The evolution of data exchange in Mass Spectrometry spans decades and has ranged from human-readable text files representing individual scans or collections thereof (McDonald et al., 2004) through the official standard XML-based (Harold, Meuns, & Udemadu, 2005) data interchange standard (Deutsch, 2012), to increasingly compressed (Teleman et al., 2014) variants of this standard sometimes requiring purely binary adjunct files (Römpf et al., 2011). While the desire to maintain even partial human readability is understandable, the inherent mismatch between XML’s textual and irregular format relative to the numeric and highly regular nature of actual spectral data, along with the explosive growth in dataset scales and the resulting need for efficient (binary and indexed) access has led to a phenomenon referred to as “technical drift” (Davis, 2013). While the drift is being continuously corrected using adjunct formats, compression schemes, and programs (Röst et al., 2015), we propose that the future of Mass Spectrometry Exchange Formats lies in the continued reliance and development of the PSI-MS (Mayer et al., 2014) controlled vocabulary, along with an expedited shift to an alternative, thriving and well-supported ecosystem for scientific data-exchange, storage, and access in binary form, namely that of HDF5 (Koranne, 2011). Indeed, pioneering efforts to leverage this universal, binary, and hierarchical data-format have already been published (Wilhelm et al., 2012; Rübel et al., 2013) though they have under-utilized self-description, a key property shared by HDF5 and XML. We demonstrate that a straightforward usage of plain (“vanilla”) HDF5 yields immediate returns including, but not limited to, highly efficient data access, platform independent data viewers, a variety of libraries (Collette, 2014) for data retrieval and manipulation in many programming languages and remote data access through comprehensive RESTful data-servers.

© 2016 The Authors. Mass Spectrometry Reviews published by Wiley Periodicals, Inc. Mass Spec Rev 36:668–673, 2017

Keywords: mass spectrometry; data exchange formats; HDF5

I. INTRODUCTION

As a general rule, mass spectrometers produce output stored in manufacturer-specific proprietary file formats. A combination of the need to efficiently store ever larger data sets, regulatory requirements for tamper-proofing in the context of medical trials and the manufacturer’s tendency to lock data produced on their platform into an exclusive analytical ecosystem lead to a largely binary on-disk representation. Beyond the inherently reduced accessibility of proprietary binary files (which cannot be accessed without prior knowledge of their structure and/or dedicated tools) the formats and tools used are often burdened with requirements for backward compatibility bridging decades, as well as software dependencies precluding the use of vendor-provided tools on UNIX-like systems central to many high-performance data analytic pipelines.

In response to this situation, the proteomics community set about to define a controlled vocabulary for mass spectrometry data as well as a standard format for data interchange. It settled on the PSI-MS for the controlled vocabulary: expressed as a 17,000 Line OBO v1.2 file (Open Biomedical Ontologies [Smith et al., 2007]) this extremely detailed and increasingly comprehensive formal standard defines and inter-relates most key concepts in the field of mass-spectrometry and mass-spectrometry-based proteomics (e.g., MS:1000628 is the formal accession for a “basepeak chromatogram” which is a kind of MS:1000810, i.e., “mass chromatogram”). Any storage and interchange format aiming to become a standard must provide a mapping from its various data-fields to these concept-codes. The official data interchange format that was selected by the community, namely mzML (Deutsch, 2008), does indeed provide this mapping, and does so in every instance (i.e., in every individual mzXML file). This is because the PSI community desired the standard to be self-describing, as well as universally supported and human readable. Collectively, these requirements led to the choice of XML as the underlying representation technology for the mzML format.

II. SCALING BEYOND HUMAN READABILITY

Following the adoption of mzML by the mass spectrometry community, a gulf of ever increasing size has emerged between the performance characteristics of XML and the requirements of sharing and efficiently accessing data sets: software systems built around the XML-based format (as well as the related mzXML [Pedrioli et al., 2004] format) suffer from a mismatch between the data being stored and the storage format being used, with a resulting penalty in terms of file size and performance characteristics. In order to maintain the positive gains of the adopted format (in the form of desirable features such as universality and human readability), the developers of mzML compliant tools have had to engineer increasingly complex adjunct software in order to keep the resulting systems performant. Examples of this include the storage of image data in adjunct files external to the main mzML file in imzML (Römpf et al., 2011), the generation
of external binary files (".cachedMzML" files) by OpenMS to support efficient access to SWATH data stored in mzML (Röst et al., 2015), and the incorporation of the MS-Numpress (Teleman et al., 2014) compression schemes into ProteoWizard (Chambers et al., 2012) to mitigate the up to 18-fold inflation in mzML file size as compared to the original vendor format (Teleman et al., 2014). These software engineering fixes are all symptomatic of the technical drift arising from the choice of a text-based format for the storage of increasingly large volumes of mass spectrometry data.

