# Free Open Source Software for Protein and Peptide Mass Spectrometry-based Science

Filippo Rusconi

# Free Open Source Software for Protein and Peptide Mass Spectrometry-based Science

Filippo Rusconi[1]

PAPPSO, Université Paris-Saclay, INRAE, CNRS, AgroParisTech, GQE - Le Moulon, 91190, Gif-sur-Yvette, France – filippo.rusconi@universite-paris-saclay.fr

**Abstract**

In the field of biology, and specifically in protein and peptide science, the power of mass spectrometry is that it is applicable to a vast spectrum of applications. Mass spectrometry can be applied to identify proteins and peptides in complex mixtures, to identify and locate post-translational modifications, to characterize the structure of proteins and peptides to the most detailed level or to detect protein–ligand non covalent interactions. Thanks to the Free and Open Source Software (FOSS) movement, scientists have limitless opportunities to deepen their skills in software development to code software that solves mass spectrometric data analysis problems. After conversion of raw data files to open standard format files, the entire spectrum of data analysis tasks can now be performed integrally on FOSS platforms, like GNU/Linux, and only with FOSS solutions. This review presents a brief history of mass spectrometry open file formats and goes on with the description of FOSS projects that are commonly used in protein and peptide mass spectrometry fields of endeavor: identification projects that involve mostly automated pipelines, like proteomics and peptidomics, and bio-structural characterization projects that most often involve manual scrutiny of the mass data. Projects of the last kind usually involve software that allows the user to delve into the mass data in an interactive graphics-oriented manner. Software projects were thus categorized on the basis of these criteria: software libraries for software developers vs desktop-based graphical user interface software for the end user and automated pipeline-based data processing vs interactive graphics-based mass data scrutiny.

**Keywords:** Free Software, Open Source, Mass Spectrometry, Proteins, Peptides, Structural Biology.

## 1. Introduction

Mass spectrometry for protein and peptide science generates vast amounts of data that need to be mined to extract reasonably meaningful information from them. Software for mass spectrometry has been produced since the beginning of the popularization of mass spectrometry in research laboratories, in the early nineties. More than a decade later, the Free and Open Source Software (FOSS) movement, in the scientific computing landscape, has elicited a surge in the production of software for mass spectrometry. The availability of the GNU/Linux FOSS platform has allowed any scientist with computer science skills to code software to solve data analysis problems. That general availability of FOSS platforms, although a remarkable opportunity, produced some undesirable side effects as detailed below.

In a very interesting paper published in 2018, Smith [1] investigated the perception that either software users or software developers had on the software offerings in the field of mass spectrometry applied to

1 Corresponding author: PAPPSO, Université Paris-Saclay, INRAE, CNRS, AgroParisTech, GQE - Le Moulon, 91190, Gif-sur-Yvette, France – filippo.rusconi@universite-paris-saclay.fr

biology. One of the main concerns raised by the interviewees is the lack of software support for users that are not developers. One other concern was the idea that many software projects were started to answer a specific research question, were then published and finally abandoned. As a software developer in the field of bio-structural mass spectrometry since the mid-nineties, the author of this review acknowledges that there is indeed a pervasive feeling of frustration among software users, related to the fact that the software development process too often looks like a “one-shot” software production process, without having in mind that any software product needs to be maintained over long time ranges to fix bugs, improve/add features and write detailed documentation. Lack of properly redacted and formatted documentation adds to the grief about unmaintained software. In this review, only software that is consistently developed and maintained, without disregarding its documentation, is reported.

The FOSS definition entails the fundamental ability to get the source code of the program without having to perform any explicit request for it in any way (no compulsory user registration or emailed request, for example). Software projects that describe their production as FOSS but that do not provide a link to a download site allowing anyone to download the source code without a prior request are not included in this review.

The “protein and peptide mass spectrometry” expression is a most generic way of describing a field of endeavor that actually encompasses a number of specialties. Not all these specialties harness the same set of software tools. For example, proteomics or peptidomics scientists use software programs that almost never display full mass spectra for human eye inspection. Their use of mass spectrum visualization software is almost always limited to human-based quality control assessments. Conversely, structural biologists working on biopolymers, like proteins and peptides, rely on software programs that allow deep inspection of mass spectrometry data sets because they are typically interested in very detailed structural information (like protein oxidation, rare/low-abundance/ undocumented protein chemical or biological post-translational modifications). In this review, the proteomics/peptidomics software offerings are going to be presented very concisely. On the contrary, the software programs for the structural biology mass spectrometrists are considered the focus of this review.

Using FOSS to mine and/or visualize mass spectrometric data requires that the mass spectrometric data be available in a non-proprietary format. The vast majority of the instrument vendors do pack the mass spectrometric data acquired on their instruments in binary files of undocumented internal format. One most notable exception, that must be underlined, is the Bruker company, that has a different stance in this regard: they provide FOSS developers with detailed information about their data file format so as to allow them to develop software that can natively use their data files. In other situations, the FOSS user will need to first convert mass spectrometric data files from the proprietary vendor format to an open data format. This review will open with a description of the various open data formats for mass spectrometry and of one software project that provides a powerful means to convert data files from almost any proprietary format to one of these open data formats.

In the field of mass spectrometry, FOSS projects can be categorized in two distinct types: the “Software for software” category, comprising mostly software libraries used by developers to craft end-user desktop programs and the “End-user software” category, comprising the end-user interactive programs,

mostly desktop-based. In this review, we will systematically make the difference between these two categories because their audiences are eminently different, from the use cases stand point.

Table 1 lists the most representative software projects in protein/peptide mass spectrometry that produce either widely used libraries, providing developers with a large set of functionalities that are meant to be strung together so as to craft sophisticated mass data analysis pipelines or desktop programs for the end-user. **Figure 1** is a flow chart illustrating the steps that are typically followed to identify the proteins in a sample, in bottom-up or top-down proteomics.

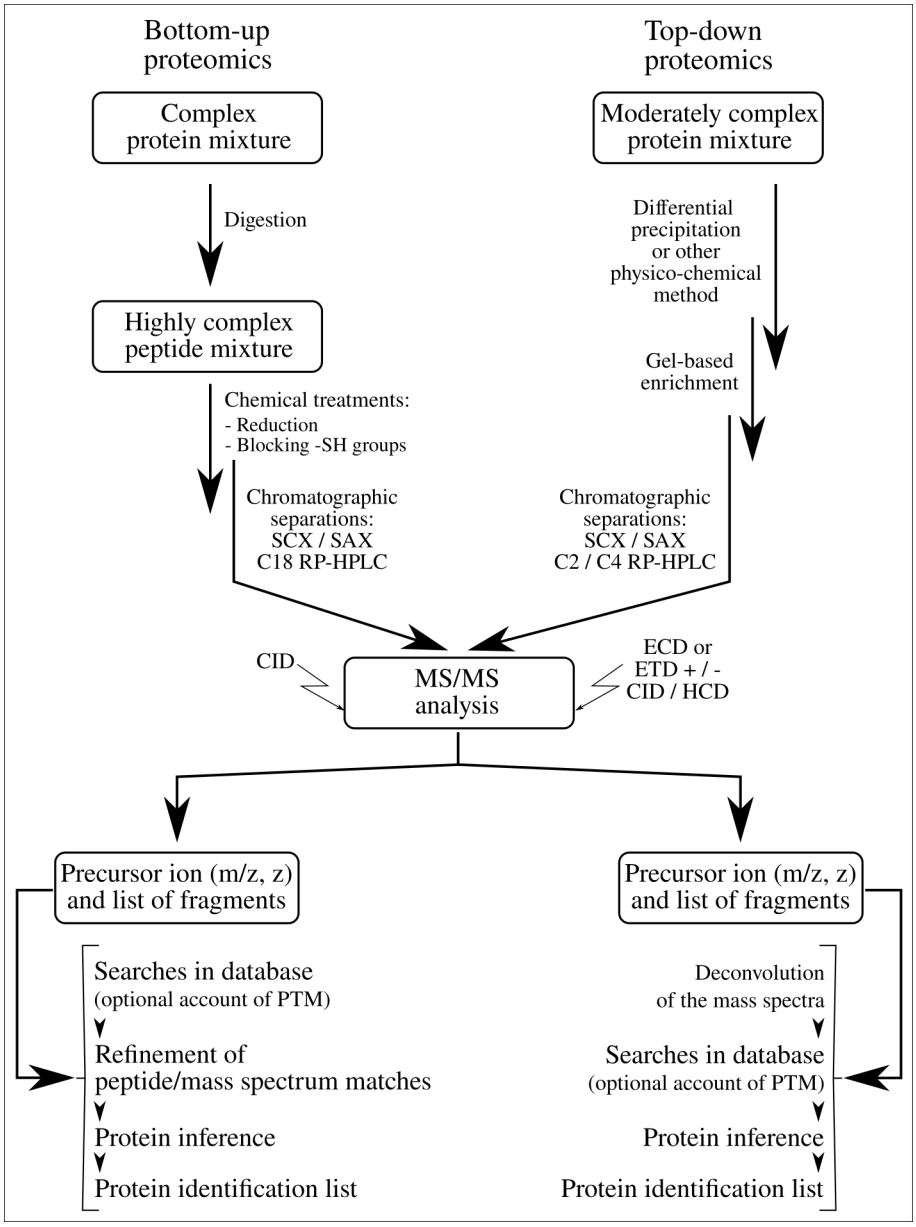


Figure 1: Typical steps followed in bottom-up and top-down proteomics. The general flow is similar in both cases, with some notable differences related to the sample preparation, the ion fragmentation techniques involved and the mass data processing before the searches in the database. SCX: strong cation exchange; SAX: strong anion exchange; RP-HPLC: reversed-phase high performance liquid chromatography; CID: collisionally-activated dissociation; ECD: electron-capture dissociation; ETD: electron-transfer dissociation; HCD: high energy collisional decomposition; PTM: post-translational modification.

