ACS Partner Journal

Journal of the American Society for Mass Spectrometry

pubs.acs.org/jasms

Article

Communicating Mass Spectrometry Quality Information in mzQC with Python, R, and Java

Published as part of Journal of the American Society for Mass Spectrometry virtual special issue "Asilomar: Computational Mass Spectrometry".

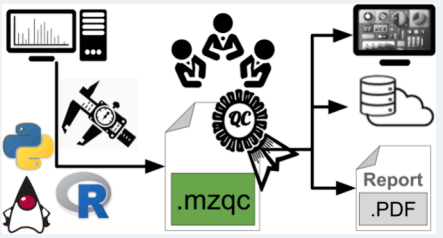
Chris Bielow,* Nils Hoffmann, David Jimenez-Morales, Tim Van Den Bossche, Juan Antonio Vizcaíno, David L. Tabb, Wout Bittremieux, and Mathias Walzer*

Cite This: *J. Am. Soc. Mass Spectrom.* 2024, 35, 1875−1882

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Mass spectrometry is a powerful technique for analyzing molecules in complex biological samples. However, inter- and intralaboratory variability and bias can affect the data due to various factors, including sample handling and preparation, instrument calibration and performance, and data acquisition and processing. To address this issue, the Quality Control (QC) working group of the Human Proteome Organization's Proteomics Standards Initiative has established the standard mzQC file format for reporting and exchanging information relating to data quality. mzQC is based on the JavaScript Object Notation (JSON) format and provides a lightweight yet versatile file format that can be easily implemented in software. Here, we present open-source software libraries to process mzQC data in three programming languages: Python, using pymzqc; R, using rmzqc; and Java, using jmzqc. The libraries follow a common data model and provide shared functionalities, including the (de)serialization and validation of mzQC files. We demonstrate use of the software libraries in a workflow for extracting, analyzing, and visualizing QC metrics from different sources. Additionally, we show how these libraries can be integrated with each other, with existing software tools, and in automated workflows for the QC of mass spectrometry data. All software libraries are available as open source under the MS-Quality-Hub organization on GitHub (https://github.com/MS-Quality-Hub).



INTRODUCTION

Mass spectrometry (MS) is a powerful analytical technique for analyzing molecules in complex biological samples. However, MS data are inherently prone to variability and bias. Even between related MS experiments, subtle variations in sample preparation, instrument performance, and data processing can lead to hidden data inconsistency.[1] To inspire confidence in the results of an MS experiment and ensure consistency and comparability across different measurements, implementing appropriate quality assurance (QA) and quality control (QC) strategies is essential. QC is essential for generating high-quality MS data that can support meaningful and reproducible scientific discoveries. This is especially relevant in light of the reproducibility crisis in science.[2] By improving the quality assessment and reproducibility of MS experiments, QC ensures the credibility and confidence of the resulting scientific findings.

Historically, coordinated efforts for QC in MS were first advocated for in proteomics with the Amsterdam Principles,[3] in 2008, closely followed by proposals for potential metrics by Kinsinger et al.[4] and the NIST MSQC metrics.[5] Since then, several dedicated software packages have emerged for the QC and QA of various mass spectrometry applications.[6−12] Developed to serve a wide range of use cases and workflows, these tools employ a heterogeneous set of QC approaches and metrics. Additionally, several consortia and community initiatives, such as the Metabolomics Quality Assurance & Quality Control Consortium[13] and the Lipidomics Standard Initiative,[14,15] are actively developing QC best practices in their respective fields by establishing community-driven guidelines.

Despite these important efforts, so far no unified approaches toward QC in biological MS have been established. One of the limitations is the lack of a standard file format to store and communicate QC metrics, which are numerical or graphical indicators that describe the quality of MS data at different levels, such as sample quality, instrument performance, completeness of the measurements, and data consistency.[16] Currently, QC metrics are often stored in different formats and locations, such as instrument log files, proprietary software outputs, spreadsheets, and human-readable QC reports. This makes it difficult to access, compare, and share QC

Received: April 29, 2024
Revised: June 7, 2024
Accepted: June 11, 2024
Published: June 26, 2024

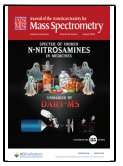


Table 1. Overview of the High-Level Functionality Provided by the mzQC Software Libraries

Functionality	Software library	API
Read (deserialize): consume an mzQC file (optionally from a JSON string, a local file, or a remote file) and return a data object representing the file contents	pymzqc	MZQCFile.JsonSerialisable.FromJson(..)
	rmzqc	rmzqc::readMZQC(..)
		MzQC$fromData(..)
	jmzqc	Converter.of(..)
Write (serialize): export an mzQC data object to a JSON file or JSON string	pymzqc	MZQCFile.JsonSerialisable.ToJson(..)
	rmzqc	rmzqc::writeMZQC(..)
		jsonlite::toJSON(..)
	jmzqc	Converter.toJsonString(..)
		Converter.toJsonFile(..)
Syntactic validation: verify that an mzQC file conforms to the mzQC schema specification	pymzqc	SyntaxCheck().validate(..)
	rmzqc	rmzqc::validateFromFile(..)
		rmzqc::validateFromString(..)
		rmzqc::validateFromObj(..)
	jmzqc	Converter.validate(..)
Semantic validation: verify that an mzQC file conforms to the mzQC semantic content constraints	pymzqc	SemanticCheck().validate(..)

information over time, across different instruments, sample preparation techniques, and laboratories.

To address this issue, the Quality Control working group[17] of the Human Proteome Organization's Proteomics Standards Initiative (HUPO-PSI)[18] has recently established the standard mzQC file format (https://github.com/HUPO-PSI/mzQC) to report and exchange data quality-related information for MS experiments and the associated analysis results. mzQC is based on the widespread JavaScript Object Notation (JSON) format to provide a lightweight yet versatile file format that can be easily implemented in software to produce or consume mzQC files, and its goal is to support diverse workflows in proteomics, metabolomics, and other MS applications. It is important to note that mzQC aims to provide a standardized framework for storing and exchanging QC metrics in MS data analysis in a transparent manner, rather than to directly judge the quality of the data it describes.

