PEFF: A Common Sequence Database Format for Proteomics

Status of this document

This document provides information to the proteomics community about a common sequence database format for proteomics. Distribution is unlimited.

Version Draft 24 - this is a draft of version 1.0

# 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. This document presents a unified format for protein and nucleotide sequence databases to be used by sequence search engines and other associated tools (spectra library search tools, sequence alignment software, data repositories, etc.). This format enables consistent extraction, display and processing of information such as protein/nucleotide sequence database entry identifier, description, taxonomy, etc. across software platforms. It also allows the representation of structural annotations such as post-translational modifications, mutations and other processing events. The proposed format has the form of a flat file that extends the formalism of the individual sequence entries as presented in a FASTA format and that includes a header of meta data to describe relevant information about the database(s) from which the sequence has been obtained (i.e., name, version, etc.). The format is named PEFF (PSI Extended FASTA Format). Sequence database providers are encouraged to generate this format as part of their release policy or to provide appropriate converters that can be incorporated into processing tools.

Contents

[Abstract 1](#_Toc482532565)

[1. Introduction 2](#_Toc482532566)

[1.1 Description of the need 2](#_Toc482532567)

[1.2 Requirements 2](#_Toc482532568)

[1.3 Issues to be addressed 3](#_Toc482532569)

[2. Notational Conventions 3](#_Toc482532570)

[3. The Format Implementation 3](#_Toc482532571)

[3.1 The documentation 3](#_Toc482532572)

[3.2 Relationship to other specifications 4](#_Toc482532573)

[3.3 The common sequence database format description 4](#_Toc482532574)

[3.4 Additional considerations 12](#_Toc482532575)

[4. Authors Information 13](#_Toc482532576)

[5. Contributors 13](#_Toc482532577)

[6. Intellectual Property Statement 14](#_Toc482532578)

[7. Copyright Notice 14](#_Toc482532579)

[8. Glossary 14](#_Toc482532580)

[9. References 14](#_Toc482532581)

# Introduction

## Description of the need

One of the main goals of proteomics is to identify and quantify proteins in complex biological samples. This is achieved using mass spectrometry (MS) as a major analytical tool and sequence search engines as the bioinformatics interpretation tool. Sequence search engines aim at matching experimental MS spectra with protein or peptide sequences from a protein or nucleotide sequence database. Thousands of copies of sequence databases are searched by so called sequence search algorithms in proteomics labs all over the world. These algorithms regularly need to download the databases in the available formats; then they extract information including an identifier, taxonomy, description and sometimes other information such as alternative splicing variants, sequence processing leading to active forms and post-translational modifications in addition to the sequence itself. Most of the software convert the original format into a vendor-specific format to process the data. Currently available sequence databases are made available in FASTA format [Pearson88] (<http://en.wikipedia.org/wiki/FASTA_format>, <http://www.ncbi.nlm.nih.gov/BLAST/fasta.shtml>) or in other native formats (UniProtKB/Swiss-Prot and UniProtKB/TrEMBL in .dat or even XML for instance [THE\_UNIPROT\_CONSORTIUM1] [APWEILER1]). For the same database, the information might be richer or poorer according to the format. For instance, the current FASTA format does not generally store information such as splicing forms, mutations or post-translational modifications. To access information about these, one needs to choose another format, for instance a richer XML format, or for UniProtKB the native .dat format (<http://www.expasy.org/sprot/userman.html>).

Mass spectrometry-based peptide identification software tools deliver, in their graphical interfaces or their export formats, protein and peptide hits with information such as a protein accession code, sequence coverage, matching score, taxonomy and description. The same entry identified by different tools is not necessarily displayed in a unique manner, which renders it difficult, if not impossible, to map results between the tools. One reason for this is that these tools do not “parse” and interpret the database content in a consistent manner. In order to create a standardized manner to represent a protein in a search engine result (entry identifier, description, taxonomy, etc.), and to enable a consistent link to a protein from third party software, we are proposing a unified format for sequence databases that can be interpreted in a uniform manner by all sequence search software and other associated tools. Converters generated by the database providers or elsewhere have to be made available and maintained for the generation and parsing of these databases.

## Requirements

The main requirements to be fulfilled are:

* The format should allow more than one sequence database to be represented in one flat file.
* The format should require minimal changes to the existing parsers.
* The format should formalize the representation of all non-sequence associated information (identifiers, description, taxonomy, other structural or functional annotation data).
* The format should include meta-information about the database itself (name, version, type of content, etc.).
* Controlled vocabularies (CVs) should be pragmatically used for keys and values (i.e. database names, prefixes, entry keys such as NcbiTaxId, Protein/Gene Name).
* The format should be compatible with MIAPE guidelines (<http://www.psidev.info/miape>), for instance MIAPE MSI.

