PSI ~~final~~ draft recommendation
Working Group

PSI Proteomics Informatics

~~David Creasy, Matrix Science~~
~~Florian Reisinger, European Bioinformatics Institute~~
~~Johannes Griss, European Bioinformatics Institute~~
~~Juan Antonio Vizcaíno, European Bioinformatics Institute~~
~~Matthew Chambers, Vanderbilt University Medical Center~~
~~Gerhard Mayer, Medizinisches Proteom-Center, Ruhr-Universität Bochum~~
~~Martin Eisenacher, Medizinisches Proteom-Center, Ruhr-Universität Bochum~~
~~Andrew R Jones, University of Liverpool~~Author list as for mzid 1.1 or additions of those contributing to protein grouping part?

~~Note: see version 1.0 specifications for the author list that contributed to that version.~~
Authors listed above worked specifically on the update to version 1.~~1~~2

~~August~~ Dec 201~~1~~3

mzIdentML: exchange format for peptides and proteins identified from mass spectra

Status of This Document

This document presents a final specification for the mzIdentML data format developed by the HUPO Proteomics Standards Initiative. Distribution is unlimited.

Version of This Document
The current version of this document is: version 1.~~1~~2~~.0~~ , draft Dec 2013~~release August 2011~~.

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 quantitation processes for proteins, peptides and protein modifications from mass spectrometry. This document defines an XML schema that can be used to describe the outputs of proteomics search engines.

Contents

1. Introduction

1.1 Background

This document addresses the systematic description of (poly)peptide identification and characterisation based upon mass spectrometry. A large number of different proteomics search engines are available that produce output in a variety of different formats. It is intended that mzIdentML will provide a common format for the export of identification results from any search engine. The format was originally developed under the name AnalysisXML as a format for several types of computational analyses performed over mass spectra in the proteomics context. It has been decided to split development into two formats: mzIdentML for peptide and protein identification (described here) and mzQuantML (to be described in a future specification document), covering quantitative proteomic data derived from MS (see Section 5.1.1).

mzIdentML has been developed with a view to supporting the following general tasks (more specific use cases are provided in Section 2):

T1. *The discovery of relevant results,* so that, for example, data sets in a database that use a particular technique or combination of techniques can be identified and studied by experimentalists during experiment design or data analysis.
T2. *The sharing of best practice*, so that, for example, analyses that have been particularly successful at identifying a certain group of peptides/proteins can be interpreted by consumers of the data.
T3. *The evaluation of results*, so that, for example, sufficient information is provided about how a particular analysis was performed to allow the results to be critically evaluated.
T4. *The sharing of data sets,* so that, for example, public repositories can import or export data, or multi-site projects can share results to support integrated analysis.
T5. *The creation of a format for input to analysis software*, for example, allowing software to be designed that provides a meta-score over the output from several search engines.
T6. *An internal format for pipeline analysis software*, for example, allowing analysis software to store intermediate results from different stages of an identification pipeline, prior to the final results being assembled in a single mzIdentML file.

The description of the analysis of proteomics mass spectra requires that models describe: (i) the identity and configuration of software used to perform the analysis and the protocol used to apply this software to the analysis; (ii) the identity of molecules; and (iii) the way in which these relate to other techniques to form a proteomics workflow. Most of this document is concerned with (i) and (ii) – the identification of the key features of different techniques that are required to support the tasks T1 to T5 above. Models of type (iii) are created by developments in the context of the Functional Genomics Experimental Object Model (FuGE), which defines model components of relevance to a wide range of experimental techniques. Several components from FuGE are re-used in the development of mzIdentML.

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 when the specification is stable.

1.2 Document Structure

The remainder of this document is structured as follows. Section 2 lists use cases mzIdentML 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 for the XML schema which is generated automatically and several parts of the schema are documented in more detail in Section 6.1~~7~~. Conclusions are presented in Section 8.

2. Use Cases for mzIdentML

The following use cases have driven the development of the mzIdentML data model and XML schema, and are used to define the scope of the format in version 1.

1. It should be possible to create a tool that loads an mzIdentML document and enables users to examine results from an MS or MSn analysis. However, there is no support for aggregating evidence from multiple MS levels. There should be sufficient information for the tool to generate output reports that conform to the requirements made by journals for publication and that conform to the relevant MIAPE guidelines. For example:
   - For a Peptide Mass Fingerprint (PMF) search, it should be possible to display the spectrum and show the matches of the peaks to the relevant peptides, but only if the spectrum is available.
   - For an MS-MS search, it should be possible to locate which spectrum matched to which peptide in the original file.
2. There should be sufficient information stored in the instance document to enable a user to run the same or a similar search on the same or another search engine. This means that all search parameters should be described in sufficient detail and that sufficient information is available to determine which database (if any) the data were searched against. The peak lists data (if any) do not need to be included in the instance document, but do need to be suitably referenced.
3. It should be possible to save the results of searching a decoy database in the same instance document as the results from the target database. It should then be possible to write a viewer application that enables a user to investigate the effect of changing, for example, a threshold value on the false discovery rate. This would only be possible if results that are generally considered lower quality from the search are also saved in the mzIdentML document (rather than just top matches) and if the results from the decoy search are also saved. It would only be possible to do this at the peptide level for an MS-MS search, because changing thresholds would normally have some effect on the protein grouping algorithm.
4. It should be possible to save manual or automated annotation of proteins/peptides in an instance document. A third party tool could be used to save annotations and validations of identified proteins/peptides to an existing instance document.
5. It should be possible to save the results from a search of a metabolically labelled sample. For example, with a 14N/15N experiment, two separate sets of amino acid masses are used, and it must be possible to tell which masses were used for each peptide result.
6. For a search of multiple peak lists, it should be possible to identify the spectrum that matched a particular peptide or protein reported by the search engine. For example, in an LC-MS-MS run, it should be possible to refer back to the spectrum in the peak list file that was searched and from there, if the information is available, to be able to determine the retention time of the spectrum. For an mzML file, the unique 'id' of the spectrum should be available. For all peak list formats, a unique identifier for each spectrum should be stored. For example, for mzML and vendor formats, a PSI "native ID" can be used to unambiguously identify the spectrum in the raw data that matched to a peptide. There is no requirement to store other redundant information in the mzIdentML file that will be available in the peak list data (see Section 5.1.3).
7. It should be possible to search a file to retrieve all molecules that have a specified modification.
8. It should be possible to store the results of a search of spectra against other spectra - i.e. a spectral library search.
9. It should be possible to store the results of a top down search i.e. analysis of complete proteins.
10. Support should be provided for storing fragmentation data so that for example viewers could display which ions in the input data match predicted ion fragment masses.
11. There should be support for storing the results of searches of peptides against nucleic acid databases, including the information about which translation frame the matches were found in.
12. It should be possible to combine the results from multiple search engines into one mzIdentML document. For example, the peptide identification results from two different search engines could be combined using a third tool to give one set of protein results.

There will be limited support for the following use cases:

1. *De novo*. *De novo* peptide sequencing results will be supported to the extent that it will be possible to enumerate and record all possible matches found by a *de novo* technique, however, we anticipate that this will produce extremely large files. In later versions of mzIdentML, solutions will be investigated for defining a standard way of reporting ambiguous combinations of residues and we invite proposals in this area.
2. Support for sequence tagging, in which short sequences defined by a *de novo* process are used to characterize spectra. The final results from a sequence-tag-filtered search can be stored in mzIdentML, but the details of tag generation and filtering cannot.

The following use cases will not be supported in version 1.1 of mzIdentML:

1. It should be possible to store relative and absolute quantitation information at the peptide and protein level using all the popular techniques [to be developed in a separate format called mzQuantML].

2. Support for LC-MS biomarker discovery.
3. Support for complex workflows where multiple data processing algorithms are combined in a pipeline; i.e. only “final” results are represented in mzIdentML v1.1, i.e. only one protein list, no intermediate results. Intermediate analyses can be represented by using multiple mzIdentML files.

3. Concepts and Terminology

This document assumes familiarity with XML Schema notation (www.w3.org/XML/Schema). 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 [RFC2119].

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. *MIAPE MSI* (http://www.psidev.info/miape/msi/). The Minimum Information About a Proteomics Experiment: Mass spectrometry Informatics document defines a checklist of information that should be reported about such a study. It is expected that mzIdentML will be used to support MIAPE:MSI compliant submissions to public repositories (as demonstrated in mzIdentML_MIAPE_1.1.0.doc).
2. *FuGE* (http://fuge.sourceforge.net). FuGE is a data model in UML, and an associated XML rendering, that represents various high-level concepts that are characteristic of functional genomics, such as investigations and protocols. FuGE has been developed by representatives of several standards bodies, with a view to making the representation of functional genomic data sets more consistent, and as such more easily shared and compared. The FuGE specifications are available from [Jones 07].
3. *mzML* (http://www.psidev.info/index.php?q=node/80). mzML is the PSI standard for capturing mass spectra / peak lists resulting from mass spectrometry in proteomics. It is RECOMMENDED that mzIdentML should be used in conjunction with mzML, although it will be possible to use mzIdentML with other formats of mass spectra. This document does not assume familiarity with mzML.

4.1 Important concepts from FuGE

mzIdentML makes use of several components from FuGE to allow the format to be more easily integrated with other FuGE-based formats. However, FuGE is a large, flexible specification that can cover a variety of concepts not required for mzIdentML. The previous mzIdentML v1.0 release imported a separate “FuGE-light” XML schema. In this release, mzIdentML v1.1, the concepts from FuGE have been directly incorporated into the mzIdentML v1.1 schema. Additional knowledge of FuGE is thus not required beyond this specification document.

4.2 The PSI Mass Spectrometry Controlled Vocabulary (CV)

The PSI-MS controlled vocabulary is intended to provide terms for annotation of mzML and mzIdentML 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 by have a “has_units” key in the CV itself (relationship: has_units: UO:0000221 ! dalton). The details of which terms are allowed or required in a given schema section is reported in the mapping file (Section 4.3).

As recommended by the PSI CV guidelines, psi-ms.obo should be dynamically maintained via the 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

4.3 Validation of controlled vocabulary terms

The correct usage of controlled vocabulary terms within mzIdentML is governed by the use of a mapping file which defines each XML location (XPath) where a <cvParam> instance can be used, and the allowed terms from the PSI-MS, or other, controlled vocabularies. The mapping file is read and interpreted by validation software, checking that the data annotation is consistent. The mapping file needs to be checked and updated when the structure of CV is changed, and in some instances when new terms are added to the CV. The draft specifications for the mapping file can be found here: http://www.psidev.info/files/validator/PSI-Mapping.doc. XML paths are associated with CV terms along with a requirement level (MAY, SHOULD or MUST) defining what should be reported by validation software if one of the mapped terms is not provided in an instance document. Example validation software based on the mapping file has been implemented as part of OpenMS: www.psidev.info/validator, which has been used to perform syntactic and semantic validation of the example files listed in Section 5.4.

4.4 Changes from version 1.0

The following changes have been made to the schema in version 1.1.0 compared with version 1.0:

- PeptideEvidence now resides in the SequenceCollection rather than within each SpectrumIdentificationItem to reduce redundancy in the file.
- PeptideEvidence now represents a unique mapping from a peptide sequence (reported in the Peptide element) to a given position with a protein sequence (DBSequence element).
- The combination of Peptide sequence and modifications must be unique in the file within the PeptideElement.
- ProteinDetectionHypothesis (PDH) has been altered to group references to the SpectrumIdentificationItem (SII) elements on which it is based, under a PeptideHypothesis element that references the unique PeptideEvidence element. This allows a file reader to find the peptide sequences on which a PDH is based quickly, without traversing a long list of redundant SII elements.
- The case of a number of elements has been changed to make the case of element names and attributes consistent.
- The FuGElight schema is no longer imported, the elements used are now included directly in one single mzIdentML 1.1.0 schema.
- The missed cleavages attribute has been removed from PeptideEvidence since it can be derived easily for simple cases (such as full trypsin cleavage) but can cause ambiguities for more complex digestion protocols.
- Some attributes have been removed from the Contact element that are already present in the CV for other uses, such as email, address, telephone etc.
- The order of elements has been made more systematic, such as cvParam elements always come at the end of a sequence.

4.5 Changes from version 1.1.0

Con formato: Título 2

The primary update requiring the change from version 1.1.0 to version 1.2.0 is in the inclusion of guidelines for encoding protein group results (Section 5.1.6). Several examples referenced throughout the document are annotated with version 1.1.0. In these cases, it can be assumed that these files are also valid 1.2.0 files, since they do not include protein inference results. Other minor changes have been made to the specification since version 1.1.0, with regards to the encoding of specific workflows – notably searches where pre-fractionation has been performed (Section 5.3.2), and searches employing multiple search engines (Section 5.3.1). In addition, minor changes have been made to the recommended encoding of information, such as retention time (Section 5.1.5). No changes have been made to the XML Schema or mapping file from version 1.1.0, although several open issues have been identified that may be resolved in future major version updates (Section 5.2).

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.

5.1.1 Quantitation

There is a clear requirement for a standard data format that supports quantitative data from studies of peptide and proteins. During the development process, several attempts were made to model the range of analysis procedures currently used in proteome studies that produce quantitative data under the name AnalysisXML. The variability in the different techniques employed (e.g. labelled, label-free) and the continual evolution of new techniques resulted in considerable delays in getting a version 1.0 of AnalysisXML produced. It was decided at the 2008 PSI meeting in Toledo that the best course of action would be to get a stable format released without support for quantitative data, rather than facing further delays. At the 2009 PSI meeting in Turku, several quantification use cases were examined and it was demonstrated that a format for quantification would be simpler to develop independently of the format for identification. It was thus decided to split the development of AnalysisXML into two formats: mzIdentML and mzQuantML. It is expected that mzQuantML will follow a broadly similar structure as the upper level hierarchy of mzIdentML. An instance of mzQuantML will reference back to <SpectrumIdentificationItem> and <ProteinDetectionHypothesis> within an mzIdentML file for peptide and protein identifications respectively. This design is relatively intuitive since software typically performs identification and quantification in independent processes.

5.1.2 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 [Deutsch08], and mzIdentML 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.psidev.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.

5.1.3 Use of identifiers for input spectra to a search

A <SpectrumIdentificationResult> is linked to the source spectrum (in an external file) from which the identifications are made by way of a reference in the spectrumID attribute and via the <SpectraData> 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. A CV term MUST be imported into the <SpectraData> element to demonstrate which system for identifying input spectra is being used in the spectrumID attribute of <SpectrumIdentificationResult>. *Note, this table shows examples from the CV but will be extended.*

Update in version 1.2:
Version 1.1.0 of the specification document states "~~T~~the CV holds the definite specification for legal encodings of spectrumID values". In version 1.2, the only legal ways of referencing a spectrum identification format are provided below in Table 1. Any new spectral formats that cannot fit into this schema require an update to this~~s.~~ document.

Con formato: Fuente: Sin Negrita

Con formato: Fuente: Sin Negrita

Con formato: Fuente: Sin Negrita

Con formato: Fuente: Sin Negrita

ID	**Term**	**Data type**	**Comment**

MS:1000768	Thermo nativeID	controllerType=xsd:nonNegativeInteger
		controllerNumber=xsd:positiveInteger
			controller=0 is usually the mass spectrometer

	format	scan=xsd:positiveInteger.	

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 referencing 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 referencing 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 referencing mzXML, or a DTA folder where native scan numbers can be derived.
MS:1000777	spectrum identifier nativeID format	spectrum=xsd:nonNegativeInteger	Used for referencing mzData. The spectrum id attribute is referenced.
MS:1001530	mzML unique identifier	xsd:string	Used for referencing mzML. The value of the spectrum id attribute is referenced directly.

~~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~~	~~scan=xsd:nonNegativeInteger~~	

	~~format~~		
~~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 mzIdentML, the spectrumID 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:

```
<SpectrumIdentificationResult id="Res1" spectrumID="index=2" SpectraData_ref="InputSpectra1">
```

...

```
<SpectraData location="local/mgf/merge.mgf" id="SD_1" >
  <FileFormat>
    <cvParam accession="MS:1001062" name="Mascot MGF file" cvRef="PSI-MS" />
  </FileFormat>
  <SpectrumIDFormat>
    <cvParam accession="MS:1000774" name="multiple peak list nativeID format" cvRef="PSI-MS" />
  </SpectrumIDFormat>
</SpectraData>
```

Spectra represented in mzML (in the <Spectrum> element) have a unique identifier within the "id" attribute, formulated as above depending on the source of the file. If the source file is mzML, <SpectrumIdentificationResult> MUST reference the value in "id" attribute to reference the spectrum that was searched.

5.1.4 Recommendations for reporting multiple spectrum identifications and protein hypotheses

There has been discussion of including a recommendation in this specification for what should be reported to allow statistical processing of results. For example, it has been noted that without peptide identifications reported for all (or most) spectra, it is difficult to perform comparative statistical analysis without a reference point. As discussed in Section 7.4~~7.1.4~~, mzIdentML allows multiple peptide and protein identifications to be included with a flag for those identifications that the file producer deems to have passed a threshold. This structure MAY be used to provide sufficient information to allow further statistical processing to be carried out but it has been decided that recommendations about the level of detail to report are handled as part of the MIAPE MSI document.

5.1.5 Exclusion of information relating to mass spectral data

It has been decided that the peak list that was searched should remain external to the format, for example referenced as an mzML file. Similarly other data items that may be used during a search, but can be retrieved from the source spectra file ~~are not~~should not generally be duplicated in mzIdentML~~, such as retention time~~.

Certain exceptions are made for data types that are sometimes difficult to retrieve from source spectra or are typically expected to be present in search results. The version 1.1 specification implied that retention time should not be captured in mzIdentML. It has now been agreed that the retention time of a given spectrum MAY be captured as part of <SpectrumIdentificationResult>, using the same terminology employed within mzML.

5.1.6 Protein grouping encoding

Con formato: Título 3

This section is newly inserted in the mzIdentML version 1.2 specifications. In version 1.1.0, CV terms had been proposed for representing set relationships between different proteins within groups, but there was requirement that particular terms were used. A given data structure from software could be mapped onto the hierarchy <ProteinAmbiguityGroup> and <ProteinDetectionHypothesis> in mzIdentML in different ways, leading to difficulties for data consumers. As such, a working group has now agreed a more rigid encoding detailed as follows and in Table 2 /Table 3.

1. As in mzIdentML version 1.1, a single protein accession that has been cited by software (Figure 1A) is captured in mzIdentML in <ProteinDetectionHypothesis> (PDH).
    a. A PDH MAY contain scores or statistical values produced by the export software, encoded as CV terms.

Con formato: Sangría: Sangría francesa: 0,63 cm, Espacio Después: 0 pto, Interlineado: sencillo

Comentario [JV1]: I think that the addition of code snippets to describe the rules would improve readibility

2. A "protein group" (Figure 1B), representing a "biological entity" for which the software claims independent evidence is present, MUST be mapped onto <ProteinAmbiguityGroup> (PAG).
    a. A PAG MAY have additional scores produced by the export software, encoded as CV terms.
3. The reporting of protein identification thresholds is now mapped onto PAGs. There is no desire to change the core XML Schema Document (XSD) for mzIdentML and as such, a new CV term "protein group passes threshold" value= "xsd:boolean" MUST be present on every PAG (MS:TODO_ACCESSION).
    a. The attribute *passThreshold* = "true|false" remains present on PDH and MAY be used if software packages wish to report a two-level hierarchy of thresholds applied, however, it is not expected that consuming software will use this attribute to determine which proteins have been reported as identified.
4. The <ProteinDetectionList> MUST contain the CV term "count of identified proteins" value= "xsd:integer" (MS:TODO_ACCESSION). The value MUST be derived from the count of PAGs passing the threshold reported in the file and will be checked by validation software.
5. Few software packages report "protein clusters" at present (Figure 1C), but for those packages that wish to report clusters, a CV term "cluster identifier" value = "xsd:integer" SHOULD be used (MS:TODO_ACCESSION). The integer identifier MUST be shared by all PAGs belonging to the same cluster. An optional term "count of identified clusters" value = "xsd:integer" MAY be annotated on the ProteinDetectionList.
6. Every PDH MUST be annotated as either a "leading protein" or a "non-leading protein" (accessions to add), as defined in Table 1, within a PAG. This recommendation thus makes it explicit for consuming software whether one or more proteins have stronger evidence than others in the group (see Table 2 for examples).
    a. An additional term, "group representative" MAY be used to annotate one PDH, which is also flagged as a "leading protein", if the export software wishes to enforce that only one of potential several "leading proteins" will be interpreted by the consuming software as the representative of the group.
    b. If the export software does not explicitly flag one protein as the "group representative", it is assumed that if consuming software requires a single accession to represent the group, an arbitrary choice will be made (among "leading proteins" only if these exist).
7. Any PDHs MAY be annotated with terms present in the CV for spectrum/sequence same-set, spectrum/sequence subset, spectrum/sequence subsumable, marginally distinguished and so on (Table 1).
    a. A PDH MAY be annotated with more than one of these terms if appropriate to describe the complex set relationships that exist within a group.
    b. Developers of software packages MAY propose additional terms for describing group membership of PDHs.

c. The associated value for these CV terms MAY be used to annotate which PDH(s) are the super/same-set of the annotated PDH.

d. There is no expectation that consuming software should be aware of these terms, but may be useful in internal pipeline or visualization software packages that are specifically designed to work with this terminology set.

8. Some PDHs could be mapped to more than one group (PAG), for example where proteins are multiply subsumed. To capture these cases, multiple PDHs in different PAGs MAY reference the same <DBSequence>.

The semantic validation software has been updated to encode these rules and report errors (“MUST” rule), warnings (“SHOULD” rule) or informational messages (“MAY” rule).

mzIdentML context	CV term	Values	Require-ment level	Description
ProteinDetection-List	count of identified proteins	xsd:integer	MUST	The value reported MUST equal the number of PAGs with “protein group passes threshold” value = “true”
ProteinDetection-List	count of identified clusters	xsd:integer	MAY	If protein clusters have been reported in the file, the exporter may choose to annotate the ProteinDetectionList with the number identified above threshold.
ProteinAmbiguity-Group	cluster identifier	xsd:integer	MAY	A common identifier reported allows multiple PAGs to be linked, for example indicating some peptides are shared between different PAGs.
ProteinAmbiguity-Group	number of distinct protein sequences	xsd:integer	MAY	The number of distinct protein sequences among the PDHs in the group. For example, if there are two PDHs with different identifiers that have identical full length sequences, the value would be 1.
ProteinDetection-Hypothesis	leading protein OR non-leading protein	-	MUST OR MUST	Every PDH in each PAG MUST be flagged as a leading protein or a non-leading protein and each PAG MUST contain at least one leading protein. A “leading protein” is defined as a protein that has the strongest or near strongest (further explained in Table 2) set of evidence for being present in the sample studied, amongst the grouped protein accessions. A “non-leading protein” is defined as a protein that has (substantially) less evidence than other proteins within the same group, and is thus less likely to have been present in the sample studied.
ProteinDetection-Hypothesis	group representative	-	MAY	Each PAG ~~MUST~~ MAY contain zero or one PDH flagged as the group representative, if the software wishes to flag a preference (often arbitrary or for example based on alphabetical ordering) amongst the leading proteins. The group representative term can thus be viewed a “tiebreaker” if the export software wishes to make this distinction.
ProteinDetection-Hypothesis	Sequence Same-Set Protein	xsd: “list of strings” space separated list of PDH IDs that are same-set.	MAY	A protein that is indistinguishable or equivalent to another protein in the group, having matches to an identical set of peptide sequences.
ProteinDetection-Hypothesis	Spectrum Same-Set Protein	xsd: “list of strings” space separated list of PDH IDs that are same-set.	MAY	A protein that is indistinguishable or equivalent to another protein in the group, having PSMs derived from the same set of spectra.
ProteinDetection-Hypothesis	Sequence Subset Protein	xsd: “list of strings” space separated list of PDH IDs that are super-set.	MAY	A protein for which the matched peptide sequences are a subset of the matched peptide sequences for another protein in the group.
ProteinDetection-Hypothesis	Spectrum Subset Protein	xsd: “list of strings” space separated list of PDH IDs that are	MAY	A protein for which the matched spectra are a subset of the matched spectra for another protein in the group.