QC metrics in an mzQC file are grouped in "runQuality" or "setQuality" elements, depending on whether the metrics pertain to a single or multiple MS runs, respectively. Each runQuality or setQuality element contains a "metadata" section that provides information to track the provenance of the QC metrics, such as the originating MS run(s) and the software tool(s) used to calculate the metrics. QC metric values are stored in "qualityMetric" elements and can consist of single values, tuples, or tabular data. Additionally, each QC metric is defined by a corresponding term in the PSI-MS controlled vocabulary[19] for semantic annotation of the data and to ensure an unambiguous definition of each QC metric. Further technical details of the mzQC format and the official PSI specification document (version 1.0, released in February 2024) are available at https://github.com/HUPO-PSI/mzQC.

To ensure the adoption of the mzQC format, supporting software tools are needed. There is a vibrant open-source community of bioinformaticians developing software to analyze MS data in various programming languages, among which some of the most popular are Python,[20−23] which is widely used for data analysis and machine learning; R,[24,25] a language designed for statistical computing and graphics; and Java,[26,27] a multiplatform programming language that is suitable for large-scale applications.

In this manuscript, we present open-source software libraries to read, write, and validate QC data in the mzQC format in the three programming languages mentioned above: Python, R, and Java. We describe the design and implementation of these libraries, which follow a common data model and provide shared functionality to operate on mzQC files. We demonstrate the use of these software libraries for extracting, analyzing, and visualizing QC metrics from different sources. We also show how these libraries can be integrated with existing software tools and workflows for performing QC of MS data, with mzQC acting as the glue between various workflow steps. All software libraries are available as open source under the MS-Quality-Hub organization on GitHub (https://github.com/MS-Quality-Hub/).

METHODS

mzQC Software Libraries. The mzQC software libraries are implemented in three popular programming languages (Table 1): pymzqc in Python, rmzqc in R, and jmzqc in Java. Each library builds on the mzQC schema definition, which formally defines the syntax of mzQC documents using a JSON schema, and provides a high-level abstraction of data quality-related information in mzQC files. Rather than a single application programming interface (API) that all libraries share, they each follow the conventions and best practices of their respective programming languages.

The primary functionality provided by all three software libraries is the serialization and deserialization of mzQC files, which facilitates the reading and writing of QC information, respectively. This enables users to create mzQC files containing newly computed QC information, read existing mzQC files with previously computed QC metrics, and manipulate the QC information for further data processing and analysis. The software libraries automatically perform native value type matching where possible, such as converting tabular data to data.frame objects in R or Pandas DataFrame objects in Python.

It is important to note that the mzQC software libraries do not calculate QC metrics directly. Instead, they facilitate the import and export of metric values obtained using external software from/to mzQC files. As such, they are agnostic to input file formats for MS-related data, such as mzML,[28] mzTab,[29] or custom formats, and operate at the level of QC metrics instead. The mzQC libraries provide functionality to conveniently create mzQC-related data structures, such as a

https://doi.org/10.1021/jasms.4c00174
J. Am. Soc. Mass Spectrom. 2024, 35, 1875−1882

"runQuality" or "setQuality", and assemble these into a complete mzQC report in the respective programming language.

In addition to (de)serialization, the software libraries provide user-friendly functionality to validate mzQC files. Syntactic validation checks if the structure of mzQC data conforms to the defined syntax rules, ensuring that the data are structured correctly and contain all necessary pieces of information. Semantic validation, on the other hand, involves verifying that the data make sense in their specific context, ensuring that they meaningfully and logically represent MS-related QC concepts and information. All three libraries support syntactic validation, which is based on the mzQC JSON schema. The pymzqc library also supports semantic validation, which interprets the content of mzQC files to ensure the correctness of the QC information, including verification that all QC metrics are represented in an accessible controlled vocabulary or ontology and that the data value types match the definition in the controlled vocabularies. Additionally, a web application to validate mzQC files, powered by pymzqc, is available at https://hupo-psi.github.io/mzQC/validator/.

Code availability. All mzQC supporting software libraries are freely available on GitHub as open source (Table 2), collected in the MS-Quality-Hub organization (https://github.com/MS-Quality-Hub). Additionally, all software libraries can be easily installed using their respective language-preferred toolchains (Table 2). All software libraries follow development best practices, including extensive code documentation, detailed installation instructions, and automated testing using continuous integration.

Table 2. Availability of the Software Libraries in Their Respective Software Package and Source Code Repositories

Software library	URL
pymzqc	PyPI: https://pypi.org/project/pymzqc/
	GitHub: https://github.com/MS-Quality-Hub/pymzqc
rmzqc	CRAN: https://cran.r-project.org/web/packages/rmzqc/index.html
	GitHub: https://github.com/MS-Quality-Hub/rmzqc
jzmqc	Maven Central: https://central.sonatype.com/artifact/org.lifs-tools/jmzqc
	GitHub: https://github.com/MS-Quality-Hub/jmzqc

All code used in this manuscript to demonstrate the library's capabilities can be found as open source in a dedicated GitHub repository under the MS-Quality-Hub organization at https://github.com/MS-Quality-hub/mzqclib-manuscript.

Data. We have reanalyzed MS data from a proteomics study of anaerobic respiration in *E. coli* grown in sulforaphane, obtained via ProteomeXchange[30] with data set identifier PXD040621.[31] In this study, bacterial cultures were grown in the presence of either sulforaphane (10 μM) or 0.034% dimethyl sulfoxide (DMSO) as a control. The study comprised four biological replicates of bacterial growth in both conditions, acquired using a 120 min liquid chromatography gradient measured on an Orbitrap Q-Exactive using data-dependent acquisition.

