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*

January 10, 2023

**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 in peptides. Distribution is unlimited.

Version: Draft 5 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/>.

Contents

[Abstract 1](#_Toc124242590)

[1. Introduction 4](#_Toc124242591)

[1.1 Description of the need 4](#_Toc124242592)

[1.2 Requirements 4](#_Toc124242593)

[2. Notational Conventions 5](#_Toc124242594)

[3. The Peak Annotation Format Definition 5](#_Toc124242595)

[3.1 The documentation 5](#_Toc124242596)

[3.2 Relationship to other specifications 5](#_Toc124242597)

[4. The Basic Form of the Peak Annotation Format 5](#_Toc124242598)

[4.1 Annotation of multiple analytes 6](#_Toc124242599)

[4.2 Multiple annotations 7](#_Toc124242600)

[4.3 Deviation of observed m/z from the theoretical m/z values 7](#_Toc124242601)

[4.4 Ion notation 8](#_Toc124242602)

[**4.4.1** Ion types overview 8](#_Toc124242603)

[**4.4.2** Unknown ions 9](#_Toc124242604)

[**4.4.3** Primary series ions 9](#_Toc124242605)

[**4.4.4** Internal fragment ions 10](#_Toc124242606)

[**4.4.5** Immonium ions 11](#_Toc124242607)

[**4.4.6** Intact precursor ions 12](#_Toc124242608)

[**4.4.7** Isobaric labelling related ions 13](#_Toc124242609)

[**4.4.8** External fragment ions 14](#_Toc124242610)

[**4.4.9** Chemical Formulas 14](#_Toc124242611)

[**4.4.10** SMILES strings for chemical compounds 15](#_Toc124242612)

[4.5 Neutral losses 16](#_Toc124242613)

[4.6 Isotopes 17](#_Toc124242614)

[4.7 Charge state 18](#_Toc124242615)

[4.8 Adduct Type 18](#_Toc124242616)

[4.9 Multiple peaks belonging to the same fragment ion 19](#_Toc124242617)

[4.10 Confidence estimates for the annotations 20](#_Toc124242618)

[5. Object Model 20](#_Toc124242619)

[5.1 Definition 20](#_Toc124242620)

[5.2 Examples 22](#_Toc124242621)

[6. Regular Expressions 24](#_Toc124242622)

[6.1 Formal Grammar for the Peak Annotation Format 25](#_Toc124242623)

[7. Pending Issues - Future developments 25](#_Toc124242624)

[7.1 Side-chain fragments and other fragment ions 25](#_Toc124242625)

[8. Appendix A. Parsing multiple annotations strategy 26](#_Toc124242626)

[9. Appendix B. Isobaric Label Ions 27](#_Toc124242627)

[10. Appendix C. Complex Neutral Loss Groups 29](#_Toc124242628)

[11. Oligosaccharides 29](#_Toc124242629)

[12. Authors Information 30](#_Toc124242630)

[13. Contributors 30](#_Toc124242631)

[14. Intellectual Property Statement 32](#_Toc124242632)

[15. Copyright Notice 32](#_Toc124242633)

[16. Glossary 33](#_Toc124242634)

[17. References 33](#_Toc124242635)

# Introduction

## 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 NIST (National Institute for Standards and Technology) (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 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. Participants from NIST and SpectraST have led the development of this standard.

This format, as currently described, is designed for unbranched 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, glycopeptides, cross-linked peptides...), 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 Dagstuhl/PSI journal article “Expanding the Use of Spectral Libraries in Proteomics”.3

## 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 unbranched 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.

# 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.

# The Peak Annotation Format Definition

## 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 example annotated spectra that demonstrate the use of mzPAF

## 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 containsrules on how to encode peptidoforms and molecular formulas

# 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 currently no limit in its maximum length. Line breaks MUST NOT be used.

The basic format of each annotation is:

*annotation1/delta,annotation2/delta*,...

or

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

e.g.

b2-18/3.2ppm,b4-18^2/3.2ppm

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:

b2-18/3.2ppm\*0.75,b4-18^2/3.2ppm\*0.25

The sections below define each component of these annotations.

## 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 number 0 is reserved as one or more unspecified contaminant molecules (see example below for common y1 ion observations for unidentified contaminant peptide).