Con formato: Fuente: (Predeterminado) Arial, 10 pto

Con formato: Espacio Después: 0 pto

Tabla con formato

Tabla con formato

Comentario [j2]: We can define this data type in XML as follows:

```
<xs:simpleType name="list of strings">
  <xs:list itemType="xsd:string"/>
</xs:simpleType>
```

But I don’t know if this can be translated to the CV

		super-set.		
ProteinDetection-Hypothesis	Sequence Multiply Subsumable Protein	xsd: "list_of_strings" space separated list of PDH IDs that subsume this PDH.	MAY	A protein for which the matched peptide sequences are the same, or a subset of, the matched peptide sequences for two or more other proteins combined. These other proteins need not all be in the same group.
ProteinDetection-Hypothesis	Spectrum Multiply Subsumable Protein	xsd: "list_of_strings" space separated list of PDH IDs that subsume this PDH.	MAY	A protein for which the matched spectra are the same, or a subset of, the matched spectra for two or more other proteins combined. These other proteins need not all be in the same group.
ProteinDetection-Hypothesis	Marginally distinguished protein	-	MAY	Assigned to a non-leading PDH that has some independent evidence to support its presence relative to the leading protein(s) e.g. the PDH may have a unique peptide but not sufficient to be promoted as, for example, a leading protein of another a PAG.

Table 2 New CV terms for reporting protein set (group) relationships and global statistics about the protein identification results. The semantic validation software for mzIdentML (v.1.2) reports an error (MUST), a warning (SHOULD) or an informational message (MAY) if these terms are not reported within the file.

Con formato: Izquierda

Scenario	Software preference	Encoding

Software scores A and B as same-set, C and D as subset.	Software wishes to make A the group representative (arbitrary)	A = leading protein & group representative
		B = leading protein
		C = non-leading protein
		D = non-leading protein
		(Use of formal same-set and subset notation is also allowed but optional)

As above	Software does not wish to choose which is the group representative	A = leading protein
		B = leading protein
		C = non-leading protein
		D = non-leading protein

Software scores A as best protein, B, C and D are all subset or subsumed	N/A	A = leading protein
		B = non-leading protein
		C = non-leading protein
		D = non-leading protein

Software scores all four proteins as same-set or as more generally as having equal evidence	Software wishes to make A the group representative (arbitrary)	A = leading protein & group representative
		B = leading protein
		C = leading protein
		D = leading protein

As above	Software does not wish to choose which is the group representative	A = leading protein
		B = leading protein
		C = leading protein
		D = leading protein

Software scores A as having slightly more evidence than B. B has additional weak independent evidence relative to A. C and D have less evidence than either A or B.	Software wishes to assign A as the leading protein and the independent evidence for B is not sufficient for it to form a new PAG.	A = leading protein
		B = non-leading protein & marginally distinguished (optional)
		C = non-leading protein
		D = non-leading protein

As above	Software does not wish to choose which is the leading protein out of A and B or group representative	A = leading protein
		B = leading protein
		C = non-leading protein
		D = non-leading protein

As above	Software does not wish to choose which is the leading protein but does select a group representative	A = leading protein & group representative
		B = leading protein
		C = non-leading protein
		D = non-leading protein

Table 3 A summary of grouping options and recommendation for CV term annotations, assuming a group of four related proteins A-D.

Con formato: Epígrafe, Izquierda

5.2 Open Issues

~~None at present, any issues identified during the document process will appear here.~~

Several minor issues have been identified in the mzIdentML 1.1 and 1.2 schema but at present the overall stability of the schema is considered of utmost importance and, as such, these issues are documented here. If there is sufficient need for a major version update, these issues will be resolved then.

5.2.1 The passThreshold attribute for protein results

Con formato: Título 3

The attribute passThreshold is present on <ProteinDetectionHypothesis> rather than <ProteinAmbiguityGroup>. It has been discussed that a more logical placing of the attribute would be at the <ProteinAmbiguityGroup> level, since this element represents the biological entity identified and at which many software packages perform thresholding. An update to the specification document is expected shortly to improve on the reporting of protein grouping, which is likely to recommend the use of new CV terms on <ProteinAmbiguityGroup> as a workaround.

Con formato: Sin Resaltar

5.2.2 Optional dbSequence_ref attribute on <ProteinDetectionHypothesis>

Con formato: Título 3

The attribute dbSequence_ref is OPTIONAL on <ProteinDetectionHypothesis>, with the following documentation "A reference to the corresponding DBSequence entry. This optional and redundant, because the PeptideEvidence elements referenced from here also map to the DBSequence". It has been acknowledged that it would be easier for consuming software if this attribute was mandatory, to prevent having to traverse down to the first <PeptideHypothesis> element and following two onwards links. However, since the code work around is already described in the documentation, it has been determined that no change will be made to the schema at this time.

5.2.3 Optional <FileFormat> element within <SearchDatabase>, <SourceFile> and <SpectraData>

Con formato: Sin Resaltar

The elements <SearchDatabase>, <SourceFile> and <SpectraData> all have an optional sub-element <FileFormat>. It has been acknowledged that file reading software would be more straightforward to write if this element was MANDATORY, but this change is not going to be made at the present time. It is RECOMMENDED that all export software includes the <FileFormat> element in all cases, although reading software will have to cope with files in which the element is absent and will have to fall back on looking at the file extension if this information is essential for onward processing.

Con formato: Sin Resaltar

5.3 Comments on Specific Use Cases

Many special use cases for mzIdentML were considered during its development. Each of these use cases has a corresponding example file that exercises the relevant part of the schema and provides a reference implementation example (see supporting documentation). Authors of software that create mzIdentML are encouraged to examine the examples that accompany this format release before implementing the writer. Further, such authors are encouraged to use the validator before releasing any new writer code and working with the PSI PI Working Group to resolve any issues. In the subsections below, we comment on a few of the notable use cases that were considered.

5.3.1 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 mzIdentML as follows (see Section 6 for documentation of the model elements). Note that the RECOMMENDED encoding has changed since the version 1.1.0 specification as a result of community feedback.

The results of searching with multiple search engines SHOULD be captured in one file that contains one <SpectrumIdentificationList> containing the results after they have been combined from different search engines, for example containing a "combined score" annotated on <SpectrumIdentificationItem> and, OPTIONALLY, the source scores from the original search engines. In addition, the file MAY contain one <SpectrumIdentificationList> per search engine employed, in which the source scores and ranking of identifications are captured representing the initial results before they were combined. A new CV term has been

Con formato: Normal

added (MS:1002315, “consensus result”) to the PSI-MS CV which SHOULD be added to the <SpectrumIdentificationList> containing the combined results to indicate this status. Any software that employs multiple search engines but does not produce a “combined” score, e.g. prior to protein inference, SHOULD still create a “combined” <SpectrumIdentificationList> that can be referenced from the <ProteinDetectionList>.

If source results are included as additional instances of <SpectrumIdentificationList>, each <SpectrumIdentification> element MUST reference the relevant <SpectrumIdentificationProtocol> to capture the search parameters and software employed for each search engine. One <SpectrumIdentification> element is required to capture the process of combining search engine results, referencing the “combined” <SpectrumIdentificationList> produced. This <SpectrumIdentification> element MUST reference an artificially created <SpectrumIdentificationProtocol> holding representative parameters (i.e. search tolerances, enzyme and modifications), since these are MANDATORY elements. If the same search parameters were not employed in all source searches, the parameters should be set with superset or widest values i.e. all modifications that have been searched; widest tolerances and so on. All search engines that have been employed SHOULD be represented within the <AnalysisSoftwareList>. It must also be highlighted that mzIdentML cannot be used to model the order in which the software was used (it does not support workflows).

It is acknowledged that mzIdentML consuming software could find it challenging to interpret results derived from multiple search engines in the same file. As such, the default behaviour expected of consuming software is that the first instance of spectrum in a file (<SpectrumIdentificationResult>) is taken to contain the “final” score of the quality of the identification. Further instances of the same spectrum found in other <SpectrumIdentificationList> instances are taken to be supporting information that can be safely ignored.

The protein-level results produced (assuming protein inference has taken place), MUST be represented in one <ProteinDetectionList>. Each <ProteinDetectionHypothesis> SHOULD reference the <SpectrumIdentificationItem> elements on which it is based in the “combined” <SpectrumIdentificationList> only.

Use cases in which multiple different searches are made with the same search engine but different parameters SHOULD be modelled using the same general structures as described above.

5.3.2 Pre-fractionation of samples prior to MS

Con formato: Título 3

~~Each database search SHOULD be represented by an instance of <SpectrumIdentification> (application of the protocol) which references the <SpectrumIdentificationProtocol> and the output data: an instance of <SpectrumIdentifcationList>. As such, if three database search engines are used, there SHOULD be three instances each of <SpectrumIdentification>, <SpectrumIdentificationProtocol> and <SpectrumIdentifcationList>. Results are then combined into a list of proteins by a separate process, represented as one instance of <ProteinDetection> (application of the protocol), which references one instance of <ProteinDetectionProtocol> and references (as input) the three instances of <SpectrumIdentificationList>. The output of <ProteinDetection> is one instance of <ProteinDetectionList>. If a secondary scoring scheme is used to weigh evidence for peptide-spectrum matches according to the search engines that have identified them, any consensus or composite scores should be assigned to each <SpectrumIdentificationItem> within parallel lists.~~

~~It was decided that more complex arrangements of workflows cannot be represented in mzIdentML version 1.1, such as different protein lists produced by each search engine, then combined by an additional process, since it becomes difficult to define which are “final” and which are “intermediate” results for data consumers and implementers of databases. Such workflows may be incorporated into later versions of the format.~~

It is common in many workflows for pre-fractionation of a sample to be performed prior to MS, for example via 1D or 2D gel electrophoresis or 2D LC. In some scenarios results of database searches are combined prior to protein inference and in other instances there is no combination of results prior to protein inference. We have identified the following scenarios and describe the RECOMMENDED encoding in each case.

Scenario	Encoding
i) A sample is fractionated into *n* sub-samples, prior to	*n* mzIdentML files SHOULD be produced, each

n runs on the MS; ii) the search engine performs *n* searches, producing *n* protein-lists.[1]	containing 1 <SpectrumIdentificationList>, 1 <SpectrumIdentificationProtocol>, 1<SpectrumIdentification>, 1 <ProteinDetection>, 1 <ProteinDetectionList>.
i) A sample is fractionated into *n* sub-samples, prior to *n* runs on the MS; ii) the search engine imports *n* peak lists but internally integrates results to produce one protein list.[3]	The final file SHOULD contain *n* <SpectrumIdentificationList>s, *n* <SpectrumIdentificationProtocol>s, *n* <SpectrumIdentification>s, 1 <ProteinDetection>, 1 <ProteinDetectionList>.
i) A sample is fractionated into *n* sub-samples, prior to *n* runs on the MS; ii) the search engine performs *n* searches, producing *n* lists of spectrum identifications; iii) post-processing software integrates results to produce one protein list.[2]	As above.

It is acknowledged that mzIdentML consuming software could find it challenging to interpret results derived from multiple fractions in the same file.

The protein-level results produced (assuming protein inference has taken place), MUST be represented in one <ProteinDetectionList>. Each <ProteinDetectionHypothesis> SHOULD reference the <SpectrumIdentificationItem> elements on which it is based in the “combined” <SpectrumIdentificationList> only.

~~5.3.2~~5.3.3 Spectral library searches

An alternative to sequence database searches for identifying peptides from MS data is to search a pre-compiled library of peptide-spectrum matches. These spectral library searches are supported in mzIdentML. The recommended encoding is similar to sequence database search results, the main difference being that rather than protein sequences represented in the <DBSequence> element, the peptide sequence for each library entry is stored here instead. Additional information about the peptide-spectrum match, such as observed modifications and consensus scores, can be stored as CV terms within each <DBSequence> entry.

5.4 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 manually from:
http://code.google.com/p/psi-pi/source/browse/#svn%2Ftrunk%2Fexamples%2F1_1examples

a) 55merge_mascot_full.mzid - example MS-MS search results including decoy matches from Mascot.
b) 55merge_omssa.mzid - example MS-MS search results including decoy matches from OMSSA.
c) 55merge_tandem.mzid - example MS-MS search results including decoy matches from X!Tandem.
d) MPC_example.mzid – an example of PSMs from different search engines, assembled into proteins using a third-party algorithm; false-discovery estimation using decoy database.
e) Mascot_NA_example.mzid - an example of a search against an EST database with Mascot.
f) Mascot_top_down_example.mzid - a single MS/MS spectra from an intact protein, searched with Mascot.
g) Sequest_example_ver1.1.mzid - a simple example derived from a “.out” file produced by SEQUEST.
h) mascot_pmf_example.mzid - example Peptide Mass Fingerprint search with Mascot.

Comentario [JV3]: What's the meaning of this superscript numbers in the table?

Comentario [JV4]: Maybe highlight that this is the same case explained above for the search engines. In this case it could also be needed to have a consensus <SpectrumIdentificationList>. In that case, the same CV param should be used?

Comentario [JV5]: Add a CV param to indicate that different fractions are present in the same file? Otherwise, if no consensus list is available, it si going to be difficult for readers to know what's going on.

Comentario [JV6]: Specify that it is case 2. Above can be interpreted as any of the top two.

Comentario [JV7]: I would add a paragraph to indicate how to treat this case for readers. What happens with peptides that are identified multiple times in the different fractions? The reader is supposed to take all of them since they will be different spectrum identifications or maybe only the <Spectrum IdentificationItems> that are referenced by each <ProteinDetectionHypothesis> in the <ProteinDetectionList>?
If there is a consensus list, things get easier, but if not, things can be a bit messy.

Comentario [JV8]: I copied/pasted this from the previous section, since I feel something similar is needed here. Again, this is only possible if there is a consensus list.

6. Model in XML Schema

An overview of the schema is presented in Figure 1. The following documentation is automatically generated from the XML Schema.

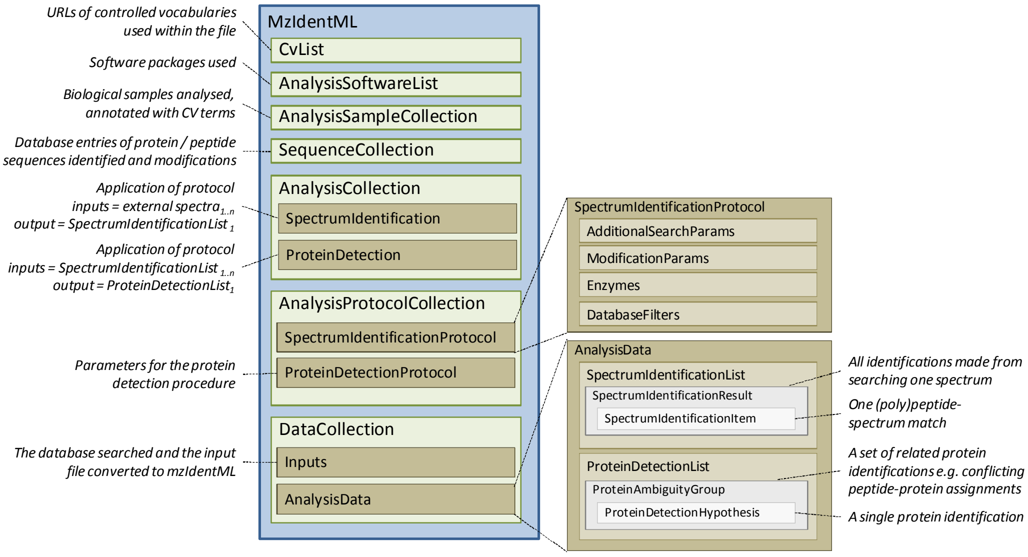

URLs of controlled vocabularies used within the file
Software packages used
Biological samples analysed, annotated with CV terms
Database entries of protein / peptide sequences identified and modifications
Application of protocol inputs = external spectra$_{1..n}$ output = SpectrumIdentificationList$_1$
Application of protocol inputs = SpectrumIdentificationList$_{1..n}$ output = ProteinDetectionList$_1$
Parameters for the protein detection procedure
The database searched and the input file converted to mzIdentML
MzIdentML
CvList
AnalysisSoftwareList
AnalysisSampleCollection
SequenceCollection
AnalysisCollection
SpectrumIdentification
ProteinDetection
AnalysisProtocolCollection
SpectrumIdentificationProtocol
ProteinDetectionProtocol
DataCollection
Inputs
AnalysisData
SpectrumIdentificationProtocol
AdditionalSearchParams
ModificationParams
Enzymes
DatabaseFilters
AnalysisData
SpectrumIdentificationList
SpectrumIdentificationResult
SpectrumIdentificationItem
ProteinDetectionList
ProteinAmbiguityGroup
ProteinDetectionHypothesis
All identifications made from searching one spectrum
One (poly)peptide-spectrum match
A set of related protein identifications e.g. conflicting peptide-protein assignments
A single protein identification

Figure 1 A diagrammatic overview of the mzIdentML schema.

6.1 Element <MzIdentML>

Definition: The upper-most hierarchy level of mzIdentML with sub-containers for example describing software, protocols and search results (spectrum identifications or protein detection results).

Type: MzIdentMLType

Attributes:

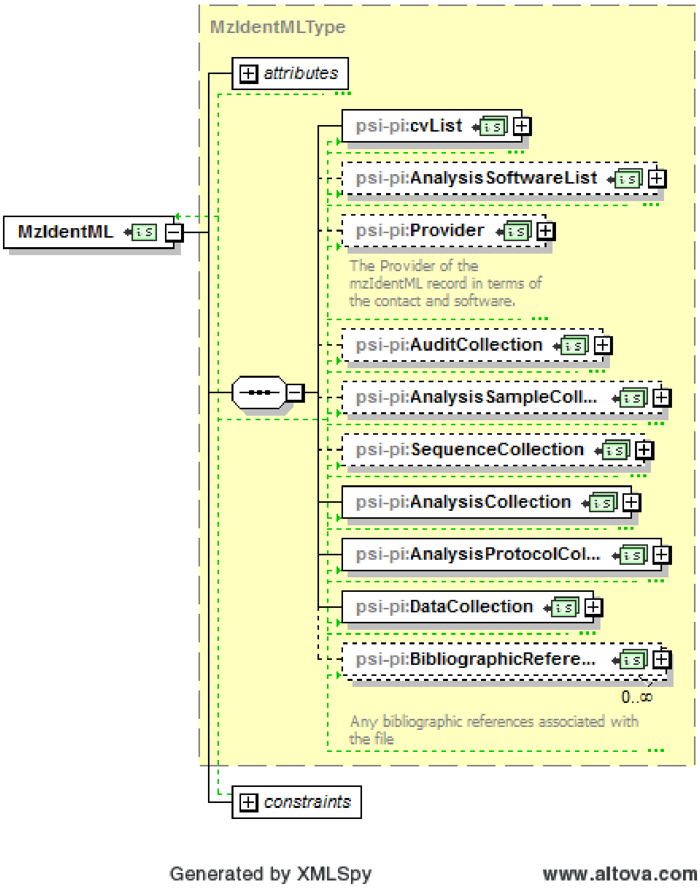
Attribute Name	Data Type	Use	Definition
creationDate	xsd:dateTime	optional	The date on which the file was produced.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
version	versionRegex	required	The version of the schema this instance document refers to, in the format x.y.z. Changes to z should not affect prevent instance documents from validating.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvList	1	1	The list of controlled vocabularies used in the file.
AnalysisSoftwareList	0	1	The software packages used to perform the analyses.
Provider	0	1	The Provider of the mzIdentML record in terms of the contact and software.

AuditCollection	0	1	The complete set of Contacts (people and organisations) for this file.
AnalysisSampleCollection	0	1	The samples analysed can optionally be recorded using CV terms for descriptions. If a composite sample has been analysed, the subsample association can be used to build a hierarchical description.
SequenceCollection	0	1	The collection of sequences (DBSequence or Peptide) identified and their relationship between each other (PeptideEvidence) to be referenced elsewhere in the results.
AnalysisCollection	1	1	The analyses performed to get the results, which map the input and output data sets. Analyses are for example: SpectrumIdentification (resulting in peptides) or ProteinDetection (assemble proteins from peptides).
AnalysisProtocolCollection	1	1	The collection of protocols which include the parameters and settings of the performed analyses.
DataCollection	1	1	The collection of input and output data sets of the analyses.
BibliographicReference	0	unbounded	Any bibliographic references associated with the file

Graphical Context:


MzIdentMLType
attributes
MzIdentML
psi-pi:cvList
psi-pi:AnalysisSoftwareList
psi-pi:Provider
The Provider of the mzIdentML record in terms of the contact and software.
psi-pi:AuditCollection
psi-pi:AnalysisSampleColl...
psi-pi:SequenceCollection
psi-pi:AnalysisCollection
psi-pi:AnalysisProtocolCol...
psi-pi:DataCollection
psi-pi:BibliographicRefere...
0..∞
Any bibliographic references associated with the file
constraints
Generated by XMLSpy
www.altova.com

Example Context:

```
<MzIdentML id="" version="1.1.0"
  xsi:schemaLocation="http://psidev.info/psi/pi/mzIdentML/1.1 ../../schema/mzIdentML1.1.0.xsd"
  xmlns="http://psidev.info/psi/pi/mzIdentML/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" creationDate="2011-03-25T13:16:49">
  <cvList>
    <cv id="PSI-MS" fullName="Proteomics Standards Initiative Mass Spectrometry Vocabularies"
      uri="http://psidev.cvs.sourceforge.net/viewvc/*checkout*/psidev/psi/psi-
ms/mzML/controlledVocabulary/psi-ms.obo"
  ...
</MzIdentML>
```

6.2 Element <AdditionalSearchParams>

Definition: The search parameters other than the modifications searched.

Type: ParamListType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	1	unbounded	A single user-defined parameter.

Example Context:

```
<AdditionalSearchParams>
  <userParam name="Mascot Instrument Name" value="Default"/>
  <cvParam accession="MS:1001211" name="parent mass type mono" cvRef="PSI-MS"/>
  <cvParam accession="MS:1001108" name="param: a ion" cvRef="PSI-MS"/>
  <cvParam accession="MS:1001146" name="param: a ion-NH3" cvRef="PSI-MS"/>
  <cvParam accession="MS:1001118" name="param: b ion" cvRef="PSI-MS"/>
  <cvParam accession="MS:1001149" name="param: b ion-NH3" cvRef="PSI-MS"/>
  ...
</AdditionalSearchParams>
```

cvParam Mapping Rules:

```
Path /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/AdditionalSearchParams
MAY supply a *child* term of MS:1001302 (search engine specific input parameter) one or more times
  e.g.: MS:1001005 (Sequest:CleavesAt)
  e.g.: MS:1001007 (Sequest:OutputLines)
  e.g.: MS:1001009 (Sequest:DescriptionLines)
  e.g.: MS:1001026 (Sequest:NormalizeXCorrValues)
  e.g.: MS:1001028 (Sequest:SequenceHeaderFilter)
  e.g.: MS:1001032 (Sequest:SequencePartialFilter)
  e.g.: MS:1001037 (Sequest:ShowFragmentIons)
  e.g.: MS:1001038 (Sequest:Consensus)
  e.g.: MS:1001042 (Sequest:LimitTo)
  e.g.: MS:1001046 (Sequest:sort_by_dCn)
  et al.
MAY supply a *child* term of MS:1001066 (ions series considered in search) one or more times
  e.g.: MS:1001108 (param: a ion)
  e.g.: MS:1001118 (param: b ion)
  e.g.: MS:1001119 (param: c ion)
  e.g.: MS:1001146 (param: a ion-NH3)
  e.g.: MS:1001148 (param: a ion-H2O)
  e.g.: MS:1001149 (param: b ion-NH3)
  e.g.: MS:1001150 (param: b ion-H2O)
  e.g.: MS:1001151 (param: y ion-NH3)
  e.g.: MS:1001152 (param: y ion-H2O)
  e.g.: MS:1001257 (param: v ion)
  et al.
MAY supply a *child* term of MS:1001210 (mass type settings) one or more times
  e.g.: MS:1001211 (parent mass type mono)
  e.g.: MS:1001212 (parent mass type average)
  e.g.: MS:1001255 (fragment mass type average)
  e.g.: MS:1001256 (fragment mass type mono)
```

Example cvParams:

```
<cvParam accession="MS:1001211" name="parent mass type mono" cvRef="PSI-MS"/>
<cvParam accession="MS:1001108" name="param: a ion" cvRef="PSI-MS"/>
<cvParam accession="MS:1001146" name="param: a ion-NH3" cvRef="PSI-MS"/>
<cvParam accession="MS:1001118" name="param: b ion" cvRef="PSI-MS"/>
<cvParam accession="MS:1001149" name="param: b ion-NH3" cvRef="PSI-MS"/>
<cvParam accession="MS:1001262" name="param: y ion" cvRef="PSI-MS"/>
<cvParam accession="MS:1001151" name="param: y ion-NH3" cvRef="PSI-MS"/>
<cvParam accession="MS:1001256" cvRef="PSI-MS" name="fragment mass type mono"/>
```

Example userParams:

```
<userParam name="Mascot Instrument Name" value="Default"/>
```

6.3 Element <Affiliation>

Definition: The organization a person belongs to.

