**mzTab: exchange format for proteomics and metabolomics results**

Status of This Document

This document presents a draft specification for the mzTab data format developed by members of the Human Proteome Organisation (HUPO) Proteomics Standards Initiative (PSI) Proteomics Informatics (PI) Working Group. Distribution is unlimited.

Version of This Document

The current version of this document is: version 1.0, release candidate 4, 14th April 2013.

# 

# Abstract

The Human Proteome Organisation (HUPO) Proteomics Standards Initiative (PSI) defines community standards for data representation in proteomics to facilitate data comparison, exchange and verification. The Proteomics Informatics Working Group is developing standards for describing the results of identification and quantification processes for proteins, peptides and protein modifications from mass spectrometry. This document defines a tab delimited text file format to report proteomics and metabolomics results.

**Contents**

[Abstract 1](#_Toc356304574)

[1. Introduction 2](#_Toc356304575)

[1.1 Background 2](#_Toc356304576)

[1.2 Document Structure 3](#_Toc356304577)

[2. Use Cases for mzTab 3](#_Toc356304578)

[3. Notational Conventions 4](#_Toc356304579)

[4. Relationship to Other Specifications 5](#_Toc356304580)

[4.1 The PSI Mass Spectrometry Controlled Vocabulary (CV) 5](#_Toc356304581)

[5. Resolved Design and scope issues 6](#_Toc356304582)

[5.1 Handling updates to the controlled vocabulary 6](#_Toc356304583)

[5.2 Use of identifiers for input spectra to a search 6](#_Toc356304584)

[5.3 Recommendations for reporting protein inference 8](#_Toc356304585)

[5.4 Recommendations for reporting replicates within experimental designs 8](#_Toc356304587)

[Recommendations for reporting quantification results 9](#_Toc356304603)

[5.5 9](#_Toc356304605)

[5.6 Encoding missing values, zeroes, nulls, infinity and calculation errors 10](#_Toc356304637)

[5.7 Number of peptides reported 10](#_Toc356304638)

[5.8 Reliability score 11](#_Toc356304639)

[5.9 Reporting modifications and amino acid substitutions 11](#_Toc356304640)

[5.10 Comments on Specific Use Cases 12](#_Toc356304641)

[5.11 Other supporting materials 14](#_Toc356304642)

[6. Format specification 14](#_Toc356304643)

[6.1 Sections 16](#_Toc356304644)

[6.2 Metadata Section 16](#_Toc356304645)

[6.3 Protein Section 23](#_Toc356304646)

[6.4 Peptide Section 28](#_Toc356304647)

[6.5 PSM Section 32](#_Toc356304658)

[6.6 Small Molecule Section 35](#_Toc356304659)

[7. Non-supported use cases 39](#_Toc356304660)

[8. Conclusions 39](#_Toc356304661)

[9. Authors 40](#_Toc356304662)

[10. Contributors 40](#_Toc356304663)

[11. References 40](#_Toc356304664)

[12. Intellectual Property Statement 41](#_Toc356304665)

[TradeMark Section 41](#_Toc356304666)

[Copyright Notice 41](#_Toc356304667)

# Introduction

## Background

This document addresses the systematic description of peptide, protein, and small molecule identification and quantification data retrieved from mass spectrometry (MS)-based experiments. A large number of software tools are available that analyze MS data and produce a variety of different output data formats. The HUPO Proteomics Standards Initiative (PSI) has developed several vendor-neutral data formats to overcome this heterogeneity of data formats for MS data. Currently, the PSI promotes the usage of three file formats to report an experiment’s data: mzML to store the pure MS data (i.e. the spectra and chromatograms), mzIdentML to store (poly)peptide identifications and potentially inferred protein identifications, and mzQuantML to store quantitative data associated with these results. All three of these formats are XML-based and require sophisticated software to access the stored data.

While full, detailed representation of MS data including provenance is essential for researchers in the field, many downstream analysis use cases are only concerned with the *results* of the experiment in an easily accessible format. In addition, there is a trend for performing more integrated experimental workflows involving both proteomics and metabolomics data. Thus, the current lack of standardization in the field of metabolomics was taken into account in the development of the format presented here, and structures were developed that can report protein, peptide, and small molecule MS based data.

mzTab is intended as a lightweight supplement to the already existing standard file formats, providing a summary, similar to the supplementary table of results of a scientific publication. mzTab files can contain protein, peptide, and small molecule identifications together with basic quantitative information. mzTab is not intended to store an experiment’s complete data / evidence but only its final reported results. This format is also intended to provide local LIMS systems as well as MS proteomics repositories a simple way to share and combine basic information.

mzTab has been developed with a view to support the following general tasks (more specific use cases are provided in Section ):

1. *Facilitate the sharing of final experimental results,* especially with researchers outside the field of proteomics that i) lack specialized software to parse the existing PSI’s XML-based standard file formats, and ii) are only interested in the final reported results and not in all the details related to the data processing due to the inherent complexity of MS proteomics data. Furthermore, this should encourage the development of small innovative tools without the requirement of parsing huge XML files, which might be outside the scope of many bioinformaticians.
2. *Export of results to external software,* that is not able to parse proteomics/metabolomics specific data formats but can handle simple tab-delimited file formats. As a guideline the file format is designed to be viewable by programs such as Microsoft Excel® and Open Office Spreadsheet.
3. *Contain the results of an experiment in a single file*, and thus not require linking two files to retrieve identification and quantification results to again simplify the processing of the data.
4. *Allow the concatenation of results,* and thus be able to combine results from multiple experiments but also multiple entries from local LIMS databases or MS proteomics repositories.
5. *Act as an output format of (web-) services* that report MS-based results and thus can produce standardized result pages.
6. *Allow the combination of MS-based proteomics and metabolomics experimental results* within a single file.
7. *Be able to link to the external experimental evidence* (i.e. the mass spectra in different formats), following the same approach used in mzIdentML and mzQuantML.

This document presents a specification, not a tutorial. As such, the presentation of technical details is deliberately direct. The role of the text is to describe the model and justify design decisions made. The document does not discuss how the models should be used in practice, consider tool support for data capture or storage, or provide comprehensive examples of the models in use. It is anticipated that tutorial material will be developed independently of this specification.

## Document Structure

The remainder of this document is structured as follows. Section lists use cases mzTab is designed to support. Section 3 describes the terminology used. Section describes how the specification presented in Section 6 relates to other specifications, both those that it extends and those that it is intended to complement. Section discusses the reasoning behind several design decisions taken. Section 6 contains the documentation of the file. Conclusions are presented in Section 7.

# Use Cases for mzTab

The following cases of usage have driven the development of the mzTab data model, and are used to define the scope of the format in version 1.0.

1. mzTab files should be simple enough to make proteomics/metabolomics results accessible to people outside the respective fields. This should facilitate the sharing of data beyond the borders of the fields and make it accessible to non-experts.
2. mzTab files should contain sufficient information to provide an electronic summary of all findings in a proteomics/metabolomics study to permit its use as a standard documentation format for ‘supplementary material’ sections of publications in proteomics and metabolomics. It should thus be able to replace PDF tables as a way of reporting peptides and proteins and make published identification and quantification information more accessible.
3. It should be possible to open mzTab files with “standard” software such as Microsoft Excel® or Open Office Spreadsheet. This should furthermore improve the usability of the format to people outside the fields of proteomics/metabolomics.
4. It should be possible to export proteomics data from, for example, mzIdentML/ mzQuantML files into mzTab to then load this data into, for example, statistical tools such as those provided through the R programming language. With the current formats, complex conversion software would be needed to make proteomics results available to such environments.
5. mzTab files should make MS derived results easily accessible to scripting languages allowing bioinformaticians to develop software without the overhead of developing sophisticated parsing code. Since mzTab files will be comparatively small, the data from multiple experiments can be processed at once without requiring special resource management techniques.
6. It should be possible to contain the complete final results of an MS-based proteomics/metabolomics experiment in a single file. This should furthermore reduce the complexity of sharing and processing an experiment’s final results. mzTab files should be able to store quantitative values for protein, peptide, and small molecule identifications. Furthermore, mzTab files should contain basic protein inference information and modification position ambiguity information. Additionally, mzTab files should be able to report merged results from multiple search engines.
7. It should be possible to merge results from multiple experiments / resources by simply concatenating the respective sections of an mzTab file. Thus, every record in an mzTab file should be self-contained. However, it must be highlighted that quantitative results cannot be directly compared between different experiments.
8. It should be useful as an output format by web-services that can then be readily accessed by tools supporting mzTab. Through simple concatenation the results from multiple tools can be aggregated and processed at once.
9. As mzTab files only contain an experiment’s core results, all entries should link back to their source. Furthermore, it should be possible to directly link a given peptide / small molecule identification to its source spectrum in an external MS data file. The same referencing system as in mzIdentML/mzQuantML should be used.

# 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 ([Bradner 1997](#_ENREF_1" \o "Bradner, 1997 #5)).

# Relationship to Other Specifications

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

1. *mzML* (<http://www.psidev.info/mzml>). mzML is the PSI standard for capturing mass spectra / peak lists resulting from mass spectrometry in proteomics (Martens, L., *et al.* 2011). mzTab files MAY be used in conjunction with mzML, although it will be possible to use mzTab with other formats of mass spectra. This document does not assume familiarity with mzML.
2. *mzIdentML* (<http://www.psidev.info/mzidentml>). mzIdentML is the PSI standard for capturing of peptide and protein identification data (Jones, A. R., *et al.* 2012). mzTab files MAY reference mzIdentML files that then contain the detailed evidence of the reported identifications.
3. *mzQuantML* (<http://www.psidev.info/mzquantml>). mzQuantML is the PSI standard for capturing quantitative proteomics data from mass spectrometry (Walzer, M. *et al.* 2013). mzTab files that report quantitative data MAY reference mzQuantML files for detailed evidence of the reported values.

## The PSI Mass Spectrometry Controlled Vocabulary (CV)

The PSI-MS controlled vocabulary is intended to provide terms for annotation of mzML, mzIdentML, and mzQuantML files. The CV has been generated with a collection of terms from software vendors and academic groups working in the area of mass spectrometry and proteome informatics. Some terms describe attributes that must be coupled with a numerical value attribute in the CvParam element (e.g. MS:1001191 “p-value”) and optionally a unit for that value (e.g. MS:1001117, “theoretical mass”, units = dalton). The terms that require a value are denoted by having a “datatype” key-value pair in the CV itself: MS:1001172 "mascot:expectation value" value-type:xsd:double. Terms that need to be qualified with units are denoted with a “has\_units” key in the CV itself (relationship: has\_units: UO:0000221 ! dalton).

As recommended by the PSI CV guidelines, psi-ms.obo should be dynamically maintained via the [psidev-ms-vocab@lists.sourceforge.net](mailto:psidev-ms-vocab@lists.sourceforge.net) mailing list that allows any user to request new terms in agreement with the community involved. Once a consensus is reached among the community the new terms are added within a few business days. If there is no obvious consensus, the CV coordinators committee should vote and make a decision. A new psi-ms.obo should then be released by updating the file on the CVS server without changing the name of the file (this would alter the propagation of the file to the OBO website and to other ontology services that rely on file stable URI). For this reason an internal version number with two decimals (x.y.z) should be increased:

* x should be increased when a first level term is renamed, added, deleted or rearranged in the structure. Such rearrangement will be rare and is very likely to have repercussion on the mapping.
* y should be increased when any other term except the first level one is altered.
* z should be increased when there is no term addition or deletion but just editing on the definitions or other minor changes.

The following ontologies or controlled vocabularies specified below may also be suitable or required in certain instances:

* Unit Ontology (<http://www.obofoundry.org/cgi-bin/detail.cgi?id=unit>)
* ChEBI (<http://www.ebi.ac.uk/chebi/>)
* OBI (Ontology of Biological Investigations - <http://obi.sourceforge.net/>)
* PSI Protein modifications workgroup - <http://psidev.cvs.sourceforge.net/psidev/psi/mod/data/PSI-MOD.obo>
* Unimod modifications database - <http://www.unimod.org/obo/unimod.obo>
* PRIDE Controlled Vocabulary (<http://ebi-pride.googlecode.com/svn/trunk/pride-core/schema/pride_cv.obo>)
* NEWT UniProt Taxonomy Database (<http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=NEWT>)
* BRENDA tissue/ enzyme source (<http://www.brenda-enzymes.info/ontology/tissue/tree/update/update_files/BrendaTissueOBO>).
* Cell Type ontology (http://obo.cvs.sourceforge.net/obo/obo/ontology/anatomy/cell\_type/cell.obo).

