PSI Recommendation
PSI Mass Spectrometry and Proteomics Informatics Working Groups
Status: DRAFT

Henry Lam, The Hong Kong University of Science and Technology
Tytus D. Mak, National Institute of Standards and Technology
Joshua Klein, Boston University
Wout Bittremieux, University of Antwerp
Ralf Gabriels, VIB-UGent Center for Medical Biotechnology
Yasset Perez-Riverol, European Molecular Biology Laboratory
Tim Van Den Bossche, VIB-UGent Center for Medical Biotechnology
Juan Antonio Vizcaíno, European Molecular Biology Laboratory
Eric W. Deutsch, Institute for Systems Biology

May 23, 2025

mzSpecLib: Mass Spectral Library Format

Status of this document

This document provides information to the proteomics community about a standardized spectral library file format for proteomics and other fields that use mass spectra. Distribution is unlimited. This Recommendation was ratified via the PSI Document Process. Any alterations of this document MUST also follow the HUPO PSI Document Process.

Version 1.0

Abstract

The Human Proteome Organization (HUPO) Proteomics Standards Initiative (PSI) defines community standards for data representation in proteomics to facilitate data comparison, exchange, and verification. This document presents a specification for a format for storing mass spectrometry (MS) spectral libraries, which are collections of mass spectra extracted from the context of the original MS runs for archival or reference purposes. These spectral libraries are often -- but not necessarily -- representative spectra sourced from replicates and accompanied by interpretations for which analyte yielded each spectrum. The specification focuses on the data model and mechanism for providing controlled vocabulary (CV)-based metadata, while leaving open the serialization mechanism to several interchangeable possibilities. Further detailed information,

including any updates to this document, implementations, and examples is available at http://psidev.info/mzSpecLib.

Table of Contents

1. Introduction

1.1 Description of the need

Spectral libraries are stored collections of previously observed mass spectra and information about what is known about them as used in the fields of proteomics and metabolomics. Commonly, each spectrum entry in the spectral library will contain the spectrum itself, information about the origin of the spectrum, and information about the putative analyte or analytes that produced the spectrum. In some cases, the source analytes may not be known. Spectral libraries are then generally used to drive or inform subsequent analyses of new data.

There have been for many years several different popular spectral library formats in proteomics, including the MSP format used by the United States (US) National Institute for Standards and Technology (NIST), the splib format used by SpectraST[1], the blib format used by Bibliospec[2], and others. While these formats are adequate for the storage of spectra, they all lack an adequate and well-documented mechanism for providing rich metadata about the spectra, including their provenance. As spectral libraries become ever more important for mature and emerging applications, an acute need for a community standard spectral library format with extensive metadata has been identified[3].

1.2 Requirements

The primary requirements of the PSI standard spectral library format are:

- Be suitable for archiving and exchange of mass spectra and their metadata.
- Enable encoding of metadata at the library level, the spectrum level, the analyte level, the peak level, and the peak annotation level.
- Use a controlled vocabulary (CV)-based mechanism for encoding metadata at the library level, spectrum levels, and analyte level.

- Provide a suitable data model that can readily be serialized as a plain text format, an object notation format such as JSON, and more complex relational or hierarchical formats.
- Primarily address the needs of the proteomics mass spectrometry community that is defining this format, but also be suitable as a base for the metabolomics, glycomics, and lipidomics mass spectrometry fields if there is a desire to adopt this format.
- Enable the encoding of consensus spectra, merged from several source replicates with provenance information.
- Enable the encoding of spectra and metadata even when the identification of the source analyte is not known or ambiguous (spectral archives).
- Enable encoding of predicted spectra
- Enable the encoding and annotation of spectra that source from the combination of several separate or linked molecules.
- Provide a data model that is suitable for exchanging well-annotated single spectra via web services or similar applications.
- Undergo a formal standardization process to gain broad community acceptance.

2. Notational Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in RFC 2119.[4] In general, “MUST” means required, “SHOULD” means recommended, and “MAY” means optional.

3. The Spectral Library Format Definition

3.1 The documentation

The documentation of the mzSpecLib format consists of four major parts:

1. This specification document
2. CV terms in the PSI-MS CV
3. Additional information in the GitHub repository https://github.com/HUPO-PSI/mzSpecLib, and additional formatted documentation (guidelines and best practices) at https://hupo-psi.github.io/mzSpecLib/guidelines/index.html
4. A reference implementation in Python can be found at https://github.com/HUPO-PSI/mzspeclib-py.

3.2 Relationship to other specifications

The specification described in this document is not being developed in isolation; indeed, it is designed to be complementary to, and thus used in conjunction with, several existing and emerging models. Related specifications include the following:

- PSI Universal Spectral Identifier (http://psidev.info/usi) The PSI Universal Spectrum Identifier (USI)[5] describes a virtual path to locate a spectrum plus a

possible interpretation of that spectrum. USIs may be used within a spectral library to refer to the origin of spectra contained therein.
- mzPAF (http://psidev.info/mzPAF). The PSI standardized format for the annotation of fragment ion peaks. mzSpecLib requires that fragment ion peaks be annotated with mzPAF, if they are annotated at all.
- PROXI (http://psidev.info/proxi). The Proteomics Expression Interface being developed by the PSI is a standardized API (Application Programming Interface) by which mass spectrometry proteomics information can be exchanged. Provision of individual spectra via PROXI is being made using the mzSpecLib data model.
- mzML (http://psidev.info/mzML). mzML[6] is the PSI open data standard for MS data and is a common source for spectral libraries.
- ProForma 2.0[7] (Proteoform and Peptidoform Notation) Specification (http://psidev.info/proforma). This specification describes a standardized way of encoding a peptidoform and will be used in mzSpecLib for this purpose.

3.3 General concepts for a spectral library

3.3.1 Purpose of the spectral library format

Historically the PSI has developed open and standardized data formats that have focused on storing mass spectra as generated by an instrument (the mzML format), and formats that have focused on storing the downstream identifications of those generated mass spectra (the mzIdentML and mzTab formats). These formats have been kept separate intentionally, with the identification formats designed to refer to the spectrum-containing format. This was done primarily to control file sizes, since the serialization of spectra takes substantial space, and replicating in the identification format seemed inefficient.

Spectral libraries are designed with an opposite approach: to unite the spectra and their interpretations into a single format. The significant difference, however, is that spectral libraries are intended to contain single or aggregated spectra that are deemed important, collated across potentially many experiments to form a reference set of spectra that may be used for downstream processing. Spectral libraries are not intended to capture the complex output of spectrum processing pipelines (for which mzIdentML is designed), but rather capture collections of annotated spectra that may be later used as a reference. The distinction is subtle and in principle the spectral library format could be used to capture experimental level output, although it is not designed for this.

3.3.2 Supported applications and design principles of the mzSpecLib format

Spectral libraries are primarily compiled as references for assigning molecular identifications to tandem mass spectra. As the fragmentation pattern of an analyte under similar experimental conditions is reproducible, a good spectral match to a library spectrum can be taken as evidence that the unknown (query) spectrum at hand comes from the same analyte. This is the basis of spectral library searching, a well-established method for spectrum identification, both for peptides and small molecules. However,

beyond spectral library searching, spectral libraries are also important resources for other data analysis workflows. In data-independent acquisition (DIA) experiments, for example, spectral libraries play a crucial role in peptide-centric approaches that look for anticipated fragment ions co-eluting over time. Spectral libraries are also useful for designing MS-based assays in targeted quantitative workflows such as selected reaction monitoring (SRM).

Besides supporting identification workflows, spectral libraries have also become a general form of data organization and a convenient mechanism for data re-use and data sharing. Unlike unprocessed experimental data, spectral libraries put the spectra and their identifications in the same place, often with various processing to distill the essential information and reduce redundancy. This makes spectral libraries the ideal data format for long-term data storage, as well as information hubs for data sharing. For this purpose, spectral libraries containing unidentified spectra (potentially organized by spectral similarity), called "spectral archives," can also be compiled.

In all these applications, it is paramount that spectral libraries contain various essential metadata, in addition to the spectra themselves. These include the library version, origins and provenance of the library spectra, the experimental conditions under which they are acquired, the method for identifying the spectra and any associated confidence metrics, annotations of the observed peaks, among others. Existing spectral library formats may also allow for storing some metadata, but these are often encoded in a haphazard and ill-defined manner, resulting in poor backward compatibility and interoperability.

To support these applications, the mzSpecLib format is designed with the following principles. First, the format should take the form of a common data model with multiple inter-convertible serialization that serve different needs. Second, metadata must be encoded with terms in an actively maintained controlled vocabulary (CV), namely the PSI-MS CV. In addition, to ensure flexibility and extensibility, mzSpecLib adopts an "open" metadata set: all terms defined in PSI-MS are legal to use. This ensures that newer concepts can be described as the field evolves, without the need to revise the format. Finally, to set a manageable scope, the present format will support primarily proteomics applications, with the expectation that it can be extended to other MS fields in the future.

