Technical note

psm_utils: A high-level Python API for parsing and handling peptide-spectrum matches and proteomics search results

Ralf Gabriels[1,2], Arthur Declercq[1,2], Robbin Bouwmeester[1,2], Sven Degroeve[1,2], and Lennart Martens[1,2,*]

[1] VIB-UGent Center for Medical Biotechnology, VIB, Belgium
[2] Department of Biomolecular Medicine, Ghent University, Belgium

[*] To whom correspondence should be addressed:
Tel: +32 9 264 93 58
Email: lennart.martens@vib-ugent.be
Address: Technologiepark 75, 9052 Ghent, Belgium

ORCID IDs:
Ralf Gabriels: https://orcid.org/0000-0002-1679-1711/
Arthur Declercq: https://orcid.org/0000-0002-9376-1399/
Robbin Bouwmeester: https://orcid.org/0000-0001-6807-7029/
Sven Degroeve: https://orcid.org/0000-0001-8349-3370/
Lennart Martens: https://orcid.org/0000-0003-4277-658X/

Abstract

A plethora of proteomics search engine output file formats are in circulation. This lack of standardized output files greatly complicates generic downstream processing of peptide-spectrum matches (PSMs) and PSM files. While standards exist to solve this problem, these are far from universally supported by search engines. Moreover, software libraries are available to read a selection of PSM file formats, but a package to parse PSM files into a unified data structure has been missing. Here, we present psm_utils, a Python package to read and write various PSM file formats and to handle peptidoforms, PSMs, and PSM lists in a unified and user-friendly Python-, command line-, and web-interface. psm_utils was developed with pragmatism and maintainability in mind, adhering to community standards and relying on existing packages where possible. The Python API and command line interface greatly facilitate handling various PSM file formats. Moreover, a user-friendly web application was built using psm_utils that allows anyone to interconvert PSM files and retrieve basic PSM statistics. psm_utils is freely available under the permissive Apache2 license at https://github.com/compomics/psm_utils.

Keywords

Proteomics; bioinformatics; data analysis; peptide identification; peptide-spectrum matches

Introduction

Peptide identification from MS/MS spectra is a key step in bottom-up mass spectrometry-based proteomics. Since the mid-1990's, a plethora of specialized software, called proteomics search engines, have been developed to automate peptide identification. These search engines generally take two inputs: a spectrum file originating from the mass spectrometer, potentially preprocessed, and a FASTA protein sequence file. Then the output in its simplest form is a list of identified peptides with peptide-spectrum match (PSM) information.[1] This PSM list is then usually passed on to other bioinformatics tools for further downstream analysis step(s), such as identification rescoring, protein inference, or protein quantification.

For optimal reproducibility and maintainability, each of these steps should be performed by a dedicated interchangeable software module, which requires standardization of the file formats that link each workflow step.[2–4] Such a comprehensive standardization effort is being undertaken by the HUPO Proteomics Standards Initiative, most notably with the development of the mzML and mzIdentML formats.[5,6] However, most search engines only support their own output file format that typically does not adhere to community standards. Consequentially, building a module for downstream use of PSM files from various search engines can be cumbersome at best, and infeasible at worst. This hurdle often results in the *ad hoc* writing of hard-to-maintain parsing scripts that are only intended for single use. While a few open-source Python libraries exist to read and/or write various PSM file formats, such as Pyteomics, psims, and pyOpenMS, a Python library that parses PSM files into a unified high-level data structure for consistent and easy handling is missing. [7–9]

Here we therefore present psm_utils (https://github.com/compomics/psm_utils), an easy-to-use Python library which reads and writes various PSM file formats, but which also handles PSMs in a unified data model and API. We also used our psm_utils library to develop both a command line interface (CLI) and a web interface to easily interconvert PSM files and to retrieve basic PSM statistics, for scripting and end-users, respectively.

Python library

psm_utils was developed with pragmatism and maintainability in mind. Instead of duplicating existing work, psm_utils relies on the existing Pyteomics and psims Python packages where possible. Furthermore, psm_utils follows HUPO-PSI community standards where applicable, and is built to be open and dynamic, allowing easy updates and extensions. As a permissively licensed open-source software, psm_utils welcomes contributions from the community.