## 2. Brief history of the mass spectrometric data formats

At the very beginning of mass spectrometry, if software was involved in an experiment, it was mainly used to store the data in very simple text-based files reporting, in each row, the m/z ratio of a detected ion along with its corresponding signal intensity. At the time, therefore, mass data analysis merely involved human scrutiny of the spectra. Because mass spectrometry evolved with new features and increased sophistication, the kind of data that had to be stored in mass data files started to require much more elaborated file formats in order to accommodate an increasing amount of metadata that were needed to fully characterize workflows and processing strategies, like data scoring, sorting or quantification, for example. Because of the huge endeavors in the advancement of mass spectrometry-based proteomics, that specific field of mass spectrometry research has been responsible for the development of most of the mass data file formats currently available.

### 2.1. Most ancient pre-proteomics formats

One of the earliest attempts to define a specific file format for mass spectrometry dates back to the beginning of the nineties with the JCAMP-DX format, building on top of a preexisting format specific for infrared spectroscopy research that had been designed by the Joint Committee on Atomic and Molecular Physical Data [2]. At the time, proteomics had not yet caught up and the design of the format did not integrate the notion of multidimensional mass spectra, specifically required to document MS/MS experiments. The *JCAMP-DX* format would eventually be abandoned around 1991 in favor of a new format–*andi/MS*–based on the *netCDF* format [3], which is in the public domain.

Because MS/MS data could not be stored in *andi/MS*-formatted files and because *netCDF* was primarily used by the spectroscopy community, the *andi/MS* format did not catch in the proteomics community. If we had to single out one of the major instrument improvements that triggered, in the early proteomics era, the design of new data file formats for mass spectrometry, that would be tandem MS. Indeed, while full scan spectra could be stored as simple (m/z, count) pairs using rudimentary text file formats, MS/MS (or $MS^n$) mass data required more metadata to be stored along with the mass peak data. Even if the new formats were not highly complex, they nonetheless had to accommodate enough new data bits to document the precursor ion's m/z ratio and charge before listing the (m/z, count) pairs obtained during the gas phase fragmentation step. Examples of such simple text file formats are:

- *dta*, that originated in the SEQUEST MS/MS-based search engine;
- *pkl*, that originated in the Micromass (now Waters) data analysis software package;
- *pks*, that originated in the Perseptive (now ThermoFisher) data analysis software package. This format specialized in post-source decay sequence analysis;
- *mgf*, (for "Mascot generic format") that originated in the Mascot search engine.

All of these files roughly have the same features: one line contains data about the precursor ion (m/z , intensity, ) and the MS/MS spectrum follows in the form of (m/z, count) pairs. One drawback is the lack of expressiveness in these kinds of formats. The advent of the extensible markup language (XML) made it possible to easily define flexible "grammars" for establishing new data formats. XML thus became the lingua franca of the open mass data formats.

## 2.2. XML-based formats

The extensible markup language (XML) is an application of the more general SGML[2] International Organization for Standardization (ISO) specification. It was elaborated starting in the middle of the nineties and quickly began a successful career in the field of open file format specification[3] (office productivity suites like LibreOffice store data in XML-based files). Below is one small excerpt from an XML-based file definition format, from the *massXpert* software package [4]. This code snippet specifies two polymer chemistry modifications:

```
<mdf>
<name>Oxidation</name>
<formula>O</formula>
<targets>M;Y;</targets>
<maxcount>1</maxcount>
</mdf>
<mdf>
<name>Phosphorylation</name>
<formula>HO3P</formula>
<targets>S;T;Y;</targets>
<maxcount>1</maxcount>
</mdf>
```

### *2.2.1. mzXML*

One of the first two attempts at using XML to define a new mass spectrometry data format was by Pedrioli *et al.* [5] at the Institute for Systems Biology (Seattle, Washington, USA). This file format, named *mzXML*, was published as the result of an extensive collaboration between no less than twelve international scientific institutions. It was designed to help with the mass data storage from a variety of mass spectrometry experiments, like MS, MS/MS or $MS^n$ experiments. Interesting features of the file format were the accounting for common mass spectrometry-based proteomics experiments, like database queries, quantitation analysis using stable isotopic labeling strategies and *de novo* sequencing. XML-based file formats for storing large numerical datasets suffer from file size problems, as converting double-sized numbers into text imposes a very large size overhead (and, consequently, slower file read/write operations). The authors got around that specific problem by base64-encoding[4] the spectral (m/z, count) pairs in the *mzXML*-formatted file. Noteworthy is the fact that the authors published, along with the *mzXML* mass data file format, a number of open source software programs aimed at reading or writing data from/to files in this format.[5]

---

2 Standard Generalized Markup Language.

3 As a standard published and maintained by the W3C consortium, the specification is available at http://www.w3.org/XML/.

4 Base64 encoding schemes are used to encode binary data into media designed to deal with textual data (that is, the XML-based data file format). This encoding ensures that the original data remains unmodified during transfer in the data file.

5 http://sourceforge.net/projects/sashimi.

### 2.2.2. *mzData*

Another XML-based format is the *mzData* data file format. This format has been developed inside the Proteomics Standards Initiative framework of the Human Proteome Organization (denominated HUPO/PSI, for Proteomics Standards Initiative). As stated at http://www.psidev.info/, *"The HUPO Proteomics Standards Initiative defines community standards for data representation in proteomics to facilitate data comparison, exchange and verification"*. The *mzData* data file format was designed to encode peak list information along with a number of other metadata of interest to the mass data users. *mzData* also made the spectral (m/z, count) pairs available under a base64 encoding and was thus on par with *mzXML* in terms of speed of file read/write operations. While *mzData* was mainly developed with the exchange and archival of mass data in mind, *mzXML* was developed out of an immediate laboratory-centric need to have long data processing and mining workflows to rest on a unique file format. Therefore, the *mzXML* format had to be able to store/encode all the mass data and metadata required to enable their use right from the start of the data mining process up to its end.

To the file format end user, the functional capabilities of *mzData* and of *mzXML* were equivalent and it turned out that, for software developers, in both the academic and the private sectors, supporting two equivalent formats was too much of a burden. The designers behind each one of the two *mzData* and *mzXML* formats finally agreed to work, under the HUPO/PSI umbrella, to the definition of a novel file format that would bestow the best of both preexisting formats.

### 2.2.3. *mzML*

The *mzML* format elaboration process involved people who participated in the design of the two first formats and people from mass spectrometer vendors. The new file format was released as version 1.0.0 in June 2008.[6, 7] The stated objectives that were agreed upon during the elaboration process were: 1) to keep the format simple; 2) to ensure that the format stays stable over long periods of time (this was a particular requirement from instrument vendors); 3) to integrate in the file format a specific support for selected reaction monitoring (SRM) data; 4) to release along with the data file format specification, free and open source software to read/write *mzML* data files (this software is collectively called "reference implementation" of the data file format).

It is to be noted that the new *mzML* format was designed in great part to help instrument manufacturers and vendors produce software that may rely on a stable mass data file format. This comes at the price of lesser ease of use due to having a specification that needs multiple files to actually work and controlled vocabulary ontologies.