Type: AffiliationType

Attributes:

Attribute Name	Data Type	Use	Definition
organization_ref	xsd:string	required	A reference to the organization this contact belongs to.

Subelements: none

Example Context:

```
<Affiliation organization_ref="ORG_DOC_OWNER"/>
```

6.4 Element <AmbiguousResidue>

Definition: Ambiguous residues e.g. X can be specified by the Code attribute and a set of parameters for example giving the different masses that will be used in the search.

Type: AmbiguousResidueType

Attributes:

Attribute Name	Data Type	Use	Definition
code	chars	required	The single letter code of the ambiguous residue e.g. X.

Subelements:

Subelement	minOccurs	maxOccurs	Definition

Name			
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	1	unbounded	A single user-defined parameter.

Example Context:

```
<AmbiguousResidue code="X">
  <cvParam accession="MS:1001360" name="alternate single letter codes" cvRef="PSI-MS"
    value="A C D E F G H I K L M N O P Q R S T U V W Y"/>
</AmbiguousResidue>
```

cvParam Mapping Rules:

```
Path /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/MassTable/AmbiguousResidue
MAY supply a *child* term of MS:1001359 (ambiguous residues) one or more times
  e.g.: MS:1001360 (alternate single letter codes)
  e.g.: MS:1001361 (alternate mass)
```

Example cvParams:

```
<cvParam accession="MS:1001360" name="alternate single letter codes" cvRef="PSI-MS"
```

6.5 Element <AnalysisCollection>

Definition: The analyses performed to get the results, which map the input and output data sets. Analyses are for example: SpectrumIdentification (resulting in peptides) or ProteinDetection (assemble proteins from peptides).

Type: AnalysisCollectionType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SpectrumIdentification	1	unbounded	An Analysis which tries to identify peptides in input spectra, referencing the database searched, the input spectra, the output results and the protocol that is run.
ProteinDetection	0	1	An Analysis which assembles a set of peptides (e.g. from a spectra search analysis) to proteins.

Example Context:

```
<AnalysisCollection>
  <SpectrumIdentification id="SI" spectrumIdentificationProtocol_ref="SIP"
    spectrumIdentificationList_ref="SIL_1" activityDate="2011-03-24T11:37:37">
    <InputSpectra spectraData_ref="SD_1"/>
    <SearchDatabaseRef searchDatabase_ref="SDB_NeoProt_tripledecoy"/>
  </SpectrumIdentification>
  <ProteinDetection id="PD_1" proteinDetectionProtocol_ref="PDP_MascotParser_1"
  ...
</AnalysisCollection>
```

6.6 Element <AnalysisData>

Definition: Data sets generated by the analyses, including peptide and protein lists.

Type: AnalysisDataType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SpectrumIdentificationList	1	unbounded	Represents the set of all search results from SpectrumIdentification.
ProteinDetectionList	0	1	The protein list resulting from a protein detection process.

Example Context:

```
<AnalysisData>
<SpectrumIdentificationList id="SII_LIST_1" xmlns="http://psidev.info/psi/pi/mzIdentML/1.1">
    <FragmentationTable>
        <Measure id="Measure_MZ">
            <cvParam accession="MS:1001225" cvRef="PSI-MS" unitCvRef="PSI-MS" unitName="m/z"
unitAccession="MS:1000040" name="product ion m/z"/>
        </Measure>
        <Measure id="Measure_Int">
  ...
</AnalysisData>
```

6.7 Element <AnalysisParams>

Definition: The parameters and settings for the protein detection given as CV terms.

Type: ParamListType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	1	unbounded	A single user-defined parameter.

Example Context:

```
<AnalysisParams>
  <cvParam accession="MS:1001316" name="mascot:SigThreshold" cvRef="PSI-MS" value="0.05"/>
  <cvParam accession="MS:1001317" name="mascot:MaxProteinHits" cvRef="PSI-MS" value="Auto"/>
  <cvParam accession="MS:1001318" name="mascot:ProteinScoringMethod" cvRef="PSI-MS"
    value="MudPIT"/>
  <cvParam accession="MS:1001319" name="mascot:MinMSMSThreshold" cvRef="PSI-MS" value="0"/>
  <cvParam accession="MS:1001320" name="mascot:ShowHomologousProteinsWithSamePeptides"
  ...
</AnalysisParams>
```

cvParam Mapping Rules:

```
Path /MzIdentML/AnalysisProtocolCollection/ProteinDetectionProtocol/AnalysisParams
MAY supply a *child* term of MS:1001302 (search engine specific input parameter) one or more
times
  e.g.: MS:1001005 (Sequest:CleavesAt)
  e.g.: MS:1001007 (Sequest:OutputLines)
  e.g.: MS:1001009 (Sequest:DescriptionLines)
  e.g.: MS:1001026 (Sequest:NormalizeXCorrValues)
  e.g.: MS:1001028 (Sequest:SequenceHeaderFilter)
  e.g.: MS:1001032 (Sequest:SequencePartialFilter)
  e.g.: MS:1001037 (Sequest:ShowFragmentIons)
  e.g.: MS:1001038 (Sequest:Consensus)
  e.g.: MS:1001042 (Sequest:LimitTo)
  e.g.: MS:1001046 (Sequest:sort_by_dCn)
  et al.
MAY supply a *child* term of MS:1001194 (quality estimation with decoy database) one or more
times
```

Example cvParams:

```
<cvParam accession="MS:1001316" name="mascot:SigThreshold" cvRef="PSI-MS" value="0.05"/>
<cvParam accession="MS:1001317" name="mascot:MaxProteinHits" cvRef="PSI-MS" value="Auto"/>
<cvParam accession="MS:1001318" name="mascot:ProteinScoringMethod" cvRef="PSI-MS"
<cvParam accession="MS:1001319" name="mascot:MinMSMSThreshold" cvRef="PSI-MS" value="0"/>
<cvParam accession="MS:1001320" name="mascot:ShowHomologousProteinsWithSamePeptides"
<cvParam accession="MS:1001321" name="mascot:ShowHomologousProteinsWithSubsetOfPeptides"
<cvParam accession="MS:1001322" name="mascot:RequireBoldRed" cvRef="PSI-MS" value="0"/>
<cvParam accession="MS:1001323" name="mascot:UseUnigeneClustering" cvRef="PSI-MS"
<cvParam accession="MS:1001324" name="mascot:IncludeErrorTolerantMatches" cvRef="PSI-MS"
<cvParam accession="MS:1001325" name="mascot:ShowDecoyMatches" cvRef="PSI-MS" value="0"/>
```

6.8 Element <AnalysisProtocolCollection>

Definition: The collection of protocols which include the parameters and settings of the performed analyses.

Type: AnalysisProtocolCollectionType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SpectrumIdentificationProtocol	1	unbounded	The parameters and settings of a SpectrumIdentification analysis.
ProteinDetectionProtocol	0	1	The parameters and settings of a ProteinDetection process.

Example Context:

```
<AnalysisProtocolCollection xmlns="http://psidev.info/psi/pi/mzIdentML/1.1">
    <SpectrumIdentificationProtocol analysisSoftware_ref="ID_software" id="SearchProtocol_1">
        <SearchType>
            <cvParam accession="MS:1001083" cvRef="PSI-MS" name="ms-ms search"/>
        </SearchType>
        <AdditionalSearchParams>
            <cvParam accession="MS:1001211" cvRef="PSI-MS" name="parent mass type mono"/>
  ...
</AnalysisProtocolCollection>
```

6.9 Element <AnalysisSampleCollection>

Definition: The samples analysed can optionally be recorded using CV terms for descriptions. If a composite sample has been analysed, the subsample association can be used to build a hierarchical description.

Type: AnalysisSampleCollectionType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
Sample	1	unbounded	A description of the sample analysed by mass spectrometry using CVParams or UserParams. If a composite sample has been analysed, a parent sample should be defined, which references subsamples. This represents any kind of substance used in an experimental workflow, such as whole organisms, cells, DNA, solutions, compounds and experimental substances (gels, arrays etc.).

Example Context:

6.10 Element <AnalysisSoftware>

Definition: The software used for performing the analyses.

Type: AnalysisSoftwareType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
uri	xsd:anyURI	optional	URI of the analysis software e.g. manufacturer's website
version	xsd:string	optional	The version of Software used.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
ContactRole	0	1	The Contact that provided the document instance.
SoftwareName	1	1	The name of the analysis software package, sourced from a CV if available.
Customizations	0	1	Any customizations to the software, such as alternative scoring mechanisms implemented, should be documented here as free text.

Example Context:

```
<AnalysisSoftware id="AS_mascot_server" name="Mascot Server" version="2.3.02"
  uri="http://www.matrixscience.com/search_form_select.html">
  <ContactRole contact_ref="ORG_MSL">
    <Role>
      <cvParam accession="MS:1001267" name="software vendor" cvRef="PSI-MS"/>
    </Role>
  </ContactRole>
  ...
</AnalysisSoftware>
```

6.11 Element <AnalysisSoftwareList>

Definition: The software packages used to perform the analyses.

Type: AnalysisSoftwareListType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
AnalysisSoftware	1	unbounded	The software used for performing the analyses.

Example Context:

```
<AnalysisSoftwareList>
  <AnalysisSoftware id="AS_mascot_server" name="Mascot Server" version="2.3.02"
    uri="http://www.matrixscience.com/search_form_select.html">
    <ContactRole contact_ref="ORG_MSL">
      <Role>
        <cvParam accession="MS:1001267" name="software vendor" cvRef="PSI-MS"/>
      </Role>
  ...
</AnalysisSoftwareList>
```

6.12 Element <AuditCollection>

Definition: The complete set of Contacts (people and organisations) for this file.

Type: AuditCollectionType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
Person	1	1	A person's name and contact details. Any additional information such as the address, contact email etc. should be supplied using CV parameters or user parameters.
Organization	1	1	Organizations are entities like companies, universities, government agencies. Any additional information such as the address, email etc. should be supplied either as CV parameters or as user parameters.

Example Context:

```
<AuditCollection xmlns="http://psidev.info/psi/pi/mzIdentML/1.1">
    <Person firstName="firstname" lastName="secondName" id="PERSON_DOC_OWNER">
        <Affiliation organization_ref="ORG_DOC_OWNER"/>
    </Person>
    <Organization name="myworkplace" id="ORG_DOC_OWNER"/>
</AuditCollection>
```

6.13 Element <BibliographicReference>

Definition: Any bibliographic references associated with the file

Type: BibliographicReferenceType

Attributes:

Attribute Name	Data Type	Use	Definition
authors	xsd:string	optional	The names of the authors of the reference.
doi	xsd:string	optional	The DOI of the referenced publication.
editor	xsd:string	optional	The editor(s) of the reference.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
issue	xsd:string	optional	The issue name or number.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
pages	xsd:string	optional	The page numbers.
publication	xsd:string	optional	The name of the journal, book etc.

publisher	xsd:string	optional	The publisher of the publication.
title	xsd:string	optional	The title of the BibliographicReference.
volume	xsd:string	optional	The volume name or number.
year	xsd:int	optional	The year of publication.

Subelements: none

Example Context:

```
<BibliographicReference
  authors="David N. Perkins, Darryl J. C. Pappin, David M. Creasy, John S. Cottrell" editor=""
  id="10.1002/(SICI)1522-2683(19991201)20:18<3551::AID-ELPS3551>3.0.CO;2-2"
  name="Probability-based protein identification by searching sequence databases using mass
spectrometry data"
  issue="18" pages="3551-3567" publication="Electrophoresis" volume="20" year="1999"
  publisher="Wiley VCH"
  title="Probability-based protein identification by searching sequence databases using mass
spectrometry data"
  ...
</BibliographicReference>
```

6.14 Element <ContactRole>

Definition: The Contact that provided the document instance.

Type: ContactRoleType

Attributes:

Attribute Name	Data Type	Use	Definition
contact_ref	xsd:string	required	When a ContactRole is used, it specifies which Contact the role is associated with.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
Role	1	1	The roles (lab equipment sales, contractor, etc.) the Contact fills.

Example Context:

```
<ContactRole contact_ref="PERSON_DOC_OWNER">
  <Role>
    <cvParam accession="MS:1001271" name="researcher" cvRef="PSI-MS"/>
  </Role>
</ContactRole>
```

6.15 Element <Customizations>

Definition: Any customizations to the software, such as alternative scoring mechanisms implemented, should be documented here as free text.

Type: xsd:string

Attributes: none

Subelements: none

Example Context:

```
<Customizations> No customisations </Customizations>
```

6.16 Element <DBSequence>

Definition: A database sequence from the specified SearchDatabase (nucleic acid or amino acid). If the sequence is nucleic acid, the source nucleic acid sequence should be given in the seq attribute rather than a translated sequence.

Type: DBSequenceType

Attributes:

Attribute Name	Data Type	Use	Definition
accession	xsd:string	required	The unique accession of this sequence.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related

			documents, or a repository) of its use.
length	xsd:int	optional	The length of the sequence as a number of bases or residues.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
searchDatabase_ref	xsd:string	required	The source database of this sequence.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
Seq	0	1	The actual sequence of amino acids or nucleic acid.
cvParam	0	unbounded	A choice of either a cvParam or userParam.
userParam	0	unbounded	A choice of either a cvParam or userParam.

Example Context:

```
<DBSequence id="DBSeq_1_Rnd2psu|NC_LIV_080090" length="16207"
  searchDatabase_ref="SDB_NeoProt_tripledecoy" accession="Rnd2psu|NC_LIV_080090">
  <Seq>DPARMTSRLSEAAPAEESMSAEATFHWYSPMLEYARAAPLWLPSRMLIEHAKPGKREGDSGHLESEATAGPPSPPAPPPEATSLPSYRGLLAFS
QASPPLCPVQLYPPRRRAHFELPLVSSDESQESRCLATCLRGVGLSWDYISPAGTSLVVAEPHGFSGPDLIQGPSADTARAELVPFFSAWSLEERVQSVSW
...
AVIRTHQADALVHEDSRTALGWLASIYXGRSPSVGSDVSDSKFPKFAMKNSTRKKLKGDDSAITSAYVASAGGSSMGILSG</Seq>
  <cvParam accession="MS:1001088" name="protein description" cvRef="PSI-MS"
    value="Rnd2psu|NC_LIV_080090 Decoy sequence, was | organism=Neospora_caninum |
product=hypothetical protein | location=Neo_chrVIIa:229175-282694(-) | length=16207"
  />
</DBSequence>
```

Example cvParams:

```
<cvParam accession="MS:1001088" cvRef="PSI-MS" value="Rnd3psu|NC_LIV_083320 Rnd3psu|NC_LIV_083320
Decoy sequence, was | organism=Neospora_caninum | product=zinc finger (CCCH type) protein, putative |
location=Neo_chrVIIa:3989308-3992771(+) | length=661" name="protein description"/>
```

6.17 Element <DataCollection>

Definition: The collection of input and output data sets of the analyses.

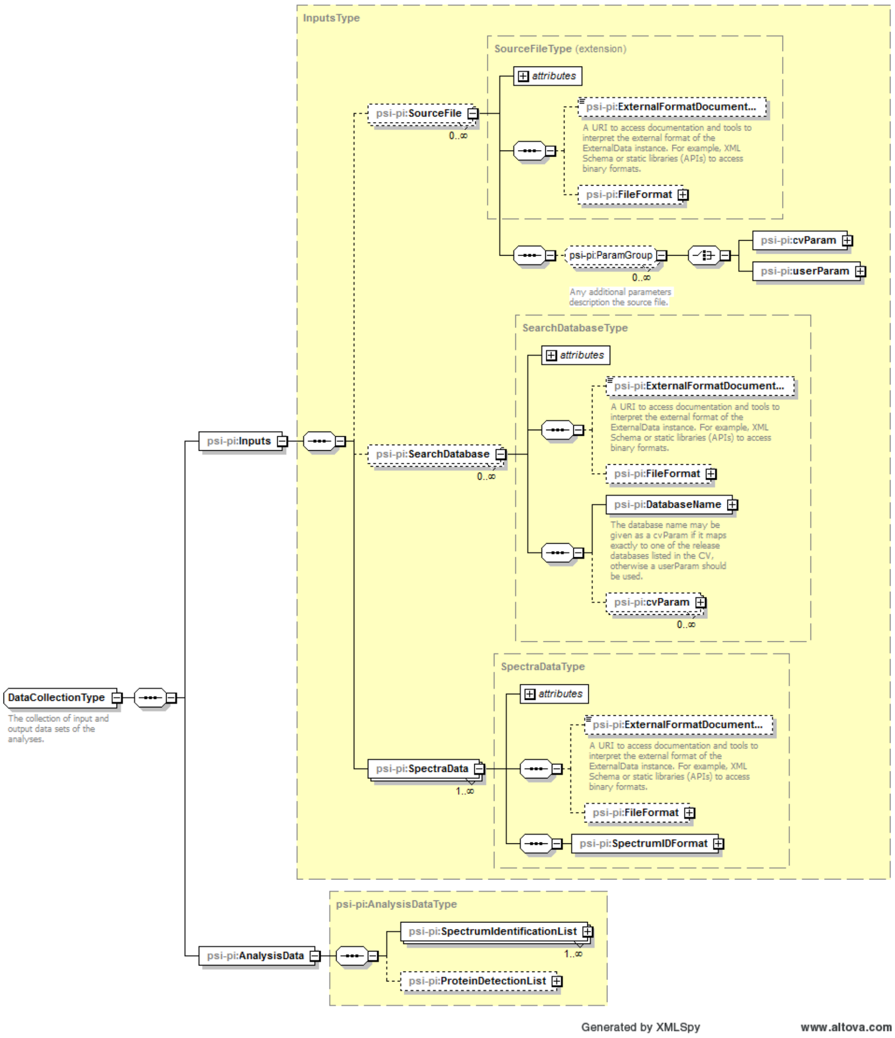
Type: DataCollectionType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
Inputs	1	1	The inputs to the analyses including the databases searched, the spectral data and the source file converted to mzIdentML.
AnalysisData	1	1	Data sets generated by the analyses, including peptide and protein lists.

Graphical Context:


InputsType
SourceFileType (extension)
attributes
psi-pi:SourceFile
0..∞
psi-pi:ExternalFormatDocument...
A URI to access documentation and tools to interpret the external format of the ExternalData instance. For example, XML Schema or static libraries (APIs) to access binary formats.
psi-pi:FileFormat
psi-pi:ParamGroup
0..∞
Any additional parameters description the source file.
psi-pi:cvParam
psi-pi:userParam
SearchDatabaseType
attributes
psi-pi:SearchDatabase
0..∞
psi-pi:ExternalFormatDocument...
A URI to access documentation and tools to interpret the external format of the ExternalData instance. For example, XML Schema or static libraries (APIs) to access binary formats.
psi-pi:FileFormat
psi-pi:DatabaseName
The database name may be given as a cvParam if it maps exactly to one of the release databases listed in the CV, otherwise a userParam should be used.
psi-pi:cvParam
0..∞
psi-pi:Inputs
DataCollectionType
The collection of input and output data sets of the analyses.
SpectraDataType
attributes
psi-pi:ExternalFormatDocument...
A URI to access documentation and tools to interpret the external format of the ExternalData instance. For example, XML Schema or static libraries (APIs) to access binary formats.
psi-pi:SpectraData
1..∞
psi-pi:FileFormat
psi-pi:SpectrumIDFormat
psi-pi:AnalysisDataType
psi-pi:SpectrumIdentificationList
1..∞
psi-pi:AnalysisData
psi-pi:ProteinDetectionList
Generated by XMLSpy
www.altova.com

Example Context:

```
<DataCollection>
<Inputs xmlns="http://psidev.info/psi/pi/mzIdentML/1.1">
    <SourceFile location="build/classes/resources/55merge_omssa.omx" id="SourceFile_1">
        <FileFormat>
            <cvParam accession="MS:1001400" cvRef="PSI-MS" name="OMSSA xml file"/>
        </FileFormat>
    </SourceFile>
  ...
</DataCollection>
```

6.18 Element <DatabaseFilters>

Definition: The specification of filters applied to the database searched.

Type: DatabaseFiltersType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition

Filter	1	unbounded	Filters applied to the search database. The filter MUST include at least one of Include and Exclude. If both are used, it is assumed that inclusion is performed first.

Example Context:

```
<DatabaseFilters>
  <Filter>
    <FilterType>
      <cvParam accession="MS:1001020" name="DB filter taxonomy" cvRef="PSI-MS"/>
    </FilterType>
  </Filter>
</DatabaseFilters>
```

6.19 Element <DatabaseName>

Definition: The database name may be given as a cvParam if it maps exactly to one of the release databases listed in the CV, otherwise a userParam should be used.

Type: ParamType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	1	A single entry from an ontology or a controlled vocabulary.
userParam	1	1	A single user-defined parameter.

Example Context:

```
<DatabaseName>
    <userParam name="D:/Software/Databases/Neospora_3rndTryp/Neo_rndTryp_3times.fasta"/>
</DatabaseName>
```

cvParam Mapping Rules:

```
Path /MzIdentML/DataCollection/Inputs/SearchDatabase/DatabaseName
MAY supply a *child* term of MS:1001013 (database name) one or more times
  e.g.: MS:1001084 (database nr)
  e.g.: MS:1001104 (database SwissProt)
  e.g.: MS:1001142 (database IPI_human)
  e.g.: MS:1001285 (database IPI_mouse)
  e.g.: MS:1001286 (database IPI_rat)
  e.g.: MS:1001287 (database IPI_zebrafish)
  e.g.: MS:1001288 (database IPI_chicken)
  e.g.: MS:1001289 (database IPI_cow)
  e.g.: MS:1001290 (database IPI_arabidopsis)
```

Example cvParams:

```
<cvParam accession="MS:1001073" name="database type amino acid" cvRef="PSI-MS"/>
```

Example userParams:

```
<userParam name="Neo_rndTryp_3times.fasta"/>
<userParam name="D:/Software/Databases/Neospora_3rndTryp/Neo_rndTryp_3times.fasta"/>
```

6.20 Element <DatabaseTranslation>

Definition: A specification of how a nucleic acid sequence database was translated for searching.

Type: DatabaseTranslationType

Attributes:

Attribute Name	Data Type	Use	Definition
frames	listOfAllowedFrames	optional	The frames in which the nucleic acid sequence has been translated as a space separated list

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
TranslationTable	1	unbounded	The table used to translate codons into nucleic acids e.g. by reference to the NCBI translation table.

Example Context:

6.21 Element <Enzyme>

Definition: The details of an individual cleavage enzyme should be provided by giving a regular expression or a CV term if a "standard" enzyme cleavage has been performed.

