



Proteomics Data Collection (ProDaC) Work Package 2, Deliverable 2.1, Report M12

Lund, September, 2007

Fredrik Levander¹, Lennart Martens², Martin Eisenacher³, Christian Stephan³, and Jari Häkkinen⁴

¹Dept of Immunotechnology, Lund University, Lund, Sweden

²EMBL Outstation, European Bioinformatics Institute (EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge, UK

³Medizinisches Proteom-Center (MPC), Ruhr-Universitaet Bochum, ZKF E.143, Bochum, Germany

⁴Dept of Theoretical Physics, Complex Systems Division, Lund University, Lund, Sweden

Corresponding authors

Fredrik Levander, Fredrik.Levander@immun.lth.se, +46 46 222 3835

Jari Häkkinen, jari@thep.lu.se, +46 46 222 9347

Contents

Work Package 2: Standards Implementation – Short Summary	2
Objectives.....	2
Involved Partners	2
Deliverable W2.D1	2
Background	2
Introduction.....	2
Overview of test needs among ProDaC partners	3
Generation of test files	3
Survey of existing validation tools.....	3
Creation of semantic validation tools.....	3
Online validation of data files	4
Testing /certification	5
Conclusions.....	5
Funding	5



ProDaC website: <http://www.fp6-prodac.eu/>

ProDaC mailing list: <http://lists.ruhr-uni-bochum.de/mailman/listinfo/prodac>

Abbreviations

MPC: Medizinisches Proteom-Center, Germany
EBI: European Bioinformatics Institute, United Kingdom
ETH: Federal Inst. of Technology ETH Zurich, Switzerland
VIB: Department of Medical Protein Research, University of Ghent, Belgium
UCD, Conway Institute of Biomolecular and Biomedical Research University College Dublin, Ireland
SIB, Swiss Institute of Bioinformatics, Geneva Switzerland
LundU: Lund Swegene, Bioinformatics Facility, Lund University, Sweden
UNIMAN, School of Biological Sciences, Manchester University, UK
GB: GeneBio, Switzerland
PAG: Protagen AG, Germany
BDAL: Bruker Daltonics GmbH, Germany
MS: MatrixScience, United Kingdom

PSI: Proteomics Standards Initiative
XML: eXtensible Markup Language
XSD: XML Schema Definition

Work Package 2: Standards Implementation – Short Summary

Objectives

For a data exchange standard to become accepted and used, it is essential to have support for the standard in major tools. This work package coordinates the implementation of the Proteomics Standards Initiative (PSI) standards in well-known proteomics tools.

Involved Partners

LundU (leader), MPC, EBI, MS, GB, PAG, ETH, VIB, UCD, UNIMAN, BDAL

Deliverable W2.D1

Definition of a test suite for PSI standards

Background

Previous experience has shown that tool providers sometimes claim compatibility to a certain standard, but actually only implement it in a very rudimentary or even an incorrect way. It is therefore important to be able to validate the files produced by standard implementers using a well-defined test suite. This deliverable consists of the definition of such a test suite

Introduction

As the new standards for proteomics data exchange are developed (Work Package 1), it is of importance to make example files employing these standards available for a wider audience, as well as to enable developers to validate the output from their applications. The standard for mass spectrometry data exchange, mzML (developed from mzData and mzXML), has come to such a maturity that a test suite is appreciated. The analysisXML format for search results is still subject to important changes, but the test suite is easily adapted to this standard as well. The mzData, mzML, and analysisXML data exchange formats are all implemented in XML (eXtensible Markup Language, <http://www.w3.org>). An advantage of XML is that the content is defined in an XML schema definition (XSD) file, allowing validation of actual XML file content using existing parsers. However, the complexity of the data content in PSI XML files precludes the validation of the

content in all data fields using only an XSD. Hence, more advanced validation tests are needed to ensure correct data representation.