### 2.2.4. Specialty *XML* formats

The field of mass spectrometry for the study of proteins and peptides is so vast that a single file format specification cannot suffice to document the diversity of experimental settings or of data recording schemes, for example. The following are three proteomics specialty file formats developed by the HUPO/PSI:

- *mzIdentML*: this file format has been designed to document the mass spectrometry data that permitted the identification of proteins [8]. In particular, it is able to store the methods and parameters that were used for the database searches that provided protein identification data. The file contains protein detection lists (that is, the set of protein identities that were determined using peptide identifications) and peptide identification lists (that is, lists of peptides as identified by database searches). It is these peptide identification lists that allow to make protein identifications.
- *mzQuantML*: this file format was designed to document, with the finest level of detail, the data and metadata associated with quantification experiments performed using quantification-specific software [9, 10]. Useful metadata that can be stored permit keeping a full logical link (so-called "trail") between raw MS, MS/MS or $MS^n$ mass data (as can be documented in a *mzML*-formatted file) and the quantitative *mzQuantML*-formatted file. Further, the file format makes it possible to also reference intermediate data processing steps like, for example, allowing *XML* elements to store the filename of transition SRM parameter files (like a *TraML* file; *vide infra*) or the filename of the protein identification data file (like a *mzIdentML* file). Because quantification studies require making a number of replicates, with select parameters being varied all along, the *mzQuantML* format provides containers to hold the results of any number of such replicates. The matrix-like structure of the data file format allows to store data with the aim of permitting the comparison of quantification data from one replicate to another in a perfectly arbitrary way. Finally, the *mzQuantML* format is useful with any kind of quantification method, be it based on SILAC, iTRAQ or label-free strategies.

- *TraML*: this file format was designed to overcome a serious fragmentation in the file formats used by vendors or software projects to design, validate and store transitions used in SRM/MRM mass spectrometry experiments. This new HUPO/PSI-developed file format was devised to serve as a common ground for disseminating and widely use useful and successful transitions [11]. The *XML* schema for the *TraML* format comprises ten main data elements that may contain references to a number of outer-file data, like references to publications that contain transitions or references to the software programs that were used to manage the transitions, for example.

#### 2.2.5. Conclusions

To conclude this part, summarizing it, we have made a quick overview of a wealth of mass spectrometry data file formats that have been created over time to meet the ever-expanding requirements to store new kinds of data alongside the technical improvements that were made with the instruments and the methodologies. There are many other open data formats, specializing in other fields of mass spectrometry for biology. For example, one may cite the *imzML* format [12] to document mass spectrometry imaging data or the *PeakML*/*mzMatch* formats [13] that are specialized in metabolomics.

**Table 1** specifies the formats supported by a number of software projects.

### 2.3. Conversion from proprietary vendor formats to open data formats

The *ProteoWizard* project [14] is most renowned for its *msConvert* program that can convert mass spectrometry data files from proprietary mass spectrometer vendor formats to open formats (using each vendor's Microsoft Windows-based proprietary dynamic linking library). This program converts mass

data files from closed proprietary formats to open data archival formats, like *mzXML* or *mzML*. Owing to this capability, the program is almost systematically used as a first step in any FOSS-based mass spectrometric data analysis and mining workflow.

Once the mass data have been converted to open formats, the analysis/mining workflow can be carried over entirely on FOSS computing platforms. The software projects are described in the next sections and will be categorized into two main groups: 1) projects producing non-interactive software, like libraries, that is used by developers to craft integrated mass data processing solutions and 2) projects that produce software that is mainly for use by the end-user, like mass data visualization programs, mass data analysis tools with graphical user interface (GUI) front ends that are to be used interactively.

## 3. Software Projects to Build Analytical Workflows

The vast amounts of data that mass spectrometry produces nowadays need to be handled in powerful ways to make the most out of them. To actually perform an integrated processing of mass spectrometry data, the required programs or library functions need to be chained together in an ordered fashion, with mass data flowing from one processing task to another. Some of the best-regarded software solutions of this kind, aimed at allowing rapid development of site-specific mass data processing workflows are described below, with a special interest in both their ability to handle various open data file formats and their data processing and mining features. This section does not pretend to be exhaustive. Instead, its aim is to show how FOSS has come of age these last ten years in terms of diversity, richness of features and usefulness.

### *3.1. OpenMS*

*OpenMS* is a large software project that aims at providing users with all the necessary tools to tailor mass spectrometry data processing workflows [15]. *OpenMS* is developed from the ground up with a multi-layered structure comprising a core framework providing the C++ building blocks (foundation and kernel classes) required to build file and database in/out routines, data visualization functions and a set of algorithms. The libraries produced by this project provide development functionalities for building all the software programs shipped by the project–collectively called *TOPP*[6]–available to both the developer and the end-user. *TOPP* comprises all the programs that are required by the developer to craft mass spectrometry data processing workflows tailored to any specific need [16]. These programs (more than fifty executable programs) are classified into the following categories: graphical tools, file handling, signal processing and preprocessing, quantitation, map alignment, peptide and protein identification, protein and peptide processing, targeted experiments, peptide property prediction. In the context of *OpenMS*, the task of the software developer is to chain all the needed programs so as to craft analysis pipelines that might be used in proteomics workflows. On top of all the utilities, two main end user graphical interface programs are available: *TOPPView* [17] and *TOPPAS* [18] *TOPPView* allows

6 *"The OpenMS Proteomics Pipeline"*.

the end-user and the software developer alike to view, inspect and interactively apply “effectors” on proteomics data by calling tools available in the TOPP toolbox, such as filters, for example. *TOPPAS* can be called from *TOPPView* and is a powerful graphical analysis workflow builder where the user can assemble tools and routines to easily craft customized mass data processing workflows. The KNIME workflow-specification framework is supported in *OpenMS* [19].

A rather recent addition to the OpenMS project is the Python bindings that allow one to drive the C++ libraries using Python code [20] See [15] for a useful article to bootstrap new users.

License: BSD-3-clause.
Website: https://www.openms.de.
Source code: https://github.com/OpenMS/OpenMS.

### *3.2.* Trans-Proteomic Pipeline

The *Trans-Proteomic Pipeline* (*TPP*) software project [21, 22] gathers a wealth of software programs for the analysis of tandem mass spectrometry (MS/MS) data aimed at the reliable identification of proteins. While the *TPP* developers do not develop conventional peptide identification search engines (like *X!Tandem*), *TPP* nonetheless includes the *SpectraST* program to build and search spectral libraries [23]. It is possible to use conventional search results by first converting them to the *pepXML* format. The data coming from tandem MS-based database searches are fed to a number of programs aimed at validating the peptide identifications:

- *PeptideProphet* [24]: implements a statistical model to estimate the accuracy of peptide sequence correlation with tandem mass spectral data as provided using database searches. The output of the program is a better peptide identification scoring that allows to filter off incorrectly assigned peptides;
- *iProphet* [25]*:* implements a refinement algorithm on top of *PeptideProphet*;
- *PTMProphet*: implements a scoring algorithm to compute confidence of post-translational modification localization on a given peptide.

The quantification processes are implemented in various programs:

- *XPRESS/ASAPRatio* [26, 27]: implement algorithms to accurately calculate the relative abundance of proteins out of peptide stable-isotope tagging experiments followed by ESI-LC/MS analysis;
- *Libra*: module within *TPP* to perform quantification on MS/MS spectra that have isobaric multi-reagent-labeled peptides (like iTRAQ-labeled samples).

The protein identification refinement process is handled by *ProteinProphet* [28]. This program implements an algorithm to validate protein identifications previously performed on the basis of peptides assigned to MS/MS spectra by database search programs.

The *TPP* software contains a wealth of accessory programs for performing computations ranging from format conversion to statistical modeling *via* quantification and spectrum processing.

License: GPLv2 and LGPL
Website: http://tools.proteomecenter.org/software.php

Source code: https://sourceforge.net/projects/sashimi.

### 3.3. ProteoWizard

The *ProteoWizard* software project [14] produces a number of tools along with the library. The *idConvert* command line tool converts between various open protein identification formats, like *pepXML*, *protXML* or *mzIdentML*. The *mscat* tool converts an open data format file to simple text with output to the console. The *msaccess* program provides access to mass spectrometry data files, including spectral data and metadata, and selected ion chromatograms. The program can create pseudo-2D gel images as a visualization of the mass spectrometric data accessed in the file. There are other programs produced by this project that, unfortunately, do not run on non-MS Windows platforms.

From an informatics standpoint, the library code is very well documented and well laid-out. Unfortunately, the build system that is used is extremely complex and a number of projects resolved to extract the minimal set of files needed for their use and integrate them in their own build system. This is sad because this featureful library loses somehow one of its nice characteristics: its generality.

License: Apache2.
Website: http://proteowizard.sourceforge.net.
Source code: https://github.com/ProteoWizard/pwiz.

