

Johannes Griss, European Bioinformatics Institute  
Timo Sachsenberg, University of Tübingen  
Mathias Walzer, University of Tübingen  
Oliver Kohlbacher, University of Tübingen  
Andrew R. Jones, University of Liverpool  
Henning Hermjakob, European Bioinformatics Institute  
Juan Antonio Vizcaíno, European Bioinformatics Institute

April 2012

mzTab: exchange format for proteomics and metabolomics results

Kommentar [JAV1]: Add new authors?

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 0.9.2, release candidate April 2012.

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

1. Introduction ..... 2

    1.1 Background ..... 2

    1.2 Document Structure ..... 3

2. Use Cases for mzTab..... 3

3. Concepts and Terminology ..... 4

4. Relationship to Other Specifications..... 4

    4.1 The PSI Mass Spectrometry Controlled Vocabulary (CV) ..... 5

5. Resolved Design and scope issues..... 6

    5.1 Handling updates to the controlled vocabulary..... 6

    5.2 Use of identifiers for input spectra to a search ..... 6

    5.3 Recommendations for reporting protein inference..... 7

    5.4 Recommendations for reporting replicates within experimental designs ..... 8

    5.5 Recommendations for reporting quantification results..... 8

    5.6 Encoding missing values, zeroes, nulls, infinity and calculation errors..... 9

    5.7 Number of unique peptides reported ..... 10

    5.8 Reliability score ..... 10

    5.9 Reporting modifications..... 11

5.10 Comments on Specific Use Cases.....	12
5.11 Other supporting materials.....	13
6. Format specification .....	13
6.1 Sections .....	<u>1544</u>
6.2 Metadata Section .....	15
6.3 Protein Section .....	20
6.4 Peptide Section .....	<u>2524</u>
6.5 Small Molecule Section .....	<u>2928</u>
7. Conclusions.....	<u>3332</u>
8. Authors and Contributors .....	<u>3332</u>
9. References.....	<u>3332</u>
10. Intellectual Property Statement .....	<u>3432</u>
Copyright Notice .....	<u>3433</u>

## 1. Introduction

### 1.1 Background

This document addresses the systematic description of peptide, protein, and small molecule identification and quantification data retrieved from a mass spectrometry-based experiment. 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 2):

- T1. *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.

- T2. *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 guideline the file format is designed to be viewable by programs such as Microsoft Excel and Open Office Spreadsheet.
- T3. *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.
- T4. *Allow the concatenation of results*, and thus being able to combine results from multiple experiments but also multiple entries from local LIMS databases or MS proteomics repositories.
- T5. *Act as an output format of (web-) services* that report MS-based results and thus can produce standardized result pages.
- T6. *Allow the combination of MS-based proteomics and metabolomics experimental results* within a single file.
- T7. *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.

## 1.2 Document Structure

The remainder of this document is structured as follows. Section 2 lists use cases mzTab is designed to support. Section 3 describes the terminology used. Section 4 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 5 discusses the reasoning behind several design decisions taken. Section 6 contains the documentation of the file. Conclusions are presented in Section 7.

## 2. Use Cases for mzTab

The following use cases have driven the development of the mzTab data model, and are used to define the scope of the format in version 1.0.0.

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. However, these files will not contain the complete evidence required to replicate the performed experiment or even provide information about all details of the reported identifications.
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 post-translational modification (PTM) 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.

### 3. Concepts and Terminology

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).

### 4. Relationship to Other Specifications

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

1. *mzML* (<http://www.psdev.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 proposed PSI standard for capturing quantitative proteomics data from mass spectrometry. *mzTab* files that report quantitative data MAY reference *mzQuantML* files for detailed evidence of the reported values.

#### 4.1 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 by 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 (via the form <http://www.psidev.info/index.php?q=node/440>) 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.sourceforge.net/mod/data/PSI-MOD.obo>
- Unimod modifications database - <http://www.unimod.org/obo/unimod.obo>
- PRIDE ontology ([http://ebi-pride.googlecode.com/svn/trunk/pride-core/schema/pride\\_cv.obo](http://ebi-pride.googlecode.com/svn/trunk/pride-core/schema/pride_cv.obo))

## 5. 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.

### 5.1 Handling updates to the controlled vocabulary

There is a difficult issue with respect to how software should encode CV terms, such that changes to 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 form <http://www.psdev.info/index.php?q=node/440>) 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.

### 5.2 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 spec\_ref attribute and via the ms\_file element which stores the URL of the file in the location attribute. 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 spec\_ref attribute should be constructed following the data type specification in Table 1. As an example, to reference the third spectrum (index = 2) in an MGF (Mascot Generic Format) file:

```

MTD UNIT_1-ms_file[1]-format [MS, MS:1001062, Mascot MGF file, ]
MTD UNIT_1-ms_file[1]-id_format [MS, MS:1000774, multiple peak list nativeID format, ]

...

PEH sequence ... spec_ref ...
PEP NILNELFQR ... ms_file[1]:index=2 ...

```

### 5.3 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 down-stream 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 could also be identified through the same (sub-)set of peptides but were not chosen as the primary identification. The members of the ambiguity group are not reported in the peptide table for the respective unit.

```

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 unit_id ... ambiguity_members ...
PRT P19012 EXP_1 P13646, P08779, P02533, Q7Z3Z0, Q7Z3Y9, Q7Z3Y8 ...

...
PEH sequence accession unit_id ...
PEP ALEENADLEVK P19012 EXP_1 ...

```



## 5.4 Recommendations for reporting replicates within experimental designs

Modeling the correct reporting of technical/biological replicates within experimental designs is inherently complex. mzQuantML supports the detailed reporting of such results in respect to quantitative data. mzTab is designed to be a simple data format. Therefore, the reporting of results from such experimental designs is poorly supported in mzTab.

A UNIT in an mzTab file can be any entity in which a protein is unambiguously identified by its accession (see below). For instance, a UNIT can be the overall result of an experiment after the data from the corresponding technical and/or biological replicates were processed. However, technical replicates MAY also be reported in a single mzTab file as separate UNITS. When reporting technical replicates, for example for an experiment “EXP\_1”, the replicates must have the UNIT\_IDs “EXP\_1-rep[1-n]”. Additionally, the overall results, *i.e.* the combined analysis of all technical replicates MAY be reported using the experiment prefix as UNIT\_ID:

```
COM Example highlighting the reporting of three replicates as well as the overall result
MTD EXP_1-title The overall result of experiment 1.
...
MTD EXP_1-rep[1] Replicate 1 of experiment 1.
...
MTD EXP_1-rep[2] Replicate 2 of experiment 1.
...
MTD EXP_1-rep[3] Replicate 3 of experiment 1.
```

Since every row in every section is linked to a UNIT\_ID, any entry in an mzTab file (protein, peptide, and small molecule identifications) can be unambiguously linked to the corresponding replicate or overall result.

Biological replicates are not explicitly supported in the same way in mzTab.

## 5.5 Recommendations for reporting quantification results

There are multiple quantification techniques available for MS-based experiments that often result in slightly different types of data. mzTab was explicitly not designed to capture any of these specific differences. The goal for mzTab was to provide a generic view on quantitative MS-based identification data that is applicable to as many different quantitation methods as possible.

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. When several biological samples are multiplexed these samples are referred to as “subsamples” in mzTab. Subsamples MUST furthermore be linked to the used labels in the metadata section of the mzTab file (see example below). In case a quantification method is used that does not lead to multiplexed biological samples the generated quantification values MUST be reported as subsample 1. Replicates in experimental designs are not explicitly modeled in mzTab (see section 5.4).



mzTab allows the reporting of abundance, standard deviation, and standard error for any present subsample. The unit of these values MUST be specified in the metadata section of the mzTab file. This should be used to differentiate between relative (use unit “ratio”) and absolute (use respective unit) quantitation. 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.

