An End-to-End System for Organizing and Sharing Raw and Derived Mass Spectrometry Data

Cosmin Stejerean[1], Paiboon Siwamutita[2], E. D. Frank[3], Carol S Giometti[4], Gyorgy Babnigg[5], David Angulo[6*], Kevin Drew[7], Gregor von Laszewski[8]

[1] DePaul University, cosmin@cti.depaul.edu
[2]DePaul University, paisi@yahoo.com
[3]Argonne National Laboratory, efrank@mcs.anl.gov
[4]Argonne National Laboratory, csgiometti@anl.gov
[5]Argonne National Laboratory, gbabnigg@anl.gov
[6]DePaul University, dangulo@cti.depaul.edu
[7]University of Chicago, kdrew@cs.uchicago.edu
[8]Argonne National Laboratory, gregor@mcs.anl.gov

Abstract

With the increasing amount of work being performed in the field of Mass Spectrometry (MS), a huge amount of data is being generated. This data needs to be properly managed, organized and shared among researchers at various institutions. The problem is further complicated by the different proprietary formats used by manufacturers of MS machines. We demonstrate an end-to-end system to automate the process of converting the data to an open format, and to upload the data to a centralized server where it is easily organized and managed. The system allows scientists to browse, download, and use the data with third party tools. The user-view is simple and hides the underlying data-management system.

1 Introduction

Mass spectrometry (MS), applied to proteomics, is a method for identifying molecules by their mass-to-charge ratio. MS machines sort and measure the mass and charge of individual charged molecules (ions). These ions must be formed by getting them into the gas phase (desorption) and adding one or more protons or electrons to the protein or peptide fragment. The major ionization and desorption methods for proteins are electrospray ionization (ESI) and matrix-assisted laser desorption ionization (MALDI). The ions can be fragmented further in the MS machine to get more information about the protein such as the amino acid sequence or the presence of post-translational modifications. [BUC05]

Tandem mass spectrometry is one of the most sensitive and most reliable methods of protein identification currently available. There are many manufacturers who are competing to produce MS machines: Applied Biosystems (ABI) [ABI], IonSpec [IONSPEC], Waters Corporation (MicroMass equipment) [WATERS], ThermoFinnigan [TFG], Agilent, etc. The manufacturers

* Contact author

include proprietary software to obtain digital representation of the mass spectra produced by their instruments. These spectra can then be compared against protein databases to help identify which protein is most likely to match the sample. There are many database search programs available for protein identification, both commercial such as PEAKS [PEAKS], Mascot [MASCOT], SEQUEST [SEQUEST], and non-commercial such as OMSSA [GMK05] and Lutefisk. The four major techniques to identify proteins are de novo sequencing [LUC04], the use of "Peptide Sequence Tag" [MAN94], Cross-correlation, and probability base matching. Usually the search programs use a combination of available techniques to help identify the protein.

These database search programs have different requirements for the input format. The manufacturers of the MS machines are aware of this problem, and their solution is to create a plug-in to their program to convert their proprietary format into the format of the database that they are using. This quick fix creates a problem amongst researchers: Unless they use MS machines from the same company and the same database search program, they can not share data. It also creates a problem if the researcher wants to search databases which did not come with the MS machine.

One of the goals of our system is to obtain data from the MS experiment and facilitate data sharing with other researchers without compromising data security, and to enable easy data analysis by several different analysis programs. Facilitating data sharing necessitates standardization on a widely recognized data format. Towards this end, we promote and endorse the mzXML standard [PEH03], and support its use in all of our software tools. This file format is based on XML and can be used to store all of the information from a mass spectrometer. The information gleaned directly from the MS equipment does not always include metadata describing the sample or the biological experiment. For the Applied Biosystems (ABI) Qstar Pulsar XL, the experimental data is located in text based files in proprietary formats on a dedicated control processor. Thus, a tool that converts existing data from those proprietary formats into mzXML is required.

In order to support data analysis by several different programs, each requiring a different input format, another format conversion tool is mandated. This tool takes as input mzXML files, and converts them into the specific file formats dictated by the program requested.

