Journal of proteome research

Technical Note

pubs.acs.org/jpr

Open-Source, Platform-Independent Library and Online Scripting Environment for Accessing Thermo Scientific RAW Files

Pieter Kelchtermans,[†,‡,§,∥,▽] Ana S. C. Silva,[†,‡,§,▽] Andrea Argentini,[†,‡,§] An Staes,[†,‡] Jonathan Vandenbussche,[†,‡] Kris Laukens,[⊥,#] Dirk Valkenborg,[∥,◆,○] and Lennart Martens*[,†,‡,§]

[†]Medical Biotechnology Center, VIB, Albert Baertsoenkaai 3, 9000 Ghent, Belgium

[‡]Department of Biochemistry, Faculty of Medicine and Health Sciences, Ghent University, Albert Baertsoenkaai 3, 9000 Ghent, Belgium

[§]Bioinformatics Institute Ghent, Ghent University, 9000 Ghent, Belgium

[∥]Flemish Institute for Technological Research (VITO), Boeretang 200, 2400 Mol, Belgium

[⊥]Department of Mathematics and Computer Science, University of Antwerp, 2020 Antwerp, Belgium

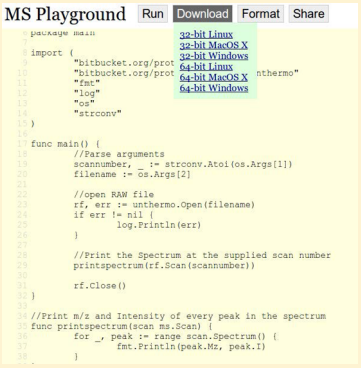
[#]Biomedical Informatics Research Center Antwerp (biomina), University of Antwerp/Antwerp University Hospital, 2020 Antwerp, Belgium

[◆]CFP, University of Antwerp, Groenenborgerlaan 171, 2020 Antwerp, Belgium

[○]I-BioStat, Hasselt University, Agoralaan, Building D, 3590 Diepenbeek, Belgium

Supporting Information

ABSTRACT: Mass spectrometers typically output data in proprietary binary formats. While converter suites and standardized XML formats have been developed in response, these conversion steps come with non-negligible computational time and storage space overhead. As a result, simple, everyday data inspection tasks are often beyond the skills of the mass spectrometrist, who is unable to freely access the acquired data. We therefore here describe the *unthermo* library for convenient, platform-independent access to Thermo Scientific RAW files and the associated online playground to transform small and easily understandable scriptlets into executable programs for end-users. By fostering the provision of code examples and snippet exchange, the interested mass spectrometrist or researcher can use this playground to quickly assemble custom scripts for their particular purpose. In this way, the data in these RAW files can be mined much more readily and directly by the user, and fast, automated raw data extraction or analysis can finally become part of the daily routine of the mass spectrometrist.



KEYWORDS: *Thermo RAW data, software library, data access*

A key milestone in any mass spectrometry experiment is the transition between wet- and dry-lab workflows. This transition is marked by the acquisition of raw data by the mass spectrometer, resulting in the output of a vendor-specific binary file. In the case of Thermo Scientific instrumentation, the typical output is a .RAW file. Immediate access to the information in this file is quite useful, for instance, as a means to perform rapid quality assessment of the data[1] or for quantification purposes.[2] Yet, because these raw data formats are usually proprietary, vendor software is required to view or analyze their content.[3] Fortunately, vendors are increasingly open about making their relevant libraries accessible to third parties (e.g., MSFileReader for Thermo Scientific, WiffReader for AB SCIEX, CompassXtract for Bruker, MHDAC for Agilent, and WRDAC for Waters). These software interfaces allow tools or workflows to be built that rely on direct access to raw data files,[4] even though access can still be cumbersome due to operating system limitations (vendor libraries are typically developed only for Microsoft Windows), inadequately commented libraries, or lack of detailed developer information.[3,5] Rather than work with the raw file directly, vendor-specific data files (or parts thereof) can be converted to a standard format (notably mzML[6]) by software tools such as ProteoWizard.[7] These standard files can, in turn, be read by generic readers such as the ms-data-core-api.[8] However, conversion comes with its own limitations. Perhaps most importantly, there is always processing overhead involved, and standard data formats come with a notable size increase compared to that of the original raw file. This overhead becomes particularly noticeable when performing simple tasks such as quality checks or rapid visualizations.

