Seeking community input for: *mzPeak* - a modern, scalable, and interoperable mass spectrometry data format for the future

*Tim Van Den Bossche[#], Samuel Wein[#], Theodore Alexandrov, Aivett Bilbao, Wout Bittremieux, Matt Chambers, Eric W. Deutsch, Andrew Dowsey, Helge Hecht, Joshua Klein, Michael Knierman, Robert L. Moritz, Elliott J. Price, Jim Shofstahl, Julian Uszkoreit, Juan Antonio Vizcaíno, Mingxun Wang and Oliver Kohlbacher**

 # Equal contributions
** Corresponding author*

Abstract

Advancements in mass spectrometry (MS) instrumentation—including higher resolution, faster scan speeds, increased throughput, and improved sensitivity—along with the growing adoption of imaging and ion mobility, have dramatically increased the volume and complexity of data produced in fields like proteomics, metabolomics, and lipidomics. While these technologies unlock new possibilities, they also present significant challenges in data management, storage, and accessibility. Existing formats, such as the XML-based community standards mzML and imzML, struggle to meet the demands of modern MS workflows due to their large file sizes, slow data access, and limited metadata support. Vendor-specific formats, while optimized for proprietary instruments, lack interoperability, comprehensive metadata support and long-term archival reliability.

This white paper lays the groundwork for mzPeak, a next-generation data format designed to address these challenges and support high-throughput, multi-dimensional MS workflows. By adopting a hybrid model that combines efficient binary storage for numerical data and human-readable metadata storage, mzPeak will reduce file sizes, accelerate data access, and offer a scalable, adaptable solution for evolving MS technologies.

For researchers, mzPeak will enable faster (random) data access, enhanced interoperability across platforms, and seamless support for complex workflows, including ion mobility and imaging. Its design will ensure data is managed in compliance with regulatory standards, essential for applications such as precision medicine and chemical safety, where long-term data integrity and accessibility are critical.

For vendors, mzPeak provides a streamlined, open alternative to proprietary formats, reducing the burden of regulatory compliance while aligning with the industry's push for transparency and standardization. By offering a high-performance, interoperable solution, mzPeak positions vendors to meet customer demands for sustainable data management tools which will be able to handle emerging and future data types and workflows.

mzPeak aspires to become the cornerstone of MS data management, empowering researchers, vendors, and developers to innovate and collaborate more effectively. We invite the MS community to join the discussion on PREreview.org and collaborate in developing and adopting mzPeak to meet the challenges of today and tomorrow.

1. Introduction

The field of mass spectrometry (MS) has transformed significantly in the past decade, driven by rapid technological advancements that have pushed the boundaries of what is possible. Modern mass spectrometers are now capable of generating very large datasets in a fraction of the time required by earlier instruments. These advancements create exciting new opportunities in MS-based omics disciplines, but they also present significant challenges, particularly in the way data is stored and accessed. There is currently no universally effective format for efficiently managing and preserving the massive, complex datasets generated by these advanced instruments.

The two main options available today, mzML and vendor-specific formats, both have critical limitations. mzML [1, 2], the widely adopted community-driven standard [3] defined by the Proteomics Standards Initiative of the Human Proteome Organization (HUPO-PSI). mzML has been a cornerstone of MS data for over a decade, but it was not designed to handle the rapidly increasing scale and complexity of modern data. Its text-based structure results in large file sizes and slow data access, making it inefficient for high-throughput workflows. Binary vendor-specific formats, while optimized for their respective instruments, pose challenges of their own. They often require proprietary software for access, lack interoperability, and are not well-suited for long-term data preservation, creating barriers for data sharing and reuse. Additionally, evolving regulatory requirements in areas such as precision medicine (e.g., personalized molecular phenotyping) and chemical safety (e.g., environmental and human monitoring programs) increasingly demand robust solutions for long-term data and metadata storage—capabilities that are inadequately supported by both mzML and vendor-based formats.

This white paper introduces mzPeak, a next-generation data format aimed at addressing these challenges. Building on the lessons from mzML and binary formats, mzPeak will offer a future-proof solution tailored for modern MS workflows. It will improve storage efficiency, accelerate data access, and enhance interoperability across platforms and vendors, while aligning with evolving regulatory demands.