3.3.3 Precepts for multiple analytes per spectrum

Past spectral library formats generally only allowed for one analyte interpretation for each spectrum. It is generally the goal of spectral library reference spectra that they only contain peaks from the intended analyte. However, in practice, many if not most spectra fall short of this ideal and contain peaks from more than one analyte. This mzSpecLib format attempts to capture this reality by enabling the encoding that multiple analytes may contribute to a single spectrum, either as a mixture of unrelated ions (a chimeric spectrum) or as a mixture of related or linked ions (as in the case of a spectrum of two crosslinked peptides). The analyte section of this specification attempts to capture this

information. Analytes are optional and may not be provided at all for unidentified spectra (See Section 3.3.4 below).

An alternative scenario is that a spectrum may have more than one possible interpretation. The format also supports multiple interpretations per spectrum, in addition to multiple analytes per interpretation. This is substantially different from previous spectral library formats that generally allowed for only one analyte per spectrum. Existing software may be unprepared to handle the complexities of multiple analytes per spectrum and multiple hypotheses per spectrum, and must be adapted to this possibility, even if it is not fully supported at first.

3.3.4 Unidentified spectra and spectral archives

Traditionally, spectral libraries are compilations of identified spectra, namely, spectra that have been assigned confidently to one or more analytes. However, applications also exist for libraries that also contain unidentified spectra (exclusively or not). Such spectral libraries are sometimes referred to as spectral archives[8]. For spectral archives to be useful, spectra are often grouped by spectral similarity into clusters by some clustering algorithm. It is often, but not always, assumed that spectra in a cluster originate from the same analyte, or from highly related analytes, on account of their spectral similarity. To support spectral archives as an extension of spectral libraries, mzSpecLib allows for the specification of spectrum clusters in the library header (see Section 4.1.5), as well as the cross-referencing of similar and/or related spectra in each library entry (see Section 4.1.6).

3.3.5 Predicted spectra

Recently, it has become possible to predict the fragmentation patterns of various analytes with reasonable accuracy. Therefore, it is anticipated that libraries of predicted spectra may also be a substantial use case for this format. Apart from a specification of the origin of the spectrum, predicted spectra can be encoded in the same way as experimentally observed spectra. More detailed information about the methods of prediction, as well as any quality and confidence metrics associated with the prediction, can be further added in the future, using newly defined CV terms as needed.

3.4 Overview of mzSpecLib data model

The mzSpecLib specification primarily takes the form of a data model, with multiple possible serializations of that data model. This specification describes a text-based serialization and a JavaScript Object Notation (JSON) serialization. Other, more efficient serializations are envisioned and are in progress. Lossless interconversion between these serialization methods is provided by a reference implementation Python library (https://github.com/HUPO-PSI/mzSpecLib/tree/master/implementations/python), and other implementations are welcome and forthcoming.

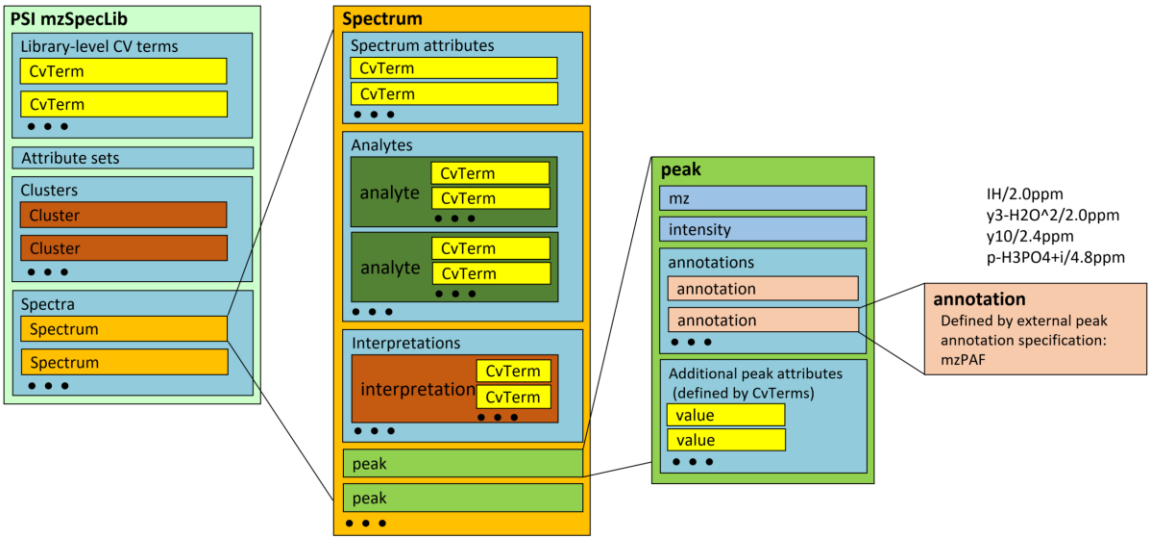

Figure 1. Overview of the mzSpecLib data model. The model consists of six main components: the library metadata, clusters, spectra, analytes, peaks, and the peak annotations.

The data model as depicted in Figure 1 consists of six main components. The top level contains basic metadata describing the library itself. At this level, attribute sets can be defined for entry-level metadata that are shared by many library entries. Optionally, a list of spectrum clusters follows, which consists of references (links) to the library entries belonging to each spectrum cluster.

The next component describes each individual spectrum entry, both the metadata about each spectrum as well as a list of peaks. Spectrum metadata describes attributes inherent to the spectrum, and is accompanied by metadata that describe the one or more putative analytes that produced it, and metadata describing the interpretations that combine one or more of the putative analytes.

The next component describes each peak, including the m/z, intensity, and aggregation metadata (for when a spectrum is an aggregated spectrum and aggregation metrics are available).

The final component describes the one or more proposed annotations for each peak. This information is encoded in the mzPAF peak annotation format, as described in a separate specification found at http://psidev.info/mzPAF.

An indexing mechanism is explicitly not specified here since it would be different for different serializations and applications. Some applications may want indexing by retention time, mass modifications, or proteins, while others would have no interest in indexes by such attributes. Tool developers are recommended to generate sidecar indexes as appropriate for their applications.

Each of these components is described in detail below.

3.4.1 Library metadata

Library-level metadata provide information about the origin of the library, such as who is the producer, when was it produced, which software was used to produce the library, and which datasets were used to build it. Library-level metadata elements are encoded as CV terms from the PSI-MS CV.

Note that spectrum metadata (next subsection) may also be provided at the library metadata level to provide inherited defaults for all or some entries in the library via the specification of "attribute sets" For example, if all or most of the spectra in a library originate from a single instrument, that instrument and any related information may be specified in an attribute set as a library-level metadata, and inherited by all library entries claiming that attribute set. For details of the mechanism, Section 4.1 describes the intended implementation in the plain-text serialization as an example.

3.4.2 Cluster metadata

In order to support the use case of spectrum clustering, where similar spectra are grouped as being proposed to originate from the same precursor ion, usually in the absence of a confident identification, the Cluster metadata section allows the encoding of which spectra are to be grouped as clusters. Cluster metadata elements are encoded as CV terms from the PSI-MS CV.

3.4.3 Library spectrum metadata

Spectrum metadata provide information about each individual library spectrum, primarily about its origin and observable attributes about it (such as its precursor m/z or total ion current), independent of the putative analyte. These attributes may be provided for each spectrum separately, or they are inherited from an attribute set as described above. There is also a mechanism to cross-reference another spectrum within the same library or external to the library within this section. Spectrum metadata elements are encoded as CV terms from the PSI-MS CV.

3.4.4 Analyte metadata

Analyte metadata provide information about the putative identity of the analyte or analytes that yielded the described spectrum. In mass spectrometry, the analyte is formally an ion, which is defined by the molecular structure and the charge state. The molecule may be a biomolecule (e.g., peptide, protein, glycan, lipid, or other metabolites), or a chemical compound. This information is optional and may not be provided at all in cases of libraries that contain unidentified spectra (spectral archives).

3.4.5 Interpretation metadata

The assignment of the spectrum to one or more analytes is referred to as an interpretation of the library spectrum. In the case of peptides, this is sometimes termed a peptide-spectrum match (PSM). Interpretation metadata provides information about this assignment and the process by which the assignment is made, such as search engine used, confidence scores, etc. The format allows for multiple possible interpretations per spectrum such that only one of several alternative interpretations is presumed to be correct but the spectrum does not provide enough information to make a confident decision between the alternative interpretations. Additionally, each alternative interpretation MAY also contain several interpretation members such that all members are present, usually in the case of a chimeric spectrum. Please refer to Section 4.1 for an example for a plain-text serialization of the interpretation and interpretation member metadata.

