Journal of proteome research
TECHNICAL NOTE
pubs.acs.org/jpr

jTraML: An Open Source Java API for TraML, the PSI Standard for Sharing SRM Transitions

Kenny Helsens,[†,‡] Mi-Youn Brusniak,[§] Eric Deutsch,[§] Robert L. Moritz,[§] and Lennart Martens*[,†,‡]

[†]Department of Medical Protein Research, VIB, Ghent, Belgium
[‡]Department of Biochemistry, Ghent University, Ghent, Belgium
[§]Computational Biology, Institute for Systems Biology, Seattle, Washington, United States

S *Supporting Information*

ABSTRACT: We here present jTraML, a Java API for the Proteomics Standards Initiative TraML data standard. The library provides fully functional classes for all elements specified in the TraML XSD document, as well as convenient methods to construct controlled vocabulary-based instances required to define SRM transitions. The use of jTraML is demonstrated via a two-way conversion tool between TraML documents and vendor specific files, facilitating the adoption process of this new community standard. The library is released as open source under the permissive Apache2 license and can be downloaded from http://jtraml.googlecode.com. TraML files can also be converted online at http://iomics.ugent.be/jtraml.

KEYWORDS: proteomics, bioinformatics, mass spectrometry, standards, selection reaction monitoring

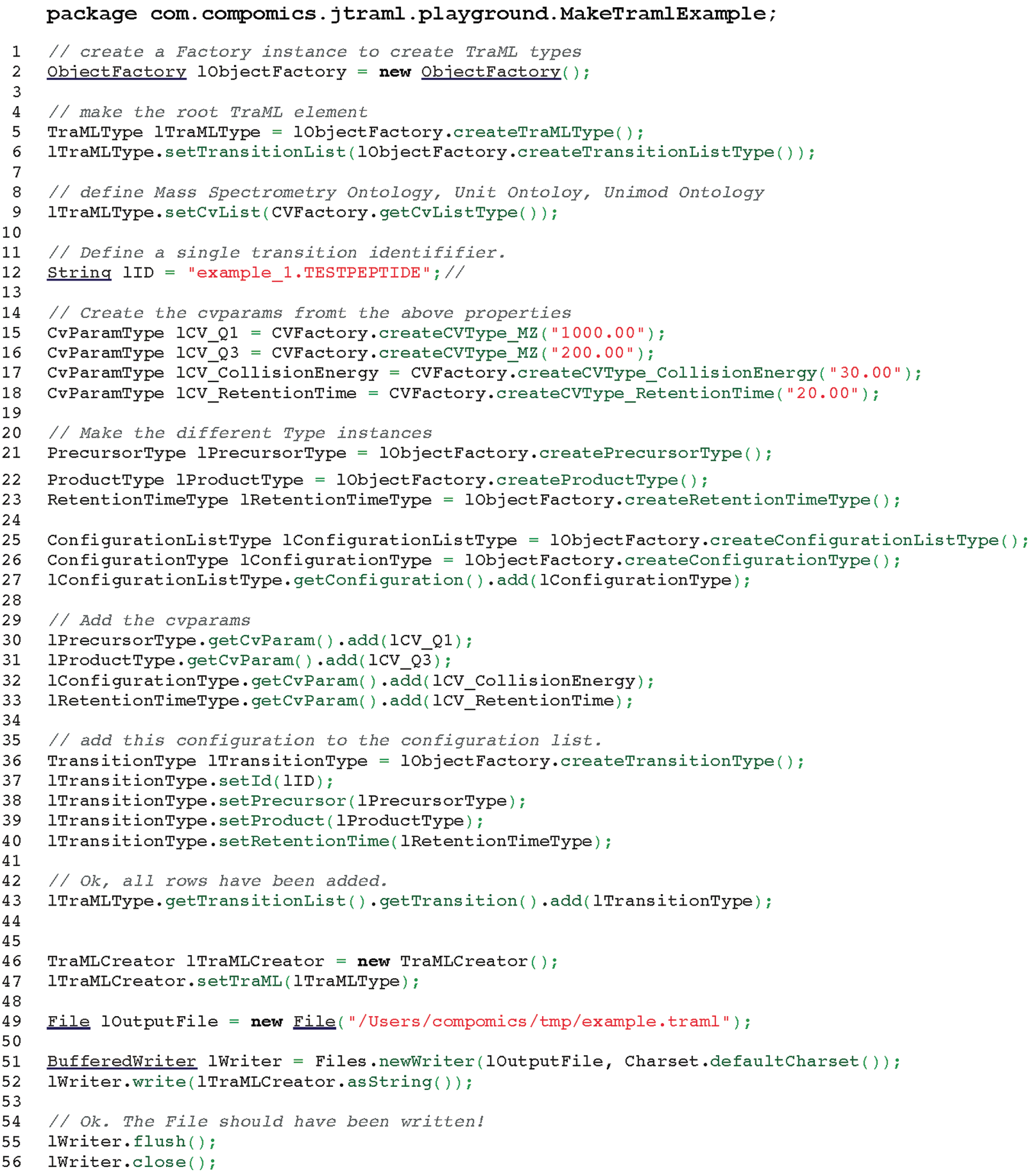
INTRODUCTION

In the last two decades, mass-spectrometry-driven proteomics has grown into a popular and powerful strategy to measure significant fractions of the expressed proteome.[1] Two major types of proteomics analysis methods can be distinguished: shotgun proteomics and targeted proteomics. Shotgun proteomics is commonly used for the explorative identification and quantification of as many detectable peptides as possible, rendering a general view on the proteome. Targeted proteomics, in contrast, focuses on a subset of peptides relevant to the study, yielding a restricted yet highly accurate view of the proteome part of interest. Clearly, both types of proteomics methodologies are highly complementary; shotgun proteomics is outstanding for screening proteome perturbations under multiple controlled conditions, whereas targeted proteomics excels in the validation of newly formulated hypotheses, derived from these observed perturbations.