## Issues to be addressed

The main issues to be addressed by the format are:

* Definition lines in FASTA and other formats vary widely for no good reason. This causes problems for end users who want to use these files with protein identification tools. The creators of these tools are faced with a significant challenge to support all of these variations while consistently extracting the same information.
* The same database file is variably processed in different search engines. A given database entry can contain multiple identifiers, which can lead to variably interpreted identifiers, which renders difficult the mapping of identical entries in different tools (for instance the UniProtKB/Swiss-Prot AC: **P02768** vs. Swiss-Prot ID: ALBU\_HUMAN).
* The same protein (and therefore also primary sequence) in different databases can have very different identifiers (for example, **P02768** in UniProtKB/Swiss-Prot, NX\_P02768 in neXtProt, gi|113576|sp|P02768.2|ALBU\_HUMAN in NCBI, and ENSP00000295897 in Ensembl.
* The identifier information extracted from the FASTA formats is heterogeneous (gi|113576 vs 113576 vs sp|P02768 vs gi|113576|sp|P02768.2|ALBU\_HUMAN etc.). The definition and format description of the identifier should come from the DB provider (documentation).
* Description and availability of taxonomy are also heterogeneous and need to be properly interpreted (Latin names, common names, NCBI TaxID).
* Choice of the description string (variations include full or partial description, including or not taxonomy information, alternative names, truncation at a defined number of characters, etc.).
* Version name or date of a specific database is often requested for traceability purposes and to allow reproducibility of results obtained from the use of a given database (number of entries, protein or gene names, descriptions, sequences, PTMs, etc. vary from one version to another).
* It should be possible to store more than one sequence database in a single flat file. As identifiers might be identical in two or more “merged” databases, a mechanism should be defined to avoid this.

# Notational Conventions

The key words “MUST“, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in RFC 2119 [BRADNER1].

# The Format Implementation

## The documentation

The documentation of the format is divided into several documents and files. These files are available from the main format description page on the HUPO-PSI website (<http://www.psidev.info/peff>).

* Main specification document (this document)
* Controlled Vocabulary (CV). The CV terms applicable for PEFF are currently still in a custom PEFF CV. However, all terms will soon move to become part of the PSI-MS CV (https://github.com/HUPO-PSI/psi-ms-CV/blob/master/psi-ms.obo). The term names will not change, but the accession numbers (rarely used) will change.
* Example files.
* Reference to example implementations

## 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>) The “Minimum Information About a Proteomics Experiment: Mass Spectrometry Informatics” document identifies the minimum information required to report the use of a mass spectrometry-based peptide and protein identification and characterization experiment. It is expected that the common sequence database format will be used to capture requirements specified in MIAPE MSI. However, the format does not enforce MIAPE compliance itself and may be valid and useful without being fully MIAPE compliant.
2. *mzIdentML* (<http://www.psidev.info/mzidentml>). The mzIdentML specification is developed by PSI as a standard to capture the output of search engines that assign mass spectra to protein or peptide sequences. For searches performed using a PEFF file, the downstream result in mzIdentML will need to encode a reference to the PEFF file used.
3. *mzTab* (<http://www.psidev.info/mztab>). The mzTab specification is developed by PSI as a standard to report proteomics and metabolomics results in a tab-delimited text file format. For searches performed using a PEFF file, the downstream result in mzTab will need to encode a reference to the PEFF file used.

## The common sequence database format description

The format has the form of a text file with two sections, a file header section and a section containing the individual sequence entries. The two sections MUST be placed in the following order

* Section 1: The file header section.
* Section 2: The individual sequence entries section.

The characters allowed are the set of ASCII characters. A more constrained set of characters can be defined for specific sections of the file.

All lines in the file MUST end with LF (ASCII 10). A CR (ASCII 13) MAY precede the LF and should be ignored by parsers.

Descriptors of the information are defined as Controlled Vocabulary (CV) terms. The CV repository is available in OBO format at <https://raw.githubusercontent.com/HUPO-PSI/PEFF/master/CV/psi-peff.obo>. Note that before final release, it is expected to move these terms into the main PSI-MS CV, but this is not yet done.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | File header section | | |  |
|  |  |  |  |  |
|  |  | File Description block |  |  |
|  |  | Sequence database description block 1 |  |  |
|  |  | ..  Sequence database description block m |  |  |
|  |  |  |  |  |
|  | Individual sequence entries section | | |  |
|  |  |  |  |  |
|  |  | Sequence Entry 1 from sequence database 1 |  |  |
|  |  | …  Sequence Entry n from sequence database 1 |  |  |
|  |  | … |  |  |
|  |  | Sequence Entry 1 from sequence database m |  |  |
|  |  | …  Sequence Entry o from sequence database m |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Figure 1: Graphical representation of the PEFF file structure. In this example, the file has m databases, database 1 has n entries, database m has o entries

### PEFF file section 1: The file header section

The file header section contains all necessary information to describe and reference the represented sequence database(s). This includes information such as the database(s) name, source, version, size, sequence type, etc. This meta-data section includes mandatory and optional elements.