In MS label-based quantitation approaches the final ratios of the various tags MUST be reported as the various subsample abundances at the peptide level. The final protein abundances MUST be reported at the protein level using the same subsample labels (see example below). MS<sup>1</sup> label-based quantification results SHOULD not report a “mass\_to\_charge” ratios. “NA” SHOULD be used instead for this column.

```
COM The following example shows how two different quantitative experiments
COM can be reported in one mzTab file. Not all labels are shown
...
MTD EXP_1-quantification_method [MS,MS:1001837,iTraq,]
MTD EXP_1-sub[1]-description Healthy human liver tissue
MTD EXP_1-sub[1]-quantification_reagent [PRIDE,PRIDE:0000114,iTRAQ reagent 114,]
MTD EXP_1-sub[2]-description Human hepatocellular carcinoma sample.
MTD EXP_1-sub[2]-quantification_reagent [PRIDE,PRIDE:0000115,iTRAQ reagent 115,]
...
MTD EXP_2-quantification_method [MS,MS:100999,SILAC,]
MTD EXP_2-sub[1]-description Healthy rat liver tissue
MTD EXP_2-sub[1]-quantification_reagent [PRIDE,PRIDE:0000325,SILAC heavy,]
MTD EXP_2-sub[2]-description Intoxicated rat liver.
MTD EXP_2-sub[2]-quantification_reagent [PRIDE,PRIDE:0000326,SILAC light,]
...
PRH accession unit_id ... protein_abundance_sub[1] ... protein_abundance_sub[2] ...
PRT P12345 EXP_1 ... 1 ... 0.82749
PRT P15151 EXP_2 ... 2.42114 ... 1
...
COM All subsample columns where the SILAC based experiment has no values but the iTraq
COM base experiment has, the SILAC experiment's proteins simply contain "-".
```

MS<sup>2</sup> spectral counting-based approaches SHOULD be reported using optional columns in the peptide table as well as the protein table as they only result in one single value per analyte. In case the approach used also generates standard deviation and standard errors the quantification results MAY also be reported using the subsample 1 columns. MS label-free quantitation techniques do not require any additional support in mzTab as they simply need to report abundance values per sample in a straight-forward manner. CV parameter accessions MAY be used as optional column names following the following format: opt\_cv\_{accession}.

```
COM Example showing how emPAI values are reported in an additional column using
COM MS CV parameter "emPAI value" (MS:1001905)
...
PRH accession ... opt_cv_MS:1001905
PRT P12345 ... 0.658
```

## 5.6 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 for an entry two symbols MUST be used to indicate why this information is not available:

“-” indicates that the given information was lost during conversion or not available in the source file (for example, modifications were not reported in the input file).

**Kommentar [JAV2]:** Robert suggested that the different between 'NA' and '-' was not clear.

**Formatiert:** Standard, Keine Aufzählungen oder Nummerierungen

- ~~“NA” should be given in case the information is not applicable (for example, a protein/peptide does not contain any modifications but modifications are reported for other proteins/peptides in the experiment).-).~~

~~Any calculation that results in “not a number” (NaN type) MUST be reported using “NA” if the value could theoretically be calculated based on the available information. If a required value for the calculation was missing “ ” MUST be used. If ratios are included and the denominator is zero, the “INF” value MUST be given. 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 (“NA” as it could have been observed and “ ” if it could not have been measured).-). In case a given property is not available “NA” MUST be used. This is, for example, the case when modifications were not identified on a given peptide (ie. the table cell MUST NOT be empty but “NA” has to be reported).~~

~~If ratios are included and the denominator is zero, the “INF” value MUST be used. 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 (“NA”).~~

**Kommentar [JG3]:** Even though MITAB uses “-” I would favor to use “NA”. When opening files in MS Excel, cells with “-” are automatically interpreted as empty numbers. Therefore, if the file is saved again the “NA” disappeared. Additionally, “NA” is already used by several statistical tools including R.

## 5.7 Number of unique peptides reported

The protein section contains three columns to report the number of peptides supporting a given protein identification: num\_peptides, num\_peptides\_distinct, num\_peptides\_unique (see below). The first column, num\_peptides, reports all peptides that are associated with the given protein in the respective unit irrespective of their charge, modifications, and sequence. num\_peptides\_distinct MUST report the number of distinct peptides associated with the given protein-based on their sequence and modifications. num\_peptides\_unique finally reports the subset of distinct peptides (see above) that are unique for the given protein based on the protein database used for the performed search. Detailed examples on how these numbers are generated are given in the respective protein table column descriptions (see below). These numbers only reflect peptides used for identification but not quantification. The peptides contributing to the protein’s quantification values can be retrieved from the peptide table by retrieving all peptides containing quantitative values assigned to the given proteins.

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. There will be cases where different tools generate different numbers for similar datasets. Nevertheless, too restrictive definitions of these columns might prevent the usage of mzTab for certain use cases.

## 5.8 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

**Kommentar [JAV4]:** Comment from Robert:  
I do not like this. First, what is it a reliability measure of? Identification? Identification+site assignment? Identification+quantitative measure? It seems so vague and undefined that I think it is not worthwhile as is. If it is peptide+modification ID reliability, then could the categories just be 1: above acceptance criteria; 2: below acceptance criteria? (This would require them to explain and justify their acceptance criteria – which is required for journal publication guidelines anyway)

documented by the data providers. If this information is not provided by the producers of mzTab files, “-” must be provided as the value for each of the protein, peptide or small molecule identification.

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

- 1: high reliability
- 2: medium reliability
- 3: poor reliability

**Kommentar [JAV5]:** We think it is still a good idea to keep the 3 levels, 1 and 3 could be interpreted as mentioned by Robert. 2 would be the 'border line' or difficult cases.

The idea behind this score was to mimic the general concept of “resource based trust”. For example, if [PeptideAtlas](#) reports identifications with a given reliability this would be interpreted differently as an identification reported from [GPMDB](#) – 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, for example 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.

**Kommentar [JAV6]:** Comment from Nuno:

This is a good aim but it would be good to give some guidance as to how to set these numbers so that non-mass-spec people have a sense of how to use these scores. Maybe indicate that high reliability should roughly correspond to 1% FDR, medium to 5% and poor to just the top candidate ID per spectrum with no error rate threshold (only useful for manual annotation explorations).

**Kommentar [JAV7]:** We think that some guidelines could be given, but this is independent from the file format. These guidelines could also evolve in time. More importantly, every data producer should document the criteria the followed.

## 5.9 Reporting modifications

Modifications are modelled using a specific modification object with the following format: {position}{reliability score}-{Modification identifier}. {position} and {reliability score} are optional depending on the section where the modification is reported (for details see 6.3.15). Terminal modifications in proteins and peptides MUST be reported with the position set to 0 or the amino acid length +1 respectively. This object allows modifications to be assigned to ambiguous locations. Furthermore, it is possible to report reliabilities for every potential location. All (identified) variable modifications as well as fixed modifications MUST be reported for every identification.

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. The list of allowed modification identifiers is therefore:

```
CHEMMOD:+NH4
CHEMMOD:-18.0913
UNIMOD:18
PSI-MOD:00815
```

CHEMMODs MUST 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 MUST NOT be reported if the given delta can be expressed through a known and unambiguous chemical formula.

To increase readability and make the information more accessible to people outside the field of proteomics, peptide modifications MAY be additionally defined in the peptide's sequence. Two approaches are allowed: specifying the modification name (preferably the PSI-MOD or UNIMOD name) in round brackets or specifying the *m/z* delta in square brackets after the

affected amino acid. When using this approach, N-terminal modifications SHOULD be reported before the first amino acid. The ambiguity when reporting C-terminal modifications can only be resolved through the detailed information in the “modifications” column.

```
PEH sequence ... modifications ...
PEP EIEILAC (Carbamidomethyl) EIR ... 7-UNIMOD:4 ...
PEP EIEILAC[57.021464]EIR ... 7-UNIMOD:4 ...
```

Irrespective of whether this approach is used or not the modifications MUST always be specified in the “modifications” column (see below). Therefore, a parser SHOULD ignore any values in a peptide’s sequence within ‘[]’ and within ‘()’ and only retrieve modification data from the “modifications” column.