While single reaction monitoring (SRM) has a long-standing history in mass spectrometry,[2] its application in targeted proteomics is relatively recent.[3−5] A single SRM assay is set to detect a single peptide using a combined filter of precursor ion mass and product ion mass, with a fragmentation step in between to derive product ions from the precursor. Such a joint mass filter is known as a transition, and a combination of transitions for a given precursor can provide an extremely specific and sensitive signature to monitor the corresponding protein. As peptide fragmentation remains difficult to predict in detail,[6,7] SRM assays are preferably based on previously identified MS/MS spectra.[8] Although such spectra have been amassing in large quantities in public repositories like PRIDE and PeptideAtlas over the last few years,[9,10] many of these were not generated using quadrupole mass spectrometers and are therefore not always the most reliable resource for designing SRM assays on the often-used triple-quadrupole instruments.[6,7,11] Furthermore, rigorous data quality control is not yet fully established in these repositories, potentially leading to the accumulation of false positive peptide identifications that would also provide poor SRM assays. Therefore, the de facto standard for developing robust SRM assays is to make use of synthetic peptides which enable a priori profiling of peptide (fragmentation) behavior on the analytical instrumentation proper. Thankfully, these synthetic peptides (if labeled with stable isotopes) can also be used to quantify the targeted peptides when mixed in with the sample during analysis[12]

Public dissemination of valid SRM assays should therefore be encouraged in order to improve the quality and throughput of targeted proteomics experiments. For this purpose, public repositories like SRMAtlas have recently been established to collect and disseminate SRM assays that have been successfully applied in previous research projects, so that they can be readily reused in subsequent studies.[13,14] Furthermore, it is important that software platforms are established to support automation and validation of SRM analyses.[15,16] Yet no standard means to communicate the specifics of such SRM assays existed until recently. As a consequence, SRM assays are now largely shared via different vendor-specific file formats. In order to solve this issue, the mass spectrometry working group of the Proteomics Standards Initiative (PSI-MS) has recently formalized the transition markup language (TraML) file format to communicate SRM assays for both peptides and small molecules in a standardized form.[17] TraML is currently going through the PSI community