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. 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. Most high S/N HCD spectra of tryptic digests contain the y1+ ions corresponding to both lysine and to arginine. If there is only one analyte defined, peak annotations SHOULD NOT be annotated with 1@. If analyte 1 is a peptide ending in R, it will not be uncommon to see:

0@IK+CO

0@IK+CO+H2O

which is the lysine y1 ion and the y1-H2O ion corresponding to some unspecified contaminating peptide ion ending with a lysine.

## 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.

There is no formal requirement about the order of the annotations, if there are more than one. However, they SHOULD be ordered by likelihood (based on existing knowledge about fragmentation), e.g. a primary series ion should be listed before a rare neutral loss.

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

* If the string starts with a ‘[‘, skip it and record it as an alternate annotation. An alternate notation MUST be closed with ‘]’.
* 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.

## Deviation of observed m/z from the theoretical m/z values

Each annotation SHOULD include an m/z delta representing the observed m/z – theoretical m/z (as calculated from the sum of the atoms charged particles of the annotated peak), unless the m/z values provided are all theoretical values anyway, as in the case for a library of predicted spectra. 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

## 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)(charge)(adduct type)

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^2[M+H+Na]

Here, a peak from the second isotope of a doubly charged protonated and sodiated y4 ion with a water loss is annotated.

### 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. Thus far, the specification is nearly entirely peptide centric. Although there is desire to support small molecules, lipids, and glycans in the future as well, it has not yet been decided on an accepted nomenclature for specifying such ion types when this current specification was developed. The following is a list of ion type prefixes, as described in detail in the subsections below:

|  |  |
| --- | --- |
| Prefix | Description |
| ? | Unknown ion |
| a | Peptide a series |
| b | Peptide b series |
| c | Peptide c series |
| x | Peptide x series |
| y | Peptide y series |
| z | Peptide z series |
| I | Immonium ion |
| m | Internal fragment (‘m’iddle) |
| \_ | External fragment ions |
| p | Precursor ion |
| r | Reporter ion, such as TMT or iTRAQ |
| f | Chemical formula |
| s | 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.

### Unknown ions

If a spectrum has been annotated and peak annotations are included in 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.

Unknown ions MAY be assigned an integer number in order to annotate relationships between 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

### 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 ions 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 Ʃ(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). is the proton mass multiplied by the charge state. In some cases, the may be modified if one or more of the charge-carrying particles is heavier than a proton.

|  |  |  |
| --- | --- | --- |
|  | Formula | Remarks |
| a | ∑(AA) – CO + (H+)z |  |
| b | ∑(AA) + (H+)z |  |
| c | ∑(AA) + NH3 + (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 loss” below. |
| x | ∑(AA) + CO2 + (H+)z |  |
| y | ∑(AA) + H2O + (H+)z |  |
| z | ∑(AA) + H2O – NH3 + (H+)z | Note: This is also known as the z.+ radical ion, which is found in ETD spectra. The first positive charge is due to the loss of an electron, creating the radical.  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 loss” below. |

### 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 n1:n2 where n1 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 n2 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 x = 1, one MUST NOT denote any b ion as m1:y. In cases where there is an N-terminal mass modification that is lost, it MUST be encoded as a neutral loss. 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-C bond (an a/x cleavage) on one end and a C-N bond (a b/y cleavage) on the other end produces an “a like” ion. 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

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 “a like ion” 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 m:4:5, although they have the same mass both might be produced.

### 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 is 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. 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, it is a little complex: the “Unimod entry name” is the “PSI-MS Name” column if it is not null, or if null, then the “Interim Name” column.

### 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

The phosphate neutral loss of a 2+ precursor ion, which is often an intense peak in CID spectra of serine phosphopeptides 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^3), a charge-reduced 2+ unfragmented precursor ion (p^2), and a charge-reduced 1+ unfragmented precursor ion (p). ETD spectra can also have charge-reduced precursors due to lost protons (not always addition of electrons), so that would be denoted as neutral losses of hydrogens, e.g. p-2H^4.

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.

### Isobaric labelling related ions

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 basic residues (though not necessarily), and the cleavage of the tag releases the reporter ion. Typical examples of this are TMT and iTRAQ reporter ions.