3.4.6 Peak metadata

Peak metadata provide information about the observable characteristics of a peak in the spectrum, namely its m/z and intensity. A mechanism also exists to include other peak metadata besides its m/z, intensity, and annotation in a customized manner. For example, aggregation information can be specified if the described spectrum is a representative spectrum derived from a set of replicate spectra. Please refer to Section 4.1 for an example for the plain-text serialization.

3.4.7 Peak annotation metadata

Peak annotation metadata provide information about the identity of the proposed fragment ion that yielded the peak. There may be several proposed interpretations for a peak, either as alternatives or as multiple contributors. Peak annotation metadata can be quite complex and have a special encoding to keep it compact. These annotations differ on whether the analyte is a peptide, a small molecule, or a different type of analyte, and different encoding schemes are required for these different analyte types. The mzPAF peak interpretation format for peptides has been developed as a separate PSI standard, since it has broader applicability than spectral libraries, and should be used here in mzSpecLib to annotate spectrum peaks. See the mzPAF specification (http://psidev.info/mzPAF) for complete information on this format.

4. mzSpecLib serializations

The same mzSpecLib data model is intended to be applicable for both communicating individual or small sets of spectra via web services as well as for large spectral libraries. For this reason, the community proposed several serializations for the mzSpecLib data model, each suitable for one or more applications. Software is provided for reading and for interconversion between these different serializations. The current primary serializations are described below. Additional serializations that are optimized for

different applications are likely and encouraged as deemed necessary as long as they follow the above data model, and thus make interconversion easy.

4.1 Plain text serialization

The plain text serialization is based loosely on the NIST MSP format and SpectraST sptxt formats, with many important differences. Its primary goal is to be human readable with less emphasis on compactness and efficiency. It is suitable for smaller libraries where human readability is a desired aspect. An implementation-specific external sidecar index can provide fast access into the library if necessary, but is not part of this format and should not be included in the same file. Empty lines (or lines consisting only of whitespace characters) are permitted throughout the file for readability, but not required and are to be ignored by readers. A text serialized mzSpecLib file SHOULD have the extension .mzSpecLib.txt.

4.1.1 Beginning of the file

The plain-text serialization of a spectral library MUST begin with the line:

```
<mzSpecLib>
```

and is followed by a series of library-level controlled vocabulary terms. The first CV term MUST be the mzSpecLib format specification version number used by the writer of the file in the form:

```
MS:1003186|library format version=1.0
```

4.1.2 Encoding of controlled vocabulary terms

CV terms MUST be encoded as follows with the three components (optional group designator, subject, and object):

- Optional group designators must be provided in square brackets. These must be integers, unique within each CV attributes section (e.g. Spectrum, Analyte, etc.) (e.g. [1], [2]). All terms prepended with the same group designator within a CV section are to be interpreted as belonging to the same group.
- The subject CV term compact URI identifier and name, separated with a | character (e.g. MS:1003186|library format version) MUST be followed WITHOUT spaces by an = character and then the object. The subject MUST never have an = character in it, and thus the first = character in a line is always assumed to be the delimiter.
- If the subject CV term does not have a data type according to the CV, then the object after the = character MUST be a single CV term in the same format as the subject (e.g. NCBITaxon:9606|Homo sapiens)
- If the subject CV term DOES have a data type according to the CV, then the object after the = character MUST be a single numerical value, a string, or another complex data type (e.g. list of integers) following the data type in the CV.

- It is permitted for the object to be empty. It MAY be interpreted as an empty string or a null value.
- If the object is a string, it MAY contain an = character, but the subject-object delimiter is always the FIRST = character.
- If the object is a string, all quotation marks are to be interpreted as part of the string, not as a container of the string.
- If the object data type is “list of strings”, “list of integers” or “list of floats”, the values MUST be comma separated without spaces.
- The object must match the data type defined in the PSI-MS CV
- The object MUST NOT be a list of numbers UNLESS the PSI-MS CV explicitly defines the “has_value_type” of the subject as a list (e.g. MS:1002712|list of integers).
- All numerical values MUST be expressed with a decimal point for decimal notation instead of a comma. Numerical values MUST NOT use any thousands separator.
- CV Terms SHOULD come from the following ontologies/CVs if possible:
    - PSI-MS
    - Unit Ontology
    - NCBITaxon

Examples:

```
MS:1000888|stripped peptide sequence=DSDDVPMVLVGNKCDLAAR
MS:1001112|n-terminal flanking residue=K
MS:1000044|dissociation method=MS:1002472|trap-type collision-induced dissociation
MS:1003208|experimental monoisotopic precursor m/z=880.8902
[1]MS:1003209|monoisotopic m/z deviation=0.333
[1]UO:0000000|unit=UO:0000169|parts per million
```

The final two example lines show an example where the two terms are together in the same group, since the delta m/z and the units must be interpreted together.

Requirements:

- Subject CV terms MUST be provided as the accession with a CV prefix and the reference identifier in the CV separated by a colon, followed by a | character and the exact name of the CV term as provided in the CV.
- If the exact name of the CV term contains an = character (e.g., for regular expressions of proteases), then the name MUST be enclosed in double quotation characters ("xxxx").
- There MUST be no spaces around the following = character.
- If the object is a CV term, it MUST follow the same convention as the subject.
- If the object CAN be a CV term, it MUST be a CV term.
- The PSI-MS CV MUST be used if the concept is present therein; other CVs MAY be used if not.
- For PSI-MS CV terms, a free-text subject value MUST be used if and only if the PSI-MS CV term explicitly allows a value (as defined in the CV with “has_value_type”). Most other CVs do not define whether a term can take on a value and are exempt.

- If a suitable CV term for the desired concept cannot be found, then the term should be requested.

4.1.3 Library CV terms

The first set of CV terms in the file are assumed to be library-level CV terms (initiated with "MS:1003186|library format version" as described in 4.1.1). For example, attributes that describe the library as an entity are listed here without a preceding header (see Metadata file for commonly used terms):

```
MS:1003186|library format version=1.0
MS:1003187|library identifier=ecoli_hcd_consensus
MS:1003188|library name=NIST E. coli HCD consensus library
MS:1003189|library description=Consensus library of HCD spectra from E.
coli tryptic peptides compiled by the National Institute of Standards
and Technology, USA
MS:1003190|library version=2020
MS:1001017|release date=2020-06-01
MS:1003198|copyright notice=...
…
```

(Note that the 4th line in the example above is a single line in the file, although it is shown as wrapped in this document)

The section of library CV terms ends when a new section begins, specified with an angle bracket, e.g. <Spectrum=_>, <AttributeSet Spectrum=_>, <Cluster=_>.

4.1.4 Attribute set definitions

If all, or a significant subset of, spectra in a library share some common spectrum-level properties, repeating them in every entry may be burdensome, especially if there are more than one such subsets in the library. To reduce clutter and facilitate the construction of heterogeneous libraries, attribute sets can be defined in the library header, and claimed by the individual spectra to which such attribute sets apply. As a common use case, when multiple libraries of different types of spectra are combined (such as a target library and its associated decoy library, libraries acquired from different species or different instrument settings), attribute sets can be defined for each type.

Attribute sets, if any, SHOULD be defined immediately following the library-level metadata, denoted by one of the following:

```
<AttributeSet Spectrum=S>
<AttributeSet Analyte=S>
<AttributeSet Interpretation=S>
<AttributeSet Cluster=S>
```

where the S is replaced by a string of alphanumeric (characters - and _ are allowed) text naming the set. The name "all" is reserved for the attribute set that is applied to all entries in the library. Four types of attribute sets are allowed: Spectrum metadata MUST be

listed under <AttributeSet Spectrum=S>. Analyte metadata MUST be listed under <AttributeSet Analyte=S>. Spectrum interpretation metadata MUST be listed under <AttributeSet Interpretation=S>. Cluster metadata MUST be listed under <AttributeSet Cluster=S> (see Metadata table).

The CV terms listed here will be applied to each library entry that has the "MS:1003212|library attribute set name" attribute specified whose value matches the attribute set's name (or all entries in the library in the case of "all").

Multiple attribute sets (of same or different types) MAY be defined sequentially, but cannot be added after the first Spectrum is defined.

Nested attribute sets are not allowed.

If a library entry claims multiple attribute sets of the same type, the union of all the CV terms of those attribute sets will be inherited and applied to that library entry. If the claimed attribute sets contain the same CV term, the instance(s) in the last-referenced attribute set will prevail. If the same term is specified in the library entry itself, its value will override any corresponding one(s) specified in the attribute set(s). When multiple instances of the same attribute are present in an attribute set, they are treated as an inseparable group, and thereby inherited and overridden together.

See section 4.1.12 for detailed examples of defining and using attribute sets.