## 5.10 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.

### 5.10.1 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.

### 5.10.2 Merging mzTab files

All sections of an mzTab file were designed that they can easily be merged by simply concatenating the information. When merging mzTab files the rows of every section should be concatenated. For the table-based sections (proteins, peptides, and small molecules) the header row MUST only be reported once at the top of the section. Custom columns MUST NOT be included in merged files unless it was made sure that these columns report identical information.

### 5.10.3 Adding optional columns

Additional columns can be added to the end of rows in all the table-based sections (protein, peptide, and small molecule). These column headers MUST start with the prefix “opt\_”. Column names MUST only contain the following characters: ‘A’-‘Z’, ‘a’-‘z’, ‘0’-‘9’, ‘\_’, ‘-’, ‘[’, ‘]’, and ‘.’.

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\_cv\_{accession}.

```
COM Example showing how emPAI values are reported in an additional column using
COM MS CV parameter "emPAI value" (MS:1001905)
...
PRH accession ... opt_cv_MS:1001905
```

PRT P12345 ... 0.658

#### 5.10.4 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 files 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.

#### 5.11 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>

- mztab\_SILAC\_example.txt - (hand crafted) mzTab file showing how SILAC data can be reported.
- mztab\_itraq\_example.txt - (hand crafted) mzTab file showing how iTRAQ data can be reported.
- mztab\_merged\_example.txt - merged version of the example file a and b.
- PRIDE\_Exp\_Complete\_Ac\_16649.xml-mztab.txt - file generated using the mztab-exporter (converted PRIDE experiment accession 16649) containing iTRAQ data.
- mztab\_lipidomics\_example.txt – Example containing MS lipidomics data produced by the Lipid Data Analyzer tool ([http://genome.tugraz.at/lda/lda\\_download.shtml](http://genome.tugraz.at/lda/lda_download.shtml)).

## 6. 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
- SMH for small molecule table header line
- SML for rows of the small molecule table
- COM for comment lines

- **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 should be left empty.

```
[MS, MS:1001207, Mascot,]
[MS, MS:1001171, Mascot:score, 40.21]
[, , A user parameter, The value]
```

- **Unit IDs**

To link identifications to certain bits of metadata identifications are grouped into so-called "units". In a unit, a specific protein must be unambiguously identified by its accession number. For PRIDE generated data a unit would, for example, be an experiment while for PeptideAtlas generated data a unit would be a specific build. UNIT\_IDs SHOULD consist of the resource identifier plus the resources internal unit id. A resource is anything that is generating mzTab files. UNIT\_IDs do not have to be unique across multiple mzTab files. Duplication of UNIT\_IDs MUST be prevented when merging mzTab files.

UNIT\_IDs MUST only contain the following characters: 'A'-'Z', 'a'-'z', '0'-'9', and '\_'. Unit IDs MUST NOT contain the prefix "\_rep[1-n]" unless reporting technical replicate data (see 5.4).

- **Subsample IDs**

In several experimental approaches biological samples are multiplexed and analyzed in one single experiment. To be able to supply metadata specific to such a subsample, subsample ids in the format sub[1-n] are used.

```
MTD PRIDE_1234-sub[1]-species [NEWT, 9606, Homo sapiens (Human), ]
```

## 6.1 Sections

mzTab files can contain four different sections. All of these sections are optional. The metadata section is made up of key-value pairs. The other three sections, protein, peptide, and small molecule section are table-based.

Every section in an mzTab file **MUST** only occur once if present. Therefore, proteins, for example, from multiple UNITS (*i.e.* experiments) **MUST** be reported in the same protein section / table.

To group information between the four different sections together, every entry contains a UNIT\_ID. Units in mzTab files are a loose concept and will change with the circumstances under which the file was generated. In the PRIDE repository, a UNIT will, for example, represent one experiment. In PeptideAtlas a UNIT might represent a whole PeptideAtlas build. The only requirement is, that within a given UNIT a protein identification **MUST** be unambiguously identified by its accession.

