



Spoms documentation

PeptidOMS-ANR-24-CE45-3296

11/07/2025 - 23/10/2026

Olivier Langella
PAPPSO
2025-2026
technical documentation

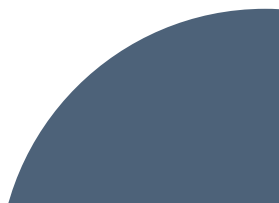


Table of content

1	Introduction	3
2	Source code	3
3	PSM CBOR file workflow	3
4	Command line interface	3
5	JSON parameter file	3
5.1	aa_table section <i>optional</i>	4
5.2	precursor section	5
5.3	spectrum section	5
5.4	pretreatment section	5
5.5	alignement section	6
5.6	post_treatment section	6
5.7	output section	6
5.8	alignment section	6
5.9	post_treatment section	6
6	CBOR peptidoms eval PSM structure	6
7	Tests for Spoms	7
7.1	Smaller dataset	7
	Bibliography	7

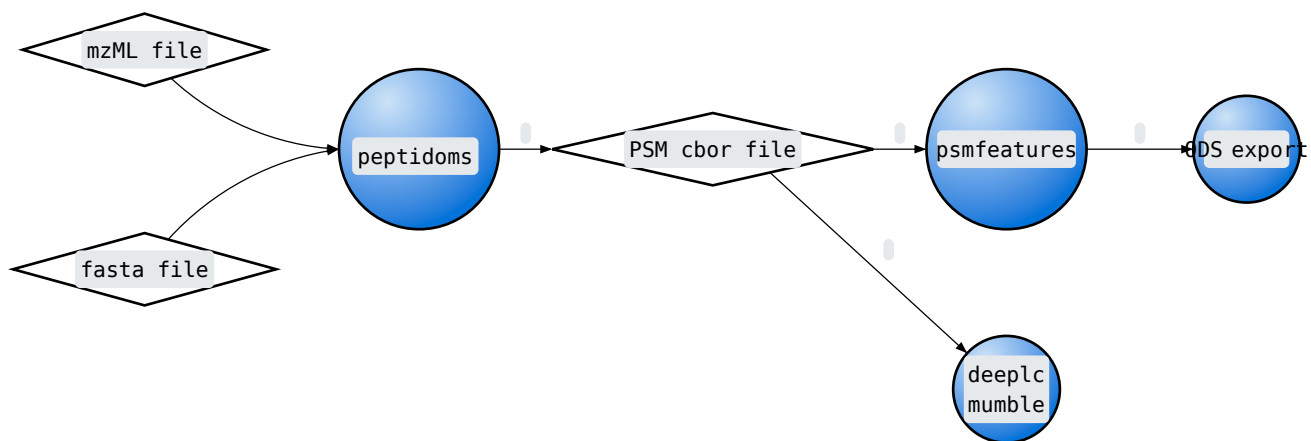
1 Introduction

Spoms aligns MS2 fragmentation spectra directly to protein sequences. Algorithm is described in (Benoist et al., 2025).

2 Source code

Current source code is available at <https://forge.inrae.fr/peptidoms/specpeptidoms>.

3 PSM CBOR file workflow



4 Command line interface

Spoms works on a collection of MS/MS spectra (mzML file typically) and a protein sequence database (FASTA file) and produces a PSM CBOR file as output.

```
spoms -p params.json -m mzdata.mzml -f protein_sequences.fasta -o results.cbor
```

To get a readable file results, use “spoms-export” to produce JSON or ODS files:

```
spoms-export -i results.cbor --ods results.ods
```

```
spoms-export -i results.cbor --json results.json
```

5 JSON parameter file

Spoms parameters are specified in a JSON file

JSON parameter file

```
{
  "aa_table": {
    "C": { "modifications": ["MOD:00397"] }
  },
  "precursor": {
    "precursor_process": true,
    "ms1_tolerance_unit": "ppm",
    "ms1_tolerance": 10,
    "minimum_complement_ion": 1,
    "intensity_ratio_threshold": 0.3,
    "potential_charge_states": [1, 2, 3, 4]
  },
  "spectrum": {
    "fragment_tolerance_unit": "dalton",
    "fragment_tolerance": 0.02,
    "deisotope": true,
    "minimum_mz": 150,
    "n_most_intense": 60
  },
  "alignment": {
    "nb_peaks_min": 30,
    "nb_min_aa_found": 20,
    "min_scenario_score": 10,
    "locations_score_threshold": 0.9,
    "nb_locations_saved": 1,
    "nb_returned_alignments": 2
  },
  "post_treatment": {
    "shut_down_peaks_cleaning": false,
    "shut_down_non_aligned_mass": false,
    "nb_interpretations_saved": 1
  },
  "output": {
    "max_interpretations_per_spectrum": 10
  }
}
```

5.1 aa_table section optional

This section allows to define your own amino acid table.