# Resolved Design and scope issues

There were several issues regarding the design of the format that were not clear cut, and a design choice was made that was not completely agreeable to everyone. So that these issues do not keep coming up, we document the issues here and why the decision that is implemented was made.

## Handling updates to the controlled vocabulary

There is a difficult issue with respect to how software should encode CV terms, such that changes to the core can be accommodated. This issue is discussed at length in the mzML specification document (Martens, L *et al.* 2011), and mzTab follows the same convention. In brief, when a new term is required, the file producers must contact the CV working group (via the mailing list [psidev-ms-vocab@lists.sourceforge.net](mailto:psidev-ms-vocab@lists.sourceforge.net)) and request the new term. It is anticipated that problems may arise if a consumer of the file encounters a new CV term and they are not working from the latest version of the CV file. It has been decided that rather than aim for a workaround to this issue, it can be expected that data file consumers must ensure that the OBO file is up-to-date.

## Use of identifiers for input spectra to a search

Peptides and small molecules MAY be linked to the source spectrum (in an external file) from which the identifications are made by way of a reference in the spectra\_ref attribute and via the ms\_file element which stores the URL of the file in the location attribute. If assays are reported or if a PSM section is present, the ms\_file[1\_n]-location MUST be present, since back references to these files will be provided.

It is advantageous if there is a consistent system for identifying spectra in different file formats. The following table is implemented in the PSI-MS CV for providing consistent identifiers for different spectrum file formats. This is the exact same approach followed in mzIdentML and mzQuantML. *Note, this table shows examples from the CV but will be extended. The CV holds the definite specification for legal encodings of spectrumID values.*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Term** | **Data type** | **Comment** |
| MS:1000768 | Thermo nativeID format | controllerType=xsd:nonNegativeInteger controllerNumber=xsd:positiveInteger scan=xsd:positiveInteger. | controller=0 is usually the mass spectrometer |
| MS:1000769 | Waters nativeID format | function=xsd:positiveInteger process=xsd:nonNegativeInteger scan=xsd:nonNegativeInteger |  |
| MS:1000770 | WIFF nativeID format | sample=xsd:nonNegativeInteger period=xsd:nonNegativeInteger cycle=xsd:nonNegativeInteger experiment=xsd:nonNegativeInteger |  |
| MS:1000771 | Bruker/Agilent YEP nativeID format | scan=xsd:nonNegativeInteger |  |
| MS:1000772 | Bruker BAF nativeID format | scan=xsd:nonNegativeInteger |  |
| MS:1000773 | Bruker FID nativeID format | file=xsd:IDREF | The nativeID must be the same as the source file ID |
| MS:1000774 | multiple peak list nativeID format | index=xsd:nonNegativeInteger | Used for conversion of peak list files with multiple spectra, i.e. MGF, PKL, merged DTA files. Index is the spectrum number in the file, starting from 0. |
| MS:1000775 | single peak list nativeID format | file=xsd:IDREF | The nativeID must be the same as the source file ID. Used for conversion of peak list files with one spectrum per file, typically in a folder of PKL or DTAs, where each sourceFileRef is different |
| MS:1000776 | scan number only nativeID format | scan=xsd:nonNegativeInteger | Used for conversion from mzXML, or a DTA folder where native scan numbers can be derived. |
| MS:1000777 | spectrum identifier nativeID format | spectrum=xsd:nonNegativeInteger | Used for conversion from mzData. The spectrum id attribute is referenced. |

Table 1 Controlled vocabulary terms and rules implemented in the PSI-MS CV for formulating the “nativeID” to identify spectra in different file formats.

In mzTab, the spectra\_ref attribute should be constructed following the data type specification in . As an example, to reference the third spectrum (index = 2) in an MGF (Mascot Generic Format) file:

MTD ms\_file[1]-format [MS, MS:1001062, Mascot MGF file, ]

MTD ms\_file[1]-id\_format [MS, MS:1000774, multiple peak list nativeID format, ]

...

PSH sequence ... spectra\_ref ...

PSM NILNELFQR ... ms\_file[1]:index=2 ...

## Recommendations for reporting protein inference

There are multiple approaches to how protein inference can be reported. mzTab is designed to only hold experimental results, which in proteomics experiments can be very complex. At the same time, for downstream statistical analysis there is a need to simplify this problem. It is not possible to model detailed protein inference data without a significant level of complexity at the file format level. Therefore, it was decided to “mention” the protein inference problem in mzTab files but not provide detailed information on how it was resolved. Protein entries in mzTab files contain the field ambiguity\_members. The protein accessions listed in this field should identify proteins that were also be identified through the same-set of peptides or spectra, or proteins supported by a largely overlapping set of evidence, and could also be a viable candidate for the “true” identification of the entity reported. It is RECOMMENDED that “subset proteins” that are unlikely to have been identified SHOULD NOT be reported here. The mapping of a single peptide-spectrum match (PSM) to multiple accessions is supported through the reporting of the same PSM on multiple rows of the PSM section, as exemplified below.

COM In the following example only one peptide was identified that can be attributed to

COM multiple proteins. The choice which one to pick as primary accession depends on the

COM resource generating the mzTab file.

...

PRH accession … ambiguity\_members …

PRT P19012 P13646, P08779, P02533, Q7Z3Z0, Q7Z3Y9, Q7Z3Y8 …

...

PEH sequence accession …

PEP ALEEANADLEVK P19012 …

## Recommendations for reporting replicates within experimental designs

Modeling the correct reporting of technical/biological replicates within experimental designs is supported in mzTab using an adaptation of the system originally developed for mzQuantML comprising three components:

* Sample – a biological material that has been analysed, to which descriptors of species, cell/tissue type etc can be attached.
  + In mzTab, these MAY be reported in the metadata section as sample[1-n].
* Assay – The application of a measurement about the sample (in this case through MS) – producing values about small molecules, peptides or proteins. An assay is typically mapped to one MS run (in the case of label-free MS analysis) or one label/tag within an MS run for multiplexed techniques.
  + In mzTab, these MAY be reported in the metadata section as assay[1-n].
  + Quantitative values MAY be reported as columns: {[protein|peptide|smallmolecule]\_abundance}\_assay[1-n].
  + If sample information is provided in the file, assays[1-n] MUST refer to the sample that has been analysed.
* Study variable – A grouping of assays to capture replication in the experimental design.
  + In mzTab, these MAY be reported in the metadata section as study\_variable[1-n].
  + If assays have been reported in the same file, study variables MUST have references to the assays they are grouping.
  + Quantitative values MAY be reported as columns: {[protein|peptide|smallmolecule]\_abundance}\_study\_variable[1-n]
  + If quantitative values have been provided for study variables, standard deviation and standard error columns SHOULD be present.

Clear definitions of biological and technical replicates are difficult to provide as these are somewhat dependent upon the biological domain. However, we use the following general definitions in mzTab.

* Biological replicates are where different samples have been analysed by MS
* Technical replicates are where same samples are analysed multiple times by (LC)-MS.

*Note: there is deliberately no attempt to define the boundary of the term “sample”.*

If sample level information is provided optimally, it is expected that *n* biological replicates SHOULD be mapped to sample[1-n] (and assay[1-n] exist for measurements of these samples); *n* technical replicates SHOULD be mapped to assay[1-n] referencing sample[1] (for example). However, an open challenge remains since analysis software is often not aware of whether replicates (multiple MS runs) are originally biological or technical in nature. As such, the default behavior for mzTab exporters from quantitative software is to exclude sample level information and report quantitative data for assay[1-n] and/or study\_variable[1-n]. Additional annotation software would typically be required to add the sample-level information, as provided (often manually) by the research group.

## Recommendations for reporting quantification results

Quantitative technologies generally result in some kind of abundance measurement of the identified analyte. Several of the available techniques, furthermore, allow/require multiple similar samples to be multiplexed and analyzed in a single MS run – for example in label-based techniques, such as SILAC/N15 where quantification occurs on MS1 data or in tag-based techniques, such as iTRAQ/TMT where quantification occurs in MS2 data.

One measurement of a small molecule, peptide or protein is mapped to the concept of assay for both multiplexed techniques and label-free techniques. Each assay MUST have a reference to the quantification reagent/label used (“unlabelled” in the label-free case and the “light” channel in SILAC/N15) and each assay MUST have a reference to the ms\_file[1\_n] from which it originated. As such, in multiplexed techniques where *n* reagents are used within one analysis, assay[1-n] MUST reference the same ms\_file.

If the data exporter wishes to report only “final results” (i.e. following averaging over replicates), then these MUST be reported as quantitative values attached to study\_variable[1-n]. mzTab allows the reporting of abundance, standard deviation, and standard error for any study\_variable. The unit of these values MUST be specified in the metadata section of the mzTab file. The reported values SHOULD represent the final result of the performed data analysis. The exact meaning of the values will thus depend on the used analysis pipeline and quantitation method and is not expected to be comparable across multiple mzTab files.

[Need to insert examples from label-free, MS1 label, MS2 tag, spectral counting and SRM here]

[Do we want to add the explicit concept of a Ratio, as used in mzQuantML, as one Assay/one Assay or one SV/SV – this is useful, but there is a workaround to normalize individual values to one, as used before]

## Encoding missing values, zeroes, nulls, infinity and calculation errors

In the table-based sections (protein, peptide, and small molecule) there MUST NOT be any empty cells. In case a given property is not available “null” MUST be used. This is, for example, the case when modifications were not identified on a given peptide (*i.e.* the table cell MUST NOT be empty but “null” has to be reported). If ratios are included and the denominator is zero, the “INF” value MUST be used. If the result leads to calculation errors (for example 0/0), this MUST be reported as “not a number” (“NaN”). In some cases, there is ambiguity with respect to these cases: e.g. in spectral counting if no peptide spectrum matches are observed for a given protein, it is open for debate as to whether its abundance is zero or missing (“null”).

## Number of peptides reported

There are OPTIONAL columns allowed in the protein section to report the number of peptides supporting a given protein identification/quantification. Exporters MAY include these values for none, some or all of the ms\_files reported.

* num\_psms\_ms\_file[1\_n]
  + The count of the total significant PSMs that can be mapped to the reported protein
* num\_peptides\_distinct\_ms\_file[1\_n]
  + The count of the number of different peptide sequences that have been identified above the significance threshold. Different modifications or charge states of the same peptide are not counted.
* num\_peptides\_unique\_ms\_file[1\_n]
  + The number of peptides that can be mapped uniquely to the protein reported. If ambiguity members have been reported, the count MUST be derived from the number of peptides that can be uniquely mapped to the group of accessions, since the assumption is that these accessions are supported by the same evidence.

The idea of these three columns is to give the researcher a quick overview of how well a given protein identification is supported by peptide identifications for a given ms\_file reported. These values are not mandatory since it is recognized that many quantitative approaches do not assemble all identification-level evidence in this way. The num\_psms column also provides the opportunity for reporting pseudo-quantitative (label-free) values from approaches in which no explicit quantification has been performed and such it is RECOMMENDED that this column is provided for identification results exported to mzQuantML.

## Reliability score

All protein, peptide and small molecule identifications reported in an mzTab file SHOULD be assigned a reliability score (column “reliability” in all tables). This reliability only applies to the identification reliability but not to modification and or quantification reliabilities. The idea is to provide a way for researcher and/or MS proteomics or metabolomics repositories to score the reported identifications based on their own criteria. This score is completely resource-dependent and MUST NOT be interpreted as a comparable score between mzTab files generated from different resources. The criteria used to generate this score SHOULD be documented by the data providers. If this information is not provided by the producers of mzTab files, “null“ MUST be provided as the value for each of the protein, peptide or small molecule identification.