Format of the file header section

The file header section contains two types of information blocks: the file description block and the sequence database description block. The file header section MUST start with a file description block that MUST be followed by at least one sequence database description block. All lines in the file header section start with the character #, followed by a space (ASCII 32) character.

The format of the file description block is the following:

* The first line of this section is also the first line of the file. It MUST be

*# PEFF N.N*

where N.N represents the version number of the PEFF format, most likely 1.0. Parsers SHOULD check this value and compare it to what they are prepared to interpret;

* It MAY be followed by one of more general comment lines, which each have the following format:

*# GeneralComment=value* (where *value* is a string of text)

If there is a GeneralComment, it MUST not be empty

The format of the sequence database description blocks is as follows:

* All lines of a sequence database description block contain one piece of information.
* Each piece of information MUST have the following format:

*# key=value*

where the element *key* MUST be the value of a special attribute labelled “PEFFKey” attached to a CV term in the PSI-MS CV under the “PEFF keyword” branch. The format of the *value* is defined for each key in the CV.

* The block MUST start with a sequence database line description and follow the following format:

*# DbName=value,* (where *value* is the database name)

* The following five *key* elements MUST also be present:

Prefix; DbVersion, DbSource, NumberOfEntries, SequenceType

* Additional key=values pairs that are used in the sequence description blocks later in the document MUST be defined here using the SpecificKey key.
* A sequence database information block MUST end with the following separation line:

# //

One or more sequence description blocks MUST be present. Each sequence database description block has the following format:

Example:

# PEFF 1.0

# GeneralComment=This is a hand-crafted example comment

# DbName=neXtProt-extract

# Prefix=nxp

# DbDescription=extract of neXtProt with manual modifications

# Decoy=false

# DbVersion=2016-01-11

# DbSource=<http://www.nextprot.org>

# NumberOfEntries=62

# SequenceType=AA

# //

# DbName=myDB

# Prefix=md

# DbDescription=manually annotated FGF21 proteoforms from top-down experiment PX12345

# DbVersion=1.1

# DbSource=

# NumberOfEntries=2

# SequenceType=AA

# ProteoformDb=yes

# //

Non mandatory key-value pairs in the file header section MAY be used to add meta-data on the database description level (see section 3.3.2 and 3.3.3). They MUST be used to define keys that are not declared in the CV and used in the individual sequence database entries. This might include information such as protein function, ligands, links to experimental evidences, other custom-defined information. They also can imply an impact on the interpretation of the data provided in the individual sequence database section (sequence and annotation).

### Defining custom keys in the sequence database description block for use in the sequence entries section

Most of the keys found in each of the individual sequence entries (described below in 3.3.3) are defined in the CV. However, it is possible to define custom keys that may be used within custom pipelines. It is recommended that, for PEFF files that will be exported publicly or for any generally reusable keys, any new keys be proposed to the PSI for inclusion in the CV. However, whenever a key that is not is the CV is used, it MUST be defined in the file header block like this:

*# SpecificKey=KeyName:”KEYDEFINITION”:VALUEREGEXP*

*KeyName* MUST be written using CamelCase

For example, to define a SecondaryStructure term:

# SpecificKey=SecondaryStructure:"Secondary structure element and position":\([0-9]+\|[0-9]+\|[\w:]\*\|\S+?\)

And then use in the sequence entries description line:

\SecondaryStructure=(617|673|ncithesaurus:C47937|Helix)

### File header section: recommendations for Proteoforms: The ProteoformDb=yes key-value pair

Specific proteoforms can be described in PEFF entries. In a proteoform, structural annotations such as PTMs, sequence variations and maturation events are to be considered as present. The key-value pair ProteoformDb=yes covers this use-case: If this key-value pair is provided in a sequence database description block, it implies that the encoded proteins are to be considered as specific proteoforms. If structural annotations such as ModRes, ModResUnimod, ModResPsi, VariantSimple, VariantComplex, Processed MUST be considered as present and not optional.