### 3.4. Python pyMzML

The *pyMzML* project [29] is highly regarded as a solution in the Python world to parse and load data from *mzML* data files such as those obtained from raw files using *msConvert*. In its most recent development iteration, *pyMzML* incorporates innovative gzip compression format classes that allow writing or reading compressed mass spectrometry data. Because their new gzip-compressed data are indexed, it becomes possible to perform random-access reading of the compressed data. Their new format is called *igzip*. The *igzip*-compressed data format is almost as dense as the raw format.

License: MIT.
Website: https://pymzml.readthedocs.io/en/latest.
Source code: https://github.com/pymzml/pymzML.

### 3.5. MzJava

The *MzJava* project [30] is the successor of the *Java Proteomic Library* project (*JPL*) both developed at the Proteome Informatics Group at the Swiss Institute of Bioinformatics. The proteomics applications and the services are made available to the proteomics community *via* the ExPASy server, known for serving– among others–the well-regarded UniProtKB/Swiss-Prot database. This library was born as a means to maximize code reuse inside the software development projects of the group. A

recent project makes use of the *MzJava* library to perform open modification search proteomics. This kind of proteomics does not require prior knowledge of the potential modifications of the proteins. The *Liberator* and *MzMod* programs harness the Apache Spark large scale computing framework to process millions of spectra [31]. First, the *Liberator* program builds spectral libraries and then the *MzMod* searches them in the open modification mode.

License: Apache2.
Website: https://mzjava.expasy.org.
Source code: https://bitbucket.org/sib-pig/mzjava.

### *3.6. Pyteomics* Python proteomics library

The Python language has gained the status of scientific scripting language of choice these last ten years and is more and more used where speed of development is crucial. Its close interpenetration with C/C++- based libraries makes it powerful enough even for the most demanding applications. The *Pyteomics* project [32] produces a Python-based proteomics libraries framework aimed at easing the development of software in which the following features are required: reading LC-MS/MS data, search engine output processing, protein sequence database handling, theoretical prediction of retention times, electrochemical properties of polypeptides, mass and m/z calculations.

License: Apache2.
Website: https://pyteomics.readthedocs.io/en/latest.
Source code: https://github.com/levitsky/pyteomics.

### *3.7. GNU R-based software and Bioconductor repository*

This section deals with GNU R-based software for mass spectrometry that is typically geared towards both developers and computer-savvy end-users. GNU R[7] is a FOSS environment for statistical computing and graphics. Another project, Bioconductor[8], established itself as a repository of tools for the analysis and comprehension of high-throughput genomic data. Bioconductor uses the R statistical programming language and as such is a FOSS project. A large number of R packages for mass spectrometry are developed inside the Bioconductor framework. The following are some examples of R packages designed to be used in mass spectrometry-based research.

- MzR [33]: Parser for netCDF, mzXML, mzData and mzML and mzIdentML.

7 https://www.R-project.org/
8 http://www.bioconductor.org/

- MSnbase [34]: Efficient and elegant R-based processing and visualisation of raw mass spectrometry data
- mzId [35]: An mzIdentML parser for R.
- pepXMLTab [36]: Parsing pepXML files and filter based on peptide FDR.
- MSnID [37]: Utilities for Exploration and Assessment of Confidence of LC-MSn Proteomics Identifications.
- Isobar [38]: Analysis and quantitation of isobarically tagged MSMS proteomics data.
- MALDIquant [39]: Quantitative Analysis of Mass Spectrometry Data
- IsoSpecR [40]: Isotopic cluster calculations.
- Topdownr [41]: Investigation of Fragmentation Conditions in Top-Down Proteomics.
- protViz: Visualizing and Analyzing Mass Spectrometry Related Data in Proteomics (https://cran.r-project.org/package=protViz).
- MSstats [42]: Protein Significance Analysis in DDA, SRM and DIA for Label-free or Label-based Proteomics Experiments.
- Synapter [43]: Label-free data analysis pipeline for optimal identification and quantitation.

Table 1. Synoptic view of the most relevant software projects. a: mzXML, b: MzData, c: mzML, d: mzIdentML, e: mzQuantML, f: TraML, g: pepXML, h: DTA, i: MGF, j: FASTA, k: UniProt XML, l: NCBI, m: ENSEMBL, P: protein, S: saccharide, DBS: database search (for peptide/protein identification), L: Linux, W: Windows, M: macOS.

| Project | File formats | MS data types | Specialty | Operating system | License | Language | Library or desktop | Main polymer | PTM |
|---|---|---|---|---|---|---|---|---|---|
| BatMass [44] | acg | MS<br>MS/MS | Visualization Mining | LWM | Apache2 | Java | Desk | A | n/a |
| Comet [45] | acig | MS<br>MS/MS | Database search (bottom-up) | LW | Apache2 | C++ | Desk | P | + |
| Emzed [46] | ac | MS<br>MS/MS | Visualization Processing Mining | LWM | GPL3 | Python | Both | A | n/a |
| MetaMorpheus [47] | cijk | MS<br>MS/MS | Database search (bottom-up) & Refinement | LWM | MIT | C# | Desk | P | + |
| massXpert [4] | n/a | n/a | Mass data simulation | LWM | GPL3 | C++ | Desk | A | + |
| mineXpert [48] | abci | MS<br>MS/MS<br>$MS^n$<br>IM-MS | Visualization Processing Mining (supports IM-MS data) | LWM | GPL3 | C++ | Desk | A | n/a |
| MzJava [30] | cdghi | MS<br>MS/MS<br>$Ms^n$ | Proteomics pipeline creation | LWM | AGPL3 | Java | Lib | P S | + |
| MZmine2 [49] | abch | MS<br>MS/MS<br>$Ms^n$ | Visualization Statistics Processing Mining | LWM | GPL2 | Java | Desk | A | n/a |
| OpenMS [15] | abcdefi | MS<br>MS/MS<br>IM-MS | Proteomics pipeline creation (TOPPAS) &<br>Data processing/visualization (TOPPView) (supports IM-MS data) | LWM | BSD | C++ | Both | P | + |
| PeptideShaker [50] | di | MS<br>MS/MS | Database search (bottom-up) & Refinement | LW | Apache2 | Java | Desk | P | + |
| | | | | | | | | | |
| Philosopher [51] | klmg | MS<br>MS/MS | Database search (bottom-up) & Refinement | LW | GPL3 | Go | Both | P | + |
| ProteoWizard [14] | abcdfi | MS<br>MS/MS<br>$Ms^{n,}$ | Data format conversion Proteomics utilities (supports IM-MS data) | LW | Apache2 | C++ | Both | P | n/a |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | IM-MS | | | | | | | |
| pyMzML [52] | c | MS<br>MS/MS<br>$Ms^n$<br>IM-MS | MzML data handling | LWM | MIT | Python | Lib | A | n/a |
| Pyteomics [32] | acdfgij | MS<br>MS/MS | Proteomics pipeline creation | LWM | Apache2 | Python | Lib | P | + |
| searchGui [53] | di | MS<br>MS/MS | Database search (bottom-up) | LW | Apache2 | Java | Desk | P | + |
| TopPIC [54] | cj | MS<br>MS/MS | Deconvolution &<br>Database search<br>(top-down/intac protein MS) | LW | Apache2 | C++ | Both | P | + |
| TPP [21] | abcdefi | MS<br>MS/MS | Proteomics pipeline creation | LWM | GPL2, LGPL2 | C/C++,Perl, Java | Both | P | + |
| UniDec [55] | | MS<br>MS/MS<br>IM-MS | Deconvolution (supports IM-MS data) | LWM | BSD-based | C Python | Both | A | n/a |
| xiSPEC [56] | cd | MS<br>MS/MS | Visualization Mining | LWM | Apache2 | PHP JavaScript | Desk | A | n/a |
| X!Tandem [57] | d | MS<br>MS/MS | Database search (bottom-up) & Refinement | LW | Artistic | C | Lib | P | + |
| X!TandemPipeline [58] | ac | MS<br>MS/MS | Protein inference | LW | GPL3 | C++ | Desk | P | + |

# 4. Software projects for the end-user

This section describes some of the most significant software pieces that are designed to be both interactive and intuitive and also to be cross-platform, running on at least two of the GNU/Linux, Microsoft Windows and macOS computing environments. The software offerings described here are of two kinds: software for (semi-)automated protein identification and software for the structural biologist requiring protein or peptide mass spectrometric data visualization, inspection, processing and mining.

## 4.1. Proteomics, Peptide, Protein and Proteoform identification

In the field of protein or proteoform identification, most software, as described above, is in the form of libraries. In this section, end-user software is presented that either provides an integrated environment featuring all the required functionalities or provides a user-friendly interface to extern libraries or programs that perform the database searches.