Type: EnzymeType

Attributes:

Attribute Name	Data Type	Use	Definition
cTermGain	xsd:string with restriction `[A-Za-z0-9 ]+`	optional	Element formula gained at CTerm.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
minDistance	xsd:int	optional	Minimal distance for another cleavage (minimum: 1).
missedCleavages	xsd:int	optional	The number of missed cleavage sites allowed by the search. The attribute MUST be provided if an enzyme has been used.
nTermGain	xsd:string with restriction `[A-Za-z0-9 ]+`	optional	Element formula gained at NTerm.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
semiSpecific	xsd:boolean	optional	Set to true if the enzyme cleaves semi-specifically (i.e. one terminus MUST cleave according to the rules, the other can cleave at any residue), false if the enzyme cleavage is assumed to be specific to both termini (accepting for any missed cleavages).

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SiteRegexp	0	1	Regular expression for specifying the enzyme cleavage site.
EnzymeName	0	1	The name of the enzyme from a CV.

Example Context:

```
<Enzyme id="ENZ_0" cTermGain="OH" nTermGain="H" semiSpecific="0">
  <SiteRegexp><![CDATA[(?<=[KR])(?!P)]]></SiteRegexp>
  <EnzymeName>
    <cvParam accession="MS:1001251" name="Trypsin" cvRef="PSI-MS"/>
  </EnzymeName>
</Enzyme>
```

6.22 Element <EnzymeName>

Definition: The name of the enzyme from a CV.

Type: ParamListType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	1	unbounded	A single user-defined parameter.

Example Context:

```
<EnzymeName>
    <cvParam accession="MS:1001251" cvRef="PSI-MS" name="Trypsin"/>
</EnzymeName>
```

cvParam Mapping Rules:

```
Path
/MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/Enzymes/Enzyme/EnzymeName
MAY supply a *child* term of MS:1001045 (cleavage agent name) only once
  e.g.: MS:1001091 (NoEnzyme)
  e.g.: MS:1001251 (Trypsin)
  e.g.: MS:1001303 (Arg-C)
  e.g.: MS:1001304 (Asp-N)
  e.g.: MS:1001305 (Asp-N_ambic)
  e.g.: MS:1001306 (Chymotrypsin)
  e.g.: MS:1001307 (CNBr)
  e.g.: MS:1001308 (Formic_acid)
  e.g.: MS:1001309 (Lys-C)
  e.g.: MS:1001310 (Lys-C/P)
  et al.
```

Example cvParams:

```
<cvParam accession="MS:1001251" name="Trypsin" cvRef="PSI-MS"/>
```

6.23 Element <Enzymes>

Definition: The list of enzymes used in experiment

Type: EnzymesType

Attributes:

Attribute Name	Data Type	Use	Definition
independent	xsd:boolean	optional	If there are multiple enzymes specified, this attribute is set to true if cleavage with different enzymes is performed independently.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
Enzyme	1	unbounded	The details of an individual cleavage enzyme should be provided by giving a regular expression or a CV term if a "standard" enzyme cleavage has been performed.

Example Context:

```
<Enzymes>
  <Enzyme id="ENZ_0" cTermGain="OH" nTermGain="H" semiSpecific="0">
    <SiteRegexp><![CDATA[(?<=[KR])(?!P)]]></SiteRegexp>
    <EnzymeName>
      <cvParam accession="MS:1001251" name="Trypsin" cvRef="PSI-MS"/>
    </EnzymeName>
  </Enzyme>
  ...
</Enzymes>
```

6.24 Element <Exclude>

Definition: All sequences fulfilling the specifed criteria are excluded.

Type: ParamListType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	1	unbounded	A single user-defined parameter.

Example Context:

cvParam Mapping Rules:

```
Path
/MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/DatabaseFilters/Filter/Exclude
MAY supply a *child* term of MS:1001512 (Sequence database filters) one or more times
  e.g.: MS:1001090 (taxonomy nomenclature)
  e.g.: MS:1001201 (DB MW filter maximum)
  e.g.: MS:1001202 (DB MW filter minimum)
  e.g.: MS:1001203 (DB PI filter maximum)
  e.g.: MS:1001204 (DB PI filter minimum)
  e.g.: MS:1001467 (taxonomy: NCBI TaxID)
```

```
e.g.: MS:1001468 (taxonomy: common name)
e.g.: MS:1001469 (taxonomy: scientific name)
e.g.: MS:1001470 (taxonomy: Swiss-Prot ID)
e.g.: MS:1001513 (DB sequence filter pattern)
et al.
```

6.25 Element <ExternalFormatDocumentation>

Definition: A URI to access documentation and tools to interpret the external format of the ExternalData instance. For example, XML Schema or static libraries (APIs) to access binary formats.

Type: xsd:anyURI

Attributes: none

Subelements: none

Example Context:

6.26 Element <FileFormat>

Definition: The format of the ExternalData file, for example "tiff" for image files.

Type: FileFormatType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	1	A single entry from an ontology or a controlled vocabulary.

Example Context:

```
<FileFormat>
    <cvParam accession="MS:1001062" cvRef="PSI-MS" name="Mascot MGF file"/>
</FileFormat>
```

cvParam Mapping Rules:

```
Path /MzIdentML/DataCollection/Inputs/SearchDatabase/FileFormat
MUST supply a *child* term of MS:1001347 (database file formats) one or more times
  e.g.: MS:1001348 (FASTA format)
  e.g.: MS:1001349 (ASN.1)
  e.g.: MS:1001350 (NCBI *.p*)
  e.g.: MS:1001351 (clustal aln)
  e.g.: MS:1001352 (embl em)
  e.g.: MS:1001353 (NBRF PIR)
  e.g.: MS:1001462 (PEFF format)
Path /MzIdentML/DataCollection/Inputs/SourceFile/FileFormat
MUST supply a *child* term of MS:1001040 (intermediate analysis format) only once
  e.g.: MS:1000742 (Bioworks SRF file)
  e.g.: MS:1001107 (data stored in database)
  e.g.: MS:1001199 (Mascot DAT file)
  e.g.: MS:1001200 (Sequest out file)
  e.g.: MS:1001242 (Sequest out folder)
  e.g.: MS:1001243 (Sequest summary)
  e.g.: MS:1001275 (ProteinScape SearchEvent)
  e.g.: MS:1001276 (ProteinScape Gel)
  e.g.: MS:1001399 (OMSSA csv file)
  e.g.: MS:1001400 (OMSSA xml file)
  et al.
Path /MzIdentML/DataCollection/Inputs/SpectraData/FileFormat
MUST supply a *child* term of MS:1000560 (mass spectrometer file format) one or more times
  e.g.: MS:1000526 (Waters raw file)
  e.g.: MS:1000562 (ABI WIFF file)
  e.g.: MS:1000563 (Thermo RAW file)
  e.g.: MS:1000564 (PSI mzData file)
  e.g.: MS:1000565 (Micromass PKL file)
  e.g.: MS:1000566 (ISB mzXML file)
  e.g.: MS:1000567 (Bruker/Agilent YEP file)
  e.g.: MS:1000584 (mzML file)
  e.g.: MS:1000613 (DTA file)
  e.g.: MS:1000614 (ProteinLynx Global Server mass spectrum XML file)
  et al.
```

Example cvParams:

```
<cvParam accession="MS:1001199" name="Mascot DAT file" cvRef="PSI-MS"/>
<cvParam accession="MS:1001348" name="FASTA format" cvRef="PSI-MS"/>
<cvParam accession="MS:1001062" name="Mascot MGF file" cvRef="PSI-MS"/>
<cvParam accession="MS:1001400" cvRef="PSI-MS" name="OMSSA xml file"/>
```

6.27 Element <Filter>

Definition: Filters applied to the search database. The filter MUST include at least one of Include and Exclude. If both are used, it is assumed that inclusion is performed first.

Type: FilterType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
FilterType	1	1	The type of filter e.g. database taxonomy filter, pi filter, mw filter
Include	0	1	All sequences fulfilling the specifed criteria are included.
Exclude	0	1	All sequences fulfilling the specifed criteria are excluded.

Example Context:

```
<Filter>
  <FilterType>
    <cvParam accession="MS:1001020" name="DB filter taxonomy" cvRef="PSI-MS"/>
  </FilterType>
</Filter>
```

6.28 Element <FilterType>

Definition: The type of filter e.g. database taxonomy filter, pi filter, mw filter

Type: ParamType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	1	A single entry from an ontology or a controlled vocabulary.
userParam	1	1	A single user-defined parameter.

Example Context:

```
<FilterType>
  <cvParam accession="MS:1001020" name="DB filter taxonomy" cvRef="PSI-MS"/>
</FilterType>
```

cvParam Mapping Rules:

```
Path
/MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/DatabaseFilters/Filter/FilterType
MUST supply a *child* term of MS:1001511 (Sequence database filter types) one or more times
  e.g.: MS:1001020 (DB filter taxonomy)
  e.g.: MS:1001021 (DB filter on accession numbers)
  e.g.: MS:1001022 (DB MW filter)
  e.g.: MS:1001023 (DB PI filter)
  e.g.: MS:1001027 (DB filter on sequence pattern)
```

Example cvParams:

```
<cvParam accession="MS:1001020" name="DB filter taxonomy" cvRef="PSI-MS"/>
```

6.29 Element <FragmentArray>

Definition: An array of values for a given type of measure and for a particular ion type, in parallel to the index of ions identified.

Type: FragmentArrayType

Attributes:

Attribute Name	Data Type	Use	Definition
measure_ref	xsd:string	required	A reference to the Measure defined in the FragmentationTable
values	listOfFloats	required	The values of this particular measure, corresponding to the index defined in ion type

Subelements: none

Example Context:

6.30 Element <FragmentTolerance>

Definition: The tolerance of the search given as a plus and minus value with units.

Type: ToleranceType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.

Example Context:

```
<FragmentTolerance>
  <cvParam accession="MS:1001412" name="search tolerance plus value" value="0.8"
    cvRef="PSI-MS" unitAccession="UO:0000221" unitName="dalton" unitCvRef="UO"/>
  <cvParam accession="MS:1001413" name="search tolerance minus value" value="0.8"
    cvRef="PSI-MS" unitAccession="UO:0000221" unitName="dalton" unitCvRef="UO"/>
</FragmentTolerance>
```

cvParam Mapping Rules:

```
Path /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/FragmentTolerance
MUST supply term MS:1001412 (search tolerance plus value) only once
MUST supply term MS:1001413 (search tolerance minus value) only once
```

Example cvParams:

```
<cvParam accession="MS:1001412" name="search tolerance plus value" value="0.8"
<cvParam accession="MS:1001413" name="search tolerance minus value" value="0.8"
```

6.31 Element <Fragmentation>

Definition: The product ions identified in this result.

Type: FragmentationType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
IonType	1	unbounded	IonType defines the index of fragmentation ions being reported, importing a CV term for the type of ion e.g. b ion. Example: if b3 b7 b8 and b10 have been identified, the index attribute will contain 3 7 8 10, and the corresponding values will be reported in parallel arrays below

Example Context:

6.32 Element <FragmentationTable>

Definition: Contains the types of measures that will be reported in generic arrays for each SpectrumIdentificationItem e.g. product ion m/z, product ion intensity, product ion m/z error

Type: FragmentationTableType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
Measure	1	unbounded	References to CV terms defining the measures about product ions to be reported in SpectrumIdentificationItem

Example Context:

```
<FragmentationTable>
    <Measure id="Measure_MZ">
        <cvParam accession="MS:1001225" cvRef="PSI-MS" unitCvRef="PSI-MS" unitName="m/z"
unitAccession="MS:1000040" name="product ion m/z"/>
    </Measure>
    <Measure id="Measure_Int">
        <cvParam accession="MS:1001226" cvRef="PSI-MS" name="product ion intensity"/>
        </Measure>
      ...
    </FragmentationTable>
```

6.33 Element <Include>

Definition: All sequences fulfilling the specifed criteria are included.

Type: ParamListType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	1	unbounded	A single user-defined parameter.

Example Context:

cvParam Mapping Rules:

```
Path
/MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/DatabaseFilters/Filter/Include
MAY supply a *child* term of MS:1001512 (Sequence database filters) one or more times
  e.g.: MS:1001090 (taxonomy nomenclature)
  e.g.: MS:1001201 (DB MW filter maximum)
  e.g.: MS:1001202 (DB MW filter minimum)
  e.g.: MS:1001203 (DB PI filter maximum)
  e.g.: MS:1001204 (DB PI filter minimum)
  e.g.: MS:1001467 (taxonomy: NCBI TaxID)
  e.g.: MS:1001468 (taxonomy: common name)
  e.g.: MS:1001469 (taxonomy: scientific name)
  e.g.: MS:1001470 (taxonomy: Swiss-Prot ID)
  e.g.: MS:1001513 (DB sequence filter pattern)
  et al.
```

6.34 Element <InputSpectra>

Definition: One of the spectra data sets used.

Type: InputSpectraType

Attributes:

Attribute Name	Data Type	Use	Definition
spectraData_ref	xsd:string	optional	A reference to the SpectraData element which locates the input spectra to an external file.

Subelements: none

Example Context:

```
<InputSpectra spectraData_ref="SID_1"/>
```

6.35 Element <InputSpectrumIdentifications>

Definition: The lists of spectrum identifications that are input to the protein detection process.

Type: InputSpectrumIdentificationsType

Attributes:

Attribute Name	Data Type	Use	Definition
spectrumIdentificationList_ref	xsd:string	required	A reference to the list of spectrum identifications that were input to the process.

Subelements: none

Example Context:

```
<InputSpectrumIdentifications spectrumIdentificationList_ref="SIL_1"/>
```

6.36 Element <Inputs>

Definition: The inputs to the analyses including the databases searched, the spectral data and the source file converted to mzIdentML.

Type: InputsType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SourceFile	0	unbounded	A file from which this mzIdentML instance was created.
SearchDatabase	0	unbounded	A database for searching mass spectra. Examples include a set of amino acid sequence entries, or annotated spectra libraries.
SpectraData	1	unbounded	A data set containing spectra data (consisting of one or more spectra).

Example Context:

```
<Inputs xmlns="http://psidev.info/psi/pi/mzIdentML/1.1">
    <SourceFile location="build/classes/resources/55merge_omssa.omx" id="SourceFile_1">
        <FileFormat>
            <cvParam accession="MS:1001400" cvRef="PSI-MS" name="OMSSA xml file"/>
        </FileFormat>
    </SourceFile>
    <SearchDatabase numDatabaseSequences="22348"
location="D:/Software/Databases/Neospora_3rndTryp/Neo_rndTryp_3times.fasta" id="SearchDB_1">
  ...
</Inputs>
```

6.37 Element <IonType>

Definition: IonType defines the index of fragmentation ions being reported, importing a CV term for the type of ion e.g. b ion. Example: if b3 b7 b8 and b10 have been identified, the index attribute will contain 3 7 8 10, and the corresponding values will be reported in parallel arrays below

Type: IonTypeType

Attributes:

Attribute Name	Data Type	Use	Definition
charge	xsd:int	required	The charge of the identified fragmentation ions.
index	listOfIntegers	optional	The index of ions identified as integers, following standard notation for a-c, x-z e.g. if b3 b5 and b6 have been identified, the index would store "3 5 6". For internal ions, the index contains pairs defining the start and end point - see specification document for examples. For immonium ions, the index is the position of the identified ion within the peptide sequence - if the peptide contains the same amino acid in multiple positions that cannot be distinguished, all positions should be given.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
FragmentArray	0	unbounded	An array of values for a given type of measure and for a particular ion type, in parallel to the index of ions identified.
cvParam	1	1	A single entry from an ontology or a controlled vocabulary.

Example Context:

cvParam Mapping

```
Path
/MzIdentML/DataCollection/AnalysisData/SpectrumIdentificationList/SpectrumIdentificationResult/Spectrum
IdentificationItem/Fragmentation/IonType
MAY supply a *child* term of MS:1001221 (fragmentation information) one or more times
```

Rules:

```
e.g.: MS:1000903 (product ion series ordinal)
e.g.: MS:1000904 (product ion m/z delta)
e.g.: MS:1000926 (product interpretation rank)
e.g.: MS:1001220 (frag: y ion)
e.g.: MS:1001222 (frag: b ion - H2O)
e.g.: MS:1001223 (frag: y ion - H2O)
e.g.: MS:1001224 (frag: b ion)
e.g.: MS:1001225 (product ion m/z)
e.g.: MS:1001226 (product ion intensity)
e.g.: MS:1001227 (product ion m/z error)
et al.
```

6.38 Element <MassTable>

Definition: The masses of residues used in the search.

Type: MassTableType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
msLevel	listOfIntegers	required	The MS spectrum that the MassTable refers to e.g. "1" for MS1 "2" for MS2 or "1 2" for MS1 or MS2.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
Residue	0	unbounded	The specification of a single residue within the mass table.
AmbiguousResidue	0	unbounded	Ambiguous residues e.g. X can be specified by the Code attribute and a set of parameters for example giving the different masses that will be used in the search.
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	0	unbounded	A single user-defined parameter.

Example Context:

```
<MassTable id="MT" msLevel="1 2">
  <Residue code="A" mass="71.037114"/>
  <Residue code="C" mass="103.009185"/>
  <Residue code="D" mass="115.026943"/>
  <Residue code="E" mass="129.042593"/>
  <Residue code="F" mass="147.068414"/>
  <Residue code="G" mass="57.021464"/>
  ...
</MassTable>
```

cvParam Mapping Rules:

```
Path /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/MassTable
MAY supply a *child* term of MS:1001354 (mass table options) one or more times
  e.g.: MS:1001346 (AAIndex mass table)
```

6.39 Element <Measure>

Definition: References to CV terms defining the measures about product ions to be reported in SpectrumIdentificationItem

Type: MeasureType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.

name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.

Example Context:

```
<Measure id="Measure_Error">
    <cvParam accession="MS:1001227" cvRef="PSI-MS" unitCvRef="PSI-MS" unitName="m/z"
unitAccession="MS:1000040" name="product ion m/z error"/>
</Measure>
```

cvParam Mapping Rules:

```
Path /MzIdentML/DataCollection/AnalysisData/SpectrumIdentificationList/FragmentationTable/Measure
MUST supply term MS:1001226 (product ion intensity) only once
MUST supply term MS:1001225 (product ion m/z) only once
MUST supply term MS:1001227 (product ion m/z error) only once
```

Example cvParams:

```
<cvParam cvRef="PSI-MS" accession="MS:1001225" name="product ion m/z"/>
<cvParam cvRef="PSI-MS" accession="MS:1001226" name="product ion intensity"/>
<cvParam cvRef="PSI-MS" accession="MS:1001227" name="product ion m/z error"
```

6.40 Element <Modification>

Definition: A molecule modification specification. If n modifications have been found on a peptide, there should be n instances of Modification. If multiple modifications are provided as cvParams, it is assumed that the modification is ambiguous i.e. one modification or another. A cvParam MUST be provided with the identification of the modification sourced from a suitable CV e.g. UNIMOD. If the modification is not present in the CV (and this will be checked by the semantic validator within a given tolerance window), there is a “unknown modification” CV term that MUST be used instead. A neutral loss should be defined as an additional CVParam within Modification. If more complex information should be given about neutral losses (such as presence/absence on particular product ions), this can additionally be encoded within the FragmentationArray.

Type: ModificationType

Attributes:

Attribute Name	Data Type	Use	Definition
avgMassDelta	xsd:double	optional	Atomic mass delta considering the natural distribution of isotopes in Daltons.
location	xsd:int	optional	Location of the modification within the peptide - position in peptide sequence, counted from the N-terminus residue, starting at position 1. Specific modifications to the N-terminus should be given the location 0. Modification to the C-terminus should be given as peptide length + 1. If the modification location is unknown e.g. for PMF data, this attribute should be omitted.
monoisotopicMassDelta	xsd:double	optional	Atomic mass delta when assuming only the most common isotope of elements in Daltons.
residues	listOfChars	optional	Specification of the residue (amino acid) on which the modification occurs. If multiple values are given, it is assumed that the exact residue modified is unknown i.e. the modification is to ONE of the residues listed. Multiple residues would usually only be specified for PMF data.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.

Example Context:

```
<Modification location="11" residues="M" monoisotopicMassDelta="15.994915">
  <cvParam accession="UNIMOD:35" name="Oxidation" cvRef="UNIMOD"/>
  <cvParam accession="MS:1001524" name="fragment neutral loss" cvRef="PSI-MS"
```

```
    value="63.998285" unitAccession="UO:0000221" unitName="dalton" unitCvRef="UO"/>
</Modification>
```

Example cvParams:

```
<cvParam accession="UNIMOD:4" name="Carbamidomethyl" cvRef="UNIMOD"/>
<cvParam accession="UNIMOD:35" name="Oxidation" cvRef="UNIMOD"/>
<cvParam accession="MS:1001524" name="fragment neutral loss" cvRef="PSI-MS" value="0"
```

6.41 Element <ModificationParams>

Definition: The specification of static/variable modifications (e.g. Oxidation of Methionine) that are to be considered in the spectra search.

Type: ModificationParamsType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SearchModification	1	unbounded	Specification of a search modification as parameter for a spectra search. Contains the name of the modification, the mass, the specificity and whether it is a static modification.

Example Context:

```
<ModificationParams>
    <SearchModification residues="C" massDelta="57.021465" fixedMod="true">
        <cvParam accession="UNIMOD:4" cvRef="UNIMOD" name="Carbamidomethyl"/>
    </SearchModification>
    <SearchModification residues="M" massDelta="15.994915" fixedMod="false">
        <cvParam accession="UNIMOD:35" cvRef="UNIMOD" name="Oxidation"/>
    </SearchModification>
  ...
</ModificationParams>
```

6.42 Element <Organization>

Definition: Organizations are entities like companies, universities, government agencies. Any additional information such as the address, email etc. should be supplied either as CV parameters or as user parameters.

Type: OrganizationType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	0	unbounded	A choice of either a cvParam or userParam.
userParam	0	unbounded	A choice of either a cvParam or userParam.
Parent	0	1	The containing organization (the university or business which a lab belongs to, etc.)

Example Context:

```
<Organization id="ORG_MSL" name="Matrix Science Limited"/>
```

6.43 Element <Parent>

Definition: The containing organization (the university or business which a lab belongs to, etc.)

Type: ParentOrganizationType

Attributes:

Attribute Name	Data Type	Use	Definition

organization_ref	xsd:string	required	A reference to the organization this contact belongs to.

Subelements: none

Example Context:

6.44 Element <ParentTolerance>

Definition: The tolerance of the search given as a plus and minus value with units.

Type: ToleranceType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.

Example Context:

```
<ParentTolerance>
  <cvParam accession="MS:1001412" name="search tolerance plus value" value="1.5"
    cvRef="PSI-MS" unitAccession="UO:0000221" unitName="dalton" unitCvRef="UO"/>
  <cvParam accession="MS:1001413" name="search tolerance minus value" value="1.5"
    cvRef="PSI-MS" unitAccession="UO:0000221" unitName="dalton" unitCvRef="UO"/>
</ParentTolerance>
```

cvParam Mapping Rules:

```
Path /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/ParentTolerance
MUST supply term MS:1001412 (search tolerance plus value) only once
MUST supply term MS:1001413 (search tolerance minus value) only once
```

Example cvParams:

```
<cvParam accession="MS:1001412" name="search tolerance plus value" value="1.5"
<cvParam accession="MS:1001413" name="search tolerance minus value" value="1.5"
```

6.45 Element <Peptide>

Definition: One (poly)peptide (a sequence with modifications). The combination of Peptide sequence and modifications MUST be unique in the file.