Received: July 15, 2011
Published: October 04, 2011



```
package com.compomics.jtraml.playground.MakeTramlExample;

// create a Factory instance to create TraML types
ObjectFactory lObjectFactory = new ObjectFactory();

// make the root TraML element
TraMLType lTraMLType = lObjectFactory.createTraMLType();
lTraMLType.setTransitionList(lObjectFactory.createTransitionListType());

// define Mass Spectrometry Ontology, Unit Ontoloy, Unimod Ontology
lTraMLType.setCvList(CVFactory.getCvListType());

// Define a single transition identififier.
String lID = "example_1.TESTPEPTIDE";//

// Create the cvparams fromt the above properties
CvParamType lCV_Q1 = CVFactory.createCVType_MZ("1000.00");
CvParamType lCV_Q3 = CVFactory.createCVType_MZ("200.00");
CvParamType lCV_CollisionEnergy = CVFactory.createCVType_CollisionEnergy("30.00");
CvParamType lCV_RetentionTime = CVFactory.createCVType_RetentionTime("20.00");

// Make the different Type instances
PrecursorType lPrecursorType = lObjectFactory.createPrecursorType();
ProductType lProductType = lObjectFactory.createProductType();
RetentionTimeType lRetentionTimeType = lObjectFactory.createRetentionTimeType();

ConfigurationListType lConfigurationListType = lObjectFactory.createConfigurationListType();
ConfigurationType lConfigurationType = lObjectFactory.createConfigurationType();
lConfigurationListType.getConfiguration().add(lConfigurationType);

// Add the cvparams
lPrecursorType.getCvParam().add(lCV_Q1);
lProductType.getCvParam().add(lCV_Q3);
lConfigurationType.getCvParam().add(lCV_CollisionEnergy);
lRetentionTimeType.getCvParam().add(lCV_RetentionTime);

// add this configuration to the configuration list.
TransitionType lTransitionType = lObjectFactory.createTransitionType();
lTransitionType.setId(lID);
lTransitionType.setPrecursor(lPrecursorType);
lTransitionType.setProduct(lProductType);
lTransitionType.setRetentionTime(lRetentionTimeType);

// Ok, all rows have been added.
lTraMLType.getTransitionList().getTransition().add(lTransitionType);


TraMLCreator lTraMLCreator = new TraMLCreator();
lTraMLCreator.setTraML(lTraMLType);

File lOutputFile = new File("/Users/compomics/tmp/example.traml");

BufferedWriter lWriter = Files.newWriter(lOutputFile, Charset.defaultCharset());
lWriter.write(lTraMLCreator.asString());

// Ok. The File should have been written!
lWriter.flush();
lWriter.close();
```

Figure 1. A source code example to create a TraML file using the jTraML library.

review process,[18] involving both researchers from academia and companies, with version 1.0 expected to be released in 2011.

Now that it is has become possible to describe SRM assays in a standardized way, it is crucial to promote the uptake of the TraML standard by software developers as well as mass spectrometrists. We therefore developed jTraML: a freely available, open-source and cross-platform Java library that provides a fully featured programming interface to work with TraML files (http://jtraml.googlecode.com), and we have used jTraML to implement a user-friendly Web application to perform the two-way transformation between TraML and several vendor-specific formats.

■ MATERIALS AND METHODS

The jTraML library is fully written in the Java programming language. The data representation layer has been generated from the TraML schema definition (http://www.psidev.info/index.php?q=node/405) using JAXB (http://download.oracle.com/javase/6/docs/technotes/guides/xml/jaxb/index.html). The jTraML library is licensed under the permissive Apache2 open source license and the project Web site is located at http://jtraml.googlecode.com. The project is managed and built using Maven2 (http://maven.apache.org).

The API is documented on the project Web site wiki and using JavaDoc.