Raw data from the MS machine contains noise that must be eliminated in order to get good quality data. This noise can be caused by low intensity signals and from impurity in the sample. The algorithms used to eliminate this noise from the MS data are considered trade secrets of the manufacturer. This can lead to significantly different results when converting the same data with different utilities. Therefore, we have decided to store the raw data and create our own utilities for noise reduction, peak baseline determination, and peak detection and integration. A nascent standard for proteomics, the Minimum Information for A Proteomics Experiment (MIAPE), dictates that the description of any algorithm utilized and all of the parameter settings used in any data transformations shall be reported in addition to the results. The purpose of this requirement it to ensure that the results can be duplicated in the future. Most scientists also do not interrogate the raw data directly from an MS machine but instead prefer to use data that has already undergone nose elimination transformation. This leads to another challenge: managing data from

experiments as well as the converted, computed and annotated data. We have implemented a management system that stores the raw data as well as the intermediate processed data.

While the physical storage of the data can be easily accomplished, there are two things that must be taken into consideration: how will the scientists access the data and how will the system administrators manage the data. Their needs can often conflict with each other. Scientists need to access the data in a way that is natural to them.. The system administrator however needs to worry about other things such as the integrity and availability of the entire system. They might sometimes need to move data from one physical location to another due to hardware problem or other circumstances. This might mean that scientists would have to learn where the data was relocated or perhaps learn how to use the new system. Therefore we designed a system that could address all of these problems. We decided to design the system with scientists in mind from the beginning, and to get feedback during the development process.

2 Use Case Scenario

An archetypal work-flow example (use case scenario) was used to discover the requirements for the system. The scenario starts with a researcher registering a sample with the analysis lab. Information such as sample name, researcher name, and sample preparation description are entered into a database. The researcher then gives the sample to the MS Lab Technician.

The MS Lab technician updates the database with experimental information including information about equipment and sample preparation. Once the experiment is complete, the lab technician runs a script to convert the data from proprietary format to an open, exchangeable format, and uploads the result to a central server. Once the data is on the central server, another script is run to do any final processing of the files, e.g., administrative processing or standard computations. The files are then made accessible through a web interface.

The researcher can now access the results via a web interface. The researcher can download the data or view it directly from the web interface. The researcher might decide to repeat the experiment, perhaps modifying experimental details. If so, the experiment is run and the new data will be made available as a new version on the server.

Once satisfied with the basic data, the researcher may choose to run analysis tools. The researcher uses a launcher script to run a tool on a desktop computer in a way that automatically fetches data from the central repository. Also, the researcher can request the administrator to restrict or enable data sharing on the web server.

3 Requirements

Several requirements are derived from the usage scenario:

1. The system must convert data from proprietary formats to the open mzXML format. The conversion tool should be a command line utility to support batch processing and automated workflows.

2. The system must store meta-data describing each sample and describing the sample processing.
3. It must be possible to store the exchangeable-format data on a central server. A tool to process and upload the data will be needed. It should work from the command.
4. Basic data-management tools are needed to manage data on the server after uploading.
    a. The tool should place the data in storage according to organizational policies.
    b. Users can reference data by a unique identifier that is easily remembered. This will be independent from the administrative (physical view): even if an administrator moves data to a new location, the users view will not change, i.e., the easily remembered identifier will not change.
    c. The system should be able to record different versions of the same data.
    d. The data must be available to researchers via a simple web interface.
5. A client-side launcher will adapt command line utilities to use data directly from the central repository.

4 Architecture

The system architecture, summarized in Figure 1, eight elements: Mass Spec Database, MS Connector (server), MS Connector (client), MS Launcher, Local tools, Web Browser, Kah, and Vendor code. Each is described here as well as their collaboration.

The Mass Spec Database stores all of the meta-data associated with the experimental data, e.g., instrumentation, conditions, researcher name, etc. A table containing sample and experiment parameters also contains allowable values to facilitate easy data entry and validation. The database does not store the mzXML data files although it does store the physical location of each file and the user-view identifier for that file. This is described in more detail below.

The MS instrument data is stored by Vendor code in proprietary format on local disk in the lab and needs to be converted to the open mzXML format. A configurable command line utility does the conversion, typically making use of licensed vendor code. Although inaccurate, this step is lumped into the MS Connector in Figure 1 for sake of simplicity.