The data were reanalyzed by converting the raw files to mzML[28] using ThermoRawFileParser (biocontainer version 1.4.0)[32] and sequence database searching using Tide (Crux toolkit version 4.2).[33] We performed target−decoy searching against the UniProtKB[34] *E. coli* K12 reference proteome (UP000000625, downloaded on October 20, 2023) and configured the search for tryptic peptides with up to three missed cleavages, variable oxidation of methionine, and a precursor mass tolerance of 50 ppm. Spectrum identifications were filtered at a 1% protein-level false discovery rate with crema (version 0.0.10).[35]

Results. To illustrate the functionality and interoperability of the mzQC software libraries, we first used data analysis scripts in different programming languages to compute various QC metric values. Next, the respective mzQC software libraries were used to produce separate mzQC files containing these QC metrics, after which the individual mzQC files were combined into a final QC report. While this demonstration deliberately splits QC metric calculation and mzQC file generation across three programming languages to demonstrate the interoperability of the software libraries, a similar result could be achieved using the respective mzQC software library within a single programming language of preference.

Our example workflow consists of several steps (Figure 1). First, the raw files were converted to mzML and processed using sequence database searching. Next, various QC metrics (Supplementary Table 1) were computed using dedicated scripts in Java, Python, and R and exported to mzQC files using the mzQC software library for the corresponding programming language: (i) jmzqc: As a compiled language, Java is excellently suited to process large amounts of data. Consequently, we used jmzqc combined with jmzml[36] and MSDK[37] to efficiently read the mzML peak files and compute basic QC metrics from the MS data. We calculated the number of chromatograms, the m/z range of the acquired spectra, the retention time range of the acquired spectra, the total ion chromatogram, and the base peak intensities. (ii) rmzqc: Based on R's emphasis on statistical processing, we used rmzqc to collect statistics of the ion injection parameters at the level of MS and MS/MS spectra. (iii) pymzqc: We used pymzqc in combination with Pyteomics[21] and Pandas[38] to read the peak and identification data; count the number of MS/MS spectra, the number of identified MS/MS spectra, the number of identified peptides, and the number of identified proteins; compute the distribution of the precursor mass deviation of the identifications; evaluate the number of missed cleavages; and find the retention time range during which spectra could be successfully annotated.

This process results in the creation of three mzQC files, each generated by one of the three programming languages. While these can work as independent quality reports, containing a limited set of QC metrics, they can also be combined into a single mzQC report that contains the full information. Therefore, pymzqc was used to merge the data into a final mzQC file. A Jupyter notebook[39] was used for subsequent interactive data analysis and to produce a report that summarizes all QC metrics. Metrics across all MS runs were visualized using a clustered heatmap, with metric values percentile rank scaled (Figure 2).

As a brief example, we performed a visual inspection of the QC metrics to illustrate how QC data can be used to explore the implications of MS data quality (Figure 2). Note that this description is not provided by mzQC directly but is based on the authors' interpretation of the heatmap. When examining the heatmap and its clustering of QC metrics across the eight MS runs, we observe a distinct separation between two specific runs from the rest: the Ecoli_DMSO_rep1_EG-1 control run

https://doi.org/10.1021/jasms.4c00174
J. Am. Soc. Mass Spectrom. 2024, 35, 1875−1882

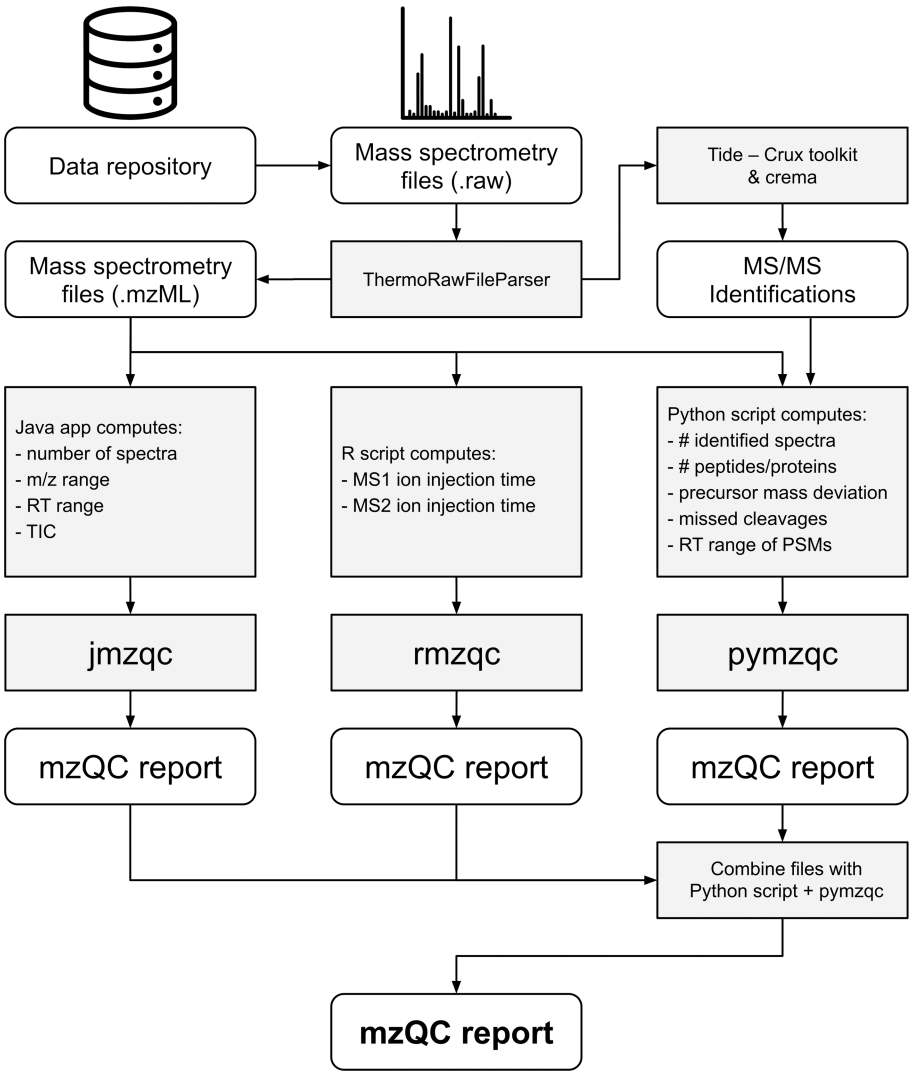


Figure 1. mzQC processing workflow. Each software library is separately used to process different QC metrics, which are ultimately combined into a single QC report. Note that the mzQC software libraries do not calculate QC metric values themselves, but rather this functionality is provided by external scripts ("Java app", "R script", "Python script") that subsequently use the respective mzQC library to produce the corresponding mzQC reports.

