

A common open representation of mass spectrometry data and its application to proteomics research

Patrick G A Pedrioli¹, Jimmy K Eng¹, Robert Hubley¹, Mathijs Vogelzang¹, Eric W Deutsch¹, Brian Raught¹, Brian Pratt², Erik Nilsson², Ruth H Angeletti³, Rolf Apweiler⁴, Kei Cheung⁵, Catherine E Costello⁶, Henning Hermjakob⁴, Sequin Huang⁶, Randall K Julian, Jr⁷, Eugene Kapp⁸, Mark E McComb⁶, Stephen G Oliver⁹, Gilbert Omenn¹⁰, Norman W Paton¹¹, Richard Simpson⁸, Richard Smith¹², Chris F Taylor⁴, Weimin Zhu⁴ & Ruedi Aebersold¹

A broad range of mass spectrometers are used in mass spectrometry (MS)-based proteomics research. Each type of instrument possesses a unique design, data system and performance specifications, resulting in strengths and weaknesses for different types of experiments. Unfortunately, the native binary data formats produced by each type of mass spectrometer also differ and are usually proprietary. The diverse, nontransparent nature of the data structure complicates the integration of new instruments into preexisting infrastructure, impedes the analysis, exchange, comparison and publication of results from different experiments and laboratories, and prevents the bioinformatics community from accessing data sets required for software development. Here, we introduce the 'mzXML' format, an open, generic XML (extensible markup language) representation of MS data. We have also developed an accompanying suite of supporting programs. We expect that this format will facilitate data management, interpretation and dissemination in proteomics research.

Proteomics research is supported by many different types of mass spectrometers, which differ in the manner in which data are acquired and stored¹. The modular and combinatorial assembly of different

components creates a wide range of MS instruments, each one with particular strengths and weaknesses for certain types of proteomics experiments. It is thus common to find more than one type of mass spectrometer in a typical proteomics laboratory. After acquisition, an MS instrument's output (from now on referred to as native data) is subjected to a number of analytical steps, commonly supported by proprietary programs supplied by the instrument manufacturer. Thus, in a laboratory environment that operates different types of mass spectrometers and multiple data analysis programs, it becomes extremely challenging to make meaningful comparisons of results obtained from different experiments or different instruments. This is a daunting issue for the proteomics community as a whole.

To alleviate this problem, we have developed the mzXML format, a common, open representation for mass spectrometric (MS), tandem mass spectrometric (MS/MS) or multiple mass spectrometric (MSⁿ) data, based on XML (Supplementary Notes online). Through the use of instrument-specific converters, the mzXML format provides a universal data interface between mass spectrometers and data analysis pipelines (Fig. 1). This approach eliminates the need to support multiple input formats in downstream data analysis software and significantly simplifies the integration of new mass spectrometers into an analysis framework. Furthermore, we expect the new format to greatly facilitate the exchange and publication of MS-based proteomics data and to provide a consistent platform for the development of new analytical tools. The flexibility of the format facilitates the introduction of new parameters as they become relevant to new experimental techniques. Finally, a public forum and a supervising committee enable the community to provide feedback and direct the evolution of the format.

Flexible stability

To keep pace with innovations in the MS and proteomics fields and to provide a platform stable enough for the development of related software tools, a common representation for MS data must achieve a delicate balance between flexibility and stability. The ANDI/netCDF format (ASTM E2078-00 'Standard Guide for Analytical Data Interchange Protocol for Mass Spectrometric Data') is one of the most successful attempts at creating a vendor-neutral MS data format. Unfortunately, because of its complex and rigid dictionaries, the format is difficult to keep up-to-date. As a result, ANDI is still unable to store MS/MS scans, the essence of most proteomics experiments.

¹Institute for Systems Biology, 1441 North 34th Street, Seattle, Washington 98103-8904 USA. ²Insilicos LLC, 4509 Interlake Avenue North, no. 223, Seattle, Washington 98103-6773, USA. ³Albert Einstein College of Medicine, LMAP Room 405, Ullman Bldg., 1300 Morris Park Avenue, Bronx, New York 10461 USA. ⁴EMBL Outstation European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK. ⁵Center for Medical Informatics, Department of Anesthesiology, Yale University School of Medicine, PO Box 208009, New Haven, Connecticut 06520, USA. ⁶Boston University School of Medicine, 715 Albany Street, R-806, Boston, Massachusetts 02118-2526, USA. ⁷Lilly Research Laboratories, One Lilly Corporate Center, Indianapolis, Indiana 46285, USA. ⁸Joint Proteomics Laboratory, Ludwig Institute For Cancer Research & The Walter and Eliza Hall Institute of Medical Research, Royal Melbourne Hospital, Parkville, Victoria, Australia 3050. ⁹School of Biological Sciences, University of Manchester, The Michael Smith Building, Oxford Road, Manchester M13 9PT, UK. ¹⁰The University of Michigan Medical School, 1150 W. Medical Center Drive, Ann Arbor, Michigan 48109-0656, USA. ¹¹Department of Computer Science, University of Manchester, Oxford Road, Manchester, M13 9PL, UK. ¹²Biological Sciences Division and Environmental Molecular Sciences Laboratory, Pacific Northwest National Laboratory, PO Box 999, Richland, Washington 99352, USA. Correspondence should be addressed to R.A. (raebersold@systemsbiology.org).

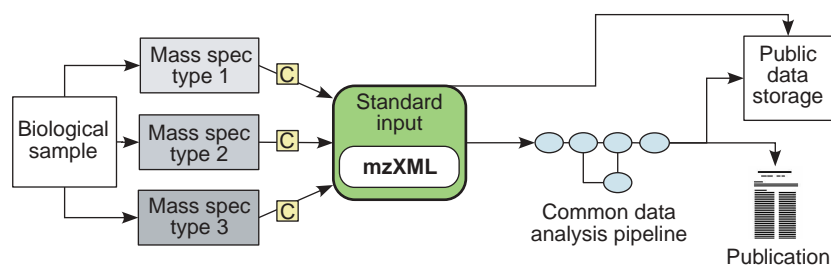


Figure 1 The mzXML file acts as a mediator, allowing multiple input formats to be subjected to a common data analysis pipeline. New types of instruments can be integrated into a preexisting analysis framework with only a utility (here represented by C) to convert MS native output to the mzXML format. The open structure of mzXML instance documents makes them suitable for data exchange such that, for example, they may be submitted to a data repository to support the results presented in a publication.