Overview of test needs among ProDaC partners

A request for the need for example files and validation tools was sent out among the ProDaC partners. The response showed that some data files complementary to the PSI example files would be useful, and furthermore a validation tool accessible via the Internet would be very useful for standards implementers. There was little feedback regarding the choice of programming language for the validation tool, and as such any validation tool could be chosen as long as it is freely available. A platform-independent distributable tool would be beneficial as it could be used for validation in ongoing implementations by mass spectrometer vendors and in software packages such as Proteios, ProteinScape, Mascot, and Phenyx.

Generation of test files

As implementation of the mzML standard is being conducted, files corresponding to the present format are being made available on a Trac Wiki web site (<http://trac.thep.lu.se/trac/fp6-prodac> linked to from the official ProDaC website). This enables discussion of the format and possible issues, which are sent back to Work Package 1. The site is divided into areas for mzData, mzML, analysisXML, and PRIDE XML. Each area contains example files and editable documents where obstacles and ideas for developments of the standards are recorded.

Support for peak list conversion to the mzML format is under development for Proteios (<http://www.proteios.org>), and output files from this conversion tool are supplied on the fp6-prodac trac site. Sample files from the conversion tool ProCon (work package 3) are also included on the trac site.

Survey of existing validation tools

While there exist numerous implementations for XML validation with XSD schemas, the PSI standards also involve requirements that are not possible to validate with XSD schemas only. The extensive use of controlled vocabularies, counts, references and unique identifiers require specific implementations to validate the semantics in the file as well as the logical structure. For these purposes, the Proteomics Services Team at the EBI has developed a generic, extensible framework for the implementation of such validators for the PSI standards, as well as other XML formats. All of this code is available as open source under the Apache2 license on the PSI Subversion (svn) server. A prototype mzML validator has already been written using this framework, and a Molecular Interactions (PSI-MI) version is nearing completion.

The mzML version 0.93 release candidate package also contained a Perl-based schema validator for testing generated files. This validator could be extended with semantic control, but the current version only performs XSD validation, and is therefore meant as a simple means to verify structural correctness of an mzML file. A disadvantage is that some of the required libraries are not available for the Active State version of Perl which is the most common Microsoft Windows Perl engine, and that the validator will therefore require Cygwin or a similar Unix simulator in order to work under Windows. For these reasons, this tool can be considered as a part of a development kit for programmers rather than an end-user tool.

Creation of semantic validation tools

The EBI participants have built a prototype semantic validator for mzML using the generic framework they have developed. This validator is highly flexible and supports full separation of concerns according to well-defined roles. Software developers can leverage the power of custom-written validation rules, which are declaratively specified in an 'objectrules' XML file. Adding such

a rule consists simply of implementing the ‘ObjectRule’ interface, and specifying the rule in the XML file. The code of the validator proper thus need not be changed or updated at all. Ontology or domain experts on the other hand can use so-called ‘CV rules’ to validate the correct usage (and check for (dis-)allowed repetition) of ontology terms at any location in the XML file. These rules are completely declaratively defined in a ‘cvrules’ XML file, which again can be updated or extended using a simple text editor without having to alter any part of the validator or its code. Access to controlled vocabularies or ontologies is provided by the OntologyAccess framework that transparently supports both direct access to local or remote Open Biomedical Ontology (OBO) files, as well as access via the EBI’s Ontology Lookup Service (OLS; <http://www.ebi.ac.uk/ols>) API’s. This OntologyAccess framework has built-in caching functionalities to ensure performance. The choice between OLS or direct ontology access is again declarative and handled by a third and last XML file.

Finally, fast yet low-memory access to very large XML files (files of several gigabytes are to be expected in the case of mzML, for instance) is handled by the ‘XPathIndexedXMLParser’ framework, which indexes any XML file on-the-fly by xpath, allowing extremely fast random-access to an indexed file after an initial delay caused by the indexing.

Online validation of data files

To enable easy validation of files, and also possible logging of common errors in standard files, a web site for data file validation was set up. This site provides a front end to a general Java XML to XSD validator, and will provide a front-end to more advanced validators that include semantic tests as they become available. A screenshot of the online tools is shown in Figure 1.