The reliability value, if provided, MUST be an integer between 1-3 and SHOULD be interpreted as follows:

1: high reliability

2: medium reliability

3: poor reliability

The idea behind this score was to mimic the general concept of “resource based trust”. For example, if one resource reports identifications with a given reliability this would be interpreted differently as an identification reported from another resource - depending on who is responsible for the given resource and how it is build. If resources now report their reliabilities using this metric and document how this metric is generated, a user can base his own interpretation of the results based on his trust in the resource. Furthermore, approaches to make various search engine scores comparable have failed so far. To prevent the notion that the reported scores represent comparable probabilities this very abstract metric was chosen. Resources MUST explicitly specify how these reliability scores are calculated and what metric they represent.

## Reporting modifications and amino acid substitutions

Modifications or substitutions are modelled using a specific modification object with the following format:

{position}{Parameter}-{Modification or Substitution identifier}|{neutral loss}

The number of modification (or substitution) objects MUST correspond to the number of identified modifications (or substitutions) in the protein or peptide.

**{position}** is optional depending on the section where the modification is reported. Terminal modifications in proteins and peptides MUST be reported with the position set to 0 (N-terminal) or the amino acid length +1 (C-terminal) respectively. N-terminal modifications that are specifically on one amino acid MUST still be reported at the position 0. This object allows modifications to be assigned to ambiguous locations. Ambiguous positions can be reported by separating the {position} and (optional) {cvParam} by an ‘|’ from the next position. Thereby, it is possible to report reliabilities / scores / probabilities etc. for every potential location.

Here only the modification field is given:

3-MOD:00412, 8-MOD:00412 TESTPEPTIDES with two known phosphorylation sites

3|4-MOD:00412, 8-MOD:00412 First phosphorylation site can be either on S or T

3|4|8-MOD:00412, 3|4|8-MOD:00412 Three possible positions for two phosphorylation sites

**{Parameter}** is optional. It MAY be used to report a numerical value e.g. a probability score associated with the modification or location.

Reporting the first two possible sites for the phosphorylation with given probability score

Here only the modification field is given:

3[MS,MS:1001876, modification probability, 0.8]|4[MS,MS:1001876, modification probability, 0.2] MOD:00412, 8-MOD:00412

This option is not allowed though:

(3|4)[MS,MS:1001876, modification probability, 0.8]|7[MS,MS:1001876, modification probability, 0.2]-MOD:00412

**{Modification or Substitution identifier}** for proteins and peptides modifications SHOULD be reported using either UNIMOD or PSI-MOD accessions. As these two ontologies are not applicable to small molecules, so-called CHEMMODs can also be defined. Two types of CHEMMODs are allowed: specifying a chemical formula or specifying a given *m/z* delta. Additionally, it is possible to report substitutions of amino acids using SUBST:{amino acid}. In these cases, the “sequence” column MUST contain the original, unaltered sequence. The list of allowed {Modification or Substitution identifier}s therefore is:

CHEMMOD:+NH4

CHEMMOD:-18.0913

UNIMOD:18

MOD:00815

SUBST:{amino acid}

CHEMMODs SHOULD NOT be used for protein/peptide modifications if the respective entry is present in either the PSI-MOD or the UNIMOD ontology. Furthermore, mass deltas SHOULD NOT be reported if the given delta can be expressed through a known and unambiguous chemical formula.

All (identified) variable modifications as well as fixed modifications MUST be reported for every identification.

**{neutral loss}** is optional. Neutral losses are reported as cvParams. Neutral losses MAY be associated with certain modifications. In this case the neutral loss is reported after the modification object separated by the ‘|’ character. Otherwise, they are reported in the same way that modification objects are (as separate, comma-separated objects in the modification column).

PEH sequence … modifications …

COM Phosphorylation with a neutral loss:

PEP EISILACEIR … 3-UNIMOD:21|[MS, MS:1001524, fragment neutral loss, 63.998285],7-UNIMOD:4 …

COM Neutral loss without an associated modification:

PEP EISILACEIR … [MS, MS:1001524, fragment neutral loss, 63.998285],7-UNIMOD:4 …

## Comments on Specific Use Cases

Many special use cases for mzTab were considered during its development. Each of these use cases has a corresponding example file that exercises the relevant part of the format and provides a reference implementation example (see supporting documentation). Authors of software that create mzTab are encouraged to examine the examples that accompany this format release before implementing the writer.

### Multiple database search engines

Proteomics groups now commonly analyze MS data using multiple search engines and combine results to improve the number of peptide and protein identifications that can be made. The output of such approaches can be represented in mzTab as follows: mzTab files SHOULD only contain the “final” protein list generated by any such workflow. Any protein, peptide, and small molecule can be associated with any number of search engines as well as multiple search engine scores. Thus, it is possible to report which element was identified by which search engine together with the resulting scores.

### Merging mzTab files

A simple merge of different mzTab files (for example arising from different technical replicates) is not explicitly supported without specialised software (or careful manual alignment of results into valid columns), since the Protein, Peptide and SmallMolecule sections are intended for reporting results that have been “aligned” across replicates. The alignment process ensures that the data reported in each column refers to the same identified entity i.e. in the Protein section – the protein accession and the ambiguity members; in the Peptide section – a Peptide “Feature” of a given m/z, charge and modification state, and in the SmallMolecule section – a molecule with a given m/z, charge and molecular formula. For the PSM section, merging is relatively straightforward, so long as the spectra\_ref attributes contain the correct identifier of the ms\_file analysed and reported in the metadata section.

For identification-only results, it is expected that search engine results for each replicate may initially be reported in separate mzTab files, but software will be made available to assist in the merge/alignment of replicates into a single mzTab file for example suitable for database submission or for spectral count analysis.

### Adding optional columns

Additional columns MAY be added to the end of rows in all the table-based sections (protein, peptide, PSM and small molecule). These column headers MUST start with the prefix “opt\_” followed by the identifier of the object they reference: assay, study variable, MS file or “global” (if the value relates to all replicates). Column names MUST only contain the following characters: ‘A’-‘Z’, ‘a’-‘z’, ‘0’-‘9’, ‘\_’, ‘-’, ‘[’, ‘]’, and ‘:’. CV parameter accessions MAY be used for optional columns following the format: opt\_{OBJECT\_ID}\_cv\_{accession}\_{parameter name}. Spaces within the parameter’s name MUST be replaced by ‘\_’.

The information stored within an optional column is completely up to the resource that generates the file. It MUST not be assumed that optional columns having the same name in different mzTab files contain the same type of information. CV parameter accessions MAY be used as optional column names according to the following convention: opt\_{OBJECT\_ID}\_cv\_{accession}\_{parameter name}. Spaces within the parameter’s name MUST be replaced by ‘\_’.

COM Example showing how emPAI values are reported in an additional column from MS file 1 using

COM MS CV parameter “emPAI value” (MS:1001905)

…

PRH accession … opt\_ms\_file[1]\_cv\_MS:1001905\_emPAI\_value

PRT P12345 … 0.658

### Referencing external resources (i.e. mzIdentML or mzQuantML files)

In mzTab all identifications SHOULD reference external resources that contain detailed evidence for the identification. This link MUST be stored in the “uri” column of the respective table. This field MUST NOT be used to reference an external MS data file. MS data files should be referenced using the method described in Section 5.2.

Where these URIs point to depends on the resource that generated the mzTab file. If, for example, PeptideAtlas was exporting data in the mzTab format the URI would be expected to point to the identification’s entry within the respective PeptideAtlas build. mzTab files originating from an mzIdentML file MAY reference the mzIdentML file using the URI column. In case quantitative values are reported coming from an mzQuantML file, the mzQuantML file SHOULD be referenced as it contains the reference to the underlying mzIdentML file.

### Reporting sequence ambiguity

In MS based proteomics approaches, some amino acids cannot be unambiguously identified. To report such ambiguous amino acid identifications, the following symbols SHOULD be used:

Asparagine or aspartic acid B

Glutamine or glutamic acid Z

Leucine or Isoleucine J

Unspecified or unknown amino acid X

### Reporting decoy peptide identifications

To report the results of a target-decoy search, decoy identifications MAY be labeled using the optional column “opt\_global\_cv\_MS:1002217\_decoy\_peptide”. The value of this column MUST be a Boolean (1/0).

## Other supporting materials

The following example instance documents are available and between them cover all the use cases supported.

All example files can be downloaded from:

<http://code.google.com/p/mztab/wiki/ExampleFiles>

1. mztab\_SILAC\_example.txt - (hand crafted) mzTab file showing how SILAC data can be reported.
2. mztab\_itraq\_example.txt - (hand crafted) mzTab file showing how iTRAQ data can be reported.
3. mztab\_merged\_example.txt - merged version of the example file a and b.
4. PRIDE\_Exp\_Complete\_Ac\_16649.xml-mztab.txt - file generated using the mztab-exporter (converted PRIDE experiment accession 16649) containing iTRAQ data.
5. mztab\_lipidomics\_example.txt – Example containing MS lipidomics data produced by the Lipid Data Analyzer tool (<http://genome.tugraz.at/lda/lda_download.shtml>).
6. PXD000002\_mztab.txt.gz - Summary file of ProteomeXchange submission PXD000002 (the complete submission can be found at <ftp://ftp.pride.ebi.ac.uk/2012/03/PXD000002/>).
7. CPTAC\_Progenesis\_label\_free\_mzq.txt - Label free example. Created by an exporter from an mzQuantML file.

# Format specification

This section describes the structure of an mzTab file.

* **Field separator**  
  The column delimiter is the Unicode Horizontal Tab character (Unicode codepoint 0009).
* **File encoding**  
  The UTF-8 encoding of the Unicode character set is the preferred encoding for mzTab files. However, parsers should be able to recognize commonly used encodings.
* **Case sensitivity**  
  All column labels and field names are case-sensitive.
* **Line prefix**  
  Every line in an mzTab file MUST start with a three letter code identifying the type of line delimited by a Tab character. The three letter codes are as follows:
  + MTD for metadata
  + PRH for the protein table header line (the column labels)
  + PRT for rows of the protein table
  + PEH for the peptide table header line (the column labels)
  + PEP for rows of the peptide table
  + PSH for the PSM table header (the column labels)
  + PSM for rows of the PSM table
  + SMH for small molecule table header line (the column labels)
  + SML for rows of the small molecule table
  + COM for comment lines
* **Header lines**Each table based section (protein, peptide, PSM and small molecule) MUST start with the corresponding header line. These header lines MUST only occur once in the document since each section also MUST only occur once.
* **Dates**  
  Dates and times MUST be supplied in the ISO 8601 format (“YYYY-MM-DD”, “YYYY-MM-DDTHH:MMZ” respectively).
* **Decimal separator**  
  In mzTab files the dot (“.”) MUST be used as decimal separator. Thousand separators MUST NOT be used in mzTab files.
* **Comment lines and empty lines**  
  Comment lines can be placed anywhere in an mzTab file. These lines must start with the three-letter code COM and are ignored by most parsers. Empty lines can also occur anywhere in an mzTab file and are ignored.
* **Params**  
  mzTab makes use of CV parameters. As mzTab is expected to be used in several experimental environments where parameters might not yet be available for the generated scores etc. all parameters can either report CV parameters or user parameters that only contain a name and a value.  
  Parameters are always reported as [CV label, accession, name, value]. Any field that is not available MUST be left empty.  
    
  [MS, MS:1001207, Mascot,]  
  [MS, MS:1001171, Mascot:score, 40.21]  
  [,,A user parameter, The value]

In case, the name of the param contains commas, quotes MUST be added to avoid problems with the parsing: [label, accession, “first part of the param name , second part of the name”, value].

[MOD, MOD:00648, “N,O-diacetylated L-serine”,]

* **Sample IDs**To be able to supply metadata specific to each sample, ids in the format sample[1-n] are used.  
    
  MTD sample[1]-species [NEWT, 9606, Homo sapiens (Human), ]

* **Assay IDs**To be able to supply metadata specific to each assay, ids in the format assay[1-n] are used.  
    
  MTD CPTAC-assay[1]-quantification\_reagent [MS,MS:1002038,unlabeled sample,]
* **Study variable IDs**To be able to supply metadata specific to each study variable (grouping of assays), ids in the format study\_variable[1-n] are used.