### Section 2: The individual sequence entries section

The individual sequence entries section contains the actual sequences, their associated identifiers and additional descriptors. The format is similar to a FASTA format. The informative elements appearing in the FASTA description lines are structured in the below described format. This section MUST immediately follow the file header section.

The format of each individual sequence entry is described below. The individual sequence entries are placed in one single block of individual sequence entries within a file. There MUST NOT be any empty lines between individual sequence entries.

Format of the individual sequence entries.

For each sequence entry:

* A sequence entry is composed of a description line and a sequence block line.
* The description line has the following structure:

*>Prefix:DbUniqueId \key=value \key=value …*

* The header line MUST start with *>Prefix:DbUniqueId* where *Prefix* is the database Prefix, as defined in the sequence database description block, of the corresponding sequence database. This is the unique mandatory information of the description line.
* The description line MAY include optional information, separated by at least one space character, each of them described as *\key=value* pairs.
  + The order of the *\key=value* pairs is not important.
  + The element *key* is a CV term. The format of the *value* is defined for each key in the CV repository.
  + The *value* can contain one item or a list of items. In the latter case, items are placed in parentheses: *(item1)(item2)…* There MUST NOT be spaces between items.

*Generic example: \key=(item1)(item2)*

* + In case *item* contains multiple components, the “|” (pipe character) MUST be used as separator between components. The item therefore has the form

*(component1|component2)*

* + Characters allowed for a key: Key: [A-Za-z0-9\_]; Use CamelCase .Characters allowed for an item (if not complex) or a component of an item: [A-Za-z0-9\_?]
* The description line MUST contain only a single *>Prefix:DbUniqueId \key=value* block. Some FASTA files such as the NCBI non-redundant (nr) database have been seen to have multiple headers per sequence separated by delimiter ASCII 001 (CTRL+A). It has been decided that PEFF does not support this and readers therefore do not need to support this. It is recommended either to split the header and create one entry for each such sequence header block or to make a selection of the most appropriate block to create a PEFF file.
* The sequence block has the following structure:
* The sequence block contains the actual sequence, coded as one-letter code for both protein and nucleotide sequences. Allowed characters are described in the table below [IUPAC1999; UniProtKB user manual]:

|  |  |
| --- | --- |
| **1 one-letter code** | **Amino acid name** |
| A | Alanine |
| R | Arginine |
| N | Asparagine |
| D | Aspartic acid |
| C | Cysteine |
| Q | Glutamine |
| E | Glutamic acid |
| G | Glycine |
| H | Histidine |
| I | Isoleucine |
| L | Leucine |
| K | Lysine |
| M | Methionine |
| F | Phenylalanine |
| P | Proline |
| O | Pyrrolysine |
| S | Serine |
| U | Selenocysteine |
| T | Threonine |
| W | Tryptophan |
| Y | Tyrosine |
| V | Valine |
| B | Aspartic acid or Asparagine |
| Z | Glutamic acid or Glutamine |
| X | Any amino acid |
| J | Leucine or Isoleucine |
| \* | Sequence interruption (stop codon, unknown linkage) |

* The sequence block MAY be a single long line with only a single line ending. We however suggest wrapping the sequences to 60-100 characters per line for better human readability.
* There MUST NOT be any blank lines in the individual sequence entries section.

