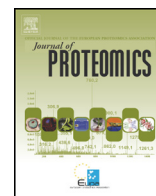




Contents lists available at ScienceDirect

Journal of Proteomics

journal homepage: www.elsevier.com/locate/jprot

ProCon – PROteomics CONversion tool

Gerhard Mayer^a, Christian Stephan^{a,b}, Helmut E. Meyer^{a,c}, Michael Kohl^a,
Katrin Marcus^a, Martin Eisenacher^{a,*}

^a Medizinisches Proteom Center (MPC), Ruhr-Universität Bochum, D-44801 Bochum, Germany

^b Kairos GmbH, Universitätsstraße 136, D-44799 Bochum, Germany

^c Leibniz-Institut für Analytische Wissenschaften, ISAS, e.V., D-44139 Dortmund, Germany

ARTICLE INFO

Article history:

Received 27 February 2015

Received in revised form 19 May 2015

Accepted 28 June 2015

Available online xxxx

Keywords:

Proteomics

Conversion tool

ProCon

mzIdentML

PRIDE

ProteomeXchange

ABSTRACT

With the growing amount of experimental data produced in proteomics experiments and the requirements/recommendations of journals in the proteomics field to publicly make available data described in papers, a need for long-term storage of proteomics data in public repositories arises. For such an upload one needs proteomics data in a standardized format. Therefore, it is desirable, that the proprietary vendor's software will integrate in the future such an export functionality using the standard formats for proteomics results defined by the HUPO-PSI group. Currently not all search engines and analysis tools support these standard formats. In the meantime there is a need to provide user-friendly free-to-use conversion tools that can convert the data into such standard formats in order to support wet-lab scientists in creating proteomics data files ready for upload into the public repositories. ProCon is such a conversion tool written in Java for conversion of proteomics identification data into standard formats mzIdentML and Pride XML. It allows the conversion of Sequest™/Comet .out files, of search results from the popular and often used ProteomeDiscoverer® 1.x (x = versions 1.1 to 1.4) software and search results stored in the LIMS systems ProteinScape® 1.3 and 2.1 into mzIdentML and PRIDE XML.

© 2015 Published by Elsevier B.V.

1. Introduction

In proteomics one has traditionally a plentitude of proprietary, vendor specific file formats [1]. In order to allow interoperability and to promote the development of sophisticated new software for reanalyzing proteomics search and quantitation results, the proteomics community lead by the HUPO-PSI (HUPO-Proteomics Standards Initiative) consortium developed modern and open XML-based standard formats together with software to process them [2]. These standard formats allow the storage of proteomics data in public repositories [3,4] like for instance

PRIDE [5] or PeptideAtlas [6] for reproducing and reanalyzing the results by other groups or with new or updated software tools or algorithms. Therefore it should be mandatory to store all the results of proteomics experiments into public repositories [7]. Furthermore, the assessment of the performance of newly developed software algorithms analyzing the data is made easier by having access to the data in such data repositories. These XML-based formats defined by the HUPO-PSI working group [8,9] incorporate also semantic annotations of the data by the use of controlled vocabulary terms from a set of ontologies [10,11]. The use of these ontology terms ensures that standardized terms are used and thereby avoids problems resulting from the usage of synonyms, the use of capital and small initial letters and misspellings. Furthermore the use of such standardized CV (Controlled Vocabulary) terms allows additional semantic correctness checks by validator software [12] and makes the file formats more flexible and thereby more stable, because it suffices to add new CV annotations, so that one must not always change the underlying XML schema in order to support new upcoming technologies or MS instruments.

To augment the adoption of these standard formats [13–16], some converter tools [17], libraries for Java [18–23] and other programming languages [24,25] for accessing data stored in these standard formats, and upload tools [26] have been developed and described in detail in [2,27]. There are also already some tools available, which read in such standard formats. For the identification results standard

Abbreviations: API, Application Programming Interface; CID, Collision Induced Dissociation; CSV, Comma Separated Values; CV, Controlled Vocabulary; EBI, European Bioinformatics Institute; ECD, Electron Capture Dissociation; ETD, Electron Transfer Dissociation; GUI, Graphical User Interface; GUID, Globally Unique Identifier; HUPO, Human Proteome Organization; JAXB, Java Architecture for XML Binding; JDBC, Java DataBase Connectivity; JRE, Java Runtime Environment; LC/MS, Liquid Chromatography/Mass Spectrometry; LIMS, Laboratory Information Management System; MSF, Mass Spec Format/Magellan Storage File; NEWT, NEW Taxonomy database of the Swiss-Prot group; PD, ProteomeDiscoverer®; PFF, Peptide Fragment Fingerprint; ProCon, Proteomics Conversion Tool; ProDac, Proteomics Data Collection; PS, ProteinScape®; PSI, Proteomics Standards Initiative; PURE, Protein Unit for Research in Europe; SAX, Simple API for XML; TSV, Tab Separated Values.

* Corresponding author at: Universitätsstraße 150, D-44801 Bochum, Germany.

E-mail address: martin.eisenacher@rub.de (M. Eisenacher).

mzIdentML [13] these are for instance the converter ProteoWizard [25], PeptideShaker [28] for the reanalysis of data sets and PRIDE Inspector [29], a program, which can be used for quality checking and viewing proteome identifications.