Once the data has been converted, the MS Connector client transfers the data to the server. The MSConnector Client runs on a machine in the MS lab: It gets metadata out of the Lab database and emits a load script for the mzXML files to be uploaded. The MSConnector Client then uploads the data files and the load file to the server where it is processed by the MSConnector Server utility. The MSConnector Server moves the mzXML files to appropriate locations in the file system (enforcing server-side organizational rules) and records the relationship between the

physical location and logical name (user-view identifier) in the Kah Catalog.

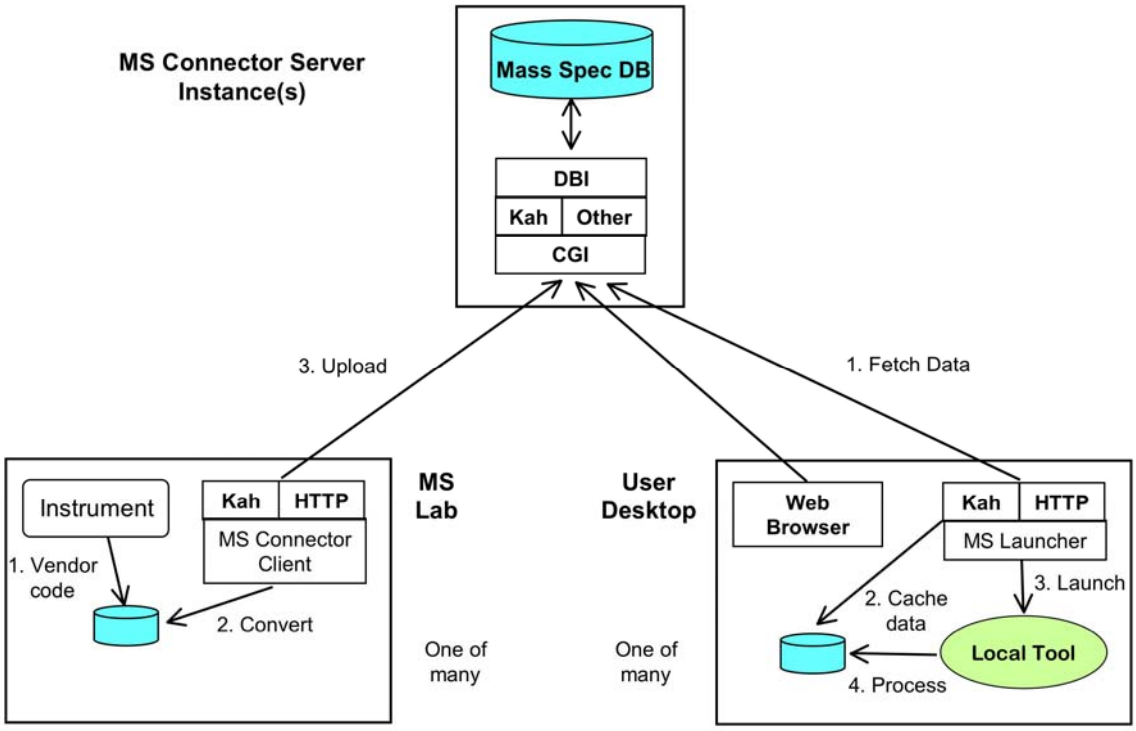

Figure 1. System Architecture

The Kah Catalog relates the physical location and logical name of the datafile. Its role is to decouple the physical file management and the user view. The catalog organizes the user view into a tree structure of names analogous to folders and files. The pathname in the tree is the unique, stable, identifier users employ to access data. This allows scientists to browse data in an intuitive fashion since most scientists can use a computer system to browse for files.

Local tools can process the mzXML data with the help of the MSLauncher. This utility retrieves an mzXML file via the user-view pathname and launches the local tool, directing the tool to the cached copy of the mzXML file. Users can write workflows in terms of Kah pathnames. These scripts will be stable against administrative rearrangement of data on the server.

A web-browser interface is provided with which users can locate MS data, review meta-data, determine user-view pathnames for the data, and download the files without installing additional software.

The entire workflow is shown in Figure 2.