**Generic illustration:**

>*Prefix:DbUniqueID1 \key=value \key=value*

SEQUENCESEQUENCE

>*Prefix:DbUniqueID2 \key=value \key=value*

SEQUENCESEQSEQUENCE

**Real example:**

>nxp:NX\_Q06418-1 \DbUniqueId=NX\_Q06418-1 \PName=Tyrosine-protein kinase receptor TYRO3 isoform Iso 1 \Gname=TYRO3 \NcbiTaxId=9606 \TaxName=Homo Sapiens \Length=890 \SV=135 \EV=357 \PE=1 \Processed=(1|40|signal peptide)(41|890|mature protein) \ModResPsi=(681|MOD:00048|O4'-phospho-L-tyrosine)(685|MOD:00048|O4'-phospho-L-tyrosine)(686|MOD:00048|O4'-phospho-L-tyrosine)(804|MOD:00048|O4'-phospho-L-tyrosine) \ModRes=(63||N-linked (GlcNAc...))(191||N-linked (GlcNAc...))(230||N-linked (GlcNAc...))(240||N-linked (GlcNAc...))(293||N-linked (GlcNAc...))(366||N-linked (GlcNAc...))(380||N-linked (GlcNAc...))(64||Disulfide)(117||Disulfide)(160||Disulfide)(203||Disulfide) \VariantSimple=(21|L)(68|R)(74|M)(85|K)(90|H)(95|G)(114|G)(119|E)(119|L)(129|R)(144|K)(156|S)(178|M)(185|S)(187|L)(200|I)(208|P)(210|D)(215|H)(228|S)(235|R)(240|I)(251|S)(260|L)(265|D)(273|G)(277|L)(283|Y)(290|S)(299|H)(302|S)(302|K)(303|V)(306|S)(311|H)(314|L)(331|T)(333|C)(333|H)(346|N)(348|K)(351|S)(352|D)(353|S)(371|D)(392|I)(396|I)(399|T)(416|C)(433|F)(445|S)(452|Q)(455|Q)(455|W)(468|V)(470|Q)(487|K)(489|K)(511|M)(521|S)(522|Q)(523|L)(533|Q)(542|S)(545|G)(549|G)(566|F)(567|G)(580|L)(590|N)(596|R)(600|I)(605|L)(619|Q)(623|K)(635|L)(638|N)(647|R)(648|F)(659|W)(669|L)(675|R)(690|R)(705|V)(717|T)(719|R)(723|C)(723|L)(728|C)(734|S)(750|C)(756|Q)(759|D)(773|S)(776|L)(777|A)(785|K)(788|S)(797|F)(815|V)(817|D)(819|M)(824|G)(829|N)(831|T)(833|N)(842|D)(169|I)(343|K)(620|T)(819|Q)(848|W)(875|R)

MALRRSMGRPGLPPLPLPPPPRLGLLLAALASLLLPESAAAGLKLMGAPVKLTVSQGQPV

KLNCSVEGMEEPDIQWVKDGAVVQNLDQLYIPVSEQHWIGFLSLKSVERSDAGRYWCQVE

DGGETEISQPVWLTVEGVPFFTVEPKDLAVPPNAPFQLSCEAVGPPEPVTIVWWRGTTKI

GGPAPSPSVLNVTGVTQSTMFSCEAHNLKGLASSRTATVHLQALPAAPFNITVTKLSSSN

ASVAWMPGADGRALLQSCTVQVTQAPGGWEVLAVVVPVPPFTCLLRDLVPATNYSLRVRC

ANALGPSPYADWVPFQTKGLAPASAPQNLHAIRTDSGLILEWEEVIPEAPLEGPLGPYKL

SWVQDNGTQDELTVEGTRANLTGWDPQKDLIVRVCVSNAVGCGPWSQPLVVSSHDRAGQQ

GPPHSRTSWVPVVLGVLTALVTAAALALILLRKRRKETRFGQAFDSVMARGEPAVHFRAA

RSFNRERPERIEATLDSLGISDELKEKLEDVLIPEQQFTLGRMLGKGEFGSVREAQLKQE

DGSFVKVAVKMLKADIIASSDIEEFLREAACMKEFDHPHVAKLVGVSLRSRAKGRLPIPM

VILPFMKHGDLHAFLLASRIGENPFNLPLQTLIRFMVDIACGMEYLSSRNFIHRDLAARN

CMLAEDMTVCVADFGLSRKIYSGDYYRQGCASKLPVKWLALESLADNLYTVQSDVWAFGV

TMWEIMTRGQTPYAGIENAEIYNYLIGGNRLKQPPECMEDVYDLMYQCWSADPKQRPSFT

CLRMELENILGQLSVLSASQDPLYINIERAEEPTAGGSLELPGRDQPYSGAGDGSGMGAV

GGTPSDCRYILTPGGLAEQPGQAEHQPESPLNETQRLLLLQQGLLPHSSC

### Recommendations on and order of the keys in a description line

After the sequence identifier, which MUST start the description line, there is no mandatory order for placing the keys. However it is recommended to place the potentially longer keys at the end of the description lines. These are typically: ModRes, ModResUnimod, ModResPsi, VariantSimple, VariantComplex, The Length key should be provided.

Several key-values allow for an OptionalTag that MAY be specified. When present, it MUST be be placed at the end of the value and separated from the value by a pipe Character (“|”).If a tag is not provided, the trailing pipe character (“|”) MUST NOT be written. The tags are free text strings that are not constrained by a controlled vocabulary. It is recommended that this OptionalTag be used for evidence purposes primarily. The terminology or vocabulary used for that purpose MAY be declared in the file header section via the CustomTag=Tag:TagDescription construction.

In general, and by default, molecular features (such as ModRes, ModResUnimod, ModResPsi, VariantSimple, VariantComplex, Processed) encoded in keys SHOULD be considered as features that CAN be applied to the sequence. In the case these need to be reported as MUST be present and applied to the sequence, the sequence database section MUST contain a ProteoformDb=true *key-value* pair (see section 3.3.3).