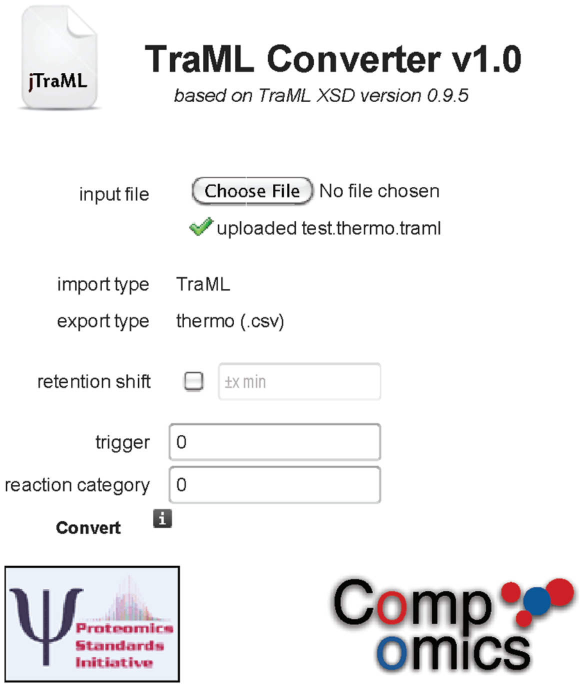
Vendor-specific SRM assay file formats, including those from Thermo Scientific, Agilent, and ABI, are tested for two-way conversion to and from TraML using JUnit (http://junit.org).

The jTraML converter Web application was developed using Vaadin (http://www.vaadin.com), a contemporary user interface framework for building Web applications. Example files for testing the converter application are available in the download section of the jTraML project Web site (http://code.google.com/p/jtraml/download).

■ RESULTS

A simplified UML diagram of the library is shown in Figure S1 (Supporting Information). The TraMLType object is the root of a TraML document containing multiple object lists. The CompoundListType for instance contains either CompoundType or PeptideType objects, with each such object describing a single targeted analyte. TransitionType object then describes a single SRM transition that references out to a CompoundType or PeptideType object, thus representing the one-to-many relation that derives from the monitoring of multiple transitions for a single analyte. Additional details can be specified for all objects using controlled vocabulary (CV) parameters that are modeled by the CVParamType object. For instance, retention time is expressed either in minutes or seconds, both defined in the unit ontology (UO). Also, the retention time is described either by start and stop times or by a single centroid time, both defined by the mass spectrometry ontology (MS). To facilitate access to the required CVs, the OboManager object makes use of the Ontology Lookup Service, a continuously updated ontology service hosted by the European Bioinformatics Institute (EBI),[19] thus ensuring the correct use of the relevant standardized terms throughout the library. The CVFactory object is made available to provide convenient methods for the efficient creation of such CV parameters. API usage is illustrated through a source code example (Figure 1) that shows how to create a simple TraML file with a single transition. Finally, we have implemented the PSI Validator framework in order to create a TraMLValidator object that can verify the syntactic and semantic validity of a TraML file on the basis of the XML schema and a set of semantic rules.[20]

To provide the community with a readily deployable conversion tool and to illustrate the use of our jTraML library, we built a TraML converter that supports two-way conversion between a TraML document and the vendor specific file formats from ABI-SCIEX, Agilent, and Thermo Scientific. This conversion application is available in two forms: a command line interface that will greatly facilitate the adoption of TraML by bioinformaticians in existing SRM software pipelines (http://code.google.com/p/jtraml/wiki/converter#TraML_Converter_Command_Line), and an end-user oriented Web application located at http://iomics.ugent.be/jtraml/. An illustration of this easy-to-use Web application interface is shown in Figure 2. The user can upload either a TraML formatted file or a vendor specific file format, and after specifying the correct import and desired export file type, this file will be converted and the output file made available for immediate download. In those specific cases where certain vendor-specific parameters may be missing from a TraML file, feedback is asked from the user. The scenario where such an issue could occur is a conversion of one vendor's format to another through a TraML intermediate, where one vendor uses an (asymmetrical) lower and upper retention time window, while another vendor might require a (symmetrical) centroid retention time. In this case, the converter will suggest to calculate the centroid retention time as the average time between the upper and lower retention time, or, the other way round, the converter will suggest to calculate the upper and lower retention time by subtracting and adding half of the retention time window to the centroid retention time. Furthermore, if the retention time window is missing or when an asymmetrical window is required, then an input dialogue appears where the user can fill in a default retention time window. Furthermore, each vendor format has instrument specific parameters that defaulted such that they are fully operational. Note, however, that the TraML converter cannot be used to add meta information about the instrument or the sample in the TraML file because the Web application focuses on the conversion of operational parameters such as peptide retention time or collision energy in order to produce instrument-ready output files.