The project is split into the main `psm_utils` package and the `psm_utils.io` subpackage. The former provides the main API for peptidoforms, PSMs, and PSM lists; the latter provides modules for reading and writing various PSM file formats (Figure 1). More specifically, in the main `psm_utils` package, PSM information is represented by three distinct classes. (1) The `Peptidoform` class accepts a combination of peptide sequence, residue modifications, and optionally charge state, represented in the HUPO-PSI ProForma 2.0 notation.[10] Through the `proforma` and `mass` modules of Pyteomics, this permits a direct implementation of useful methods, such as the calculation of theoretical mass or fragmentation spectrum, while considering any resolvable residue modification. (2) The `PeptideSpectrumMatch` class connects a `Peptidoform` instance to a spectrum defined by a collection, run, and spectrum identifier – analogous to the keys of HUPO-PSI Universal Spectrum Identifier (USI)[11] - and holds all relevant match information and PSM metadata. (3) The `PSMList` class represents a collection of PSMs as a simple Python list with additional functionality. Any PSM file can therefore be parsed into a `PSMList` with `PeptideSpectrumMatch` instances, each in turn holding a `Peptidoform` instance alongside relevant PSM information. We opted for a simple Pythonic list-like object to allow for flexible downstream implementations of psm_utils. Both the `PeptideSpectrumMatch` and `PSMList` classes are based on the Pydantic data class model for efficient data validation and type coercion (https://github.com/pydantic/pydantic). For effortless use of psm_utils in data analysis or machine learning contexts, `PSMList` instances can be transformed into the commonly used tabular Pandas `DataFrame` object. The fully documented Python API and a quickstart guide can be found on https://psm-utils.readthedocs.io/.

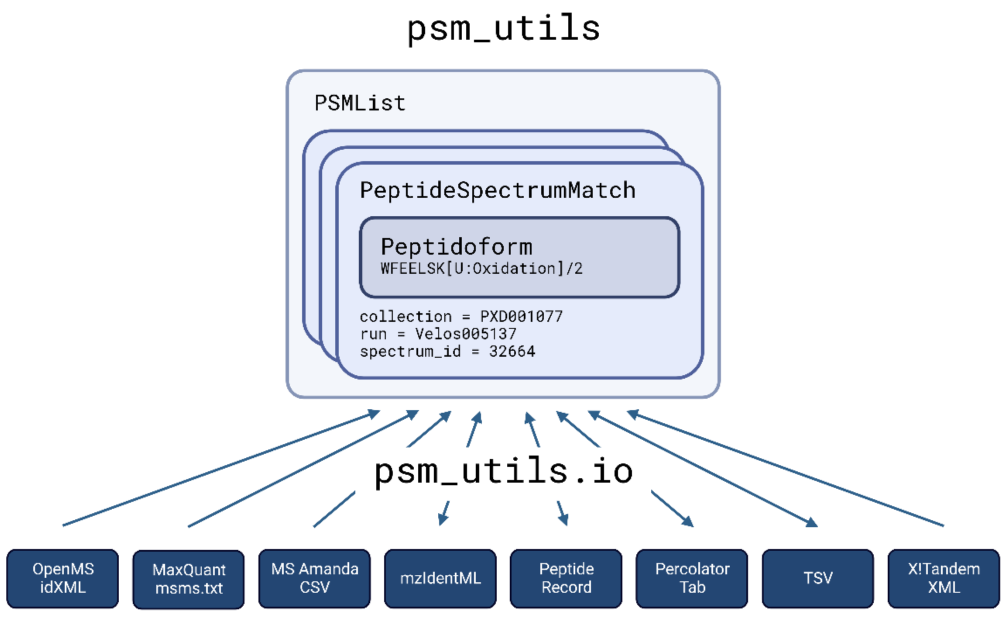


Figure 1. Overview of the psm_utils structure. Various file formats can be read to, or written from, a PSMList object, which holds multiple PeptideSpectrumMatch objects, which in turn each hold a Peptidoform object along with the relevant metadata. Created with BioRender.com.

The `psm_utils.io` subpackage contains a separate module for each of the (currently) eight supported PSM file formats (Table 1). Every module implements a reader and/or writer class to convert the specific PSM file format to, or from, the unified `PSMList` object.

Importantly, these readers and writers also interconvert the various proprietary peptidoform notations from or to ProForma 2.0, which can otherwise be a cumbersome process to perform *ad hoc*. Consistency between various readers and writers is achieved by inheriting from an abstract base class, providing a blueprint for future implementations of new PSM file formats. Generic `read_file` and `write_file` functions could therefore be implemented, where the file type can be specified or set to be inferred from the filename. Similarly, a high-level `convert` function was implemented for quick interconversion between PSM file formats. Due to this blueprint and the hierarchical architecture of psm_utils, support for more file types can easily be added in the future without requiring changes to the central API.