### Definition of complex header keys

Most keys in the CV are self-explanatory in the CV itself. However, some terms are sufficiently complex and central to the format that they are described in detail in this document in the following sections.

In all header keys that allow an optional tag component, this optional tag is placed as the last component, example: (item|item|…|OptionalTag). The optional tag MAY be specified or not, as desired by the writer. If such a tag is not provided, the trailing pipe character (“|”) MUST NOT be written. The tags are free text strings that are not constrained by a controlled vocabulary. The tags MAY be defined in the file header via the CustomTag=Tag:TagDescription keyword (Tag is the text string of the tag and the TagDescription may be used to further describe the meaning of that tag.)

### Variant header key

The header key “Variant” was deprecated in 2015 during final development of the format in favor of “VariantSimple” and “VariantComplex”. Some PEFF files, e.g. from neXtProt, were produced with the “Variant” header key before it was deprecated. This term should no longer be used.

### VariantSimple header key

The header key “*VariantSimple*” is used to encode all single-amino acid substitutions. The format of the value for this term is *(position|newAminoAcid|optionalTag),* e.g. “(223|A)” or “(225|C|dbSNP)”. The first example indicates that at position 233 (count starting at 1) the default amino acid in the sequence may be substituted with the amino acid A, and the second example shows that at position 225 the default amino acid in the sequence may be substituted with the amino acid C (and that change is tagged with the string “dbSNP”. The position MUST be greater than 0 and less than or equal to the length of the protein. This key may not be used to extend a protein. The “*newAminoAcid*” part of the value must be a valid amino acid code (ambiguity codes such as J or X are permitted) or an asterisk (\*). It may not be empty, or space, or any non-alphabetic character except asterisk. The asterisk is to be interpreted as a nonsense mutation (stop codon) over which a peptide sequence may not span. Regular expressions may not be used. Insertions or deletions (indels) MUST NOT be specified with this term.

The rationale for separating these variants into a separate term from more complex variants is to more easily allow reader software and sequence search engines to support these simple variations in advance of more complex variations, which are considerably more difficult to implement.

### VariantComplex header key

The header key “*VariantComplex*” is used to encode all sequence variations more complex than a single amino acid substitution. The format of the value for this key is (*startPosition|endPosition|newSequence|optionalTag*). Variations that can fit the description of a VariantSimple MUST NOT be encoded using this term. See the table below for a series of examples, both legal and illegal. Position counting begins with 1.

|  |  |
| --- | --- |
| Example Value | Interpretation |
| (100|100|) | Position 100 may be nothing, signifying a single amino acid deletion. No character should be used to denote deletions, i.e. no dashes (-) or any other characters. |
| (100|100||10kexomes) | Same as above, but labeled with a tag “10kexomes”. |
| (100|102|) | A 3-AA deletion starting at position 100. |
| (100|100|APT) | A replacement of the original residue at position 100 by APT. It represents X -> APT, where X can be any residue. Examples are: 1) A->APT where A is the residue at position 100 in the sequence line; this corresponds to an insertion of PT after A; 2) L->APT where L is the residue at position 100 in the sequence line; this corresponds to a replacement of L by APT (for instance in an alternative splicing between exons. For an insertion, the following convention SHOULD be used: inserted amino acids SHOULD come after the residue defined in the sequence line. In the example 1 above, a PT is inserted after A at position 100 |
| (100|100|A) **ILLEGAL** | Not a legal VariantComplex. This must be encoded as a VariantSimple. |
| (100|102|KPA) | A 3-AA substitution as a cassette. If the AAs can be substituted individually, then they should be encoded as 3 separate VariantSimple entries. |
| (100|101|P) | A deletion and substitution. AAs at position 100 and 101 are both removed and replaced with a single P. Neither position was originally a P. If either position already had a P, then either (100|100|) or (101|101|) SHOUD be used. |
| (100|100|[AEQ]P) | An insertion before the P originally at position 100 with any of A or E or Q. |
| **ILLEGAL** | Not a legal VariantComplex. This must be encoded as three separate VariantComplex. No regular expressions are allowed in this item. |

### ModResUnimod header key

The header key “ModResUnimod” is used to encode mass modifications on amino acids (residues) using the Unimod controlled vocabulary. Two other terms (ModResPsi and ModRes) are used for other controlled vocabularies. The format of this term is (position|accession|name|OptionalTag). If the specified position cannot take on the specific amino acid modification in its default or variant form, this is an error in the file. If the sequence entry has a variant that is modified (for instance an alanine -> O-phospho-L-serine), a new entry MUST be created that contains this variant (i.e. serine) in the main sequence. In this case the modified residue (O-phospho-L-serine) can be added in the new entry. The specified modification name should be the one found in the “name:” field in the OBO file, not a synonym. See the table below for a series of examples, both legal and illegal. The position counting begins with 1. Unimod entries that specify an amino acid substitution MUST not be used. The \VariantSimple mechanism MUST be used instead. The tags MAY be defined in the file header via the CustomTag keyword as described in section 3.3.4.