and the Ecoli_Suf_rep1_EG-5 sulforaphane run. This divergence is driven by a comparatively lower number of MS/MS spectra acquired, peptides and proteins identified, as well as related QC metrics. Interestingly, these two runs exhibit above average total ion currents and the rate of MS/MS spectra that could be identified, suggesting that the lower peptide and protein identification rate is due to a decrease in the number of MS/MS spectra acquired, rather than due to a reduction in the quality of the MS/MS spectra. Additionally, the two outlier runs have a higher proportion of peptides with no missed tryptic cleavages, which might impact the subsequent protein inference. Consequently, deriving biological interpretations from the full experiment may require robust statistical models that are resistant to outliers.

The large contrast in the number of acquired MS/MS spectra and identifications indicates that some caution might need to be exercised when interpreting the results for these two runs to study the effect of sulforaphane on *E. coli* growth. The heatmap generated by the mzQC pipeline suggests the need for a deeper investigation into the cause of these discrepancies. This will necessitate a broader QC analysis incorporating long-term instrument performance monitoring, including repeated analysis of consistent QC samples. This would provide a more informed basis for interpreting these outlier results within the context of the study.

CONCLUSION

The introduction of the mzQC standard file format for quality control in biological mass spectrometry has numerous potential benefits, including increased reproducibility, improved interoperability, and enhanced data sharing among researchers. However, adopting new file formats can be challenging without suitable software libraries to facilitate their integration into bioinformatics software. The development of open software libraries such as pymzqc, rmzqc, and jmzqc is therefore essential for the successful adoption of mzQC. These libraries provide a consistent interface for accessing mzQC files, allowing bioinformatics software developers to easily incorporate mzQC into their tools and workflows. Third-party support for mzQC is already emerging[12] and will be further strengthened by the presented software libraries.

https://doi.org/10.1021/jasms.4c00174
J. Am. Soc. Mass Spectrom. 2024, 35, 1875−1882