To provide a way to incorporate new types of data as they become relevant, particular attention has been given to the flexibility of the mzXML format. It has been designed with XML, which is by definition an extensible language and provides a way to define optional as well as required content. This intrinsic flexibility has further been complemented with the use of name-value-type elements. Information that is more likely to require regular updates (e.g., the list of MS instrument models) have been incorporated in an external ontology instead of enumerating them into the mzXML schema. Finally, the supporting software is released under an open source license agreement, to facilitate its dissemination, modification and upgrades.

Although it is desirable to maintain a certain level of flexibility, too much freedom can lead to the formation of 'dialects,' as has occurred with microarray gene expression (MAGE) suite². Although difficult to prevent entirely, dialects reduce the usefulness of a common format and should be minimized. It is therefore important that different groups working on a common representation communicate and work under the supervision of an advisory board. For the mzXML format, this has been addressed by providing users with an online public forum (located at http://sourceforge.net/forum/forum.php?forum_id=235607) to exchange opinions and provide feedback on the format, and the formation of the mzXML-associated standard solutions (MASS) committee.

The MASS committee will be responsible for (i) adapting the mzXML schema to address the needs of the proteomics community (ii) ensuring that the mzXML format stays up-to-date with the

innovations in the MS and proteomics fields and (iii) coordinating the efforts of groups working on different applications of the mzXML format. The committee will also interact and share resources with other groups working on the standardization of proteomics data. For instance, significant input has been given to the Proteomics Standard Initiative (PSI) group working on the MS region of the PSI Object Model. This will continue in the process of further developing the standard. Within the scope of the mzXML format, we will also provide export functions into the PSI standard format once it has stabilized (Supplementary Fig. 1 online).

Optimizing XML for MS data representation

XML's advantages of portability and extensibility make it increasingly popular for the representation of an enormous variety of data types, yet XML has some limitations compared with binary file formats. Of particular importance to us was the potential for greatly increased file size and reduced speed of access to the information represented in XML. The following describes how these limitations were addressed.

Modern mass spectrometers can generate over a gigabyte of compressed binary data per hour. However, XML cannot directly incorporate binary data and the conversion to a human readable clear text representation is not possible without a significant size increase (e.g., to represent a 4-byte floating point number at full precision at least 13 bytes are required; that is, +1.234567E-01). A more efficient solution is highly desirable, especially considering that for drug developers to comply with the US Food and Drug Administration (Rockville, MD, USA) 21 Code of Federal Regulation Part 11, native data should not be deleted, even after having been converted to the mzXML format.

This problem is addressed in the mzXML format by encoding the mass/charge (m/z) intensity binary pairs in base64 (rfc1341; <http://www.faqs.org/rfcs/rfc1341.html>). Base64-encoded binary data are somewhat larger (1.3-fold) than a binary representation. However, unlike binary data, they can be integrated into an XML file. For some mass spectrometer models, including the Q-TOF Ultima (Waters, Beverly, MA, USA) and the LCQ Classic (ThermoFinnigan, Waltham, MA, USA), it is also possible to reduce the file size by removing all

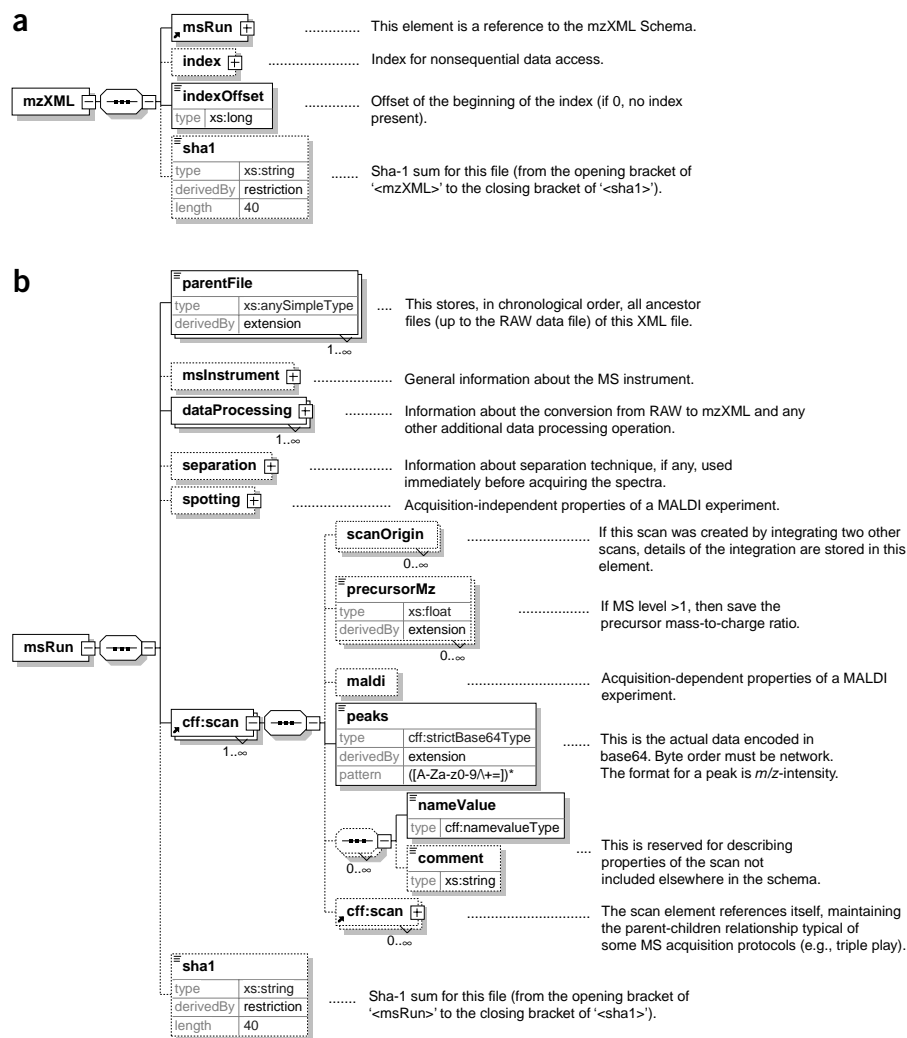
peaks with intensity equal to zero. The effects of these operations on file size are indicated in Table 1. With these simple strategies, information from large native binary files may be saved as text format files of manageable size.

