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
Douwe Schulte, Utrecht University
Yasset Perez-Riverol, European Bioinformatics Institute
Tim Van Den Bossche, VIB-UGent Center for Medical Biotechnology
Juan Antonio Vizcaíno, European Bioinformatics Institute
Eric W. Deutsch, Institute for Systems Biology

October 15, 2024

mzPAF: Peak Annotation Format - Peptides

Status of this document

This document provides information to the proteomics community about a proposed Peak Annotation Format specification for fragment ion mass spectra. The current specification document is focused on peptides. Distribution is unlimited.

Version: Draft 15 of 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 fragment ion peak annotation format for mass spectra, focused on peptides. This provides for a standardized format for describing the origin of fragment ions to be used in spectral libraries, other formats that aim to describe fragment ions, and software tools that annotate fragment ions. Further detailed information, including any updates to this document, implementations, and examples is available at http://psidev.info/mzPAF/.

Table of Contents

1. Introduction

1.1 Description of the need

As part of the PSI spectral library format mzSpecLib, it is possible to annotate individual peaks, as is already done in spectral libraries from the National Institute for Standards and Technology (NIST, https://chemdata.nist.gov/dokuwiki/doku.php?id=peptidew:start), SpectraST,[1] and PeptideAtlas.[2] However, there have been several different styles of annotations in the past (even from a single provider), and therefore this document describes a single common peak annotation format for peptides that is recommended for all peptide libraries and related applications from which peak annotations are desirable.

The specification is heavily based on the formatting used in the NIST MSP format and the SpectraST sptxt format. These precursor formats were quite similar, but not exactly the same, and were never fully documented. NIST MSP annotations have undergone small changes over the years. Those annotation formats are based on the original nomenclature proposals published by Roepstorff and Fohlman[3], which was further refined by Biemann[4]. Participants from NIST and SpectraST have led the development of this standard.

This format, as currently described, is designed for linear peptides with "simple" modifications, i.e. those routinely identified by typical proteomics pipelines (with or without enrichment methods), and for fragmentation methods commonly used in proteomics such as collision-induced dissociation (CID), higher-energy C-trap dissociation (HCD), and electron-transfer dissociation (ETD). Although there are some provisions for annotating small molecules (e.g., contaminants in a predominantly peptide spectrum), as well as unusual fragments, it is expected that for other major classes of analytes (metabolites, glycans, lipids, glycopeptides, cross-linked peptides, etc.), alternative peak annotation formats should be defined, ideally compatible with this format.

Throughout this document, when referring to mzPAF, it will primarily be in the context of supporting peptides in proteomics.

The content of this specification is inspired in part by and addresses some of the wishes laid out by the journal article "Expanding the Use of Spectral Libraries in Proteomics"[5] following extensive discussions at the Dagstuhl Seminar and at different PSI workshops.

1.2 Requirements

The main requirements to be fulfilled for the peak annotation format are:

- It MUST be machine parsable as well as easily human readable.
- It MUST be compatible with existing PSI file formats (especially mzSpecLib), where it will be used.

- It MUST support the encoding of linear peptides with "simple" modifications, i.e. those routinely identified by typical proteomics pipelines, but not including glycans.
- It MUST support fragmentation methods commonly used in proteomics such as CID, HCD, and ETD.
- It MUST support all reasonably common peaks observed in fragment ion spectra.

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.[6] In general, "MUST" means required, "SHOULD" means recommended, and "MAY" means optional.

3. The Peak Annotation Format Definition

3.1 The documentation

The document provides the full specification of the mzPAF peak annotation format. It is accompanied by several other products. All products in their most recent form are available at the HUPO-PSI website (http://psidev.info/mzPAF/) or at the GitHub version control repository (https://github.com/HUPO-PSI/mzPAF). Additional components that accompany this specification are:

- An extendable list of isobaric label ions in text and JSON formats
- An extendable list of neutral losses in text and JSON formats
- A set of exemplary annotated spectra that demonstrate the use of mzPAF

3.2 Relationship to other specifications

The format specification described in this document is not being developed in isolation; it is designed to be complementary to, and thus used in conjunction with, other PSI standards. Current related specifications include the following:

1. *mzSpecLib, the PSI spectrum library format* (http://psidev.info/mzSpecLib). The PSI spectrum library format is being developed as a standard mechanism for storing spectrum libraries. Individual peaks from mass spectra encoded in the libraries are annotated using this format.
2. *ProForma 2.0.* This PSI format contains rules on how to encode peptidoforms and molecular formulae[7] (https://www.psidev.info/proforma).

4. The Basic Form of the Peak Annotation Format

The mzPAF peak annotation format is composed of a string of characters. It is case sensitive. There is no limit in its maximum length. Line breaks MUST NOT be used. The

character encoding is not enforced, but all characters shown in this specification are UTF-8 characters below 128 unless otherwise stated in this specification.

The basic format of each annotation is:

annotation1/delta,annotation2/delta,...
or
*annotation1/delta*confidence,annotation2/delta*confidence,...*

e.g.
`b2-H2O/3.2ppm,b4-H2O^2/3.2ppm`

where multiple possible explanations are separated with a comma. Deltas of (observed – theoretical) *m*/*z* values are prefixed with a slash (/). Scores MAY be provided for different annotations prefixed with an asterisk (*), such as:

`b2-H2O/3.2ppm*0.75,b4-H2O^2/3.2ppm*0.25`

The sections below define each component of these annotations.

4.1 Annotation of multiple analytes

It is common for there to be multiple analytes co-fragmented together to produce a spectrum, or there may be alternative interpretations of the spectrum. These can take the form of two or more separately described precursors, or just low-level contamination from background peptide ions, or other miscellaneous molecules. It is possible to define more than one analyte in the context of a single spectrum, and these MUST be assigned numbers 1, 2, etc. The number 1 is assumed to be the primary analyte, the one that the annotation writer recommends as first in a list of analytes. The number 0 is reserved as one or more unspecified contaminant molecules (see example below for common y1 ion observations for unidentified contaminant peptides).

For spectra that have multiple analytes associated with them, peak annotations MUST be prefixed with their analyte number as defined for the spectrum and an @ symbol. For such cases of multiple specified analytes, the prefix notation MUST be present on every ion to indicate to which analyte the annotation applies. If there is only one analyte defined, peak annotations SHOULD NOT be annotated with 1@. For example,

`1@y12/0.13,2@b9-NH3/0.23`

indicates that the peak may be either the y12 ion from analyte 1 or the b9-NH3 ion from analyte 2. As another example, most high signal-to-noise ratio HCD spectra of tryptic digests contain the y1 ions corresponding to both lysine and arginine. If analyte 1 is a peptide ending in R, it will be common to see:

`0@y1{K}`

```
0@y1{K}-NH3
```

which is the lysine y1 ion and the y1-NH3 ion corresponding to some unspecified (hence the number 0) contaminating peptide ion ending with a lysine. See section 4.4.3 for more information about specifying sequences for ion series fragments.

4.2 Multiple annotations

Each peak may have multiple annotations separated by commas. These multiple annotations MAY represent AND or OR. There is no distinction between whether the annotation system intends that there are two contributors to a peak or whether they are mutually exclusive.

If there are several annotations, they SHOULD be ordered by decreasing likelihood (based on existing knowledge about fragmentation), e.g. a primary series ion should be listed before a rare neutral loss.

If a regular expression-based annotation parser designed for a single annotation is used, additional logic is required to handle multiple annotations. A procedure like the following SHOULD be applied:

- From the current position, attempt to pattern-match the longest possible regular expression and record it.
- If the next unmatched character is a comma, skip the comma and begin again, and repeat until done.
- If the regular expression goes to the end of the string, then parsing is complete.
- If the next unmatched character is not a comma, this is a parsing error.