By default, spoms uses this table :

```
{
  "C": { "modifications": ["MOD:00397"]}
}
```

for the Carbamidomethyl fixed modification on Cystein.

This term is defined in the [PSI-MOD ontology](#) as:

```
[Term]
id: MOD:00397
name: iodoacetamide derivatized residue
def: "A protein modification that is produced by reaction with iodoacetamide, usually replacement of a reactive hydrogen with a methylcarboxamido group." [PubMed:11327326, PubMed:11510821, PubMed:12422359, Unimod:4]
subset: PSI-MOD-slim
synonym: "Carbamidomethyl" RELATED PSI-MS-label []
synonym: "Iodoacetamide derivative" RELATED Unimod-description []
```

It is possible to override this amino acid modification table by providing you own definition.

13C residue modifications can be found in the PSI-MOD ontology with accession MOD:00842:

is_a: MOD:00842 ! (13)C labeled residue

15N residue modifications can be found in the PSI-MOD ontology with accession MOD:00843:

is_a: MOD:00843 ! (15)N labeled residue

For a complete 13C + 15N list of labelled residue, you can use :

```
{
  "A": {"modifications": ["UNIMOD:1297"]},
  "R": {"modifications": ["MOD:00587"]},
  "N": {"modifications": ["UNIMOD:1266", "UNIMOD:995"]},
  "D": {"modifications": ["UNIMOD:1298"]},
  "C": {"modifications": ["UNIMOD:1297", "MOD:00397"]},
  "E": {"modifications": ["UNIMOD:268"]},
  "Q": {"modifications": ["UNIMOD:772", "UNIMOD:995"]},
  "G": {"modifications": ["UNIMOD:2088"]},
  "H": {"modifications": ["UNIMOD:188", "UNIMOD:996"]},
  "I": {"modifications": ["MOD:01286"]},
  "L": {"modifications": ["MOD:01285"]},
  "K": {"modifications": ["MOD:00582"]},
  "M": {"modifications": ["UNIMOD:772", "UNIMOD:994"]},
  "F": {"modifications": ["MOD:00589"]},
  "P": {"modifications": ["UNIMOD:772", "UNIMOD:994"]},
  "S": {"modifications": ["UNIMOD:1296", "UNIMOD:994"]},
  "T": {"modifications": ["UNIMOD:1298"]},
  "W": {"modifications": ["UNIMOD:188", "UNIMOD:772", "UNIMOD:995"]},
  "Y": {"modifications": ["UNIMOD:269"]},
  "V": {"modifications": ["MOD:00588"]},
}
```

5.2 precursor section

precursor_process **boolean** enable MS1 signal process to find alternative precursors. false by default, only works with mzML or mzCBOR peak list files

ms1_tolerance_unit **string** MS1 tolerance unit to use ("dalton" or "ppm", ppm by default)

ms1_tolerance **float** MS1 tolerance to look for peaks in MS1 signal (+/- this value)

minimum_complement_ion **integer** given a possible MS1 precursor mass : check the number of complement ion in the MS2 spectra.

intensity_ratio_threshold **float** Filter the MS1 signal for peaks of a minimum intensity compared to the official precursor intensity.

potential_charge_states **integer array** Look for potential charge states in the precursor isolation window (default is : [1,2,3]).

5.3 spectrum section

fragment_tolerance_unit **string** "dalton" by default

fragment_tolerance **float** mass delta precision (0.02 dalton by default)

deisotope **boolean** if true, process spectrum to merge isotopes into single charge peaks

minimum_mz **float** minimum m/z ratio to consider peaks in the spectrum

n_most_intense **int** consider only the N most intense peaks in the spectrum

5.4 pretreatment section

use_neutral_losses **boolean** if true, uses neutral losses peaks to score alignments