The second limitation of XML as an operational format for the representation of MS data is a consequence of some XML parsers (e.g., SAX; <http://www.saxproject.org/>) that read a document sequentially, from the beginning of the file to the end. Applications requiring nonsequential access to data, such as a peptide quantification strategy based on stable isotope labeled-substrates (e.g., ASAPRatio³), would have unacceptable performance using purely sequential data access. We obviated this problem by creating a second schema that wraps the mzXML schema

Table 1 Data size comparison

Sample ^a	Native data ^b (Mb)	Clear text ^c		Base 64 ^c	
		All ^d (Mb)	Omit peaks with zero intensity ^d (Mb)	All ^d (Mb)	Omit peaks with zero intensity ^d (Mb)
A	3.8	15	8.1	5.9	3.1
B	234	816	30	312	13
C	20	60	60	26	26
D	14	45	45	18	18
E	151	1,100	681	397	243
F	53.4	1,000	1,000	390.3	390.3

^aSamples A and B were acquired in profile mode on a Q-TOF Ultima (Waters) instrument; samples C and D were acquired in centroid mode on a LCQ Classic (ThermoFinnigan) instrument; sample E was acquired in profile mode on an LCQ Classic (ThermoFinnigan) instrument; sample F was acquired in profile mode on a Qstar (ABI/Sciex) instrument. ^bNative refers to the size of the original binary native file created by the mass spectrometer. ^cClear text and Base64 refer to the size of mzXML instance documents translated from the native file and with the m/z intensities pairs stored as clear text or Base64-encoded binary data, respectively. ^dAll peaks included in data or peaks of zero intensity were omitted.

**Figure 2** Overview of the mzXML format.

(a) Schema for the indexed mzXML format. XML parsers such as the SAX parser can only read a document sequentially from the beginning to the end. Other parsers avoid this problem by reading the entire XML file into memory, then allowing data to be accessed in any order. Given the amount of data from a typical LC-MS/MS experiment none of these scenarios is desirable. For example, consider a situation in which data from scan 5,000 must be read and analyzed before it can be determined that scan 4,999 should be read next. A typical XML reader implementation would have to read the first 5000 scans, then return to the beginning of the file and read the first 4,999 scans. In the mzXML format this problem is addressed by indexing the position of each scan in the document. (b) Overview of the mzXML schema version 2.0. This version is compatible with LC-ESI-MSⁿ and with MALDI-MSⁿ experiments. In creating the mzXML format, we have tried to address some of the needs of an operational format, as well as those of a machine-independent data representation. For instance, we have allowed a certain degree of redundancy to simplify the design and improve the performance of programs that process mzXML documents. As an example, to allow the rapid reconstruction of a base peak chromatogram, the scan element has attributes to describe the *m/z* and the intensity values of the base peak, despite the fact that these could both be derived from the content of the peaks element. For the sake of clarity attributes have been omitted from the representation. A full version of the schema and a detailed description of the mzXML structure is available on the project homepage (http://sashimi.sourceforge.net; http://sashimi.sourceforge.net/schema_revision/mzXML_2.0/Doc/mzXML_2.0_tutorial.pdf and http://sashimi.sourceforge.net/schema_revision/mzXML_2.0/Doc/mzXML_2.0.html). The principal elements are described in the main text in the section 'Structure of the mzXML format.'

and indexes the position of each scan in a given XML file (Fig. 2a). At parsing time, this index can be used to adjust the input stream to a scan-specific offset (Supplementary Fig. 2 online).

We tested this strategy by comparing access times on the same data set represented both in a ThermoFinnigan binary format and in the mzXML format (see Supplementary Methods online). In a first test (Table 2), *m/z* intensity pairs were read for two full liquid chromatography (LC)-MS/MS runs of different sizes, starting from the last scan and moving toward the first. The average time to read a scan was less than a millisecond, with a slight (0.2 ms versus 0.3 ms) speed advantage for the data represented in the mzXML format. In a second test, we measured the times required by XPRESS⁴ to calculate relative isotopic quantification of peptides detected in an LC-MS/MS run, again using both a ThermoFinnigan native binary and the mzXML formats. The data in Table 3 indicate an approximately twofold speed advantage in favor of the mzXML representation. These tests thus demonstrate

that for a prominent binary MS format, after restoring random access with a simple indexing technique, no performance penalty is incurred in accessing data stored in XML as compared with data stored in a binary format.

Structure of the mzXML schema

Version 2 of the mzXML schema accounts for the most common applications of MS-based proteomics research: database searching,

Table 2 Nonsequential access times for binary and mzXML formats

Sample ^a	Total number of scans	Binary ^b		mzXML ^b	
		Total time (s)	Time/scan (ms)	Total time (s)	Time/scan (ms)
A	2,254	0.59	0.3	0.42 (71%) ^c	0.2
B	9,358	2.59	0.3	1.60 (62%) ^c	0.2

^aData for samples A and B were acquired on a ThermoFinnigan LCQ instrument. ^bThe times to read all the peaks starting from the last scan and moving back to the first one are shown for a native binary and an mzXML file. The binary files were read with an application based on ThermoFinnigan's proprietary file access libraries. An application based on RAMP was used to access the XML files. ^cThe numbers in parentheses indicate the percentage of time used for accessing data in mzXML relative to the time used for accessing data stored in the binary format.