Indeed, despite the availability of the above-mentioned approaches for accessing raw data, very few mass spectrometrists today are able to interact directly with the data they obtain from their instrument, hampering their ability to make

Received: August 19, 2015
Published: October 19, 2015

DOI: 10.1021/acs.jproteome.5b00778
J. Proteome Res. 2015, 14, 4940−4943

MS Playground Run Download Format Share Plot About

```
//The printspectrum tool prints out the spectrum (mz and intensity values) of a
//Thermo RAW File
//
//  Every line of the output is a peak registered by the mass spectrometer
//  characterized by an m/z value in Da and an intensity in the mass spectrometer's unit of abundance
package main

import (
        "bitbucket.org/proteinspector/ms"
        "bitbucket.org/proteinspector/ms/unthermo"
        "flag"
        "fmt"
        "log"
)

func main() {
        var scannumber int
        var filename string

        //Parse arguments
        flag.IntVar(&scannumber, "sn", 1, "the scan number")
        flag.StringVar(&filename, "raw", "small.RAW", "name of the RAW file")
        flag.Parse()

        //open RAW file
        file, err := unthermo.Open(filename)
        if err != nil {
                log.Fatal(err)
        }
```

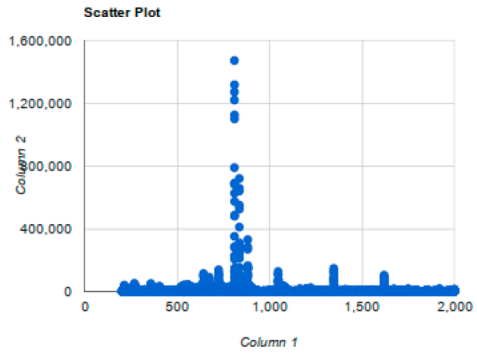


Figure 1. Web interface for MS Playground. The yellow text box holds the Go code. The top buttons allow the user to run or download the code with textual output or to format or share the code. In this case, the Plot option is activated, and numeric results are displayed in a scatter plot that is displayed at the bottom. (This figure is a screenshot taken by the authors and is in the public domain.)

the most of their analytics. In order to provide convenient access to the raw data, we describe our platform-independent system for straightforward access to Thermo Scientific RAW files. On the basis of the approach followed by the *unfinnigan* project,[9] we have built the *unthermo* open-source Go library to read Thermo Scientific raw data. While our library provides full access to the RAW files, our library is not intended to provide a perfect duplication of all existing calls in other libraries such as the Thermo Visual Basic API or ProteoWizard API (which is built on top of the Thermo API). Furthermore, we have provided an online code playground around this library that can be used to transform simple scriptlets to standalone, platform-independent executables. The playground is also accompanied by a code snippet wiki to stimulate developing and sharing of common code.

Importantly, our choice to reimplement *unfinnigan* in Go (instead of forking the existing Perl code) is motivated by the much easier path from Go to compiled and statically linked executables. This allows our *unthermo* library to be used on many different platforms or even entire compute clusters[10] without requiring any installation of libraries or interpreters. A thorough description of the Go Playground is already published by the Go authors.[11] Our MS Playground starts from this basic functionality and adds three major features: (i) it can serve downloadable binaries for a variety of platforms, (ii) it can graphically display results, and (iii) it also hosts a version of *unthermo* that runs on specific server-hosted test RAW files.

Upon opening a RAW file with the library, only scan indices are read, so random access to any indexed part of the binary files is possible within *unthermo*. Details about the internal data structures read by the library are described in the documentation, which can be found online at https://godoc.org/bitbucket.org/proteinspector/ms/unthermo. It is important to note that the *unthermo* library is a subpackage of *ms*, a general purpose mass spectrometry library. This means that other instrument file readers can be implemented in a similar way under the umbrella of this general purpose library. It should be noted, however, that the *unthermo* library does not yet support the parsing of RAW files from the latest Orbitrap Fusion instruments.

Using a simple example that prints every spectrum in a RAW file, we will illustrate the use of our library. When navigating a web browser to http://ms-playground.appspot.com/, a code box and a set of buttons are presented as displayed in Figure 1. In this code box, anyone can edit Go code that can subsequently be either run online or compiled for offline use. Note that installation of a local copy of MS Playground is also possible using the tools mentioned above, but this procedure is

DOI: 10.1021/acs.jproteome.5b00778
J. Proteome Res. 2015, 14, 4940−4943

rather technical and will require local support from a system administrator or developer.

The default program provided is tasked with loading the required libraries and opening and closing the RAW file. It stops with an error message if the file cannot be opened. Function *printspectrum* calls *unthermo* methods, looping over all peaks in the provided scan's spectrum. For every peak, it simply prints the m/z value and intensity. A thoroughly annotated copy of the code can be found in the Supporting Information.