2. Challenges with current formats

Limitations of XML-based formats

The development of mzML was a significant milestone in MS data standardization, providing a flexible, human-readable format that greatly improved data exchange and interoperability at the time of its introduction. Designed as an XML-based format to be both human and machine-readable, mzML marked a departure from vendor-specific, proprietary formats that required specialized, often limited-access software to read. It addressed many challenges faced by the MS community more than a decade ago and has since served as a cornerstone for mass spectrometry data handling. However, the landscape of mass spectrometry has evolved dramatically, and mzML's text-based XML structure now introduces inefficiencies that are increasingly problematic in today’s high-throughput, multi-dimensional workflows.

One of mzML's design bottlenecks is that it is a text-based XML format, which makes file sizes much larger than necessary. See the supplementary material for further technical discussion. This file size expansion can be a significant burden, particularly in labs generating data in the terabyte range per instrument and month, with researchers often expected to store this data for more than 10 years or even indefinitely. Additionally, users rely on various cloud storage solutions, highlighting the need for open standards that are compatible with different storage options and support efficient, long-term data storage.

Moreover, mzML's reliance on XML makes it suboptimal for high-throughput environments where rapid data access is critical. Parsing large XML files is computationally expensive, and as datasets grow in size and complexity, this becomes a limiting factor in processing speed. Responding to these limitations, some computational groups have developed non-standard binary formats as intermediate data structures (**Supplementary Table 1**), but these formats lack interoperability, further complicating data management. The growth of mass spectrometry imaging (MSI), also called imaging mass spectrometry, as a data acquisition modality has also highlighted the need for a unified format that can handle the unique requirements of MSI data, including the large size of files, a large number of pixels, need for efficient access to either spectra or ion images, increasing use of ion mobility in MSI, as well as potential for including MS/MS or QQQ information.

Researchers working with high-resolution instruments, multidimensional ion mobility workflows, or MSI data are often hindered due to software supporting only part of the capabilities of their mass spectrometers, with some software not being able to load the files due to the memory requirements, or inefficient data access due to the legacy data structures used in the XML-based formats. Ultimately, these barriers limit the potential of mass spectrometry applications. This is particularly important for users without the in-depth knowledge and experience in programming who rely on existing software implementations. Furthermore, newer users expressed difficulty finding a well advertised, complete, and/or documented implementation of mzML/imzML or its validator had hindered its uptake and created problems both for the users as well as for software developers, emphasizing the need for robust support and documentation in future formats.

Limitations of vendor-specific formats

In contrast to mzML, vendor-specific formats are optimized for the performance of the instruments they support, tailoring data access and storage to specific instrumentation, acquisition parameters, and diagnostic metadata. However, these proprietary formats often lack long-term archival considerations and limit interoperability. Access to vendor formats typically requires restrictive licenses, as well as specific platforms, operating systems, or programming languages, which makes extended future data access uncertain. This restriction on accessibility poses significant risks for data preservation over time. In addition, for an academic developer of software it creates an additional burden to support software formats from multiple vendors and software distribution is hindered by vendors' license restrictions (e.g., redistribution restrictions).

By comparison, a unified community-supported format helps to abstract from this complexity and open formats facilitate reproducible science by ensuring long-term, license-free, one-entry-point and system-agnostic data access—all of which are indispensable for scientific

reproducibility. A universal open format like mzPeak is therefore essential to support seamless data exchange and reproducibility across diverse mass spectrometers, addressing these limitations effectively.