Table 3 Quantification times for binary and mzXML datasets

Sample ^a	Binary ^b time (s)	mzXML ^b time (s)
A	48.4	23.8 (49%) ^c
B	79.0	37.2 (47%) ^c

^aSamples A and B were isotope-coded affinity tag (ICAT)-labeled and analyzed on a ThermoFinnigan LCQ mass spectrometer. ^bThe times to calculate relative quantification for the isotopic pairs identified in the database search step are shown for native binary and mzXML formats. The binary files were quantified using the XPRESS version based on ThermoFinnigan's proprietary libraries, whereas XML files were quantified using the RAMP-based version of XPRESS. ^cThe numbers in parentheses indicate the percentage of time used for calculating quantities from data in mzXML relative to the time used for data stored in the binary format.

de-novo sequencing, quantification using stable isotopic labeling and quantification of LC-MS traces. The following is an overview of the schema elements shown in Figure 2b.

The 'parentFile' element stores a chronological list of all files used to generate a given instance document. For example, if a native data file is converted to a first mzXML document, from which a second mzXML file is created, the second mzXML document will have two 'parentFile' elements. The first 'parentFile' element will represent the universal resource identifier (URI) of the native data and the second 'parentFile' element will represent the URI of the first mzXML document. Each URI is also associated with a sha1-sum (a digital signature generated by a secure hash algorithm; SHA1 version 1.0; http://www.w3.org/PICS/DSig/SHA1_1_0.html).

The 'msInstrument' element stores the specifications of the MS instrument (e.g., resolution, manufacturer, model, ionization type, mass analyzer type, detector type) and acquisition software used to generate the data. A 'nameValue' element (see 'nameValue' element under scan element) provides a means to store laboratory-specific instrument modifications. Even in a vendor-neutral representation, it is important to preserve this information because the analytical software should account for the strengths and weaknesses of different instruments.

The 'dataProcessing' element describes any type of data processing (e.g., centroiding, noise reduction, peak finding) performed during the creation of the current instance document. The description will include the name and version of the program used, a list of the input parameters and a comment field, which can reference a publication illustrating the processing algorithm.

We felt compelled to add a 'separation' element because, although the mzXML format only represents information generated by mass spectrometers, some MS applications are tightly coupled to a separation technique (e.g., online microcapillary liquid chromatography mass spectrometry). Because in a strict sense this is outside the scope of the mzXML format, the separation element has not been developed, but acts only as a placeholder (creating variable content container elements; <http://www.xfront.com/BestPracticesHomepage.html>) for connecting an additional XML schema describing a separation technique. A simple example on how to implement such a schema for a liquid chromatography separation system can be found at the project homepage (<http://sashimi.sourceforge.net/>).

The 'spotting' element stores those characteristics of matrix-assisted laser desorption/ionization (MALDI) experiments that are constant for each acquisition, such as the matrix composition, the plate type and geometry, and the spotting robot model used, if applicable.

The 'scan' element has attributes to describe, among others, the retention time, the MS level, the polarity of the ion source, the ionization energy and the mode of acquisition (e.g., full, selected ion

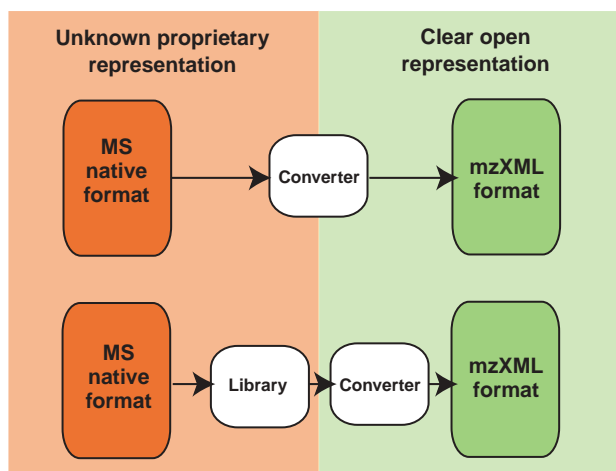


Figure 3 Although most mass spectrometers are capable of exporting data in formats recognized by sequence search engines (e.g., .dta for SEQUEST, .mgf for Mascot and .pkl for ProteinLynx), other data analysis operations, such as peptide quantification after stable isotope labeling, are not supported by these formats. The complete information required for such operations can be extracted from the native outputs using one of the following three strategies: first, reverse engineering the structure of the native output; second, obtaining, most likely under a nondisclosure agreement, the specifications for the structure of the native output; or third, using functions from a data access library provided by the vendor. These functions automatically retrieve specific information from a native format so that the programmer does not need to know the actual structure of the format being read. Programs developed using this strategy are easier to redistribute, easier to maintain, more robust against upgrades in the native format and shift the responsibility for guaranteeing the integrity of the data conversion from the developer of the analytical tool to the instrument vendor. The main disadvantage of this approach is that instrument manufacturers typically provide interface code only for the computer environment that their instrument controllers use. Consequently, it is impossible to make a single analytical program that works with data from instruments that use different computer (hardware or operating system) environments. This problem can easily be addressed by converting the native data into a platform-independent intermediary format. This intermediary format can then be accessed by the various components of a software analysis pipeline in a vendor-neutral way.

monitoring, selected reaction monitoring) for the scan being described. The 'scan' element contains a reference to itself. This provides an intuitive way to store scans sharing a common ancestor (e.g., a common survey scan). The scan element possesses seven subelements: the 'scanOrigin' element, the 'precursorMz' element, the 'maldi' element, the 'peaks' element, the 'nameValue' element and the 'comment' element.