Reporter ions are designated by compact names. Although there are currently a relatively small number of reporter ions, this may grow slowly over time and therefore the full set of possible options is defined at a version-controlled document in GitHub that may be updated as the field advances. There is a human-readable markdown version:

<https://github.com/HUPO-PSI/SpectralLibraryFormat/blob/master/specification/IsobaricLabelIons.md>

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

<https://github.com/HUPO-PSI/SpectralLibraryFormat/blob/master/specification/IsobaricLabelIons.json>

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

Reporter ions or other isobaric tag-related ions are prefixed with the letter ‘r’ with the ion name following in square brackets in order to provide a reliable handle for software parsers such as:

r[TMT127N]

r[iTRAQ114]

Reporter ion names MUST NOT contain “[“ or “]” characters.

Each of these names MAY also be used as a neutral (or frequently a charged) loss. It is also common in TMT fragmentation spectra to see

p-[TMT6nterm]

It is also not unusual to see a series of peaks such as:

p-[iTRAQ114]-CO

p-[iTRAQ115]-CO

p-[iTRAQ116]-CO-H2O

etc.

### External fragment ions

Ions that do not come from the peptide precursor, but rather from external contaminants, can often be found in tandem mass spectra. Any ions attributed to a precursor ion not explicitly listed in the spectrum header MUST be prefixed with a 0@ as described above. If the annotation is not a chemical formula or other type of annotation described elsewhere in this specification, then it 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@\_{Adenosine}

attributes the annotated peak as coming from singly charged adenosine (0 charge would not be detected, and doubly charged should be postfixed with ^2, etc.). The string after the underscore and enclosed in curly braces should be as informative and concise as possible. It is not specifically obligatory that tools supporting this format would be able to understand a peak annotation of 0@\_{Adenosine}+HPO3^2 as the 2+ ion of adenosine monophosphate, but this is the aspirational goal. 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 external named ion, but are not obligated to understand more than that.

As another example, y1 and a2 and b2 ions are commonly very strong in HCD spectra, and one can easily assign such peaks otherwise not associated with the putatively identified analyte. For example, if the primary analyte were a tryptic peptide MYPEPTIDEK, it would not be unusual to see annotations such as:

0@\_{y1(R)}

0@\_{a2(LP)}

0@\_{b2(LP)}

which signify that 3 reasonably prominent peaks are not explained by the analyte, but yet, are explainable as common contaminant ions. Nearly every high S/N HCD spectrum has both y1(K) and y1(R) although only one matches the likely analyte.

### 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.

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

The ProForma 2.0 specification5 (<https://psidev.info/proforma>) provides further rules on how to encode molecular formulas (e.g. recommended element ordering).

### 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}/-0.55ppm

s{Coc(c1)cccc1C#N}^2/2.29ppm

Other suffixes such as for isotope and charge state, as described elsewhere in this document, MAY be used following the SMILES string. SMILES strings are able to encode the putative locations of charges, but overall charge notation as described below in this document MUST be followed, irrespective of whether it is specified within the SMILES string. For example, a doubly charged ion should be written s{…}^2 irrespective of whether charge information is present in the SMILES string.

At present, only SMILES is support, while InChi is not supported, because SMILES is more human readable.

## Neutral losses

The neutral loss (or gain) component may be 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 MUST be preceded by an integer indicating the number (e.g., -2H2O signifies a double water loss). Single losses MUST NOT have a preceding 1.

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 in the following format: CaHbNcOdSe… (followed by other elements in any order), where a, b, c, d, e are the stoichiometric number of the element that precedes it. This table is not complete, and other losses are possible.

|  |  |  |  |
| --- | --- | --- | --- |
| Neutral Loss/Gain Group | Common name | Exact (monoisotopic) mass | 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” |
| 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 |
| C2H5NOS | Mercaptoacetamide | 91.009195 | From carbamidomethyl cysteine |
| C2H4O2S | Mercaptoacetic acid | 91.993211 | From carboxymethyl cysteine |
| H3PO4 | Phosphoric acid | 97.976896 | From phosphoserine and phosphothreonine |
| [Isobaric tags]  Note: The square bracket is mandatory for non-chemical formulae. |  | (variable) | See the isobaric tags section above since these can generate losses as well. |
| [Other complex groups], e.g. [Hex] |  |  | A list of permissible complex neutral loss ions is available at <https://github.com/HUPO-PSI/mzSpecLib/tree/master/specification> |

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

y2+CO-H2O

y2-H2O-NH3

y2-[Hex]

p-[iTRAQ115]

p-[iTRAQ116]-H2O

etc.

If there are multiple neutral gains/losses, alphabetical 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.

## 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 ~1.003 Da greater than that of the parent fragment ion, where this delta depends a bit on the elemental composition and relative isotope ratios.