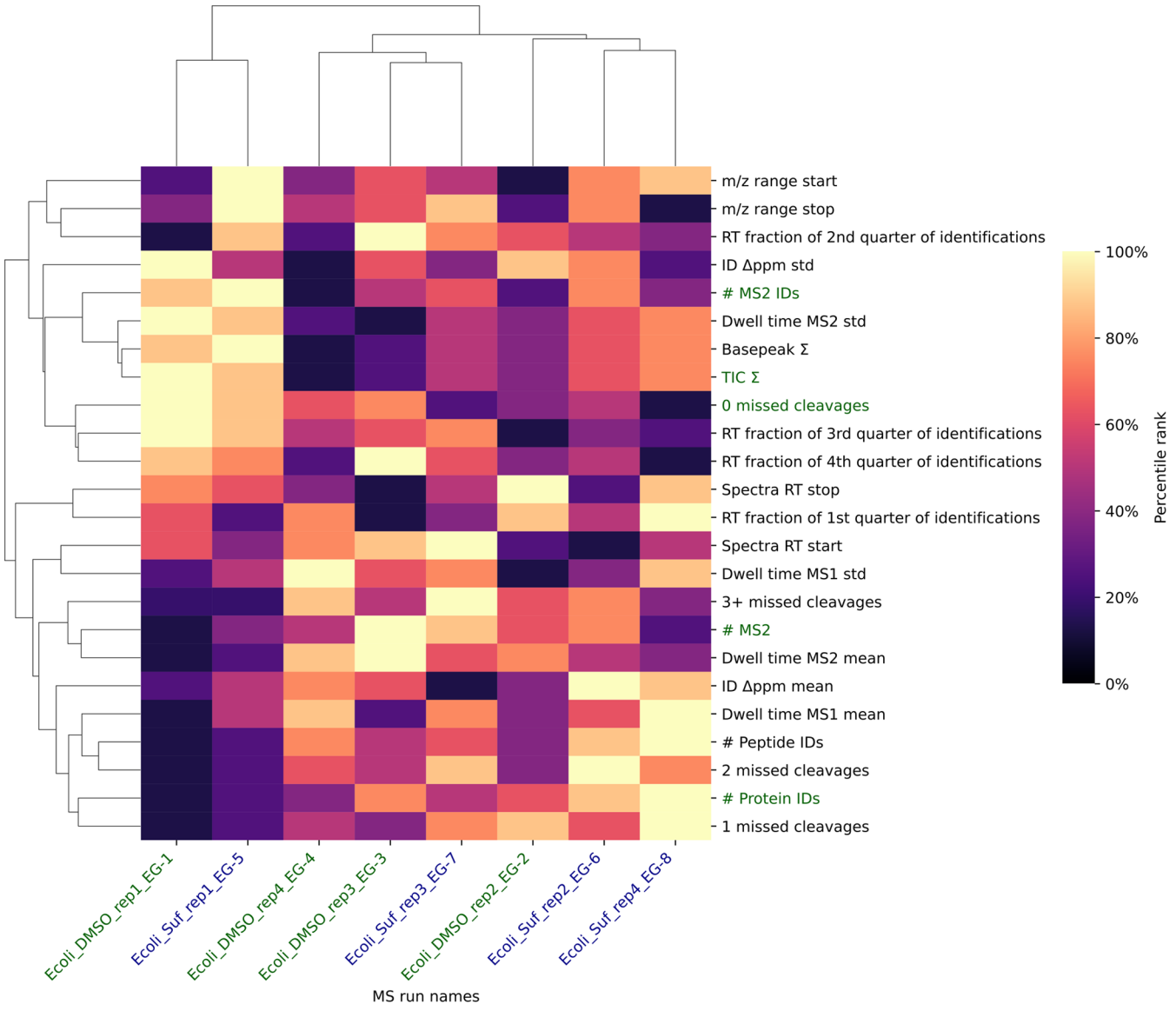


Figure 2. Heatmap with dendrogram displaying QC metrics across eight MS runs (DMSO controls colored in green, sulforaphane samples colored in blue), clustered by MS runs on the horizontal axis and by QC metrics on the vertical axis. Colors in the heatmap represent percentile ranks calculated from the combined data set, with darker shades indicating lower percentile ranks and lighter shades indicating higher ranks. The QC metrics include the number of acquired MS/MS spectra, MS/MS identifications, peptide identifications, protein identifications, summed total ion current, and number of missed cleavages, among others. The metrics discussed in the text are highlighted in green. See the Jupyter Notebook on our GitHub repository (https://github.com/MS-Quality-Hub/mzqclib-manuscript/) for data analysis and code to generate the plot.

On the basis of a worked use case, we have demonstrated that only a small amount of code is needed to construct an mzQC object in memory, populate it with calculated metric values, and export the data to an mzQC file on disk. Likewise, the interactive notebooks showcase the libraries for conveniently reading data from mzQC for further processing and reporting. This illustrates how the high-level abstractions provided by the mzQC libraries presented here facilitate interactions with mzQC files in different programming languages.

The mzQC software libraries offer several key benefits, including the ability to validate mzQC files, extract information from them, and convert to and from other formats. Additionally, the complexity of the mzQC software libraries is limited, building on native JSON support in the various programming languages. These capabilities are crucial for the development of new QC tools and workflows that can help to improve the reliability and reproducibility of mass spectrometry experiments. Especially as data analysis pipelines become increasingly complex, with separate processing steps implemented in different programming languages, this multilanguage support will be highly beneficial. This could even take the form of polyglot programming, where operations in multiple programming languages are combined in a single analysis notebook. As such, we anticipate that our software libraries will foster a vibrant ecosystem of general and bespoke bioinformatics tools for QC of MS experiments, which will be able to seamlessly interoperate through a common mzQC interface.

In light of these benefits, we invite software developers to start using the mzQC software libraries. Additionally, we welcome any contributions to the mzQC software libraries and

https://doi.org/10.1021/jasms.4c00174
J. Am. Soc. Mass Spectrom. 2024, 35, 1875−1882

the mzQC format, for example by developing complementary libraries in alternative programming languages, such as C++, C#, Rust, or JavaScript. By doing so, the adoption of mzQC in the mass spectrometry and bioinformatics communities will be further facilitated, ultimately leading to better-quality data and more reliable scientific discoveries. To ensure the continued evolution and application of mzQC, we are dedicated to enhancing its ecosystem, including broadening its integration into diverse bioinformatics tools and developing extensive use cases for QC across various biological mass spectrometry applications.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/jasms.4c00174.

Supplementary Table 1: List of QC metrics considered during the analysis and their accession numbers in the PSI-MS controlled vocabulary (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Chris Bielow − *Bioinformatics Solution Center, Institut für Mathematik und Informatik, Freie Universität Berlin, 14195 Berlin, Germany;* orcid.org/0000-0001-5756-3988; Email: chris.bielow@bsc.fu-berlin.de

Mathias Walzer − *European Molecular Biology Laboratory, EMBL-European Bioinformatics Institute (EMBL-EBI), Cambridge CB10 1SD, United Kingdom;* orcid.org/0000-0003-4538-2754; Email: walzer@ebi.ac.uk

Authors

Nils Hoffmann − *Institute for Bio- and Geosciences (IBG-5), Forschungszentrum Jülich GmbH, 52428 Jülich, Germany;* orcid.org/0000-0002-6540-6875

David Jimenez-Morales − *Department of Medicine, Stanford University School of Medicine, Stanford, California 94305, United States*

Tim Van Den Bossche − *Department of Biomolecular Medicine, Faculty of Medicine and Health Sciences, Ghent University, 9052 Ghent, Belgium; VIB-UGent Center for Medical Biotechnology, 9052 Ghent, Belgium;* orcid.org/0000-0002-5916-2587

Juan Antonio Vizcaíno − *European Molecular Biology Laboratory, EMBL-European Bioinformatics Institute (EMBL-EBI), Cambridge CB10 1SD, United Kingdom;* orcid.org/0000-0002-3905-4335

David L. Tabb − *European Research Institute for the Biology of Ageing, University Medical Center Groningen, Groningen 9713 AV, The Netherlands;* orcid.org/0000-0001-7223-578X

Wout Bittremieux − *Department of Computer Science, University of Antwerp, Antwerpen 2020, Belgium;* orcid.org/0000-0002-3105-1359

Complete contact information is available at: https://pubs.acs.org/10.1021/jasms.4c00174

Author Contributions

C.B. and N.H. contributed equally.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

M.W. would like to acknowledge funding from the H2020 EPIC-XS grant [Grant number 823839], BBSRC 'Proteomics DIA' [BB/P024599/1], and from The Wellcome Trust [208391/Z/17/Z]. Additionally, J.A.V. would like to acknowledge EMBL core funding. This work was supported in part by the de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) and ELIXIR-DE (Forschungszentrum Jülich and W-de.NBI-001, W-de.NBI-004, W-de.NBI-008, W-de.NBI-010, W-de.NBI-013, W-de.NBI-014, W-de.NBI-016, and W-de.NBI-022). T.V.D.B. acknowledges funding from the Research Foundation Flanders (FWO) [1286824N]. W.B. acknowledges support by the University of Antwerp Research Fund.

