simultaneously discovering the fate and biochemical effects of pharmaceuticals through untargeted metabolomics. Nat Commun. 2023; 14(1):4653. Epub 2023/08/04. https://doi.org/10.1038/s41467-023-40333-7 PMID: 37537184; PubMed Central PMCID: PMC10400635.

58. Brahmer JR, Lee JS, Ciuleanu TE, Bernabe Caro R, Nishio M, Urban L, et al. Five-Year Survival Outcomes With Nivolumab Plus Ipilimumab Versus Chemotherapy as First-Line Treatment for Metastatic Non-Small-Cell Lung Cancer in CheckMate 227. J Clin Oncol. 2023; 41(6):1200–12. Epub 2022/10/13. https://doi.org/10.1200/JCO.22.01503 PMID: 36223558; PubMed Central PMCID: PMC9937094.

59. Ansone L, Briviba M, Silamikelis I, Terentjeva A, Perkons I, Birzniece L, et al. Amino Acid Metabolism is Significantly Altered at the Time of Admission in Hospital for Severe COVID-19 Patients: Findings from Longitudinal Targeted Metabolomics Analysis. Microbiol Spectr. 2021; 9(3):e0033821. Epub 2021/12/09. https://doi.org/10.1128/spectrum.00338-21 PMID: 34878333; PubMed Central PMCID: PMC8653833.

60. Halbert CL, Tretyakov K. intervaltree. v3.1.0 ed: github; 2023. p. A mutable, self-balancing interval tree for Python 2 and 3. Queries may be by point, by range overlap, or by range envelopment.

61. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal statistical society: series B (Methodological). 1995; 57(1):289–300.

62. Abdi H, Williams LJ. Tukey's honestly significant difference (HSD) test. Encyclopedia of research design. 2010; 3(1):1–5.

63. Schousboe P, Ronit A, Nielsen HB, Benfield T, Wiese L, Scoutaris N, et al. Reduced levels of pulmonary surfactant in COVID-19 ARDS. Sci Rep. 2022; 12(1):4040. Epub 2022/03/10. https://doi.org/10.1038/s41598-022-07944-4 PMID: 35260704; PubMed Central PMCID: PMC8904856 Paediatrics, Holbaek Hospital, Region Zealand, Denmark and SIME Diagnostics Ltd. HV, NS, PV and PS reported being consultants and shareholders of SIME Diagnostics Ltd. TB reports personal fees and nonfinancial support from Bristol-Myers Squibb and Gilead. The rest of the authors have no conflicts of interest.