The 'scanOrigin' subelement stores the details of the integration process if the current scan has been created by merging multiple scans. The 'precursorMz' subelement stores the *m/z*, intensity, charge state, width of the selection window and collision energy values for the precursor ion fragmented in the current scan. Multiple instances of the 'precursorMz' subelement per scan element can be included to account for fragmentation spectra possessing more than one precursor ion (e.g., as in shotgun sequencing experiments with fragments generated by in-source decay⁵). The 'maldi' subelement stores those parts of data from a MALDI experiment that can vary between multiple scans acquired on the same spot (e.g., the laser intensity or the duration of the laser excitation). The 'peaks' subelement contains the *m/z* intensity pairs as base64-encoded binary data. This element can

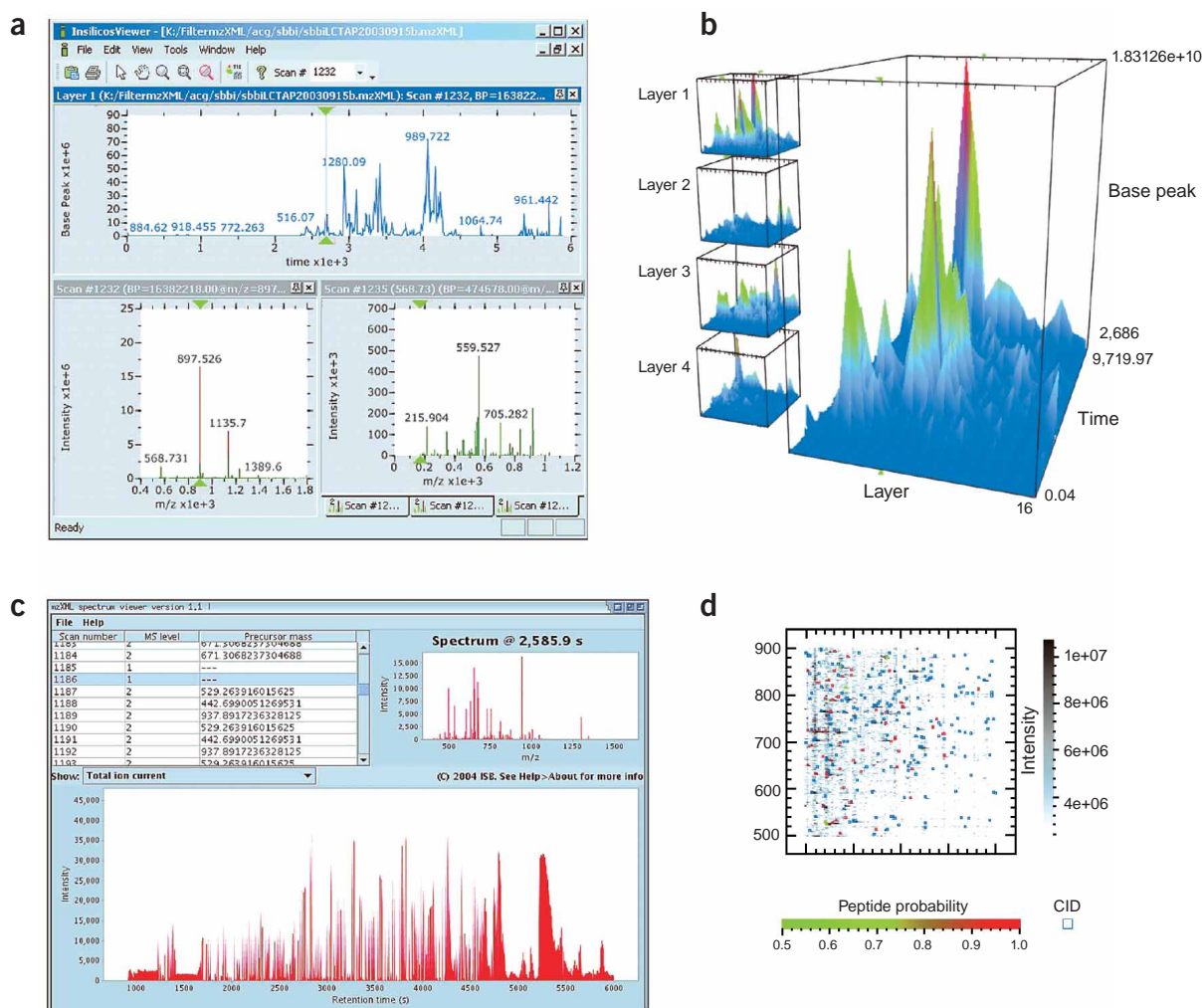


Figure 4 Various methods for visualization of data contained in mzXML documents. (a) The figure shows different displays supported by InsilicosViewer for mzXML files. It provides most of the features offered by viewers in commercial software packages, including visualization of base peak chromatographs (top panel), MS survey spectra (lower left panel) and MS/MS spectra (lower right panel). (b) InsilicosViewer mzXML overview window. This window can be particularly helpful in experiments where one sample has been split into multiple fractions that are sequentially analyzed, as when a two-dimensional (e.g., ion exchange and reverse phase) peptide fractionation is applied. In this case, four different proteomics experiments are shown. Each experiment consists of 16 mzXML files; each mzXML file contains ~6,000 spectra. Consequently, well over 300,000 spectra are represented. The indexing scheme of the mzXML format allows any spectrum in this massive collection to be quickly accessed. In the case of InsilicosViewer, this requires only a few mouse clicks. (c) The mzXML viewer main window contains a list of all scans in an mzXML instance document (top left panel), a graphical representation of the currently selected scan (top right panel) and the total ion current (lower panel). (d) A two-dimensional density plot created by Pep3D, an in-house written software tool designed to monitor the performance of LC-MS/MS experiments. Specific areas of the plot are contoured by colored boxes to indicate a collision induced dissociation (CID) attempt (blue) or the probability assigned by PeptideProphet to a particular MS/MS scan (green to red gradient).