■ REFERENCES

(1) Bittremieux, W.; Tabb, D. L.; Impens, F.; Staes, A.; Timmerman, E.; Martens, L.; Laukens, K. Quality Control in Mass Spectrometry-Based Proteomics. *Mass Spectrom. Rev.* **2018**, *37* (5), 697−711.

(2) Baker, M. 1,500 Scientists Lift the Lid on Reproducibility. *Nature* **2016**, *533* (7604), 452−454.

(3) Rodriguez, H.; Snyder, M.; Uhlén, M.; Andrews, P.; Beavis, R.; Borchers, C.; Chalkley, R. J.; Cho, S. Y.; Cottingham, K.; Dunn, M.; Dylag, T.; Edgar, R.; Hare, P.; Heck, A. J. R.; Hirsch, R. F.; Kennedy, K.; Kolar, P.; Kraus, H.-J.; Mallick, P.; Nesvizhskii, A.; Ping, P.; Pontén, F.; Yang, L.; Yates, J. R.; Stein, S. E.; Hermjakob, H.; Kinsinger, C. R.; Apweiler, R. Recommendations from the 2008 International Summit on Proteomics Data Release and Sharing Policy: The Amsterdam Principles. *J. Proteome Res.* **2009**, *8* (7), 3689−3692.

(4) Kinsinger, C. R.; Apffel, J.; Baker, M.; Bian, X.; Borchers, C. H.; Bradshaw, R.; Brusniak, M.-Y.; Chan, D. W.; Deutsch, E. W.; Domon, B.; Gorman, J.; Grimm, R.; Hancock, W.; Hermjakob, H.; Horn, D.; Hunter, C.; Kolar, P.; Kraus, H.-J.; Langen, H.; Linding, R.; Moritz, R. L.; Omenn, G. S.; Orlando, R.; Pandey, A.; Ping, P.; Rahbar, A.; Rivers, R.; Seymour, S. L.; Simpson, R. J.; Slotta, D.; Smith, R. D.; Stein, S. E.; Tabb, D. L.; Tagle, D.; Yates, J. R. I.; Rodriguez, H. Recommendations for Mass Spectrometry Data Quality Metrics for Open Access Data (Corollary to the Amsterdam Principles). *Mol. Cell. Proteomics* **2011**, *10* (12), O111.015446.

(5) Rudnick, P. A.; Clauser, K. R.; Kilpatrick, L. E.; Tchekhovskoi, D. V.; Neta, P.; Blonder, N.; Billheimer, D. D.; Blackman, R. K.; Bunk, D. M.; Cardasis, H. L.; Ham, A.-J. L.; Jaffe, J. D.; Kinsinger, C. R.; Mesri, M.; Neubert, T. A.; Schilling, B.; Tabb, D. L.; Tegeler, T. J.; Vega-Montoto, L.; Variyath, A. M.; Wang, M.; Wang, P.; Whiteaker, J. R.; Zimmerman, L. J.; Carr, S. A.; Fisher, S. J.; Gibson, B. W.; Paulovich, A. G.; Regnier, F. E.; Rodriguez, H.; Spiegelman, C.; Tempst, P.; Liebler, D. C.; Stein, S. E. Performance Metrics for Liquid Chromatography-Tandem Mass Spectrometry Systems in Proteomics Analyses. *Mol. Cell. Proteomics* **2010**, *9* (2), 225−241.

(6) Ma, Z.-Q.; Polzin, K. O.; Dasari, S.; Chambers, M. C.; Schilling, B.; Gibson, B. W.; Tran, B. Q.; Vega-Montoto, L.; Liebler, D. C.; Tabb, D. L. QuaMeter: Multivendor Performance Metrics for LC-MS/MS Proteomics Instrumentation. *Anal. Chem.* **2012**, *84* (14), 5845−5850.

(7) Pichler, P.; Mazanek, M.; Dusberger, F.; Weilnböck, L.; Huber, C. G.; Stingl, C.; Luider, T. M.; Straube, W. L.; Köcher, T.; Mechtler, K. SIMPATIQCO: A Server-Based Software Suite Which Facilitates Monitoring the Time Course of LC-MS Performance Metrics on Orbitrap Instruments. *J. Proteome Res.* **2012**, *11* (11), 5540−5547.

(8) Bielow, C.; Mastrobuoni, G.; Kempa, S. Proteomics Quality Control: Quality Control Software for MaxQuant Results. *J. Proteome Res.* **2016**, *15* (3), 777−787.

(9) Chiva, C.; Olivella, R.; Borràs, E.; Espadas, G.; Pastor, O.; Solé, A.; Sabidó, E. QCloud: A Cloud-Based Quality Control System for Mass Spectrometry-Based Proteomics Laboratories. *PLOS ONE* **2018**, *13* (1), No. e0189209.