neutral_losses **dict** list of neutral losses to use and associated masses

neutral_losses_scores **dict** bonus scores associated to each neutral loss

5.5 alignment section

nb_peaks_min **int** Only spectra above that number of peaks are processed

nb_min_aa_found **int** Only spectra above that number of labels are processed

min_scenario_score **int** Only interpretations with a score over this threshold are returned to the user

locations_score_threshold **float** Excludes LssSM with a score below the best score multiplied by this factor when several are returned per spectrum by the preliminary alignment

nb_locations_saved **int** Number of Located subsequence Spectrum Matches (LssSM) per spectrum saved in memory at a time

nb_returned_alignments **int** Number of spectra alignments returned to the user

score/undefined_nter_penalty **int** if an unidentified mass remains in nter, apply this penalty to the alignment

score/undefined_cter_penalty **int** if an unidentified mass remains in cter, apply this penalty to the alignment

score/intensity_bonus_thresh **float** between 0 and 1. Above this proportion of the maximum peak intensity of the spectra, the peaks receive a bonus defined by **intensity_bonus**

score/intensity_bonus **int** bonus applied in the case described above

score/intensity_penalty_thresh **float** between 0 and 1. Below this proportion of the maximum peak intensity of the spectra, the peaks receive a penalty defined by **intensity_penalty**

score/intensity_penalty **int** penalty applied in the case described above

5.6 post_treatment section

shut_down_peaks_cleaning **boolean** Option (true/false) to correct specific alignments in the second alignment round

shut_down_non_aligned_mass **boolean**

nb_interpretations_saved **int** Number of backup interpretations per spectrum saved in memory at a time

5.7 output section

max_interpretations_per_spectrum **int** Number of interpretations per spectrum returned to the user

5.8 alignment section

5.9 post_treatment section

6 CBOR peptidoms eval PSM structure

Spoms writes results in a PSM CBOR file. For each PSM, Spoms uses the “eval” section to store its scores :

spoms PSM eval structure

```
{
  "proforma": "[-1.01]?MISPSNVAVDIGR",
  "protein_list": [{ "accession": "GRMZM2G105644_P01", "positions": [110] }],
  "eval": {
    "spoms": {
      "isotope": "0",
      "precursor": {
        "mz": 679.8544921875,
        "z": 2,
        "intensity": 21616796
      },
      "first": {
        "bracket": "[243.11]SPSNVAVD[I][G]R",
        "spc": 11,
        "score": 63,
        "nam": -0.001982735625688292
      },
      "post": {
        "bracket": "[M][I]SPSNVAVD[I][G]R",
        "spc": 13,
        "score": 66,
        "nam": -1.0136910826447547
      }
    }
  }
}
```

7 Tests for Spoms

```
./src/spoms -p ../doc/typst/json/spoms_parameters.json -f /gorgone/pappso/moulon/database/Genome_Z_mays_5a.fasta -m /gorgone/pappso/data_extraction_pappso/mzML/20120906_balliau_extract_1_A01_urnb-1.mzML -o /tmp/test.cbor -c 10
```

```
./src/spoms -p ../doc/typst/json/spoms_parameters.json -f /gorgone/pappso/moulon/database/Genome_Z_mays_5a.fasta -m /gorgone/pappso/data_extraction_pappso/mzcbor/20120906_balliau_extract_1_A01_urnb-1.mzcbor -o /tmp/test.cbor -c 10
```

Process for spectrum index 2406 finished. Slowest one is 2287 2290
total 236

7.1 Smaller dataset

```
./src/spoms -p ../doc/typst/json/spoms_parameters_test.json -f ../tests/data/fasta/small_zea_mays.fasta -m ../tests/data/peaklists/peaklist_15046.mgf -o /tmp/test.cbor
```

```
./src/spoms -p ../tests/data/parameters/DATNVGDEGGFAPNIIENK_parameters.json -f ../tests/data/fasta/databankThreeProteinsHuman.fasta -m ../tests/data/peaklists/DATNVGDEGGFAPNIIENK.mgf -o DATNVGDEGGFAPNIIENK.cbor -c 10
```

Bibliography

Benoist, É., Jean, G., Rogniaux, H., Fertin, G., and Tessier, D. (2025). SpecPeptidOMS Directly and Rapidly Aligns Mass Spectra on Whole Proteomes and Identifies Peptides That Are Not Necessarily Tryptic: Implications for Peptidomics. *Journal of Proteome Research* 24, 2159–2172. doi: [10.1021/acs.jproteome.4c00870](https://doi.org/10.1021/acs.jproteome.4c00870)