The program in the code box can be compiled by selecting the Run button, which will yield output in the form of two floating point numbers per line. The first column corresponds to the m/z value, and the second column, to the intensity value. Clicking the Download button opens a list box wherein command-line executables can be downloaded for multiple platforms (i.e., Linux, MacOS X, or Windows) and hardware architectures (i.e., 32-bit or 64-bit). The playground also offers a scatter plot generator for this type of two-column output by enabling the Plot checkbox and selecting the Run button again. This action will plot the first column on the horizontal axis and the second column on the vertical axis. If a user wants to store the program in the code box for later use or to share it with colleagues, then the Share button should be used. This button provides a URL that will link to the adjusted code, a functionality that is especially convenient for data provenance.

Note that the name of the RAW file is hard-coded in the example program. This file, small.RAW, is stored on the server, so users of the MS Playground can test their algorithms directly online by using this example file. Obviously, hardcoding file names would be very impractical in practice, as users typically want to run the command-line executable on different RAW files. Therefore, Go provides the flag package that allows command-line flags to be supplied to executables such that users are able to define the RAW file on which the executable should operate at invocation time. An example of the *printspectrum* program that uses flags can be found on the MS Playground Web site and in the Supporting Information.

It should also be noted that additional test RAW files can be submitted to the developers by users who may have a specific type of RAW file that would be useful to have available in the online environment. Adding such a RAW file to the system is on a per-request basis, however, to avoid duplication and to avoid the creation of a *de facto* RAW file repository, which is not the purpose of MS Playground.

In the next two paragraphs, we will briefly introduce three other, more advanced, applications that are all implemented in the MS Playground appspot. The code can be found in the Supporting Information and on the *github* wiki page for the MS Playground: https://github.com/pkelchte/ms-playground/wiki. This wiki page also contains more commonly used snippets. github users are encouraged to extend this wiki with new examples, which is facilitated by the lack of access restrictions on that part of the repository. A typical use of the wiki would be a developer going through the wiki on a user's computer, picking relevant examples on the wiki, and extending these examples to the user's liking in the MS Playground, ultimately yielding a shareable and downloadable tool.

The first more advanced example tool extracts an ion chromatogram from all MS1 scans. It loops over all scans and loads the spectrum for MS1 scans only. From each MS1 spectrum, the algorithm selects the highest peak within a certain tolerance around a specific m/z value. This m/z tolerance is specified at the start of the program. The output is a list of the times that a peak was detected with this m/z value, along with the intensity of the peak.

The second example program inserts iTRAQ (or TMT) reporter ions from HCD scans into CID spectra to simplify identification and quantification, as described in ref 12. Here, the mass spectrometer is set up to perform both HCD and CID scans for each selected precursor, with the CID spectra intended for identification, while the HCD scans will contain quantitative data in the form of iTRAQ (or TMT) reporter ions. Ideally, the quantitative data from the reporter ions should be added to the CID spectra, allowing identification and quantification to occur on the same spectra. The program, therefore, loops over all scans in a file, accumulating CID scans and HCD reporter ion peaks in the respective hash maps *cidScans* and *hcdPeakSpectra*. These maps are indexed by precursor m/z, which is identical for the pairs of HCD and CID MS2 scans following an MS1 scan. If a complete set of MS2 scans is collected, then the HCD-derived reporter ion peaks are merged into the corresponding CID scans. This extended scan is subsequently printed in MGF format. The HCD-derived reporter ion peaks, *reporterIons(scan.Spectrum())*, are selected as the highest peaks in the m/z windows centered on the theoretical reporter ion m/z and with a width corresponding to a given tolerance in parts per million.

The third example tool, *peakstats*, calculates the CPTAC quality control parameters used to monitor chromatographic peak shifts over time[1] for peptides derived from a given standard protein (e.g., bovine serum albumin or enolase). It extracts the maximal peak intensity, the retention time at that peak, and the full width at half-maximum of relevant peptide ions.

Small programs such as these illustrate the usefulness of having direct and convenient access to RAW files for quality control and data inspection purposes by mass spectrometrists. Furthermore, the convenient transition from existing examples to actual executables via the MS Playground makes it easy to start using the library. It should also be noted that the main intended use of our library is in the form of the described executables. This is because most programming languages and workflow tools in common use make it very easy to integrate executables in their operations.