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

```
MTD PRIDE_1234-species [NEWT, 9606, Homo sapiens (Human),]
MTD PRIDE_1234-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. For example, title **MAY** only be specified once per unit.

### 6.2.1 {UNIT\_ID}-title

<b>Description:</b>	The unit's human readable title.
<b>Type:</b>	String
<b>Multiplicity:</b>	0 .. 1
<b>Example:</b>	MTD PRIDE_1234-title My first test experiment

### 6.2.2 {UNIT\_ID}-description

<b>Description:</b>	The unit's human readable description.
<b>Type:</b>	String
<b>Multiplicity:</b>	0 .. 1
<b>Example:</b>	MTD PRIDE_1234-description An experiment investigating the effects of Il-6.

### 6.2.3 {UNIT\_ID}-sample\_processing[1-n]

<b>Description:</b>	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 should be separated by a " ".
<b>Type:</b>	Parameter List
<b>Multiplicity:</b>	0 .. *



<b>Example:</b>	<pre>MTD PRIDE_1234-sample_processing[1] [SEP, SEP:00173, SDS PAGE,] MTD PRIDE_1234-sample_processing[2] [SEP, SEP:00142, enzyme digestion,][MS, ...  MS:1001251, Trypsin, ]</pre>
-----------------	--

#### 6.2.4 {UNIT\_ID}-instrument[1-n]-source

<b>Description:</b>	The instrument's source used in the experiment. Multiple instruments are numbered 1..n
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-instrument[1]-source [MS, MS:1000073, ESI,] ... MTD PRIDE_1234-instrument[2]-source [MS, MS:1000598, ETD,]</pre>

#### 6.2.5 {UNIT\_ID}-instrument[1-n]-analyzer

<b>Description:</b>	The instrument's analyzer type used in the experiment. Multiple instruments are enumerated 1..n
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-instrument[1]-analyzer [MS, MS:1000291, linear ion trap,] ... MTD PRIDE_1234-instrument[2]-analyzer [MS, MS:1000484, orbitrap,]</pre>

#### 6.2.6 {UNIT\_ID}-instrument[1-n]-detector

<b>Description:</b>	The instrument's detector type used in the experiment. Multiple instruments are numbered 1..n
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-instrument[1]-detector [MS, MS:1000253, electron multiplier,] ... MTD PRIDE_1234-instrument[2]-analyzer [MS, MS:1000348, focal plane collector,]</pre>

#### 6.2.7 {UNIT\_ID}-software[1-n]

<b>Description:</b>	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.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-software[1] [MS, MS:1001207, Mascot, 2.3] MTD PRIDE_1234-software[2] [MS, MS:1001561, Scaffold, 1.0]</pre>

#### 6.2.8 {UNIT\_ID}-false\_discovery\_rate

<b>Description:</b>	The unit's false discovery rate(s) reported at the peptide and/or protein level. Multiple parameters MUST be separated by " ".
<b>Type:</b>	Parameter List
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-false_discovery_rate [MS, MS:1001364, pep:global FDR, 0.01] ...  [MS, MS:1001214, prot:global FDR, 0.08]</pre>

### 6.2.9 {UNIT\_ID}-publication

<b>Description:</b>	A publication on this unit. PubMed ids must be prefixed by "pubmed:", DOIs by "doi:". Multiple identifiers MUST be separated by " ".
<b>Type:</b>	String List
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	MTD PRIDE_1234-publication pubmed:21063943 doi:10.1007/978-1-60761-987-1_6 MTD PRIDE_1234-publication pubmed:20615486 doi:10.1016/j.jprot.2010.06.008

### 6.2.10 {UNIT\_ID}-contact[1-n]-name

<b>Description:</b>	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).
<b>Type:</b>	String
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	MTD PRIDE_1234-contact[1]-name James D. Watson ... MTD PRIDE_1234-contact[2]-name Francis Crick

### 6.2.11 {UNIT\_ID}-contact[1-n]-affiliation

<b>Description:</b>	The contact's affiliation.
<b>Type:</b>	String
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	MTD PRIDE_1234-contact[1]-affiliation Cambridge University, UK MTD PRIDE_1234-contact[2]-affiliation Cambridge University, UK

### 6.2.12 {UNIT\_ID}-contact[1-n]-email

<b>Description:</b>	The contact's e-mail address.
<b>Type:</b>	String
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	MTD PRIDE_1234-contact[1]-email watson@cam.ac.uk ... MTD PRIDE_1234-contact[2]-email <a href="mailto:crick@cam.ac.uk">crick@cam.ac.uk</a>

### 6.2.13 {UNIT\_ID}-uri

<b>Description:</b>	A URI pointing to the unit's source data (e.g., a PRIDE experiment or a PeptideAtlas built)
<b>Type:</b>	URI
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	MTD PRIDE_1234-uri <a href="http://www.ebi.ac.uk/pride/url/to/experiment">http://www.ebi.ac.uk/pride/url/to/experiment</a>

### 6.2.14 {UNIT\_ID}-mod

<b>Description:</b>	A list of " " separated parameters describing all (distinct) PTMs reported in this unit.
<b>Type:</b>	Parameter List
<b>Multiplicity:</b>	0 .. 1
<b>Example:</b>	MTD PRIDE_1234-mod [MOD, MOD:00397, iodoacetamide derivatized residue, ] ... [MOD, MOD:00675, oxidized residue, ]

### 6.2.15 {UNIT\_ID}-mod-probability\_method

<b>Description:</b>	Method used to report modification (position) probabilities.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. 1
<b>Example:</b>	COM As no CV parameters exist to describe such methods, a user parameter was used COM for this example MTD PRIDE_1234-mod-probability_method [,,Ascore,]

### 6.2.16 {UNIT\_ID}-quantification\_method

<b>Description:</b>	The quantification method used in this unit (most of the times experiment) (if any).
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. 1
<b>Example:</b>	MTD PRIDE_1234-quantification_method [MS, MS:1001837, iTraq, ]

### 6.2.17 {UNIT\_ID}-protein-quantification\_unit

<b>Description:</b>	Defines what type of units is reported in the protein quantification fields.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. 1
<b>Example:</b>	MTD PRIDE_1234-protein-quantification_unit [PRIDE, PRIDE:0000395, Ratio, ]

### 6.2.18 {UNIT\_ID}-peptide-quantification\_unit

<b>Description:</b>	Defines what type of units is reported in the peptide quantification fields.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. 1
<b>Example:</b>	MTD PRIDE_1234-peptide-quantification_unit [PRIDE, PRIDE:0000395, Ratio, ]

### 6.2.19 {UNIT\_ID}-ms\_file[1-n]-format

<b>Description:</b>	A parameter specifying the data format of the external MS data file.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	MTD PRIDE_1234-ms_file[1]-format [MS, MS:1000584, mzML file, ] ... MTD PRIDE_1234-ms_file[2]-format [MS, MS:1001062, Mascot MGF file, ]

### 6.2.20 {UNIT\_ID}-ms\_file[1-n]-location

<b>Description:</b>	Location of the external data file.
<b>Type:</b>	<a href="#">URI</a>
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	MTD PRIDE_1234-ms_file[1]-location file://C:\path\to\my\file ... MTD PRIDE_1234-ms_file[2]-location ftp://ftp.ebi.ac.uk/path/to/file

### 6.2.21 {UNIT\_ID}-ms\_file[1-n]-id\_format

<b>Description:</b>	Parameter specifying the id format used in the external data file.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *

<b>Example:</b>	<pre>MTD PRIDE_1234-ms_file[1]-id_format [MS, MS:1001530, mzML unique identifier, ] ... MTD PRIDE_1234-ms_file[2]-id_format [MS, MS:1000774, multiple peak list ...                                      nativeID format, ]</pre>
-----------------	---

### 6.2.22 {UNIT\_ID}-custom

<b>Description:</b>	Any additional parameters describing the unit.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-custom [,,MS operator, Florian]</pre>

### 6.2.23 {UNIT\_ID}{-(SUB\_ID)}-species[1-n]

<b>Description:</b>	The respective species. Multiple species can be supplied. If there were multiple subsamples analyzed in the respective unit these species should be given using the additional “-{SUB_ID}” part. Subsample specific parameters describing one sample should all contain the same number between the brackets.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>COM Experiment where all subsamples (if any) consisted of the same two species MTD PRIDE_1234-species[1] [NEWT, 9606, Homo sapiens (Human), ] MTD PRIDE_1234-species[2] [NEWT, 12059, Rhinovirus, ]  COM Experiment where different samples from different species (combinations) COM where pooled in one single MS analysis.  MTD PRIDE_1235-sub[1]-species[1] [NEWT, 9606, Homo sapiens (Human), ] MTD PRIDE_1235-sub[1]-species[2] [NEWT, 573824, Human rhinovirus 1, ] MTD PRIDE_1235-sub[2]-species[1] [NEWT, 9606, Homo sapiens (Human), ] MTD PRIDE_1235-sub[2]-species[2] [NEWT, 12130, Human rhinovirus 2, ]</pre>