|  |  |
| --- | --- |
| Example Value | Interpretation |
| (100|UNIMOD:21|Phospho) | Potential phosphorylation on position 100 |
| (100||Phospho) ILLEGAL | Not legal. The UNIMOD:nn accession must be provided |
| (100|UNIMOD:21|) ILLEGAL | Not legal. The full name from the OBO file (or equivalent) must be provided |

### ModResPsi header key

The header key “ModResPsi” is used to encode mass modifications on amino acids (residues) using the PSI-MOD controlled vocabulary. Two other terms (ModResUnimod and ModRes) are used for other controlled vocabularies. The format of this term is (position|accession|name|OptionalTag). See the table below for a series of examples, both legal and illegal. Position counting begins with 1. Note that the ModResPsi CV entry encodes the amino acid that is modified. If the specified position cannot take on the specific amino acid modification in its default or variant form, this is an error in the file. If the sequence entry has a variant that is modified (for instance an alanine -> O-phospho-L-serine), a new entry MUST be created that contains this variant (i.e. serine) in the main sequence. In this case the modified residue (O-phospho-L-serine) can be added in the new entry. The specified modification name should be the one found in the “name:” field in the OBO file, not a synonym. The tags MAY be defined in the file header via the CustomTag keyword as described in section 3.3.4.

|  |  |
| --- | --- |
| Example Value | Interpretation |
| (100|MOD:00046|O-phospho-L-serine) | Potential phosphorylation of a serine at position 100 |
| (100||O-phospho-L-serine) ILLEGAL | Not legal. The MOD:00046 accession must be provided |
|  |  |
| (100|MOD:00046|) ILLEGAL | Not legal. The full name from the OBO file (or equivalent) must be provided. |
| (?|MOD:00046|O-phospho-L-serine) | A phosphoserine for which a position is unknown. If a position range is known, it MAY be encoded in the Optional tag component; however a reader is not supposed to be able to interpret this. |

### ModRes header key

The header key “ModRes” is used to encode mass modifications on amino acids (residues) where a controlled vocabulary entry is available in neither Unimod nor PSI-MOD, or for custom applications. Two other terms (ModResPsi and ModResUnimod) are preferred and should be used when possible. The format of this term is (position|accession|name|OptionalTag). See the table below for a series of examples, both legal and illegal. Position counting begins with 1. The accession field may be empty if no accession number is available. However, the name field must be provided. Since no amino acid may be specified, the modification is presumed to apply to all possible residues in that position, unless specified in the custom lookup file. If the reading software can understand the modification, it is up to the reading software to ensure that the modification is applicable to the target residue. The tags MAY be defined in the file header via the CustomTag keyword as described in section 3.3.4.

|  |  |
| --- | --- |
| Example Value | Interpretation |
| (100||N-linked (GlcNAc...)) | The amino acid at position 100 has N-linked glycosylation modification/s of unknown composition. |
| (100||Disulfide) | The amino acid at position 100 is half of a disulfide modification. This is probably not useful for sequence database searching, but may be valuable information for other purposes. |
| (100|CustomMod:22|Floxilation) | The amino acid at position 100 has a potential floxilation modification as described in a custom CV. This will not be usable by most reading software, but could potentially be used by custom workflows. |
| (100||Phosphorylation) | The amino acid at position 100 has a potential phosphorylation. Note that although this is permitted, the use of either ModResPsi or ModResUnimod CV when available is strongly encouraged. |

### Processed header key

The header key “Processed” is used to encode post-translational processing of the protein, such that the mature form of the protein is only a subset of the entire provided sequence. The format of this term is (startPosition|endPosition|accession|name|OptionalTag). See the table below for a series of examples, both legal and illegal. Position counting begins with 1. The coordinates are presumed to apply to the default sequence, not taking into account possible indels.

|  |  |
| --- | --- |
| Example Value | Interpretation |
| (1|40|PEFF:1027|signal sequence) | Residues 1-40 are a signal peptide sequence that is cleaved off after translation |
| (41|890|PEFF:1028|mature protein) | Residues 41-890 are the mature form of the protein after the signal sequence is removed |
| (1|40||signal sequence) ILLEGAL | Not legal; an accession number from the PEFF CV must be provided. |
| (1|40|PEFF:1027|) ILLEGAL | Not legal; the term name from the PEFF CV must be provided. |

## Additional considerations

### Representation of splicing variants

When splicing variants (alternative exon splicing products) are to be represented for a given gene/protein in a sequence database, they SHOULD be represented in separate sequence entries; in this case, the DbUniqueId MUST be different for each of these sequence entries. Such corresponding sequences MAY be discriminated by a different suffix (>sp:P01234-1 and >sp:P01234-2).