Lessons learned from mzMLb and other binary formats

The development of mzMLb represented an attempt to address some of the limitations of mzML, particularly around data compression and access speed (**Supplementary Table 1**) [4]. By using HDF5, a binary storage format, mzMLb reduced file sizes and improved data retrieval times, making it a more efficient option for handling large datasets. However, despite these technical improvements, mzMLb did not gain widespread adoption. One critical issue arising from the mzML specification; left unresolved by the conversion process; was the limited availability of metadata, which hindered its usability in workflows that depend on detailed experimental information, such as multi-dimensional and regulatory-driven studies. The lack of metadata integration reduced its usability, as researchers found it challenging to store and manage essential contextual information within mzMLb just as it was within mzML. Because the conversion still relied upon only the limited information the vendor libraries exposed and the implementation(s) were vastly complicated to modify, it was impractical to fix this limitation here in the MS data ecosystem. Furthermore, the community perceive mzMLb as offering limited additional value over mzML, as it did not address broader needs like vendor interoperability or regulatory compliance, while greatly increasing the technical burden to support it. These lessons emphasize the need for mzPeak to offer both improved data handling efficiency and robust metadata support to ensure broad utility and adoption across the MS community.

3. Our vision for mzPeak

A scalable, open solution

mzPeak will be designed to overcome the limitations of the aforementioned formats by adopting a hybrid model that combines efficient binary storage for numerical data and human-readable metadata storage. This hybrid approach ensures efficient storage and faster read/write times, making it well-suited for the vast and complex datasets generated by high-resolution, multi-dimensional workflows such as ion mobility spectrometry and MSI. By using Parquet, or a similar highly performant format widely adopted in high-volume data applications, mzPeak will ensure scalability without sacrificing flexibility. Selection of the underlying format is still in active discussion, see supplementary for details.

The format will use the HUPO-PSI Mass Spectrometry controlled vocabulary, which ensures that mzPeak will align with widely accepted terms and definitions. Designed with a natively binary format, mzPeak will enable random access to spectra, chromatograms, ion images, and mobilograms, ensuring fast and efficient data retrieval. Additionally, it will allow for lossless interconversion with vendor-based formats, preserving data integrity while ensuring compatibility across platforms. Released under an open license and free of patent restrictions, mzPeak will be designed to support both the immediate and future needs of the MS community, offering a robust foundation for managing the growing complexity of mass spectrometry data.

Comprehensive metadata

One of the critical shortcomings of older formats is their limited ability to store and annotate comprehensive metadata, particularly at sample and run levels. mzPeak will address this limitation by enabling detailed annotation of both sample characteristics and mass spectrometer configurations, while integrating the community-supported metadata standard SDRF-Proteomics [5], enabling detailed annotation of both sample characteristics and mass spectrometer configurations. This includes crucial information such as experimental conditions, run-specific parameters, and sample descriptions, ensuring the data can be fully utilized and interpreted across different MS platforms from various vendors.

By combining sample-level metadata with operational details, such as pump pressures across an LC gradient (even when not directly retrievable from vendor MS files), mzPeak will support seamless metadata annotation and export. These capabilities are essential for ensuring data comparability in public repositories and for meeting regulatory requirements, where complete and accurate metadata are critical for long-term usability and integrity.

Flexible and future-proof design

mzPeak will not just be a solution for today's challenges, but will be designed to accommodate the future evolution of mass spectrometry. It should have a flexible yet machine-readable structure that allows for the incorporation of new data types and workflows, ensuring that the format remains relevant even with ongoing technological advancements. It should be able to support recently emerged and yet emerging modes of data acquisition such as MSI and single-cell mass spectrometry analyses where the experimental set up and data structures can substantially differ from more traditional chromatography-based mass spectrometry used in bulk proteomics, metabolomics, or lipidomics. Therefore, this format can evolve alongside the MS field, supporting new analytical techniques, instrumentation, and data analysis workflows.

4. Benefits of mzPeak

For researchers

The adoption of mzPeak will bring immediate benefits to the research community, with faster data access being one of the most tangible improvements. By storing data in a binary format, mzPeak reduces the time required for search engines to retrieve large datasets, enabling quicker and more efficient analysis.

Interoperability is another major advantage. mzPeak will facilitate seamless data sharing between software platforms, breaking down barriers to compatibility and enabling more flexible data analysis. At the data level, proteomics, metabolomics, and lipidomics share substantial similarities apart from aspects like polarity, with differentiation largely driven by analytical and sample preparation techniques. This commonality ensures that mzPeak can serve these fields effectively while remaining adaptable to emerging -omics fields. Moreover, its interoperability is particularly beneficial for multi-omics studies, where the integration of data from proteomics,

metabolomics, and lipidomics is critical for providing a comprehensive view of biological systems.