Type: PeptideType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
PeptideSequence	1	1	The amino acid sequence of the (poly)peptide. If a substitution modification has been found, the original sequence should be reported.
Modification	0	unbounded	A molecule modification specification. If n modifications have been found on a peptide, there should be n instances of Modification. If multiple modifications are provided as cvParams, it is assumed that the modification is ambiguous i.e. one modification or another. A cvParam MUST be provided with the identification of the modification sourced from a suitable CV e.g. UNIMOD. If the modification is not present in the CV (and this will be checked

			by the semantic validator within a given tolerance window), there is a “unknown modification” CV term that MUST be used instead. A neutral loss should be defined as an additional CVParam within Modification. If more complex information should be given about neutral losses (such as presence/absence on particular product ions), this can additionally be encoded within the FragmentationArray.
SubstitutionModification	0	unbounded	A modification where one residue is substituted by another (amino acid change).
cvParam	0	unbounded	A choice of either a cvParam or userParam.
userParam	0	unbounded	A choice of either a cvParam or userParam.

Example Context:

```
<Peptide id="TVDVGMGGVDLANLKACSGSGVSQELHIWGK_000000100000000000000000000000000">
  <PeptideSequence>TVDVGMGGVDLANLKACSGSGVSQELHIWGK</PeptideSequence>
  <Modification location="6" residues="M" monoisotopicMassDelta="15.994915">
    <cvParam accession="UNIMOD:35" name="Oxidation" cvRef="UNIMOD"/>
    <cvParam accession="MS:1001524" name="fragment neutral loss" cvRef="PSI-MS"
      value="63.998285" unitAccession="UO:0000221" unitName="dalton" unitCvRef="UO"/>
  </Modification>
  ...
</Peptide>
```

6.46 Element <PeptideEvidence>

Definition: PeptideEvidence links a specific Peptide element to a specific position in a DBSequence. There MUST only be one PeptideEvidence item per Peptide-to-DBSequence-position.

Type: PeptideEvidenceType

Attributes:

Attribute Name	Data Type	Use	Definition
dBSequence_ref	xsd:string	required	A reference to the protein sequence in which the specified peptide has been linked.
end	xsd:int	optional	The index position of the last amino acid of the peptide inside the protein sequence, where the first amino acid of the protein sequence is position 1. Must be provided unless this is a de novo search.
frame	allowed_frames	optional	The translation frame of this sequence if this is PeptideEvidence derived from nucleic acid sequence
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
isDecoy	xsd:boolean	optional	Set to true if the peptide is matched to a decoy sequence.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
peptide_ref	xsd:string	required	A reference to the identified

			(poly)peptide sequence in the Peptide element.
post	xsd:string with restriction [ABCDEFGHIJKLMNOPQRSTUVWXYZ?\-]{1}	optional	Post flanking residue. If the peptide is C-terminal, post="-" and not post="". If for any reason it is unknown (e.g. denovo), post="?" should be used.
pre	xsd:string with restriction [ABCDEFGHIJKLMNOPQRSTUVWXYZ?\-]{1}	optional	Previous flanking residue. If the peptide is N-terminal, pre="-" and not pre="". If for any reason it is unknown (e.g. denovo), pre="?" should be used.
start	xsd:int	optional	Start position of the peptide inside the protein sequence, where the first amino acid of the protein sequence is position 1. Must be provided unless this is a de novo search.
translationTable_ref	xsd:string	optional	A reference to the translation table used if this is PeptideEvidence derived from nucleic acid sequence

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	0	unbounded	A choice of either a cvParam or userParam.
userParam	0	unbounded	A choice of either a cvParam or userParam.

Example Context:

```
<PeptideEvidence id="DDDHSDQGGEVQGGR_0000000000000000_1_Rnd2psu|NC_LIV_123110_2144_2158"
  start="2144" end="2158" pre="R" post="G" isDecoy="false"
  dBSequence_ref="DBSeq_1_Rnd2psu|NC_LIV_123110" peptide_ref="DDDHSDQGGEVQGGR_0000000000000000"/>
<PeptideEvidence id="SVAGKGLADEHTACR_0000000000000000_1_Rnd3psu|NC_LIV_114310_1396_1410"
  start="1396" end="1410" pre="R" post="E" isDecoy="false"
  dBSequence_ref="DBSeq_1_Rnd3psu|NC_LIV_114310" peptide_ref="SVAGKGLADEHTACR_0000000000000000"/>
<PeptideEvidence id="FASCCGEDDGEAPR_0000000000000000_1_psu|NC_LIV_113540_2484_2497" start="2484"
  ...
</PeptideEvidence>
```

6.47 Element <PeptideEvidenceRef>

Definition: Reference to the PeptideEvidence element identified. If a specific sequence can be assigned to multiple proteins and or positions in a protein all possible PeptideEvidence elements should be referenced here.

Type: PeptideEvidenceRefType

Attributes:

Attribute Name	Data Type	Use	Definition
peptideEvidence_ref	xsd:string	required	A reference to the PeptideEvidenceItem element(s).

Subelements: none

Example Context:

```
<PeptideEvidenceRef
  peptideEvidence_ref="SGALR_0000000_1_Rnd1psu|NC_LIV_081600_1089_1093"/>
<PeptideEvidenceRef peptideEvidence_ref="SGALR_0000000_1_Rnd1psu|NC_LIV_020100_977_981"/>
<PeptideEvidenceRef peptideEvidence_ref="SGALR_0000000_1_Rnd1psu|NC_LIV_102910_725_729"/>
<PeptideEvidenceRef peptideEvidence_ref="SGALR_0000000_1_Rnd1psu|NC_LIV_122850_402_406"/>
<PeptideEvidenceRef peptideEvidence_ref="SGALR_0000000_1_Rnd1psu|NC_LIV_060960_32_36"/>
<PeptideEvidenceRef peptideEvidence_ref="SGALR_0000000_1_Rnd2psu|NC_LIV_145280_820_824"/>
  ...
</PeptideEvidenceRef>
```

Example cvParams:

```
<cvParam accession="MS:1001171" name="mascot:score" cvRef="PSI-MS" value="13.49"/>
<cvParam accession="MS:1001172" name="mascot:expectation value" cvRef="PSI-MS"
<cvParam accession="MS:1001363" name="peptide unique to one protein" cvRef="PSI-MS"/>
<cvParam accession="MS:1001371" name="mascot:identity threshold" cvRef="PSI-MS" value="42"/>
```

```
<cvParam accession="MS:1001370" name="mascot:homology threshold" cvRef="PSI-MS" value="24"/>
<cvParam accession="MS:1001030" name="number of peptide seqs compared to each spectrum"
<cvParam accession="MS:1000796" name="spectrum title" cvRef="PSI-MS"
```

6.48 Element <PeptideHypothesis>

Definition: Peptide evidence on which this ProteinHypothesis is based by reference to a PeptideEvidence element.

Type: PeptideHypothesisType

Attributes:

Attribute Name	Data Type	Use	Definition
peptideEvidence_ref	xsd:string	required	A reference to the PeptideEvidence element on which this hypothesis is based.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SpectrumIdentificationItemRef	1	unbounded	Reference(s) to the SpectrumIdentificationItem element(s) that support the given PeptideEvidence element. Using these references it is possible to indicate which spectra were actually accepted as evidence for this peptide identification in the given protein.

Example Context:

```
<PeptideHypothesis
  peptideEvidence_ref="KDLYGNVVLSGGTTMYEGIGER_0000000000000000100000000_1_psu|NC_LIV_020800_292_313">
  <SpectrumIdentificationItemRef spectrumIdentificationItem_ref="SII_303_1"/>
  <SpectrumIdentificationItemRef spectrumIdentificationItem_ref="SII_304_1"/>
</PeptideHypothesis>
```

Example cvParams:

```
<cvParam accession="MS:1001171" name="mascot:score" cvRef="PSI-MS"
<cvParam accession="MS:1001093" name="sequence coverage" cvRef="PSI-MS" value="11"/>
<cvParam accession="MS:1001097" name="distinct peptide sequences" cvRef="PSI-MS"
```

6.49 Element <PeptideSequence>

Definition: The amino acid sequence of the (poly)peptide. If a substitution modification has been found, the original sequence should be reported.

Type: sequence

Attributes: none

Subelements: none

Example Context:

```
<PeptideSequence>GELLGLGGVSGCPLRSGGTEAGGALEQPPLKPK</PeptideSequence>
```

6.50 Element <Person>

Definition: A person's name and contact details. Any additional information such as the address, contact email etc. should be supplied using CV parameters or user parameters.

Type: PersonType

Attributes:

Attribute Name	Data Type	Use	Definition
firstName	xsd:string	optional	The Person's first name.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
lastName	xsd:string	optional	The Person's last/family name.
midInitials	xsd:string	optional	The Person's middle initial.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a

			human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	0	unbounded	A choice of either a cvParam or userParam.
userParam	0	unbounded	A choice of either a cvParam or userParam.
Affiliation	0	unbounded	The organization a person belongs to.

Example Context:

```
<Person firstName="firstname" lastName="secondName" id="PERSON_DOC_OWNER">
    <Affiliation organization_ref="ORG_DOC_OWNER"/>
</Person>
```

6.51 Element <ProteinAmbiguityGroup>

Definition: A set of logically related results from a protein detection, for example to represent conflicting assignments of peptides to proteins.

Type: ProteinAmbiguityGroupType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
ProteinDetectionHypothesis	1	unbounded	A single result of the ProteinDetection analysis (i.e. a protein).
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	0	unbounded	A single user-defined parameter.

Example Context:

```
<ProteinAmbiguityGroup id="PAG_hit_2">
  <ProteinDetectionHypothesis id="PDH_psu|NC_LIV_105380_0"
    dBSequence_ref="DBSeq_1_psu|NC_LIV_105380" passThreshold="true">
    <PeptideHypothesis

peptideEvidence_ref="VIDENFGLVEGLMTTVHAATGTQK_0000000000000100000000000_1_psu|NC_LIV_105380_842_865">
      <SpectrumIdentificationItemRef spectrumIdentificationItem_ref="SII_307_1"/>
    </PeptideHypothesis>
  ...
</ProteinAmbiguityGroup>
```

6.52 Element <ProteinDetection>

Definition: An Analysis which assembles a set of peptides (e.g. from a spectra search analysis) to proteins.

Type: ProteinDetectionType

Attributes:

Attribute Name	Data Type	Use	Definition
activityDate	xsd:dateTime	optional	When the protocol was applied.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
proteinDetectionList_ref	xsd:string	required	A reference to the ProteinDetectionList in the DataCollection section.

proteinDetectionProtocol_ref	xsd:string	required	A reference to the detection protocol used for this ProteinDetection.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
InputSpectrumIdentifications	1	unbounded	The lists of spectrum identifications that are input to the protein detection process.

Example Context:

```
<ProteinDetection id="PD_1" proteinDetectionProtocol_ref="PDP_MascotParser_1"
  proteinDetectionList_ref="PDL_1" activityDate="2011-03-25T13:33:51">
  <InputSpectrumIdentifications spectrumIdentificationList_ref="SIL_1"/>
</ProteinDetection>
```

6.53 Element <ProteinDetectionHypothesis>

Definition: A single result of the ProteinDetection analysis (i.e. a protein).

Type: ProteinDetectionHypothesisType

Attributes:

Attribute Name	Data Type	Use	Definition
dBSequence_ref	xsd:string	optional	A reference to the corresponding DBSequence entry. This optional and redundant, because the PeptideEvidence elements referenced from here also map to the DBSequence.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
passThreshold	xsd:boolean	required	Set to true if the producers of the file has deemed that the ProteinDetectionHypothesis has passed a given threshold or been validated as correct. If no such threshold has been set, value of true should be given for all results.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
PeptideHypothesis	1	unbounded	Peptide evidence on which this ProteinHypothesis is based by reference to a PeptideEvidence element.
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	0	unbounded	A single user-defined parameter.

Example Context:

```
<ProteinDetectionHypothesis id="PDH_psu|NC_LIV_105380_0"
  dBSequence_ref="DBSeq_1_psu|NC_LIV_105380" passThreshold="true">
  <PeptideHypothesis

peptideEvidence_ref="VIDENFGLVEGLMTTVHAATGTQK_000000000000010000000000000_1_psu|NC_LIV_105380_842_865">
    <SpectrumIdentificationItemRef spectrumIdentificationItem_ref="SII_307_1"/>
  </PeptideHypothesis>
  <cvParam accession="MS:1001171" name="mascot:score" cvRef="PSI-MS" value="99.72"/>
  ...
</ProteinDetectionHypothesis>
```

cvParam Mapping Rules:

```
Path
/MzIdentML/DataCollection/AnalysisData/ProteinDetectionList/ProteinAmbiguityGroup/ProteinDetectionHypo
thesis
MAY supply a *child* term of MS:1001153 (search engine specific score) one or more times
  e.g.: MS:1001154 (Sequest:probability)
  e.g.: MS:1001155 (Sequest:xcorr)
  e.g.: MS:1001156 (Sequest:deltacn)
  e.g.: MS:1001157 (Sequest:sp)
  e.g.: MS:1001158 (Sequest:Uniq)
  e.g.: MS:1001159 (Sequest:expectation value)
  e.g.: MS:1001160 (Sequest:sf)
```

```
    e.g.: MS:1001161 (Sequest:matched ions)
    e.g.: MS:1001162 (Sequest:total ions)
    e.g.: MS:1001163 (Sequest:consensus score)
    et al.
  MAY supply a *child* term of MS:1001060 (quality estimation method details) one or more times
    e.g.: MS:1001058 (quality estimation by manual validation)
    e.g.: MS:1001194 (quality estimation with decoy database)
    e.g.: MS:1001447 (prot:FDR threshold)
    e.g.: MS:1001448 (pep:FDR threshold)
    e.g.: MS:1001454 (quality estimation with implicite decoy sequences)
    e.g.: MS:1001494 (no threshold)
    e.g.: MS:1001574 (report only spectra assigned to identified proteins)
  MAY supply a *child* term of MS:1001085 (protein result details) one or more times
    e.g.: MS:1001088 (protein description)
    e.g.: MS:1001090 (taxonomy nomenclature)
    e.g.: MS:1001093 (sequence coverage)
    e.g.: MS:1001097 (distinct peptide sequences)
    e.g.: MS:1001098 (confident distinct peptide sequences)
    e.g.: MS:1001099 (confident peptide qualification)
    e.g.: MS:1001100 (confident peptide)
    e.g.: MS:1001125 (manual validation)
    e.g.: MS:1001157 (Sequest:sp)
    e.g.: MS:1001158 (Sequest:Uniq)
    et al.
```

6.54 Element <ProteinDetectionList>

Definition: The protein list resulting from a protein detection process.

Type: ProteinDetectionListType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
ProteinAmbiguityGroup	0	unbounded	A set of logically related results from a protein detection, for example to represent conflicting assignments of peptides to proteins.
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	0	unbounded	A single user-defined parameter.

Example Context:

```
<ProteinDetectionList id="PDL_1">
  <ProteinAmbiguityGroup id="PAG_hit_1">
    <ProteinDetectionHypothesis id="PDH_psu|NC_LIV_020800_0"
      dBSequence_ref="DBSeq_1_psu|NC_LIV_020800" passThreshold="true">
      <PeptideHypothesis

peptideEvidence_ref="LCYIALDFDEEMKAAEDSSDIEK_0000000000000100000000000_1_psu|NC_LIV_020800_217_239">
        <SpectrumIdentificationItemRef spectrumIdentificationItem_ref="SII_308_1"/>
  ...
</ProteinDetectionList>
```

cvParam Mapping Rules:

```
Path /MzIdentML/DataCollection/AnalysisData/ProteinDetectionList
MAY supply a *child* term of MS:1001184 (search statistics) one or more times
  e.g.: MS:1001035 (date / time search performed)
  e.g.: MS:1001036 (search time taken)
  e.g.: MS:1001177 (number of molecular hypothesis considered)
```

6.55 Element <ProteinDetectionProtocol>

Definition: The parameters and settings of a ProteinDetection process.

Type: ProteinDetectionProtocolType

Attributes:

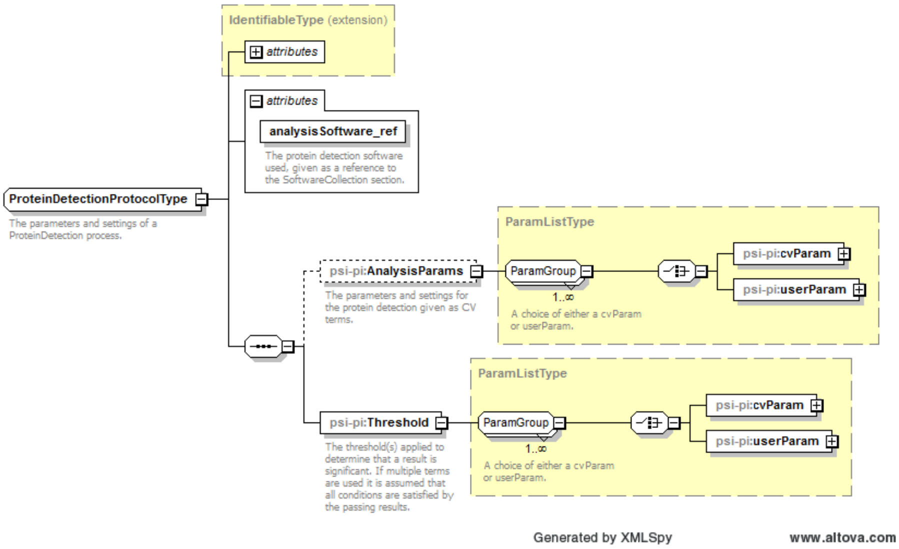
Attribute Name	Data Type	Use	Definition

analysisSoftware_ref	xsd:string	required	The protein detection software used, given as a reference to the SoftwareCollection section.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
AnalysisParams	0	1	The parameters and settings for the protein detection given as CV terms.
Threshold	1	1	The threshold(s) applied to determine that a result is significant. If multiple terms are used it is assumed that all conditions are satisfied by the passing results.

Graphical Context:


IdentifiableType (extension)
attributes
attributes
analysisSoftware_ref
The protein detection software used, given as a reference to the SoftwareCollection section.
ProteinDetectionProtocolType
The parameters and settings of a ProteinDetection process.
ParamListType
psi-pi:AnalysisParams
The parameters and settings for the protein detection given as CV terms.
ParamGroup
1..∞
A choice of either a cvParam or userParam.
psi-pi:cvParam
psi-pi:userParam
ParamListType
psi-pi:Threshold
The threshold(s) applied to determine that a result is significant. If multiple terms are used it is assumed that all conditions are satisfied by the passing results.
ParamGroup
1..∞
A choice of either a cvParam or userParam.
psi-pi:cvParam
psi-pi:userParam
Generated by XMLSpy
www.altova.com

Example Context:

```
<ProteinDetectionProtocol id="PDP_MascotParser_1" analysisSoftware_ref="AS_mascot_parser">
  <AnalysisParams>
    <cvParam accession="MS:1001316" name="mascot:SigThreshold" cvRef="PSI-MS" value="0.05"/>
    <cvParam accession="MS:1001317" name="mascot:MaxProteinHits" cvRef="PSI-MS" value="Auto"/>
    <cvParam accession="MS:1001318" name="mascot:ProteinScoringMethod" cvRef="PSI-MS"
      value="MudPIT"/>
    <cvParam accession="MS:1001319" name="mascot:MinMSMSThreshold" cvRef="PSI-MS" value="0"/>
  ...
</ProteinDetectionProtocol>
```

6.56 Element <Provider>

Definition: The Provider of the mzIdentML record in terms of the contact and software.

Type: ProviderType

Attributes:

Attribute Name	Data Type	Use	Definition
analysisSoftware_ref	xsd:string	optional	The Software that produced the document instance.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related

			documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
ContactRole	0	1	The Contact that provided the document instance.

Example Context:

```
<Provider id="PROVIDER">
  <ContactRole contact_ref="PERSON_DOC_OWNER">
    <Role>
      <cvParam accession="MS:1001271" name="researcher" cvRef="PSI-MS"/>
    </Role>
  </ContactRole>
</Provider>
```

6.57 Element <Residue>

Definition: The specification of a single residue within the mass table.

Type: ResidueType

Attributes:

Attribute Name	Data Type	Use	Definition
code	chars	required	The single letter code for the residue.
mass	xsd:float	required	The residue mass in Daltons (not including any fixed modifications).

Subelements: none

Example Context:

```
<Residue code="C" mass="103.009185"/>
```

6.58 Element <Role>

Definition: The roles (lab equipment sales, contractor, etc.) the Contact fills.

Type: RoleType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	1	A single entry from an ontology or a controlled vocabulary.

Example Context:

```
<Role>
  <cvParam accession="MS:1001267" name="software vendor" cvRef="PSI-MS"/>
</Role>
```

Example cvParams:

```
<cvParam accession="MS:1001267" name="software vendor" cvRef="PSI-MS"/>
<cvParam accession="MS:1001271" name="researcher" cvRef="PSI-MS"/>
```

6.59 Element <Sample>

Definition: A description of the sample analysed by mass spectrometry using CVParams or UserParams. If a composite sample has been analysed, a parent sample should be defined, which references subsamples. This represents any kind of substance used in an experimental workflow, such as whole organisms, cells, DNA, solutions, compounds and experimental substances (gels, arrays etc.).

Type: SampleType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within

			the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
ContactRole	0	unbounded	The Contact that provided the document instance.
SubSample	0	unbounded	References to the individual component samples within a mixed parent sample.
cvParam	0	unbounded	A choice of either a cvParam or userParam.
userParam	0	unbounded	A choice of either a cvParam or userParam.

Example Context:

6.60 Element <SearchDatabase>

Definition: A database for searching mass spectra. Examples include a set of amino acid sequence entries, or annotated spectra libraries.

Type: SearchDatabaseType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
location	xsd:anyURI	required	The location of the data file.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
numDatabaseSequences	xsd:long	optional	The total number of sequences in the database.
numResidues	xsd:long	optional	The number of residues in the database.
releaseDate	xsd:dateTime	optional	The date and time the database was released to the public; omit this attribute when the date and time are unknown or not applicable (e.g. custom databases).
version	xsd:string	optional	The version of the database.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
ExternalFormatDocumentation	0	1	A URI to access documentation and tools to interpret the external format of the ExternalData instance. For example, XML Schema or static libraries (APIs) to access binary formats.
FileFormat	0	1	The format of the ExternalData file, for example "tiff" for image files.
DatabaseName	1	1	The database name may be given as a cvParam if it maps exactly to one of the release databases listed in the CV, otherwise a userParam

			should be used.
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.

Example Context:

```
<SearchDatabase numDatabaseSequences="22348"
location="D:/Software/Databases/Neospora_3rndTryp/Neo_rndTryp_3times.fasta" id="SearchDB_1">
    <FileFormat>
        <cvParam accession="MS:1001348" cvRef="PSI-MS" name="FASTA format"/>
    </FileFormat>
    <DatabaseName>
        <userParam name="D:/Software/Databases/Neospora_3rndTryp/Neo_rndTryp_3times.fasta"/>
    </DatabaseName>
  ...
</SearchDatabase>
```

cvParam Mapping Rules:

```
Path /MzIdentML/DataCollection/Inputs/SearchDatabase
MAY supply a *child* term of MS:1000561 (data file checksum type) one or more times
  e.g.: MS:1000568 (MD5)
  e.g.: MS:1000569 (SHA-1)
MAY supply a *child* term of MS:1001011 (search database details) one or more times
  e.g.: MS:1001014 (database local file path)
  e.g.: MS:1001015 (database original uri)
  e.g.: MS:1001016 (database version)
  e.g.: MS:1001017 (database release date)
  e.g.: MS:1001020 (DB filter taxonomy)
  e.g.: MS:1001021 (DB filter on accession numbers)
  e.g.: MS:1001022 (DB MW filter)
  e.g.: MS:1001023 (DB PI filter)
  e.g.: MS:1001024 (translation frame)
  e.g.: MS:1001025 (translation table)
  et al.
```

6.61 Element <SearchDatabaseRef>

Definition: One of the search databases used.

Type: SearchDatabaseRefType

Attributes:

Attribute Name	Data Type	Use	Definition
searchDatabase_ref	xsd:string	optional	A reference to the database searched.

Subelements: none

Example Context: `<SearchDatabaseRef searchDatabase_ref="SDB_NeoProt_tripledecoy"/>`

6.62 Element <SearchModification>

Definition: Specification of a search modification as parameter for a spectra search. Contains the name of the modification, the mass, the specificity and whether it is a static modification.