### 4.1.1. The *X!TandemPipeline/MassChroQ* integrated environment

The main FOSS project that provides a full-featured desktop-based solution to protein identification is *X!TandemPipeline* [58]. The published version was developed using Java. That software piece was fully rewritten in C++ for highly increased performance and numerous feature additions, like peptide theoretical isotope cluster visualization and extracted ion chromatogram browsing. By harnessing the very good features of the *X!Tandem* protein database search program [57]. *X!TandemPipeline* provides a user-friendly interface to the intricacies of database search configuration, on the one hand and protein inference, on the other hand, thus helping the user make the most out of the database search program output. Phosphosite identification is also based on protein database searches and implements grouping algorithms based on the principle of parsimony to provide the user with manageable data. Contrary to what its name seems to imply, *X!TandemPipeline* is able to load identification data obtained using other database search engines than *X!Tandem* in order to perform its protein inference tasks. The companion software, MassChroQ, [59] handles the quantification of the identified proteins, with either labelled or label-free peptides. MassChroQ is also able to handle data from other sources as long as the identification data are formatted according to the MassChroQML format, that is documented.

The programs are coded in C++11 with a portable widget set (Qt libraries) that make them portable to GNU/Linux, Microsoft Windows. Binaries are provided for these two platforms.

License: GPLv3+.
Website: http://pappso.inrae.fr/bioinfo/xtandempipeline.
Source code: https://forgemia.inra.fr/pappso/xtpcpp.

#### 4.1.2. *searchGui* and *PeptideShaker* database search interfacing programs

Not unlike the *X!TandemPipeline* software program described above, *searchGui* and *PeptideShaker* are programs that provide a user-friendly interface to either database searching engines (*searchGui*) or to their output (*PeptideShaker*) [50 , 53]. Written in Java, these two software pieces run on all of the three common computing platforms. The usage logic involves first running *searchGui* to actually perform the database search step and then loading the program's output into *PeptideShaker*. One strength of the *searchGui* program is its support for no less than ten proteomics and *de novo* search engines.

License: Apache2.
Website: https://www.compomics.com.
Source code: https://github.com/compomics/searchgui.
Source code: https://github.com/compomics/peptide-shaker.

#### 4.1.3. *TopPIC*

The *TopPIC* software project [54] deals with intact protein mass spectrometry data analysis. This software builds on top of pioneering software like THRASH [60] and MS-Deconv [61]. The overall aim of the software is the characterization of proteoforms in a sample. The software operates according to a conceptual schema comparable to what has been practiced in bottom-up proteomics since decades. The software first takes in *mzML* data files with centroided peaks (the *msConvert* program allows the conversion to be configured so). Then, the fragmentation mass spectral data are deconvoluted to monoisotopic masses which are in turn used to perform database searches to identify proteoforms. One powerful feature of *TopPIC* is its ability to track down unknown protein chemical modifications. In 2016, Kou *et al.* [54] identified 301 proteins from 1914 proteoform spectrum matches. The data set used for this analysis was from a top-down proteomics experiment on an *E. coli* sample. One interesting result from this study is that combining fragmentation data from CID and ETD provided significantly better proteoform identification results than when only one or the other was used (roughly 20% more than when CID or ETD data were used alone).

From an informatics stand point, the program code is very well laid-out and the build system that is used makes it easy to build the project. Note that the GUI programs are merely user-friendly front ends to the command line-driven programs. The man pages are detailed and the general operation of the software is clearly described.

License: Apache2.
Website: http://proteomics.informatics.iupui.edu/software/toppic.
Source code: https://github.com/toppic-suite/toppic-suite.

## 4.2. Mass spectrometry data visualization and mining software

When mass spectrometry data cannot be processed via automated workflows, in pipeline-only software tools, interactive visualization becomes desirable. This situation is encountered particularly often in structural biology mass spectrometry where the mass spectrometrists use mass spectrometry to probe hyperfine structural properties of peptides or proteins. For example, scientific projects involving particularly complex post-translational modification patterns or chemical modifications of purified proteins and peptides require manual data scrutiny. One most eloquent example is the study of tubulin poly-glutamylation or poly-glycylation, where the substrate is made of a large number of protein isoforms and the post-translational poly-modifications themselves are poly-disperse [62, 63]. In a large spectrum of fields of endeavor, involving induced chemical modifications of proteins or peptides, human eye scrutiny of the obtained mass spectrometric data is crucial to extract data bits not predictably modelled by automated workflows [64, 65]. Finally, one example of compulsory mass spectral data visualization is when isotopic cluster profiles need to be compared between experiments with differential isotopic labelling, like in pulse-chase experiments.

In this section, we will review mass spectrometric data visualization and mining software projects that implement interesting visualization paradigms, that allow one to interrogate the data using a number of criteria, making it possible to probe the mass spectral data sets through all their depth levels.

### *4.2.1. MZmine2*

The *MZmine2* project [49] is a highly regarded software project for LC-MS data processing. Written in Java, it is inherently cross-platform. The capabilities of the software are too large to enumerate. *MZmine2* loads data in various open data formats, like *mzML*, *mzXML*, *mzData*, for example. The features of the program can be categorized in raw data methods, peak list methods, statistical analysis and data visualization. Each category has a number of features that should provide solutions to most data visualization tasks. One missing feature of MZmine2 is support for ion mobility mass spectrometry data.

License: GPLv2.
Website: http://mzmine.github.io.
Source code: https://github.com/mzmine/mzmine2.

### *4.2.2. xiSPEC*

The *xiSPEC* project [56] is an interesting project that provides a web-based mass spectrometry data viewer with interesting capabilities. The user is tasked with the loading of the mass spectrometry data files, in the *mzML* or *MGF* format. While the program is designed to be used in a bottom-up proteomics context, it might be used for deep inspection of the mass data by manual scrutiny of the mass spectra in other contexts. The data loading and handling is performed using the *pyMzML* library and is coded in Python. The data visualization component is written in the ubiquitous JavaScript language. Mass

spectrum visualization can be performed like in other desktop-based software programs, with zooming and panning operations, distance measurement between mass peaks and deconvolution to the mass of the analytes based on their charge state. One interesting feature of the program is the ability given to the user to annotate the spectra. By providing a peptide sequence, the user may explore the MS/MS spectrum by setting select post-translational modifications to try matching the measured data with structural hypotheses. In this respect, the capability of the software to cope with cross-links can be underlined. It is easy to change the cross-linker position to search for relevant matches in the mass spectra.

The source code is available and instructions are provided to build the software so as to self-host the website. The project appears to be maintained. A fully working demonstration website is located at https://spectrumviewer.org.

License: Apache2.
Website: https://spectrumviewer.org.
Source code: https://github.com/Rappsilber-Laboratory/xiSPEC_website.

#### *4.2.3. BatMass*

The *BatMass* project [44] produces a mass spectral data visualization software program that loads *mzML* or *mzXML* data files. One particular strength of the software is its ability to plot mass spectral data in the form of heat maps for multiple files at once. The data navigation in the heat maps allows one to dynamically compare multiple samples for any mass spectral feature of interest by locking all the heat maps' regions to one reference heat map. Another interesting feature is the ability to display the mass spectral data in the form of a table view that may be linked to the heat map representation of the data. By selecting rows of interest in the table view, the heat map region gets updated with a zoom-in operation matching the data selected in the table view. Being able to drive the heat map data exploration from the table view is a powerful data navigation paradigm.

License: Apache2.
Website: https://batmass.org.
Source code: https://github.com/chhh/batmass.

#### *4.2.4. emzed*

The *emzed* software project [46] is an interesting project in the Python language that aims at allowing users to either simply interact with a graphical user interface or scripting that interface without needing too much development skills. The user interface comprises TIC chromatograms, heat maps and tabulated data views that allow integrations to mass spectra. One feature of special interest is the integrated development environment that makes it easy to develop scripts right into the target application. The new software development iteration (*emzed3*) with full Python3 support is already available on the web site, although the documentation is not yet ready.

License: GPLv3+.
Website: https://emzed.ethz.ch.
Source code: https://github.com/uweschmitt/emzed2.

#### 4.2.5. UniDec

The *UniDec* software project [55] is aimed at providing a wide array of features related to mass spectral data deconvolution and analysis. Its support for ion mobility mass spectrometry data is remarkable. It is developed in Python, although intensive numerical computations are handled by C code. Data can be loaded in the software either using proprietary software (harnessing proprietary dynamically linked libraries) or in text format (*mzML* is supported). The *UniDec* software allows one to set some pretty low-level parameters for processing the data. For example, it is possible to define the mass of the charging agent (for protonation, that would be 1 Da, for example). Equally useful, the binning of the m/z values can be configured, which is an important setting to control whenever one starts to scrutinize mass spectral data to the finest detail. One strength of the software is its ability to plot the processed data in a wide variety of ways, in particular relating m/z ratios to charge or to arrival time, in IM-MS experiments. A particularly useful representation of IM-MS data is in the form of cubes where, for example, the three visible faces show heat maps relating the m/z data vs arrival time, the arrival time vs the charge and finally the m/z data vs the charge. The graph plotting can be configured in various ways, producing publication-ready figures.