III. HDF5 IS A NATURAL CHOICE FOR LARGE SCIENTIFIC DATASETS

The Hierarchical Data Format version 5 (HDF5) is an open source, binary file format for storing large and complex datasets, and is particularly suited for scientific and technical applications. The format supports extremely efficient access to and manipulation of complex and hierarchical data while maintaining universality both in terms of supported architectures, operating systems and programming languages. The format effectively offers facilities for an unlimited variety of data types, thanks in part to its self-describing nature. This in turn implies that computer systems accessing a file can adjust and optimize access patterns to the specific (potentially unique) data type encountered. Work on the format began at the NCSA in 1987 and has since been funded by the NSF, NASA, and the DOE. It has been heavily adopted by many US national labs including Lawrence Livermore, Los Alamos, and Sandia National Laboratories. Over the years, a thriving software ecosystem has emerged around the format including data viewers, programmer support, and parallel file processing to support massively parallel high performance computing environments.

IV. OpenMSI and mz5—HDF5 PIONEERS IN MASS SPECTROMETRY

The shortcomings of XML were perhaps most keenly felt in the domain of imaging mass spectrometry where even the first proposed standard for this sub-domain of mass spectrometry (imzML [Römpf et al., 2011]) abandoned the notion of a pure XML format. The raw data produced by these experiments is, in fact, never stored in XML but rather in a separate imaging binary data file (.ibd file). The overall format is still considered a variant of mzML since the metadata for the experiment is stored in "classic" mzML with empty binaryDataArray tags referring to the external "ibd" file. In response to this situation, and in order to better support parallel data access, the OpenMSI (Rübel et al., 2013) format was defined, which is based entirely on a single HDF5 file. The format leverages HDF5’s built-in gzip support and achieves excellent storage efficiency as well as support for both parallel read and write operations. The metadata and provenance information are also stored in the same unique HDF5 file without disrupting the data format thanks to the extensibility and self-describing nature of the format.

Another instance of HDF5 being used in mass-spectrometry was the incorporation of support for mz5 (Wilhelm et al., 2012) in ProteoWizard as an alternative storage model for mzML. The idea behind this format is that software which accesses raw mass spectrometry data through ProteoWizard can be made to use HDF5 behind the scenes without any change to the high-level application programming interfaces (APIs) used by the programmer. The benefit of this approach is a significant increase in storage and data access performance with no disruption to the sizable existing software base. Unfortunately, the proposed mz5 encoding of the raw data does not fully leverage the self-describing nature of HDF5. In pursuit of optimal performance the authors of the format adopted, for example, a delta mass representation, which, while yielding smaller data files, does reduce their accessibility by standard software/viewers (which rely solely on the data-typing explicitly declared in the file format). As a result, it is not possible to, for example, browse an mz5 data file using the standard HDFView data viewer (in contrast to a straightforward representation which is trivially accessible as seen in Fig. 1).

V. ANNOTATING HDF WITH THE PSI-MS CONTROLLED VOCABULARY (CV)

A positive aspect of mz5’s 1-to-1 compatibility with mzML is the reuse of PSI’s comprehensively curated controlled vocabulary for mass spectrometry (henceforth, PSI-MS CV). Every data element in mz5 is annotated with a PSI-MS CV accession, which gives it a definitive meaning, that can be referenced by humans and software alike. We hypothesize that a straightforward encoding of mass spectrometry data in HDF5 coupled with annotations using the PSI-MS CV can serve as a natural evolution away from XML as the underlying storage format in mass spectrometry. We have initiated a research program to assess this strategy by experimenting with a range of format designs all having in common a few key characteristics: (i) datasets must have self-explanatory names and be encoded in efficient yet standard representations of data (binary vectors of doubles wherever possible with names such as “m/z,” “intensity,” or “noise”); (ii) leverage yet another extremely useful feature of HDF5, namely the ability to associate attributes with datasets, in order to tag each dataset with the ID (“PSI-MS_ID”) and name (“PSI-MS_NAME”) of the CV term which best describes it; and (iii) in cases where PSI has not issued an appropriate controlled term, we fall back on a more general parent term which will apply while being necessarily over-generalized. For example, when storing noise information per centroid as provided by Thermo Fisher Scientific (Waltham, MA), we lack a term for noise array (not to be confused with the signal-to-noise array which does have an official accession, MS:1000517). In this situation we fall back on “binary data array” (MS:1000513).