4.1.5 Spectrum Cluster section

To support spectral archives, a section in the library header can be added to specify spectrum clusters of similar spectra. The mechanism is to refer to each spectrum by its unique library entry key if that spectrum is in the same library (preferred), or by its Universal Spectrum Identifier (USI) if the spectrum is stored externally. Since a cluster represents a group of spectra, there MUST NOT be Analyte or Interpretation sections within <Cluster>. In case a cluster member is identified to some analyte, that information should be specified within the <Spectrum> section of that cluster member. It is also possible to refer to a "consensus" spectrum or a "best representative" spectrum of the cluster.

An example is below (note that some lines are wrapped in this display):

```
<Cluster=N>
MS:1003320|spectrum cluster size=6
MS:1003268|spectrum cluster member keys=1,6,23,63,89
MS:1003269|spectrum cluster member
USI=mzspec:PXD000561:Adult_Frontalcortex_bRP_Elite_85_f09:scan:17555
[1] MS:1003321|summary statistics of clustered
spectra=MS:1003304|spectral dot product
[1] MS:1003176|attribute mean=0.7
[2] MS:1003321|summary statistics of clustered
spectra=MS:1003208|experimental precursor monoisotopic m/z
```

```
[2] MS:1003176|attribute mean=1029.05
[2] MS:1003177|attribute standard deviation=0.41
MS:1003322|spectrum cluster best representative=63
…

<Cluster=N+1>
…
```

where N is the spectrum cluster key, which MUST be a positive integer and unique in the file. Cluster key numbers are independent of Spectrum key numbers, and shared Cluster and Spectrum key numbers MUST NOT be assumed to be related. This specifies that there exists a spectrum cluster with key N, which consists of 6 member spectra. Five of the members are in this library (with library entry keys 1, 6, 23, 63, 89) and one is not in the library and is specified as a USI. Any additional information about the cluster can be specified in the same subsection. The subsection containing information about this cluster ends with the next <Cluster=...> line, or with the beginning of the first library spectrum marked by <Spectrum=...>. If a text-format mzSpecLib file contains both Spectrum entries and Cluster entries, they MAY be interleaved.

4.1.6 Library spectrum metadata section

The start of an individual library spectrum or entry in the library is signaled by the line:

```
<Spectrum=N>
```

where N is the library spectrum key (MS:1003237), a positive integer uniquely identifying the spectrum in the library. By custom, the library spectrum key SHOULD begin at 1. Spectra are normally, but should not be assumed to be, placed in ascending order of their library spectrum keys with no gaps. However, if entries in a library are deleted, the remaining entries should retain their original keys and gaps are permitted. Only the uniqueness of a key within a library MUST be enforced.

Optionally, a library spectrum index (MS:1003062) MAY be included to refer to the ordered position of the spectrum within the library, starting with 0 for the first spectrum. A library spectrum may have its index changed as the library evolves, and therefore SHOULD only be used internally by the library management software (e.g. for random access retrieval). To refer to a library spectrum unambiguously from outside (e.g. using a Universal Spectrum Identifier), the library spectrum key MUST be used.

This is followed by a series of CV terms describing the spectrum itself and 0, 1, or more analytes and interpretations for the library spectrum. The CV terms in the <Spectrum> section MUST contain only information pertaining to the library spectrum itself. Information pertaining to the proposed analyte(s) and interpretation(s) MUST be placed in the <Analyte=N> and <Interpretation=M> sections (described below).

The MS level of a spectrum is always assumed to be 2 unless otherwise specified. In cases of MS3 or higher, the MS level MUST be specified via the “ms level” term, e.g.:

MS:1000511|ms level=3

The precursor m/z specified MUST be the selected fragment ion from the previous level of MS, and the peptidoform specified MUST be the peptidoform of the selected fragment ion, including mass modifications. For example, if the precursor is a b ion, the specified peptidoform may need to be encoded with a C terminal "-[b-type-ion]" modification.

To describe the provenance of the spectrum, the origin type and aggregation type of the spectrum can be specified here, e.g.:

MS:1003072|spectrum origin type=MS:1003073|observed spectrum

MS:1003072|spectrum origin type=MS:1003074|predicted spectrum

MS:1003065|spectrum aggregation type=MS:1003068|best replicate spectrum

Within this section, it is possible to cross-reference another spectrum in the library (via its library entry key) or a spectrum external to the library (via a Universal Spectrum Identifier). Several kinds of cross references are defined: contributing replicate spectrum, similar spectrum, and related spectrum. A "contributing replicate spectrum" can be specified in case of a representative spectrum or an aggregated spectrum derived from a set of replicates, to enable the user to trace the library spectrum to its original source(s). A "similar spectrum" is one with high spectral similarity to the present one, and usually presumed to originate from the same analyte. A "related spectrum" is another spectrum that has some relationship with this one, but is not necessarily spectrally similar. This is a flexible mechanism to encode any relationship between spectra, e.g. between the unprocessed and processed version of the same spectrum, between a fragment ion spectrum and its precursor's spectrum when multiple rounds of fragmentations are applied, between spectra from different fragmentation methods of the same analyte, between spectra of the heavy and light isotopic versions of the same peptide ion, etc. Some examples are shown below:

```
[1]MS:1003065|spectrum aggregation type=MS:1003067|consensus spectrum
[1]MS:1003070|number of replicate spectra used=3
[1]MS:1003069|number of replicate spectra available=3
[1]MS:1003297|contributing replicate spectrum keys=18,19,20

MS:1003264|similar spectrum
USI=mzspec:PXD000561:Adult_Frontalcortex_bRP_Elite_85_f09:scan:17555

[2]MS:1003263|similar spectrum keys=888
[2]MS:1003304|spectral dot product=0.89

[3]MS:1003259|related spectrum keys=1234
[3]MS:1003261|related spectrum description=heavy isotope version of the
same peptide ion

[4]MS:1003259|related spectrum keys=567
```

```
[4]MS:1003261|related spectrum description=fragment ion spectrum of
peak at m/z 678.19 (isolation window 678.0 - 680.0)

[5]MS:1003259|related spectrum keys=1083
[5]MS:1003261|related spectrum description=unprocessed spectrum before
deisotoping
```

4.1.7 Analyte metadata section (optional)

An analyte is a charged biomolecule or small molecule that may be a component of the hypothesis about what yielded a spectrum. The start of an individual analyte description is signaled by the line:

```
<Analyte=N>
```

where the N is replaced by an integer that begins with 1 and is unique within a Spectrum entry. This is followed by a series of CV terms that describe analyte N. This section is omitted only if there is no analyte because the interpretation of the spectrum is not known.

In most cases, a MS spectrum is considered an experimental observation of an ion, not the neutral chemical or biomolecule. Therefore, a full specification of the analyte should contain both the chemical or biomolecule AND the charge state. For peptides, the analyte SHOULD be specified using the ProForma 2 notation for peptide ions, which includes the charge, e.g.:

```
MS:1003270|proforma peptidoform ion notation=[iTRAQ4plex]-
LHFFM[Oxidation]PGFAPLTSR/3
```

which represents the peptide sequence, modification(s) if any, the charge state, and, optionally, the adduct ion type in a clear and concise manner. For small molecules, adduct ion information MAY be specified via the “adduct ion formula” (MS:1002813) term. This usage is described in more detail in section 4.7 of the mzPAF 1.0 specification.

Specifications for other analyte types (e.g. lipids, small molecules, glycopeptides etc) will follow in later specifications similar to ProForma.

If the charge state needs to be specified separately from the molecular structure, it SHOULD be specified using the CV term MS:1000041|charge state.

4.1.8 Interpretation metadata section (optional)

An interpretation is a hypothesis about how one or more analytes have yielded a spectrum. The start of an individual spectrum interpretation description is signaled by the line:

```
<Interpretation=M>
```

where the M is replaced by an integer that begins with 1 and is unique among Interpretations within a Spectrum entry. This is followed by a series of CV terms that describes interpretation M. The section MUST contain the CV term to list its member analyte(s) unless there is only one analyte for a spectrum:

```
MS:1003163|analyte mixture members=1,2
```

If there is only one analyte, this term SHOULD be omitted. The Interpretation section SHOULD also contain one or more CV terms providing confidence metrics for that Interpretation.

This section SHOULD be omitted if there is no analyte because the interpretation of the spectrum is not known, as in the case of unidentified spectra in spectral archives.

4.1.9 InterpretationMember metadata section (optional)

This optional section SHOULD only be used if there are CV terms that describe the interpretation member in the context of a specific interpretation. The start of an individual spectrum interpretation member description is signaled by the line:

```
<InterpretationMember=N>
```

where the N is replaced by an integer that corresponds to an Analyte that is a member of the interpretation. This is followed by a series of CV terms that describes the member within the context of a specific interpretation. For example, this section might contain the total intensity fraction that this member contributes to a multi-member Interpretation. Two or more InterpretationMember sections MAY only be provided after an Interpretation section. It is invalid without an Interpretation section.