Long-term data preservation is crucial in fields such as precision medicine and monitoring programs. mzPeak is designed with archival durability in mind, providing a stable and secure format that aligns with regulatory requirements for data retention and accessibility. By supporting robust metadata and standardized data structures, mzPeak ensures critical information remains intact and accessible, even as technologies and analytical platforms evolve. This ensures that data generated today will remain interpretable well into the future, enabling longitudinal studies and regulatory reviews without the risk of data degradation or incompatibility.

Additionally, mzPeak's ability to present all available vendor raw data, unlike current standards that often omit or fail to store certain instrument-specific details, makes it a more complete and efficient solution. This, combined with its open design, reduces the burden on public repositories such as Metabolomics Workbench, MassIVE, and EMBL-EBI's PRIDE and MetaboLights, contributing to more sustainable long-term storage and archiving practices. A side-effect of data submission in mzPeak would be easier access for web services to the majority of MS data in an archive instead of only those projects that deposit open formats alongside vendor raw files.

For vendors

For mass spectrometry vendors, mzPeak offers strategic advantages beyond just technical performance. The increasing focus on regulatory compliance, particularly in fields such as precision medicine (e.g. personalized molecular phenotyping), requires auditable data formats that can be archived for long periods while remaining accessible and usable and with an assurance of integrity. mzPeak addresses this need by providing a format that ensures both long-term data preservation and rapid access when needed.

By adopting an open standard, vendors can also reduce the costs associated with maintaining proprietary data formats. The transition to mzPeak allows vendors to focus on their core innovations while leveraging a community-driven standard for routine data management tasks.

5. Roadmap for mzPeak development

Community-driven collaboration

The success of mzPeak hinges on collaboration between the MS communities and key stakeholders. Formed to address the need for a new open file format, the mzPeak Committee has focused on archival and reanalysis needs for large-scale omics studies. The initiative has held round-table discussions at key conferences and conducted a community survey to gather insights. Moving forward, the Committee will continue collaborating with researchers, instrument vendors, and software developers to ensure that mzPeak aligns with the diverse needs of the MS ecosystem. HUPO-PSI, with its history of fostering community standards, will

play a critical role as well in guiding this development, with inclusion of other communities representing metabolomics, lipidomics, MSI, and single-cell MS.

Technical implementation

mzPeak will be released with reference implementations in multiple programming languages, to ensure broad accessibility for both researchers and vendors. To prevent issues like those seen with earlier formats, the development process will explicitly include a plan for several well-advertised, fully functional reference implementations from the outset. These implementations will include a validator to ensure compatibility with the mzPeak standard and facilitate seamless integration across different operating systems through cross-platform interoperability.

The data storage model will be natively binary, allowing for random access to spectra, chromatograms, ion images, and mobilograms, ensuring efficiency in handling large and complex datasets. Vendors will also be able to store essential technical metadata, including MS and LC settings and run-specific parameters. Each run will be stored as a single, self-contained file, providing a comprehensive archive of all data and metadata.

Additionally, the reference implementations will be designed to integrate with existing tools, minimizing disruptions to established workflows by providing converters or wrappers. This approach addresses one of the shortcomings of previous standards by prioritizing compatibility and encouraging adoption through developer-friendly tools and clear documentation.

All technical comments collected before the submission of this preprint are summarized in the **Supplementary notes**.

Adoption strategy

The adoption of mzPeak will be driven by both its technical advantages and strategic collaborations with key stakeholders. To ensure broad acceptance, strategic partnerships will be established not only with major MS vendors but also with prominent academic institutions and influential users who can advocate for the format's adoption. Regulatory agencies will also play a critical role by encouraging or requiring data deposits in standardized formats like mzPeak for compliance purposes. This multifaceted approach will position mzPeak as the default option for data storage and analysis. The goal is to create a format that is technically superior, aligned with regulatory and community needs, and widely supported across the MS community.