The simplicity of the approach is most evident upon opening an HDF5 dataset in, for example, the standard and cross-platform tool HDFView (Fig. 1). It is immediately obvious what datasets are being provided and what their types are—this is because we store all the data in ragged-arrays that store each element type (e.g., m/z, intensity, charge state [z], noise etc.) across all the scans acquired in the dataset. For complex acquisitions (e.g., when attempting to identify small molecules via hierarchical product ion spectra scan types, aka scan trees) HDF5 may be used through another one of its features, namely the ability to create a hierarchy of groups—these can naturally represent the tree-like nature of many product ion spectrum (MS²) acquisitions. While in the regular format annotations with PSI-MS CV accessions may be made for all scans in a single...
non-redundant location, the hierarchical mode allows for independent annotation of every individual element, as in the mzML data model. Note that Shaduf, our HDF5 generating tool (described below) is able to generate both styles of HDF5 (the default being ragged mode, and the alternative being hierarchical mode).

VI. HDF5 SUPPORTS THE EVOLVING FUNCTION OF DATA INTERCHANGE FORMATS

Clearly, we are not yet in a position to propose a complete HDF5-based standard. We suggest that a period of trial and error should be considered, somewhat analogously to the current strategy of the PRIDE repository (Martens et al., 2005)—which today accepts a very broad range of data files where it initially mandated rather strictly which exact data formats should be uploaded. This tolerance to variation will serve the community well as, for example, we discover that certain data vectors are worth storing routinely (e.g., “signal to noise array,” MS:100517 has assumed much greater importance in quantitative Orbitrap experiments following the publication of observations by Makarov and Denisov (2009). Furthermore, as leading journals in the field restore the mandate to deposit all raw data in public repositories (Burlingame et al., 2015) and as repositories enable authors to upload raw vendor data, the function of the interchange data format may shift: authors will comply by uploading their raw vendor data files and use the standard interchange format to communicate a more useful subset of the underlying data (e.g., the authors may upload the full profile data in its original RAW file, while keeping only the vendor-defined centroids in the HDF5 file).

VII. PROOF OF CONCEPT IMPLEMENTATION: SHADUF

Shaduf is a python script that can export Thermo Fisher Scientific raw files into a ragged-array based HDF5 format for mass spectrometry data, without the need for ProteoWizard. The script

FIGURE 1. HDFView enables cross-platform access to a Shaduf generated HDF5 file.
requires Python including the NumPy and h5py extensions, as well as a copy of the MSFileReader (https://thermo.
flexnetoperations.com/control/thmo/download?element=6306677) software from Thermo Fisher Scientific. It is only approximately 500 lines long and offers the user the option to export full profile data or the vendor generated centroids (along with Noise and charge state estimation).

Shaduf emphasizes self-description over compression ratio, (e.g., by avoiding delta-encoding, employed by the designers of the mz5 format) and as a result the files it produces can be browsed easily using pre-existing HDF5 viewers, such as the platform-independent HDFView application, which is distributed by the HDF5 Group (https://www.hdfgroup.org/products/java/hdfview/). Figure 1 shows the ability to inspect individual peak entries along with the ability to observe scan filter data as well as the basepeak chromatogram. Note that the bpc dataset has an attribute named MS:1000628 with a value of “basepeak chromatogram.” In fact, the presence of a red “A” in the dataset icons represents the fact that attributes are present (which, in our case always corresponds to a PSI-MS CV entry being applied).

FIGURE 2. Accessing the same HDF5 file using both the Python and R scripting languages to produce a base peak chromatogram and an individual spectrum, respectively. The spectrum in question is an ms/ms spectrum of a doubly charged precursor ion at 858.92 m/z which corresponds to peptide DALSSVQESQVAQQR from Bovine Apolipoprotein C-III.