jTraML

TraML Converter v1.0

based on TraML XSD version 0.9.5

input file Choose File No file chosen

✓ uploaded test.thermo.traml

import type TraML

export type thermo (.csv)

retention shift □ ±x min

trigger 0

reaction category 0

Convert

Figure 2. jTraML Converter (http://iomics.ugent.be/jtraml). A screenshot of the jTraML Converter Web application. First, the user has to upload one file to the server, and after specifying the import and export types, the transitions will be converted and the resulting file can be downloaded from the server. The PSI logo is used by permission of the Proteomics Standards Initiative.

Because TraML is able to faithfully represent the various vendor formats, our conversion application will thus easily allow anyone interested to manage their SRM assays in uniformly formatted, vendor-independent TraML documents.

■ DISCUSSION

We have created a freely available and permissively licensed open source Java library to create and parse TraML files. As the TraML standard is currently being finalized, the jTraML library we describe here is the only Java implementation of the TraML standard to date. Our library can thus be of great use to facilitate the adoption and uptake of this emerging standard in diverse software pipelines for targeted proteomics. Because of the importance of facilitating the two-way conversion between vendor specific file formats and standardized TraML files, we used our jTraML library to create developer-oriented command line and end-user-oriented Web application tools that can be used for batch-mode or user-driven conversion of existing SRM assays into and out of the TraML format.

■ ASSOCIATED CONTENT

S Supporting Information

The (shortened) UML diagram shows the key classes and their relations that are present in the jTraML library. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: lennart.martens@ugent.be. Tel.: 32-92649458. Fax: 32-92649484.

■ ACKNOWLEDGMENT

K.H is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO-Vlaanderen). L.M. acknowledges the support of Ghent University (Multidisciplinary Research Partnership "Bioinformatics: from nucleotides to networks"), the PRIME-XS project, Grant Agreement No. 262067 funded by the European Union seventh Framework Program. L.M. and E.D acknowledge the ProteomeXchange project, Grant Agreement No. 260558 funded by the European Union Seventh Framework Program. R.M. acknowledges the support of National Institutes of Health - National Human Genome Research Institute (Grant HG005805), the National Institute of General Medical Sciences (Grant GM087221 to E.D.), the Center for Systems Biology (Grant GMO76547 to R.M.) and the Duchy of Luxembourg (Systems Biology initiative to R.M.).

■ ABBREVIATIONS:

API, application programmer's interface; SRM, selected reaction monitoring; XSD, XML schema definition; CV, controlled vocabulary

■ REFERENCES

(1) Nilsson, T; Mann, M; Aebersold, R; Yates, J. R., III; Bairoch, A; Bergeron, J. J. M. Mass spectrometry in high-throughput proteomics: Ready for the big time. *Nat. Methods* **2010**, *7*, 681–685.

(2) Kondrat, R. W.; McClusky, G. A.; Cooks, R. G. Multiple reaction monitoring in mass spectrometry/mass spectrometry for direct analysis of complex mixtures. *Anal. Chem.* **1978**, *50*, 2017–2021.

(3) Stahl-Zeng, J; Lange, V; Ossola, R; Eckhardt, K; Krek, W; Aebersold, R; Domon, B. High sensitivity detection of plasma proteins by multiple reaction monitoring of *N*-glycosites. *Mol. Cell. Proteomics* **2007**, *6*, 1809–1817.

(4) Lange, V; Picotti, P; Domon, B; Aebersold, R. Selected reaction monitoring for quantitative proteomics: A tutorial. *Mol. Syst. Biol.* **2008**, *4*, 222.

(5) Kuzyk, M. A.; Smith, D; Yang, J; Cross, T. J.; Jackson, A. M.; Hardie, D. B.; Anderson, N. L.; Borchers, C. H. Multiple reaction monitoring-based, multiplexed, absolute quantitation of 45 proteins in human plasma. *Mol. Cell. Proteomics* **2009**, *8*, 1860–1877.