See Appendix A for a Python-like pseudo code description of this procedure.

4.3 Deviation of observed *m*/*z* from the theoretical *m*/*z* values

Each annotation SHOULD include an *m*/*z* delta (observed m/z - theoretical m/z), UNLESS the m/z values provided are all theoretical values anyway, as in the case for a library of predicted spectra. The theoretical m/z is calculated from the sum of the atoms and charged particles comprising the ion reportedly explaining the annotated peak. A negative delta MUST be preceded by a minus sign. A non-negative delta MUST NOT be preceded by any sign. There are two possible units, either *m*/*z* units (Daltons per elementary charge) or parts per million (ppm). Any *m*/*z* deltas in parts per million MUST have the suffix “ppm” in lower case without a preceding space. *m*/*z* deltas in *m*/*z* units (Daltons divided by charge) MUST NOT have any suffix. Examples:

```
y1/-1.4ppm
y1/-0.0002
```

4.4 Ion notation

Each annotation begins with an ion notation describing the putative peak origin. The ion notation has multiple components described as follows:

[ion type](neutral loss)(isotope)(adduct type)(charge)

Of these five components, only the first (ion type) is always required. The others are optional. Each of these components is described in the following subsections. A complex example with all five components is:

```
y4-H2O+2i[M+H+Na]^2
```

Here, a peak from two isotopes above the monoisotope of a doubly charged protonated and sodiated y4 ion with a water loss is annotated.

4.4.1 Ion types overview

The ion type component is required and describes the basic type of ion being described. Examples are b ions, y ions, immonium ions, unfragmented precursor ions, internal fragmentation ions, isobaric tag ions, etc. Each of these is described in the subsections below. As mentioned above, the specification is peptide centric. Although there is desire to support other kinds of molecules (e.g. small molecules, lipids, and glycans) in the future as well, an accepted nomenclature for specifying such ion types has not yet been decided. However, this specification offers limited support for such ion types through the use of chemical formulae, SMILES strings or a free-text name of the non-peptide molecule.

The following is a list of ion type prefixes, as described in detail in the subsections below:

Prefix	Code point	Description
?	63	Unknown ion
a	97	Peptide a ion series
b	98	Peptide b ion series
c	99	Peptide c ion series
d	100	Peptide d ion series
v	118	Peptide v ion series
w	119	Peptide w ion series
x	120	Peptide x ion series
y	121	Peptide y ion series
z	122	Peptide z ion series

I	73	Immonium ion
m	109	Internal fragment (‘m’iddle)
_	95	Named compound (underscore)
p	112	Precursor ion
r	114	Reference molecule, such as a TMT or iTRAQ ion or Hex or Adenosine
f	102	Chemical formula
s	115	SMILES string

For most ion types, one or two ordinal numbers or additional characters follow the ion type prefix to complete the specification; see examples below.

The following prefixes are reserved for future extensions for custom annotation formats for other molecule types: G for glycan ion fragments; L for lipid ion fragments; X for cross-linked peptide fragments.

4.4.2 Unknown ions

If a spectrum has been annotated and peak annotations are included for other peaks, those peaks that cannot be interpreted SHOULD be marked with ‘?’. The charge state and isotopic state MAY be specified after the ‘?’ if they can be determined (e.g. by charge deconvolution), such as:

```
?
?^3
?+2i^4
```

See the “Isotope” and “Charge State” sections below for more information on those components.

An unknown ion MAY be assigned a positive integer in order to annotate relationships with other unknown ions. In the following examples, one ion is designated as number 17 and then an isotope and neutral loss of this ion are annotated to indicate their proposed relationship to a primary unknown ion:

```
?17
?17+i/1.45ppm
?17-H2O/-0.87ppm
```

Unknown ion annotations MAY be preceded with an analyte identifier such as 0@ or 1@ or left unspecified. Writers MAY specify their preference if orthogonal information indicates whether an ion belongs to a specified analyte or a contaminant.

4.4.3 Primary series ions

The primary fragmentation series ions include a, b, and c ions from the N terminus and x, y, and z ions from the C terminus. The ion types are followed by an ordinal to indicate how many residues from the terminus are included in the fragment. For example, a b2 ion indicates 2 residues from the N terminus.

The following table describes how the theoretical mass of each type of primary ion is calculated. The formulae as shown assume that the N and C termini are unmodified, and the adduct type is [M + zH]. Σ(AA) is the sum of masses of the neutral, modified amino acid residues (i.e. the structure -NH-CHR-CO- where R is the side-chain, for non-proline). $(H^+)z$ is the proton mass multiplied by the charge state.

	Formula	Remarks
a	$\sum(AA) - CO + (H^+)z$	
b	$\sum(AA) + (H^+)z$	
c	$\sum(AA) + NH_3 + (H^+)z$	Note: The “c-1” radical ion, which arises when the c ion loses a hydrogen to the z.+ ion, is also observed in ETD. This is denoted with a “neutral loss” of -H, e.g. “c12-H^2”. See “Neutral losses” section below.
d (da) (db)	$\sum_{\square}^{n-1} \square (AA) + C_2H_3N + (H^+)z$	Note: the d ion is partial loss of the side chain from an a ion, breaking at the second carbon atom. For Threonine and Isoleucine there are two possible d ions these can be listed as ‘daN’ and ‘dbN’ where a is the heaviest of the two options. For Threonine da has an additional group of OH and db a group of CH_3. Isoleucine da has an additional group of C_2H_5, and db a group of CH_3. Valine is a special case; it has CH3 as additional mass. Glycine, alanine, and proline have no d ions.
v	$\sum_{\square}^{c-1} \square (AA) + C_2H_2NO + (H^+)z$	Note: the v ion is complete loss of the side chain of a y ion.
w (wa) (wb)	$\sum_{\square}^{c-1} \square (AA) + C_3H_3O + (H^+)z$	Note: the w ion is partial loss of the side chain from an z ion, breaking at the second carbon atom. For Threonine and Isoleucine there are two possible w ions these can be listed as ‘waN’ and ‘wbN’ where a is the heaviest of the two options. For more details see the d ion.
x	$\sum(AA) + CO - 2H + (H^+)z$	

y	$\sum(AA) + H_2O + (H^+)z$	
z	$\sum(AA) + H_2O - NH_2 + (H^+)z$	Note: This is also known as the z.+ radical ion, which is found in ETD spectra. This ion is heavier by one hydrogen atom than the non-radical "z ion" originally defined by Biemann (1990).[4] In addition, the "z+1" ion, which arises when the z.+ abstracts a hydrogen atom from the c ion, is also observed in ETD. This is denoted with a "neutral gain" of +H, e.g. "z12+H^2". See "Neutral losses" section below.

For any peptide fragmentation annotations that do not come from the proposed peptide analyte(s) used to annotate other peaks in the same spectrum, the ion type and ordinal for any additional asserted interpretations MAY be indicated by following the peak annotation with a peptide sequence in curly braces, such as:

```
0@b2{LL}
0@y1{K}
0@b2{LC[Carbamidomethyl]}
0@b1{[Acetyl]-M}
0@y4{M[Oxidation]ACK}-CH4OS[M+H+Na]^2
```

The peptide sequence MAY contain modifications which MUST be specified using Unimod entry names in square brackets following the amino acid residue letter, as in the above examples. Note that the "Unimod entry name" corresponds to the "Name:" field in the Unimod OBO file (http://www.unimod.org/obo/unimod.obo). On the unimod.org web site, the "Unimod entry name" is the "PSI-MS Name" column if it is not null, or if null, then the "Interim Name" column. The exact capitalization as listed in Unimod SHOULD be used (mostly capital first letter as shown above) for consistency, but interpreting software SHOULD handle these in a case insensitive manner since operating in a case insensitive manner does not lead to conflation of different terms.