(10) Broadhurst, D.; Goodacre, R.; Reinke, S. N.; Kuligowski, J.; Wilson, I. D.; Lewis, M. R.; Dunn, W. B. Guidelines and Considerations for the Use of System Suitability and Quality Control Samples in Mass Spectrometry Assays Applied in Untargeted Clinical Metabolomic Studies. *Metabolomics* **2018**, *14* (6), 72.
(11) Stratton, K. G.; Webb-Robertson, B.-J. M.; McCue, L. A.; Stanfill, B.; Claborne, D.; Godinez, I.; Johansen, T.; Thompson, A. M.; Burnum-Johnson, K. E.; Waters, K. M.; Bramer, L. M. pmartR: Quality Control and Statistics for Mass Spectrometry-Based Biological Data. *J. Proteome Res.* **2019**, *18* (3), 1418−1425.
(12) Naake, T.; Rainer, J.; Huber, W. MsQuality: An Interoperable Open-Source Package for the Calculation of Standardized Quality Metrics of Mass Spectrometry Data. *Bioinformatics* **2023**, *39* (10), btad618.
(13) Kirwan, J. A.; Gika, H.; Beger, R. D.; Bearden, D.; Dunn, W. B.; Goodacre, R.; Theodoridis, G.; Witting, M.; Yu, L.-R.; Wilson, I. D. the metabolomics Quality Assurance and Quality Control Consortium (mQACC). Quality Assurance and Quality Control Reporting in Untargeted Metabolic Phenotyping: mQACC Recommendations for Analytical Quality Management. *Metabolomics* **2022**, *18* (9), 70.
(14) Köfeler, H. C.; Ahrends, R.; Baker, E. S.; Ekroos, K.; Han, X.; Hoffmann, N.; Holčapek, M.; Wenk, M. R.; Liebisch, G. Recommendations for Good Practice in MS-Based Lipidomics. *J. Lipid Res.* **2021**, *62*, 100138.
(15) McDonald, J. G.; Ejsing, C. S.; Kopczynski, D.; Holčapek, M.; Aoki, J.; Arita, M.; Arita, M.; Baker, E. S.; Bertrand-Michel, J.; Bowden, J. A.; Brügger, B.; Ellis, S. R.; Fedorova, M.; Griffiths, W. J.; Han, X.; Hartler, J.; Hoffmann, N.; Koelmel, J. P.; Köfeler, H. C.; Mitchell, T. W.; O'Donnell, V. B.; Saigusa, D.; Schwudke, D.; Shevchenko, A.; Ulmer, C. Z.; Wenk, M. R.; Witting, M.; Wolrab, D.; Xia, Y.; Ahrends, R.; Liebisch, G.; Ekroos, K. Introducing the Lipidomics Minimal Reporting Checklist. *Nat. Metab.* **2022**, *4* (9), 1086−1088.
(16) Bittremieux, W.; Valkenborg, D.; Martens, L.; Laukens, K. Computational Quality Control Tools for Mass Spectrometry Proteomics. *PROTEOMICS* **2017**, *17* (3−4), 1600159.
(17) Bittremieux, W.; Walzer, M.; Tenzer, S.; Zhu, W.; Salek, R. M.; Eisenacher, M.; Tabb, D. L. The Human Proteome Organization-Proteomics Standards Initiative Quality Control Working Group: Making Quality Control More Accessible for Biological Mass Spectrometry. *Anal. Chem.* **2017**, *89* (8), 4474−4479.
(18) Deutsch, E. W.; Vizcaíno, J. A.; Jones, A. R.; Binz, P.-A.; Lam, H.; Klein, J.; Bittremieux, W.; Perez-Riverol, Y.; Tabb, D. L.; Walzer, M.; Ricard-Blum, S.; Hermjakob, H.; Neumann, S.; Mak, T. D.; Kawano, S.; Mendoza, L.; Van Den Bossche, T.; Gabriels, R.; Bandeira, N.; Carver, J.; Pullman, B.; Sun, Z.; Hoffmann, N.; Shofstahl, J.; Zhu, Y.; Licata, L.; Quaglia, F.; Tosatto, S. C. E.; Orchard, S. E. Proteomics Standards Initiative at Twenty Years: Current Activities and Future Work. *J. Proteome Res.* **2023**, *22* (2), 287−301.
(19) Mayer, G.; Montecchi-Palazzi, L.; Ovelleiro, D.; Jones, A. R.; Binz, P.-A.; Deutsch, E. W.; Chambers, M.; Kallhardt, M.; Levander, F.; Shofstahl, J.; Orchard, S.; Vizcaino, J. A.; Hermjakob, H.; Stephan, C.; Meyer, H. E.; Eisenacher, M. The HUPO Proteomics Standards Initiative- Mass Spectrometry Controlled Vocabulary. *Database* **2013**, *2013*, bat009.
(20) Röst, H. L.; Schmitt, U.; Aebersold, R.; Malmström, L. pyOpenMS: A Python-Based Interface to the OpenMS Mass-Spectrometry Algorithm Library. *PROTEOMICS* **2014**, *14* (1), 74−77.
(21) Levitsky, L. I.; Klein, J. A.; Ivanov, M. V.; Gorshkov, M. Pyteomics 4.0: Five Years of Development of a Python Proteomics Framework. *J. Proteome Res.* **2019**, *18* (2), 709−714.
(22) Huber, F.; Verhoeven, S.; Meijer, C.; Spreeuw, H.; Castilla, E.; Geng, C.; van der Hooft, J.; Rogers, S.; Belloum, A.; Diblen, F.; Spaaks, J. Matchms - Processing and Similarity Evaluation of Mass Spectrometry Data. *J. Open Source Softw.* **2020**, *5* (52), 2411.
(23) Bittremieux, W.; Levitsky, L.; Pilz, M.; Sachsenberg, T.; Huber, F.; Wang, M.; Dorrestein, P. C. Unified and Standardized Mass Spectrometry Data Processing in Python Using Spectrum_utils. *J. Proteome Res.* **2023**, *22* (2), 625−631.
(24) Smith, C. A.; Want, E. J.; 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), 779−787.
(25) Gatto, L.; Gibb, S.; Rainer, J. MSnbase, Efficient and Elegant R-Based Processing and Visualization of Raw Mass Spectrometry Data. *J. Proteome Res.* **2021**, *20* (1), 1063−1069.
(26) Barsnes, H.; Vaudel, M.; Colaert, N.; Helsens, K.; Sickmann, A.; Berven, F. S.; Martens, L. Compomics-Utilities: An Open-Source Java Library for Computational Proteomics. *BMC Bioinformatics* **2011**, *12* (1), 70.
(27) Schmid, R.; Heuckeroth, S.; Korf, A.; Smirnov, A.; Myers, O.; Dyrlund, T. S.; Bushuiev, R.; Murray, K. J.; Hoffmann, N.; Lu, M.; Sarvepalli, A.; Zhang, Z.; Fleischauer, M.; Dührkop, K.; Wesner, M.; Hoogstra, S. J.; Rudt, E.; Mokshyna, O.; Brungs, C.; Ponomarov, K.; Mutabdžija, L.; Damiani, T.; Pudney, C. J.; Earll, M.; Helmer, P. O.; Fallon, T. R.; Schulze, T.; Rivas-Ubach, A.; Bilbao, A.; Richter, H.; Nothias, L.-F.; Wang, M.; Orešič, M.; Weng, J.-K.; Böcker, S.; Jeibmann, A.; Hayen, H.; Karst, U.; Dorrestein, P. C.; Petras, D.; Du, X.; Pluskal, T. Integrative Analysis of Multimodal Mass Spectrometry Data in MZmine 3. *Nat. Biotechnol.* **2023**, *41* (4), 447−449.
(28) Martens, L.; Chambers, M.; Sturm, M.; Kessner, D.; Levander, F.; Shofstahl, J.; Tang, W. H.; Römpp, A.; Neumann, S.; Pizarro, A. D.; Montecchi-Palazzi, L.; Tasman, N.; Coleman, M.; Reisinger, F.; Souda, P.; Hermjakob, H.; Binz, P.-A.; Deutsch, E. W. mzML—a Community Standard for Mass Spectrometry Data. *Mol. Cell. Proteomics* **2011**, *10* (1), R110.000133.
(29) Griss, J.; Jones, A. R.; Sachsenberg, T.; Walzer, M.; Gatto, L.; Hartler, J.; Thallinger, G. G.; Salek, R. M.; Steinbeck, C.; Neuhauser, N.; Cox, J.; Neumann, S.; Fan, J.; Reisinger, F.; Xu, Q.-W.; del Toro, N.; Perez-Riverol, Y.; Ghali, F.; Bandeira, N.; Xenarios, I.; Kohlbacher, O.; Vizcaíno, J. A.; Hermjakob, H. The mzTab Data Exchange Format: Communicating Mass-Spectrometry-Based Proteomics and Metabolomics Experimental Results to a Wider Audience. *Mol. Cell. Proteomics* **2014**, *13* (10), 2765−2775.
(30) Deutsch, E. W.; Bandeira, N.; Sharma, V.; Perez-Riverol, Y.; Carver, J. J.; Kundu, D. J.; García-Seisdedos, D.; Jarnuczak, A. F.; Hewapathirana, S.; Pullman, B. S.; Wertz, J.; Sun, Z.; Kawano, S.; Okuda, S.; Watanabe, Y.; Hermjakob, H.; MacLean, B.; MacCoss, M. J.; Zhu, Y.; Ishihama, Y.; Vizcaíno, J. A. The ProteomeXchange Consortium in 2020: Enabling "big Data" Approaches in Proteomics. *Nucleic Acids Res.* **2019**, *48* (D1), D1145−D1152.
(31) Marshall, S. A.; Young, R. B.; Lewis, J. M.; Rutten, E. L.; Gould, J.; Barlow, C. K.; Giogha, C.; Marcelino, V. R.; Fields, N.; Schittenhelm, R. B.; Hartland, E. L.; Scott, N. E.; Forster, S. C.; Gulliver, E. L. The Broccoli-Derived Antioxidant Sulforaphane Changes the Growth of Gastrointestinal Microbiota, Allowing for the Production of Anti-Inflammatory Metabolites. *J. Funct. Foods* **2023**, *107*, 105645.
(32) Hulstaert, N.; Shofstahl, J.; Sachsenberg, T.; Walzer, M.; Barsnes, H.; Martens, L.; Perez-Riverol, Y. ThermoRawFileParser: Modular, Scalable, and Cross-Platform RAW File Conversion. *J. Proteome Res.* **2020**, *19* (1), 537−542.
(33) Park, C. Y.; Klammer, A. A.; Käll, L.; MacCoss, M. J.; Noble, W. S. Rapid and Accurate Peptide Identification from Tandem Mass Spectra. *J. Proteome Res.* **2008**, *7* (7), 3022−3027.
(34) The UniProt Consortium UniProt: A Hub for Protein Information. *Nucleic Acids Res.* **2015**, *43* (D1), D204−D212, .
(35) Lin, A.; See, D.; Fondrie, W. E.; Keich, U.; Noble, W. S. Target-Decoy False Discovery Rate Estimation Using Crema. *PROTEOMICS* **2024**, *24* (8), 2300084.
(36) Côté, R. G.; Reisinger, F.; Martens, L. jmzML, an Open-Source Java API for mzML, the PSI Standard for MS Data. *PROTEOMICS* **2010**, *10* (7), 1332−1335.