(6) Mueller, M; Vizcaíno, J. A.; Jones, P; Côté, R; Thorneycroft, D; Apweiler, R; Hermjakob, H; Martens, L. Analysis of the experimental detection of central nervous system-related genes in human brain and cerebrospinal fluid datasets. *Proteomics* **2008**, *8*, 1138–1148.

(7) Barsnes, H.; Eidhammer, I.; Martens, L. A global analysis of peptide fragmentation variability. *Proteomics* **2011**, *11* (6), 1181–1188.

(8) Prakash, A; Tomazela, D. M.; Frewen, B; Maclean, B; Merrihew, G; Peterman, S; Maccoss, M. J. Expediting the development of targeted SRM assays: Using data from shotgun proteomics to automate method development. *J. Proteome Res.* **2009**, *8*, 2733–2739.

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

(10) Desiere, F; Deutsch, E. W.; King, N. L.; Nesvizhskii, A. I.; Mallick, P; Eng, J; Chen, S; Eddes, J; Loevenich, S. N.; Aebersold, R. The PeptideAtlas project. *Nucleic Acids Res.* **2006**, *34*, D655–658.

(11) Jansen, R; Lachatre, G; Marquet, P. LC—MS/MS systematic toxicological analysis: Comparison of MS/MS spectra obtained with different instruments and settings. *Clin. Biochem.* **2005**, *38*, 362–372.

(12) Anderson, L; Hunter, C. L. Quantitative mass spectrometric multiple reaction monitoring assays for major plasma proteins. *Mol. Cell Proteomics* **2006**, *5*, 573–588.

(13) Picotti, P; Lam, H; Campbell, D; Deutsch, E; Mirzaei, H; Ranish, J; Domon, B; Aebersold, R. A database of validated assays for the targeted mass spectrometric analysis of the *S. cerevisiae* proteome. *Nat. Methods* **2008**, *5*, 913–914.

(14) Picotti, P; Rinner, O; Stallmach, R; Dautel, F; Farrah, T; Domon, B; Wenschuh, H; Aebersold, R. High-throughput generation of selected reaction-monitoring assays for proteins and proteomes. *Nat. Methods* **2010**, *7*, 43–46.

(15) Reiter, L; Rinner, O; Picotti, P; Hüttenhain, R; Beck, M; Brusniak, M-Y; Hengartner, M. O.; Aebersold, R. mProphet: Automated data processing and statistical validation for large-scale SRM experiments. *Nat. Method.* **2011**, *8*, 430–435.

(16) Brusniak, M-YK; Kwok, S-T; Christiansen, M; Campbell, D; Reiter, L; Picotti, P; Kusebauch, U; Ramos, H; Deutsch, E. W.; Chen, J; Moritz, R. L.; Aebersold, R. ATAQS: A computational software tool for high throughput transition optimization and validation for selected reaction monitoring mass spectrometry. *BMC Bioinf.* **2011**, *12*, 78.

(17) Deutsch, E. W.; Chambers, M.; Bertsch, A.; Levander, F.; Neumann, S.; Shofstahl, J.; Campbell, D.; Christiansen, M.; Martens, L.; Binz, P.-A.; Aebersold, R.; Brusniak, M. Y. TraML - A standard format for exchange of selected reaction monitoring transition lists. http://www.psidev.info/index.php?q=node/405 (2010).

(18) Vizcaíno, J. A.; Foster, J. M.; Martens, L. Proteomics data repositories: Providing a safe haven for your data and acting as a springboard for further research. *J. Proteomics* **2010**, *73*, 2136–2146.

(19) Côté, R; Reisinger, F; Martens, L; Barsnes, H; Vizcaino, J. A.; Hermjakob, H. The Ontology Lookup Service: Bigger and better. *Nucleic Acids Res.* **2010**, *38*, W155–160.

(20) Montecchi-Palazzi, L; Kerrien, S; Reisinger, F; Aranda, B; Jones, A. R.; Martens, L; Hermjakob, H. The PSI semantic validator: A framework to check MIAPE compliance of proteomics data. *Proteomics* **2009**, *9*, 5112–5119.