Table 1. Supported file formats for reading and writing peptide-spectrum match lists

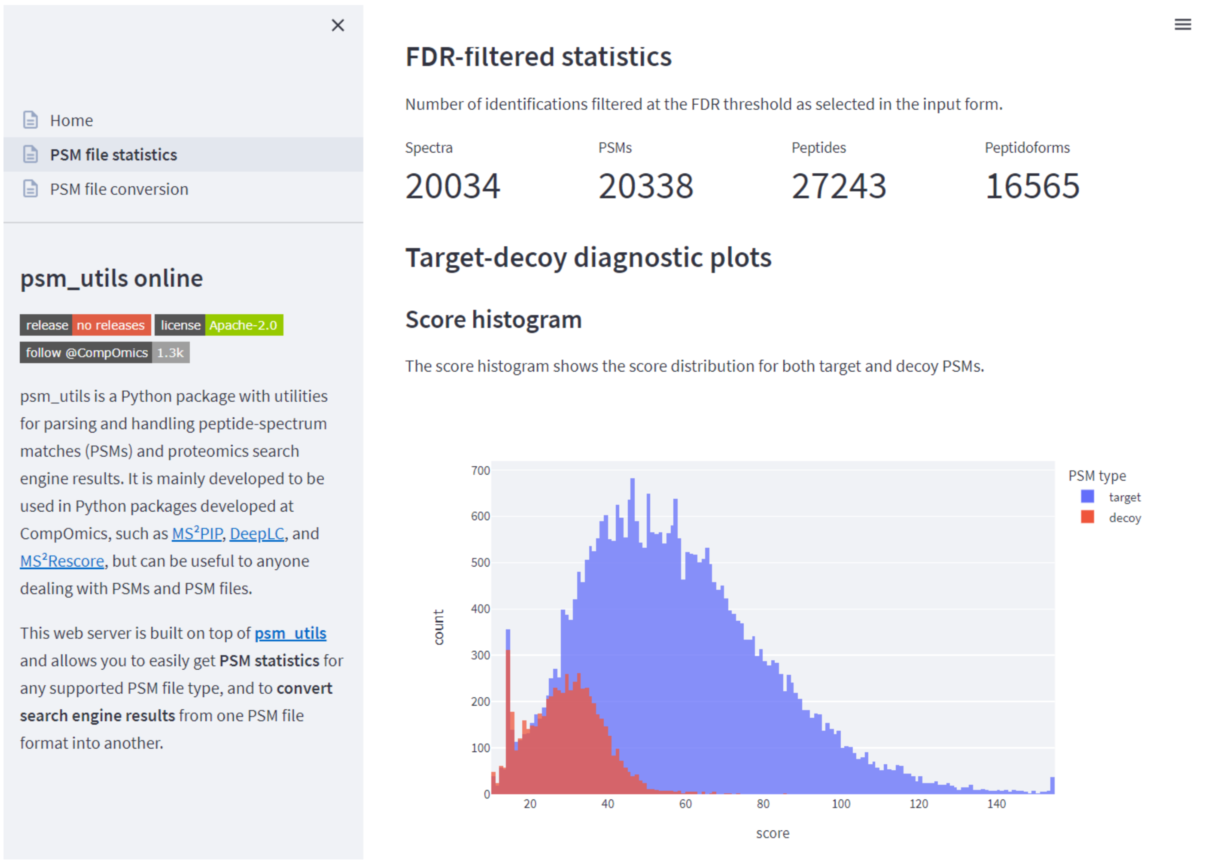
File format	Read support	Write support
OpenMS idXML	✅	❌
MaxQuant msms.txt	✅	❌
MS Amanda CSV	✅	❌
mzIdentML	✅	✅
PeptideRecord	✅	✅
Percolator Tab	✅	✅
TSV	✅	✅
X!Tandem XML	✅	❌

Command line interface

The `psm_utils.io.convert` function is also accessible through a CLI. This facilitates the implementation of psm_utils within proteomics data analysis pipelines where two sequential steps would otherwise be incompatible due to different PSM file types being used. While currently only file conversion is implemented, the use of subcommands allows for more functionality to be added to the psm_utils CLI in the future.

Web application

The psm_utils functionality can also be accessed through a Streamlit (https://streamlit.io) web application at https://psm-utils.streamlit.app (Figure 2). This allows any researcher to interconvert between any supported file formats, regardless of programming skills. Next to the interconversion of PSM file formats, PSM files can be uploaded to retrieve basic PSM statistics, such as the total number of identified spectra at a preset false discovery rate (FDR) threshold. Moreover, several diagnostic target-decoy plots are automatically generated, allowing users to easily assess the quality of the FDR estimation.