### 6.2.24 {UNIT\_ID}{-(SUB\_ID)}-tissue[1-n]

<b>Description:</b>	The respective tissue. For detailed documentation see 6.2.23
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-tissue[1] [BTO, BTO:0000759, liver, ]</pre>

### 6.2.25 {UNIT\_ID}{-(SUB\_ID)}-cell\_type[1-n]

<b>Description:</b>	The respective cell type. For detailed documentation see 6.2.23
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-cell_type[1] [CL, CL:0000182, hepatocyte, ]</pre>

### 6.2.26 {UNIT\_ID}{-(SUB\_ID)}-disease[1-n]

<b>Description:</b>	The respective disease. For detailed documentation see 6.2.23
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-disease[1] [DOID, DOID:684, hepatocellular carcinoma, ] MTD PRIDE_1234-disease[2] [DOID, DOID:9451, alcoholic fatty liver, ]</pre>

### 6.2.27 {UNIT\_ID}-{SUB\_ID}-description

<b>Description:</b>	A human readable description of the subsample.
<b>Type:</b>	String
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-sub[1]-description Hepatocellular carcinoma samples. MTD PRIDE_1234-sub[2]-description Healthy control samples.</pre>

### 6.2.28 {UNIT\_ID}-{SUB\_ID}-quantification\_reagent

<b>Description:</b>	The reagent used to label the given subsample.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-sub[1]-reagent [PRIDE,PRIDE:0000114,iTRAQ reagent,114] MTD PRIDE_1234-sub[2]-reagent [PRIDE,PRIDE:0000115,iTRAQ reagent,115]</pre>

### 6.2.29 {UNIT\_ID}-{SUB\_ID}-custom

<b>Description:</b>	Parameters describing the subsample's additional properties.
<b>Type:</b>	Parameter
<b>Multiplicity:</b>	0 .. *
<b>Example:</b>	<pre>MTD PRIDE_1234-sub[1]-custom [,,Extraction date, 2011-12-21] MTD PRIDE_1234-sub[1]-custom [,,Extraction reason, liver biopsy]</pre>

## 6.3 Protein Section

The protein section is table-based. The protein section must always come after the metadata section if the metadata section is present in the file. All table columns MUST be tab-separated. There MUST NOT be any empty cells.

The columns in the protein section MUST be in the order they are presented in this document.

### 6.3.1 accession

<b>Description:</b>	The accession of the protein in the source database. A protein accession MUST be unique within one UNIT. Together with the UNIT_ID the accession creates a unique identifier within a given mzTab file.
<b>Type:</b>	String
<b>Example:</b>	<pre>PRH accession ... PRT P12345 ... PRT P12346 ...</pre>

### 6.3.2 unit\_id

<b>Description:</b>	The unit the protein comes from. See section 6 for detailed information on how the unit_id should be generated.
<b>Type:</b>	String
<b>Example:</b>	<pre>PRH accession unit_id ... PRT P12345 PRIDE_1234 ... PRT P12346 PRIDE_1234 ...</pre>

### 6.3.3 description

<b>Description:</b>	The protein's name and or description line.
---------------------	---

<b>Type:</b>	String
<b>Example:</b>	<pre> PRH  accession  unit_id  description  ... PRT  P12345     PRIDE_1234  Aspartate aminotransferase, mitochondrial  ... PRT  P12346     PRIDE_1234  Serotransferrin  ... </pre>

#### 6.3.4 taxid

<b>Description:</b>	The NCBI/NEWT taxonomy id for the species the protein was identified in.
<b>Type:</b>	Integer
<b>Example:</b>	<pre> PRH  accession  ...  taxid  ... PRT  P12345     ...  10116  ... PRT  P12346     ...  10116  ... </pre>

#### 6.3.5 species

<b>Description:</b>	The human readable species the protein was identified in (this should be the NCBI entry's name).
<b>Type:</b>	String
<b>Example:</b>	<pre> PRH  accession  ...  taxid  species  ... PRT  P12345     ...  10116  Rattus norvegicus (Rat)  ... PRT  P12346     ...  10116  Rattus norvegicus (Rat)  ... </pre>

#### 6.3.6 database

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

#### 6.3.7 database\_version

<b>Description:</b>	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. <u>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)".</u>
<b>Type:</b>	String
<b>Example:</b>	<pre> 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  ... </pre>

**Kommentar [JAV8]:** This has been incorporated to the format following a suggestion made by Robert.

#### 6.3.8 search\_engine

<b>Description:</b>	A "[" delimited list of search engine(s) used to identify this protein. Search engines MUST be supplied as parameters.
<b>Type:</b>	Parameter List
<b>Example:</b>	<pre> 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,]  ... </pre>

### 6.3.9 search\_engine\_score

<b>Description:</b>	A "[" delimited list of search engine score(s) for the given protein. Scores SHOULD be reported using CV parameters whenever possible.
<b>Type:</b>	Parameter List
<b>Example:</b>	<pre> PRH  accession  ...  search_engine_score PRT  P12345      ...  [MS,MS:1001171,Mascot  score,50] [MS,MS:1001155,Sequest:xcorr,2] ... PRT  P12346      ...  [MS,MS:1001171,Mascot  score,47.2] ... </pre>

### 6.3.10 reliability

<b>Description:</b>	<p>The reliability of the given protein identification. This must be supplied by the resource and has to be one of the following values:</p> <ul style="list-style-type: none"> <li>1: high reliability</li> <li>2: medium reliability</li> <li>3: poor reliability</li> </ul> <p>Important: An identification's reliability is resource-dependent.</p>
<b>Type:</b>	Integer
<b>Example:</b>	<pre> PRH  accession  ...  reliability  ... PRT  P12345      ...  3           ... PRT  P12346      ...  1           ... </pre>

### 6.3.11 num\_peptides

<b>Description:</b>	The total number of peptides identifying this protein.
<b>Type:</b>	Integer
<b>Example:</b>	<pre> COM  P12345 is identified through ABCM, ABCM+Oxidation, CDE, CDE ... PRH  accession  ...  num_peptides  ... PRT  P12345      ...  4             ... </pre>

### 6.3.12 num\_peptides\_distinct

<b>Description:</b>	The number of distinct peptides identifying this protein. Distinct peptides are defined based on their sequence + modifications.
<b>Type:</b>	Integer
<b>Example:</b>	<pre> COM  P12345 is identified through ABCM, ABCM+Oxidation, CDE, CDE ... PRH  accession  ...  num_peptides_distinct  ... PRT  P12345      ...  3                     ... </pre>

### 6.3.13 num\_peptides\_unambiguous

<b>Description:</b>	The number of unambiguous distinct (only fitting this protein in the used search database) peptides identifying this protein.
<b>Type:</b>	Integer
<b>Example:</b>	<pre> 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_unambiguous  ... PRT  P12345      ...  2                       ... </pre>

### 6.3.14 ambiguity\_members

<b>Description:</b>	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 but were not chosen by the researcher or resource. The members of the ambiguity group are not reported in the protein table for the respective unit. The exact semantics of how the ambiguity members were defined depends on the resource.
<b>Type:</b>	String List
<b>Example:</b>	COM P12345, P12347, and P12348 can all be identified through the same peptides ... PRH accession ... ambiguity_members ... PRT P12345 ... P12347,P12348 ...

### 6.3.15 modifications

<b>Description:</b>	<p>A comma delimited list of modifications found in the given protein. Modifications have to be reported in the following format: {position in protein}{reliability score}{Modification identifier}</p> <p>Modification location probabilities / abundances can optionally be supplied and MUST be represented as a Double between 0-1. In case the position of the modification is uncertain multiple positions can be supplied delimited by a " ". Furthermore, in case a position is unknown no position information MAY be supplied.</p> <p>Terminal modifications <u>MUST-SHOULD</u> be reported at position 0 or protein size + 1 respectively.</p> <p>Valid modification identifiers are either PSI-MOD or UNIMOD accession (including the "MOD:" / "UNIMOD:" prefix) or <b>CHEMMODS</b>. CHEMMODS have the format CHEMMOD:±{chemical formula or <i>m/z</i> 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.</p>