The project code is very cleanly laid out and the software build process is properly handled with scripts for MS Windows or GNU/Linux.

License: https://github.com/michaelmarty/UniDec/blob/master/LICENSE.
Website: http://unidec.chem.ox.ac.uk/.
Source code: https://github.com/michaelmarty/UniDec.

#### 4.2.6. msXpertSuite

The *msXpertSuite* project gathers two subprojects: *massXpert* and *mineXpert*. *massXpert*, which dates back to the late nineties has gone through a number of full rewrites [4, 66, 67]. *mineXpert* is a more recent project that was published rather recently [48]. Both software pieces are witten in C++.

The *massXpert* software was devised to allow biochemists to model any kind of linear polymer. A grammar allows the biochemist to describe any characteristic of the polymer chemistry to model. For example, it is possible to define isotopes, atoms, monomers, chemical modifications, cross-linkers, enzymatic/chemical cleaving reagents, gas phase fragmentation patterns, ionization rules, for example. The program comes with highly detailed protein, saccharide and nucleic acids polymer chemistry definitions. Once a given polymer chemistry definition is ready, polymer sequences can be defined and subjected to any typical sample processing, either in liquid or gas phase, like chemical modifications, cross-linkings, digestions, fragmentations. For each of these treatments, the matching set of

protein/peptide masses is produced. The program features a large array of functionalities that cannot be described here. The software was designed to be used in parallel with the actual sample analysis, as a decision making aid program. For example, when analyzing peptides from a digestion of a protein that is modified in unusual ways, being able to test structural hypotheses is of paramount importance when “fragmenting everything or anything” is not an option because the mass spectrometry session is being carried over interactively. On the basis of a structural hypothesis that could be verified in the program, the actual fragmentation of a given peptide might make sense to establish the correctness of the hypothesis.

The *mineXpert* program specializes in mass spectral data visualization and mining and is therefore a companion program to *massXpert*. The program loads *mzML*, *mzXML*, *MGF* or text data files. Once the data file is loaded, a TIC chromatogram is automatically computed. In case of IM-MS data, a heat map relating m/z and drift time is also computed. These two plots serve as the starting point for data integrations to any of the available kind of data plots: XIC chromatograms, mass spectra, drift spectra and drift spectrum vs mass spectrum heat maps. A powerful feature of the program is the fact that it is possible to integrate data from any kind of plot to any kind of plot.

Because *mineXpert* was designed with hyperfine mass spectral feature inspection, the user can configure the binning of the m/z data when integrating them to a mass spectrum. The bin size might be defined as an absolute, a ppm-relative or a resolution-relative value. This is useful when handling mass data originating from different vendor software. Smoothing of the traces is possible using the Savitzky-Golay algorithm, of which the user can control the full set of parameters.

*mineXpert* was designed to fulfill the requirements of the author, who has specialized in structural biology mass spectrometry for decades. Manual inspection of complex spectra has been the core of his activity and no software featured the possibility to perform semi-automated external annotation of mass spectra. That feature is central to the process of data mining and is designed like so: the user selects the output of the mining discoveries, like the clipboard, the program’s console window, a file on disk or any combination thereof. Once this setting is defined, the user can start mining the data, by zooming-in a mass spectral feature, for example. Deconvolutions are available at a mouse drag operation and compute the charge of the analyte and its Mr. The intensity of the mass spectral feature might be computed also. All these computed values are stored in memory until the next computation. At any time the user can hit the SPACE keyboard key and all these data are written to the output configured initially. The user can configure the stanza that the program should write to the output using a set of placeholder format strings that will be replaced with the values calculated above. Another key feature of *mineXpert* is the isotopic cluster calculation interface to the *IsoSpec* library [68]. The user enters the formula and charge state of the analyte, specifies the resolution of the instrument and *mineXpert* computes a mass spectrum that matches the theoretical isotopic distribution returned by *IsoSpec*. The *IsoSpec* library is not easy to configure, as it is clearly optimized for accuracy and speed. *mineXpert* allows the user to configure intuitively and to the finest detail the isotopic abundances of the chemical elements in the formula so as to model isotopic distributions for labelled molecules.

As mentioned above, *mineXpert* features a full support for IM-MS data and was tested with data sets acquired on Waters Synapt2 and Agilent instruments.

One limitation of *mineXpert* is that it only handles MS1 data, a limitation that was addressed in a full rewrite of the code. The new *mineXpert2* software program is now available and will replace gradually the prior version. It features arbitrary $MS^n$ capabilities, very much improved performance, with the pervasive use of multi-threading and a fully rewritten plot widget framework allowing for a much greater flexibility in the handling of plots. A table view of the data is also available for data filtering using a number of criteria that can be logically combined: the MS level, the retention time, the drift time, the precursor ion m/z value, the precursor ion charge or the precursor ion spectrum index. The set of spectra that remain after the filtering can be integrated to all the possible destinations: mass spectrum, TIC chromatogram, drift spectrum, various heat maps relating either retention time with mass spectra or drift spectra or drift time with mass spectra.

*massXpert* and *mineXpert/2* are shipped with detailed user manuals in HTML and PDF formats. Video tutorials are available from the website. The software is available in Debian GNU/Linux and derivative distributions, like Ubuntu. Binaries for MS Windows and Apple macOS are provided.

License: GPLv3+.
Website: http://msxpertsuite.org.
Source code: https://salsa.debian.org/debichem-team/massxpert.
Source code: https://salsa.debian.org/debichem-team/minexpert2.

## Conclusion

This review has shown that the Free and Open Source Software movement has allowed the flourishing of innumerable projects aimed at accomplishing a variety of tasks like, for example, mass spectrometric data archiving and storage, pipelined processing for analysis and sophisticated interactive visualization methods. Indeed, in that FOSS context, any scientist willing to learn computer science can set out to code software specifically tailored to any particular mass spectrometric data processing need. The author of this review is himself a self-taught FOSS developer with biochemistry and organic chemistry training backgrounds. For this review, only FOSS projects that produce correctly maintained software and properly redacted documentation were retained. An interesting observation is that the software projects described here produce two kinds of perfectly complementary software artifacts: libraries sporting a broad set of useful functions and end-user desktop-based programs that make use of these libraries for their inner workings and that provide end users with intuitive graphical interfaces. Cross-platform software development is fundamental in ensuring the widest dissemination of software. In general, creating FOSS projects that run only on proprietary platforms seems like a contradiction (one notable exception is the software that converts data files from proprietary formats to open standard formats). Sadly, some good software projects are developed with proprietary languages or are designed to run only on proprietary platforms. These software projects could not be integrated in this review. I would like to conclude this review with one appeal to mass spectrometry vendors: they could make their dynamic linking libraries commonly shipped with their own software compatible with UNIX-like systems (GNU/Linux and Apple macOS, to name the most common ones). These libraries could be

coded using standard languages supported anywhere, like the industry-standard C++, making these libraries usable on these UNIX-like platforms. Alternatively, they could open their data format just as Bruker did with their timsTOF data format.

## List of Abbreviations

FOSS - Free and Open Source Software.

GUI- Graphical User Interface.

**Consent for publication**

Not applicable

**Funding**

**None**

**Conflict of Interest**

Authors declare no conflict of interest.

## Acknowledgements

FR is a senior research scientist at CNRS, Institut de Chimie, working in UMR8120 CNRS.
Prof. Christine Dillmann and the PAPPSO team at GQE-Le Moulon are acknowledged for their strong support and commitment to the use of Free and Open Source Software. The Debian GNU/Linux distribution project is acknowledged for providing any scientist with a totally FOSS computing platform suitable for any kind of mass spectrometry data handling task.

## References

[1] R. Smith, “Conversations with 100 Scientists in the Field Reveal a Bifurcated Perception of the State of Mass Spectrometry Software.,” *J Proteome Res*, vol. 17, no. 4, pp. 1335–1339, 2018, doi: 10.1021/acs.jproteome.8b00015.

[2] P. Lampen, H. Hillig, A. N. Davies, and M. Linscheid, “JCAMP-DX for Mass Spectrometry,” *Appl Spectrosc*, vol. 48, pp. 1545–1552, 1994.

[3] R. K. Rew and G. P. Davis, “NetCDF: An Interface for Scientific Data Access,” *IEEE Comput. Graph. Appl.*, vol. 10, no. 4, pp. 76–82, 1990.