MTD CPTAC-study\_variable[1]-description Group B (spike-in 0.74 fmol/uL)

## Sections

mzTab files can contain five different sections. The MANDATORY metadata section is made up of key-value pairs. The other four sections are OPTIONAL: protein, peptide, PSM and small molecule section are table-based.

Every section in an mzTab file MUST only occur once if present. If the Peptide and Protein Section are present, the information MUST be consistent between both sections.Field names with indices in square brackets MUST be numbered sequentially and non-decreasing (starting at the first value indicated in the bracket; single integer steps).

## Metadata Section

The metadata section can provide additional information about the dataset(s) reported in the mzTab file. All fields in the metadata section are optional apart from two exceptions:

- “mzTab\_version” MUST always be reported.

- “ms\_file[1-n]-location” MUST be reported if spectra external spectra are referenced using the “spectra\_ref” column in any of the table based sections or if PSMs are reported, since the ms\_file[1-n] serves as a unique identifier for the MS run that was analysed.

The fields in the metadata section should be reported first in order of the UNIT\_IDs then in the order of the various fields listed here. The field’s name and value MUST be separated by a tab character:

MTD publication [PRIDE, PRIDE:00000029, PubMed, 12345]

In the following list of fields any term encapsulated by {} is meant as a variable which MUST be replaced accordingly. The multiplicity numbers given in the descriptions below refer to one unit of analysis i.e. one file. For example, title MAY only be specified once per file.

### mzTab-version

|  |  |
| --- | --- |
| **Description:** | The version of the mzTab file. |
| **Type:** | String |
| **Multiplicity:** | 1 |
| **Example:** | MTD mzTab-Version 1.0 rc4 |

### mzTab-ID

|  |  |
| --- | --- |
| **Description:** | The ID of the mzTab file. |
| **Type:** | String |
| **Multiplicity:** | 0 .. 1 |
| **Example:** | MTD mzTab-ID PRIDE\_1234 |

### title

|  |  |
| --- | --- |
| **Description:** | The file’s human readable title. |
| **Type:** | String |
| **Multiplicity:** | 0 .. 1 |
| **Example:** | MTD title My first test experiment |

### description

|  |  |
| --- | --- |
| **Description:** | The file’s human readable description. |
| **Type:** | String |
| **Multiplicity:** | 0 .. 1 |
| **Example:** | MTD description An experiment investigating the effects of Il-6. |

### sample\_processing[1-n]

|  |  |
| --- | --- |
| **Description:** | A list of parameters describing a sample processing step. The order of the data\_processing items should reflect the order these processing steps were performed in. If multiple parameters are given for a step these MUST be separated by a “|”. |
| **Type:** | Parameter List |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD sample\_processing[1] [SEP, SEP:00173, SDS PAGE,] MTD sample\_processing[2] [SEP, SEP:00142, enzyme digestion,]|[MS, …  MS:1001251, Trypsin, ] |

### instrument[1-n]-name

|  |  |
| --- | --- |
| **Description:** | The name of the instrument used in the experiment. Multiple instruments are numbered 1..n. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD instrument[1]-name [MS, MS:1000449, LTQ Orbitrap,]  … MTD instrument[2]-name [MS, MS:1000031, Instrument model, name of the instrument not included in the CV] |

### instrument[1-n]-source

|  |  |
| --- | --- |
| **Description:** | The instrument's source used in the experiment. Multiple instruments are numbered 1..n. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD instrument[1]-source [MS, MS:1000073, ESI,] … MTD instrument[2]-source [MS, MS:1000598, ETD,] |

### instrument[1-n]-analyzer

|  |  |
| --- | --- |
| **Description:** | The instrument’s analyzer type used in the experiment. Multiple instruments are enumerated 1..n. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD instrument[1]-analyzer [MS, MS:1000291, linear ion trap,] … MTD instrument[2]-analyzer [MS, MS:1000484, orbitrap,] |

### instrument[1-n]-detector

|  |  |
| --- | --- |
| **Description:** | The instrument's detector type used in the experiment. Multiple instruments are numbered 1..n. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD instrument[1]-detector [MS, MS:1000253, electron multiplier,] … MTD instrument[2]-detector [MS, MS:1000348, focal plane collector,] |

### software[1-n]

|  |  |
| --- | --- |
| **Description:** | Software used to analyze the data and obtain the results reported. The parameter’s value SHOULD contain the software’s version. The order (numbering) should reflect the order in which the tools were used. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD software[1] [MS, MS:1001207, Mascot, 2.3] MTD software[2] [MS, MS:1001561, Scaffold, 1.0] |

### software[1-n]-setting

|  |  |
| --- | --- |
| **Description:** | A software setting used. This field MAY occur multiple times for a single software. The value of this field is deliberately set as a String, since there currently do not exist cvParams for every possible setting. |
| **Type:** | String |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD software[1]-setting Fragment tolerance = 0.1 Da  MTD software[1]-setting Parent tolerance = 0.5 Da |

### false\_discovery\_rate

|  |  |
| --- | --- |
| **Description:** | The file’s false discovery rate(s) reported at the PSM, peptide, and/or protein level. Multiple parameters MUST be separated by “|”. |
| **Type:** | Parameter List |
| **Multiplicity:** | 0 .. 1 |
| **Example:** | MTD false\_discovery\_rate [MS, MS:1001364, pep:global FDR, 0.01]|…  [MS, MS:1001214, prot:global FDR, 0.08] |

### publication[1-n]

|  |  |
| --- | --- |
| **Description:** | A publication associated with this file. Several publications can be given by indicating the number in the square brackets after “publication”. PubMed ids must be prefixed by “pubmed:”, DOIs by “doi:”. Multiple identifiers MUST be separated by “|”. |
| **Type:** | String List |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD publication[1] pubmed:21063943|doi:10.1007/978-1-60761-987-1\_6 MTD publication[2] pubmed:20615486|doi:10.1016/j.jprot.2010.06.008 |

### contact[1-n]-name

|  |  |
| --- | --- |
| **Description:** | The contact's name. Several contacts can be given by indicating the number in the square brackets after "contact". A contact has to be supplied in the format [first name] [initials] [last name] (see example). |
| **Type:** | String |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD contact[1]-name James D. Watson … MTD contact[2]-name Francis Crick |

### contact[1-n]-affiliation

|  |  |
| --- | --- |
| **Description:** | The contact’s affiliation. |
| **Type:** | String |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD contact[1]-affiliation Cambridge University, UK MTD contact[2]-affiliation Cambridge University, UK |

### contact[1-n]-email

|  |  |
| --- | --- |
| **Description:** | The contact’s e-mail address. |
| **Type:** | String |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD contact[1]-email watson@cam.ac.uk … MTD contact[2]-email crick@cam.ac.uk |

### uri

|  |  |
| --- | --- |
| **Description:** | A URI pointing to the file's source data (e.g., a PRIDE experiment or a PeptideAtlas built). |
| **Type:** | URI |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD uri http://www.ebi.ac.uk/pride/url/to/experiment  MTD uri http://proteomecentral.proteomexchange.org/cgi/GetDataset |

### mod

|  |  |
| --- | --- |
| **Description:** | A list of “|” separated parameters describing all (distinct) modifications reported in this unit. |
| **Type:** | Parameter List |
| **Multiplicity:** | 0 .. 1 |
| **Example:** | MTD mod [MOD, MOD:00397, iodoacetamide derivatized residue, ]|…  [MOD, MOD:00675, oxidized residue, ] |

### quantification\_method

|  |  |
| --- | --- |
| **Description:** | The quantification method used in the experiment reported in the file. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. 1 |
| **Example:** | MTD quantification\_method [MS, MS:1001837, iTRAQ quantitation analysis, ] |

### protein-quantification\_unit

|  |  |
| --- | --- |
| **Description:** | Defines what type of units is reported in the protein quantification fields. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. 1 |
| **Example:** | MTD protein-quantification\_unit [PRIDE, PRIDE:0000395, Ratio, ] |

### peptide-quantification\_unit

|  |  |
| --- | --- |
| **Description:** | Defines what type of units is reported in the peptide quantification fields. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. 1 |
| **Example:** | MTD peptide-quantification\_unit [PRIDE, PRIDE:0000395, Ratio, ] |

### small\_molecule-quantification\_unit

|  |  |
| --- | --- |
| **Description:** | Defines what type of units is reported in the small molecule quantification fields. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. 1 |
| **Example:** | MTD small\_molecule-quantification\_unit [PRIDE, PRIDE:0000395, Ratio, ] |

### ms\_file[1-n]-format

|  |  |
| --- | --- |
| **Description:** | A parameter specifying the data format of the external MS data file. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD ms\_file[1]-format [MS, MS:1000584, mzML file, ] … MTD ms\_file[2]-format [MS, MS:1001062, Mascot MGF file, ] |

### ms\_file[1-n]-location

|  |  |
| --- | --- |
| **Description:** | Location of the external data file. If assays are reported or if a PSM section is present in the file, these attributes are MANDATORY, since back references to the MS file MUST be provided in certain sections. If the actual location of the MS file is unknown, a place holder value SHOULD be inserted. |
| **Type:** | URL |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD ms\_file[1]-location file://C:\path\to\my\file … MTD ms\_file[2]-location <ftp://ftp.ebi.ac.uk/path/to/file> |

### ms\_file[1-n]-id\_format

|  |  |
| --- | --- |
| **Description:** | Parameter specifying the id format used in the external data file. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD ms\_file[1]-id\_format [MS, MS:1001530, mzML unique identifier, ] … MTD ms\_file[2]-id\_format [MS, MS:1000774, multiple peak list …  nativeID format, ] |

### custom

|  |  |
| --- | --- |
| **Description:** | Any additional parameters describing the analysis reported. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD custom [,,MS operator, Florian] |

### {SAMPLE\_ID}-species[1-n]

|  |  |
| --- | --- |
| **Description:** | The respective species of the samples analysed. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | COM Experiment where all samples consisted of the same two species MTD sample[1]-species[1] [NEWT, 9606, Homo sapiens (Human), ] MTD sample[1]-species[2] [NEWT, 12059, Rhinovirus, ]  COM Experiment where different two samples from different species (combinations) COM were analysed as biological replicates.  MTD sample[1]-species[1] [NEWT, 9606, Homo sapiens (Human), ] MTD sample[1]-species[2] [NEWT, 573824, Human rhinovirus 1, ] MTD sample[2]-species[1] [NEWT, 9606, Homo sapiens (Human), ] MTD sample[2]-species[2] [NEWT, 12130, Human rhinovirus 2, ] |

### {SAMPLE\_ID}-tissue[1-n]

|  |  |
| --- | --- |
| **Description:** | The respective tissue(s) of the sample. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD sample[1]-tissue[1] [BTO, BTO:0000759, liver, ] |

### {SAMPLE\_ID}-cell\_type[1-n]

|  |  |
| --- | --- |
| **Description:** | The respective cell type(s) of the sample. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD sample[1]-cell\_type[1] [CL, CL:0000182, hepatocyte, ] |

### {SAMPLE\_ID}-disease[1-n]

|  |  |
| --- | --- |
| **Description:** | The respective disease(s) of the sample. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD sample[1]-disease[1] [DOID, DOID:684, hepatocellular carcinoma, ] MTD sample[1]-disease[2] [DOID, DOID:9451, alcoholic fatty liver, ] |

### {SAMPLE\_ID}-description

|  |  |
| --- | --- |
| **Description:** | A human readable description of the sample. |
| **Type:** | String |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD sample[1]-description Hepatocellular carcinoma samples. MTD sample[2]-description Healthy control samples. |



### {SAMPLE\_ID}-custom

|  |  |
| --- | --- |
| **Description:** | Parameters describing the sample’s additional properties. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD sample[1]-custom [,,Extraction date, 2011-12-21] MTD sample[1]-custom [,,Extraction reason, liver biopsy] |

### {ASSAY\_ID}-quantification\_reagent

|  |  |
| --- | --- |
| **Description:** | The reagent used to label the sample in the assay. For label-free analyses and for the “light” channel in label-based experiments, the “unlabeled sample” CV term SHOULD be used. If assays are reported in the file, the quantification reagent is MANDATORY. |
| **Type:** | Parameter |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD assay[1]-quantification\_reagent [PRIDE,PRIDE:0000114,iTRAQ reagent,114] MTD assay[2]-quantification\_reagent [PRIDE,PRIDE:0000115,iTRAQ reagent,115]  OR  MTD assay[1]-quantification\_reagent [MS,MS:1002038,unlabeled sample,] |