The amino acid count of the peptide sequence SHOULD NOT be greater than the ion series ordinal. The amino acid count of the peptide sequence MUST NOT be less than the ion series ordinal. This notation can be useful for annotating identifiable low-mass ions from contaminants that are clearly not associated with a specified analyte, or for marking sequence tags identified by de novo sequencing algorithms.

4.4.4 Internal fragment ions

Canonical internal fragments result from two amide bond cleavages – those forming b/y ions – of the peptide precursor ion. As such, they do not contain either terminus. An

internal fragment is denoted by the ion type ‘m’ (for “middle”). To describe an internal fragment, specify a range n_1:n_2 where n_1 is the ordinal (beginning with 1, counting from the N terminus) of the N-terminal amino acid residue of the fragment in the original peptide sequence, and n_2 is the ordinal (beginning with 1, counting from the N terminus) of the C-terminal amino acid residue of the fragment in the original peptide sequence.

For example, for the peptide precursor ion MYPEPTIDEK/2, the 1+ internal fragment ion of PEPT (the 3rd to 6th amino acids in the original peptide sequence) should be denoted by:

```
m3:6
```

Internal fragments of only one residue SHOULD be encoded as immonium ions, not as internal fragments of length 1. Even though b ions have the same masses as internal fragments with $n_1 = 1$, one MUST NOT denote any b ion as m1:n_2. In cases where there is an N- or C-terminal mass modification that is lost, it MUST be encoded as a neutral loss as described below. For example, for the precursor [Carbamidomethyl]-MYPEPTIDEK/2, a fragment ion that is MYP without the n-terminal Carbamidomethyl should be denoted as:

```
b3-C2H3NO
```

instead of:

```
m1:3
```

Other internal fragment ions are possible. For instance, one that results from the cleavage of the C-N bond (a b cleavage) on the N-terminal side and a C-C bond (an x cleavage) on the C-terminal side produces a “bx” ion. The “by” and “bx” (-CO) internal fragments are commonly seen in HCD spectra. These are encoded as a neutral loss/gain (see section 4.5 on Neutral Loss (or gain) below) of the corresponding canonical internal fragment, e.g.

```
m3:6-CO
```

The resulting neutral loss or gain from any internal ion can be found in the table below.

Backbone cleavage	x	y	z
a	No difference	+CO	+CHNO
b	-CO	No difference	+NH
c	-CHNO	-NH	No difference

Following the primary ion series convention, singly-charged internal fragment ions are not labeled with a charge component, but multiply-charged internal fragments should be labeled with ^N at the end, where N is 2, 3, 4, etc.

A doubly charged “bx” internal fragment with a water loss would be written

```
m3:6-CO-H2O^2
```

In case there are multiple instances of the same combination of residues (with differing order) of the internal fragment ion in the sequence of the precursor, all MAY be specified, and MAY be ordered from most likely to least likely. For example, for a precursor of MYPEPTIDEK/2, m3:4 is far more likely than m4:5, although they have the same mass and both might be produced. In this case one MAY annotate the peak as

m3:4/1.1ppm,m4:5/1.1ppm

On the other hand, if the multiple possible internal fragments are identical in sequence, only the annotation with the smallest ordinals SHOULD be used. For example, for the precursor GGAAAAAAK/2, the internal fragment ion of AAA SHOULD only be annotated as:

m3:5

since m3:5, m4:6, m5:7 and m6:8 are all equivalent.

4.4.5 Immonium ions

An internal fragment with just a single amino acid formed by a b/y cleavage on the N-terminal side and an a/x cleavage on the C-terminal side is called an immonium ion (forming an “a like” ion that is equivalent to the residue mass minus CO). Immonium ions are denoted by capital ‘I’, followed by the one-letter amino acid abbreviation. For example, the common tyrosine immonium and histidine immonium ions are denoted by:

```
IY
IH
```

There are other related ions from a single amino acid residue that are sometimes referred to as “immonium ions.” These will be handled as a neutral loss/gain of the corresponding true immonium ion (see section 4.5 on Neutral Loss below). For example, the leucine immonium ion can lose a CH2, which will be denoted by:

```
IL-CH2
```

Some immonium ions are derived from residues that already have a mass modification attached. In this case the modification MUST be specified as the Unimod entry name in

square brackets following the residue letter. For example, common immonium ions from carbamidomethylated cysteine and phosphorylated tyrosine are:

```
IC[Carbamidomethyl]
IY[Phospho]
```

Neutral losses may be added to these as usual.

4.4.6 Intact precursor ions

The intact, unfragmented, precursor ion, as well as its neutral losses, can often be found in the tandem mass spectrum in CID. For an MS3 spectrum, the precursor is the MS2 fragment ion selected for fragmentation for MS3. The precursor is denoted by the ion type ‘p’. For example, the intact 2+ precursor ion is denoted by:

```
p^2
```

As with all peak annotations, the omission of an adduct type specification (see Section 4.8) implies that 2 protons are added to the neutral precursor to yield the 2+ charge. For another example, the phosphate neutral loss of a 2+ precursor ion, which is often an intense peak in CID spectra of peptides containing a phosphorylated serine is given by:

```
p-H3PO4^2
```

Note that the charge state needs to be included explicitly, even though it is implied by the precursor charge state. This is needed to distinguish charge-reduced precursor ions, which are common in ETD spectra. For instance, in the ETD spectrum of a 4+ precursor ion, one might find the 4+ unfragmented precursor ion (p^4), a charge-reduced 3+ unfragmented precursor ion (p+H^3) after the capture of an electron, a charge-reduced 2+ unfragmented precursor ion (p+2H^2) after the capture of two electrons, and a charge-reduced 1+ unfragmented precursor ion (p+3H) after the capture of three electrons. ETD spectra can also have charge-reduced precursors due to lost protons (not always addition of electrons), so that would be denoted as fewer added hydrogens, e.g. p^2 for the loss of two protons from a 4+ precursor. The possible intact precursor ions from a 4+ precursor are:

Annotation	Adduct type	Description
p^4	[M+4H]4+	original 4+ precursor with four additional protons
p+H^3	[M+4H]3+	capture of one electron
p^3	[M+3H]3+	loss of one proton
p+2H^2	[M+4H]2+	capture of two electrons
p^2	[M+2H]2+	loss of two protons
p+H^2	[M+3H]2+	capture of one electron, loss of one proton
p+3H	[M+4H]+	capture of three electrons
p+2H	[M+3H]+	capture of two electrons, loss of one proton
p+H	[M+2H]+	capture of one electron, loss of two protons
p	[M+H]+	loss of three protons

To summarize, to distinguish the possible charge-reduced precursor ions, one attaches a "neutral gain" of hydrogens to the ion type whenever the number of protons is different from what one would expect from a purely protonated adduct at that charge state. As with the regular fragment ion types (b, y, etc.), a 1+ precursor does NOT get a charge suffix (i.e, use p instead of p^1).

WARNING: This is a different convention than used by the NIST MSP format where the original charged precursor did not get the charge suffix and all other charge states, including 1+, did.

4.4.7 Reference ions

Peptide MS2 spectra can contain a substantial number of non-peptidic fragment ions, and many of them are well known. In MS2-based labeled quantification strategies employing an isobaric tag, such as iTRAQ and TMT, fragment ions of the tags are used for quantification. The isobaric tag is typically attached to the N terminus or to the side-chain amino group of certain residues, and the cleavage of the tag releases the reporter ion. Typical examples of this are TMT and iTRAQ reporter ions, and other isobaric labeling reagent derived ions. In addition to isobaric tags, some peptide PTMs may also be labile, resulting in fragments corresponding to the protonated modification group which can be useful as a diagnostic marker for the PTM. Other reference ions derived from monosaccharides and nucleotides are often seen in peptide MS2 spectra.