<b>Type:</b>	String
<b>Example:</b>	<p>COM P12345 with MOD:00412@10 probability = 0.8</p> <p>COM P12345 with MOD:00412@140 OR 145 with associated probabilities</p> <p>... PRH accession ... modifications PRT P12345 ... [10[0.8]-MOD:00412,140[0.2] 145[0.3]-MOD:00412] ...</p>
	<p><u>Support for identification of cross-linked peptides could be added by using PEPMOD instead of MOD and allowing for extra localization information to localize the linkage site on the PEPMOD sequence. A possible syntax for this could be something like PEPMOD:&lt;peptide sequence&gt;@&lt;position in protein&gt;{reliability score}-{Modification identifier}, where the last three fields are similar to the 'normal' PTMs case and would represent position in protein = location of the linker on the second peptide, reliability score = localization certainty on the second peptide and Modification identifier = ID of the cross-linker reagent or post-translational modification (e.g., disulphide bridge).</u></p> <p><u>Support for identification of lipo- and glycol-peptides could also be supported by allowing for the same identifiers as used for SMH - maybe have an option like SMHMOD here to implement <a href="#">this</a>?</u></p>

**Kommentar [NB9]:** Is the choice of term CHEMMODS motivated by historic reasons (eg the same term used in other formats) or is it new to mzTab? If it is new then it may be worth considering changing it to something like MODELTA since CHEMMODS seems to imply chemical modifications whereas in blind modification searches one rarely knows whether the new mods are chemical or enzymatic until after thorough follow-up studies.

**Kommentar [JG10]:** The name is up to us. We are open to any suggestions.

**Kommentar [JAV13]:** A

**Kommentar [NB11]:** It would be nice to also be able to assign a probability/quantity to a set of sites. Using the data from this example, a set of sites could look like "140|145[0.4]-MOD:00412". A less-elegant solution would be to technically adhere to the format and use ""140[0.0]|145[0.4]-MOD:00412" but that makes difficult to distinguish two cases: a) 100% of the peptide abundance is on some combination of two sites (140 and 145 in this example) versus b) a specific breakdown where 50% of the intensity is on one site (eg 140) and 50% on the other (eg 145). This distinction is more important/clearer at the peptide/PSM level but will also affect protein-level PTM localization.

**Kommentar [JG12]:** In my opinion this is getting very complex. If there are different abundance levels (and I guess they were measured somehow) why not report them as two peptide entries with different mods? This way scenario B could be easily reported.

**Formatiert:** Block

**Kommentar [JAV14]:** Suggestion made by Nuno.

### 6.3.16 uri

<b>Description:</b>	A URI pointing to the protein's source entry in the unit it was identified in (e.g.,
---------------------	--

<http://www.psdev.info/>

	the PRIDE database or a local database / file identifier).
<b>Type:</b>	URI
<b>Example:</b>	PRT accession ... uri PRH P12345 ... http://www.ebi.ac.uk/pride/url/to/P12345 ...

### 6.3.17 go\_terms

<b>Description:</b>	A comma-delimited list of GO accessions for this protein.
<b>Type:</b>	String List
<b>Example:</b>	PRT accession ... go_terms PRH P12345 ... GO:0006457, GO:0005759, GO:0005886, GO:0004069 ...

### 6.3.18 protein\_coverage

<b>Description:</b>	A value between 0 and 1 defining the protein coverage.
<b>Type:</b>	Double
<b>Example:</b>	PRT accession ... protein_coverage ... PRH P12345 ... 0.4 ...

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

<b>Description:</b>	<b>Optional</b> (this column MAY be present) The protein's abundance in the given subsample. This information can only be interpreted when identifying the subsample's properties through the experiment id + subsample number in the metadata section of the file. The protein abundance reflects the protein's quantitative information after it was interpreted by the user as, for example when using peptide-based quantification methods like iTRAQ.
<b>Type:</b>	Double
<b>Example:</b>	PRT accession ... protein_abundance_sub[1] ... protein_abundance_sub[2] ... PRH P12345 ... 0.4 ... 0.2 ...

### 6.3.20 protein\_abundance\_stdev\_sub[1-n] (Optional)

<b>Description:</b>	<b>Optional</b> (this column MAY be present) The standard deviation of the protein's abundance. If a protein's abundance is given for a certain subsample, the corresponding standard deviation column MUST also be present (in case the value is not available "-" or "NA" should be given respectively).
<b>Type:</b>	Double
<b>Example:</b>	PRT accession ... protein_abundance_sub[1] ... protein_abundance_stdev_sub[1] ... PRH P12345 ... 0.4 ... 0.05 ...

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

<b>Description:</b>	<b>Optional</b> (this column MAY be present) The standard error of the protein's abundance. If a protein's abundance is given for a certain subsample, the corresponding standard error column MUST also be present (in case the value is not available "-" or "NA" should be given respectively).
<b>Type:</b>	Double
<b>Example:</b>	PRT accession ... protein_abundance_sub[1] ... protein_abundance_std_error_sub[1] ...

	PRH	P12345	...	0.4	...	0.03	...
--	-----	--------	-----	-----	-----	------	-----

### 6.3.22 opt\_\*

<b>Description:</b>	<b>Optional</b> (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_". 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_cv_{accession}.
<b>Type:</b>	Column
<b>Example:</b>	<pre>PRT  accession  ...  opt_my_value  opt_another_value PRH  P12345    ...  My value      some other value</pre>

## 6.4 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. The columns in the peptide section MUST be in the order they are presented in this document.

### 6.4.1 sequence

<b>Description:</b>	The peptide's sequence – can optionally contain modifications / delta masses in UNIMOD syntax (see section 5.9).
<b>Type:</b>	String
<b>Example:</b>	<pre>PEH  sequence  ... PEP  KVPQVSTPTLVEVSR  ... PEP  EIEILAC(Carbamidomethyl)EIR  ...</pre>

### 6.4.2 accession

<b>Description:</b>	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 "-" or "NA" respectively.
<b>Type:</b>	String
<b>Example:</b>	<pre>PEH  sequence  accession  ... PEP  KVPQVSTPTLVEVSR  P02768  ...</pre>

### 6.4.3 unit\_id

<b>Description:</b>	The unit the peptide was identified in.
<b>Type:</b>	String
<b>Example:</b>	<pre>PEH  sequence  accession  unit_id  ... PEP  KVPQVSTPTLVEVSR  P02768  PRIDE_1234  ...</pre>

### 6.4.4 unique

<b>Description:</b>	Indicates whether the peptide is unique for this protein in respect to the searched database.
<b>Type:</b>	Boolean
<b>Example:</b>	<pre>PEH  sequence  accession  unit_id  unique  ... PEP  KVPQVSTPTLVEVSR  P02768  PRIDE_1234  0  ... PEP  VFDEFKPLVEEPQNLIK  P02768  PRIDE_1234  1  ...</pre>

**Kommentar [NB15]:** Maybe this could be allowed to be "UNKNOWN:<number>" to represent a peptide in a cluster <number>? This would make sense when reporting quantitation or clustering results where the same spectra/LCMS features are found in multiple conditions/species even before an ID is assigned.

**Kommentar [JG16]:** This can potentially cause a very long discussion. Since the field of reporting clustering results is not very advanced/stable yet, it might be better to wait until this is consolidated and maybe later add a new optional section to mzTab (which wouldn't affect existing parsers) to report clustering results. We think it is not a good idea to reuse the existing protein structure for reporting clusters.

## 6.4.5 database

<b>Description:</b>	The protein database used for the search (could theoretically come from a different species) and the peptide sequence comes from. <b>6-5</b> 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)".					
<b>Type:</b>	String					
<b>Example:</b>	PEH sequence	accession	unit_id	unique	database	...
	PEP KVPQVSTPTLVEVSR	P02768	PRIDE_1234	0	UniProtKB	...