https://doi.org/10.1021/jasms.4c00174
J. Am. Soc. Mass Spectrom. 2024, 35, 1875−1882

(37) Pluskal, T.; Hoffmann, N.; Du, X.; Weng, J.-K. Mass Spectrometry Development Kit (MSDK): A Java Library for Mass Spectrometry Data Processing. In *New Developments in Mass Spectrometry*; Winkler, R., Ed.; Royal Society of Chemistry: Cambridge, 2020; pp 399−405, DOI: 10.1039/9781788019880-00399.
(38) McKinney, W. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*; van der Walt, S.; Millman, J., Eds.; Austin, Texas, USA, 2010; pp 51−56, DOI: 10.25080/Majora-92bf1922-00a.
(39) Thomas, K.; Benjamin, R.-K.; Fernando, P.; Brian, G.; Matthias, B.; Jonathan, F.; Kyle, K.; Jessica, H.; Jason, G.; Sylvain, C.; Paul, I.; Damián, A.; Safia, A.; Carol, W. Jupyter Development Team. Jupyter Notebooks - A Publishing Format for Reproducible Computational Workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*; IOS Press, 2016; pp 87−90.

https://doi.org/10.1021/jasms.4c00174
J. Am. Soc. Mass Spectrom. 2024, 35, 1875−1882