Reference ions are prefixed with the letter 'r' with the name following in square brackets. The name MAY be a Unimod entry name, or defined in the in a version-controlled JSON document in GitHub that may be updated as the field advances. There is also a human-readable markdown version autogenerated from the JSON. The markdown version is viewable here:

https://github.com/HUPO-PSI/mzPAF/blob/main/specification/reference_data/reference_molecules.md

and the software-parsable JSON version is available here:

Viewable:
https://github.com/HUPO-PSI/mzPAF/blob/main/specification/reference_data/reference_molecules.json
Raw:
https://raw.githubusercontent.com/HUPO-PSI/mzPAF/main/specification/reference_data/reference_molecules.json

The current version of this list is provided as Appendix B of this document, but the above URLs should be checked for updates.

When appearing as independent ions, reference ions are prefixed with the letter ‘r’ with the name following in square brackets in order to provide a reliable handle for software parsers such as:

```
r[TMT127N]
r[iTRAQ114]
r[TMT6plex]
r[Hex]
r[Adenosine]
```

These names MAY contain paired “[“ and “]” characters. They are assumed to be protonated (as are other ions in the specification) (and thus heavier than the neutral mass by a proton and singly charged) unless followed by an adduct component that specifies some other charge-bearing moiety (e.g. [M+Na]). For example, the Unimod entry name of HexNAc(2) specifies a mass modification of 406.1587 Da. The corresponding 1+ protonated reference ion (i.e., the oxonium ion of GlcNAc-GlcNAc) denoted r[HexNAc(2)], should have a mass of 407.1660 Da.

4.4.8 Named compounds

If a fragment ion does not fit into the previous categories, yet is more easily understood by a name rather than a chemical formula or a SMILES string, it may be specified as a compound name. Such a name MUST be prefixed with an underscore (_) followed by a string enclosed in curly braces ({}). The string within curly braces does not need to be software-interpretable. It may be followed by neutral loss, isotope, adduct type, and charge information in the usual manner. As an example,

```
0@_{Urocanic Acid}
```

attributes the annotated peak as coming from a singly charged urocanic acid molecule (uncharged would not be detected, and doubly charged MUST be postfixed with ^2, etc.). The string after the underscore and enclosed in curly braces should be as informative and concise as possible. The string MUST imply the neutral molecule to which charge is then added. It is not obligatory that tools supporting this format are able to understand a peak annotation of 0@_{Urocanic Acid}+HPO3^2 as the 2+ ion of some molecule monophosphate and display it to a user, but a more formal specification MAY be developed in the future. Tools MAY encode a list of known named compounds to annotate, but this is not required. Readers of this notation MUST simply understand that it is some named compound ion, but are not obligated to understand more than that.

4.4.9 Chemical Formulas

Chemical formulas for identified peaks may be encoded if they are non-peptidic in origin. This may be useful for contamination peaks or for small molecule spectra where the small molecule is a named analyte. Chemical formulas MUST be prefixed by ‘f’,

enclosed in curly braces, and follow the formatting described below in the neutral loss subsection.

The chemical formula enclosed in the curly braces is understood to represent all the nuclei in the charged molecule. For example, if the compound is singly charged with a proton, there will be an extra H in the formula over what would be the neutral molecule. Adduct information MAY be specified as described in section 4.8 to denote the charged atoms; however, it does NOT add mass to the molecular formula. In the first line of the example below, C13H9 corresponds directly to all nuclei present in the ion and theoretical *m/z* is computed as the mass of C13H9 minus the mass of one electron (since it has a charge of 1+) to yield the *m/z* of 165.069988, and it may be assumed that one of the H atoms is a proton.

An example from MassBank https://massbank.eu/MassBank/RecordDisplay.jsp?id=SM858102 adapted for this format is as follows:

```
165.0698   104629.2  f{C13H9}/-0.55ppm
167.0730   479334.8  f{C12H9N}/0.06ppm
179.0726    82567.1  f{C13H9N}/-2.01ppm
180.0808 27526884.0  f{C13H10N}/-0.11ppm
181.0886   783300.1  f{C13H11N}/-0.09ppm
182.0965  6583053.0  f{C13H12N}/0.26ppm
192.0808   189835.2  f{C14H10N}/0.19ppm
193.0887    66613.2  f{C14H11N}/0.45ppm
208.0757  1080071.1  f{C14H10NO}/0.03ppm
```

Other suffixes such as for isotope and charge state, as described elsewhere in this document, may be used following the chemical formulae, e.g.:

```
f{C16H22O}+i^3
```

Stable isotopes can be encoded as in ProForma 2.0 by prefixing the atom with its isotopic number in square brackets. This notation MUST only be used for stably labeled molecules. Any peak that is isotopically related to another annotated peak MUST use the isotope notation as written in section 4.6.

```
f{C15[13C1]H22O}^3
```

The ProForma 2.0 specification[7] (https://psidev.info/proforma) provides further rules on how to encode molecular formulas regarding white space and recommended element ordering .

4.4.10 SMILES strings for chemical compounds

Chemical compounds may also be expressed using SMILES notation (https://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system). SMILES strings MUST be prefixed by ‘s’ and enclosed in curly braces. For example:

```
s{CN=C=O}[M+H]/-0.55ppm
s{COc(c1)cccc1C#N}[M+H+Na]^2/1.29ppm
```

Since the structure of the adduct ion is often unclear, the SMILES string MUST NOT include the charge-bearing moieties (H+, Na+, etc.) which may be attached at unspecified sites. Therefore, the SMILES string MUST correspond to the neutral molecule, and the charge-bearing moieties should be specified by the adduct notation as described in Section 4.8.

In the example above s{CN=C=O}[M+H] should be interpreted as the neutral molecule methyl isocyanate H3C-N=C=O with an additional proton attached to it at an unspecified location, to yield a 1+ charge (a charge state of 1+ is assumed for all peak annotations unless specified with a ^c suffix). This also means that the mass of the ion should be calculated from the neutral mass of methyl isocyanate plus that of a proton. Likewise, s{COc(c1)cccc1C#N}[M+H+Na]^2 denotes an adduct ion of 2+ charge state, in which the neutral molecule of COc(c1)cccc1C#N (3-methoxybenzonitrile) is attached to one proton and one Na+ ion at some unspecified sites.

Note that while the SMILES format allows for the specification of charged nuclei, it only supports this use case where the molecular structure is fully known. Although sometimes it is possible to place the charge-bearing moiety at an exact location with high confidence, in order to avoid confusion about how the mass should be calculated, the SMILES string MUST encode neutral molecules only. The charge state and the charge-bearing moiety(-ies) MUST be encoded in the adduct notation following the SMILES string, as described in Section 4.8. In addition, if SMILES strings are used, the [M+nH] suffix MUST always be included even if the adduct is only protonated (unlike for peptides for which protonated adducts are assumed if omitted).

For the purpose of mass calculation, it is assumed that what follows the M in the adduct notation are ions which should confer the correct charge state, i.e., [M+Na] means the addition of a 1+ sodium ion, not a sodium atom, whereas [M+HCOO] means the addition of formate HCOO-, not the neutral radical of HCOO. It should be noted that sometimes the adduct notation is used in the literature to also specify neutral loss/gain to the neutral analyte, in addition to the charge-bearing moiety(-ies). Since mzPAF already provides for a mechanism to specify neutral loss/gain (Section 4.5), it is recommended that what follows the M in the adduct notation SHOULD only include charge-bearing ions, and neutral loss/gain SHOULD NOT be included in the adduct notation. For example, s{OCCCC=OOH}-H2O[M+H] is preferred over s{OCCCC=OOH}[M-H2O+H] to denote 4-hydroxybutyric acid with a water loss (with exact structure unknown), charged with an additional proton.