4.1.10 Example of Analyte, Interpretation, and InterpretationMember metadata

The simplest and, initially, most common case in spectrum libraries is that a Spectrum contains only a single Analyte section:

```
<Spectrum=1>
MS:1003208|experimental precursor monoisotopic m/z=880.8902
<Analyte=1>
MS:1003270|proforma peptidoform ion notation=DSDDVPMVLVGNKCDLAAR/3

<Interpretation=1>
MS:1002357|PSM-level probability=0.974
```

A complex case that contains two different interpretations encompassing 3 analytes might look like this example:

```
<Spectrum=1>
MS:1003208|experimental precursor monoisotopic m/z=880.8902
```

```
<Analyte=1>
MS:1003270|proforma peptidoform ion
notation=DSDDVPM[Oxidation]VLVGNKCDLAAR/2
<Analyte=2>
MS:1003270|proforma peptidoform ion notation=VLVGNKCDLDSDDVPMVLR/2
<Analyte=3>
MS:1003270|proforma peptidoform ion notation=CDLDSDDSDDVPMVLSAR/2
<Interpretation=1>
MS:1003163|analyte mixture members=1,2
MS:1002357|PSM-level probability=0.974
<Interpretation=2>
MS:1003163|analyte mixture members=3
MS:1002357|PSM-level probability=0.937
```

Note that all involved analytes are listed first, then two alternative interpretations (with probabilities 0.974 and 0.937, respectively) are listed, one of which contains two analytes (is chimeric). See section 4.1.17 for a complete example of a chimeric spectrum with multiple analytes and peaks that refer to different analytes.

A maximally complex example that includes InterpretationMembers:

```
<Spectrum=1>
MS:1003208|experimental precursor monoisotopic m/z
<Analyte=1>
MS:1003270|proforma peptidoform ion
notation=DSDDVPM[Oxidation]VLVGNKCDLAAR/2
<Analyte=2>
MS:1003270|proforma peptidoform ion notation=VLVGNKCDLDSDDVPMVLR/2
<Analyte=3>
MS:1003270|proforma peptidoform ion notation=CDLDSDDSDDVPMVLSAR/2
<Interpretation=1>
MS:1003163|analyte mixture members=1,2
MS:1002357|PSM-level probability=0.974
<InterpretationMember=1>
MS:1003166|assigned intensity fraction=0.287
<InterpretationMember=2>
MS:1003166|assigned intensity fraction=0.423
<Interpretation=2>
MS:1003163|analyte mixture members=3
MS:1002357|PSM-level probability=0.937
<InterpretationMember=3>
MS:1003166|assigned intensity fraction=0.512
```

The InterpretationMember number is an integer that corresponds to an Analyte number (e.g. 1 for <Analyte=1>), which MUST be a member of this interpretation. For example, if MS:1003163|analyte mixture members=3,4 then valid values for InterpretationMember are 3 and 4.

NOTE: As seen above, the charge state SHOULD be specified as part of the Analyte, and not as part of the Spectrum if the Analyte section is included.

Only if the Analyte is not known (and the Analyte section is absent), should the charge state of the precursor (e.g. as inferred by examining the precursor isotope pattern in the MSn-1 spectrum) be specified as part of the Spectrum, using the same CV term MS:1000041|charge state. If the charge state of the precursor is ambiguous but can be determined to be among several possibilities, multiple instances of the attribute MS:1000633|possible charge state can be used to list all the possible charge states, e.g.:

```
<Spectrum=234>
MS:1003208|experimental precursor monoisotopic m/z=702.1033
MS:1000633|possible charge state=2
MS:1000633|possible charge state=3
MS:1000633|possible charge state=4
…
```

If the spectrum has more than one precursor m/z in the isolation window and they have different charges, the terms SHOULD be grouped, e.g.:

<Spectrum=234>
[1]MS:1003208|experimental precursor monoisotopic m/z=702.1033
[1]MS:1000041|charge state=2
[2]MS:1003208|experimental precursor monoisotopic m/z=702.3078
[2]MS:1000041|charge state=3

4.1.11 Multiple instances of the same attribute

By default, each attribute SHOULD appear only once under the same context. If multiple instances of the same attribute exist for the same context, and their values differ and cannot be reconciled, the one that appears last in the file takes precedence. However, some attributes MAY be repeated under the same context, to encode the fact that the attribute takes on multiple values for that library/spectrum/analyte/interpretation. Such multiple values MAY have “AND” or “OR” relationships between them. The format does not imply one way or another.

For clarity, any such multiple instances of the same attributes SHOULD be placed adjacent to each other. For example, if a spectrum has undergone multiple steps of data processing, it can be specified as follows:

```
MS:1000543|data processing action=MS:1000033|deisotoping
MS:1000543|data processing action=MS:1003242|rank transform
MS:1000543|data processing action=MS:1001485|m/z calibration
MS:1000543|data processing action=MS:1000594|low intensity data point
removal

MS:1000885|protein accession=sp|HUMAN|xxxx
MS:1000885|protein accession=sp|HUMAN|yyyy
```

It is also possible to have multiple instances of groups of attributes. For example, if a peptide is mapped to multiple proteins, the Analyte section of the library entry MAY be written:

```
<Analyte=1>
MS:1003270|proforma peptidoform ion notation=MYPEPTIDEK/2
MS:1003053|theoretical monoisotopic m/z=367.3043
[1] MS:1000885|protein accession=sp|HUMAN|xxxx
[1] MS:1000886|protein name=protein XYZ
[1] MS:1003047|protein sequence offset=25
[1] MS:1003048|number of enzymatic termini=2
[1] MS:1001045|cleavage agent name=MS:1001251|Trypsin
[1] MS:1001112|n-terminal flanking residue=K
[1] MS:1001113|c-terminal flanking residue=E
[2] MS:1000885|protein accession=sp|HUMAN|yyyy
[2] MS:1000886|protein name=homolog of protein XYZ
[2] MS:1003047|protein sequence offset=27
[2] MS:1003048|number of enzymatic termini=1
[2] MS:1001112|n-terminal flanking residue=H
[2] MS:1001113|c-terminal flanking residue=E
```

Attributes that also have an instance in a claimed attribute set are OVERRIDDEN by the more dominant (last appearing) context. To avoid confusion, multiple instances of the same term within the same context are treated as a set that is claimed and overridden together. For example, if a term is provided in the “all” attribute set, and if a named attribute set is invoked and has two instances of the term (e.g. two instances of MS:1000885|protein accession), the two instances supersede the original single term. Furthermore, if the library spectrum itself contains three instances of the same term, the two claimed instances are discarded, and the final state is the set of three that is locally specified. In other words, new instances of a term that occur in attribute sets referenced later replace those that are referenced earlier, and are not accumulative.

4.1.12 Example of Attribute Sets

The following examples serve to clarify the aforementioned rules.

Example 1

If the following attribute set specifications are found in the library header:

```
<AttributeSet Spectrum=all>
MS:1000465|scan polarity=MS:1000130|positive scan
MS:1000044|dissociation method=MS:1000133|collision-induced
dissociation
MS:1003072|spectrum origin type=MS:1003073|observed spectrum
<AttributeSet Spectrum=ETD>
MS:1000031|instrument model=MS:1000639|LTQ Orbitrap XL ETD
MS:1000044|dissociation method=MS:1000598|electron transfer
dissociation
<AttributeSet Spectrum=HCD>
MS:1000031|instrument model=MS:1002416|Orbitrap Fusion
```

```
MS:1000044|dissociation method=MS:1000598|beam-type collision-induced
dissociation
MS:1000419|collision gas=helium
MS:1000138|normalized collision energy=35
<AttributeSet Spectrum=Decoy>
MS:1003072|spectrum origin type=MS:1003195|shuffle and reposition decoy
spectrum
```

And the following library spectrum is specified:

```
<Spectrum=1>
MS:1003212|library attribute set name=ETD
MS:1003212|library attribute set name=Decoy
MS:1003072|spectrum origin type=MS:1003194|precursor shift decoy
spectrum
```

Then for this library spectrum, the following should apply after resolving conflicts:

`MS:1000465|scan polarity=MS:1000130|positive scan` (inherited from all without conflict)
`MS:1000031|instrument model=MS:1000639|LTQ Orbitrap XL ETD`
`MS:1000044|dissociation method=MS:1000598|electron transfer dissociation` (overriding collision-induced dissociation)
`MS:1003072|spectrum origin type=MS:1003194|precursor shift decoy spectrum` (overriding "shuffle and reposition decoy spectrum", which in turns overrides "observed spectrum")

On the other hand, if this is specified:

```
<Spectrum=1>
MS:1003212|library attribute set name=HCD
MS:1003212|library attribute set name=ETD
```

Then for this library spectrum, the following should apply

```
MS:1000031|instrument model=MS:1000639|LTQ Orbitrap XL ETD
MS:1000044|dissociation method=MS:1000598|electron transfer
dissociation
MS:1000419|collision gas=helium
MS:1000138|normalized collision energy=35
MS:1000465|scan polarity=MS:1000130|positive scan
MS:1003072|spectrum origin type=MS:1003073|observed spectrum
```

This is because the "ETD" set is specified last, so its "dissociation method" instance overrides the same attribute in the "HCD" set, which in turns overrides that in the "all" set. Similarly, the instrument model of "LTQ Orbitrap XL ETD" will override that in the "HCD" set. But the "ETD" set does not specify the attributes of "collision gas" or "normalized collision energy", so those attributes claimed from the "HCD" set still remain. Note that it is the order at which the attribute set is claimed by the library

spectrum entry that matters in terms of precedence, not the order at which the attribute sets are defined in the library header.