We have here presented a simple, open-source interface to access Thermo Scientific RAW files. Using this library in the MS Playground can speed up tool generation and provides executables for these tools for all common platforms. Over time, we aim to keep the unthermo library and the MS Playground system up-to-date with newer RAW file versions and are currently working on incorporating RAW file from the Fusion instrument. Moreover, the MS Playground wiki allows code to be archived for provenance and to be exchanged and reused by others as well. As a result, our *unthermo* library and online playground provides mass spectrometrists and proteomics researchers with the ability to directly interface with the raw data acquired by their Thermo Scientific instruments.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jproteome.5b00778.

Thoroughly annotated copies of the relevant code (PDF)

DOI: 10.1021/acs.jproteome.5b00778
J. Proteome Res. 2015, 14, 4940−4943

AUTHOR INFORMATION

Corresponding Author

*E-mail: lennart.martens@vib-ugent.be. Tel.: +32-92649358. Fax: +32-92649484.

Author Contributions

∇P.K. and A.S.C.S. contributed equally to this work.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

P.K., A.S.C.S., A.A., K.L., D.V., and L.M. acknowledge support from an SBO grant, "InSPECtor" (120025), of the Flemish agency for Innovation by Science and Technology (IWT). We thank Gene Selkov for reverse-engineering the Thermo raw-data object and for making the corresponding code available as open-source, and Bernard Delanghe from Thermo Fisher Scientific.

ABBREVIATIONS

MS, mass spectrometry; URL, uniform resource locator

REFERENCES

(1) Rudnick, P. A.; Clauser, K. R.; Kilpatrick, L. E.; Tchekhovskoi, D. V.; Neta, P.; Blonder, N.; Billheimer, D. D.; Blackman, R. K.; Bunk, D. M.; Cardasis, H. L.; et al. Performance metrics for liquid chromatography-tandem mass spectrometry systems in proteomic analyses. *Mol. Cell. Proteomics* **2010**, *9*, 225−241.

(2) Vaudel, M.; Sickmann, A.; Martens, L. Peptide and protein quantification: a map of the minefield. *Proteomics* **2010**, *10*, 650−670.

(3) Martens, L.; Nesvizhskii, A. I.; Hermjakob, H.; Adamski, M.; Omenn, G. S.; Vandekerckhove, J.; Gevaert, K. Do we want our data raw? Including binary mass spectrometry data in public proteomics data repositories. *Proteomics* **2005**, *5*, 3501−3505.

(4) Chambers, M. C.; Maclean, B.; Burke, R.; Amodei, D.; Ruderman, D. L.; Neumann, S.; Gatto, L.; Fischer, B.; Pratt, B.; Egertson, J.; et al. A cross-platform toolkit for mass spectrometry and proteomics. *Nat. Biotechnol.* **2012**, *30*, 918−920.

(5) Amstadt, B Wine. *Linux J.* **1994**, *3*, 4es.

(6) Martens, L.; Chambers, M.; Sturm, M.; Kessner, D.; Levander, F.; Shofstahl, J.; Tang, W. H.; Rompp, A.; Neumann, S.; Pizarro, A. D.; et al. mzML - a community standard for mass spectrometry data. *Mol. Cell. Proteomics* **2011**, *10*, R110.000133.

(7) Kessner, D.; Chambers, M.; Burke, R.; Agus, D.; Mallick, P. ProteoWizard: Open Source Software for Rapid Proteomics Tools Development. *Bioinformatics* **2008**, *24*, 2534−2536.

(8) Perez-Riverol, Y.; Uszkoreit, J.-; Sanchez, A.; Ternent, T.; Del Toro, N.; Hermjakob, H.; Vizcaíno, J. A.; Wang, R. ms-data-core-api: an open-source, metadata-oriented library for computational proteomics. *Bioinformatics* **2015**, *31*, 2903−2905.

(9) Selkov, G. *unfinnigan: painless extraction of mass spectra from Thermo "raw" files.* http://code.google.com/p/unfinnigan/.

(10) Verheggen, K.; Barsnes, H.; Martens, L. Distributed computing and data storage in proteomics: many hands make light work, and a stronger memory. *Proteomics* **2014**, *14*, 367−377.

(11) Gerrand, A. *Inside the Go Playground.* http://blog.golang.org/playground.

(12) Köcher, T.; Pichler, P.; Schutzbier, M.; Stingl, C.; Kaul, A.; Teucher, N.; Hasenfuss, G.; Penninger, J. M.; Mechtler, K. High precision quantitative proteomics using iTRAQ on an LTQ orbitrap: a new mass spectrometric method combining the benefits of all. *Proteome Res.* **2009**, *8*, 4743−4752.

DOI: 10.1021/acs.jproteome.5b00778
J. Proteome Res. 2015, 14, 4940−4943