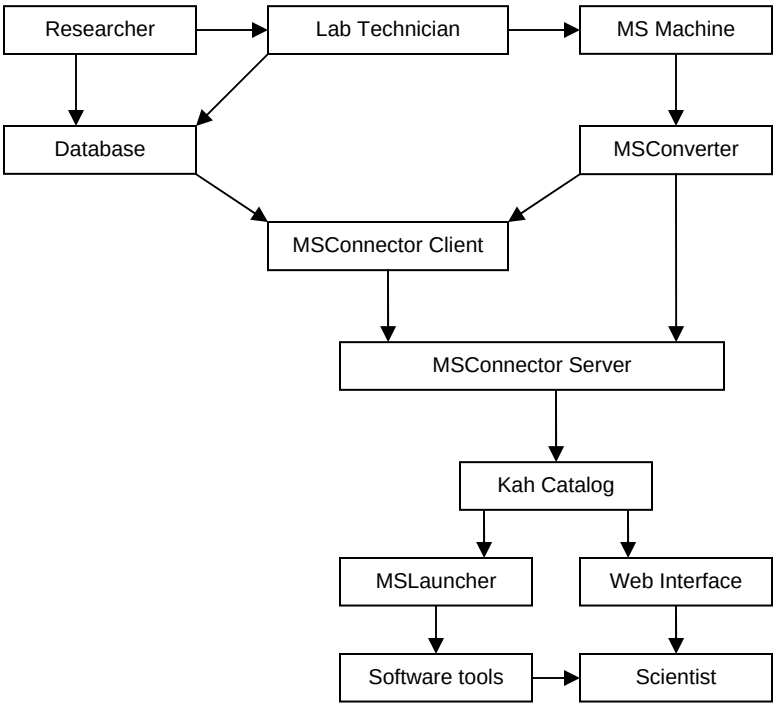

Figure 2. End-to-End Workflow

5 Implementation

To implement this system we worked closely with the Mass Spectrometry Lab in the Biosciences Division at the Argonne National Laboratory. The laboratory had approximately 400 files in ABI's proprietary wiff format with which we could experiment during development.

We modified an open source utility called mzStar [MZSTAR] to convert wiff files to mzXML. The original version could not be executed as a command line utility and generated improper XML for nested scan structures. The errors in the code were identified and corrected so that the program would produce well-formed mzXML. The new program, which we named MSwiff, was redesigned to use a command line interface. The settings for centroiding the data as well as the information for the MS machine were moved to a configuration file. Because the MSwiff programs uses the vendor-supplied libraries, it can only be run on a computer with a licensed version of Analyst installed.

We created a utility in JAVA that can convert all the wiff files in a folder to mzXML. It logs which file is being converted and, if interrupted, it can quickly resume the conversion from the last unconverted file. In addition to converting files, this integrates with our other tools on the client machine to complete the workflow. The program will run the MSConnector client on the converted files and then upload all the files to the central server using pscp [PSCP]. The tool also has a feature to be run on a single file when given the full path to the file.

This satisfies the requirements for the client side of the system with the exception of the database to record the metadata. A database for the capturing of the metadata was developed. The database was implemented in Microsoft Access because it was the easiest way to design and test the database without installing additional software on the Analyst machine.

The MsConnector provides an interface that can be used to create specific instances of the MsConnector for various organizations. Different organizations can have different databases or ways to store metadata as well as different requirements about how the data should be organized on the server. The MsConnector client has three parts: reader, metadata and writer. The reader will read the files to be processed. The current implementation is to read the list of files in a given path. However the list of files could also be read from a text file, spreadsheet, database, etc. The second part is the metadata. Once the reader puts together a list of files it passes it to the metadata handler which looks up the information for each one in the database and stores the information for each file in a custom data structure. An array is created for all the different samples in the list and this array is passed to the writer. The writer then creates a load file. Right now the load file contains tab-separated items for the following: path on the server, path in the catalog and sample name.

The load file is processed by the server side MSConnector. The first part is the reader which will read the load file created by the MsConnector client writer. The reader will generate two sets of commands, one for the catalog connector and the second for the file handler. The catalog connector will update the Kah Catalog. The file handler will move mzXML data to the proper location on the server.

The Kah system can map a user-view pathnames to one of several handlers, called data factories, to map the user-view to phyical. Our implementation uses one data factory per MS Lab that wants to use of our centralized server. This means that the catalog information for their data can be stored on different database servers but the system appears to be a single web interface. When a user requests data via a pathname (user-view) the corresponding URL is looked up in the database (administrators-view); the data are retrieved and sent to the user. This allows the administrator to move the data to another server and update the URLs in the catalog without any changes visible to the user.