If the ion is generated by the gain/loss of electrons only, then the adduct notations of [M+/-ne] MUST be used. For example, [M-e] stands for the 1+ ion generated when the neutral molecule specified by the SMILES string loses an electron. [M+2e] stands for the 2- ion generated when two electrons are added to the neutral molecule. The shorthand of [M+] to mean [M-e] MUST NOT be used.

In this version of the specification, only SMILES is supported as opposed to InChI, because SMILES is more human readable. If both SMILES strings and chemical formulas are provided, they MAY be equivalents or alternatives.

4.5 Neutral losses

The neutral loss (or gain) component MAY be denoted as a string of 0 to n loss (or gain) components, described by their molecular formula. Most losses are negative and preceded by a minus sign. However, neutral gains may be signified with a plus sign. Double (and triple and up) losses SHOULD be preceded by an integer indicating the number (e.g., -2H2O signifies a double water loss). Single losses MUST NOT have a preceding 1.

Reference group names in square brackets, which can either be defined in the version-controlled JSON document as described in Section 4.4.7 or a Unimod entry name, MAY also be used as a neutral loss. Common use cases of reference group neutral losses are the losses of glycan groups in the fragmentation of glycopeptides, and of the quantitation tag in the fragmentation of TMT- or iTRAQ-labeled peptides. Here, the lost mass is equal to the neutral mass of the reference group, e.g.:

p-[Hex]

which denotes the precursor ion with a neutral loss of about 162 Da (C6H10O5, corresponding to the “Hex” modification in Unimod).

The following are a table of common neutral losses/gains. If the neutral loss/gain in question is listed below, they SHOULD be followed as a matter of convention (e.g., do not write an ammonia loss (NH3) as H3N). If the neutral loss/gain in question is not listed, one may prescribe new ones as chemical formulae as described in section 4.4.9 . This table is not complete, and other losses are possible.

Neutral loss/ gain group	Common name	Exact mass (monoisotopic)	Remark
H	Hydrogen	1.007825	e.g., for specifying hydrogen transfer from c to z ions in ETD
NH3	Ammonia	17.026549	From amine groups
H2O	Water	18.010565	From -OH and -COOH groups

CO	Carbon monoxide	27.994915	For backbone fragments, use “a” instead of “b-CO”. But for internal fragments, use “mx:y-CO”. Also seen as a neutral loss from formylated serine or threonine.
CO2	Carbon dioxide	43.989829	From -COOH groups
HCONH2	Formamide	45.021464	From -CONH2 groups
HCOOH	Formic acid	46.005479	From -COOH groups
CH4OS	Methanesulfenic acid	63.998301	From oxidized methionine
SO3	Sulfur trioxide	79.956818	From sulfotyrosine
HPO3	Metaphosphoric acid	79.966331	From phosphotyrosine and sometimes from phosphoserine and phosphothreonine
C2H5NOS	Mercaptoacetamide	91.009195	From carbamidomethyl cysteine
C2H4O2S	Mercaptoacetic acid	91.993211	From carboxymethyl cysteine
H3PO4	Phosphoric acid	97.976896	From phosphoserine and phosphothreonine
[reference group] Note: The square bracket is mandatory for non-chemical formulae.		(variable)	See the reference ion s section above . The reference group name inside the square bracket can either be a Unimod entry name, or defined in the aforementioned version-controlled JSON document.

Such neutral gains and losses may be strung together as in these examples:

```
y2+CO-H2O
y2-H2O-NH3
p-[Hex]
p-[TMT6plex]-2H2O-HPO3

p-2[iTRAQ115]
p-[iTRAQ116]-CO-H2O-HPO3
```

etc.

If there are multiple neutral gains/losses, alphanumeric order SHOULD be followed, e.g. y2-H2O-NH3 rather than y2-NH3-H2O. Annotations such as y2-H2O-NH3 and y2-NH3-H2O are considered identical, and SHOULD not be listed as multiple plausible annotations.

4.6 Isotopes

The isotope component is optional. If the monoisotopic ion is being described, then there MUST NOT be any isotope component. However, if another isotope is being described, then this component MUST be "+ni" or "-ni" where n is the isotope number above or below the monoisotope; however, an n of 1 SHOULD be suppressed, following the precedent from the NIST MSP format. Examples are: +i, +2i, +3i, -i, -2i, etc.

This notation does not differentiate among isotopes of different constituent atoms of the fragment (C, H, N). The theoretical mass of the +i isotope is taken to be the difference between 13C and 12C (1.003355 Da) greater than that of the parent fragment ion, where this delta depends on the elemental composition and relative isotope ratios.

4.7 Adduct Type

The adduct type component is optional. Typical conditions for peptide ionization for proteomics generates protonated cations [M+H], [M+2H], etc, which are implied if the adduct type component is omitted. When the adduct type is omitted, the number of added protons is taken to be equal to the charge state, which is specified with the ^ notation as described in the following section. However, under some conditions other charged atoms convey the charge, such as the sodiated peptide ion. If the fragment ion is anything except a purely protonated adduct, it MUST be specified in the form [M+nA] or [M-nA] where M is the neutral fragment being annotated, A is the atom/molecule added to the neutral fragment to form the ion, and n is the number of specified atoms/molecules added. If more than one kind of atom/molecule is added, it will be specified by [M+n_1A+n_2B…], etc. Some examples are:

[M+Na] denotes a sodiated adduct ion

[M+NH4] denotes an ammonium adduct ion

[M+2Na] denotes an adduct ion with two sodium atoms (which MUST be followed by ^2 to specify the charge state, see above under "Charge State")

[M+2H+Na] denotes an adduct ion with two hydrogen atoms and one sodium atom (which MUST be followed by ^3 to specify the charge state, see below under "Charge State")

If there are multiple types of atoms/molecules, alphabetical order SHOULD be followed, e.g. [M+2H+Na] rather than [M+Na+2H].

A useful reference for adducts may be found at the web page: https://fiehnlab.ucdavis.edu/staff/kind/Metabolomics/MS-Adduct-Calculator/. Note that at the time of this writing at this web page, the concept of mass and *m*/*z* are conflated. Table columns that are labeled "mass" are actually "*m*/*z*".

Complete Examples:

A sodiated y4 singly charged ion is written:

```
y4[M+Na]
```

A doubly charged y5 ion with one charge from Na+ and one from a proton and a water loss:

```
y5-H2O[M+H+Na]^2
```

IMPORTANT NOTE: In this context, the M represents the annotated neutral fragment, NOT the precursor or some other parent analyte. In the above example, M represents a neutral y5-H2O. Therefore, it is not valid to specify 2M to indicate that the primary analyte has formed a dimer, or similar.

4.8 Charge state

If the charge is 1+ (most common), this component MUST be suppressed. If the charge is not 1+, then the charge MUST be provided as ^n where n is the charge number (without a + symbol). Examples are ^2, ^3, etc.

Charge 0 MUST NOT be used. If the charge state is not known, the peak annotation SHOULD be marked as '?'.

For spectra acquired in the negative ion mode, the spectrum identification should be a negative precursor ion. In this case, the charge state is interpreted to be negative n. The charge state component in the peak annotation MUST NOT include the minus sign.

WARNING: This is a different rule than implemented in the NIST MSP format, where the precursor p does not carry a charge component if it is the fully charged unfragmented precursor, but does include a charge if it is a charge-reduced precursor, even a charge 1+. For example, in NIST MSP, a singly charged charge-reduced precursor is p^1 and the doubly charged original precursor is just p. The NIST MSP convention MUST NOT be used in this standard.