## 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. 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+. I.e. in NIST MSP, 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.

## Adduct Type

The adduct type component is optional. Typical conditions for peptide ionization for proteomics generates protonated ions [M+H], [M+2H], etc, which are implied if the adduct type component is omitted. 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 and A is the atom/molecule added to the precursor 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+n1A+n2B...], 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 should be preceded 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 preceded by ^3 to specify the charge state, see above 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^2[M+H+Na]

IMPORTANT NOTE: In this context, the M represents the annotated neutral fragment, NOT the precursor. In the above example, M represents a neutral y5-H2O.

## Multiple peaks belonging to 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 will be assigned the same peak annotation string with a bracket [ ] around it, with the m/z deviation changed accordingly. For example:

677.298 69 [y7/-0.002]

677.299 572 [y7/-0.001]

677.300 5681 y7/0.000

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 delineate a peak 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 be responsible for the annotated annotation. However, the present format does not mandate such kind of peak annotations.

## 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 as a spectrum attribute.

If multiple annotations are offered, the most likely 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 SHOULD sum to a number equal to or less than 1.0. The difference between the sum and 1.0 is presumed to be the probability 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.

# 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.

## Definition

analyte\_reference:

type: integer|string

description: Label of analyte to which this annotation belongs.

required: true

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:

description: Error between observed and theoretical mass

value:

type: number

unit:

type: string

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

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

series\_label: internal

precursor:

series\_label: precursor

immonium:

amino\_acid: The amino acid represented by this immonium ion

modification: An 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

external:

label: The name of the external ion being marked

series\_label: external

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

## 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"

],

"rest": null

}

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,

"rest": 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,

"rest": null,

"molecule\_description": {

"series\_label": "precursor"

}

}

# Regular Expressions

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

**ECMAScript Regex**

^(?:(?<analyte\_reference>[^@\s]+)@)?(?:(?:(?<series>[axbycz]\.?)(?<ordinal>\d+))|(?<series\_internal>[m](?<internal\_start>\d+):(?<internal\_end>\d+))|(?<precursor>p)|(:?I(?<immonium>[ARNDCEQGHKMFPSTWYVIL])(?:\[(?<immonium\_modification>(?:[^\]]+))\])?)|(?<reporter>r(?:(?:\[(?<reporter\_label>[^\]]+)\])))|(?:f\{(?<formula>[A-Za-z0-9]+)\})|(?:\_(?<external\_ion>(?:[^"\s,/]+)|(?:"(?:[^"]+)")))|(?<unknown>\?))(?<neutral\_losses>(?:[+-]\d\*(?:(?:[A-Z][A-Za-z0-9]\*)|(?:\[(?:(?:[A-Za-z0-9:\.]+))\])))+)?(?:(?<isotope>[+-]\d\*)i)?(?:\^(?<charge>[+-]?\d+))?(?:\[(?<adducts>M(:?[+-]\d\*[A-Z][A-Za-z0-9]\*)+)\])?(?:/(?<mass\_error>[+-]?\d+(?:\.\d+)?)(?<mass\_error\_unit>ppm)?)?(?:\\*(?<confidence>\d\*(?:\.\d+)?))?

**Python SRE in verbose mode**

^(?:(?P<analyte\_reference>[^@\s]+)@)?

(?:(?:(?P<series>[axbycz]\.?)(?P<ordinal>\d+))|

(?P<series\_internal>[m](?P<internal\_start>\d+):(?P<internal\_end>\d+))|

(?P<precursor>p)|

(:?I(?P<immonium>[ARNDCEQGHKMFPSTWYVIL])(?:\[(?P<immonium\_modification>(?:[^\]]+))\])?)|

(?P<reporter>r(?:

(?:\[

(?P<reporter\_label>[^\]]+)

\])

))|

(?:f\{(?P<formula>[A-Za-z0-9]+)\})|

(?:\_(?P<external\_ion>(?:[^"\s,/]+)|(?:"(?:[^"]+)")))|

(?P<unknown>\?)

)

(?P<neutral\_losses>(?:[+-]\d\*

(?:(?:[A-Z][A-Za-z0-9]\*)|

(?:\[

(?:

(?:[A-Za-z0-9:\.]+)

)

\])

)

)+)?

(?:(?P<isotope>[+-]\d\*)i)?

(?:\^(?P<charge>[+-]?\d+))?