Type: SearchModificationType

Attributes:

Attribute Name	Data Type	Use	Definition
fixedMod	xsd:boolean	required	True, if the modification is static (i.e. occurs always).
massDelta	xsd:float	required	The mass delta of the searched modification in Daltons.
residues	listOfCharsOrAny	required	The residue(s) searched with the specified modification. For N or C terminal modifications that can occur on any residue, the . character should be used to specify any, otherwise the list of amino acids should be provided.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SpecificityRules	0	1	The specificity rules of the searched modification including for example the probability of a modification's presence or peptide or protein termini. Standard fixed or variable status should be

			provided by the attribute fixedMod.
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.

Example Context:

```
<SearchModification residues="C" massDelta="57.021465" fixedMod="true">
    <cvParam accession="UNIMOD:4" cvRef="UNIMOD" name="Carbamidomethyl"/>
</SearchModification>
```

cvParam Mapping Rules:

```
Path
/MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/ModificationParams/SearchModifi
cation
MUST supply a *child* term of UNIMOD:0 (UNIMOD root) one or more times
MUST supply a *child* term of MS:1001471 (peptide modification details) one or more times
  e.g.: MS:1001460 (unknown modification)
  e.g.: MS:1001524 (fragment neutral loss)
  e.g.: MS:1001525 (precursor neutral loss)
MUST supply a *child* term of MOD:00000 (protein modification) one or more times
```

Example cvParams:

```
<cvParam accession="UNIMOD:4" name="Carbamidomethyl" cvRef="UNIMOD"/>
<cvParam accession="UNIMOD:35" name="Oxidation" cvRef="UNIMOD"/>
```

6.63 Element <SearchType>

Definition: The type of search performed e.g. PMF, Tag searches, MS-MS

Type: ParamType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	1	A single entry from an ontology or a controlled vocabulary.
userParam	1	1	A single user-defined parameter.

Example Context:

```
<SearchType>
  <cvParam accession="MS:1001083" name="ms-ms search" cvRef="PSI-MS" value=""/>
</SearchType>
```

cvParam Mapping Rules:

```
Path /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/SearchType
MUST supply a *child* term of MS:1001080 (search type) one or more times
  e.g.: MS:1001010 (de novo search)
  e.g.: MS:1001031 (spectral library search)
  e.g.: MS:1001081 (pmf search)
  e.g.: MS:1001082 (tag search)
  e.g.: MS:1001083 (ms-ms search)
  e.g.: MS:1001584 (combined pmf + ms-ms search)
```

Example cvParams:

```
<cvParam accession="MS:1001083" name="ms-ms search" cvRef="PSI-MS" value=""/>
```

6.64 Element <Seq>

Definition: The actual sequence of amino acids or nucleic acid.

Type: sequence

Attributes: none

Subelements: none

Example Context:

```
<Seq>MPSRSNSRGASADPASDALSDADAASSAVPGSASERSFPFVHPSRDQLTADKPAKRDADQEAFAMTEDTLPPVPPAPPPGEEGVPSSRFTSSEAFH
DPPASPACASPPRRRCAAASPELEALGAFFARYACCLERVAVVDGAAECPGSLFGCALLPHVEASPAFAVSPAAWTSRWEADPFAWSGQGETRHGGALASR
...
RVERRRPDHRRRAGDCGEKGPKRGARRGRKHGGARAPSRAGPETEPAAASPAASARQRLVQAMALPACAFLDLQAQQPSSFSVSPDGTPDPVYPQDLVSLD
ALRQGFPCGAPTGKSLGRIQDWSSTGATFWSRRVRAALDAFVLLPSWYGGIENRLLLEEAAVLLANTATCALLEAYAVHCLKRQAAAPIPRMYAAGHAAVS
DASLRSVQIAENQTVDDRLPTASTCFFMIKLPKYSSKEVLRKKLKLAIMSCVDIDLDALHHDLDFAQFE</Seq>
```

Example cvParams:

```
<cvParam accession="MS:1001088" name="protein description" cvRef="PSI-MS"
```

6.65 Element <SequenceCollection>

Definition: The collection of sequences (DBSequence or Peptide) identified and their relationship between each other (PeptideEvidence) to be referenced elsewhere in the results.

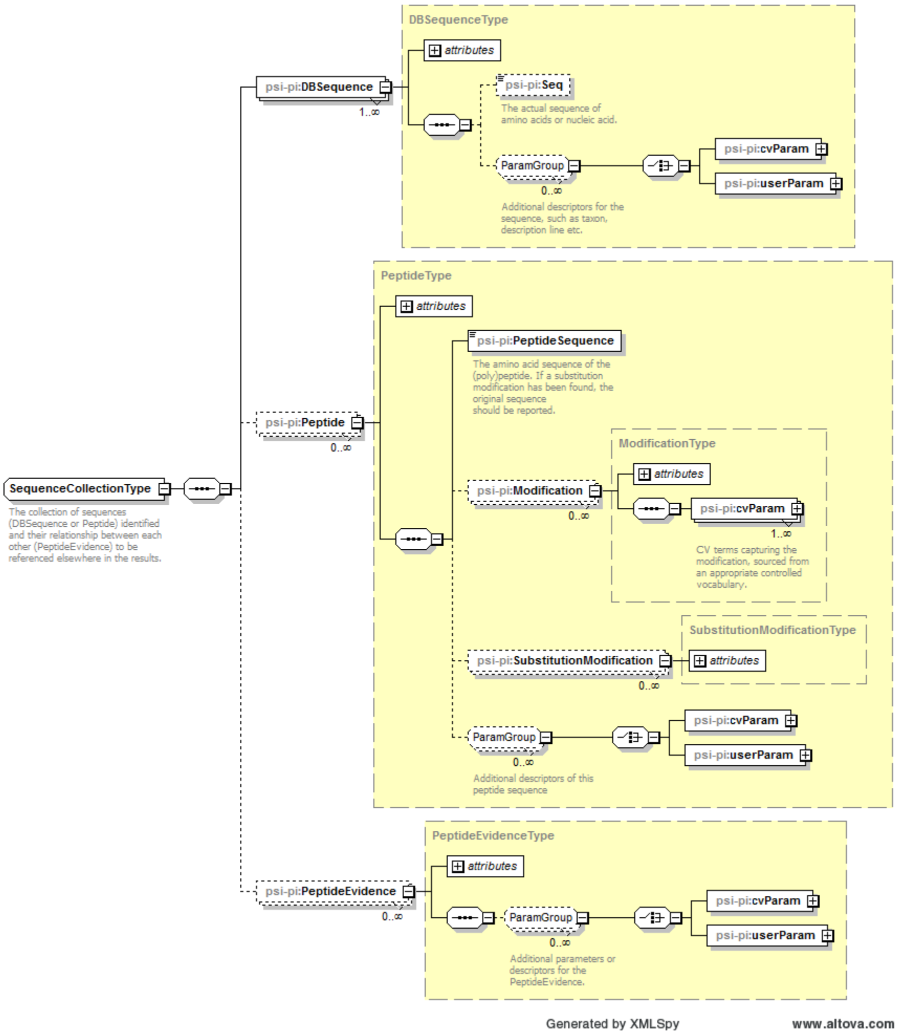
Type: SequenceCollectionType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
DBSequence	1	unbounded	A database sequence from the specified SearchDatabase (nucleic acid or amino acid). If the sequence is nucleic acid, the source nucleic acid sequence should be given in the seq attribute rather than a translated sequence.
Peptide	0	unbounded	One (poly)peptide (a sequence with modifications). The combination of Peptide sequence and modifications MUST be unique in the file.
PeptideEvidence	0	unbounded	PeptideEvidence links a specific Peptide element to a specific position in a DBSequence. There MUST only be one PeptideEvidence item per Peptide-to-DBSequence-position.

Graphical Context:


DBSequenceType
attributes
psi-pi:DBSequence
1..∞
psi-pi:Seq
The actual sequence of amino acids or nucleic acid.
ParamGroup
0..∞
Additional descriptors for the sequence, such as taxon, description line etc.
psi-pi:cvParam
psi-pi:userParam
PeptideType
attributes
psi-pi:PeptideSequence
The amino acid sequence of the (poly)peptide. If a substitution modification has been found, the original sequence should be reported.
psi-pi:Peptide
0..∞
SequenceCollectionType
The collection of sequences (DBSequence or Peptide) identified and their relationship between each other (PeptideEvidence) to be referenced elsewhere in the results.
ModificationType
attributes
psi-pi:Modification
0..∞
psi-pi:cvParam
1..∞
CV terms capturing the modification, sourced from an appropriate controlled vocabulary.
SubstitutionModificationType
psi-pi:SubstitutionModification
attributes
0..∞
ParamGroup
0..∞
Additional descriptors of this peptide sequence
psi-pi:cvParam
psi-pi:userParam
PeptideEvidenceType
attributes
psi-pi:PeptideEvidence
0..∞
ParamGroup
0..∞
Additional parameters or descriptors for the PeptideEvidence.
psi-pi:cvParam
psi-pi:userParam
Generated by XMLSpy
www.altova.com

Example Context:

```
<SequenceCollection xmlns="http://psidev.info/psi/pi/mzIdentML/1.1">
    <DBSequence accession="Rnd3psu|NC_LIV_083320" searchDatabase_ref="SearchDB_1" length="661"
id="dbseq_Rnd3psu|NC_LIV_083320">
        <cvParam accession="MS:1001088" cvRef="PSI-MS" value="Rnd3psu|NC_LIV_083320
Rnd3psu|NC_LIV_083320 Decoy sequence, was | organism=Neospora_caninum | product=zinc finger (CCCH
type) protein, putative | location=Neo_chrVIIa:3989308-3992771(+) | length=661" name="protein
description"/>
    </DBSequence>
    <DBSequence accession="Rnd1psu|NC_LIV_123020" searchDatabase_ref="SearchDB_1" length="2986"
id="dbseq_Rnd1psu|NC_LIV_123020">
        <cvParam accession="MS:1001088" cvRef="PSI-MS" value="Rnd1psu|NC_LIV_123020
Rnd1psu|NC_LIV_123020 Decoy sequence, was | organism=Neospora_caninum | product=hypothetical protein |
location=Neo_chrX:3202583-3213218(-) | length=2986" name="protein description"/>
    </DBSequence>
  ...
</SequenceCollection>
```

6.66 Element <SiteRegexp>

Definition: Regular expression for specifying the enzyme cleavage site.

Type: xsd:string

Attributes: none

Subelements: none

Example Context: `<SiteRegexp><![CDATA[(?<=[KR])(?!P)]]></SiteRegexp>`

6.67 Element <SoftwareName>

Definition: The name of the analysis software package, sourced from a CV if available.

Type: ParamType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	1	A choice of either a cvParam or userParam.
userParam	1	1	A choice of either a cvParam or userParam.

Example Context:

```
<SoftwareName>
  <cvParam accession="MS:1001478" name="Mascot Parser" cvRef="PSI-MS"/>
</SoftwareName>
```

cvParam Mapping Rules:

```
Path /MzIdentML/AnalysisSoftwareList/AnalysisSoftware/SoftwareName
MUST supply a *child* term of MS:1001456 (analysis software) one or more times
  e.g.: MS:1000532 (Xcalibur)
  e.g.: MS:1000533 (Bioworks)
  e.g.: MS:1000534 (MassLynx)
  e.g.: MS:1000535 (FlexAnalysis)
  e.g.: MS:1000536 (Data Explorer)
  e.g.: MS:1000537 (4700 Explorer)
  e.g.: MS:1000539 (Voyager Biospectrometry Workstation System)
  e.g.: MS:1000551 (Analyst)
  e.g.: MS:1000600 (Proteios)
  e.g.: MS:1000601 (ProteinLynx Global Server)
  et al.
```

Example cvParams:

```
<cvParam accession="MS:1001207" name="Mascot" cvRef="PSI-MS"/>
<cvParam accession="MS:1001478" name="Mascot Parser" cvRef="PSI-MS"/>
<cvParam accession="MS:1001475" cvRef="PSI-MS" name="OMSSA"/>
```

6.68 Element <SourceFile>

Definition: A file from which this mzIdentML instance was created.

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
location	xsd:anyURI	required	The location of the data file.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
ExternalFormatDocumentation	0	1	A URI to access documentation and tools to interpret the external format of the ExternalData instance. For example, XML Schema or static libraries (APIs) to access binary formats.
FileFormat	0	1	The format of the ExternalData file, for example "tiff" for image files.
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.

userParam	0	unbounded	A single user-defined parameter.

Example Context:

```
<SourceFile location="file:///D:/TestSpace/NeoTestMarch2011/55merge_mascot.dat" id="SF_1">
  <FileFormat>
    <cvParam accession="MS:1001199" name="Mascot DAT file" cvRef="PSI-MS"/>
  </FileFormat>
</SourceFile>
```

cvParam Mapping Rules:

```
Path /MzIdentML/DataCollection/Inputs/SourceFile
MAY supply a *child* term of MS:1000561 (data file checksum type) one or more times
  e.g.: MS:1000568 (MD5)
  e.g.: MS:1000569 (SHA-1)
```

6.69 Element <SpecificityRules>

Definition: The specificity rules of the searched modification including for example the probability of a modification's presence or peptide or protein termini. Standard fixed or variable status should be provided by the attribute fixedMod.

Type: SpecificityRulesType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.

Example Context:

cvParam Mapping Rules:

```
Path
/MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/ModificationParams/SearchModifi
cation/SpecificityRules
MUST supply a *child* term of MS:1001056 (modification specificity rule) one or more times
  e.g.: MS:1001189 (modification specificity N-term)
  e.g.: MS:1001190 (modification specificity C-term)
```

6.70 Element <SpectraData>

Definition: A data set containing spectra data (consisting of one or more spectra).

Type: SpectraDataType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
location	xsd:anyURI	required	The location of the data file.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
ExternalFormatDocumentation	0	1	A URI to access documentation and tools to interpret the external format of the ExternalData instance. For example, XML Schema or static libraries (APIs) to access binary formats.
FileFormat	0	1	The format of the ExternalData file, for example "tiff" for image files.
SpectrumIDFormat	1	1	The format of the spectrum identifier within the source file

Example

```
<SpectraData location="D:/TestSpace/NeoTestMarch2011/55merge.mgf" id="SID_1">
    <FileFormat>
```

Context:

```
        <cvParam accession="MS:1001062" cvRef="PSI-MS" name="Mascot MGF file"/>
    </FileFormat>
    <SpectrumIDFormat>
        <cvParam accession="MS:1000774" cvRef="PSI-MS" name="multiple peak list nativeID format"/>
    </SpectrumIDFormat>
  ...
</SpectraData>
```

6.71 Element <SpectrumIDFormat>

Definition: The format of the spectrum identifier within the source file

Type: SpectrumIDFormatType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	1	A single entry from an ontology or a controlled vocabulary.

Example Context:

```
<SpectrumIDFormat>
    <cvParam accession="MS:1000774" cvRef="PSI-MS" name="multiple peak list nativeID format"/>
</SpectrumIDFormat>
```

cvParam Mapping Rules:

```
Path /MzIdentML/DataCollection/Inputs/SpectraData/SpectrumIDFormat
MUST supply a *child* term of MS:1000767 (native spectrum identifier format) only once
  e.g.: MS:1000768 (Thermo nativeID format)
  e.g.: MS:1000769 (Waters nativeID format)
  e.g.: MS:1000770 (WIFF nativeID format)
  e.g.: MS:1000771 (Bruker/Agilent YEP nativeID format)
  e.g.: MS:1000772 (Bruker BAF nativeID format)
  e.g.: MS:1000773 (Bruker FID nativeID format)
  e.g.: MS:1000774 (multiple peak list nativeID format)
  e.g.: MS:1000775 (single peak list nativeID format)
  e.g.: MS:1000776 (scan number only nativeID format)
  e.g.: MS:1000777 (spectrum identifier nativeID format)
  et al.
MUST supply a *child* term of MS:1001529 (spectra data details) only once
  e.g.: MS:1001530 (mzML unique identifier)
  e.g.: MS:1001531 (spectrum from ProteinScape database nativeID format)
  e.g.: MS:1001532 (spectrum from database nativeID format)
```

Example cvParams:

```
<cvParam accession="MS:1000774" name="multiple peak list nativeID format" cvRef="PSI-MS"/>
```

6.72 Element <SpectrumIdentification>

Definition: An Analysis which tries to identify peptides in input spectra, referencing the database searched, the input spectra, the output results and the protocol that is run.

Type: SpectrumIdentificationType

Attributes:

Attribute Name	Data Type	Use	Definition
activityDate	xsd:dateTime	optional	When the protocol was applied.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
spectrumIdentificationList_ref	xsd:string	required	A reference to the SpectrumIdentificationList produced by this analysis in the DataCollection section.
spectrumIdentificationProtocol_ref	xsd:string	required	A reference to the search protocol used for this SpectrumIdentification.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition

InputSpectra	1	unbounded	One of the spectra data sets used.
SearchDatabaseRef	1	unbounded	One of the search databases used.

Example Context:

```
<SpectrumIdentification id="SI" spectrumIdentificationProtocol_ref="SIP"
  spectrumIdentificationList_ref="SIL_1" activityDate="2011-03-24T11:37:37">
  <InputSpectra spectraData_ref="SD_1"/>
  <SearchDatabaseRef searchDatabase_ref="SDB_NeoProt_tripledecoy"/>
</SpectrumIdentification>
```

6.73 Element <SpectrumIdentificationItem>

Definition: An identification of a single (poly)peptide, resulting from querying an input spectra, along with the set of confidence values for that identification. PeptideEvidence elements should be given for all mappings of the corresponding Peptide sequence within protein sequences.

Type: SpectrumIdentificationItemType

Attributes:

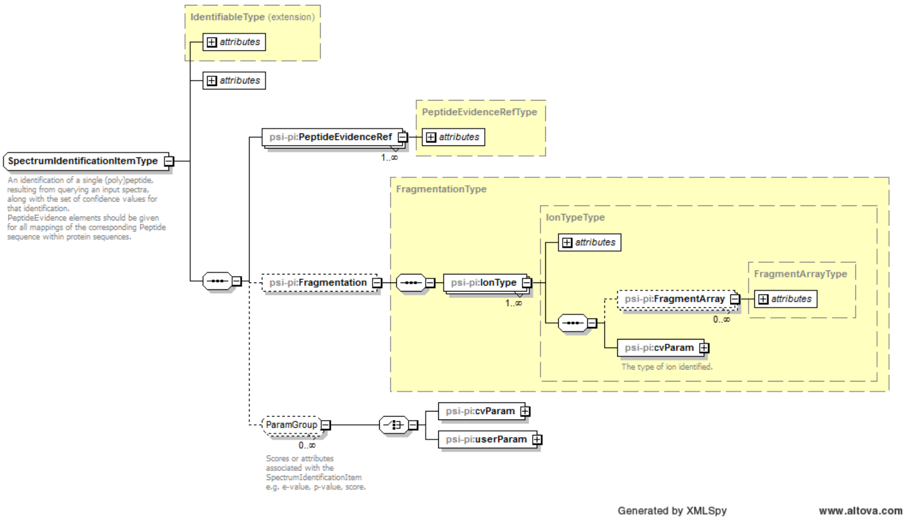
Attribute Name	Data Type	Use	Definition
calculatedMassToCharge	xsd:double	optional	The theoretical mass-to-charge value calculated for the peptide in Daltons / charge.
calculatedPI	xsd:float	optional	The calculated isoelectric point of the (poly)peptide, with relevant modifications included. Do not supply this value if the PI cannot be calcuated properly.
chargeState	xsd:int	required	The charge state of the identified peptide.
experimentalMassToCharge	xsd:double	required	The mass-to-charge value measured in the experiment in Daltons / charge.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
massTable_ref	xsd:string	optional	A reference should be given to the MassTable used to calculate the sequenceMass only if more than one MassTable has been given.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
passThreshold	xsd:boolean	required	Set to true if the producers of the file has deemed that the identification has passed a given threshold or been validated as correct. If no such threshold has been set, value of true should be given for all results.
peptide_ref	xsd:string	optional	A reference to the identified (poly)peptide sequence in the Peptide element.
rank	xsd:int	required	For an MS/MS result set, this is the rank of the identification quality as scored by the search engine. 1 is the top rank. If multiple identifications have the same top score, they should all be assigned rank =1. For PMF data, the rank attribute may be meaningless and values of rank = 0 should be given.
sample_ref	xsd:string	optional	A reference should be provided to link the SpectrumIdentificationItem to a Sample if

			more than one sample has been described in the AnalysisSampleCollection.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
PeptideEvidenceRef	1	unbounded	Reference to the PeptideEvidence element identified. If a specific sequence can be assigned to multiple proteins and or positions in a protein all possible PeptideEvidence elements should be referenced here.
Fragmentation	0	1	The product ions identified in this result.
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	0	unbounded	A single user-defined parameter.

Graphical Context:


IdentifiableType (extension)
attributes
attributes
SpectrumIdentificationItemType
An identification of a single (poly)peptide, resulting from querying an input spectra, along with the set of confidence values for that identification.
PeptideEvidence elements should be given for all mappings of the corresponding Peptide sequence within protein sequences.
psi-pi:PeptideEvidenceRef
1..∞
PeptideEvidenceRefType
attributes
FragmentationType
IonTypeType
attributes
psi-pi:Fragmentation
psi-pi:IonType
1..∞
FragmentArrayType
psi-pi:FragmentArray
attributes
0..∞
psi-pi:cvParam
The type of ion identified.
ParamGroup
0..∞
Scores or attributes associated with the SpectrumIdentificationItem e.g. e-value, p-value, score.
psi-pi:cvParam
psi-pi:userParam
Generated by XMLSpy
www.altova.com

Example Context:

```
<SpectrumIdentificationItem id="SII_69_10" calculatedMassToCharge="515.329999"
  chargeState="1" experimentalMassToCharge="514.242" peptide_ref="GIGLR_0000000" rank="10"
  passThreshold="false">
  <PeptideEvidenceRef peptideEvidence_ref="GIGLR_0000000_1_psu|NC_LIV_020600_1026_1030"/>
  <PeptideEvidenceRef peptideEvidence_ref="GIGLR_0000000_1_Rnd1psu|NC_LIV_145070_441_445"/>
  <PeptideEvidenceRef peptideEvidence_ref="GIGLR_0000000_1_Rnd1psu|NC_LIV_120130_148_152"/>
  <PeptideEvidenceRef peptideEvidence_ref="GIGLR_0000000_1_Rnd3psu|NC_LIV_072540_650_654"/>
  ...
</SpectrumIdentificationItem>
```

cvParam Mapping Rules:

```
Path
/MzIdentML/DataCollection/AnalysisData/SpectrumIdentificationList/SpectrumIdentificationResult/Spectrum
IdentificationItem
MAY supply a *child* term of MS:1001405 (spectrum identification result details) one or more times
  e.g.: MS:1000796 (spectrum title)
  e.g.: MS:1000797 (peak list scans)
  e.g.: MS:1000798 (peak list raw scans)
  e.g.: MS:1000903 (product ion series ordinal)
  e.g.: MS:1000904 (product ion m/z delta)
  e.g.: MS:1000926 (product interpretation rank)
  e.g.: MS:1001030 (number of peptide seqs compared to each spectrum)
  e.g.: MS:1001035 (date / time search performed)
  e.g.: MS:1001036 (search time taken)
  e.g.: MS:1001088 (protein description)
  et al.
```

Example cvParams:

```
<cvParam accession="MS:1001171" name="mascot:score" cvRef="PSI-MS" value="13.49"/>
<cvParam accession="MS:1001172" name="mascot:expectation value" cvRef="PSI-MS"
<cvParam accession="MS:1001363" name="peptide unique to one protein" cvRef="PSI-MS"/>
<cvParam accession="MS:1001371" name="mascot:identity threshold" cvRef="PSI-MS" value="43"/>
<cvParam accession="MS:1001370" name="mascot:homology threshold" cvRef="PSI-MS" value="26"/>
<cvParam accession="MS:1001030" name="number of peptide seqs compared to each spectrum"
<cvParam accession="MS:1000796" name="spectrum title" cvRef="PSI-MS"
```

```
<cvParam accession="MS:1001328" cvRef="PSI-MS" value="0.866331351956052" name="OMSSA:evalue"/>
<cvParam accession="MS:1001329" cvRef="PSI-MS" value="2.0805267818349E-4" name="OMSSA:pvalue"/>
```

6.74 Element <SpectrumIdentificationItemRef>

Definition: Reference(s) to the SpectrumIdentificationItem element(s) that support the given PeptideEvidence element. Using these references it is possible to indicate which spectra were actually accepted as evidence for this peptide identification in the given protein.