VIII. ACCESSING HDF5 DATA WITH STANDARD AND CUSTOM APIs

HDF5 is very well supported across programming languages of interest to the scientific community. The resulting ease of access...
to HDF5 files is demonstrated for Python (Ramalho, 2015) and R (R Core Team, 2016) with the following snippets, using the standard h5py (Collette, 2014) and h5 (Annau, 2016) Libraries to access and plot the base peak chromatogram and an individual spectrum contained in the file (Fig. 2).

Another example where HDF5 can be used with minimal overhead is in providing remote access to individual scans from large data files using h5serv (https://www.hdfgroup.org/projects/hdfserver/), a RESTful (Richardson & Ruby, 2007) API, and server implementation for remote access to HDF5 files (including the ability to remotely edit the file if necessary).

Remote browsing Shaduf generated HDF5 data files using an unmodified h5serv instance and visualizing the data (Fig. 3) with the Lorikeet (Sharma et al., 2012) spectral viewer (http://uwpr-github.io/Lorikeet/) required less than 150 Lines of JavaScript code to query the h5serv instance and pass the results to Lorikeet. This example demonstrates the benefit of optimizing for self-description: h5serv is able to provide individual spectra to the client because they are stored explicitly as a set of ragged arrays. Attempting a similar setup with mzML files (using, e.g., a JAX-RX (Graf, Lewandowski, & Grün, 2010) enabled server) would require additional server-side code to extract the actual spectral data from the opaque Base64 encoding scheme used to store it.

IX. CONCLUSION

We have explored a historic trend in mass-spectrometry data exchange from text-based representation of individual spectra to increasingly compressed and indexed formats favoring efficiency over human readability. In light of these trends we have highlighted the existing use of HDF5 as a means of preserving universality in a binary context and have demonstrated that a fully self-describing binary representation of mass spectrometry data can be achieved using plain (“vanilla”) HDF5 as the underlying storage format. A judicious annotation of the resulting datasets with attributes containing PSI-MS CV accessions yields a candidate interchange format that restores efficient access and storage, scalable up to, and including parallel data handling in HPC data centers. The examples provided, including data access through the standard HDFView file viewer, access from within python and R, as well as remotely through h5serv, all demonstrate that HDF5 constitutes a file format associated with a rich, standard-compliant computational ecosystem ready to be adopted as the data storage format of choice for the mass spectrometry community. We argue that while use of mzML and related XML-based formats is possible, a transition to simple (“vanilla”) HDF5 makes data access and sharing effortless, bringing into agreement the requirements of current mass

FIGURE 3. The Lorikeet spectral viewer connected to HDF5-JSON data provided by h5serv. The ms/ms spectrum being visualized is identical to the one accessed through an R script in the previous figure.
spectrumometry data, and storage technology and thus canceling (for a time) the effect of inevitable technical drift.

ACKNOWLEDGMENTS

We would like to acknowledge John Readey of the HDFGroup for helpful discussions regarding HDF5 in general and h5serv in particular.

We are grateful to Mario Annau, the author of the h5 package for R, for implementing access to ragged arrays and helpful discussion.

J.G., H.B.H., and the Proteomics Core at WCM-Q are supported by “Biomedical Research Program” funds at Weill Cornell Medicine – Qatar, a program funded by Qatar Foundation.

REFERENCES

Annau M. 2016. h5: Interface to the ‘HDF5’ library. https://CRAN.R-project.org/package=h5

Burlingame A, Carr SA, Bradshaw RA, Chalkley RJ. 2015. On credibility, clarity, and compliance. Mol Cell Proteomics 14:1731–1733.

Chambers MC, Maclean B, Burke R, Amodei D, Ruderman DL, Neumann S, Gatto L, Fischer F, Pratt B, Egerton J, Hoff K, Kessner D, Tasman N, Shulman N, Frewen B, Baker TA, Brusniak M, Paule C, Creasy D, Flashner L, Kani K, Moulding C, Seymour SL, Nuwaysir LM, Lefebvre B, Kuhlmann F, Roark J, Rainer P, Detlev S, Hemenway T, Huhmer A, Langridge J, Connolly B, Chadick T, Holly K, Eckels J, Deutsch EW, Moritz RL, Katz JE, Agus DB, MacCoss M, Tabb DL, Mallick P. 2012. A cross-platform toolkit for mass spectrometry and proteomics. Nat Biotechnol 30:918–920.

Collette A. 2014. Python and HDF5. Beijing: O’Reilly.