The user interface was implemented using HTML requests and CGI on an Apache web server. The CGI scripts were written in Python. When a user first visits the webpage the CGI script initializes the catalog and returns the available data factories presenting them as the first nodes in the user-view tree. This is illustrated in Figure 3 where the two available data factories are for the Proteomics Lab at the University of Chicago and the Mass Spectrometry Lab at Argonne National Laboratory.

The user can browse deeper into the tree. The CGI determines when browsing has reached a leaf node and thus has located a data file. The script will then display available information including the full catalog path.

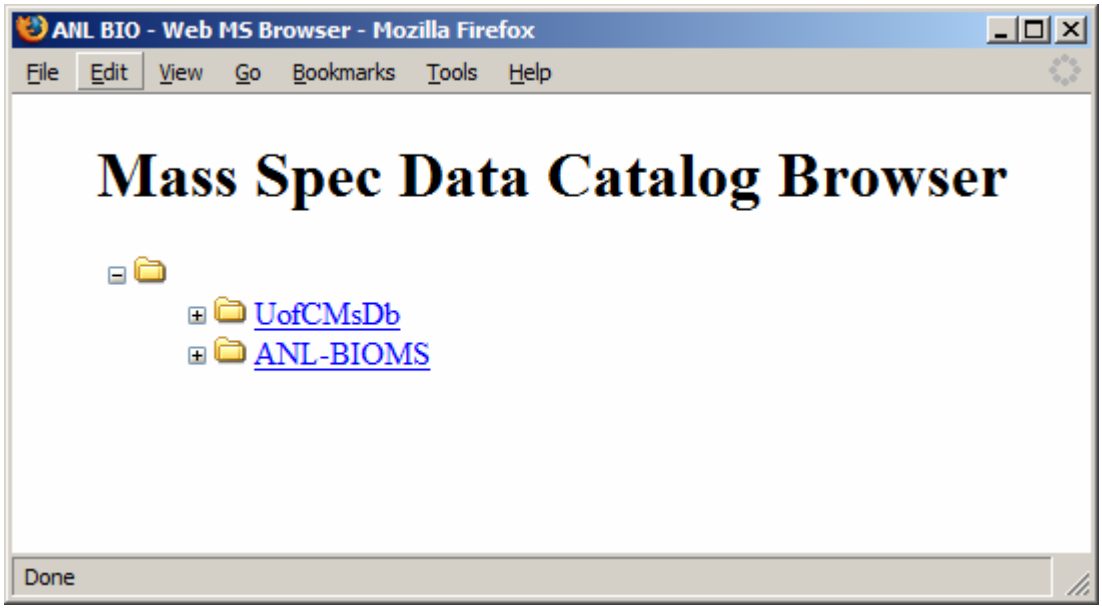

Figure 3. Top Level View of the Browser: One branch per MS Laboratory.

6 Performance Results

We were successful in using this system to convert 400 wiff files to mzXML and upload them to a centralized server. The conversion of the files is relatively slow because of the current implementation of the MSwiff utility. The slow performance is due to the design of the API for extraction of mass spectra from the raw files and the way the original mzStar handled information. Because of the XML tags in the mzXML format the converted files are much larger than the original wiff files. This means that uploading mzXML files requires significantly more bandwidth than uploading wiff files. This can be addressed in the future by compressing the files before uploading.

We have tested the end-to-end workflow. We started with 400 wiff files that were converted to mzXML, processed with the metadata in the database, uploaded to the server and entered in the catalog. We were able to browse the catalog using the web interface and run Lutefisk using our MSLauncher utility.

Future work on this project could include improving the speed and size problems as well as converting some of the utilities to use a common language to make it easier to maintain them. Additional requirements may be addressed in the future such as providing access control at either the organizational level or the user level. If access control is implemented, more work will be necessary to create mechanisms to share data between scientists at different organizations.