4.9 Multiple peaks associated with the same fragment ion

As a general rule, a specific peak annotation SHOULD be placed on one peak only (the most likely one), even if multiple peaks might be within some tolerance around the theoretical fragment ion *m*/*z*, to avoid excessive cluttering. However, this format allows for placing the same peak annotation on multiple peaks, which may be useful in some cases, according to the following guideline.

The peak annotation string, the format of which is described above, will still be placed on the most likely peak. Any other peaks that are regarded as possibly belonging to the same fragment ion MAY be assigned the same peak annotation string prefixed with an ampersand (&) before the analyte identifier (if one is provided), with the *m*/*z* deviation changed accordingly. For example:

```
677.298    69       &1@y7/-0.002
677.299    572      &y7/-0.001
677.300    5681     y7/0.000*0.95
677.301    1320     &y7/0.001
677.302    240      &y7/0.002
677.303    34       b6-H2O/-0.005,&y7/0.003
```

This can be used as the mechanism to annotate multiple peaks belonging to one fragment ion in profile-mode data. It can also be used if there is more than one centroided peak within the tolerance of the annotation and it is unclear which peak might best correspond to the ion annotation. However, the present format does not mandate such kind of peak annotations.

4.10 Confidence estimates for the annotations

Annotations MAY be annotated at the end with a confidence estimate by placing an asterisk followed by a number that MUST be between 0.0 and 1.0, inclusive, with 1.0 being the highest confidence. This signifies that confidence or probability that the offered annotation is correct. This example:

```
y12-H2O^2/7.4ppm*0.70
```

indicates that the offered annotation is only judged to be 70% likely by the interpreting software (perhaps based on a model of mass deltas). To allow for the use of other confidence measures with more precise statistical definitions, a proper CV term of such a metric can be defined as a child of MS:1003274 “peak annotation confidence metric”, and specified in the spectrum metadata.

If there is no peak annotation confidence metric specified, this number SHOULD be interpreted as the probability of the peak being this annotation (in the Bayesian sense that the probability indicates a degree of belief, which may be subjective). If multiple annotations are offered, the one with the highest confidence SHOULD come first. If

multiple annotations are present, they SHOULD all have a confidence estimate or none should. If there are multiple annotations with confidence scores, the confidence scores MUST sum to a number equal to or less than 1.0. The difference between the sum and 1.0 MAY be interpreted to mean that the confidence that the true source of the peak is something else not listed. The following:

```
y12/3.4ppm*0.85,b9-NH3/5.2ppm*0.05
```

would signify that the first annotation is judged to be 85% likely and the second 5% likely, with the balance of 10% reserved for some other origin not listed.

5. Object Model

All the above data elements MAY be encoded in the following object model in memory or some other data serialization format such as JSON. The formal JSON Schema definition is available at https://github.com/HUPO-PSI/mzPAF/blob/main/specification/annotation-schema.json.

5.1 Definition

analyte_reference:
type: integer|null
description: Label of analyte to which this annotation belongs.
required: true
default: null
molecule_description:
type: molecule_description_type
description: A description of the molecule or molecule fragment that this peak is annotated with
required: true
neutral_loss:
type: array[string]
description: Any additional gains or losses of chemical groups defined by formula or by name. Multiple may be specified.
required: false
isotope:
type: integer
description: An isotopic peak offset from the monoisotopic peak
required: false
default: 0
mass _error:
type: value-unit object|null
description: Error between observed and theoretical mass
value:
type: number

unit:
type: string
required: false
confidence:
description: Number defining confidence in peak annotation. Higher is better. 1.0 is the highest confidence level, while 0.0 is the lowest.
type: number|null
required: false
adduct:
type: array[string]
description: The charge carrier(s) for the given annotation
required: false
charge:
type: integer
description: The charge state of the ion generating this peak. This value is unsigned
required: false
default: 1
additionalProperties: true

molecule_description_type: one of:
peptide:
series: The peptide ion series this ion belongs to
position: The position from the appropriate terminal along the peptide this ion was fragmented at (starting with 1)
series_label: peptide
internal:
start_position: N-terminal amino acid residue of the fragment in the original peptide sequence (beginning with 1, counting from the N-terminus)
end_position: C-terminal amino acid residue of the fragment in the original peptide sequence (beginning with 1, counting from the N-terminus)
series_label: internal
precursor:
series_label: precursor
immonium:
amino_acid: The amino acid represented by this immonium ion
modification: Optional modification that may be attached to this immonium ion
series_label: immonium
reporter:
reporter_label: The labeling reagent's name or channel information
series_label: reporter
named_compound:
label: The name of the compound
series_label: named_compound
formula:
formula: The elemental formula of the ion being marked

series_label: formula

smiles:

smiles: The SMILES string of the ion being marked
series_label: smiles

unknown:

series_label: unknown
unknown_label: An optional digital label for an unknown peak

5.2 Examples

```
1@y7-H2O+i^2[M+NH4]/-0.2ppm*0.5
```


```
{
 "adducts": [
  "NH4"
 ],
 "analyte_reference": "1",
 "charge": 2,
 "confidence": 0.5,
 "isotope": 1,
 "mass_error": {
  "unit": "ppm",
  "value": -0.2
 },
 "molecule_description": {
  "position": 7,
  "series": "y",
  "series_label": "peptide"
 },
 "neutral_losses": [
  "-H2O"
 ]
}
```


```
m5:8-H2O/14.4ppm
```


```
{
 "neutral_loss": ["-H2O"],
 "isotope": 0,
 "adduct":[],
 "charge": 1,
 "analyte_reference": 1,
 "mass_error": {
  "value": 14.4,
  "unit": "ppm"
```

```
  },
  "confidence": null,
  "molecule_description": {
    "series_label": "internal",
    "start_position": 5,
    "end_position": 8
  }
}
```


`p/-1.7ppm`


```
{
  "neutral_loss":[],
  "isotope": 0,
  "adduct":[],
  "charge": 1,
  "analyte_reference": 1,
  "mass_error": {
    "value": -1.7,
    "unit": "ppm"
  },
  "confidence": null,
  "molecule_description": {
    "series_label": "precursor"
  }
}
```


6. Regular Expressions

The compact ion notation may be encoded via the following regular expressions:

ECMAScript Regex


```
^(?<is_auxiliary>&)?(?:(?<analyte_reference>\d+)@)?(?:(?:(?
<series>(?:da|db|wa|wb)|[axbyczdwv]\.?)(?<ordinal>\d+)(?:\{
(?<sequence_ordinal>.+)\})?)|(?<series_internal>[m](?<inter
nal_start>\d+):(?<internal_end>\d+)(?:\{(?<sequence_interna
l>.+)\})?)|(?<precursor>p)|(:?I(?<immonium>[ARNDCEQGHKMFPST
WYVIL])(?:\[(?<immonium_modification>(?:[^\]]+))\])?)|(?<re
ference>r(?:(?:\[(?<reference_label>[^\]]+)\]))))|(?:f\{(?<f
ormula>[A-Za-z0-
9\[\]]+)\})|(?:_\{(?<named_compound>[^\{\}\s,/]+)\})|(?:s\{
(?<smiles>[^\}]+)\})|(?:(?<unannotated>\?)(?<unannotated_la
bel>\d+)?))(?<neutral_losses>(?:[+-]\d*(?:(?:[A-Z][A-Za-z0-
9]*)|(?:\[(?:(?:[A-Za-z0-9:\.]+)(?:\[(?:[A-Za-z0-9\.:\-\
```

```
]+)\])?)\])))+)?(?:(?<isotope>[+-
]\d*)i)?(?:\[(?<adducts>M(:?[+-]\d*[A-Z][A-Za-z0-
9]*)+)\])?(?:\^(?<charge>[+-]?\d+))?(?:\/(?<mass_error>[+-
]?\d+(?:\.\d+)?)(?<mass_error_unit>ppm)?)?(?:\*(?<confidenc
e>\d*(?:\.\d+)?))?
```