Davis N. 2013. Are you experiencing technical drift? Code Climate. http://blog.codeclimate.com/blog/2013/12/19/are-you-experiencing-technical-drift/

Deutsch E. 2008. MsML: A single, unifying data format for mass spectrometer output. Proteomics 8:2776–2777.

Deutsch EW. 2012. File formats commonly used in mass spectrometry proteomics. Mol Cell Proteomics MCP 11:1612–1621.

Graf S, Lewandowski L, Grün C. 2010. JAX-RX—Unified REST access to XML resources.

Harold ER, Means WS, Udemadu K. 2005. XML in a nutshell. 3. Auflage. Cologne, Germany: O’Reilly Verlag GmbH.

Koranne S. 2011. Handbook of open source tools. US: Springer. pp. 191–200.

Makarov A, Denisov E. 2009. Dynamics of ions of intact proteins in the Orbitrap mass analyzer. J Am Soc Mass Spectrom 20:1486–1495.

Martens L, Hermjakob H, Jones P, Adamski M, Taylor C, States D, Gevaert K, Vandekerckhove J, Apweiler R. 2005. PRIDE: The proteomics identifications database. Proteomics 5:3537–3545.

Mayer G, Jones AR, Binz P-A, Deutsch EW, Orchard S, Montecchi-Palazzi L, Vizcaíno JA, Hermjakob H, Oveiidero D, Julian R, Meyer HE, Eisenacher M. 2014. Controlled vocabularies and ontologies in proteomics: Overview, principles and practice. Biochim Biophys Acta 1844:98–107.

McDonald WH, Tabb DL, Sadygov RG, MacCoss MJ, Venable J, Grauman J, Johnson CR, Coccirova D, Yates JR. 2004. MS1, MS2, and SQT—Three unified, compact, and easily parsed file formats for the storage of shotgun proteomic spectra and identifications. Rapid Commun Mass Spectrom 18:2162–2168.

Pedrioli PGA, Eng JK, Hubyre R, Vogelzang M, Deutsch EW, Raught B, Pratt B, Nilsson E, Angeletti RH, Apweiler R, Cheung K, Costello CE, Hermjakob H, Huang S, Julian RK, Kapp E, McComb ME, Oliver SG, Omenn G, Paton NW, Simpson R, Smith R, Taylor CF, Zhu W, Aebersold R. 2004. A common open representation of mass spectrometry data and its application to proteomics research. Nat Biotech 22:1459–1466.

Ramalho L. Fluent python. 2015. Beijing [China]: O’Reilly.

R Core Team. 2016. R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/

Richardson L, Ruby S. 2007. RESTful web services. Farnham: O’Reilly.

Römmpp A, Schramm T, Hester A, Klinkert I, Both JP, Heeren RMA, Stöckl M, Spengler B. 2011. ImzML: Imaging Mass Spectrometry Markup Language: A common format for mass spectrometry imaging. Methods Mol Biol Clifton NJ 696:205–224.

Röst HL, Schmitt U, Aebersold R, Malmström L. 2015. Fast and efficient XML data access for next-generation mass spectrometry. PLoS ONE 10:e0125108.

Rübel O, Greiner A, Cholia S, Louie K, Bethel EW, Northen TR, Bowen BP. 2013. OpenMSI: A high-performance web-based platform for mass spectrometry imaging. Anal Chem 85:10354–10361.

Sharma V, Eng JK, Macoss MJ, Riffle M. 2012. A mass spectrometry proteomics data management platform. Mol Cell Proteomics MCP 11:824–831.

Smith B, Ashburner M, Rosse C, Bard J, Bag W, Ceusters W, Goldberg LJ, Eilbeck K, Ireland A, Mangalil CJ, OBI Consortium, Leontis N, Rocca-Serra P, Ruttenberg A, Sansone SA, Scheuermann RH, Shah N, Whetzel PL, Lewis S. 2007. The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol 25:1251–1255.

Teleman J, Dowsey AW, Gonzalez-Galarza FF, Perkins S, Pratt B, Röst HL, Malmström L. 2014. Numerical compression schemes for proteomics mass spectrometry data. Mol Cell Proteomics MCP 13:1537–1542.

Wilhelm M, Kirchner M, Steen JAJ, Steen H. 2012. Mz5: Space- and time-efficient storage of mass spectrometry data sets. Mol Cell Proteomics MCP 11:O111.011379.