Also, since the “normalized collision energy” in the “HCD” set does not appear in the “ETD” set that appears last, it is not overridden and remains an attribute claimed by the library spectrum.

Example 2

If the following attribute set specifications are found in the library header:

```
<AttributeSet Spectrum=all>
MS:1000543|data processing action=MS:1000033|deisotoping
MS:1000543|data processing action=MS:1003242|rank transform
```

The “data processing action” attribute in the following spectrum:

```
<Spectrum=1>
MS:1003212|data processing action=MS:1003241|square root transform
```

will override ALL the instances of the same term (both “rank transform” and “deisotoping”) inherited from the “all” library spectrum attribute set. In other words, when multiple instances of the same term are specified in one context, they are inherited or overridden as an inseparable set.

Example 3

If it is desired that the inherited attributes from an attribute set are placed in a group, then one MUST claim the attribute set inside the group. For example, for a spectral library consisting of mostly spectra of tryptic peptides from human samples, the following attribute set is placed in the library header:

```
<AttributeSet Analyte=human_tryptic>
MS:1001469|taxonomy: scientific name=Homo sapiens
MS:1001045|cleavage agent name=MS:1001251|Trypsin
MS:1003048|number of enzymatic termini=2
```

In a library spectrum of a human tryptic peptide, if the following is specified:

```
<Analyte=1>
MS:1003270|proforma peptidoform ion notation=DLGEENFK/2
[1] MS:1003053|theoretical monoisotopic m/z=476.2245
[1] UO:0000000|unit=MS_1000040|m/z
[2] MS:1000885|protein accession=sp|P02768|ALBU_HUMAN
[2] MS:1000886|protein name=human serum albumin
[2] MS:1003047|protein sequence offset=37
[2] MS:1001112|n-terminal flanking residue=K
[2] MS:1001113|c-terminal flanking residue=A
[2] MS:1003212|library attribute set name=human_tryptic
```

then, the attributes of “MS:1001469|taxonomy: scientific name,” “MS:1001045|cleavage agent name” and “MS:1003048|number of enzymatic termini” will be inherited into the group indicated by the prefix [2] to specify that this is a human protein, and in the context of this protein, the peptide is fully tryptic.

Note that one MAY NOT inherit the “all” attribute set into any group automatically because the set is automatically inherited into the whole (e.g. Analyte) section. The MS:1003212|library attribute set name attribute must be placed in the group to claim the attribute set explicitly. In other words, if some frequently-repeated attributes are to be inherited into a group, they should not be put into the “all” attribute set.

To denote the fact that a spectrum is identified to a peptide that maps to both a human protein and a non-human protein, the following can be specified in the Analyte section of the spectrum:

```
<Analyte=1>
MS:1003270|proforma peptidoform ion notation=KYLYEIAR/2
[1] MS:1003053|theoretical monoisotopic m/z=528.29789
[1] UO:0000000|unit=MS_1000040|m/z
[2] MS:1000885|protein accession=sp|P02768|ALBU_HUMAN
[2] MS:1000886|protein name=human serum albumin
[2] MS:1003047|protein sequence offset=161
[2] MS:1001112|n-terminal flanking residue=K
[2] MS:1001113|c-terminal flanking residue=R
[2] MS:1003212|library attribute set name=human_tryptic
[3] MS:1000885|protein accession=sp|P02769|ALBU_BOVIN
[3] MS:1000886|protein name=bovine serum albumin
[3] MS:1003047|protein sequence offset=160
[3] MS:1001112|n-terminal flanking residue=G
[3] MS:1001113|c-terminal flanking residue=R
[3] MS:1001469|taxonomy: scientific name=Bos taurus
[3] MS:1001045|cleavage agent name=MS:1001251|Trypsin
[3] MS:1003048|number of enzymatic termini=1
```

Here, note that the attribute set “human_tryptic” is claimed by the group prefixed [2] only, and it does not affect the group prefixed [3]. The same peptide can be mapped to a human protein (specified in group [2]) or a bovine protein, in which it appears as a semitryptic peptide (specified in group [3]).

4.1.13 Peak data

The start of the peak data for a spectrum is signaled by the line:

```
<Peaks>
```

Each line of peak data MUST be tab-separated values of 2, 3, or more columns as follows:

Column 1 encodes the m/z value of the peak.

Column 2 encodes the intensity value of the peak.

Column 3 is optional and contains one or more annotations (assignments to fragment ions) of the peak if present. The content of this column is described in detail in section 4.1.15 of this specification.

Additional columns are optional and MAY contain additional attributes of the peak, such as aggregation information. If a column 4 is needed (see below in 4.1.16), then column 3 must be present, even if blank (i.e., an empty string). A description of these additional columns is described in subsection 4.1.16.

After the peak data is complete, the line:

```
<Spectrum=N>
```

signals the start of the next spectrum. A blank line between entries is permitted but not required.

4.1.14 Peak annotations

Peaks MAY be annotated to describe the fragment ions that generated them. If peptide ion annotations are present, they MUST be specified using the mzPAF format (http://psidev.info/mzPAF). At present mzPAF only supports peptides and small molecules. For other use cases, such as glycopeptides and lipids, other annotation formats MAY be used.

4.1.15 Specifying the Peak Annotation Format used

The choice of peak annotation format for a spectrum SHOULD be specified in as an attribute of the spectrum in this manner:

```
<Spectrum=N>
…
MS:1003103|ion annotation format=MS:1003104|mzPAF peptide ion
annotation format
…
```

Typically, all spectra in a library would use the same peak annotation format. To reduce repetition, this ion annotation format attribute can be specified in a “spectrum attribute set” in the library header and inherited by all library spectra. Moreover, there exist mechanisms to specify multiple spectrum attribute sets that apply to different subsets of spectra, or to override the global specification locally. If no format is specified, the mzPAF peptide ion annotation format MUST be assumed.

If one spectrum contains multiple types of peaks, multiple formats may be specified as in:

```
MS:1003103|ion annotation format=MS:1003104|mzPAF peptide ion
annotation format
MS:1003103|ion annotation format=MS:1003106|glycan ion annotation
format
```

to indicate that some annotations are in peptide ion format, and some are in glycan format. The different format MUST be differentiable based on the first character of the 'ion type' component of each annotation (see section 'Ion Type' on page 8 below).

This is primarily a signal to the reader which annotation formats should be expected, and immediate action taken when unsupported formats are encountered.

Supported formats SHOULD be parsed by a standardized parser utilizing well-defined regular expressions or context-free grammars. Users SHOULD NOT write their own parser. The recommended regular expression is provided in the mzPAF specification at https://psidev.info/mzPAF.

4.1.16 Additional peak attributes

Besides the m/z, the intensity, and the annotation, other attributes of the peak can be specified in columns following the first three, separated by tabs.

For example, some spectral libraries contain spectra that are representative of other replicate spectra in existence but not provided in the library, often in the form of consensus (merged) spectra or best-replicate spectra. In such cases, each peak may have associated information about what is determined to be the same peak in many of the replicate spectra. The peak aggregation information can be specified in the additional columns. Other information such as ion mobility, intensity rank, and the original charge of the peak (for a decharged spectrum), etc., can also be included in this manner. Tabs are NOT permitted within values.

The meaning of each column MUST be defined by an order-dependent attribute group within the <Spectrum> section of the library spectrum using CV terms. (This can also be specified in an attribute set in the library header and claimed by all spectra.) All peaks in the same spectrum MUST have the same number of columns, and the specified meanings of the columns MUST apply to all peaks in a spectrum. One MAY skip a column for a given peak (e.g. if the attribute is unknown or not applicable for that peak) by having consecutive tabs. This is to be interpreted as a null value for that column. If a peak row has fewer columns than are defined, the missing values are to be interpreted as nulls.

As an example, an attribute group may look like:

```
[1]MS:1003254|peak attribute=MS:1003279|observation frequency of peak
```

```
[1]MS:1003254|peak attribute=MS:1003285|standard deviation of m/z
values of peak among replicates
[1]MS:1003254|peak attribute=MS:1003286|coefficient of variation of
intensity of peak among replicates
```

The keys MUST be MS:1003254|peak attribute and the values MUST be child terms of MS:1003254|peak attribute.