### {ASSAY\_ID}-sample\_ref

|  |  |
| --- | --- |
| **Description:** | An association from a given assay to the sample analysed. |
| **Type:** | {SAMPLE\_ID} |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD assay[1]-sample\_ref sample[1] MTD assay[2]-sample\_ref sample[2] |

### {ASSAY\_ID}-ms\_file\_ref

|  |  |
| --- | --- |
| **Description:** | An association from a given assay to the source MS file. If assays are reported in the file, the ms\_file reference is MANDATORY. |
| **Type:** | {MS\_FILE\_ID} |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD assay[1]-ms\_file\_ref ms\_file[1] |

### {STUDY\_VARIABLE\_ID}-assay\_refs

|  |  |
| --- | --- |
| **Description:** | Comma-separated references to the IDs of assays grouped in the study variable. If both assays and study variables are reported in the file, the references are MANDATORY. |
| **Type:** | {ASSAY\_ID}, ... |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD study\_variable[1]-assay\_refs assay[1], assay[2], assay[3] |

### {STUDY\_VARIABLE\_ID}-sample\_refs

|  |  |
| --- | --- |
| **Description:** | Comma-separated references to the samples that were analysed in the study variable. |
| **Type:** | {SAMPLE\_ID}, ... {SAMPLE\_ID} |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD study\_variable[1]-sample\_refs sample[1] |

### {STUDY\_VARIABLE\_ID}-description

|  |  |
| --- | --- |
| **Description:** | A textual description of the study variable. |
| **Type:** | String |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD study\_variable[1]-description Group B (spike-in 0.74 fmol/uL) |

### colunit-protein

|  |  |
| --- | --- |
| **Description:** | Defines the unit for the data reported in a column of the protein section. The format of the value has to be {column name}={Parameter defining the unit}  This field MUST NOT be used to define a unit for quantification columns. The unit used for protein quantification values MUST be set in *protein-quantification\_unit*. |
| **Type:** | String |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD colunit-protein protein\_abundance\_sub[1] =[EFO, EFO:0004374, milligram per deciliter,] |

### colunit-peptide

|  |  |
| --- | --- |
| **Description:** | Defines the used unit for a column in the peptide section. The format of the value has to be {column name}={Parameter defining the unit}  This field MUST NOT be used to define a unit for quantification columns. The unit used for peptide quantification values MUST be set in peptide-quantification\_unit. |
| **Type:** | String |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD colunit-peptide retention\_time=[UO,UO:0000031, minute,] |

### colunit-small\_molecule

|  |  |
| --- | --- |
| **Description:** | Defines the used unit for a column in the small molecule section. The format of the value has to be {column name}={Parameter defining the unit}  This field MUST NOT be used to define a unit for quantification columns. The unit used for small molecule quantification values MUST be set in small\_molecule-quantification\_unit. |
| **Type:** | String |
| **Multiplicity:** | 0 .. \* |
| **Example:** | MTD colunit-small\_molecule retention\_time=[UO,UO:0000031, minute,] |

## Protein Section

The protein section is table-based. The protein section MUST always come after the metadata section. All table columns MUST be tab-separated. There MUST NOT be any empty cells. Missing values MUST be reported using “null”. All columns are mandatory unless specified otherwise. The order of columns is not specified although for ease of human interpretation, it is RECOMMENDED to follow the order specified below.

### accession

|  |  |
| --- | --- |
| **Description:** | The accession of the protein in the source database. A protein accession MUST be unique within one mzTab file. If different quantification values are required for the same underlying accession, for example if differentially modified forms of a protein have been quantified, a suitable suffix SHOULD be appended to the accession. |
| **Type:** | String |
| **Example:** | PRH accession … PRT P12345 … PRT P12346 … |



### description

|  |  |
| --- | --- |
| **Description:** | The protein’s name and or description line. |
| **Type:** | String |
| **Example:** | PRH accession description … PRT P12345 Aspartate aminotransferase, mitochondrial … PRT P12346 Serotransferrin … |

### taxid

|  |  |
| --- | --- |
| **Description:** | The NCBI/NEWT taxonomy id for the species the protein was identified in. |
| **Type:** | Integer |
| **Example:** | PRH accession … taxid … PRT P12345 … 10116 … PRT P12346 … 10116 … |

### species

|  |  |
| --- | --- |
| **Description:** | The human readable species the protein was identified in - this SHOULD be the NCBI entry’s name. |
| **Type:** | String |
| **Example:** | PRH accession … taxid species … PRT P12345 … 10116 Rattus norvegicus (Rat) … PRT P12346 … 10116 Rattus norvegicus (Rat) … |

### database

|  |  |
| --- | --- |
| **Description:** | The protein database used for the search (could theoretically come from a different species). Wherever possible the Miriam (<http://www.ebi.ac.uk/miriam>) assigned name SHOULD be used. |
| **Type:** | String |
| **Example:** | PRH accession … taxid species database … PRT P12345 … 10116 Rattus norvegicus (Rat) UniProtKB … PRT P12346 … 10116 Rattus norvegicus (Rat) UniProtKB … |

### database\_version

|  |  |
| --- | --- |
| **Description:** | The protein database's version – in case there is no version available (custom build) the creation / download (e.g., for NCBI nr) date SHOULD be given. Additionally, the number of entries in the database MAY be reported in round brackets after the version in the format: {version} ({#entries} entries), for example “2011-11 (1234 entries)”. |
| **Type:** | String |
| **Example:** | PRH accession … taxid species database database\_version … PRT P12345 … 10116 Rattus norvegicus (Rat) UniProtKB 2011\_11 … PRT P12346 … 10116 Rattus norvegicus (Rat) UniProtKB 2011\_11 … |

### search\_engine

|  |  |
| --- | --- |
| **Description:** | A “|” delimited list of search engine(s) used to identify this protein. Search engines MUST be supplied as parameters. |
| **Type:** | Parameter List |
| **Example:** | COM In this example the first protein was identified by Mascot and Sequest while COM the second protein was only identified by Mascot. PRH accession … search\_engine … PRT P12345 … [MS,MS:1001207,Mascot,]|[MS,MS:1001208,Sequest,] … PRT P12346 … [MS,MS:1001207,Mascot,] … |

### best\_search\_engine\_score

|  |  |
| --- | --- |
| **Description:** | A “|” delimited list of the best search engine score(s) for the given protein across all replicates reported. Scores SHOULD be reported using CV parameters whenever possible. |
| **Type:** | Parameter List |
| **Example:** | PRH accession … best\_search\_engine\_score\_ms\_file[1] … PRT P12345 … [MS,MS:1001171,Mascot score,50]|[MS,MS:1001155,Sequest:xcorr,2] … PRT P12346 … [MS,MS:1001171,Mascot score,47.2] … |

### search\_engine\_score\_ms\_file[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present)  A “|” delimited list of search engine score(s) for the given protein. Scores SHOULD be reported using CV parameters whenever possible. |
| **Type:** | Parameter List |
| **Example:** | PRH accession … search\_engine\_score\_ms\_file[1] … PRT P12345 … [MS,MS:1001171,Mascot score,50]|[MS,MS:1001155,Sequest:xcorr,2] … PRT P12346 … [MS,MS:1001171,Mascot score,47.2] … |

### reliability

|  |  |
| --- | --- |
| **Description:** | The reliability of the given protein identification. This must be supplied by the resource and has to be one of the following values:  1: high reliability  2: medium reliability  3: poor reliability  Important: An identification's reliability is resource-dependent. |
| **Type:** | Integer |
| **Example:** | PRH accession … reliability … PRT P12345 … 3 … PRT P12346 … 1 … |

### num\_psms\_ms\_file[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present)  The total number of PSMs that were used to identify this protein from a given ms\_file. |
| **Type:** | Integer |
| **Example:** | COM P12345 is identified through ABCM, ABCM+Oxidation, CDE, CDE … PRH accession … num\_psms\_ms\_file[1] … PRT P12345 … 4 … |

### num\_peptides\_distinct\_ms\_file[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present)  The number of distinct peptide sequences identifying this protein in a given ms\_file. Distinct peptides are defined based on their sequence, ignoring different modifications or charge states. |
| **Type:** | Integer |
| **Example:** | COM P12345 is identified through ABCM, ABCM+Oxidation, CDE, CDE … PRH accession … num\_peptides\_distinct\_ms\_file[1] … PRT P12345 … 3 … |

### num\_peptides\_unique\_ms\_file[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present)  The number of peptides that are mapped uniquely to this protein and the other ambiguity members in this ms\_file. |
| **Type:** | Integer |
| **Example:** | COM P12345 is identified through ABCM, ABCM+Oxidation, CDE, CDE COM ABCM is only from P12345, CDE from P12345 and P12346 … PRH accession … num\_peptides\_unique\_ms\_file[1] … PRT P12345 … 2 … |

### ambiguity\_members

|  |  |
| --- | --- |
| **Description:** | A comma-delimited list of protein accessions. This field should be set in the representative protein of the ambiguity group (the protein identified through the accession in the first column). The accessions listed in this field should identify proteins that could also be identified through these peptides (e.g. “same-set proteins”) but were not chosen by the researcher or resource, often for arbitrary reasons. It is NOT RECOMMENDED to report subset proteins as ambiguity\_members, since the proteins reported here, together with the representative protein are taken to be a group that cannot be separated based on the peptide evidence. |
| **Type:** | String List |
| **Example:** | COM P12345, P12347, and P12348 can all be identified through the same peptides … PRH accession … ambiguity\_members … PRT P12345 … P12347,P12348 … |

### modifications

|  |  |
| --- | --- |
| **Description:** | A comma delimited list of modifications found in the given protein. Modifications have to be reported in the following format:  {position in protein}{Parameter}-{Modification or Substitution identifier}|{neutral loss}  Modification location scores can be supplied using the optional Parameter object. In case the position of the modification is uncertain multiple positions cMAY be supplied delimited by a "|".  Furthermore, in case a position is unknown no position information MAY be supplied.  Terminal modifications MUST be reported at position 0 or protein size + 1 respectively.  Valid modification identifiers are either PSI-MOD or UNIMOD accession (including the “MOD:” / “UNIMOD:” prefix) or CHEMMODS. CHEMMODS have the format CHEMMOD:+/-{chemical formula or *m/z* delta}. Valid CHEMMODS are for example “CHEMMOD:+NH4” or “CHEMMOD:-10.1098”. CHEMMODs MUST NOT be used if the modification can be reported using a PSI-MOD or UNIMOD accession. Mass deltas MUST NOT be used for CHEMMODs if the delta can be expressed through a known chemical formula.  Neutral losses MAY be reported as cvParams. If a neutral loss is not associated with an existing modification it is reported as separated comma-separated entry. Otherwise, the neutral loss MUST be reported after the modification it is associated with and separated by a ‘|’ from the modification.  Additionally, it is possible to report substitutions of amino acids using SUBST:{amino acid}.  If different modifications are identified from different ms\_files, a superset of the identified modifications SHOULD be reported here. Detailed modification mapping to individual ms\_files is provided through the PSM table. |
| **Type:** | String |
| **Example:** | COM Protein P12345 TESTPEPTIDES with 2 phosphorylation sites: TEpSTPEPpTIDES  COM Common use cases without score:  COM Example 1: Both locations have been determined  PRH accession … modifications …  PRT P12345 … 3-MOD:00412,8-MOD:00412 …  COM Example 2: Like Ex. 1, but first site localization is ambiguous (S or T)  PRH accession … modifications …  PRT P12345 … 3|4-MOD:00412,8-MOD:00412 …  COM Example 3: Protein only known to contain two phosphor sites in the range 3 to 8  PRH accession … modifications …  PRT P12345 … 3|4|8-MOD:00412, 3|4|8-MOD:00412 …  COM Example 4: No position information or only accurate mass available  PRH accession … modifications …  PRT P12345 … CHEMMOD:+159.93 …  COM Common use cases with probability scores:  COM Example 5: MOD:00412 with associated probabilities at position 3 and 4  COM and a probability of 0.3 at position 8  PRH accession … modifications …  PRT P12345 … 3[MS,MS:1001876, modification probability, 0.8]|4[MS,MS:1001876, modification probability, 0.2]-MOD:00412,8[MS,MS:1001876, modification probability, 0.3]-MOD:00412 …  COM Reporting substitutions  COM Example 6: Substitution of amino acid at position 3 with R (Original sequence is reported in sequence column)  PRH accession … modifications  PRT P12345 … 3-SUBST:R |