	PEP VFDEFKPLVEEPQNLIK	P02768	PRIDE_1234	1	UniProtKB	...

**Formatiert:** Block, Keine, Abstand  
Vor: 0 pt, Nach: 0 pt, Keine  
Aufzählungen oder Nummerierungen,  
Vom nächsten Absatz trennen

**Kommentar [JAV17]:** This has been incorporated to the format following a suggestion made by Robert.

**Kommentar [NB18]:** How should one specify a combination of sequence databases? What if a peptide sequence is from a combined database of say, human+influenza and the peptide occurs in both databases?

**Kommentar [JG19]:** In that case you have to decide to which protein the peptide should be assigned to. Then this protein's database should be reported here. If your report multiple protein databases you also have to allow protein assignment ambiguity.

**Kommentar [JAV20]:** Basically if you report one peptide, one peptide-protein mapping is allowed to be reported as the representative protein. Otherwise, they should be considered as ambiguity members.

## 6-5-16-4.6 database\_version

<b>Description:</b>	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.					
<b>Type:</b>	String					
<b>Example:</b>	PEH sequence	accession	unit_id	unique	database	database_version ...
	PEP KVPQVSTPTLVEVSR	P02768	PRIDE_1234	0	UniProtKB	2011_11 ...
	PEP VFDEFKPLVEEPQNLIK	P02768	PRIDE_1234	1	UniProtKB	2011_11 ...

## 6-5-26-4.7 search\_engine

<b>Description:</b>	A " " delimited list of search engine(s) used to identify this peptide. Search engines must be supplied as parameters.					
<b>Type:</b>	Parameter List					
<b>Example:</b>	PEH sequence	...	search_engine			...
	PEP KVPQVSTPTLVEVSR	...	[MS,MS:1001207,Mascot,]	[MS,MS:1001208,Sequest,]		...
	PEP VFDEFKPLVEEPQNLIK	...	[MS,MS:1001207,Mascot,]			...

## 6-5-36-4.8 search\_engine\_score

<b>Description:</b>	A " " delimited list of search engine score(s) for the given peptide. Scores SHOULD be reported using CV parameters whenever possible.					
<b>Type:</b>	Parameter List					
<b>Example:</b>	PRH sequence	...	search_engine	score		...
	PEP KVPQVSTPTLVEVSR	...	[MS,MS:1001155,Sequest:xcorr,2]			...
	PEP VFDEFKPLVEEPQNLIK	...	[MS,MS:1001171,Mascot score,47.2]			...

## 6-5-46-4.9 reliability

<b>Description:</b>	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.					
<b>Type:</b>	Integer					
<b>Example:</b>	PRH sequence	...	reliability			...
	PEP KVPQVSTPTLVEVSR	...	3			...
	PEP VFDEFKPLVEEPQNLIK	...	1			...

**6.5.56.4.10 modifications**

<b>Description:</b>	The peptide's modifications. The position of the modification MUST be given relative to the peptide's start not the protein. For detailed information see protein table.
<b>Type:</b>	String
<b>Example:</b>	<pre> PRH sequence      ... modifications      ... PEP KVPQVSTPTLVEVSR ... 10[0.8]-MOD:00412 ... PEP VFDEFKPLVEEPQNLIK ... NA              ... </pre>

**6.5.66.4.11 retention\_time**

<b>Description:</b>	A comma-separated list of time points in seconds. Semantics may vary. This time should refer to the peptide's retention time if determined or the mid point between the first and last spectrum identifying the peptide.
<b>Type:</b>	Double List
<b>Example:</b>	<pre> PRH sequence      ... retention_time      ... PEP KVPQVSTPTLVEVSR ... 10.2            ... PEP VFDEFKPLVEEPQNLIK ... 15.8            ... </pre>

**6.5.76.4.12 charge**

<b>Description:</b>	The precursor's charge. 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 "-" or "NA" should be used respectively.
<b>Type:</b>	Double
<b>Example:</b>	<pre> PRH sequence      ... charge      ... PEP KVPQVSTPTLVEVSR ... 2            ... PEP VFDEFKPLVEEPQNLIK ... 3            ... </pre>

**6.5.86.4.13 mass\_to\_charge**

<b>Description:</b>	The precursor's experimental mass to charge (m/z).
<b>Type:</b>	Double
<b>Example:</b>	<pre> PRH sequence      ... mass_to_charge      ... PEP KVPQVSTPTLVEVSR ... 1234.4            ... PEP VFDEFKPLVEEPQNLIK ... 123.4            ... </pre>

**6.5.96.4.14 uri**

<b>Description:</b>	A URI pointing to the peptide's entry in the experiment it was identified in (e.g., the peptide's PRIDE entry).
<b>Type:</b>	URI
<b>Example:</b>	<pre> PRH sequence      ... uri PEP KVPQVSTPTLVEVSR ... http://www.ebi.ac.uk/pride/link/to/peptide ... PEP VFDEFKPLVEEPQNLIK ... http://www.ebi.ac.uk/pride/link/to/peptide ... </pre>

**6.5.106.4.15 spectra\_ref**

<b>Description:</b>	Reference to a spectrum in a spectrum file. The reference must be in the format <code>ms_file[1-n]:{SPEC_REF}</code> where SPEC_REF MUST follow the format defined in mzIdentML. Multiple spectra MUST be referenced using a "[" delimited list.
<b>Type:</b>	String
<b>Example:</b>	<pre> PRH sequence      ... spec_ref      ... </pre>

PEP	KVPQVSTPTLVEVSR	...	ms_file[1]:index=5	...
PEP	VFDEFKPLVEEPQNLIK	...	ms_file[2]:index=7 ms_file[2]:index=9	...

**6.5.116.4.16 peptide\_abundance\_sub[1-n] (Optional)**

<b>Description:</b>	<b>Optional</b> (this column MAY be present) The peptide's abundance in the given subsample. This information can only be interpreted when identifying the subsample's properties through the experiment id + subsample number in the metadata section of the file.						
<b>Type:</b>	Double						
<b>Example:</b>	PRH	sequence	...	peptide_abundance_sub[1]	...	peptide_abundance_sub[2]	...
	PEP	KVPQVSTPTLVEVSR	...	0.4	...	0.2	...

**6.5.126.4.17 peptide\_abundance\_stdev\_sub[1-n] (Optional)**

<b>Description:</b>	<b>Optional</b> (this column MAY be present) The standard deviation of the peptide's abundance. In case a peptide's abundance is given for a certain subsample the corresponding standard deviation column <b>MUST</b> also be present (in case the value is not available “-“ or “NA” should be given respectively).					
<b>Type:</b>	Double					
<b>Example:</b>	PRH	sequence	...	peptide_abundance_sub[1]	peptide_abundance_stdev_sub[1]	...
	PEP	KVPQVSTPTLVEVSR	...	0.4	0.2	...

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

<b>Description:</b>	<b>Optional</b> (this column MAY be present) The standard error of the peptide's abundance. In case a peptide's abundance is given for a certain subsample the corresponding standard error column MUST also be present (in case the value is not available "-" or "NA" should be given respectively).														
<b>Type:</b>	Double														
<b>Example:</b>	<table><tr><td>PRH</td><td>sequence</td><td>...</td><td>peptide_abundance_sub[1]</td><td>...</td><td>peptide_abundance_std_error_sub[1]</td><td>...</td></tr><tr><td>PEP</td><td>KVPQVSTPTLVEVSR</td><td>...</td><td>0.4</td><td>...</td><td>0.2</td><td>...</td></tr></table>	PRH	sequence	...	peptide_abundance_sub[1]	...	peptide_abundance_std_error_sub[1]	...	PEP	KVPQVSTPTLVEVSR	...	0.4	...	0.2	...
PRH	sequence	...	peptide_abundance_sub[1]	...	peptide_abundance_std_error_sub[1]	...									
PEP	KVPQVSTPTLVEVSR	...	0.4	...	0.2	...									

**6.5.146.4.19 opt\_\***

<b>Description:</b>	<b>Optional</b> (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_”. 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_cv_{accession}.										
<b>Type:</b>	Column										
<b>Example:</b>	<table><tr><td>PRH</td><td>sequence</td><td>...</td><td>opt_my_value</td><td>opt_another_value</td></tr><tr><td>PEP</td><td>KVPQVSTPTLVEVSR</td><td>...</td><td>My value</td><td>some other value</td></tr></table>	PRH	sequence	...	opt_my_value	opt_another_value	PEP	KVPQVSTPTLVEVSR	...	My value	some other value
PRH	sequence	...	opt_my_value	opt_another_value							
PEP	KVPQVSTPTLVEVSR	...	My value	some other value							

**6.6.5 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. The columns in the small molecule section MUST be in the order they are presented in this document.

**6.6.16.5.1 identifier**

<b>Description:</b>	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).
<b>Type:</b>	String List
<b>Example:</b>	SMH identifier  ... SML CID:00027395 ... SML HMDB:HMDB12345 ...

**Kommentar [NB21]:** Maybe this could be allowed to be “UNKNOWN:<number>” to represent a small molecule in a cluster <number>?

**6.6.26.5.2 unit\_id**

<b>Description:</b>	The unit the small molecule was identified in.