6. Conclusion

As mass spectrometry continues to evolve, so too must the tools we use to manage the data it generates. mzPeak is designed to address the challenges of today's high-throughput data environments while preparing the field for future advancements. By building on the lessons of mzML, mzMLb, imzML, and others, mzPeak strives to offer a scalable, future-proof solution that benefits researchers, vendors, and regulators. We therefore invite the MS community to

join the discussion on PREreview.org and collaborate in developing and adopting mzPeak, ensuring that we are equipped to meet the challenges and opportunities that lie ahead.

References

(1) 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. https://doi.org/10.1074/mcp.R110.000133.

(2) Deutsch, E. mzML: A Single, Unifying Data Format for Mass Spectrometer Output. *Proteomics* **2008**, *8* (14), 2776–2777. https://doi.org/10.1002/pmic.200890049.

(3) 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. https://doi.org/10.1021/acs.jproteome.2c00637.

(4) Bhamber, R. S.; Jankevics, A.; Deutsch, E. W.; Jones, A. R.; Dowsey, A. W. mzMLb: A Future-Proof Raw Mass Spectrometry Data Format Based on Standards-Compliant mzML and Optimized for Speed and Storage Requirements. *J Proteome Res* **2021**, *20* (1), 172–183. https://doi.org/10.1021/acs.jproteome.0c00192.

(5) Dai, C.; Füllgrabe, A.; Pfeuffer, J.; Solovyeva, E. M.; Deng, J.; Moreno, P.; Kamatchinathan, S.; Kundu, D. J.; George, N.; Fexova, S.; Grüning, B.; Föll, M. C.; Griss, J.; Vaudel, M.; Audain, E.; Locard-Paulet, M.; Turewicz, M.; Eisenacher, M.; Uszkoreit, J.; Van Den Bossche, T.; Schwämmle, V.; Webel, H.; Schulze, S.; Bouyssié, D.; Jayaram, S.; Duggineni, V. K.; Samaras, P.; Wilhelm, M.; Choi, M.; Wang, M.; Kohlbacher, O.; Brazma, A.; Papatheodorou, I.; Bandeira, N.; Deutsch, E. W.; Vizcaíno, J. A.; Bai, M.; Sachsenberg, T.; Levitsky, L. I.; Perez-Riverol, Y. A Proteomics Sample Metadata Representation for Multiomics Integration and Big Data Analysis. *Nat Commun* **2021**, *12* (1), 5854. https://doi.org/10.1038/s41467-021-26111-3.

Supplementary

Technical comments collected before preprint v01

Limitations of XML-based formats

XML repeats structural information explicitly while offloading semantics to a shared schema, instead of both structure and meaning, to provide a degree of human readability. XML is also inefficient for storing large numerical datasets, either using two to ten times more space to encode 32-bit float imprecisely but human readable, or using 33% more space to base64 encode the numerical bytes exactly. Even with optional compression, mzML files still end up being four to eighteen times larger than the original proprietary formats [4] but also prevents random access to the data using commonly available tools.

A scalable, open solution

One of the core developments for mass spectrometry in recent years has been the addition of further dimensions of separation or analysis. The former is seen in the explosive growth of ion-mobility data and the latter is seen in the rise of imaging MS. Looking to the future mzPeak needs to be designed in such a way that future developments that would further increase the dimensionality of our data do not break or indeed require additions to the format itself.

There is still active discussion as to what format mzPeak will use for serialization. Parquet is mentioned by name in this document as an example and should not be construed as constituting a final technical decision. In choosing what serialization format we are going to use for mzPeak, we need to keep in balance the specificity of the library to handle the data types that we are working with, versus the level of general support that the format has in the wild.

Parquet, for example, would be a good choice, as there are lots of use cases for it across a wide variety of different areas of data science, and as a result, there are lots of supporting implementations out there. There are at least three different independent compatible reference implementations in C++, Java, Rust, and it's good at interoperability with other languages.

The predicted longevity of the container format is also important. We need to make sure that whatever decision we make today, there is going to be continued maintenance on that library through the predicted lifespan of mzPeak. Again, using Parquet as an example, Parquet has good support through the Apache Foundation, which is indicative of good long-term survivability.