### Representation of processed sequences

### Processed sequences (removal of precursor peptide, active chain, etc.) SHOULD be represented with annotations in the sequence description line. In cases where reading software cannot interpret this annotation, or in cases where the complexity of interpretation of additional annotation (such as active forms specific PTMs), processed sequences MAY be represented in separate sequence entries; in this case, the DbUniqueId MUST be different for each of these sequence entries.

### File extension

The suggested file extension is .peff (PSI Extended FASTA Format).

# Authors Information

Pierre-Alain Binz

CHUV Centre Universitaire Hospitalier Vaudois, CH-1011 Lausanne 14, Switzerland

pierre-alain.binz@chuv.ch

Sean L. Seymour

Seymour Data Science, USA

sean@seymourdatascience.com

Eugene A. Kapp

Walter & Eliza Hall Institute of Medical Research and the University of Melbourne, Australia

kapp@wehi.edu.au

Jim Shofstahl

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

jim.shofstahl@thermofisher.com

David Creasy

Matrix Science Ltd | 64 Baker Street | London W1U 7GB | UK

dcreasy@matrixscience.com

Lydie Lane

SIB Swiss Institute of Bioinformatics, 1 Michel-Servet CH-1211 Genève 14, Switzerland

Lydie.Lane@isb-sib.ch

Harald Barsnes

Proteomics Unit, Department of Biomedicine, University of Bergen, Norway

harald.barsnes@uib.no

Matt Chambers

Vanderbilt University, Nashville, Tennessee

matt.chambers@vanderbilt.edu

Robert Chalkley

University of California, San Francisco

chalkley@cgl.ucsf.edu

Eric W. Deutsch

Institute for Systems Biology, Seattle WA, USA

edeutsch@systemsbiology.org

# Contributors

In addition to the authors, a number of additional contributions have been made during the preparation process. The contributors who actively participated to the recommendation documentation are:

Members of the UniProt consortium that mapped the proposal with UniProt:

Nicole Redaschi, Swiss Institute of Bioinformatics, Swiss-Prot group, Geneva, Switzerland

Maria Jesus Martin, European Bioinformatics Institute, Hinxton, UK

Claire O Donovan, European Bioinformatics Institute, Hinxton, UK

Peter McGarvey, Protein Information Resource, Washington, USA

Amos Bairoch, Swiss Institute of Bioinformatics, CALIPHO group, Geneva, Switzerland

Philip C Andrews, University of Michigan, Ann Arbor, MI, USA

Gerben Menschaert, Ghent University, Ghent, Belgium

Karl Clauser, Broad Institute, Cambridge MA, USA,

Joel ??, ??

Daril ??, ??

David ??, ??

# Intellectual Property Statement

The PSI takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the PSI Chair.

The PSI invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the PSI Chair (see contacts information at PSI website).

# Copyright Notice

Copyright (C) Proteomics Standards Initiative (2017). All Rights Reserved.

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

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

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

# Glossary

Not used.

# References

[APWEILER1] Apweiler R., Bairoch A., Wu C.H., Barker W.C., Boeckmann B., Ferro S., Gasteiger E., Huang H., Lopez R., Magrane M., Martin M.J., Natale D.A., O'Donovan C., Redaschi N., Yeh L.S.  
UniProt: the Universal Protein knowledgebase. Nucleic Acids Res. 32:D115-119(2004).

[BRADNER1] Bradner, S. Key Words for Use in RFCs to Indicate Requirement Levels, RFC 2119. March 1997.

[IUPAC1999] IUPAC-IUBMB Joint Commission on Biochemical Nomenclature (JCBN) and Nomenclature Committee of IUBMB (NC-IUBMB). Eur. J. Of Biochemistry 264(2), 607-609 (1999); DOI: 10.1046/j.1432-1327.1999.news99.x

[PEARSON1] Pearson WR, Lipman DJ.Improved tools for biological sequence comparison. Proc Natl Acad Sci U S A. 1988 Apr;85(8):2444-8.

[THE\_UNIPROT\_CONSORTIUM1] The UniProt Consortium, The Universal Protein Resource (UniProt). Nucleic Acids Res. 35:D193-D197(2007)