[4] F. Rusconi, “MassXpert 2: A cross-platform software environment for polymer chemistry modelling and simulation/analysis of mass spectrometric data.,” *Bioinformatics*, vol. 25, no. 20, pp. 2741–2742, 2009, doi: 10.1093/bioinformatics/btp504.
[5] P. G. A. Pedrioli *et al.*, “A common open representation of mass spectrometry data and its application to proteomics research.,” *Nat Biotechnol*, vol. 22, no. 11, pp. 1459–1466, 2004, doi: 10.1038/nbt1031.
[6] E. Deutsch, “MzML: A single, unifying data format for mass spectrometer output.,” *Proteomics*, vol. 8, no. 14, pp. 2776–2777, 2008, doi: 10.1002/pmic.200890049.
[7] L. Martens *et al.*, “MzML–a community standard for mass spectrometry data.,” *Mol Cell Proteomics*, vol. 10, no. 1, pp. R110-000133, 2011, doi: 10.1074/mcp.R110.000133.
[8] A. R. Jones *et al.*, “The mzIdentML data standard for mass spectrometry-based proteomics results.,” *Mol Cell Proteomics*, vol. 11, no. 7, pp. M111-014381, 2012, doi: 10.1074/mcp.M111.014381.
[9] S. Orchard *et al.*, “Tackling quantitation: A report on the annual Spring Workshop of the HUPO-PSI 28-30 March 2010, Seoul, South Korea.,” *Proteomics*, vol. 10, no. 17, pp. 3062–3066, 2010, doi: 10.1002/pmic.201090075.
[10] M. Walzer *et al.*, “The mzQuantML data standard for mass spectrometry-based quantitative studies in proteomics.,” *Mol Cell Proteomics*, vol. 12, no. 8, pp. 2332–2340, 2013, doi: 10.1074/mcp.O113.028506.
[11] E. W. Deutsch *et al.*, “TraML–a standard format for exchange of selected reaction monitoring transition lists.,” *Mol Cell Proteomics*, vol. 11, no. 4, pp. R111-015040, 2012, doi: 10.1074/mcp.R111.015040.
[12] T. Schramm *et al.*, “ImzML–a common data format for the flexible exchange and processing of mass spectrometry imaging data.,” *J Proteomics*, vol. 75, no. 16, pp. 5106–5110, 2012, doi: 10.1016/j.jprot.2012.07.026.
[13] R. A. Scheltema, A. Jankevics, R. C. Jansen, M. A. Swertz, and R. Breitling, “PeakML/mzMatch: A file format, Java library, R library, and tool-chain for mass spectrometry data analysis.,” *Anal Chem*, vol. 83, no. 7, pp. 2786–2793, 2011, doi: 10.1021/ac2000994.
[14] D. Kessner, M. Chambers, R. Burke, D. Agus, and P. Mallick, “ProteoWizard: Open source software for rapid proteomics tools development.,” *Bioinformatics*, vol. 24, no. 21, pp. 2534–2536, 2008, doi: 10.1093/bioinformatics/btn323.
[15] H. L. Rost *et al.*, “OpenMS: A flexible open-source software platform for mass spectrometry data analysis.,” *Nat Methods*, vol. 13, no. 9, pp. 741–748, 2016, doi: 10.1038/nmeth.3959.
[16] A. Bertsch, C. Gropl, K. Reinert, and O. Kohlbacher, “OpenMS and TOPP: Open source software for LC-MS data analysis.,” *Methods Mol Biol*, vol. 696, pp. 353–367, 2011, doi: 10.1007/978-1-60761-987-1_23.
[17] M. Sturm and O. Kohlbacher, “TOPPView: An open-source viewer for mass spectrometry data.,” *J Proteome Res*, vol. 8, no. 7, pp. 3760–3763, 2009, doi: 10.1021/pr900171m.
[18] J. Junker, C. Bielow, A. Bertsch, M. Sturm, K. Reinert, and O. Kohlbacher, “TOPPAS: A graphical workflow editor for the analysis of high-throughput proteomics data.,” *J Proteome Res*, vol. 11, no. 7, pp. 3914–3920, 2012, doi: 10.1021/pr300187f.
[19] S. Aiche *et al.*, “Workflows for automated downstream data analysis and visualization in large-scale computational mass spectrometry.,” *Proteomics*, vol. 15, no. 8, pp. 1443–1447, 2015, doi: 10.1002/pmic.201400391.
[20] H. L. Rost, U. Schmitt, R. Aebersold, and L. Malmstrom, “PyOpenMS: A Python-based interface to the OpenMS mass-spectrometry algorithm library.,” *Proteomics*, vol. 14, no. 1, pp. 74–77, 2014, doi: 10.1002/pmic.201300246.
[21] E. W. Deutsch *et al.*, “A guided tour of the Trans-Proteomic Pipeline.,” *Proteomics*, vol. 10, no. 6, pp. 1150–1159, 2010, doi: 10.1002/pmic.200900375.

[22] E. W. Deutsch, L. Mendoza, D. Shteynberg, J. Slagel, Z. Sun, and R. L. Moritz, “Trans-Proteomic Pipeline, a standardized data processing pipeline for large-scale reproducible proteomics informatics.,” *Proteomics Clin Appl*, vol. 9, no. 7–8, pp. 745–754, 2015, doi: 10.1002/prca.201400164.
[23] H. Lam *et al.*, “Development and validation of a spectral library searching method for peptide identification from MS/MS.,” *Proteomics*, vol. 7, no. 5, pp. 655–667, 2007, doi: 10.1002/pmic.200600625.
[24] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold, “Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search.,” *Anal Chem*, vol. 74, no. 20, pp. 5383–5392, 2002.
[25] D. Shteynberg *et al.*, “IProphet: Multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates.,” *Mol Cell Proteomics*, vol. 10, no. 12, pp. M111-007690, 2011, doi: 10.1074/mcp.M111.007690.
[26] D. K. Han, J. Eng, H. Zhou, and R. Aebersold, “Quantitative profiling of differentiation-induced microsomal proteins using isotope-coded affinity tags and mass spectrometry.,” *Nat Biotechnol*, vol. 19, no. 10, pp. 946–951, 2001, doi: 10.1038/nbt1001-946.
[27] X.-J. Li, H. Zhang, J. A. Ranish, and R. Aebersold, “Automated statistical analysis of protein abundance ratios from data generated by stable-isotope dilution and tandem mass spectrometry.,” *Anal Chem*, vol. 75, no. 23, pp. 6648–6657, 2003, doi: 10.1021/ac034633i.
[28] A. I. Nesvizhskii, A. Keller, E. Kolker, and R. Aebersold, “A statistical model for identifying proteins by tandem mass spectrometry.,” *Anal Chem*, vol. 75, no. 17, pp. 4646–4658, 2003.
[29] M. Kosters *et al.*, “PymzML v2.0: Introducing a highly compressed and seekable gzip format.,” *Bioinformatics*, vol. 34, no. 14, pp. 2513–2514, 2018, doi: 10.1093/bioinformatics/bty046.
[30] O. Horlacher, F. Nikitin, D. Alocci, J. Mariethoz, M. Muller, and F. Lisacek, “MzJava: An open source library for mass spectrometry data processing.,” *J Proteomics*, vol. 129, pp. 63–70, 2015, doi: 10.1016/j.jprot.2015.06.013.
[31] O. Horlacher, F. Lisacek, and M. Muller, “Mining Large Scale Tandem Mass Spectrometry Data for Protein Modifications Using Spectral Libraries.,” *J Proteome Res*, vol. 15, no. 3, pp. 721–731, 2016, doi: 10.1021/acs.jproteome.5b00877.
[32] L. I. Levitsky, J. A. Klein, M. V. Ivanov, and M. V. Gorshkov, “Pyteomics 4.0: Five Years of Development of a Python Proteomics Framework.,” *J Proteome Res*, vol. 18, no. 2, pp. 709–714, 2019, doi: 10.1021/acs.jproteome.8b00717.
[33] S. N. Bernd Fischer, *mzR*. Bioconductor, 2017.
[34] L. Gatto, S. Gibb, and J. Rainer, “MSnbase, Efficient and Elegant R-Based Processing and Visualization of Raw Mass Spectrometry Data,” *J. Proteome Res.*, Sep. 2020, doi: 10.1021/acs.jproteome.0c00313.
[35] V. A. P. W. C. F. G. Thomas Lin Pedersen, *mzID*. Bioconductor, 2017.
[36] Xiaojing Wang, *pepXMLTab*. Bioconductor, 2017.
[37] V. P. W. C. F. L. Gatto, *MSnID*. Bioconductor, 2017.
[38] F. P. Breitwieser *et al.*, “General statistical modeling of data from protein relative expression isobaric tags.,” *J Proteome Res*, vol. 10, no. 6, pp. 2758–2766, 2011, doi: 10.1021/pr1012784.
[39] S. Gibb and K. Strimmer, “MALDIquant: A versatile R package for the analysis of mass spectrometry data.,” *Bioinformatics*, vol. 28, no. 17, pp. 2270–2271, 2012, doi: 10.1093/bioinformatics/bts447.
[40] M. K. Łącki, D. Valkenborg, and M. P. Startek, “IsoSpec2: Ultrafast Fine Structure Calculator,” *Anal. Chem.*, vol. 92, no. 14, pp. 9472–9475, Jul. 2020, doi: 10.1021/acs.analchem.0c00959.
[41] P. V. Shliaha *et al.*, “Maximizing Sequence Coverage in Top-Down Proteomics By Automated Multimodal Gas-Phase Protein Fragmentation,” *Anal. Chem.*, vol. 90, no. 21, pp. 12519–12526, Nov. 2018, doi: 10.1021/acs.analchem.8b02344.