Another consideration is accessibility, because Parquet is implemented across multiple languages, it's easier for users to read the raw container file, and therefore easier for us to implement a reader on top of it. Compared to a case like HDF5 where if you didn't have access to the C library because A) no C ABI or B) can't use the must-have plugins because of issues with linking stage of compilation, an implementer may have to try to use a partial reimplementation of the format with 8 different kinds of strings.

On the other hand Parquet is not obviously the best solution from a strictly technical perspective. Parquet is a column-based data format, so what are the columns here? Instead, some multi-dimensional tensor format seems appropriate, particularly for the raw data where we have at least two arrays or dimensions per spectrum and often more with ion mobility.

A tensor is just a strided/nested array with more than two dimensions. Nothing stops us from representing one as either an in-row list or an unzipped long table with all the metadata with run-length encoding (RLE). There are costs associated with either decision, there's no free lunch after all, but this price is paid either once per dimension per row (wide), or once per run group per page (long). These choices each introduce trade offs that impact how the file would be read, and how different tools might interact with them. The same constraints apply to the tensor format when it comes time for compression and storage, but it may be able to make certain assumptions during the layout of data in the byte stream to reduce the costs in space or access time. These are examples of more general array storage formats which, due to the recentness of their emergence in the field, have only a single implementation, often governed by a single institution or corporation, and may not even have any formal guarantee of stability of the container file format. For example, DuckDB's file format would satisfy most of the same needs that Parquet would while adding many other desirable capabilities like multiple logical tables per file and search indices, only recently announced their serialization format would be stable going forwards, but there remains only the one canonical implementation of the reader written by DuckDB themselves which they provide bindings to for most major languages. We greatly hope that DuckDB will be with us to welcome the year 2050 along with SQLite, but there are no guarantees.

Parquet's main weaknesses are granular random access and storage heterogeneity. Storage heterogeneity is usually addressed with multiple files or by schema partitioning. Schema partitioning here means each row in the table is a structure containing *n* distinct types of nested nullable structures, one for each type of entity we wish to store, e.g. spectrum, chromatogram, custom index block, et. cetera, and then organize the file so that most rows containing the same type of substructure are grouped together for RLE to compress away the other branches. While our hope is that we can achieve our goals while storing all data in a single file, it may be necessary to split a Parquet file across ordering dimensions to achieve the desired performance as Parquet does not itself encode out-of-stream sort indices. Random access is the bit that can't be fixed without ahead-of-time planning on the schema, the sorting order, and expected anticipation of access patterns. Parquet uses blocked and compressed storage, which means that in order to read a single value, all the values in the block must be decompressed together. The smaller the block, the less efficient the compression but the faster that block can be decompressed on average. While block sizes can be tuned, there are limits and they cannot be configured granularly per column. The canonical way to address this problem with Parquet is to split (and sort) files so that you are more likely to want to read whole blocks at a time, and so that block-level statistics make it easier to filter out blocks entirely. This is undesirable for a format intended to be uploaded/downloaded with conventional web browsers that cannot operate on directories as a unit.

Exactly what's in the schema is undecided yet. We could end up mirroring the mzML schema very easily with a few recurring CVparams burned in as columns, or something resembling mzTab, depending upon what works best. Many facets of the mzML schema were designed

for use cases that never emerged, or required a degree of nesting that was not useful in practice.

Parquet as a container format checksums each data page for integrity checking and granular file encryption to protect sensitive information and prevent tampering.
https://parquet.apache.org/docs/file-format/data-pages/checksumming/

https://parquet.apache.org/docs/file-format/data-pages/encryption/

Table 1. Comparison of existing interchange formats

Format Name	Container Format	Implementations	Data-Metadata disposition	Metadata Format	Ion Mobility Compatibility	Uses mzML Data Model	Reference
mzML	XML	Multiple (https://www.psidev.info/mzml)	In-band	XML Text	Compatible	Yes	[1]
mz5	HDF5	C++, R (via C++)	In-band	Binary Structures	Not compatible	Yes	[2]
mzMLb	HDF5	C++, Python, Rust, R (via C++)	Out-of-band	XML Text	Compatible	Yes	[3]
imzML	XML + Custom binary (two different files)	C++, R, Python, Java	Out-of-band	XML Text	Not compatible	No	[4]
mzDB	SQLite3	C++, Java	In-band, Replicated out-of-band Index	XML Text	Not compatible	Yes	[5]
Toffee	HDF5	C++, Python	Out-of-band	XML Text	Required	No	[6]
mzA	HDF5	Python	Out-of-band	Binary Structures	Compatible	No	[7]