7 Conclusions

We created an end-to-end workflow for sharing and organizing MS data from various laboratories. The system addressed problems including the proprietary formats of MS data and managing and accessing data. We reused and improved existing open source tools to meet our requirements and plan to release the utilities written in the course of this project as open source tools as well. This system will likely be used to store private MS data before it is reviewed, annotated, curated and uploaded to a public database.

Acknowledgements

The authors would like to thank DePaul University for allowing us to host the centralized MS server on their network. We would also like to thank Argonne National Laboratory for providing us with access to their facilities in order to develop this project. This work was supported in part by the National Science Foundation program Research Experience for Undergraduates under Grant No. 0353989, 2004-2006. http://reu.cti.depaul.edu

References

[ILH92] Yannis E. Ioannidis, Miron Livny and Eben M. Haber, *Graphical User Interfaces for the Management of Scientific Experiments and Data*, Sigmod Record, Vol. 21, No. 1, March 1992

[BUC05] Michelle Buchanan. *Genomes to Life: Technology Assessment for Mass Spectrometry.* Oak Ridge National Laboratory. Online Publication. Available at http://www.doegenomestolife.org/pubs/mass_spec_exec_summ.pdf. Last Visit July 28, 2005

[GMK05] Lewis Y Geer, Sanford P Markey, Jeffrey A Kowalak, Lukas Wagner, Xu Ming, Dawn M Maynard, Xiaoyu Yang, Wenyao Shi and Stephen H Bryant, *Open Mass Spectrometry Search Algorithm.* Research Articles: Journal of Proteome Research. Online Publication. Available at http://proteome.nih.gov/papers/Geer_jpr_OMSSA.pdf. Last Visit, July 28, 2005

[LUC04] Bingwen Lu and Ting Chen, *Algorithms for De Novo Peptide Sequencing using Tadem Mass Spectrometry.* DDT: Biosilico. Vol. 2, No. 2, P. 85-89, March 2004.

[MAN94] M Mann and M Wilm, *Error-Tolerant Identification of Peptide in Sequence Databases by Peptide Sequence Tag,* Protein & Peptide Group, European Molecular Biology Laboratory (EMBL), Anal. Chem., Vol. 66, pages 4390-4399, 1994.

[PEH03] Patrick Pedrioli, J Eng, R Hubley, *Creation of an Open Standard File Format for the representation of MS Data.* Poster present at 51st ASMS conference in Montreal. June 2003.

[PED05] Patrick Pedrioli, *mzXML Schema Documentation.* Webpage. Available at http://sashimi.sourceforge.net/schema_revision/mzXML_2.1/Doc/mzXML_2.1_tutorial.pdf. Last Visit July 28, 2005

[PSCP] Putty Documentation, Chapter 5, Available online at http://the.earth.li/~sgtatham/putty/0.52/htmldoc/Chapter5.html. Last Visit April 18, 2006

[ABI] Applied Biosystems Website, http://www.appliedbiosystems.com/. Last Visit April 18, 2006

[IONSPEC] IonSpec Website, http://www.ionspec.com. Last Visit April 18, 2006

[WATERS] Waters Corporation Website, http://www.waters.com. Last Visit April 18, 2006

[TFG] Thermo Electron Corporation, http://www.thermo.com. Last Visit April 18, 2006

[PEAKS] PEAKS website, www.bioinformaticssolutions.com/products/peaksstudio.php. Last Visit April 18, 2006

[MASCOT] Matrix Science Website, http://www.matrixscience.com/search_intro.html

[SEQUEST] J.K. Eng, A.L. McCormack and J.R. Yates III, *An Approach to Correlate Tandem Mass Spectral Data of Peptides with Amino Acid Sequences in a Protein Database*, J Am Soc Mass Spectrom, Volume 5, Pages 976-989, 1994.

[GPL] GNU General Public License, http://www.gnu.org/copyleft/gpl.html, Last Visit April 18, 2006

[MZSTAR] mzStar software, http://sashimi.sourceforge.net/software_glossolalia.html, Last Visit August 16, 2005

[ORACLE] Oracle Corporation, http://www.oracle.com, Last Visit August 16, 2005

[PGSQL] PostgreSQL Inc, *PostgreSQL Homepage*, http://www.postgresql.net, Last Visit August 16, 2005

[PYTHON] Python Software Foundation, *Python Programming Language Homepage*, http://www.python.org. Last Visit August 16, 2005