store raw as well as processed m/z intensity pairs. The 'nameValue' subelement is an idea borrowed from the MAGE language. It provides an extensible content model to the scan subelement (creating extensible content models; <http://www.xfront.com/BestPracticesHomepage.html>) and has three optional attributes: name, value and type. The 'nameValue' subelement can be used to add entries to the instance document without having to change the schema. This allows different laboratories to have personalized instance documents, while referring to a centralized common schema. For example, the temperature of the heated capillary of an electrospray instrument could be stored in the following way:

```
<nameValue name='heatedCapillaryTemperature'
value='203.4'
```

```
type='Celsius'/>
```

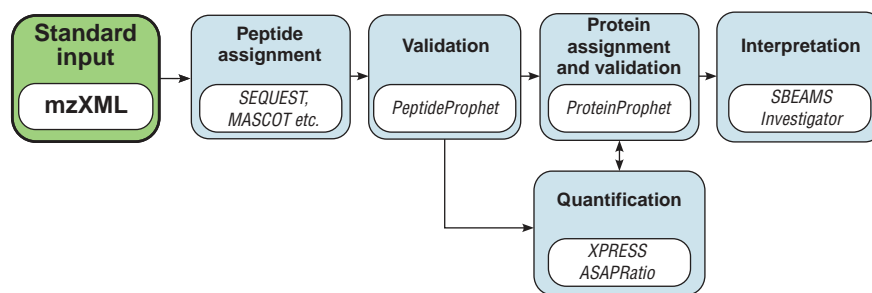
The 'comment' subelement describes in more detail, if needed, the content of the 'nameValue' subelement.

The 'sha1' element contains a sha1-sum that is calculated for the current instance document. This is a unique identifier that will change if a single bit of the file is modified. It provides a way to determine if the data have been corrupted.

The mzXML toolbox

To encourage the adoption of the mzXML format and further development of related analytical software, we have also created a series of basic software tools. All of these programs, except for the InsilicosViewer, are released under an open source license and can be

Figure 5 Role of the mzXML format in an analysis framework. Schematic representation of the Institute for Systems Biology Common Proteomics Data Analysis Pipeline. To facilitate the comparison of results generated on different instruments, a common analysis pipeline has been developed. The tasks carried out by the programs are indicated with blue backgrounds and specific implementations for carrying out each task are indicated with white backgrounds. The Institute for Systems Biology pipeline consists of the following: first, a suite of sequence database search tools, such as SEQUEST⁶, Mascot⁷ and Probid¹⁴, which assign a peptide sequence to each fragment ion spectrum; second, a tool (PeptideProphet¹⁰) that calculates an accurate probability that a peptide sequence has been correctly assigned to a fragment ion spectrum; third, a tool (ProteinProphet¹¹) that calculates an accurate probability that a protein has been identified from the available peptide identities; fourth, tools (ASAPRatio³, XPRESS⁴) that use the intensities of differentially isotopically labeled peaks⁹ representing the same peptide sequence to determine the abundance ratios for identified proteins; and fifth, tools (Cytoscape¹⁵ and Systems Biology Experiment Analysis Management System (SBEAMS); <http://www.sbeams.org>) assisting in the organization, visualization and biological interpretation of the data.



downloaded from the project homepage (<http://sashimi.sourceforge.net/>).

The first group of programs converts the MS native data into the mzXML format. Currently, converters are available for Thermo-Finnigan's Xcalibur v.1.4, Waters' MassLynx v.4.0, MDS Sciex (Toronto)/ABI's (Foster City, CA, USA) AnalystOS v.1.1 and Bruker's (Billerica, MA, USA) microTOF analysis.baf. All of these converters make use of libraries provided by the instrument vendors (Fig. 3).

The second group of programs performs the opposite task, providing analytical software with a way to access the information stored in an mzXML file. A standard XML parser may be used, but if

fast, random access to the information is required, an index-aware parser should instead be used. We have developed two such parsers, random access parser (RAP) and random access minimal parser (RAMP). The use of a parser requires modifying the source code of the analytical program, which is not always possible (in the case of proprietary source software, such as SEQUEST⁶, Mascot⁷, ProID) or convenient. We therefore wrote mzXML2Other, an application that expands the horizon of possible applications for the mzXML format by translating it into the input formats accepted by proprietary software tools. Currently, mzXML2Other can create input files for SEQUEST (.dta), Mascot (.mgf), and ProteinLynx (.pkl), in addition to a tab-delimited text format.

It is very important, especially for a file that is likely to be transmitted between different instruments, to be able to confirm the integrity of its content at any time. The validateXML program performs two types of validation on an mzXML file. First, the document is validated against its schema. Second, the sha1-sum for the document is generated and compared against the value stored in the sha1 element to determine if the file was corrupted. These two validation steps assure that data in an mzXML file conform to the document grammar and that the content of the file is exactly the same as when it was generated. At a higher level, we have used RAMP to port XPRESS and ASAPRatio, two proteomics tools for the quantification of differentially labeled samples, to directly read mzXML files.

Finally, we have developed several viewers for mzXML files. The mzXMLViewer and the InsilicosViewer (Fig. 4a-c) support the most

Swiss-Prot accession number	SEQUEST			MASCOT			PROTEINLYNX		
	SEQUEST	mzXML	centroid	SEQUEST	mzXML	centroid	PROTEINLYNX	mzXML	centroid
P02768	N	P	C	N	P	C	N	P	C
P00921	N	P	C	N	P	C	N	P	C
Q29443	N	P	C	N	P	C	N	P	C
P00489	N	P	C	N	P	C	N	P	C
P00722	N	P	C	N	P	C	N	P	C
P00432	N	P	C	N	P	C	N	P	C
P02562	N	P	C	N	P	C	N	P	C
P00634	N	P	C	N	P	C	N	P	C
P02769	N	P	C	N	P	C	N	P	C
P06278	N	P	C	N	P	C			
P02666	N	P	C						
P01012	N	P	C						C
P02754	N	P	C				N	P	C
P46406									
P02188									
P03996									
P00711									