StackZDPD	Custom binary + JSON	C#, Java	Out-of-Band	JSON	Compatible	No	[8]
mzTree	Custom binary (R-Tree based)	Java	In-Band	SQLite	Not compatible	No	[9]

Supplementary References

1. Martens, Lennart, Matthew Chambers, Marc Sturm, Darren Kessner, Fredrik Levander, Jim Shofstahl, Wilfred H. Tang, et al. “mzML--a Community Standard for Mass Spectrometry Data.” *Molecular & Cellular Proteomics: MCP* 10, no. 1 (January 2011): R110.000133. https://doi.org/10.1074/mcp.R110.000133.
2. Wilhelm, Mathias, Marc Kirchner, Judith A. J. Steen, and Hanno Steen. “Mz5: Space- and Time-Efficient Storage of Mass Spectrometry Data Sets *.” *Molecular & Cellular Proteomics* 11, no. 1 (January 1, 2012). https://doi.org/10.1074/mcp.O111.011379.
3. Bhamber, Ranjeet S., Andris Jankevics, Eric W. Deutsch, Andrew R. Jones, and Andrew W. Dowsey. “mzMLb: A Future-Proof Raw Mass Spectrometry Data Format Based on Standards-Compliant mzML and Optimized for Speed and Storage Requirements.” *Journal of Proteome Research* 20, no. 1 (January 1, 2021): 172–83. https://doi.org/10.1021/acs.jproteome.0c00192.
4. Schramm, Thorsten, Zoë Hester, Ivo Klinkert, Jean-Pierre Both, Ron M. A. Heeren, Alain Brunelle, Olivier Laprévote, et al. “imzML--a Common Data Format for the Flexible Exchange and Processing of Mass Spectrometry Imaging Data.” *Journal of Proteomics* 75, no. 16 (August 30, 2012): 5106–10. https://doi.org/10.1016/j.jprot.2012.07.026.
5. Bouyssié, David, Marc Dubois, Sara Nasso, Anne Gonzalez de Peredo, Odile Burlet-Schiltz, Ruedi Aebersold, and Bernard Monsarrat. “mzDB: A File Format Using Multiple Indexing Strategies for the Efficient Analysis of Large LC-MS/MS and SWATH-MS Data Sets.” *Molecular & Cellular Proteomics: MCP* 14, no. 3 (March 2015): 771–81. https://doi.org/10.1074/mcp.O114.039115.
6. Tully, Brett. “Toffee - a Highly Efficient, Lossless File Format for DIA-MS.” *Scientific Reports* 10, no. 1 (June 2, 2020): 8939. https://doi.org/10.1038/s41598-020-65015-y.
7. Bilbao, Aivett, Dylan H. Ross, Joon-Yong Lee, Micah T. Donor, Sarah M. Williams, Ying Zhu, Yehia M. Ibrahim, Richard D. Smith, and Xueyun Zheng. “MZA: A Data Conversion Tool to Facilitate Software Development and Artificial Intelligence Research in Multidimensional Mass Spectrometry.” *Journal of Proteome Research* 22, no. 2 (February 3, 2023): 508–13. https://doi.org/10.1021/acs.jproteome.2c00313.
8. Wang, Jinyin, Miaoshan Lu, Ruimin Wang, Shaowei An, Cong Xie, and Changbin Yu. “StackZDPD: A Novel Encoding Scheme for Mass Spectrometry Data Optimized for Speed and Compression Ratio.” *Scientific Reports* 12, no. 1 (March 30, 2022): 5384. https://doi.org/10.1038/s41598-022-09432-1.
9. Handy, Kyle, Jebediah Rosen, André Gillan, and Rob Smith. “Fast, Axis-Agnostic, Dynamically Summarized Storage and Retrieval for Mass Spectrometry Data.” *PloS One* 12, no. 11 (2017): e0188059. https://doi.org/10.1371/journal.pone.0188059.