With this specification, the corresponding peak list may look like any of the following, depending on the information available for that peak:

with both annotation and full aggregation information
```
110.04050   12342 IH    10    52.34 23.43
110.34050   12342 ?     3     522.34      22.43
```

with annotation but no aggregation information (the last 3 columns are left out)
```
110.04050   12342 IH
110.34050   12342 ?
```

with NO annotation but with full aggregation information (the 3rd column is an empty string):
```
110.04050   12342       10    52.34 23.43
110.34050   12342       3     522.34      22.43
```

with annotation but with some aggregation information missing (missing column is an empty string):
```
110.04050   12342 IH    10          23.43
110.34050   12342 ?           522.34      22.43
```

4.1.17 Example of a chimeric spectrum

As discussed in Sections 3.4.5 and 4.1.10 , a chimeric spectrum can be indicated by including multiple <Analyte> sections for the spectrum, and specifying the MS:1003163|analyte mixture members attribute. Additionally, the mzPAF specification for peak annotations allows the association of each peak to one of the analytes via the n@ prefix. An example is shown below for a chimeric spectrum.

```
<Spectrum=1>
MS:1003061|library spectrum name=AILINFIDR/2+HLAEISLLEQSFVK/3
MS:1003208|experimental precursor monoisotopic m/z=538.3234
MS:1003072|spectrum origin type=MS:1003073|observed spectrum
MS:1003059|number of peaks=222
MS:1003063|universal spectrum
identifier=mzspec:PXD046281:20230206_borrelia_5A4B31_rprp_F9:scan:39167
:AILINFIDR/2+HLAEISLLEQSFVK/3
<Analyte=1>
MS:1003270|proforma peptidoform ion notation=AILINFIDR/2
MS:1001117|theoretical mass=1074.6306
```

```
<Analyte=2>
MS:1003270|proforma peptidoform ion notation=HLAEISLLEQSFVK/3
MS:1001117|theoretical mass=1613.8897
<Interpretation=1>
MS:1003163|analyte mixture members=1,2
<Peaks>
 84.0446   11605.7  IQ-NH3/2.5ppm
 84.0811   28119.8  IK-NH3/3.9ppm
 86.0967  708095.4  IL/3.2ppm
 87.0556    7941.3  IN/3.6ppm
 87.1000   43344.2  IL+i/-0.6ppm
 88.0396    6063.7  ID/3.4ppm
 89.0709    3381.4  ?
101.0712    8654.1  IQ/2.6ppm
102.0552   16947.8  IE/2.4ppm
104.6250    3621.9  ?
107.7645    3077.3  ?
110.0715   32326.8  2@a1/2.1ppm
112.0872   12136.4  IR-NH3/2.5ppm
114.0915    5812.4  1@m2:3^2/1.4ppm
116.0711    5845.4  IR+H2O+H2O-N3H7/4.3ppm
120.0811   37238.5  IF/2.7ppm
128.0067    3493.3  ?
129.1025   42042.6  2@y1-H2O/2.0ppm
130.0866   18462.1  2@y1-NH3/2.7ppm
136.0760   25607.6  0@IY/2.3ppm
136.0814    4217.6  ?
140.1437   21269.7  ?
141.0663    3462.8  2@m9:11-CO-H2O-H2O^2/3.2ppm
143.1183    5625.6  IK[Acetyl]/2.9ppm
147.1130   30064.8  2@y1/1.3ppm
147.7529    4753.6  ?
147.7613   10702.6  ?
147.7793   22742.6  ?
147.7882   11670.4  ?
147.7962    8056.7  ?
147.8054    6234.9  ?
147.8128    4745.6  ?
157.1339  583917.3  1@a2/2.3ppm
158.0926    9620.5  1@y1-NH3/1.2ppm
158.1306    6595.5  0@a2{GK}/11.5ppm
158.1372   47934.8  1@a2+i/-0.4ppm
171.1130    4223.8  1@m3:5^2/1.1ppm
172.1165    4587.1  ?
173.1289    7568.3  2@m5:6-CO/2.6ppm
175.1193  175526.3  1@y1/2.0ppm
175.1280    8607.3  ?
176.1229    9200.3  1@y1+i/1.4ppm
178.2119   10569.1  ?
```

```
183.1133    9553.6  2@m5:6-H2O/2.7ppm
183.5190    3343.2  ?
185.1288  463714.2  1@b2/1.9ppm
185.1651   13455.5  0@a2{LV}/1.4ppm
186.1323   42196.1  1@b2+i/-14.2ppm
187.1443    7080.1  0@a2{LT}/1.0ppm
192.7906    3377.9  ?
196.1449   29221.6  ?
197.1289    8604.2  2@y3^2/2.3ppm
199.1809  195314.4  1@m2:3-CO/2.1ppm
200.1397    4190.7  1@m4:5-CO/1.7ppm
200.1841   22796.5  1@m2:3-CO+i/-0.8ppm
201.0867    4445.6  2@m3:4/-1.4ppm
201.1239   13906.0  2@m5:6/2.6ppm
209.1033    4416.2  0@b2{AH}/-0.0ppm
211.1085    6414.8  1@m7:8-H2O/3.7ppm
214.1551    4383.8  0@a2{LQ}/0.5ppm
215.1399    7895.1  2@m4:5-CO/4.1ppm
216.0990    4989.7  2@m10:11/5.2ppm
217.0824    3800.7  0@b2{DT}/2.3ppm
223.1557   10709.2  2@a2/1.6ppm
224.1397   32495.5  ?
225.1237    4280.9  2@m7:12-CO-NH3^3/1.5ppm
225.1965    6662.8  ?
226.1182    4171.9  2@b4^2/-1.8ppm
227.1760   74901.4  1@m2:3/2.6ppm
228.1348   26600.3  1@m4:5/2.3ppm
228.1795   10189.0  1@m2:3+i/-10.5ppm
229.1190    9942.9  1@m7:8/3.1ppm
229.1389    4238.0  1@m4:5+i/3.3ppm
233.1653    6992.5  1@m6:7-CO/2.0ppm
234.1240   18098.7  1@m5:6-CO/1.3ppm
240.0986   10605.4  2@m9:10-H2O/3.0ppm
… (truncated for brevity)
```

4.1.18 Comments in the text format

In the text serialization it is permissible to write a human-readable comment as a line beginning with the # character. A # character placed NOT as the first character MUST not be considered a comment.

Comments are permitted purely for human discussion and are generally NOT transcribed to other serializations. For example, the JSON serialization has no mechanism for comments.

4.2 JSON serialization

The JSON serialization of a spectral library is intended to serve as a convenient, readily parsable encoding of the spectral library. The most common use case will be transmitting a single spectrum or small numbers of spectra via web APIs. If one or more spectra are serialized in JSON to a file, the file extension SHOULD be .mzSpecLib.json. A JSON schema is provided for validation at https://github.com/HUPO-PSI/mzSpecLib/blob/master/specification/mzSpecLib_json.schema.json.

4.2.1 Spectral Library Representation

The JSON serialization of a spectral library has three top-level keys, “attributes”, “format_version”, and “spectrum”. The “format_version” key MUST map to the value of “MS:1009002|format version”, as this may drive the interpretation of subsequent keys. The “attributes” key maps to an array of CV terms describing the library itself. The “spectra” key maps to an array of spectrum entries.

```
{
        “format_version”: “1.0”,
        “attributes”: [...],
        “clusters”: [...],
        “spectra”: [...],
        “library_spectrum_attribute_sets”: { “all”: { “attributes”: [...] }, “decoy”: {
“attributes”: [...] } } },
       “library_analyte_attribute_sets”: { “all”: { “attributes”: [...] }, “decoy”: {
“attributes”: [...] } } },
        “library_interpretation_attribute_sets”: { “all”: { “attributes”: [...] } }
}
```

All requirements on CV terms from the text serialization apply to the JSON serialization.
4.2.2 Controlled Vocabulary Terms
The encoding of CV terms encode the “accession”, a string, the “name”, a string, the “value”, of varied type, “value_accession”, a string, if applicable, and “cv_param_group”, an integer, if applicable.

```
      {
        "accession": "MS:1000224",
        "name": "molecular mass",
        "value": 1710.9076
      }
```

An example where “value_accession” is present

```
      {
        "accession": "MS:1001045",
        "cv_param_group": 1,
        "name": "cleavage agent name",
        "value": "Trypsin",
        "value_accession": "MS:1001251"
```

```
    }
```

An example where “cv_param_group” is present

```
{
  "accession": "MS:1000045",
  "cv_param_group": 2,
  "name": "collision energy",
  "value": 46
},
{
  "accession": "UO:0000000",
  "cv_param_group": 2,
  "name": "unit",
  "value": "electronvolt",
  "value_accession": "UO:0000266"
}
```

In the last case, the second term in the group alters the interpretation of the first term in the group.