N Native
 P mzXML profile
 C mzXML centroid

Figure 6 Standard versus mzXML database search results. A 17-protein mixture was analyzed on a Q-TOF Ultima (Waters) instrument and searched using SEQUEST, Mascot and ProteinLynx. Positive identifications are symbolized by an N, P or C character. The native data were converted to the appropriate search formats either directly using an application from Waters (N), or by using the mzXML format as an intermediary. The same proteins were identified when searching directly or with mzXML (P) as an intermediary, demonstrating that the mzXML format is a faithful representation of the raw data. The same proteins (plus an additional one in the case of ProteinLynx) were also identified from the centroided data in the mzXML (C) format, showing that the format is also compatible with processed data. Protein references are given as accession numbers from the Swiss-Prot database. The cutoffs for SEQUEST, MASCOT and ProteinLynx were set at a ProteinProphet probability cutoff of 0.9, at the significance threshold and at a protein score of 100, respectively. Therefore, only protein identification resulting from the same search program should be compared.



common features offered by viewers in commercial software packages. These viewers can display various properties of an LC-MS run, such as the total ion current (TIC), the base peak chromatogram (BPC) and MSⁿ spectra. InsilicosViewer features scripting capability using the Python language and can load and simultaneously visualize multiple files. A third viewer, Pep3D⁸ (Fig. 4d), visualizes data using an interactive two-dimensional density plot and indicates peptide identification confidence using a color gradient. The viewers provide an instrument-neutral way to interpret the results of a MS analysis and allow data from one instrument to be displayed in the absence of the corresponding acquisition software, thereby facilitating the exchange of information.

Interestingly, some third-party applications with support for the mzXML format have also started to emerge, such as an application (Sequin, H. *et al.* MPE-053. 52nd American Society for Mass Spectrometry Conference, Nashville, TN, USA, May 23–27, 2004) for user-friendly submission of data sets from multiple MS instruments to disparate search engines.

Road test for the mzXML format

To test the robustness of the programs presented in the previous section, we used them to port the proteomic data analysis pipeline developed at the Institute for Systems Biology (Seattle, WA, USA) (Fig. 5) to directly read mzXML-formatted documents. The pipeline was then used to test the performance of the mzXML format in a series of standard proteomics experiments.

A 17-protein mixture was digested with trypsin and analyzed on a Q-TOF Ultima (Waters) instrument (see **Supplementary Methods** online). Native profile data were converted either directly to .pkl and .dta files using an application from Waters or to the mzXML format first, then to .pkl, .dta and .mgf using mzXML2Other. The spectra were searched using SEQUEST, Mascot and ProteinLynx. Results are summarized in **Figure 6**. There is a complete agreement in the proteins identified by searching with or without the mzXML format as an intermediary, confirming that the mzXML format faithfully represents native raw data.

To test the ability of the mzXML format to represent processed data, the same native data was converted to the mzXML format, but this time the *m/z* intensity pairs were centroided. Searching these data with SEQUEST, Mascot and ProteinLynx yielded at least the same number of identifications seen when the data were searched without any processing, demonstrating that the mzXML format is also compatible with processed data.

In an additional experiment, a simple mixture consisting of seven proteins was labeled on cysteine residues with isotopically heavy or light isotope-coded affinity tag (ICAT) reagents and digested with trypsin (see **Supplementary Methods** online). The resulting peptides were analyzed on an LCQ Deca (ThermoFinnigan) instrument. All operations were carried out using standard protocols as described in ref. 9. After generating .dta files using lcq_dta on the binary data and mzXML2Other on the mzXML representation, peptide sequences were assigned to the spectra using SEQUEST. The probability of each assignment was then evaluated with PeptideProphet¹⁰ and signal pairs

Table 4 Standard versus mzXML quantitative proteomics experiment^a

Protein ^b	Protein probability	Percent coverage	XPRESS ratio mean	XPRESS s.d.	XPRESS no. of peptides	No. of unique peptides	Total no. of peptides
CATA_BOVIN	0.99	6.3	0.78	0	1	1	1
	0.99	6.3	0.78	0	1	1	1
PHS2_RABIT	1	8.8	0.9	0.38	7	8	9
	1	8.8	0.9	0.38	7	8	9
LCA_BOVIN	1	63.4	1.41	1.6	41	27	43
	1	54.9	1.42	1.62	40	25	41
OVAL_CHICK	1	10.4	0.79	0.21	29	11	30
	1	10.4	0.79	0.2	30	13	32
MYG_HORSE	1	9.2	NA	NA	0	2	3
	1	9.2	NA	NA	0	2	3
ALBU_BOVIN	1	52.9	0.83	0.21	100	68	105
	1	52.9	0.83	0.21	100	68	105
TRFE_BOVIN	1	21.4	1.07	0.23	42	26	42
	1	21.4	1.11	0.36	44	27	46

^aThe output of ProteinProphet is shown for an analysis performed on the same data starting from a ThermoFinnigan binary file (upper rows) or from an equivalent mzXML file (lower rows). In both cases the same seven proteins were identified and assigned a probability score higher than 0.9. The small variations observed at the level of the quantification and numbers of identified peptides are a consequence of the different charge state determination algorithms used during the creation of the .dta files. ^bProtein names are given as entry names from the Swiss-Prot database. For further details see **Supplementary Methods** online. s.d., standard deviation.

representing isotopically labeled peptides were quantified with the traditional version of XPRESS or with the RAMP-based version of the same program. Finally, individual probabilities for each peptide were combined to derive protein probabilities using ProteinProphet¹¹.

Comparison of the outputs at the peptide level shows that corresponding spectra were assigned the same identifications, scores and quantifications. At the protein level (**Table 4**) ProteinProphet assigned a probability higher than 0.9 to all seven proteins, irrespective of the format used to create the .dta files. The only variations observed were a consequence of the different algorithms used for the determination of the charge state by lcq_dta and mzXML2Other. This study thus demonstrates that for quantitative proteomics experiments, the mzXML format is a valid intermediary between the native format and the integrated LC-MS/MS software analysis pipeline by representing all the information required for protein identification and quantification.