### uri

|  |  |
| --- | --- |
| **Description:** | A URI pointing to the protein's source entry in the unit it was identified in (e.g., the PRIDE database or a local database / file identifier). |
| **Type:** | URI |
| **Example:** | PRT accession … uri … PRH P12345 … http://www.ebi.ac.uk/pride/url/to/P12345 … |

### go\_terms

|  |  |
| --- | --- |
| **Description:** | A ’|’-delimited list of GO accessions for this protein. |
| **Type:** | String List |
| **Example:** | PRT accession … go\_terms … PRH P12345 … GO:0006457|GO:0005759|GO:0005886|GO:0004069 … |

### protein\_coverage

|  |  |
| --- | --- |
| **Description:** | A value between 0 and 1 defining the protein coverage. |
| **Type:** | Double |
| **Example:** | PRT accession … protein\_coverage … PRH P12345 … 0.4 … |

### protein\_abundance\_assay[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The protein's abundance as measured in the given assay through whatever technique was employed. |
| **Type:** | Double |
| **Example:** | PRT accession … protein\_abundance\_assay[1] … protein\_abundance\_assay[2] … PRH P12345 … 0.4 … 0.2 … |

### protein\_abundance\_study\_variable[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The protein's abundance as measured in the given Study Variable, for example mean or median of quantitative values reported in Assays. |
| **Type:** | Double |
| **Example:** | PRT accession … protein\_abundance\_assay[1] … protein\_abundance\_assay[2] … PRH P12345 … 0.4 … 0.2 … |

### protein\_abundance\_stdev\_study\_variable[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The standard deviation of the protein’s abundance. If a protein’s abundance is given for a certain study variable, the corresponding standard deviation column MUST also be present (in case the value is not available “null” should be used). |
| **Type:** | Double |
| **Example:** | PRT accession … protein\_abundance\_stdev\_study\_variable[1] … PRH P12345 … 0.4 … |

### protein\_abundance\_std\_error\_study\_variable [1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The standard error of the protein’s abundance. If a protein’s abundance is given for a certain study variable, the corresponding standard error column MUST also be present (in case the value is not available “null” should be used). |
| **Type:** | Double |
| **Example:** | PRT accession … protein\_abundance\_sub[1] … protein\_abundance\_std\_error\_sub[1] … PRH P12345 … 0.4 … 0.03 … |

### opt\_{ASSAY\_ID}|{STUDY\_VARIABLE\_ID}|{MS\_FILE\_ID}|“global”\_\*

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) Additional columns can be added to the end of the protein table. These column headers MUST start with the prefix “opt\_” followed by the identifier of the object they reference: assay, study variable, MS file or “global” (if the value relates to all replicates). Column names MUST only contain the following characters: ‘A’-‘Z’, ‘a’-‘z’, ‘0’-‘9’, ‘\_’, ‘-’, ‘[’, ‘]’, and ‘:’. CV parameter accessions MAY be used for optional columns following the format: opt\_{OBJECT\_ID}\_cv\_{accession}\_{parameter name}. Spaces within the parameter’s name MUST be replaced by ‘\_’. |
| **Type:** | Column |
| **Example:** | PRT accession … opt\_assay[1]\_my\_value opt\_global\_another\_value PRH P12345 … My value about assay[1] some other value that is across reps |

## Peptide Section

The peptide section is table based. The peptide section must always come after the metadata section and or protein section if these are present in the file. All table columns MUST be tab separated. There MUST NOT be any empty cells. All columns, unless specified otherwise, are mandatory. The order of columns is not specified although for ease of human interpretation, it is RECOMMENDED to follow the order specified below.

### sequence

|  |  |
| --- | --- |
| **Description:** | The peptide's sequence |
| **Type:** | String |
| **Example:** | PEH sequence … PEP KVPQVSTPTLVEVSR … PEP EIEILACEIR … |

### accession

|  |  |
| --- | --- |
| **Description:** | The protein's accession the peptide is associated with. In case no protein section is present in the file or the peptide was not assigned to a protein the field should be filled with “null”. If the peptide can be assigned to more than one protein... |
| **Type:** | String |
| **Example:** | PEH sequence accession … PEP KVPQVSTPTLVEVSR P02768 … |



### unique

|  |  |
| --- | --- |
| **Description:** | Indicates whether the peptide is unique for this protein in respect to the searched database. |
| **Type:** | Boolean (0/1) |
| **Example:** | PEH sequence accession unique … PEP KVPQVSTPTLVEVSR P02768 0 … PEP VFDEFKPLVEEPQNLIK P02768 1 … |

### database

|  |  |
| --- | --- |
| **Description:** | The protein database used for the search (could theoretically come from a different species) and the peptide sequence comes from. |
| **Type:** | String |
| **Example:** | PEH sequence accession unique database … PEP KVPQVSTPTLVEVSR P02768 0 UniProtKB … PEP VFDEFKPLVEEPQNLIK P02768 1 UniProtKB … |

### database\_version

|  |  |
| --- | --- |
| **Description:** | The protein database's version – in case there is no version available (custom build) the creation / download (e.g., for NCBI nr) date should be given.  Additionally, the number of entries in the database MAY be reported in round brackets after the version in the format: {version} ({#entries} entries), for example “2011-11 (1234 entries)”. |
| **Type:** | String |
| **Example:** | PEH sequence accession unique database database\_version … PEP KVPQVSTPTLVEVSR P02768 0 UniProtKB 2011\_11 … PEP VFDEFKPLVEEPQNLIK P02768 1 UniProtKB 2011\_11 … |

### search\_engine

|  |  |
| --- | --- |
| **Description:** | A “|” delimited list of search engine(s) used to identify this peptide. Search engines must be supplied as parameters. |
| **Type:** | Parameter List |
| **Example:** | PEH sequence … search\_engine … PEP KVPQVSTPTLVEVSR … [MS,MS:1001207,Mascot,]|[MS,MS:1001208,Sequest,] … PEP VFDEFKPLVEEPQNLIK … [MS,MS:1001207,Mascot,] … |

### best\_search\_engine\_score

|  |  |
| --- | --- |
| **Description:** | A “|” delimited list of best search engine score(s) for the given peptide across all replicates. Scores SHOULD be reported using CV parameters whenever possible. |
| **Type:** | Parameter List |
| **Example:** | PEH sequence … best\_search\_engine\_score … PEP KVPQVSTPTLVEVSR … [MS,MS:1001155,Sequest:xcorr,2] … PEP VFDEFKPLVEEPQNLIK … [MS,MS:1001171,Mascot score,47.2] … |

### search\_engine\_score\_ms\_file[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present)  A “|” delimited list of search engine score(s) for the given peptide from a given MS file. Scores SHOULD be reported using CV parameters whenever possible. |
| **Type:** | Parameter List |
| **Example:** | PEH sequence … search\_engine\_score\_ms\_file[1] … PEP KVPQVSTPTLVEVSR … [MS,MS:1001155,Sequest:xcorr,2] … PEP VFDEFKPLVEEPQNLIK … [MS,MS:1001171,Mascot score,47.2] … |

### reliability

|  |  |
| --- | --- |
| **Description:** | The reliability of the given peptide identification. This must be supplied by the resource and has to be one of the following values:  1: high reliability  2: medium reliability  3: poor reliability  Important: An identification's reliability is resource dependent. |
| **Type:** | Integer |
| **Example:** | PEH sequence … reliability … PEP KVPQVSTPTLVEVSR … 3 … PEP VFDEFKPLVEEPQNLIK … 1 … |

### modifications

|  |  |
| --- | --- |
| **Description:** | The peptide's modifications or substitutions. To further distinguish peptide terminal modifications, these SHOULD be reported at position 0 or *peptide size* + 1 respectively. For detailed information see the modifications section in the protein table. If substitutions are reported, the “sequence” column MUST contain the original, unaltered sequence. For quantitation approaches, the assumed modifications present on the peptide across all replicates SHOULD be reported i.e. the labels/tags identified on particular peptides SHOULD NOT be reported. It is thus also expected that modification reliability scores will typically be reported at the PSM-level only. |
| **Type:** | String |
| **Example:** | PEH sequence … modifications … PEP KVPQVSTPTLVEVSR … 10-MOD:00412 … PEP VFDEFKPLVEEPQNLIK … NA … |

### retention\_time

|  |  |
| --- | --- |
| **Description:** | A ‘|’-separated list of time points. Semantics may vary on how retention times are reported. For quantification approaches, different exporters MAY wish to export the retention times of all spectra used for quantification (e.g. in MS2 approaches) or the centre point of the feature quantified for MS1 approaches. It is assumed that the reported value(s) are for a given “master” peptide from one assay only (and the unlabeled peptide in label-based approaches). If the exporter wishes to export values for all assays, this can be done using optional columns. |
| **Type:** | Double List |
| **Example:** | PEH sequence … retention\_time … PEP KVPQVSTPTLVEVSR … 10.2 … PEP VFDEFKPLVEEPQNLIK … 15.8 … |

### charge

|  |  |
| --- | --- |
| **Description:** | The charge assigned by the search engine/software. In case multiple charge states for the same peptide are observed these should be reported as distinct entries in the peptide table. In case the charge is unknown “null” MUST be used. |
| **Type:** | Integer |
| **Example:** | PEH sequence … charge … PEP KVPQVSTPTLVEVSR … 2 … PEP VFDEFKPLVEEPQNLIK … 3 … |

### mass\_to\_charge

|  |  |
| --- | --- |
| **Description:** | The precursor’s experimental mass to charge (*m/z*). It is assumed that the reported value is for a given “master” peptide from one assay only (and the unlabeled peptide in label-based approaches). If the exporter wishes to export values for all assays, this can be done using optional columns. |
| **Type:** | Double |
| **Example:** | PEH sequence … mass\_to\_charge … PEP KVPQVSTPTLVEVSR … 1234.4 … PEP VFDEFKPLVEEPQNLIK … 123.4 … |

### uri

|  |  |
| --- | --- |
| **Description:** | A URI pointing to the peptide's entry in the experiment it was identified in (e.g., the peptide’s PRIDE entry). |
| **Type:** | URI |
| **Example:** | PEH sequence … uri … PEP KVPQVSTPTLVEVSR … http://www.ebi.ac.uk/pride/link/to/peptide … PEP VFDEFKPLVEEPQNLIK … http://www.ebi.ac.uk/pride/link/to/peptide … |

### spectra\_ref (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present)  Reference to spectra in a spectrum file. It is expected that spectra\_ref SHOULD only be used for MS2-based quantification approaches, in which retention time values cannot identify the spectra used for quantitation. The reference must be in the format ms\_file[1-n]:{SPECTRA\_REF} where SPECTRA\_REF MUST follow the format defined in 5.2. Multiple spectra MUST be referenced using a “|” delimited list. If “spectra\_ref” is present, the element “ms\_file[1-n]-location” MUST be reported in the metadata section. |
| **Type:** | String |
| **Example:** | PEH sequence … spectra\_ref … PEP KVPQVSTPTLVEVSR … ms\_file[1]:index=5 …  PEP VFDEFKPLVEEPQNLIK … ms\_file[2]:index=7|ms\_file[2]:index=9 … |

### peptide\_abundance\_assay[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The peptide’s abundance in the given assay. |
| **Type:** | Double |
| **Example:** | PEH sequence … peptide\_abundance\_assay[1] peptide\_abundance\_assay[2]…  PEP KVPQVSTPTLVEVSR … 0.4 0.5 |

### peptide\_abundance\_study\_variable[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The peptide’s abundance in the given study variable, for example calculated as an average of assay values. |
| **Type:** | Double |
| **Example:** | PEH sequence … peptide\_abundance\_study\_variable[1] …  PEP KVPQVSTPTLVEVSR … 0.4 … |

### peptide\_abundance\_stdev\_study\_variable[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The standard deviation of the peptide’s abundance for a given study variable. |
| **Type:** | Double |
| **Example:** | PEH sequence … peptide\_abundance\_sub[1] peptide\_abundance\_stdev\_sub[1] … PEP KVPQVSTPTLVEVSR … 0.4 0.2 … |

### peptide\_abundance\_std\_error\_sub[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The standard error of the peptide’s abundance for a given study variable. |
| **Type:** | Double |
| **Example:** | PEH sequence … peptide\_abundance\_sub[1] … peptide\_abundance\_std\_error\_sub[1] … PEP KVPQVSTPTLVEVSR … 0.4 … 0.2 … |

### opt\_{ASSAY\_ID}|{STUDY\_VARIABLE\_ID}|{MS\_FILE\_ID}|“global”\_\*

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) Additional columns can be added to the end of the peptide table. These column headers MUST start with the prefix “opt\_” followed by the identifier of the object they reference: assay, study variable, MS file or “global” (if the value relates to all replicates). Column names MUST only contain the following characters: ‘A’-‘Z’, ‘a’-‘z’, ‘0’-‘9’, ‘\_’, ‘-’, ‘[’, ‘]’, and ‘:’. CV parameter accessions MAY be used for optional columns following the format: opt\_{OBJECT\_ID}\_cv\_{accession}\_{parameter name}. Spaces within the parameter’s name MUST be replaced by ‘\_’. |
| **Type:** | Column |
| **Example:** | PRT accession … opt\_assay[1]\_my\_value opt\_global\_another\_value PRH P12345 … My value about assay[1] some other value that is across reps |



## PSM Section

The PSM section is table-based. The PSM section MUST always come after the metadata section, peptide section and or protein section if they are present in the file. All table columns MUST be Tab separated. There MUST NOT be any empty cells.