Home

PSM file statistics

PSM file conversion

psm_utils online

release no releases | license Apache-2.0 | follow @CompOmics 1.3k

psm_utils is a Python package with utilities for parsing and handling peptide-spectrum matches (PSMs) and proteomics search engine results. It is mainly developed to be used in Python packages developed at CompOmics, such as MS²PIP, DeepLC, and MS²Rescore, but can be useful to anyone dealing with PSMs and PSM files.

This web server is built on top of **psm_utils** and allows you to easily get **PSM statistics** for any supported PSM file type, and to **convert search engine results** from one PSM file format into another.

FDR-filtered statistics

Number of identifications filtered at the FDR threshold as selected in the input form.

Spectra	PSMs	Peptides	Peptidoforms
20034	20338	27243	16565

Target-decoy diagnostic plots

Score histogram

The score histogram shows the score distribution for both target and decoy PSMs.

Figure 2. Screenshot of the psm_utils online web application. Users can upload a PSM file for interconversion between supported file types, or retrieve PSM-related statistics and diagnostic target-decoy plots.

Conclusion

We here presented psm_utils, a Python package that greatly simplifies downstream usage of peptide identifications, regardless of PSM file format. The Python package is convenient to use in any data analysis tool that handles PSMs, and the command line interface can easily be embedded in automated workflows. The web application brings the ability to interconvert or quickly inspect any supported PSM file type to any researcher in the field. psm_utils is set up as an open and dynamic project and we welcome everyone in the computational proteomics community to make use of, and contribute to, the project.

Availability

The psm_utils Python package is available on PyPI (https://pypi.org/project/psm-utils) and Bioconda (https://anaconda.org/bioconda/psm-utils). The source code is available on GitHub (https://github.com/compomics/psm_utils) under the permissive Apache2 license. The psm_utils online web application is available on Streamlit (https://psm-utils.streamlit.app).

Acknowledgment

We thank the reviewers for their detailed and helpful suggestions to further improve the psm_utils library.

Funding

R.G. and A.D. acknowledge funding from the Research Foundation Flanders (FWO) [1SE3722; 12B7123N]. R.B. acknowledges funding from the Vlaams Agentschap Innoveren en Ondernemen [HBC.2020.2205]. S.D. and L.M. acknowledge funding from the European Union's Horizon 2020 Programme (H2020-INFRAIA-2018-1) [823839]; L.M. acknowledges funding from the Research Foundation Flanders (FWO) [G028821N] and from Ghent University Concerted Research Action [BOF21/GOA/033].

Author contributions

Ralf Gabriels: Conceptualization, Software, Writing - Original Draft. **Arthur Declercq:** Software, Writing - Review & Editing. **Robbin Bouwmeester:** Software, Writing – Review & Editing. **Sven Degroeve:** Writing - Review & Editing, Supervision. **Lennart Martens:** Writing - Review & Editing, Supervision.

References

(1) Verheggen, K.; Raeder, H.; Berven, F. S.; Martens, L.; Barsnes, H.; Vaudel, M. Anatomy and Evolution of Database Search Engines-a Central Component of Mass Spectrometry Based Proteomic Workflows. *Mass Spectrom Rev* **2017**. https://doi.org/10.1002/mas.21543.

(2) Röst, H. L.; Sachsenberg, T.; Aiche, S.; Bielow, C.; Weisser, H.; Aicheler, F.; Andreotti, S.; Ehrlich, H. C.; Gutenbrunner, P.; Kenar, E.; Liang, X.; Nahnsen, S.; Nilse, L.; Pfeuffer, J.; Rosenberger, G.; Rurik, M.; Schmitt, U.; Veit, J.; Walzer, M.; Wojnar, D.; Wolski, W. E.; Schilling, O.; Choudhary, J. S.; Malmström, L.; Aebersold, R.; Reinert, K.; Kohlbacher, O. OpenMS: A Flexible Open-Source Software Platform for Mass Spectrometry Data Analysis. *Nat Methods* **2016**, *13* (9). https://doi.org/10.1038/nmeth.3959.

(3) Perez-Riverol, Y.; Moreno, P. Scalable Data Analysis in Proteomics and Metabolomics Using BioContainers and Workflows Engines. *Proteomics* **2020**, *20* (9). https://doi.org/10.1002/pmic.201900147.