Type: SpectrumIdentificationItemRefType

Attributes:

Attribute Name	Data Type	Use	Definition
spectrumIdentificationItem_ref	xsd:string	required	A reference to the SpectrumIdentificationItem element(s).

Subelements: none

Example Context:

```
<SpectrumIdentificationItemRef spectrumIdentificationItem_ref="SII_308_1"/>
```

6.75 Element <SpectrumIdentificationList>

Definition: Represents the set of all search results from SpectrumIdentification.

Type: SpectrumIdentificationListType

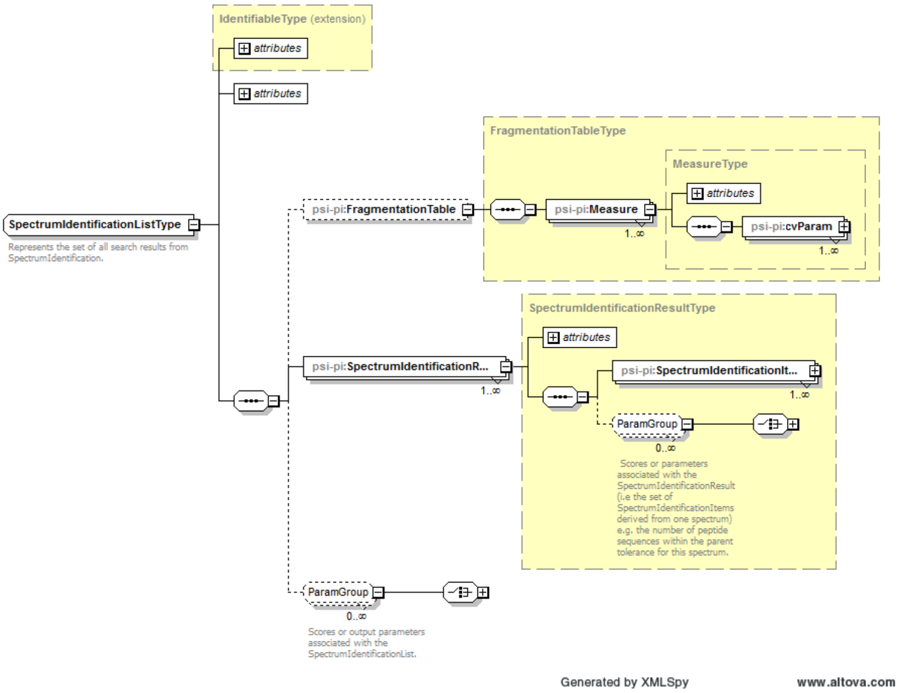
Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
numSequencesSearched	xsd:long	optional	The number of database sequences searched against. This value should be provided unless a de novo search has been performed.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
FragmentationTable	0	1	Contains the types of measures that will be reported in generic arrays for each SpectrumIdentificationItem e.g. product ion m/z, product ion intensity, product ion m/z error
SpectrumIdentificationResult	1	unbounded	All identifications made from searching one spectrum. For PMF data, all peptide identifications will be listed underneath as SpectrumIdentificationItems. For MS/MS data, there will be ranked SpectrumIdentificationItems corresponding to possible different peptide IDs.
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	0	unbounded	A single user-defined parameter.

Graphical Context:


IdentifiableType (extension)
attributes
attributes
SpectrumIdentificationListType
Represents the set of all search results from SpectrumIdentification.
FragmentationTableType
MeasureType
attributes
psi-pi:FragmentationTable
psi-pi:Measure
1..∞
psi-pi:cvParam
1..∞
SpectrumIdentificationResultType
attributes
psi-pi:SpectrumIdentificationR...
1..∞
psi-pi:SpectrumIdentificationIt...
1..∞
ParamGroup
0..∞
Scores or parameters associated with the SpectrumIdentificationResult (i.e the set of SpectrumIdentificationItems derived from one spectrum) e.g. the number of peptide sequences within the parent tolerance for this spectrum.
ParamGroup
0..∞
Scores or output parameters associated with the SpectrumIdentificationList.
Generated by XMLSpy
www.altova.com

Example Context:

```
<SpectrumIdentificationList id="SII_LIST_1" xmlns="http://psidev.info/psi/pi/mzIdentML/1.1">
    <FragmentationTable>
        <Measure id="Measure_MZ">
            <cvParam accession="MS:1001225" cvRef="PSI-MS" unitCvRef="PSI-MS" unitName="m/z"
unitAccession="MS:1000040" name="product ion m/z"/>
        </Measure>
        <Measure id="Measure_Int">
            <cvParam accession="MS:1001226" cvRef="PSI-MS" name="product ion intensity"/>
  ...
</SpectrumIdentificationList>
```

cvParam Mapping Rules:

```
Path /MzIdentML/DataCollection/AnalysisData/SpectrumIdentificationList
MAY supply a *child* term of MS:1001184 (search statistics) one or more times
  e.g.: MS:1001035 (date / time search performed)
  e.g.: MS:1001036 (search time taken)
  e.g.: MS:1001177 (number of molecular hypothesis considered)
```

6.76 Element <SpectrumIdentificationProtocol>

Definition: The parameters and settings of a SpectrumIdentification analysis.

Type: SpectrumIdentificationProtocolType

Attributes:

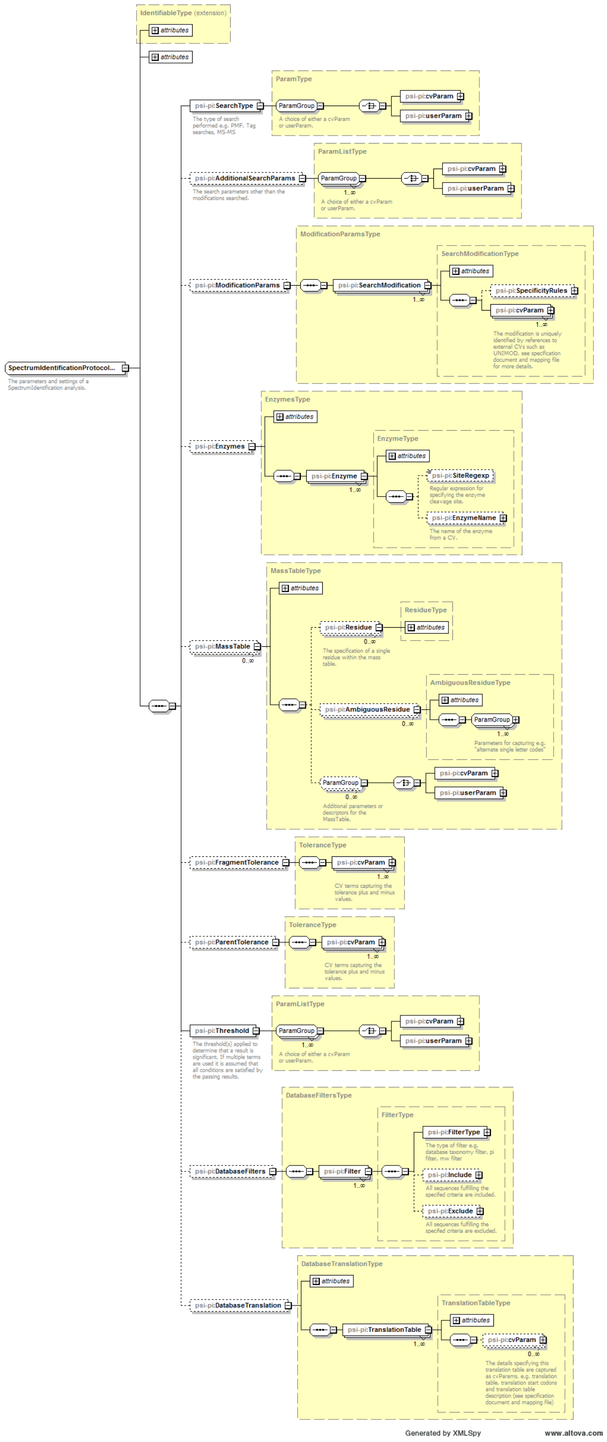
Attribute Name	Data Type	Use	Definition
analysisSoftware_ref	xsd:string	required	The search algorithm used, given as a reference to the SoftwareCollection section.
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SearchType	1	1	The type of search performed e.g. PMF,

			Tag searches, MS-MS
AdditionalSearchParams	0	1	The search parameters other than the modifications searched.
ModificationParams	0	1	The specification of static/variable modifications (e.g. Oxidation of Methionine) that are to be considered in the spectra search.
Enzymes	0	1	The list of enzymes used in experiment
MassTable	0	unbounded	The masses of residues used in the search.
FragmentTolerance	0	1	The tolerance of the search given as a plus and minus value with units.
ParentTolerance	0	1	The tolerance of the search given as a plus and minus value with units.
Threshold	1	1	The threshold(s) applied to determine that a result is significant. If multiple terms are used it is assumed that all conditions are satisfied by the passing results.
DatabaseFilters	0	1	The specification of filters applied to the database searched.
DatabaseTranslation	0	1	A specification of how a nucleic acid sequence database was translated for searching.

Graphical Context:


IdentifiableType (extension)
attributes
attributes
SpectrumIdentificationProtocol...
The parameters and settings of a SpectrumIdentification analysis.
ParamType
psi-pi:SearchType
The type of search performed e.g. PMF, Tag searches, MS-MS
ParamGroup
A choice of either a cvParam or userParam.
psi-pi:cvParam
psi-pi:userParam
ParamListType
psi-pi:AdditionalSearchParams
The search parameters other than the modifications searched.
ParamGroup
1..∞
A choice of either a cvParam or userParam.
psi-pi:cvParam
psi-pi:userParam
ModificationParamsType
psi-pi:ModificationParams
psi-pi:SearchModification
1..∞
SearchModificationType
attributes
psi-pi:SpecificityRules
psi-pi:cvParam
1..∞
The modification is uniquely identified by references to external CVs such as UNIMOD, see specification document and mapping file for more details.
EnzymesType
attributes
psi-pi:Enzymes
psi-pi:Enzyme
1..∞
EnzymeType
attributes
psi-pi:SiteRegexp
Regular expression for specifying the enzyme cleavage site.
psi-pi:EnzymeName
The name of the enzyme from a CV.
MassTableType
attributes
psi-pi:MassTable
0..∞
psi-pi:Residue
0..∞
The specification of a single residue within the mass table.
ResidueType
attributes
psi-pi:AmbiguousResidue
0..∞
AmbiguousResidueType
attributes
ParamGroup
1..∞
Parameters for capturing e.g. "alternate single letter codes"
ParamGroup
0..∞
Additional parameters or descriptors for the MassTable.
psi-pi:cvParam
psi-pi:userParam
ToleranceType
psi-pi:FragmentTolerance
psi-pi:cvParam
1..∞
CV terms capturing the tolerance plus and minus values.
ToleranceType
psi-pi:ParentTolerance
psi-pi:cvParam
1..∞
CV terms capturing the tolerance plus and minus values.
ParamListType
psi-pi:Threshold
The threshold(s) applied to determine that a result is significant. If multiple terms are used it is assumed that all conditions are satisfied by the passing results.
ParamGroup
1..∞
A choice of either a cvParam or userParam.
psi-pi:cvParam
psi-pi:userParam
DatabaseFiltersType
psi-pi:DatabaseFilters
psi-pi:Filter
1..∞
FilterType
psi-pi:FilterType
The type of filter e.g. database taxonomy filter, pi filter, mw filter
psi-pi:Include
All sequences fulfilling the specifed criteria are included.
psi-pi:Exclude
All sequences fulfilling the specifed criteria are excluded.
DatabaseTranslationType
attributes
psi-pi:DatabaseTranslation
psi-pi:TranslationTable
1..∞
TranslationTableType
attributes
psi-pi:cvParam
0..∞
The details specifying this translation table are captured as cvParams, e.g. translation table, translation start codons and translation table description (see specification document and mapping file)
Generated by XMLSpy
www.altova.com

Example Context:

```
<SpectrumIdentificationProtocol analysisSoftware_ref="ID_software" id="SearchProtocol_1">
    <SearchType>
        <cvParam accession="MS:1001083" cvRef="PSI-MS" name="ms-ms search"/>
    </SearchType>
    <AdditionalSearchParams>
        <cvParam accession="MS:1001211" cvRef="PSI-MS" name="parent mass type mono"/>
        <cvParam accession="MS:1001256" cvRef="PSI-MS" name="fragment mass type mono"/>
  ...
</SpectrumIdentificationProtocol>
```

6.77 Element <SpectrumIdentificationResult>

Definition: All identifications made from searching one spectrum. For PMF data, all peptide identifications will be listed underneath as SpectrumIdentificationItems. For MS/MS data, there will be ranked SpectrumIdentificationItems corresponding to possible different peptide IDs.

Type: SpectrumIdentificationResultType

Attributes:

Attribute Name	Data Type	Use	Definition
id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.
spectraData_ref	xsd:string	required	A reference to a spectra data set (e.g. a spectra file).
spectrumID	xsd:string	required	The locally unique id for the spectrum in the spectra data set specified by SpectraData_ref. External guidelines are provided on the use of consistent identifiers for spectra in different external formats.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
SpectrumIdentificationItem	1	unbounded	An identification of a single (poly)peptide, resulting from querying an input spectra, along with the set of confidence values for that identification. PeptideEvidence elements should be given for all mappings of the corresponding Peptide sequence within protein sequences.
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	0	unbounded	A single user-defined parameter.

Example Context:

```
<SpectrumIdentificationResult spectraData_ref="SID_1" spectrumID="index=21" id="SIR_32">
    <SpectrumIdentificationItem passThreshold="false" rank="1"
peptide_ref="VIDENFGLVEGLMTTVHAATGTQK_1@12" calculatedMassToCharge="2546268.0"
experimentalMassToCharge="2547212.0" chargeState="3" id="SII_32_1">
        <PeptideEvidenceRef peptideEvidence_ref="PE32_2_53"/>
        <cvParam accession="MS:1001328" cvRef="PSI-MS" value="7.40729329987533E-8"
name="OMSSA:evalue"/>
        <cvParam accession="MS:1001329" cvRef="PSI-MS" value="3.18593260209692E-11"
name="OMSSA:pvalue"/>
    </SpectrumIdentificationItem>
    <SpectrumIdentificationItem passThreshold="false" rank="2"
peptide_ref="APCSGSAVTGVDSPGCDGVGDLNVTR" calculatedMassToCharge="2547129.0"
experimentalMassToCharge="2547212.0" chargeState="3" id="SII_32_2">
  ...
</SpectrumIdentificationResult>
```

cvParam Mapping Rules:

```
Path /MzIdentML/DataCollection/AnalysisData/SpectrumIdentificationList/SpectrumIdentificationResult
MAY supply a *child* term of MS:1001405 (spectrum identification result details) one or more times
  e.g.: MS:1000796 (spectrum title)
  e.g.: MS:1000797 (peak list scans)
  e.g.: MS:1000798 (peak list raw scans)
  e.g.: MS:1000903 (product ion series ordinal)
  e.g.: MS:1000904 (product ion m/z delta)
  e.g.: MS:1000926 (product interpretation rank)
  e.g.: MS:1001030 (number of peptide seqs compared to each spectrum)
  e.g.: MS:1001035 (date / time search performed)
```

```
e.g.: MS:1001036 (search time taken)
e.g.: MS:1001088 (protein description)
et al.
```

6.78 Element <SubSample>

Definition: References to the individual component samples within a mixed parent sample.

Type: SubSampleType

Attributes:

Attribute Name	Data Type	Use	Definition
sample_ref	xsd:string	required	A reference to the child sample.

Subelements: none

Example Context:

6.79 Element <SubstitutionModification>

Definition: A modification where one residue is substituted by another (amino acid change).

Type: SubstitutionModificationType

Attributes:

Attribute Name	Data Type	Use	Definition
avgMassDelta	xsd:double	optional	Atomic mass delta considering the natural distribution of isotopes in Daltons. This should only be reported if the original amino acid is known i.e. it is not "X"
location	xsd:int	optional	Location of the modification within the peptide - position in peptide sequence, counted from the N-terminus residue, starting at position 1. Specific modifications to the N-terminus should be given the location 0. Modification to the C-terminus should be given as peptide length + 1.
monoisotopicMassDelta	xsd:double	optional	Atomic mass delta when assuming only the most common isotope of elements in Daltons. This should only be reported if the original amino acid is known i.e. it is not "X"
originalResidue	xsd:string with restriction [ABCDEFGHIJKLMNOPQRSTUVWXYZ?\-]{1}	required	The original residue before replacement.
replacementResidue	xsd:string with restriction [ABCDEFGHIJKLMNOPQRSTUVWXYZ?\-]{1}	required	The residue that replaced the originalResidue.

Subelements: none

Example Context:

```
<SubstitutionModification location="15" originalResidue="X" replacementResidue="P"/>
```

6.80 Element <Threshold>

Definition: The threshold(s) applied to determine that a result is significant. If multiple terms are used it is assumed that all conditions are satisfied by the passing results.

Type: ParamListType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	1	unbounded	A single entry from an ontology or a controlled vocabulary.
userParam	1	unbounded	A single user-defined parameter.

Example Context:

```
<Threshold>
  <cvParam accession="MS:1001316" name="mascot:SigThreshold" cvRef="PSI-MS" value="0.05"/>
</Threshold>
```

cvParam Mapping Rules:

```
Path /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/Threshold
MUST supply a *child* term of MS:1001302 (search engine specific input parameter) one or more times
  e.g.: MS:1001005 (Sequest:CleavesAt)
  e.g.: MS:1001007 (Sequest:OutputLines)
  e.g.: MS:1001009 (Sequest:DescriptionLines)
  e.g.: MS:1001026 (Sequest:NormalizeXCorrValues)
  e.g.: MS:1001028 (Sequest:SequenceHeaderFilter)
  e.g.: MS:1001032 (Sequest:SequencePartialFilter)
  e.g.: MS:1001037 (Sequest:ShowFragmentIons)
  e.g.: MS:1001038 (Sequest:Consensus)
  e.g.: MS:1001042 (Sequest:LimitTo)
  e.g.: MS:1001046 (Sequest:sort_by_dCn)
  et al.
MUST supply a *child* term of MS:1001153 (search engine specific score) one or more times
  e.g.: MS:1001154 (Sequest:probability)
  e.g.: MS:1001155 (Sequest:xcorr)
  e.g.: MS:1001156 (Sequest:deltacn)
  e.g.: MS:1001157 (Sequest:sp)
  e.g.: MS:1001158 (Sequest:Uniq)
  e.g.: MS:1001159 (Sequest:expectation value)
  e.g.: MS:1001160 (Sequest:sf)
  e.g.: MS:1001161 (Sequest:matched ions)
  e.g.: MS:1001162 (Sequest:total ions)
  e.g.: MS:1001163 (Sequest:consensus score)
  et al.
MUST supply term MS:1001494 (no threshold) only once
MUST supply term MS:1001448 (pep:FDR threshold) only once
Path /MzIdentML/AnalysisProtocolCollection/ProteinDetectionProtocol/Threshold
MUST supply a *child* term of MS:1001302 (search engine specific input parameter) one or more times
  e.g.: MS:1001005 (Sequest:CleavesAt)
  e.g.: MS:1001007 (Sequest:OutputLines)
  e.g.: MS:1001009 (Sequest:DescriptionLines)
  e.g.: MS:1001026 (Sequest:NormalizeXCorrValues)
  e.g.: MS:1001028 (Sequest:SequenceHeaderFilter)
  e.g.: MS:1001032 (Sequest:SequencePartialFilter)
  e.g.: MS:1001037 (Sequest:ShowFragmentIons)
  e.g.: MS:1001038 (Sequest:Consensus)
  e.g.: MS:1001042 (Sequest:LimitTo)
  e.g.: MS:1001046 (Sequest:sort_by_dCn)
  et al.
MUST supply a *child* term of MS:1001153 (search engine specific score) one or more times
  e.g.: MS:1001154 (Sequest:probability)
  e.g.: MS:1001155 (Sequest:xcorr)
  e.g.: MS:1001156 (Sequest:deltacn)
  e.g.: MS:1001157 (Sequest:sp)
  e.g.: MS:1001158 (Sequest:Uniq)
  e.g.: MS:1001159 (Sequest:expectation value)
  e.g.: MS:1001160 (Sequest:sf)
  e.g.: MS:1001161 (Sequest:matched ions)
  e.g.: MS:1001162 (Sequest:total ions)
  e.g.: MS:1001163 (Sequest:consensus score)
  et al.
MUST supply term MS:1001447 (prot:FDR threshold) only once
MUST supply term MS:1001494 (no threshold) only once
```

Example cvParams:

```
<cvParam accession="MS:1001316" name="mascot:SigThreshold" cvRef="PSI-MS" value="0.05"/>
<cvParam accession="MS:1001494" name="no threshold" cvRef="PSI-MS"/>
```

6.81 Element <TranslationTable>

Definition: The table used to translate codons into nucleic acids e.g. by reference to the NCBI translation table.

Type: TranslationTableType

Attributes:

Attribute Name	Data Type	Use	Definition

id	xsd:string	required	An identifier is an unambiguous string that is unique within the scope (i.e. a document, a set of related documents, or a repository) of its use.
name	xsd:string	optional	The potentially ambiguous common identifier, such as a human-readable name for the instance.

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cvParam	0	unbounded	A single entry from an ontology or a controlled vocabulary.

Example Context:

cvParam Mapping Rules:

```
Path
/MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/DatabaseTranslation/Translation
Table
MUST supply term MS:1001410 (translation start codons) only once
MUST supply term MS:1001025 (translation table) only once
MUST supply term MS:1001423 (translation table description) only once
```

6.82 Element <cv>

Definition: A source controlled vocabulary from which cvParams will be obtained.

Type: cvType

Attributes:

Attribute Name	Data Type	Use	Definition
fullName	xsd:string	required	The full name of the CV.
id	xsd:string	required	The unique identifier of this cv within the document to be referenced by cvParam elements.
uri	xsd:anyURI	required	The URI of the source CV.
version	xsd:string	optional	The version of the CV.

Subelements: none

Example Context:

```
<cv id="PSI-MS" fullName="Proteomics Standards Initiative Mass Spectrometry Vocabularies"
  uri="http://psidev.cvs.sourceforge.net/viewvc/*checkout*/psidev/psi/psi-
ms/mzML/controlledVocabulary/psi-ms.obo"
  version="2.32.0"/>
<cv id="UNIMOD" fullName="UNIMOD" uri="http://www.unimod.org/obo/unimod.obo"/>
<cv id="UO" fullName="UNIT-ONTOLOGY"
  uri="http://obo.cvs.sourceforge.net/*checkout*/obo/obo/ontology/phenotype/unit.obo"/>
</cvList>
  ...
</cv>
```

6.83 Element <cvList>

Definition: The list of controlled vocabularies used in the file.

Type: CVListType

Attributes: none

Subelements:

Subelement Name	minOccurs	maxOccurs	Definition
cv	1	unbounded	A source controlled vocabulary from which cvParams will be obtained.