All columns, unless specified otherwise, are mandatory. The order of columns is not specified although for ease of human interpretation, it is RECOMMENDED to follow the order specified below.

### sequence

|  |  |
| --- | --- |
| **Description:** | The peptide's sequence |
| **Type:** | String |
| **Example:** | PSH sequence … PSM KVPQVSTPTLVEVSR … PSM EIEILACEIR … |

### PSM\_ID

|  |  |
| --- | --- |
| **Description:** | A unique identifier for a PSM within the file. If a PSM can be matched to multiple proteins, the same PSM should be represented on multiple rows with different accessions and the same PSM\_ID. |
| **Type:** | Integer |
| **Example:** | PSH sequence PSM\_ID accession… PSM KVPQVSTPTLVEVSR 1 P02768 …  PSM PEPTIDR 2 P04267 …  PSM PEPTIDR 2 P04268 … |

### accession

|  |  |
| --- | --- |
| **Description:** | The protein's accession the peptide is associated with. In case no protein section is present in the file or the peptide was not assigned to a protein the field should be filled with “null”. If the peptide can be assigned to more than one protein, the same PSM should be represented on multiple rows with the same unique identifier. |
| **Type:** | String |
| **Example:** | PSH sequence accession … PSM KVPQVSTPTLVEVSR P02768 … |

### unique

|  |  |
| --- | --- |
| **Description:** | Indicates whether the peptide is unique for this protein in respect to the searched database. |
| **Type:** | Boolean (0/1) |
| **Example:** | PSH sequence accession unique … PSM KVPQVSTPTLVEVSR P02768 0 … PSM VFDEFKPLVEEPQNLIK P02768 1 … |

### database

|  |  |
| --- | --- |
| **Description:** | The protein database used for the search (could theoretically come from a different species) and the peptide sequence comes from. |
| **Type:** | String |
| **Example:** | PSH sequence accession unique database … PSM KVPQVSTPTLVEVSR P02768 0 UniProtKB … PSM VFDEFKPLVEEPQNLIK P02768 1 UniProtKB … |

### database\_version

|  |  |
| --- | --- |
| **Description:** | The protein database's version – in case there is no version available (custom build) the creation / download (e.g., for NCBI nr) date should be given.  Additionally, the number of entries in the database MAY be reported in round brackets after the version in the format: {version} ({#entries} entries), for example “2011-11 (1234 entries)”. |
| **Type:** | String |
| **Example:** | PSH sequence accession unique database database\_version … PSM KVPQVSTPTLVEVSR P02768 0 UniProtKB 2011\_11 … PSM VFDEFKPLVEEPQNLIK P02768 1 UniProtKB 2011\_11 … |

### search\_engine

|  |  |
| --- | --- |
| **Description:** | A “|” delimited list of search engine(s) used to create the PSM. Search engines must be supplied as parameters. |
| **Type:** | Parameter List |
| **Example:** | PSH sequence … search\_engine … PSM KVPQVSTPTLVEVSR … [MS,MS:1001207,Mascot,]|[MS,MS:1001208,Sequest,] … PSM VFDEFKPLVEEPQNLIK … [MS,MS:1001207,Mascot,] … |

### search\_engine\_score

|  |  |
| --- | --- |
| **Description:** | A “|” delimited list of search engine score(s) for the given PSM. |
| **Type:** | Parameter List |
| **Example:** | PSH sequence … best\_search\_engine\_score … PSM KVPQVSTPTLVEVSR … [MS,MS:1001155,Sequest:xcorr,2] … PSM VFDEFKPLVEEPQNLIK … [MS,MS:1001171,Mascot score,47.2] … |

### reliability

|  |  |
| --- | --- |
| **Description:** | The reliability of the given PSM. This must be supplied by the resource and has to be one of the following values:  1: high reliability  2: medium reliability  3: poor reliability  Important: An identification's reliability is resource dependent. |
| **Type:** | Integer |
| **Example:** | PSH sequence … reliability … PSM KVPQVSTPTLVEVSR … 3 … PSM VFDEFKPLVEEPQNLIK … 1 … |

### modifications

|  |  |
| --- | --- |
| **Description:** | The peptide's modifications or substitutions. To further distinguish peptide terminal modifications, these SHOULD be reported at position 0 or *peptide size* + 1 respectively. For detailed information see the modifications section in the protein table. If substitutions are reported, the “sequence” column MUST contain the original, unaltered sequence. |
| **Type:** | String |
| **Example:** | PSH sequence … modifications … PSM KVPQVSTPTLVEVSR … 10[MS,MS:100xxxx,Probability Score Y,0.8]-MOD:00412 … PSM VFDEFKPLVEEPQNLIK … NA … |

### retention\_time

|  |  |
| --- | --- |
| **Description:** | The retention time of the spectrum. A ‘|’-separated list of multiple time points is allowed in case multiple spectra were combined by the search engine to make the PSM. |
| **Type:** | Double List |
| **Example:** | PSH sequence … retention\_time … PSM KVPQVSTPTLVEVSR … 10.2 … PSM VFDEFKPLVEEPQNLIK … 15.8 … |

### charge

|  |  |
| --- | --- |
| **Description:** | The charge assigned by the search engine/software. |
| **Type:** | Integer |
| **Example:** | PSH sequence … charge … PSM KVPQVSTPTLVEVSR … 2 … PSM VFDEFKPLVEEPQNLIK … 3 … |

### exp\_mass\_to\_charge

|  |  |
| --- | --- |
| **Description:** | The PSM’s experimental mass to charge (*m/z*). |
| **Type:** | Double |
| **Example:** | PSH sequence … mass\_to\_charge … PSM KVPQVSTPTLVEVSR … 1234.4 … PSM VFDEFKPLVEEPQNLIK … 123.4 … |

### calc\_mass\_to\_charge

|  |  |
| --- | --- |
| **Description:** | The PSM’s calculated (theoretical) mass to charge (*m/z*). |
| **Type:** | Double |
| **Example:** | PSH sequence … mass\_to\_charge … PSM KVPQVSTPTLVEVSR … 1234.4 … PSM VFDEFKPLVEEPQNLIK … 123.4 … |

### uri

|  |  |
| --- | --- |
| **Description:** | A URI pointing to the PSM's entry in the experiment it was identified in (e.g., the peptide’s PRIDE entry). |
| **Type:** | URI |
| **Example:** | PSH sequence … uri … PSM KVPQVSTPTLVEVSR … http://www.ebi.ac.uk/pride/link/to/peptide … PSM VFDEFKPLVEEPQNLIK … http://www.ebi.ac.uk/pride/link/to/peptide … |

### spectra\_ref

|  |  |
| --- | --- |
| **Description:** | Reference to a spectrum in a spectrum file. The reference must be in the format ms\_file[1-n]:{SPECTRA\_REF} where SPECTRA\_REF MUST follow the format defined in 5.2. Multiple spectra MUST be referenced using a “|” delimited list for the (rare) cases in which search engines have combined multiple spectra to make identifications. |
| **Type:** | String |
| **Example:** | PSH sequence … spectra\_ref … PSM KVPQVSTPTLVEVSR … ms\_file[1]:index=5 …  PSM VFDEFKPLVEEPQNLIK … ms\_file[2]:index=7|ms\_file[2]:index=9 … |

### pre

|  |  |
| --- | --- |
| **Description:** | Amino acid preceding the peptide in the protein sequence. If unknown “null” MUST be used, if the peptide is N-terminal “-“ MUST be used. |
| **Type:** | String |
| **Example:** | PSH sequence … pre post … PSM KVPQVSTPTLVEVSR … K D …  PSM VFDEFKPLVEEPQNLIK … R L … |

### post

|  |  |
| --- | --- |
| **Description:** | Amino acid following the peptide in the protein sequence. If unknown “null” MUST be used, if the peptide is C-terminal “-“ MUST be used. |
| **Type:** | String |
| **Example:** | PSH sequence … pre post … PSM KVPQVSTPTLVEVSR … K D …  PSM VFDEFKPLVEEPQNLIK … R L … |

### start

|  |  |
| --- | --- |
| **Description:** | The start position of the peptide within the protein, counting 1 as the N-terminus of the protein. |
| **Type:** | String |
| **Example:** | PSH sequence … start end … PSM KVPQVSTPTLVEVSR … 45 57 …  PSM VFDEFKPLVEEPQNLIK … 34 46 … |

### end

|  |  |
| --- | --- |
| **Description:** | The end position of the peptide within the protein, counting 1 as the N-terminus of the protein. |
| **Type:** | String |
| **Example:** | PSH sequence … start end … PSM KVPQVSTPTLVEVSR … 45 57 …  PSM VFDEFKPLVEEPQNLIK … 34 46 … |

### opt\_{ASSAY\_ID}|{STUDY\_VARIABLE\_ID}|{MS\_FILE\_ID}|“global”\_\*

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) Additional columns can be added to the end of the PSM table. These column headers MUST start with the prefix “opt\_” followed by the identifier of the object they reference: assay, study variable, MS file or “global” (if the value relates to all replicates). Column names MUST only contain the following characters: ‘A’-‘Z’, ‘a’-‘z’, ‘0’-‘9’, ‘\_’, ‘-’, ‘[’, ‘]’, and ‘:’. CV parameter accessions MAY be used for optional columns following the format: opt\_{OBJECT\_ID}\_cv\_{accession}\_{parameter name}. Spaces within the parameter’s name MUST be replaced by ‘\_’. |
| **Type:** | Column |
| **Example:** | PSH sequence … opt\_assay[1]\_my\_value opt\_global\_another\_value PSM PEPTIDER … My value about assay[1] some other value that is across reps |

## Small Molecule Section

The small molecule section is table-based. The small molecule section MUST always come after the metadata section, peptide section and or protein section if they are present in the file. All table columns MUST be Tab separated. There MUST NOT be any empty cells.

All columns, unless specified otherwise, are mandatory. The order of columns is not specified although for ease of human interpretation, it is RECOMMENDED to follow the order specified below.