(4) Chambers, M. C.; MacLean, B.; Burke, R.; Amodei, D.; Ruderman, D. L.; Neumann, S.; Gatto, L.; Fischer, B.; Pratt, B.; Egertson, J.; Hoff, K.; Kessner, D.; Tasman, N.; Shulman, N.; Frewen, B.; Baker, T. A.; Brusniak, M. Y.; Paulse, C.; Creasy, D.; Flashner, L.; Kani, K.; Moulding, C.; Seymour, S. L.; Nuwaysir, L. M.; Lefebvre, B.; Kuhlmann, F.; Roark, J.; Rainer, P.; Detlev, S.; Hemenway, T.; Huhmer, A.; Langridge, J.; Connolly, B.; Chadick, T.; Holly, K.; Eckels, J.; Deutsch, E. W.; Moritz, R. L.; Katz, J. E.; Agus, D. B.; MacCoss, M.; Tabb, D. L.; Mallick, P. A Cross-Platform Toolkit for Mass Spectrometry and

Proteomics. *Nature Biotechnology*. October 2012, pp 918–920. https://doi.org/10.1038/nbt.2377.

(5) Martens, L.; Chambers, M.; Sturm, M.; Kessner, D.; Levander, F.; Shofstahl, J.; Tang, W. H.; Römpp, A.; Neumann, S.; Pizarro, A. D.; Montecchi-Palazzi, L.; Tasman, N.; Coleman, M.; Reisinger, F.; Souda, P.; Hermjakob, H.; Binz, P. A.; Deutsch, E. W. MzML - A Community Standard for Mass Spectrometry Data. *Molecular and Cellular Proteomics* **2011**, *10* (1). https://doi.org/10.1074/mcp.R110.000133.

(6) Vizcaíno, J. A.; Mayer, G.; Perkins, S.; Barsnes, H.; Vaudel, M.; Perez-Riverol, Y.; Ternent, T.; Uszkoreit, J.; Eisenacher, M.; Fischer, L.; Rappsilber, J.; Netza, E.; Walzer, M.; Kohlbacher, O.; Leitner, A.; Chalkley, R. J.; Ghali, F.; Martínez-Bartolome, S.; Deutsch, E. W.; Jones, A. R. The MzIdentML Data Standard Version 1.2, Supporting Advances in Proteome Informatics. *Molecular and Cellular Proteomics* **2017**, *16* (7). https://doi.org/10.1074/mcp.M117.068429.

(7) Levitsky, L. I.; Klein, J. A.; Ivanov, M. v.; Gorshkov, M. v. Pyteomics 4.0: Five Years of Development of a Python Proteomics Framework. *J Proteome Res* **2019**, *18* (2), 709–714. https://doi.org/10.1021/acs.jproteome.8b00717.

(8) Klein, J.; Zaia, J. Psims - A Declarative Writer for MzML and MzIdentML for Python. *Molecular and Cellular Proteomics* **2019**, *18* (3). https://doi.org/10.1074/mcp.RP118.001070.

(9) Röst, H. L.; Schmitt, U.; Aebersold, R.; Malmström, L. PyOpenMS: A Python-Based Interface to the OpenMS Mass-Spectrometry Algorithm Library. *Proteomics* **2014**, *14* (1). https://doi.org/10.1002/pmic.201300246.

(10) LeDuc, R. D.; Deutsch, E. W.; Binz, P.-A.; Fellers, R. T.; Cesnik, A. J.; Klein, J. A.; van den Bossche, T.; Gabriels, R.; Yalavarthi, A.; Perez-Riverol, Y.; Carver, J.; Bittremieux, W.; Kawano, S.; Pullman, B.; Bandeira, N.; Kelleher, N. L.; Thomas, P. M.; Vizcaíno, J. A. Proteomics Standards Initiatives ProForma 2.0 Unifying the Encoding of Proteoforms and Peptidoforms. *J Proteome Res* **2021**, *21*, 1189–1195. https://doi.org/https://doi.org/10.1021/acs.jproteome.1c00771.

(11) Deutsch, E. W.; Perez-Riverol, Y.; Carver, J.; Kawano, S.; Mendoza, L.; van den Bossche, T.; Gabriels, R.; Binz, P.-A.; Pullman, B.; Sun, Z.; Shofstahl, J.; Bittremieux, W.; Mak, T. D.; Klein, J.; Zhu, Y.; Lam, H.; Vizcaíno, J. A.; Bandeira, N. Universal Spectrum Identifier for Mass Spectra. *Nat Methods* **2021**, *18* (7). https://doi.org/10.1038/s41592-021-01184-6.