[42] M. Choi *et al.*, “MSstats: An R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments.,” *Bioinformatics*, vol. 30, no. 17, pp. 2524–2526, 2014, doi: 10.1093/bioinformatics/btu305.
[43] N. J. Bond, P. V. Shliaha, K. S. Lilley, and L. Gatto, “Improving Qualitative and Quantitative Performance for MS $^{E}$ -based Label-free Proteomics,” *J. Proteome Res.*, vol. 12, no. 6, pp. 2340–2353, Jun. 2013, doi: 10.1021/pr300776t.
[44] D. M. Avtonomov, A. Raskind, and A. I. Nesvizhskii, “BatMass: A Java Software Platform for LC-MS Data Visualization in Proteomics and Metabolomics.,” *J Proteome Res*, vol. 15, no. 8, pp. 2500–2509, 2016, doi: 10.1021/acs.jproteome.6b00021.
[45] J. K. Eng, M. R. Hoopmann, T. A. Jahan, J. D. Egertson, W. S. Noble, and M. J. MacCoss, “A Deeper Look into Comet—Implementation and Features,” *J. Am. Soc. Mass Spectrom.*, vol. 26, no. 11, pp. 1865–1874, Nov. 2015, doi: 10.1007/s13361-015-1179-x.
[46] P. Kiefer, U. Schmitt, and J. A. Vorholt, “EMZed: An open source framework in Python for rapid and interactive development of LC/MS data analysis workflows.,” *Bioinformatics*, vol. 29, no. 7, pp. 963–964, 2013, doi: 10.1093/bioinformatics/btt080.
[47] S. K. Solntsev, M. R. Shortreed, B. L. Frey, and L. M. Smith, “Enhanced Global Post-translational Modification Discovery with MetaMorpheus,” *J. Proteome Res.*, vol. 17, no. 5, pp. 1844–1851, May 2018, doi: 10.1021/acs.jproteome.7b00873.
[48] F. Rusconi, “MineXpert: Biological Mass Spectrometry Data Visualization and Mining with Full JavaScript Ability.,” *J Proteome Res*, vol. 18, no. 5, pp. 2254–2259, 2019, doi: 10.1021/acs.jproteome.9b00099.
[49] T. Pluskal, S. Castillo, A. Villar-Briones, and M. Oresic, “MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data.,” *BMC Bioinformatics*, vol. 11, p. 395, 2010, doi: 10.1186/1471-2105-11-395.
[50] M. Vaudel *et al.*, “PeptideShaker enables reanalysis of MS-derived proteomics data sets.,” *Nat Biotechnol*, vol. 33, no. 1, pp. 22–24, 2015, doi: 10.1038/nbt.3109.
[51] F. da Veiga Leprevost *et al.*, “Philosopher: a versatile toolkit for shotgun proteomics data analysis,” *Nat. Methods*, vol. 17, no. 9, pp. 869–870, Sep. 2020, doi: 10.1038/s41592-020-0912-y.
[52] T. Bald, J. Barth, A. Niehues, M. Specht, M. Hippler, and C. Fufezan, “PymzML–Python module for high-throughput bioinformatics on mass spectrometry data.,” *Bioinformatics*, vol. 28, no. 7, pp. 1052–1053, 2012, doi: 10.1093/bioinformatics/bts066.
[53] H. Barsnes and M. Vaudel, “SearchGUI: A Highly Adaptable Common Interface for Proteomics Search and de Novo Engines.,” *J Proteome Res*, vol. 17, no. 7, pp. 2552–2555, 2018, doi: 10.1021/acs.jproteome.8b00175.
[54] Q. Kou, L. Xun, and X. Liu, “TopPIC: A software tool for top-down mass spectrometry-based proteoform identification and characterization.,” *Bioinformatics*, vol. 32, no. 22, pp. 3495–3497, 2016, doi: 10.1093/bioinformatics/btw398.
[55] M. T. Marty, A. J. Baldwin, E. G. Marklund, G. K. A. Hochberg, J. L. P. Benesch, and C. V. Robinson, “Bayesian deconvolution of mass and ion mobility spectra: From binary interactions to polydisperse ensembles.,” *Anal Chem*, vol. 87, no. 8, pp. 4370–4376, 2015, doi: 10.1021/acs.analchem.5b00140.
[56] L. Kolbowski, C. Combe, and J. Rappsilber, “XiSPEC: Web-based visualization, analysis and sharing of proteomics data.,” *Nucleic Acids Res*, vol. 46, no. W1, pp. W473–W478, 2018, doi: 10.1093/nar/gky353.
[57] R. Craig and R. C. Beavis, “TANDEM: Matching proteins with tandem mass spectra.,” *Bioinformatics*, vol. 20, no. 9, pp. 1466–1467, 2004, doi: 10.1093/bioinformatics/bth092.
[58] O. Langella, B. Valot, T. Balliau, M. Blein-Nicolas, L. Bonhomme, and M. Zivy, “X! TandemPipeline: A Tool to Manage Sequence Redundancy for Protein Inference and Phosphosite

Identification.,” *J Proteome Res*, vol. 16, no. 2, pp. 494–503, 2017, doi: 10.1021/acs.jproteome.6b00632.
[59] B. Valot, O. Langella, E. Nano, and M. Zivy, “MassChroQ: A versatile tool for mass spectrometry quantification.,” *Proteomics*, vol. 11, no. 17, pp. 3572–3577, 2011, doi: 10.1002/pmic.201100120.
[60] D. M. Horn, R. A. Zubarev, and F. W. McLafferty, “Automated reduction and interpretation of high resolution electrospray mass spectra of large molecules.,” *J Am Soc Mass Spectrom*, vol. 11, no. 4, pp. 320–332, 2000, doi: 10.1016/s1044-0305(99)00157-9.
[61] X. Liu *et al.*, “Deconvolution and database search of complex tandem mass spectra of intact proteins: A combinatorial approach.,” *Mol Cell Proteomics*, vol. 9, no. 12, pp. 2772–2782, 2010, doi: 10.1074/mcp.M110.002766.
[62] S. Gadadhar *et al.*, “Tubulin glycylation controls primary cilia length.,” *J Cell Biol*, vol. 216, no. 9, pp. 2701–2713, 2017, doi: 10.1083/jcb.201612050.
[63] V. Redeker, “Mass spectrometry analysis of C-terminal posttranslational modifications of tubulins.,” *Methods Cell Biol*, vol. 95, pp. 77–103, 2010, doi: 10.1016/S0091-679X(10)95006-1.
[64] L. A. Alvarez, F. Merola, M. Erard, and F. Rusconi, “Mass spectrometry-based structural dissection of fluorescent proteins.,” *Biochemistry*, vol. 48, no. 18, pp. 3810–3812, 2009, doi: 10.1021/bi900327f.
[65] V. Berthelot *et al.*, “An analytical workflow for the molecular dissection of irreversibly modified fluorescent proteins.,” *Anal Bioanal Chem*, vol. 405, no. 27, pp. 8789–8798, 2013, doi: 10.1007/s00216-013-7326-y.
[66] F. Rusconi, “GNU polyxmass: A software framework for mass spectrometric simulations of linear (bio-)polymeric analytes.,” *BMC Bioinformatics*, vol. 7, p. 226, 2006, doi: 10.1186/1471-2105-7-226.
[67] F. Rusconi and M. Belghazi, “Desktop prediction/analysis of mass spectrometric data in proteomic projects by using massXpert.,” *Bioinformatics*, vol. 18, no. 4, pp. 644–645, 2002, doi: 10.1093/bioinformatics/18.4.644.
[68] M. K. Lacki, M. Startek, D. Valkenborg, and A. Gambin, “IsoSpec: Hyperfast Fine Structure Calculator.,” *Anal Chem*, vol. 89, no. 6, pp. 3272–3277, 2017, doi: 10.1021/acs.analchem.6b01459.