<b>Type:</b>	String
<b>Example:</b>	SMH identifier unit_id ... SML CID:00027395 PRIDE_1234 ... SML HMDB:HMDB12345 PRIDE_1234 ...

**Kommentar [JAV22]:** Same thing as before

**6.6.36.5.3 chemical\_formula**

<b>Description:</b>	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.  <b>Example:</b> N-acetylglucosamine would be encoded by the string “C8H15NO6”
<b>Type:</b>	String
<b>Example:</b>	SMH identifier unit_id chemical_formula ... SML CID:00027395 PRIDE_1234 C17H20N4O2 ...

**6.5.4 smiles**

<b>Description:</b>	The molecules structure in the simplified molecular-input line-entry system (SMILES). <b>TODO: add reference for SMILES</b>
<b>Type:</b>	String
<b>Example:</b>	SMH identifier ... chemical formula smiles SML CID:00027395 ... C17H20N4O2 C1=CC=C(C=C1)CCNC(=O)CCNCC(=O)C2=CC=NC=C2 ...

**6.5.5 inchi key**

<b>Description:</b>	The standard IUPAC International Chemical Identifier (InChI) Key of the given substance.
<b>Type:</b>	String



<b>Example:</b>	SMH identifier ... chemical formula ... inchi key ... SML CID:00027395 ... C17H20N4O2 ... QXBMEGUKVLFJAM-UHFFFAOYSA-N ...
-----------------	--

**6.6.46.5.6 description**

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

**6.6.56.5.7 mass\_to\_charge**

<b>Description:</b>	The small molecule's precursor's mass to charge ratio.
<b>Type:</b>	Double
<b>Example:</b>	SMH identifier unit_id ... mass_to_charge ... SML CID:00027395 PRIDE_1234 ... 1234.5 ...

**6.6.66.5.8 charge**

<b>Description:</b>	The precursor's charge.
<b>Type:</b>	Double
<b>Example:</b>	SMH identifier unit_id ... charge ... SML CID:00027395 PRIDE_1234 ... 2 ...

**6.6.76.5.9 retention\_time**

<b>Description:</b>	A comma-separated list of time points (in seconds). 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.
<b>Type:</b>	Double List
<b>Example:</b>	SMH identifier unit_id ... retention_time ... SML CID:00027395 PRIDE_1234 ... 10.2,11.5 ...

**6.6.86.5.10 taxid**

<b>Description:</b>	The taxonomy id coming from the NEWT taxonomy for the species (if applicable).
<b>Type:</b>	Integer
<b>Example:</b>	SMH identifier unit_id ... taxid ... SML CID:00027395 PRIDE_1234 ... NA ...

**6.6.96.5.11 species**

<b>Description:</b>	The species as a human readable string (if applicable).
<b>Type:</b>	String
<b>Example:</b>	SMH identifier unit_id ... species ... SML CID:00027395 PRIDE_1234 ... NA ...

**6.6.106.5.12 database**

<b>Description:</b>	Generally references the used spectral library (if applicable). 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)".
---------------------	---

**Formatiert:** Block**Kommentar [JAV23]:** This has been incorporated to the format following a suggestion made by Robert.

<b>Type:</b>	String
<b>Example:</b>	SMH identifier unit_id ... database ... SML CID:00027395 PRIDE_1234 ... name of used database ...

**6.6.116.5.13 database\_version**

<b>Description:</b>	Either the version of the used database if available or otherwise the date of creation.
<b>Type:</b>	String
<b>Example:</b>	SMH identifier unit_id ... database_version ... SML CID:00027395 PRIDE_1234 ... 2011-12-22 ...

**6.6.126.5.14 reliability**

<b>Description:</b>	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.
<b>Type:</b>	Integer
<b>Example:</b>	SMH identifier unit_id ... reliability ... SML CID:00027395 PRIDE_1234 ... 3 ...

**6.6.136.5.15 uri**

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

**6.6.146.5.16 spectra\_ref**

<b>Description:</b>	Reference to a spectrum in a spectrum file. The reference must be in the format ms_file[1-n]:{SPEC_REF} where SPEC_REF must follow the format defined in mzIdentML. Multiple spectra can be referenced using a "[" delimited list.
<b>Type:</b>	String
<b>Example:</b>	SMH identifier ... spectra_ref ... SML CID:00027395 ... ms_file[1]:index=1002 ...

**6.6.156.5.17 search\_engine**

<b>Description:</b>	A "[" delimited list of search engine(s) used to identify this small molecule. Search engines must be supplied as parameters.
<b>Type:</b>	Parameter List
<b>Example:</b>	SMH identifier ... search_engine ... SML CID:00027395 ... [MS, MS:1001477, SpectraST,] ...

### 6.6.166.5.18 search\_engine\_score

<b>Description:</b>	A “ ” delimited list of search engine score(s) for the given small molecule. Scores SHOULD be reported using CV parameters whenever possible.
<b>Type:</b>	Parameter List
<b>Example:</b>	SMH identifier ... search_engine_score SML CID:00027395 ... [MS, MS:1001419, SpectraST:discriminant score F, 0.7] ...

### 6.6.176.5.19 modifications

<b>Description:</b>	The small molecule's modifications. 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.
<b>Type:</b>	String
<b>Example:</b>	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 ...

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

<b>Description:</b>	<b>Optional</b> (this column MAY be present) The small molecule's abundance in the given subsample. This information can only be interpreted when identifying the subsample's properties through the experiment id + subsample number in the metadata section of the file.
<b>Type:</b>	Double
<b>Example:</b>	SMH identifier ... smallmolecule_abundance_sub[1] ... SML CID:00027395 ... 0.3 ...

### 6.6.196.5.21 smallmolecule\_abundance\_stdev\_sub[1-n] (Optional)

<b>Description:</b>	<b>Optional</b> (this column MAY be present) The standard deviation of the small molecule's abundance. In case the abundance for a respective subsample is given the standard deviation column MUST also be present (in case the value is not available “-” or “NA” should be given respectively).
<b>Type:</b>	Double
<b>Example:</b>	SMH identifier ... smallmolecule_abundance_sub[1] smallmolecule_abundance_stdev_sub[1]... SML CID:00027395... 0.3 0.04 ...

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

<b>Description:</b>	<b>Optional</b> (this column MAY be present) The standard error of the small molecule's abundance. In case the abundance for a respective subsample is given the standard error column MUST also be present (in case the value is not available “-” or “NA” should be given respectively).
<b>Type:</b>	Double
<b>Example:</b>	SMH identifier ... smallmolecule_abundance_stderror_sub[1] ... SML CID:00027395 ... 0.04 ...

**6.6.216.5.23 opt\_\***

<b>Description:</b>	<b>Optional</b> (this column MAY be present) Additional columns can be added to the end of the small molecule table. These column headers <b>MUST</b> start with the prefix "opt_". Column names <b>MUST</b> only contain the following characters: 'A'-'Z', 'a'-'z', '0'-'9', '_', '-', '[', ']', and '.'. CV parameter accessions <b>MAY</b> be used for optional columns following the format: opt_cv {accession}.
<b>Type:</b>	Column
<b>Example:</b>	<pre>SMH identifier ... opt_my_value opt_another_value SML CID:00027395 ... My value some other value</pre>

**7. 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 standardization process, which will result in a standard officially sanctioned by PSI.

**8. Authors and Contributors**

**Kommentar [JAV24]:** Add new authors?

Johannes Griss, European Bioinformatics Institute  
 Timo Sachsenberg, University of Tübingen  
 Mathias Walzer, Center for Bioinformatics, University of Tübingen, Germany  
 Oliver Kohlbacher, Center for Bioinformatics, University of Tübingen, Germany  
 Andrew R. Jones, University of Liverpool  
 Henning Hermjakob, European Bioinformatics Institute  
 Juan Antonio Vizcaíno, European Bioinformatics Institute

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

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

- Gerhard Thallinger, Graz University of Technology
- Jürgen Hartler, Graz University of Technology
- Martin Eisenacher, Medizinisches Proteom-Center, Ruhr-Universität Bochum
- Gerhard Mayer, Medizinisches Proteom-Center, Ruhr-Universität Bochum

**9. 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

## 10. 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).

## Copyright Notice

Copyright (C) Proteomics Standards Initiative (2012). 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."