PSM CBOR format documentation

26/10/2025 - 09/10/2026

Olivier Langella
PAPPSO
2026-2027
technical documentation

Table of content

1 Introduction 3
2 psm CBOR format 3
3 Root structure 5
4 Process entries 5
5 Protein description 5
6 Sample description 6
7 Scan description 6
7.1 Scan identifier description 6
7.2 Scan precursor description 6
7.3 Scan ms2 description 7
8 PSM description 7
8.1 Protein link description 7

1 Introduction

PSM CBOR file is designed to record any PSM (peptide spectrum match) in a compact binary file, easy to manipulate, versatile, extendable. This file is used as a stream in any condition, allowing the users to use unix pipes, compression algorithms, network transparency.

This way, from a DDA identification engine search result converted in PSM CBOR, any process can be added :

- Feature computations
- Prediction process (retention times, ion mobility, MS2 prediction...)
- Rescoring
- Filtering

2 psm CBOR format

It could be something like that :


```
{
  "informations": {
    "executable": "mcql",
    "type": "psm",
    "pappsomspp_version": "0.9.46",
    "sysinfo_machine_hostname": "proteus1",
    "sysinfo_product_name": "Debian GNU/Linux 12 (bookworm)",
    "timestamp": "2025-06-25T10:33:51"
  },
  "log": [],
  "parameter_map": {
    "sage": {},
    "xtandem": {},
    "specoms": {}
  },
  "target_fasta_files": ["zea_mays.fasta", "contaminant.fasta"],
  "decoy_fasta_files": ["rev_zea_mays.fasta", "rev_contaminant.fasta"],
  "protein_map": {
    "GRMZM2G083841_P01": {
      "description": "",
      "sequence": "",
      "target": true,
      "contaminant": false,
      "props": {},
      "eval": {}
    },
    "GRMZM2G083841_P02": {}
  },
  "sample_list": [
    {
      "name": "tandem2017_nopatch_20120906_balliau_extract_1_A01_urnb-1",
      "identification_file_list": [
        {
          "name": "/home/langella/data1/tandem/
```

```
tandem2017_nopatch_20120906_balliau_extract_1_A01_urnb-1.xml"
        }
      ],
      "peaklist_file": {
        "name": "tandem2017_nopatch_20120906_balliau_extract_1_A01_urnb-1.mzml"
      },
      "scan_list": [
        {
          "id": {
            "index": 1976
          },
          "precursor": {
            "z": 2,
            "mz": 1120.529471,
            "intensity": 500000,
            "alternatives": [
              {
                "z": 2,
                "mz": 1120.529471,
                "intensity": 500000
              }
            ]
          },
          "ms2": {
            "rt": 12648.87,
            "spectrum": {
              "mz": [1, 2, 3, 4],
              "intensity": [1, 2, 3, 4]
            }
          },
          "props": {},
          "psm_list": [
            {
              "proforma": "AQEEM[+15.99491]AQVAK",
              "protein_list": [
                {
                  "accession": "GRMZM2G083841_P01",
                  "positions": [15, 236]
                }
              ],
              "props": {
                "ion-series": {}
              },
              "eval": {
                "xtandem": {
                  "evalue": 0.0011
                },
                "specoms": {
                  "evalue": 0.0011
                }
              }
            }
```

```
          ]
        }
      ]
    }
  ]
}
```

3 Root structure

root sections are required and the order must be respected.

informations *(object)* *Required*

log *(array)* *Required* contains an array that logs all the previous "informations" sections. It helps to keep trace of PSM treatments.

parameter_map *(dictionary)* *Required* dictionary where each entry corresponds to a specific process (see Process entries). Each entry must contain the parameters used for this process.

target_fasta_files *(array)* *Required* List of file path to FASTA files (preferably absolute file path) used as reference target protein sequences for the identification engine.

decoy_fasta_files *(array)* *Optional* List of file path to FASTA files (preferably absolute file path) used as decoy protein sequences for the identification engine (if any and not generated on the fly).

protein_map *(dictionary)* *Required* contains an entry for each protein, using the accession as a unique identifier.

sample_list *(array)* *Required* an array of "sample" objects (see Sample description)

4 Process entries

If a CBOR PSM process is intended to give new results in "eval" sections : it must have a unique key to define it. This key must corresponds to an entry (see parameter_map in root).

Currently, several entries are already defined, but anyone can create a new one:

xtandem psm and protein evaluations computed by the X!Tandem search engine
sage psm and protein evaluations computed by the Sage search engine

5 Protein description

The protein object is stored in the protein_map in root . The protein accession is used as a dictionary key to reference each protein object. Protein accession is used in PSM description to link PSMs to proteins.

"protein" object is composed of (the order of the elements must be respected) :

description *(string)* *Required* Protein description, maybe empty

sequence *(string)* *Required* Protein amino acid sequence, maybe empty

target *(boolean)* *Required* Boolean (true by default), true if the protein belongs to the targeted sequences (target FASTA files). False otherwise.

contaminant *(boolean)* *Required* Boolean (false by default), true if the protein is tagged as a contaminant protein.

props *(object)* *Optional* Free structure designed to store any data (not related to a particular process) related to a protein as a propertie.

eval *(object)* *Required* Protein values computed by different algorithm or process (described in Process entries). Maybe empty.

6 Sample description

"sample" object is composed of :

name *(string)* *Required* Sample name

identification_file_list *(array)* *Optional* Identification engine search result files

peaklist_file *(object)* *Required*

scan_list *(array)* *Required* an array of "scan" objects (see Scan description)

7 Scan description

"scan" object is composed of :

id *(object)* *Required* Scan identifier object Scan identifier description

precursor *(object)* *Required* Precursor description (MS1 related data) Scan precursor description

ms2 *(object)* *Required* MS2 related data Scan ms2 description

props *(object)* *Optional* Free structure designed to store any data (not related to a particular process) related to a scan as a propertie.

psm_list *(array)* *Required* an array of "psm" objects (see PSM description)

7.1 Scan identifier description

index *(integer)* *Optional* Index of the scan in the original peak list file. Strongly recommended, but sometimes not available at all.

native_id *(string)* *Required* Unique identifier of the scan as defined in mzML file format.

scan *(integer)* *Optional* Scan number as written in the original peak list file.

7.2 Scan precursor description

mz *(float)* *Required* Scan precursor m/z

z *(int)* *Required* Scan precursor charge

intensity *(float)* *Optional* Scan precursor intensity

mass *(float)* *Optional* Scan precursor mass (molecule mass)

mh *(float)* *Optional* Scan precursor mass with 1 H+ (molecule mass + mhplus)

alternatives *(array)* *Optional* List of potential other precursors (not listed in the original mz files), called “alternatives”. each element is an object with “mz”, “z”, “intensity” of each potential precursor

7.3 Scan ms2 description

rt *(float)* *Optional* MS2 scan retention time

spectrum *(object)* *Optional* MS2 complete spectrum with “mz” and “intensity” arrays

8 PSM description

proforma *(string)* *Required* Peptide proforma notation

protein_list *(array)* *Required* Related proteins, described in Protein link description

props *(object)* *Optional* Peptide properties

eval *(dictionary)* *Required* Peptide values computed by different algorithm or process (described in Process entries).

8.1 Protein link description

accession *(string)* *Required* Protein accession : must be described in protein_map in root

positions *(array)* *Required* Positions of this peptide (proforma in psm) in the protein sequence. Maybe empty if the position is not known.

Important: a position starts at 0 (first amino acid) to N-1 (N is the protein length) for the last amino acid.