Example Context:

```
<cvList>
  <cv id="PSI-MS" fullName="Proteomics Standards Initiative Mass Spectrometry Vocabularies"
    uri="http://psidev.cvs.sourceforge.net/viewvc/*checkout*/psidev/psi/psi-
ms/mzML/controlledVocabulary/psi-ms.obo"
    version="2.32.0"/>
  <cv id="UNIMOD" fullName="UNIMOD" uri="http://www.unimod.org/obo/unimod.obo"/>
  <cv id="UO" fullName="UNIT-ONTOLOGY"
    uri="http://obo.cvs.sourceforge.net/*checkout*/obo/obo/ontology/phenotype/unit.obo"/>
  ...
```

```
</cvList>
```

6.84 Element <cvParam>

Definition: A single entry from an ontology or a controlled vocabulary.

Type: CVParamType

Attributes:

Attribute Name	Data Type	Use	Definition
accession	xsd:string	required	The accession or ID number of this CV term in the source CV.
cvRef	xsd:string	required	A reference to the cv element from which this term originates.
name	xsd:string	required	The name of the parameter.
unitAccession	xsd:string	optional	An accession number identifying the unit within the OBO foundry Unit CV.
unitCvRef	xsd:string	optional	If a unit term is referenced, this attribute MUST refer to the CV 'id' attribute defined in the cvList in this file.
unitName	xsd:string	optional	The name of the unit.
value	xsd:string	optional	The user-entered value of the parameter.

Subelements: none

Example Context:

```
<cvParam accession="MS:1001088" name="protein description" cvRef="PSI-MS"
  value="Rnd3psu|NC_LIV_111720 Decoy sequence, was | organism=Neospora_caninum | product=forkhead-
aasociated (FHA) domain-containing protein | location=Neo_chrIX:1702443-1710709(-) | length=1155"
/>
</DBSequence>
<DBSequence id="DBSeq_1_Rnd1psu|NC_LIV_080090" length="16207"
searchDatabase_ref="SDB_NeoProt_tripledecoy" accession="Rnd1psu|NC_LIV_080090">
<Seq>TPLRAFARSEISNPSLIVLDDVLSQSLSSPGSHLERLLPEAEPGRGAEIDTKPHKRTTAFFSLEHEAEAWPPAPPTPPPGNQYAPIDRVGSATGAD
DPPTAPFALVPPRRRNLGLAPELFALGSQQTRAGIYSGRVGATQNTGVGPSAQLAAQANPAELDTPSQLAGPGSLVARALWAPVTFFLENSSRSLSSNSSR
...
GGGLRAELAVNELLVEDRGPDNSNYVVAVKFPKAGAKECSRKKDKLDLHGSCDICAGMGTATSMSCTYA</Seq>
  ...
</cvParam>
```

6.85 Element <userParam>

Definition: A single user-defined parameter.

Type: UserParamType

Attributes:

Attribute Name	Data Type	Use	Definition
name	xsd:string	required	The name of the parameter.
type	xsd:string	optional	The datatype of the parameter, where appropriate (e.g.: xsd:float).
unitAccession	xsd:string	optional	An accession number identifying the unit within the OBO foundry Unit CV.
unitCvRef	xsd:string	optional	If a unit term is referenced, this attribute MUST refer to the CV 'id' attribute defined in the cvList in this file.
unitName	xsd:string	optional	The name of the unit.
value	xsd:string	optional	The user-entered value of the parameter.

Subelements: none

Example Context:

```
<userParam name="D:/Software/Databases/Neospora_3rndTryp/Neo_rndTryp_3times.fasta"/>
```

7. Specific Comments on schema

In this section, several points of documentation are elaborated beyond the core specification in Section 6.

7.1 File extension and compression

Con formato: Título 2, Ninguno, Interlineado: sencillo, Sin viñetas ni numeración, No conservar con el siguiente

It is noted that standard file compression algorithms greatly reduce the mzIdentML file sizes, speeding up file transfers and uploads / downloads. It is also noted that software implementing mzIdentML import or export will be expected to benefit in performance from working with compressed mzIdentML, since the compression and decompression algorithms are expected to give significant performance gains over disk access times for non-compressed files. As such, it is RECOMMENDED that mzIdentML files are compressed using gzip from all software that exports mzIdentML and software that imports SHOULD be expected to read gzipped files, as well as native (non-compressed) mzIdentML files. The file extension for native mzIdentML files SHOULD be ".mzid" and for compressed files SHOULD be "mzid.gz".

7.2 Referencing elements within the document

A number of elements within the schema have an attribute which is used to reference an element elsewhere in the file using the unique identifier of the referenced element. These attributes are named following the convention: "[elementName]_ref". The uniqueness of the value in the "id" attribute of elements is validated using xsd:key, and the integrity of the reference is validated using xsd:keyref, defined within the schema.

7.3 Searches against nucleotide sequences

Searches of Nucleic acid databases - The "seq" attribute on <DBSequence> SHOULD contain the nucleic acid sequence if a nucleic acid database was searched (rather than up to six translated sequences). <Peptide> represents the identified amino acid sequence (including modifications) and, as such, the <peptideSequence> elements SHOULD store the translated amino acid sequences. <PeptideEvidence> contains the DBSequence_Ref together with the translation frame and a TranslationTable_Ref attribute (see below). The Peptide_Ref is done in <SpectrumIdentificationItem> as in the case for an amino acid database. If a protein dectection is performed, there are <PeptideHypothesis> elements referencing <PeptideEvidence> elements from <SpectrumIdentificationItem> sections. For clarification, see the example instance document for a nucleic acid search (Section 5.4).

In the <SpectrumIdentificationProtocol>, <TranslationTable> is used to specify how nucleic acid sequences are translated into amino acid sequences as follows:

```
<DatabaseTranslation frames="1 2 3 -1 -2 -3">
  <TranslationTable id="TT_1" name="Standard">
    <cvParam accession="MS:1001025" name="translation table" cvRef="PSI-MS"
value="FFLLSSSSYY**CC*WLLLLPPPPHHQQRRRRIIIMTTTTNNKKSSRRVVVVAAAADDEEGGGG" />
    <cvParam accession="MS:1001410" name="translation start codons" cvRef="PSI-MS" value="---M---------
------M---------------M----------------------------" />
    <cvParam accession="MS:1001423" name="translation table description" cvRef="PSI-MS"
value="http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/index.cgi?chapter=cgencodes#SG1" />
  </TranslationTable>
  <TranslationTable id="TT_2" name="Vertebrate Mitochondrial">
    <cvParam accession="MS:1001025" name="translation table" cvRef="PSI-MS"
value="FFLLSSSSYY**CCWWLLLLPPPPHHQQRRRRIIMMTTTTNNKKSS**VVVVAAAADDEEGGGG" />
    <cvParam accession="MS:1001410" name="translation start codons" cvRef="PSI-MS" value="-------------
-------------------MMMM---------------M------------" />
    <cvParam accession="MS:1001423" name="translation table description" cvRef="PSI-MS"
value="http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/index.cgi?chapter=cgencodes#SG2" />
  </TranslationTable>
```

The attribute "frames" specifies which frames are considered and one or more translation tables can be specified using CV parameters. The translation table is defined here: http://www.ncbi.nlm.nih.gov/IEB/ToolBox/SDKDOCS/SEQFEAT.HTML#_Genetic_Codes:

"The genetic codes themselves are arrays of 64 amino acid codes. The index to the position in the array of the amino acid is derived from the codon by the following method:

index = (base1 16) + (base2 4) + (base3 1) where T=0, C=1, A=2, G=3”

The same encoding technique is used to specify start codons. Alphabet names are prefixed with "s" (e.g. sncbieaa) to indicate start codon arrays. Each cell of a start codon array contains either the gap code ("-" for ncbieaa) or an amino acid code if it is valid to use the codon as a start codon. Currently all starts are set to code for methionine, since it has never been convincingly demonstrated that a protein can start with any other amino acid. However, if other amino acids are shown to be used as starts, this structure can easily accommodate that information.

For each peptide, the frame and translation table should be specified in the PeptideEvidence:

```
<PeptideEvidence id=”1” TranslationTable_ref="TT_1" frame="1" />
```

7.4 Reporting peptide and protein identifications passing a significance threshold

The elements <SpectrumIdentificationItem> and <ProteinDetectionHypothesis> have a mandatory Boolean attribute passThreshold that allows a file producer to indicate that an identification has passed a given threshold or that it has been manually validated. Depending on the intended purpose of the file, the file producer MAY wish to report a number of identifications that fall below the given significance threshold, for example to allow global statistical analyses to be performed which are not possible if only identifications passing the threshold are reported. Thresholds for peptide-spectrum matches or for protein identification should be encoded as instances of <cvParam> within <SpectrumDetectionProtocol> or <ProteinDetectionProtocol> for example as follows. If the file producer does not want to indicate that a threshold has been set, all identifications MUST have passThreshold = “true” and the “no threshold” CV term should be given within the protocols.

```
<SpectrumIdentificationProtocol id="SIP" AnalysisSoftware_ref="AS_mascot_server">
       …
   <Threshold>
     <cvParam accession="MS:1001316" name="mascot:SigThreshold" cvRef="PSI-MS" value="0.05"/>
   </Threshold>


<ProteinDetectionProtocol id="PDP_MascotParser_1"  AnalysisSoftware_ref="AS_mascot_parser">
       …
  <Threshold>
    <cvParam accession="MS:1001316" name="mascot:SigThreshold" cvRef="PSI-MS" value="0.05"/>
  </Threshold>
```

7.5 Using decoy databases to set different thresholds of false discovery rate

mzIdentML supports the reporting of searches against decoy databases, constructed and searched using many of the currently known methods. A <SpectrumIdentificationItem> can be marked as matching a decoy peptide using the isDecoy attribute of the referenced <PeptideEvidence> element, thus allowing the false discovery rate to be calculated across an entire file. The DBSequence_Ref references the decoy protein record.

Implementers of the format SHOULD report the peptide identifications that pass the threshold they wish to communicate to a consumer of the data. For example, a threshold could be set by p-value, false discovery rate, by a native search engine score (or a more complex system documented with CV terms in <Threshold>), and those peptides reported (passing the threshold) are used to determine which proteins have been detected. It is not guaranteed that a consumer of an mzIdentML file will be able to calculate other results, or global false discovery rates, using different thresholds from the reported information, although in some circumstances they may be able to, for example, if a user reports the complete output of a search against a target and decoy search.

```
<SearchDatabase location="/localdirectory/18.E_coli_K12_edit.fasta" id="K12_nosignal" name="K12"
numDatabaseSequences="9376" releaseDate="01-2008-08-2008" version="1.0" >
       <FileFormat>
              <cvParam accession="MS:1001348" name="FASTA format" cvRef="PSI-MS"/>
       </FileFormat>
       <DatabaseName>
              <userParam name="18.E_coli_K12_edit.fasta" />
       </DatabaseName>
       <cvParam accession="MS:1001197" name="DB composition target+decoy" cvRef="PSI-MS"/>
       <cvParam accession="MS:1001283" name="decoy DB accession regexp" value="Rnd" cvRef="PSI-MS"/>
       <cvParam accession="MS:1001195" name="decoy DB type reverse" cvRef="PSI-MS"/>
</SearchDatabase>
```

```
<SpectrumIdentificationItem passThreshold="false" rank="1"
                        peptide_ref="HAVGGRYSSLLCK__57.0215@C$403;_"
                        experimentalMassToCharge="1448.756" chargeState="2" id="SII_6_1">
                        <PeptideEvidenceRef peptideEvidence_ref="PE6_2_4"/>
            <PeptideEvidence isDecoy="true" post="D" pre="K" end="404"
                start="392" peptide_ref="HAVGGRYSSLLCK__57.0215@C$403;_"
                dBSequence_ref="dbseq_REV_psu|NC_LIV_113200" id="PE6_2_4"/>
               <cvParam accession="MS:1001329" name="OMSSA:pvalue" cvRef="PSI-MS"  value="0.00073351"
/>

…

</SpectrumIdentificationItem>
```

7.6 Database Filter

The format can specify that a sequence database has been filtered, for example based on PI, protein mass, taxonomy or even a set of accession numbers for a second pass search. For example all animals except mice would be encoded as (NCBI:33208 is metazoa, NCBI:10090 is *Mus musculus*):

```
<DatabaseFilters>
  <Filter>
    <FilterType>
      <cvParam accession="MS:1001020" name="DB filter taxonomy" cvRef="PSI-MS" />
    </FilterType>
   <Include>
      <cvParam accession="MS:1001467" name="taxonomy: NCBI TaxID" cvRef="PSI-MS" value="33208"/>
   </Include>
   <Exclude>
     <cvParam accession="MS:1001467" name="taxonomy: NCBI TaxID" cvRef="PSI-MS" value="10090"/>
   </Exclude>
   </Filter>
</DatabaseFilters>
```

7.7 Types of parameters and values

There are several types for parameters that are used in the schema:

<ParamListType>: A list (i.e. unbounded number) of <ParamGroup>.
<ParamGroup>: A choice between <cvParam> or <userParam>.

<ParamType>: A single reference to <ParamGroup>, which allows a choice between either <cvParam> or <userParam> at the specified point in the schema.
<cvParamType>: A single entry from an ontology or a controlled vocabulary. Attributes: accession, cvRef, name, value, unitAccession, unitName, unitCvRef.
<userParamType>: A single user-defined parameter. Attributes: name, value, unitAccession, unitName, unitCvRef.

7.8 Reporting fragmentation ions

mzIdentML employs an array type structure to support the reporting of ion types identified in an MS/MS analysis, coupled with CV parameters to retain flexibility in the types of ion that can be reported. A brief example is given here to explain how these structures should be used where y11, y8 and y7 have been identified with charge = 2+. First, the types of measures to be reported are given in the <FragmentationTable> using <cvParam> instances. Second, each <SpectrumIdentificationItem> contains an index of values (11, 8 and 7 for each y ion) and parallel arrays that reference back to each <Measure> defined in the <FragmentationTable>. In the example, the y8 ion has a product ion m/z = 436.4, product ion intensity = 11 and product ion m/z error = 0.1284 (the second position in the index of each array).

```
<FragmentationTable>
  <Measure id="m_mz">
    <cvParam cvRef="PSI-MS" accession="MS:1001225" name="product ion m/z"/>
  </Measure>
  <Measure id="m_intensity">
    <cvParam cvRef="PSI-MS" accession="MS:1001226" name="product ion intensity"/>
  </Measure>
  <Measure id="m_error">
    <cvParam cvRef="PSI-MS" accession="MS:1001227" name="product ion m/z error"
unitAccession="MS:1000040" unitName="m/z" unitCvRef="PSI-MS"/>
```

```
  </Measure>
</FragmentationTable>
…

<IonType index="11 8 7" charge="2">
  <cvParam cvRef="PSI-MS" accession="MS:1001220" name="frag: y ion"/>
  <FragmentArray values="551.3 436.4 380.1 " measure_ref="m_mz"/>
  <FragmentArray values="800 11 46" measure_ref="m_intensity"/>
  <FragmentArray values="0.4752 0.1284 0.3704" measure_ref="m_error"/>
</IonType>
```

7.8.1 Internal fragments and immonium ions

mzIdentML supports the reporting of internal fragment ions, of which an immonium ion is a special case comprising a single side chain (http://www.matrixscience.com/help/fragmentation_help.html). For internal and immonium ions, the index is used in two different ways. Internal fragments are reported using the index structure to identify the start and end of the ion within the sequence. The example shows how the index performs this different role, as it identifies pairs of internal ions: ya2-5, ya3-7, ya3-8, ya4-8, ya5-8, ya5-11, ya8-11.

```
<IonType index="2 5 3 7 3 8 4 8 5 8 5 11 8 11" charge="1">
  <cvParam cvRef="PSI-MS" accession="MS:1001366" name="frag: internal ya ion"/>
  <FragmentArray values="315.2 388.1 501.4 444.1 342.8 669.901495 412.4 " measure_ref="m_mz"/>
  <FragmentArray values="44 63 10430 75 48 6420 31" measure_ref="m_intensity"/>
  <FragmentArray values="-0.0027 -0.1191 0.0969 -0.1817 -0.4340 0.4721 0.1082" measure_ref="m_error"/>
</IonType>
```

For immonium ions, the index is the position of the identified ion within the peptide sequence. If the peptide contains the same amino acid in multiple positions that cannot be distinguished, all positions should be given. Example, where immonium ions have been found matching T and G in the following peptide sequence FGGEENTY (positions 2 or 3, and position 7):

```
<IonType cvRef="PSI-PI" accession=" MS:1001239" name="frag: immonium ion" index="2 3 7" charge="1">
  <FragmentArray values="288.2 286.1 387.2 371.127841 " measure_ref="m_mz"/>
  <FragmentArray values="2137 83 656 1663" measure_ref="m_intensity"/>
  <FragmentArray values="0.0260 -0.1125 -0.0602 -0.1011" measure_ref="m_error"/>
</IonType>
```

7.9 Enzyme definition

The <SpectrumIdentificationProtocol> SHOULD contain a specification of which enzyme (if any) was applied in the search. The element <Enzyme> has optional sub-elements for specifying the <EnzymeName> using a CV term and the cleavage site, using a regular expression. Regular expressions should be encoded following the notation of Perl Compatible Regular Expressions (PCRE regex, http://www.pcre.org, matching the syntax and semantics of Perl version 5). The PSI-MS CV contains terms for the most common enzymes with pre-defined regular expressions (Table 4~~Table 2~~). If the enzyme used is present in the PSI-MS CV, the term MUST be encoded under <EnzymeName> unless the rule given in the CV does not match that used by the software or if the enzyme used is not present in the CV, in which case the regular expression used MUST be given in the element <SiteRegexp>. If the <EnzymeName> element is used, the regular expression MAY also be provided additionally. For a no enzyme search, (i.e. one where there may be a cleavage at any residue), the cvTerm MS:1001091 'NoEnzyme' MUST be specified, and the missedCleavages and semiSpecific attributes SHOULD NOT be specified. If two or more enzymes are used, multiple <Enzyme> elements SHOULD be provided rather than trying to build a regular expression covering all cleavage sites. If the software uses a name for an enzyme other than the one specified in the CV, a userParam MAY also be given.

The following guidelines SHOULD be followed when generating regular expressions in an instance document for enzymes not present in the CV: 1) use the PCRE supplied negation syntax for look-ahead and look-behind assertions and 2) use the most compact representation possible for a regex. The start of a match specifies the cleavage point. For example the enzyme Trypsin, which cleaves following a K or R residue unless the next residue is P, has the regular expression:

```
(?<=[KR])(?!P)
```

The ?<= is a "zero-width positive look-behind assertion", and [] means one of this character set. So, this rule is to look behind for a K or R. ?! is a zero-width positive look-ahead assertion, and ?!P means any character that is not P. An example of an "N-term" enzyme is Asp-N which cleaves before D or B. This can be described using the PCRE:

```
   (?=[BD])
```

The ?= is a "zero-width positive look-ahead assertion."

A simple 3 line perl program can be written to test a regular expression:

```
 $protein = "ABCDKPEFGHIJKLMNOPQRSTUVWXYZ";
 @peptides = split(/(?<=[KR])( ?!P)/, $protein);
 print join "\n", @peptides;
```

The program returns:

```
ABCDKPEFGHIJK
LMNOPQR
STUVWXYZ
```

Enzyme Name	Regular expression
Trypsin	(?<=[KR])(?!P)
Arg-C	(?<=R)(?!P)
Asp-N	(?=[BD])
Asp-N_ambic	(?=[DE])
Chymotrypsin	(?<=[FYWL])(?!P)
CNBr	(?<=M)
Formic_acid	((?<=D))|((?=D))
Lys-C	(?<=K)(?!P)
Lys-C/P	(?<=K)
PepsinA	(?<=[FL])
TrypChymo	(?<=[FYWLKR])(?!P)
Trypsin/P	(?<=[KR])
V8-DE	(?<=[BDEZ])(?!P)
V8-E	(?<=[EZ])(?!P)
Leukocyte elastase	(?<=[ALIV])(?!P)
Proline endopeptidase	(?<=[HKR]P)(?!P)
Glutamyl endopeptidase	(?<=[^E]E)
2-iodobenzoate	(?<=W)

Table 44~~2~~ Common enzymes and the cleavage site specified as regular expressions as represented in the PSI-MS CV.

7.10 Unknown modifications

In version 1.1.0 onwards of mzIdentML there has been a change with respect to how "unknown modifications" (i.e. those not present in an allowed CV) are reported on peptides. In version 1.0, userParam elements were allowed on Peptide to capture these modifications. In version 1.1.0 onwards, only cvParam elements can be given on Peptide and a term "unknown modification" has been added to the PSI-MS CV. This term MUST only be used if the identified modification is not present in UNIMOD (or other allowed CV), according to the identity of the residue modified and the delta mass, within the parent tolerance specified in the search. The semantic validator will check any uses of the "unknown modification" term (MS:1001460) and reject files if the modification is present in UNIMOD.

8. Conclusions

This document contains the specifications for using the mzIdentML format to represent results from peptide and protein identification pipelines, in the context of a proteomics investigation. This specification, in conjunction with the XML Schema, mapping file and CV constitute 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.

9. Authors and Contributors

Authors of this specification:
David Creasy, Matrix Science
Florian Reisinger, European Bioinformatics Institute
Johannes Griss, European Bioinformatics Institute
Juan Antonio Vizcaíno, European Bioinformatics Institute
Matthew Chambers, Vanderbilt University Medical Center
Gerhard Mayer, Medizinisches Proteom-Center, Ruhr-Universität Bochum
Martin Eisenacher, Medizinisches Proteom-Center, Ruhr-Universität Bochum
Andrew Jones, University of Liverpool

Correspondence - andrew.jones@liv.ac.uk

The mzIdentML version 1.0 authors were as follows:
Angel Pizarro, Center for Bioinformatics, University of Pennsylvania
David Creasy, MatrixScience
Phil Jones, European Bioinformatics Institute
Andreas Bertsch, Eberhard Karls University Tübingen
Jenny Siepen, University of Manchester
Martin Eisenacher, Medizinisches Proteom-Center, Ruhr-Universität Bochum
Andrew Jones, University of Liverpool

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

- Eric Deutsch, Institute for Systems Biology
- Simon Hubbard, University of Manchester
- Julian Selley, University of Manchester
- Zsuzsanna Bencsath-Makkai, Biomedical Engineering, McGill University
- Sean Seymour, Applied Biosystems
- Randy Julian, IndigoBio
- Pierre-Alain Binz, GeneBio Geneva
- Alex Masselot, GeneBio Geneva
- Lennart Martens, European Bioinformatics Institute
- Henning Hermjakob, European Bioinformatics Institute
- Luisa Montecchi, European Bioinformatics Institute
- Richard Côté, European Bioinformatics Institute
- Marc Sturm, Eberhard Karls University, Tübingen
- Jim Shofstahl, Thermo Fisher
- David Horn, Agilent
- Jimmy Eng, Fred Hutchinson Cancer Research
- Brian Searle, Proteome Software
- Phillip Young, Waters
- Michael Kohl, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Germany
- Christian Stephan, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Germany
- Eugene Kapp, Ludwig Institute for Cancer Research
- Michael Coleman, Stowers Institute
- Julian Uszkoreit, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Germany
- Oliver Kohlbacher, Eberhard Karls University, Tübingen
- Mathias Walzer, Eberhard Karls University, Tübingen
- David Ovelleiro, European Bioinformatics Institute
- Alberto Medina, ProteoRed Consortium, Spain

- Salvador Martinez, ProteoRed Consortium, Spain
- Laurent Gatto, University of Cambridge

10. References

- [RFC2119] Bradner, S. (1997). "Key words for use in RFCs to Indicate Requirement Levels, Internet Engineering Task Force, RFC 2119, http://www.ietf.org/rfc/rfc2119.txt.
- [Jones 07] Jones AR, Miller M, Spellman P and Pizarro A. Specification documentation for the Functional Genomics Experiment (FuGE) model: user guide. Version 1 (final): http://fuge.sourceforge.net/dev/V1Final/FuGE-v1-SpecDoc.doc.
- [Deutsch08] Deutsch EW, Martens L, Montecchi-Palazzi L, Binz P-A, Kessner D, Souda P mzML: Mass Spectrometry Markup Language, http://www.psidev.info/index.php?q=node/303.

11. 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