DISCUSSION

We expect that the proteomics community as a whole will benefit from the adoption of an open data representation format because this is a necessary prerequisite to address a key issue in MS-based proteomics: the question of how the assertion of a positive protein identification in the literature can be confirmed by readers of the publication¹². Because of the proprietary nature of native acquisition formats, papers have typically been published without being accompanied by the actual data they are presenting. The mzXML format can now be used to represent the data associated with a proteomics publication in an openly accessible format. By providing a common 'language' for multiple mass spectrometers and a set of free tools to work with this language, the mzXML format will also facilitate the exchange of data sets, promoting collaboration and the creation of public data repositories¹³.

For instance, as a special experiment within the Human Proteome Organization (HUPO) Plasma Proteome Project (the analysis of collaborative studies of plasma or tissue lysates involving very large numbers of proteins with multiple post-translational modifications and

with extreme variation in concentrations), the Institute for Systems Biology group has successfully undertaken independent analysis of the raw MS data with the many tools presented in this paper. These diverse data came from multiple instruments and multiple laboratories. The HUPO Plasma Proteome Project data set has very large numbers of peptide sequences, leading to thousands of reported protein identifications by matching via diverse search engines to diverse protein and gene databases (reported at the HUPO Plasma Proteome Project Jamboree Workshop, June 1–4, 2004; <http://www.hupo.org/hpp/hppp.htm>).

Unrestricted access to MS data is also of particular relevance for the bioinformatics community, who will now be in a position to create new analytical software using data represented in an open file format, thereby enhancing the data analysis capabilities of the proteomics community and supporting the design of novel types of experiments. With conversion to a peaklist, many important parameters (e.g., peak width, peak shape and noise level) are lost. These data are essential for advanced data analysis. For this reason, the mzXML format can represent raw or processed data, offering researchers not directly working in the proteomics field a way to access all the information required for developing data manipulation or mining algorithms (e.g., noise reduction, peak detection, charge state deconvolution), topics that are posing fundamental new questions for statisticians and computer scientists. In combination, the features of the mzXML format are expected to drive progress and add credibility and standardization to MS-based proteomics.

Note: Supplementary information is available on the Nature Biotechnology website.

ACKNOWLEDGMENTS

This project was funded in part by federal funds from the National Heart, Lung, and Blood Institute, National Institutes of Health, under contract no. N01-HV-28179 and by grant no. 1R33CA93302 from the National Cancer Institute. The Institute for Systems Biology is supported by a generous gift from Merck and Co. We are grateful to SourceForge for hosting the project and Eugene Yi for providing the seven-protein mix data set. We would also like to acknowledge the following for endorsing the mzXML format: Philip C. Andrews, Tom Blackwell, Daniel Burns, Jayson Falkner, Panagiotis Papoulias, Abhik Shah, Peter Ulintz,

Al Burlingame, Robert Chalkley, Karl Clauser, Bruno Domon, James Eddes, Robert Moritz, Daniel Figeys, Barry L. Karger, William Hancock, Tomas Rejtar, Peter James, Matthias Mann, Sanford Markey, Matthias Wilm, Ken Williams and Kratos Analytical Limited (a Shimadzu Group Company).

COMPETING INTERESTS STATEMENT

The authors declare that they have no competing financial interests.

Published online at <http://www.nature.com/naturebiotechnology/>

- Patterson, S.D. & Aebersold, R.H. Proteomics: the first decade and beyond. *Nat. Genet.* **33** suppl., 311–323 (2003).
- Spellman, P.T. *et al.* Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol.* **3**, RESEARCH0046 (2002).
- Li, X., Zhang, H., Ranish, J.A. & Aebersold, R. Automated statistical analysis of protein abundance ratios from data generated by stable-isotope dilution and tandem mass spectrometry. *Anal. Chem.* **75**, 6658–6665 (2003).
- Han, D.K., Eng, J., Zhou, H. & Aebersold, R. Quantitative profiling of differentiation-induced microsomal proteins using isotope-coded affinity tags and mass spectrometry. *Nat. Biotechnol.* **19**, 946–951 (2001).
- Purvine, S., Eppel, J.T., Yi, E.C. & Goodlett, D.R. Shotgun collision-induced dissociation of peptides using a time of flight mass analyzer. *Proteomics* **3**, 847–850 (2003).
- Eng, J.K., McCormack, A.L. & Yates, J.R. III. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.* **5**, 976–989 (1994).
- Perkins, D.N., Pappin, D.J., Creasy, D.M. & Cottrell, J.S. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* **20**, 3551–3567 (1999).
- Li, X.J. *et al.* A tool to visualize and evaluate data obtained by liquid chromatography-electrospray ionization-mass spectrometry. *Anal. Chem.* **76**, 3856–3860 (2004).
- Gygi, S.P. *et al.* Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nat. Biotechnol.* **17**, 994–999 (1999).
- Keller, A., Nesvizhskii, A.I., Kolker, E. & Aebersold, R. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.* **74**, 5383–5392 (2002).
- Nesvizhskii, A.I., Keller, A., Kolker, E. & Aebersold, R. A statistical model for identifying proteins by tandem mass spectrometry. *Anal. Chem.* **75**, 4646–4658 (2003).
- Baldwin, M.A. Protein identification by mass spectrometry: issues to be considered. *Mol. Cell. Proteomics* **3**, 1–9 (2004).
- Prince, J.T., Carlson, M.W., Wang, R., Lu, P. & Marcotte, E.M. The need for a public proteomics repository. *Nat. Biotechnol.* **22**, 471–472 (2004).
- Zhang, N., Aebersold, R. & Schwikowski, B. ProBlID: a probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics* **2**, 1406–1412 (2002).
- Shannon, P. *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).