4.2.2 Spectrum Encoding

The serialization of a spectrum in JSON MUST have an “attributes” key mapping to an array of CV terms describing the spectrum, an “analytes” key mapping to an object mapping analyte identifiers to analyte descriptions (if the analyte is not known for a spectrum, the “analytes” value should be an empty object), an “mzs” key mapping to an array of numbers giving the m/z values for each peak in the spectrum, and an “intensities” key mapping to an array of numbers giving the intensity values for each peak in the spectrum. It MAY contain a “peak_annotations” key mapping to an array of peak annotations where each annotation is an array of strings or objects, and an “aggregation_metadata” key mapping to an array of peak aggregation statistics encoded as an array of numbers or strings.

```
{
 “attributes”: [...],
 “analytes”: {“1”: {…}, “2”: {...} },
 “interpretations”: {“1”: {...}, ...}
 “mzs”: [...],
 “intensities”: [...],
 “peak_annotations”: [ [...], [...], ... ],
 “aggregation_metadata”: [ [...], [...], ... ],
}
```

The serialization of an analyte in JSON MUST have an “attributes” key mapping to an array of CV terms. It was decided that for m/z values and intensities, parallel arrays are more space efficient than a single array of maps.

4.2.3 Analyte Encoding

The serialization of an analyte in JSON MUST have an “attributes” key mapping to an array of CV terms describing the analyte, and an “id” key mapping to a string defining the within-entry identifier for this analyte. This arrangement allows an analyte to be accessed via its key in a set of analytes, but it also allows an analyte object to know its own id independently of whether it is in a set, and also allows for future extensibility.

```
{
 "attributes": [
  {
   "accession": "MS:1000224",
   "name": "molecular mass",
   "value": 1710.9076
  },
  ...
 ],
 "id": "1"
}
```

4.3 Other serializations

An HDF5 serialization has been proposed and may be forthcoming, but is not described in this specification. It should follow the same data model so that interconversion between serializations is straightforward.

A SQLite3 serialization has been proposed and may be forthcoming, but is not described in this specification. It should follow the same data model so that interconversion between serializations is straightforward.

Other serializations that optimize for access speed, size, or indexing are acceptable as long as they follow the same data model so that interconversion between serializations is straightforward.

5. Future possible developments in other fields

This specification document primarily addresses the needs of the proteomics mass spectrometry community. The format may also be suitable as a base for encoding spectral libraries in metabolomics, glycomics, and lipidomics (among other mass spectrometry fields) if there is a desire to adopt this format.

However, it is important to highlight that extensions and adaptations in the format may be needed to support appropriately the requirements of those fields. In addition, extensions to the PSI-MS CV may be needed as well.

6. Author Information

Henry Lam
The Hong Kong University of Science and Technology
kehlam@ust.hk

Tytus D. Mak
Mass Spectrometry Data Center, National Institute of Standards and Technology
tytus.mak@nist.gov

Joshua Klein
Boston University
joshua.adam.klein@gmail.com

Wout Bittremieux
University of Antwerp
wout.bittremieux@uantwerpen.be

Ralf Gabriels
VIB-UGent Center for Medical Biotechnology, VIB, Ghent, Belgium
Ralf.Gabriels@UGent.be

Yasset Perez-Riverol
European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI)
yperez@ebi.ac.uk

Tim Van Den Bossche
VIB-UGent Center for Medical Biotechnology, VIB, Ghent, Belgium
Tim.VanDenBossche@UGent.be

Juan Antonio Vizcaíno
European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI)
juan@ebi.ac.uk

Eric W. Deutsch
Institute for Systems Biology, Seattle WA, USA
edeutsch@systemsbiology.org

7. Contributors

In addition to the authors, many other contributions have been made during the preparation process. The contributors who actively participated in the development, testing, and review of the recommendation documentation are:

Nuno Bandeira
Center for Computational Mass Spectrometry, Department of Computer Science and Engineering, Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California, San Diego, 92093-0404, USA

Pierre-Alain Binz
Lausanne University Hospital, Lausanne, Switzerland

Jeremy Carver
Center for Computational Mass Spectrometry, Department of Computer Science and Engineering, Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California, San Diego, 92093-0404, USA

Tine Claeys
VIB-UGent Center for Medical Biotechnology, VIB, Ghent, Belgium

Helge Hecht
Masaryk University, Kotlářská 2, Brno, Czech Republic

Nils Hoffmann
Institute for Bio- and Geosciences (IBG-5), Forschungszentrum Jülich GmbH, 52428 Jülich, Germany

Andrew R. Jones
Institute of Systems, Molecular and Integrative Biology, University of Liverpool, Liverpool, L69 3BX, United Kingdom

Shin Kawano
Database Center for Life Science, Joint Support Center for Data Science Research, Research Organization of Information and Systems, Chiba, Japan

Luis Mendoza
Institute for Systems Biology, Seattle, Washington 98109, United States

Benjamin A. Neely
National Institute of Standards and Technology

Benjamin Pullman

Center for Computational Mass Spectrometry, Department of Computer Science and Engineering, Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California, San Diego, 92093-0404, USA

Jim Shofstahl
Thermo Fisher Scientific, 355 River Oaks Parkway San Jose, CA 95134, USA

Zhi Sun
Institute for Systems Biology, Seattle, Washington 98109, United States

Tomi Suomi
University of Turku, Turku, Finland

Yunping Zhu
National Center for Protein Sciences (Beijing), Beijing Institute of Lifeomics, #38, Life Science Park, Changping District, Beijing 102206, China

8. Intellectual Property Statement

The PSI takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the PSI Chair.

The PSI invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the PSI Chair (see contacts information at PSI website).

9. Copyright Notice

10. Glossary

All non-standard terms are already defined in detail in section 3.

11. References

1. Lam H, Deutsch EW, Eddes JS, Eng JK, King N, Stein SE, Aebersold R. Development and validation of a spectral library searching method for peptide identification from MS/MS. Proteomics. 2007 Mar;7(5):655–667. PMID: 17295354
2. Frewen B, MacCoss MJ. Using BiblioSpec for creating and searching tandem MS peptide libraries. Curr Protoc Bioinforma. 2007 Dec;Chapter 13:Unit 13.7. PMID: 18428681
3. Deutsch EW, Perez-Riverol Y, Chalkley RJ, Wilhelm M, Tate S, Sachsenberg T, Walzer M, Käll L, Delanghe B, Böcker S, Schymanski EL, Wilmes P, Dorfer V, Kuster B, Volders PJ, Jehmlich N, Vissers JPC, Wolan DW, Wang AY, Mendoza L, Shofstahl J, Dowsey AW, Griss J, Salek RM, Neumann S, Binz PA, Lam H, Vizcaíno JA, Bandeira N, Röst H. Expanding the Use of Spectral Libraries in Proteomics. J Proteome Res. 2018 07;17(12):4051–4060. PMCID: PMC6443480
4. Bradner S. RFC2119: Key words for use in RFCs to Indicate Requirement Levels (https://tools.ietf.org/html/rfc2119) [Internet]. 1997. Available from: https://tools.ietf.org/html/rfc2119
5. Deutsch EW, Perez-Riverol Y, Carver J, Kawano S, Mendoza L, Van Den Bossche T, Gabriels R, Binz PA, Pullman B, Sun Z, Shofstahl J, Bittremieux W, Mak TD, Klein J, Zhu Y, Lam H, Vizcaíno JA, Bandeira N. Universal Spectrum Identifier for mass spectra. Nat Methods. 2021 Jul;18(7):768–770. PMCID: PMC8405201
6. Martens L, Chambers M, Sturm M, Kessner D, Levander F, Shofstahl J, Tang WH, Römpp A, Neumann S, Pizarro AD, Montecchi-Palazzi L, Tasman N, Coleman M, Reisinger F, Souda P, Hermjakob H, Binz PA, Deutsch EW. mzML--a community standard for mass spectrometry data. Mol Cell Proteomics MCP. 2011 Jan;10(1):R110.000133. PMCID: PMC3013463
7. LeDuc RD, Deutsch EW, Binz PA, Fellers RT, Cesnik AJ, Klein JA, Van Den Bossche T, Gabriels R, Yalavarthi A, Perez-Riverol Y, Carver J, Bittremieux W, Kawano S, Pullman B, Bandeira N, Kelleher NL, Thomas PM, Vizcaíno JA. Proteomics Standards Initiative's ProForma 2.0: Unifying the Encoding of Proteoforms and Peptidoforms. J Proteome Res. 2022 Apr 1;21(4):1189–1195. PMCID: PMC7612572
8. Frank AM, Monroe ME, Shah AR, Carver JJ, Bandeira N, Moore RJ, Anderson GA, Smith RD, Pevzner PA. Spectral archives: extending spectral libraries to analyze both identified and unidentified spectra. Nat Methods. 2011 May 15;8(7):587–591. PMCID: PMC3128193