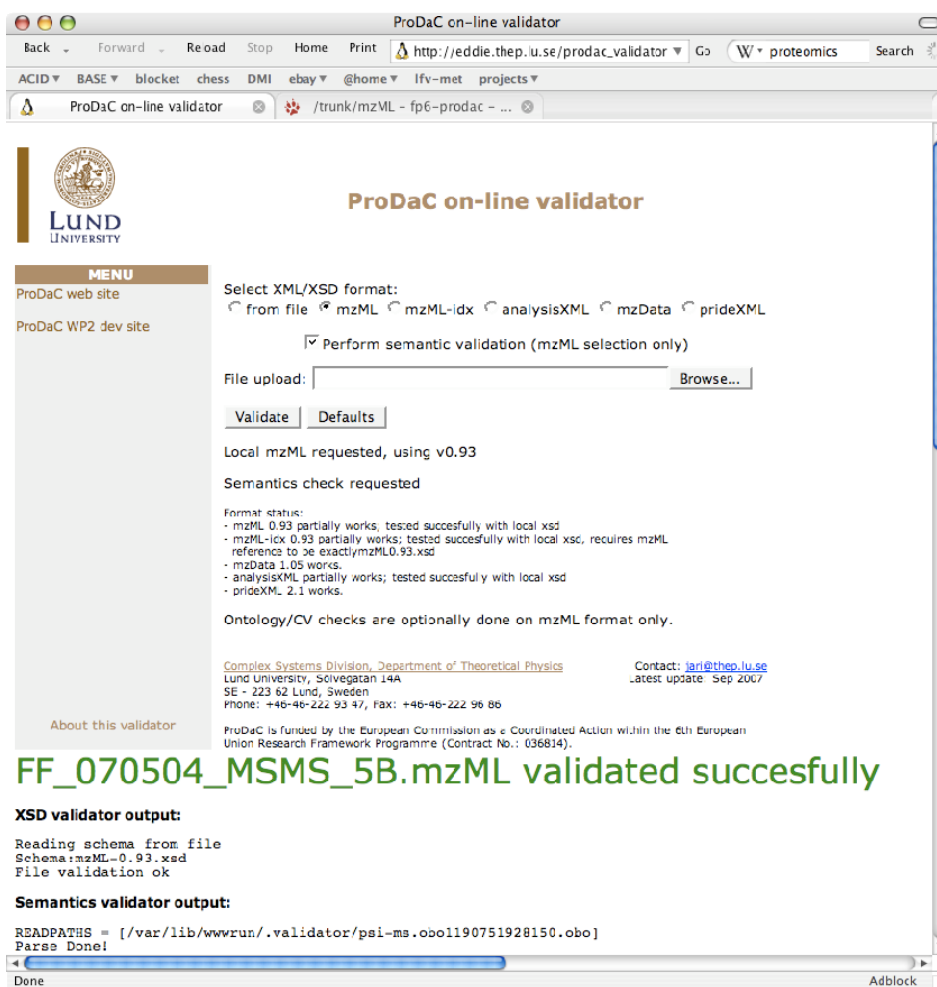


Figure 1: Screen shot of the ProDaC online validator.

Testing/Certification

To acquire ProDaC-certification, files generated by tools that pass the validator in its final form should be accepted by certified proteomics data repositories and storage tools. The validator enables the repository maintainers to work with well defined files and the validator puts pressure on tool vendors to adapt the standards as they mature to a level useful for validation. There are several standard compliant files (the test suite) at the <http://trac.thep.lu.se/trac/fp6-prodac> site for testing the ability of software to import these schematically and semantically valid data files.

In short, ProDaC-certification of a tool/repository comprises two criteria:

1. Files generated from a tool must pass the online validator described above, and if the tool imports standard files it should also pass the below item.
2. The test suite of files must be importable by the repository (or the tool). A stricter criterion is to require that a repository should import any file that passes the validator.

Conclusions

To enable for efficient implementation of standards compatible tools, a test suite was developed. A web interface which allows researchers and developers to verify the compliance of their data to the PSI standard mzML was included in the suite. The test suite and validation tools will be further adapted for application on other PSI standards as these standards are being finalised.

Funding

ProDaC is funded by the European Commission as a Coordinated Action within the 6th European Union Research Framework Programme (Contract No.: 036814).