### identifier

|  |  |
| --- | --- |
| **Description:** | A list of “|” separated possible identifiers for these small molecules.  The database identifier must be preceded by the resource description followed by a colon (in case this is not already part of the identifier format). |
| **Type:** | String List |
| **Example:** | SMH identifier … SML CID:00027395 … SML HMDB:HMDB12345 … |



### chemical\_formula

|  |  |
| --- | --- |
| **Description:** | The chemical formula of the identified compound.  This should be specified in Hill notation (EA Hill 1900), i.e. elements in the order C, H and then alphabetically all other elements. Counts of one may be omitted. Elements should be capitalized properly to avoid confusion (e.g., “CO” vs. “Co”). The chemical formula reported should refer to the neutral form. Charge state is reported by the charge field. This permits the comparison of positive and negative mode results.  **Example:** N-acetylglucosamine would be encoded by the string “C8H15NO6” |
| **Type:** | String |
| **Example:** | SMH identifier chemical\_formula … SML CID:00027395 C17H20N4O2 … |

### smiles

|  |  |
| --- | --- |
| **Description:** | The molecules structure in the simplified molecular-input line-entry system (SMILES). If there are more than one SMILES for a given small molecule, use the “|” separator. |
| **Type:** | String List |
| **Example:** | SMH identifier … chemical\_formula smiles … SML CID:00027395 … C17H20N4O2 C1=CC=C(C=C1)CCNC(=O)CCNNC(=O)C2=CC=NC=C2 … |

### inchi\_key

|  |  |
| --- | --- |
| **Description:** | The standard IUPAC International Chemical Identifier (InChI) Key of the given substance. If there are more than one InChI identifier for a given small molecule, use the “|” separator. |
| **Type:** | String List |
| **Example:** | SMH identifier … chemical\_formula … inchi\_key … SML CID:00027395 … C17H20N4O2 … QXBMEGUKVLFJAM-UHFFFAOYSA-N … |

### description

|  |  |
| --- | --- |
| **Description:** | The small molecule’s description / name. |
| **Type:** | String |
| **Example:** | SMH identifier … description … SML CID:00027395 … N-(2-phenylethyl)-3-[2-(pyridine-4-carbonyl)hydrazinyl]propanamide… |

### mass\_to\_charge

|  |  |
| --- | --- |
| **Description:** | The small molecule’s precursor’s mass to charge ratio. |
| **Type:** | Double |
| **Example:** | SMH identifier … mass\_to\_charge … SML CID:00027395 … 1234.5 … |

### charge

|  |  |
| --- | --- |
| **Description:** | The charge assigned by the search engine/software. |
| **Type:** | Integer |
| **Example:** | SMH identifier … charge … SML CID:00027395 … 2 … |

### retention\_time

|  |  |
| --- | --- |
| **Description:** | A ‘|’-separated list of time points. Semantics may vary. This time should refer to the small molecule’s retention time if determined or the mid point between the first and last spectrum identifying the small molecule. |
| **Type:** | Double List |
| **Example:** | SMH identifier … retention\_time … SML CID:00027395 … 10.2|11.5 … |

### taxid

|  |  |
| --- | --- |
| **Description:** | The taxonomy id coming from the NEWT taxonomy for the species (if applicable). |
| **Type:** | Integer |
| **Example:** | SMH identifier … taxid … SML CID:00027395 … null … |

### species

|  |  |
| --- | --- |
| **Description:** | The species as a human readable string (if applicable). |
| **Type:** | String |
| **Example:** | SMH identifier … species … SML CID:00027395 … null … |

### database

|  |  |
| --- | --- |
| **Description:** | Generally references the used spectral library (if applicable). |
| **Type:** | String |
| **Example:** | SMH identifier … database … SML CID:00027395 … name of used database … |

### database\_version

|  |  |
| --- | --- |
| **Description:** | Either the version of the used database if available or otherwise the date of creation.  Additionally, the number of entries in the database MAY be reported in round brackets after the version in the format: {version} ({#entries} entries), for example “2011-11 (1234 entries)”. |
| **Type:** | String |
| **Example:** | SMH identifier … database\_version … SML CID:00027395 … 2011-12-22 … |

### reliability

|  |  |
| --- | --- |
| **Description:** | The reliability of the given small molecule identification. This must be supplied by the resource and has to be one of the following values:  1: high reliability  2: medium reliability  3: poor reliability  Important: An identification's reliability is resource dependent. |
| **Type:** | Integer |
| **Example:** | SMH identifier … reliability … SML CID:00027395 … 3 … |

### uri

|  |  |
| --- | --- |
| **Description:** | A URI pointing to the small molecule’s entry in the experiment it was identified in (e.g., the small molecule’s PRIDE entry). |
| **Type:** | URI |
| **Example:** | SMH identifier … uri … SML CID:00027395 … http://www.ebi.ac.uk/pride/link/to/identification … |

### spectra\_ref

|  |  |
| --- | --- |
| **Description:** | Reference to a spectrum in a spectrum file. The reference must be in the format ms\_file[1-n]:{SPECTRA\_REF} where spectra\_ref MUST follow the format defined in 5.2. Multiple spectra can be referenced using a “|” delimited list. If “spectra\_ref” is not ‘null’, then the element “ms\_file[1-n]-location” MUST be reported in the metadata section. |
| **Type:** | String |
| **Example:** | SMH identifier … spectra\_ref … SML CID:00027395 … ms\_file[1]:index=1002 … |

### search\_engine

|  |  |
| --- | --- |
| **Description:** | A “|” delimited list of search engine(s) used to identify this small molecule. Search engines must be supplied as parameters. |
| **Type:** | Parameter List |
| **Example:** | SMH identifier … search\_engine … SML CID:00027395 … [MS, MS:1001477, SpectraST,] … |

### best\_search\_engine\_score

|  |  |
| --- | --- |
| **Description:** | A “|” delimited list of best search engine score(s) across replicates for the given small molecule. Scores SHOULD be reported using CV parameters whenever possible. |
| **Type:** | Parameter List |
| **Example:** | SMH identifier … search\_engine\_score … SML CID:00027395 … [MS, MS:1001419, SpectraST:discriminant score F, 0.7] … |

### search\_engine\_score\_ms\_file[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present)  A “|” delimited list of search engine score(s) in each MS file for the given small molecule. Scores SHOULD be reported using CV parameters whenever possible. |
| **Type:** | Parameter List |
| **Example:** | SMH identifier … search\_engine\_score … SML CID:00027395 … [MS, MS:1001419, SpectraST:discriminant score F, 0.7] … |

### modifications

|  |  |
| --- | --- |
| **Description:** | The small molecule’s modifications or adducts. The position of the modification must be given relative to the small molecule’s beginning. The exact semantics of this position depends on the type of small molecule identified. In case the position information is unknown or not applicable it should not be supplied. For detailed information see protein table. |
| **Type:** | String |
| **Example:** | COM example where an ammonium loss is found and the position is not COM applicable in the given small molecule  SMH identifier … modifications … SML CID:00027395 … CHEMMOD:-NH4 …  COM reporting adducts: sodiated glycine  SMH … formula … charge … modifications  SML … C2H5NO2 … 1 … CHEMMOD:+Na-H |

### smallmolecule\_abundance\_assay[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The small molecule’s abundance in the given assays. |
| **Type:** | Double |
| **Example:** | SMH identifier … smallmolecule\_abundance\_sub[1] … SML CID:00027395 … 0.3 … |

### smallmolecule\_abundance\_study\_variable[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The small molecule’s abundance in the given study variables. |
| **Type:** | Double |
| **Example:** | SMH identifier … smallmolecule\_abundance\_sub[1] … SML CID:00027395 … 0.3 … |

### smallmolecule \_abundance\_stdev\_ study\_variable [1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The standard deviation of the small molecule’s abundance in the given study variable. In case the abundance for a respective study variable is given the standard deviation column MUST also be present (in case the value is not available “null” MUST be used). |
| **Type:** | Double |
| **Example:** | SMH identifier … smallmolecule\_abundance\_sub[1] smallmolecule\_abundance\_stdev\_sub[1]… SML CID:00027395… 0.3 0.04 … |

### smallmolecule \_abundance\_std\_error\_study\_variable[1-n] (Optional)

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) The standard error of the small molecule’s abundance in the given study variable. In case the abundance for a respective study variable is given the standard error column MUST also be present (in case the value is not available “null” MUST be used). |
| **Type:** | Double |
| **Example:** | SMH identifier … smallmolecule\_abundance\_std\_error\_sub[1] …  SML CID:00027395 … 0.04 … |

### opt\_{ASSAY\_ID}|{STUDY\_VARIABLE\_ID}|{MS\_FILE\_ID}|“global”\_\*

|  |  |
| --- | --- |
| **Description:** | **Optional** (this column MAY be present) Additional columns can be added to the end of the small molecule table. These column headers MUST start with the prefix “opt\_” followed by the identifier of the object they reference: assay, study variable, MS file or “global” (if the value relates to all replicates). Column names MUST only contain the following characters: ‘A’-‘Z’, ‘a’-‘z’, ‘0’-‘9’, ‘\_’, ‘-’, ‘[’, ‘]’, and ‘:’. CV parameter accessions MAY be used for optional columns following the format: opt\_{OBJECT\_ID}\_cv\_{accession}\_{parameter name}. Spaces within the parameter’s name MUST be replaced by ‘\_’. |
| **Type:** | Column |
| **Example:** | SMH identifier … opt\_assay[1]\_my\_value opt\_global\_another\_value SML CID:00027395 … My value some other value |

# Non-supported use cases

There are a number of use cases that were discussed during the development process and it was decided that they are not explicitly supported in mzTab version 1.0. They may be implemented in future versions of the standard.

- Sequence Tag approaches.

- Grouped modification position scoring systems.

# Conclusions

This document contains the specifications for using the mzTab format to represent results from peptide, small molecule and protein identification pipelines, in the context of a proteomics investigation. This specification constitutes a proposal for a standard from the Proteomics Standards Initiative. These artefacts are currently undergoing the PSI document process, which will result in a standard officially sanctioned by PSI.

# Authors

Johannes Griss, European Bioinformatics Institute, United Kingdom

Timo Sachsenberg, University of Tübingen, Germany

Mathias Walzer, Center for Bioinformatics, University of Tübingen, Germany

Oliver Kohlbacher, Center for Bioinformatics, University of Tübingen, Germany

Andrew R. Jones, University of Liverpool, United Kingdom

Henning Hermjakob, European Bioinformatics Institute, United Kingdom

Juan Antonio Vizcaíno, European Bioinformatics Institute, United Kingdom

Correspondence – Henning Hermjakob ([hhe@ebi.ac.uk](mailto:juan@ebi.ac.uk))

# Contributors

In addition to the authors, the following people contributed to the model development, gave feedback or tested mzTab:

* Gerhard Thallinger, Graz University of Technology, Austria.
* Jürgen Hartler, Graz University of Technology, Austria.
* Martin Eisenacher, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Germany.
* Gerhard Mayer, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Germany.
* Nuno Bandeira, Center for Computational Mass Spectrometry, University of California, San Diego, CA, USA.
* Robert J. Chalkley, Department of Pharmaceutical Chemistry, University of California San Francisco, CA, USA.
* Laurent Gatto, University of Cambridge, Cambridge, United Kingdom.
* Florian Reisinger, European Bioinformatics Institute, United Kingdom.
* Ioannis Xenarios, Swiss Institute of Bioinformatics, Geneva, Switzerland.
* Qing-Wei Xu, European Bioinformatics Institute, United Kingdom.

# References

* Bradner, S. (1997). Key words for use in RFCs to Indicate Requirement Levels, Internet Engineering Task Force. RFC 2119.
* Martens, L., et al. (2011). "mzML--a community standard for mass spectrometry data." Mol Cell Proteomics 10(1): R110 000133.
* Jones, A. R., et al. (2012). "The mzIdentML data standard for mass spectrometry-based proteomics results." Mol Cell Proteomics doi:10.1074/mcp.M111.014381
* EA Hill (1900). “ON A SYSTEM OF INDEXING CHEMICAL LITERATURE; ADOPTED BY THE CLASSIFICATION DIVISION OF THE U. S. PATENT OFFICE.” J. Am. Chem. Soc. 22 (8): 478–494. doi:10.1021/ja02046a005
* Walzer at al. The mzQuantML data standard for mass spectrometry-based quantitative studies in proteomics (2013) Mol Cell Proteomics doi: 10.1074 mcp.O113.028506.

# 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 that 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).

# TradeMark Section

Microsoft Excel®

# Copyright Notice

Copyright (C) Proteomics Standards Initiative (2013). 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."