(?:\[(?P<adducts>M(:?[+-]\d\*[A-Z][A-Za-z0-9]\*)+)\])?

(?:/(?P<mass\_error>[+-]?\d+(?:\.\d+)?)(?P<mass\_error\_unit>ppm)?)?

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

## 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/mzSpecLib/blob/master/specification/peak-annotation-format/grammar.md>

Grammar:

<https://github.com/HUPO-PSI/mzSpecLib/blob/master/specification/peak-annotation-format/annotation.lark>

# 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.

## 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, 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. The chosen format for peak annotation should then be specified in the library header (for the whole library) or in the spectrum header (for that spectrum only, overriding the library-level specification if any).

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.

# 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:

is\_auxiliary = **False**

**if** text[i] == '[':

is\_auxiliary = **True**

i += 1

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** is\_auxiliary:

**if** text[i\_end] == ']':

i\_end += 1

annot.is\_auxiliary = **True**

**else**:

**raise** **Exception**(f"Unclosed auxiliary annotation " +

f"starting at **{**i - 1**}**")

**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

# Appendix B. Isobaric Label Ions

The most up-to-date list of isobaric ions can be found here:

<https://github.com/HUPO-PSI/SpectralLibraryFormat/blob/master/specification/IsobaricLabelIons.md>

And a JSON serialization with m/z and other information:

<https://github.com/HUPO-PSI/SpectralLibraryFormat/blob/master/specification/IsobaricLabelIons.json>

For the convenience of reading this specification, the state of this file as of version 1.0 of the specification is provided below:

TMT peaks

TMT126

TMT127N

TMT127C

TMT128N

TMT128C

TMT129N

TMT129C

TMT130N

TMT130C

TMT131N

TMT131C

TMT0Nterm

TMT2Nterm

TMT6Nterm

TMTproNterm

TMT132N

TMT132C

TMT133N

TMT133C

TMT134N

iTRAQ peaks

iTRAQ113

iTRAQ114

iTRAQ115

iTRAQ116

iTRAQ117

iTRAQ118

iTRAQ119

iTRAQ121

iTRAQ4Nterm

iTRAQ8Nterm

The "Nterm" elements refer to the entire N terminal label, including both the reporter and balance group. These commonly fall off together producing a peak at the full label mass plus a proton.

# Appendix C. Complex Neutral Loss Groups

The most up-to-date list of neutral loss groups can be found here:

<https://github.com/HUPO-PSI/SpectralLibraryFormat/blob/master/specification/NeutralLossGroups.md>

And a JSON serialization with m/z and other information:

<https://github.com/HUPO-PSI/SpectralLibraryFormat/blob/master/specification/NeutralLossGroups.json>

For the convenience of reading this specification, the state of this file as of version 1.0 of the specification is provided below:

In mzSpecLib library peak annotations, complex neutral loss groups that are not conveniently described by chemical formulae can be denoted using the following tags enclosed in square brackets, e.g. y6-[Hex] for a hexose loss from the y6 primary ion.

If the chemical formula is long enough that it can conceivably map to many possible functional groups, then we use this complex group mechanism to be more specific. For example, there is no point of making a [Water] tag because H2O is already clear and unambiguous.

Currently permitted values are:

# Oligosaccharides

* Hex
* HexNAc
* dHex
* NeuAc
* NeuGc

## 

# Authors 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](mailto:juan@ebi.ac.uk)

Eric W. Deutsch

Institute for Systems Biology, Seattle WA, USA

[edeutsch@systemsbiology.org](mailto:edeutsch@systemsbiology.org)

# 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:

Andrew R. Jones

Institute of Systems, Molecular and Integrative Biology, University of Liverpool, Liverpool, L69 3BX, United Kingdom

Pierre-Alain Binz

Lausanne University Hospital, Lausanne, Switzerland

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

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

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

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

Zhi Sun

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

Nils Hoffmann

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

Jim Shofstahl

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

Yunping Zhu

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

Helge Hecht

Masaryk University, Kotlářská 2, Brno, Czech Republic

# 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).

# Copyright Notice

Copyright (C) Proteomics Standards Initiative (2023). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the PSI or other organizations, except as needed for the purpose of developing Proteomics Recommendations in which case the procedures for copyrights defined in the PSI Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the PSI or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE PROTEOMICS STANDARDS INITIATIVE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

# Glossary

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

# 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. 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. 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