Python SRE in verbose mode

```
^(?P<is_auxiliary>&)?
   (?:(?P<analyte_reference>\d+)@)?

(?:(?:(?P<series>(?:da|db|wa|wb)|[axbyczdwv]\.?)(?P<ordinal
>\d+)(?:\{(?P<sequence_ordinal>.+)\})?)|

(?P<series_internal>[m](?P<internal_start>\d+):(?P<internal
_end>\d+)(?:\{(?P<sequence_internal>.+)\})?)|
   (?P<precursor>p)|

(:?I(?P<immonium>[ARNDCEQGHKMFPSTWYVIL])(?:\[(?P<immonium_m
odification>(?:[^\]]+))\])?)|
   (?P<reference>r(?:
    (?:\[
        (?P<reference_label>[^\]]+)
    \])
   ))|
   (?:f\{(?P<formula>[A-Za-z0-9\[\]]+)\})|
   (?:_\{
    (?P<named_compound>[^\{\}\s,/]+)
    \})|
   (?:s\{(?P<smiles>[^\}]+)\})|
   (?:(?P<unannotated>\?)(?P<unannotated_label>\d+)?)
)
(?P<neutral_losses>(?:[+-]\d*
    (?:(?:[A-Z][A-Za-z0-9]*)|
        (?:\[
            (?:
                (?:[A-Za-z0-9:\.]+)(?:\[(?:[A-Za-z0-9\.:\-\
]+)\])?
            )
            \])
    )
)+)?
(?:(?P<isotope>[+-]\d*)i)?
(?:\[(?P<adducts>M(:?[+-]\d*[A-Z][A-Za-z0-9]*)+)\])?
(?:\^(?P<charge>[+-]?\d+))?
(?:/(?P<mass_error>[+-
]?\d+(?:\.\d+)?)(?P<mass_error_unit>ppm)?)?
```

```
(?:\*(?P<confidence>\d*(?:\.\d+)?))?
```

6.1 Formal Grammar for the Peak Annotation Format

In addition to the regular expression, we provide two alternative presentations of the peptide peak annotation format to either aid understanding or guide implementation.

Parsing state machine diagrams:
https://github.com/HUPO-PSI/mzPAF/blob/main/specification/grammars/grammar.md

Grammar:
https://github.com/HUPO-PSI/mzPAF/blob/main/specification/grammars/annotation.lark

7. Pending Issues - Future developments

There are several use cases that are NOT currently supported in the current version of the specification. These complications are left open and will ideally be addressed in future versions, after the community has gained more experience with the common cases. The objective here is to document those cases appropriately and, in some cases, to propose some possible solutions for representing the information in future versions of mzPAF.

7.1 Side-chain fragments and other fragment ions

This format currently does not allow for the specification of side-chain fragments (which are important for glycopeptides, for example) and other fragments (unless they are simple chemical formulas). It also does not have a mechanism to denote fragments of cross-linked peptides. Moreover, in the case of metabolites/small molecules, there is no notion of a backbone and the fragment will need to be specified by a chemical formula.

To accommodate these other kinds of molecules, separate peak annotation formats will need to be defined, similar to this document. We anticipate that in the future, a number of peak annotation formats will be defined and put into use.

Without a separate peak annotation format, the prefix '_' (see "External Fragment Ions" above) can be used to denote any fragment not covered by this format. Software tools supporting this format can choose to ignore such peak annotations, or merely display them to the user as-is.

7.2 Specifying isotope origins

It is currently not allowed to specify from which element an isotope originated, as outlined in 4.6. This could be extended to allow specifying the originating element. This use case is not fully understood at the moment of writing as most mass spectrometry data is not of

sufficient resolution to accurately identify the different isotopes, for example tell [13C6][15N2] from [13C7][15N1]. Suggested syntax to specify the element is ‘+niEZ’ where E is the element, eg ‘+iC13’ or ‘+3iN15’. For the complex cases with multiple isotopes from different elements these could be listed sequentially, eg ‘+6iC13+2iN15’.

7.3 Specifying the backbone cleavages for internal ions

Internal fragments can be formed from all combinations of backbone cleavage. The current specification defaults to ‘by’ and specifies that any other backbone cleavage should be written as the corresponding neutral loss or gain from this reference point. Extending the specification to allow to directly specify which backbones led to the internal ion could be done, this would enhance the clarity for internal fragments annotations that are not formed by ‘by’. Some syntax suggestions were to append the backbone cleavage directly after the’ m’ tag ‘max3:6’, ‘mcz3:6’, another way would be to specify the backbone on the position ‘ma3:x6’, ‘mc3:z6’. If support for this notation the ‘m3:6’ would be left in as meaning any of ‘ax’/’by’/’cz’.

8. Appendix A. Parsing multiple annotations strategy

If the provided regular expression-based annotation parser is used, additional logic is required to handle multiple annotations. A procedure like the following pseudo code SHOULD be applied:

-

```
def unpack_match(match):
    ...

def match_pattern(text):
    ...

def parse_annotation_string(text):
    i = 0
    annotations = []
    n = len(text)
    while i < n:
        match = match_pattern(text[i:])
        if not match:
            raise Exception(f"{text[i:]} does not match annotation pattern!")
        annot = unpack_match(match)
        i_end = match.end()
        if i_end < n:
            if text[i_end] == ',':
                i_end += 1
            else:
                raise Exception(f"Unparsed content following annotations " +
                        f" starting at {i_end}")
        i = i_end
        annotations.append(annot)
    return annotations
```

9. Appendix B. Reference molecules

The most up-to-date list of reference molecules can be found here:

https://github.com/HUPO-PSI/mzPAF/blob/main/specification/reference_data/reference_molecules.md

and a JSON serialization with *m*/*z* and other information is available here:

Viewable:
https://github.com/HUPO-PSI/mzPAF/blob/main/specification/reference_data/reference_molecules.json
Raw:

https://raw.githubusercontent.com/HUPO-PSI/mzPAF/main/specification/reference_data/reference_molecules.json

For the convenience of reading this specification, the state of this file as of version 1.0 of the specification is provided below, although the URLs above should be checked for updates:

Name	Label type	Molecule type	Chemical formula	Ion m/z	Neutral mass
TMT126	TMT	reporter	C8N1H15	126.128	
TMT127N	TMT	reporter	C8[15N1]H15	127.125	
TMT127C	TMT	reporter	C7[13C1]N1H15	127.131	
TMT128N	TMT	reporter	C7[13C1][15N1]H15	128.128	
TMT128C	TMT	reporter	C6[13C2]N1H15	128.134	
TMT129N	TMT	reporter	C6[13C2][15N1]H15	129.131	
TMT129C	TMT	reporter	C5[13C3]N1H15	129.138	
TMT130N	TMT	reporter	C5[13C3][15N1]H15	130.135	
TMT130C	TMT	reporter	C4[13C4]N1H15	130.141	
TMT131N	TMT	reporter	C4[13C4][15N1]H15	131.138	
TMT131C	TMT	reporter	C3[13C5]N1H15	131.144	
TMT132N	TMT	reporter	C3[13C5][15N1]H15	132.142	
TMT132C	TMT	reporter	C2[13C6]N1H15	122.148	
TMT133N	TMT	reporter	C2[13C6][15N1]H15	133.145	
TMT133C	TMT	reporter	C1[13C7]N1H15	133.151	
TMT134N	TMT	reporter	C1[13C7][15N1]H15	134.148	
TMT134C	TMT	reporter	[13C8]N1H15	134.155	
TMT135N	TMT	reporter	[13C8][15N1]H15	135.152	
TMTzero	TMTzero	reporter+balance	C12H20N2O2	225.16	224.152
TMTpro_zero	TMTpro_zero	reporter+balance	C15H25N3O3	296.197	295.19
TMT2plex	TMT2plex	reporter+balance	C11[13C1]H20N2O2	226.163	225.156
TMT6plex	TMT6plex	reporter+balance	C8[13C5]H20N1[15N1]O2	230.17	229.163
TMTpro	TMTpro	reporter+balance	C8[13C7]H25[15N2]N1O3	305.214	304.207
iTRAQ113	iTRAQ	reporter	C6N2H12	113.108	
iTRAQ114	iTRAQ	reporter	C5[13C1]N2H12	114.111	
iTRAQ115	iTRAQ	reporter	C5[13C1]N1[15N1]H12	115.108	
iTRAQ116	iTRAQ	reporter	C4[13C2]N1[15N1]H12	116.112	
iTRAQ117	iTRAQ	reporter	C3[13C3]N1[15N1]H12	117.115	
iTRAQ118	iTRAQ	reporter	C3[13C3][15N2]H12	118.112	
iTRAQ119	iTRAQ	reporter	C2[13C4][15N2]H12	119.115	
iTRAQ121	iTRAQ	reporter	[13C6][15N2]H12	121.122	
iTRAQ4plex	iTRAQ4plex	reporter+balance	C4[13C3]N1[15N1]O1H12	145.109	144.102
iTRAQ8plex	iTRAQ8plex	reporter+balance	C7[13C7]N3[15N1]O3H24	305.213	304.205
TMT126-ETD	TMT	reporter	C7N1H15	114.128	
TMT127N-ETD	TMT	reporter	C7[15N1]H15	115.125	
TMT127C-ETD	TMT	reporter	C6[13C1]N1H15	114.128	
TMT128N-ETD	TMT	reporter	C6[13C1][15N1]H15	115.125	
TMT128C-ETD	TMT	reporter	C5[13C2]N1H15	116.134	
TMT129N-ETD	TMT	reporter	C5[13C2][15N1]H15	117.131	
TMT129C-ETD	TMT	reporter	C4[13C3]N1H15	116.134	
TMT130N-ETD	TMT	reporter	C4[13C3][15N1]H15	117.131	
TMT130C-ETD	TMT	reporter	C3[13C4]N1H15	118.141	
TMT131N-ETD	TMT	reporter	C3[13C4][15N1]H15	119.138	
TMT131C-ETD	TMT	reporter	C2[13C5]N1H15	118.141	
sidechain_A		sidechain	C1H3		15.0235

sidechain_C		sidechain	C1H3S1		46.9955
sidechain_D		sidechain	C2H2O2		58.0055
sidechain_E		sidechain	C3H4O2		72.0211
sidechain_F		sidechain	C7H7		91.0548
sidechain_G		sidechain	H1		1.00782
sidechain_H		sidechain	C4H5N2		81.0453
sidechain_I		sidechain	C4H9		57.0704
sidechain_J		sidechain	C4H9		57.0704
sidechain_K		sidechain	C4H10N1		72.0813
sidechain_L		sidechain	C4H9		57.0704
sidechain_M		sidechain	C3H7S1		75.0268
sidechain_N		sidechain	C2H4N1O1		58.0293
sidechain_O		sidechain	C9H17N2O1		169.134
sidechain_Q		sidechain	C3H6N1O1		72.0449
sidechain_R		sidechain	C4H10N3		100.087
sidechain_S		sidechain	C1H3O1		31.0184
sidechain_T		sidechain	C2H5O1		45.034
sidechain_U		sidechain	C1H3Se1		94.94
sidechain_V		sidechain	C3H7		43.0548
sidechain_W		sidechain	C9H8N1		130.066
sidechain_Y		sidechain	C7H7O1		107.05

name	**molecule_type**	**neutral_mass**	**chemical_formula**	**label_type**	**ion_mz**
Hex	monosaccharide	162.053	C6H10O5		
HexNAc	monosaccharide	203.079	C8H13N1O5		
dHex	monosaccharide	146.058	C6H10O4		
NeuAc	monosaccharide	291.095	C11H17N1O8		
NeuGc	monosaccharide	307.09	C11H17N1O9		
TMT126	reporter			TMT	126.128
TMT127N	reporter			TMT	127.125
TMT127C	reporter			TMT	127.131
TMT128N	reporter			TMT	128.128
TMT128C	reporter			TMT	128.134
TMT129N	reporter			TMT	129.131
TMT129C	reporter			TMT	129.138
TMT130N	reporter			TMT	130.135
TMT130C	reporter			TMT	130.141
TMT131N	reporter			TMT	131.138
TMT131C	reporter			TMT	131.144
TMT132N	reporter			TMT	132.142
TMT132C	reporter			TMT	122.148
TMT133N	reporter			TMT	133.145
TMT133C	reporter			TMT	133.151
TMT134N	reporter			TMT	134.148
TMT134C	reporter			TMT	134.155
TMT135N	reporter			TMT	135.152

TMTzero	reporter+balance	224.152		TMTzero	225.16
TMTpro_zero	reporter+balance	295.19		TMTpro_zero	296.197
TMT2plex	reporter+balance	225.156		TMT2plex	226.163
TMT6plex	reporter+balance	229.163		TMT6plex	230.17
TMTpro	reporter+balance	304.207		TMTpro	305.214
iTRAQ113	reporter			iTRAQ	113.108
iTRAQ114	reporter			iTRAQ	114.111
iTRAQ115	reporter			iTRAQ	115.108
iTRAQ116	reporter			iTRAQ	116.112
iTRAQ117	reporter			iTRAQ	117.115
iTRAQ118	reporter			iTRAQ	118.112
iTRAQ119	reporter			iTRAQ	119.115
iTRAQ121	reporter			iTRAQ	121.122
iTRAQ4plex	reporter+balance	144.102		iTRAQ4plex	145.109
iTRAQ8plex	reporter+balance	304.205		iTRAQ8plex	305.213
TMT126-ETD	reporter			TMT	114.128
TMT127N-ETD	reporter			TMT	115.125
TMT127C-ETD	reporter			TMT	114.128
TMT128N-ETD	reporter			TMT	115.125
TMT128C-ETD	reporter			TMT	116.134
TMT129N-ETD	reporter			TMT	117.131
TMT129C-ETD	reporter			TMT	116.134
TMT130N-ETD	reporter			TMT	117.131
TMT130C-ETD	reporter			TMT	118.141
TMT131N-ETD	reporter			TMT	119.138
TMT131C-ETD	reporter			TMT	118.141

10. 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

Douwe Schulte
Utrecht University
d.schulte@uu.nl

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

11. 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

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

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

12. 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).

13. Copyright Notice

14. Glossary

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

15. References

1. Lam H, Deutsch EW, Eddes JS, Eng JK, Stein SE, Aebersold R. Building consensus spectral libraries for peptide identification in proteomics. Nat Methods. 2008 Oct;5(10):873–875. PMCID: PMC2637392

2. Desiere F, Deutsch EW, King NL, Nesvizhskii AI, Mallick P, Eng J, Chen S, Eddes J, Loevenich SN, Aebersold R. The PeptideAtlas project. Nucleic Acids Res. 2006 Jan 1;34(Database issue):D655-658. PMCID: PMC1347403

3. Roepstorff P, Fohlman J. Proposal for a common nomenclature for sequence ions in mass spectra of peptides. Biomed Mass Spectrom. 1984 Nov;11(11):601. PMID: 6525415

4. Biemann K. Appendix 5. Nomenclature for peptide fragment ions (positive ions). Methods Enzymol. 1990;193:886–887. PMID: 2074849

5. 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

6. 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

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