Here we describe the free to use ProCon software tool, which allows the conversion of Sequest®.out result files (Thermo Fisher Scientific Inc., Waltham, MA, USA), Proteome Discoverer®.msf (Thermo Scientific, Waltham, MA, USA) (Mass Spec Format) files, and of identification results obtained from the ProteinScape® 2.1 (Bruker Daltonik GmbH, Bremen, Germany) LIMS system into the mzIdentML standard format for peptide and protein identifications. In addition the conversion of the LIMS database contents of ProteinScape® 1.3 into PRIDE XML is possible.

The ProCon software is freely available from the MPC (Medizinisches Proteom Center, Bochum, Germany) website at <http://www.medizinisches-proteom-center.de/ProCon>.

2. Materials and methods

ProCon is a proteomics conversion tool written in the platform independent Java SE (Standard Edition) 7 programming language. The Software comprises a GUI (Graphical User Interface), which is intended to allow laboratory personnel without access to a supporting bioinformatics unit to convert their peptide and protein identification results of proteomics experiments into formats suitable for submitting them to data repositories like for instance PRIDE [5] and PeptideAtlas [6]. The upload of these converted files can then be done by the standardized data submission pipeline developed by the ProteomeXchange [26] consortium (<http://www.proteomexchange.org>) for initial upload and automatic data exchange between the main proteomics data repositories. For the GUI of ProCon the Swing classes of the Java standard edition (<http://docs.oracle.com/javase/tutorial/uiswing/>) were used. The ProCon development was started in the Proteomics Data Collection (ProDaC) project [30] and continued as part of the ProteomeXchange project [26] (<http://www.proteomexchange.org>).

2.1. ProCon functionality

ProCon itself is implemented in Java 7, tested with Java 7 and 8 and possesses a modular structure consisting of 8 jar files: Some basic constants and functionality used by the other .jar files are defined within the BaseLib.jar, CvtBaseLib.jar and CvtLib.jar libraries. Access to ProteinScape® 2.1 is implemented in the PSLibrary.jar file and functionality for accessing the ProteomeDiscoverer® database is encapsulated in the PDLlibrary.jar file. The converters for ProteinScape® 2.1 and ProteomeDiscoverer® results are contained in the ConvertPS2MzIdent1.1.jar resp. ConvertProt2MzIdent1.1.jar files. The converters for Sequest/Comet.out files and ProteinScape® 1.3 are contained in the ProCon.jar file, which also implements the GUI of the ProCon application. This modular structure eases the further development and modification of the software. Besides that ProCon can also be started from the command line as described in the user manual, which resides in the documentation folder. By this command line interface ProCon can be called by batch processes, e.g. from cron jobs. Furthermore the command line support is a prerequisite for integration into workflow environments like KNIME [31] in the future.

The PSLibrary.jar and PDLlibrary.jar can also be used independently from ProCon, for instance for implementing viewers or other converters, e.g. for mzQuantML or other analysis software accessing the results contained in ProteinScape® 2.1 resp. ProteomeDiscoverer®.

2.1.1. Conversion of Sequest/Comet .out files into mzIdentML

For converting the Sequest.out files [32] ProCon parses the result files of a Sequest search run from the resulting HTML web pages using the HTML parser library (<http://htmlparser.sourceforge.net>) and converts the information about the found peptides and proteins into the

mzIdentML format. At first, a folder, which contains all the Sequest.out files intended for conversion, has to be selected. Then all the Sequest parameters are read in from the sequest.params parameter file, and all the .out files found under the folder tree of the specified folder are parsed. After specifying the name and the location of the mzIdentML output file to be generated, the conversion process is started.

For specifying information not contained in the Sequest .out files, but required for the .mzid files, e.g. the contact information for the 'AuditCollection' element of mzIdentML, there are XML templates residing inside the /config folder, which can be edited by the user in order to specify the 'ContactRole', the 'Person' and 'Organization' information of the researcher generating the mzIdentML file.

Since Comet [33] is an open-source implementation that can also export .out files like Sequest and has a similar .params file, we built in a radio button, which allows one also to convert such .out output folders from the Comet search engine.

2.1.2. Conversion of ProteinScape® 1.3 search results into PRIDE XML

For the conversion of ProteinScape® 1.3 data, the PFF (Peptide Fragment Fingerprint) and PMF (Peptide Mass Fingerprint) search results are converted into the PRIDE XML format, which is specified by the XML Schema definition file obtained from http://code.google.com/p/ebi-pride/source/browse/trunk/pride-web/src/main/webapp/help_resources/pride.xsd. The converter can be used either to convert the results of a single search event of ProteinScape® 1.3, i.e. the results of one search engine run, or the conversion of all search events of a complete gel into PRIDE XML. In the latter case, the program loops over all spots of the gel. During this conversion process results obtained from the SloMo tool [34] can be used for integration of additional information extracted from ETD/ECD or CID mass spectra about phosphorylation modifications (localization and confidence scores) into the PRIDE XML file. To this end specially formatted CSV (Comma Separated Values) columns are loaded during the ProteinScape® 1.3 conversion.

For consideration of modifications used by ProteinScape® 1.3 and Sequest two .obo (Open Biomedical Ontologies) [35] files were created and stored in the /config folder. The access to these .obo files is implemented in the 'OBOMapper' class, which makes use of the functionality provided by the org.geneontology.jar library. These modifications of definitions are then mapped to the corresponding PRIDE resp. PSI-MS CV [9] (Controlled Vocabulary) terms.

The conversion to the PRIDE XML format is done by instantiating objects from the PRIDE_core_2_5_4.jar library from the EBI (European Bioinformatics Institute). This library contains implementations for all the elements of a PRIDE XML file. For instance the object ExperimentImpl stands at the top of the PRIDE XML hierarchy and can be used to instantiate an experiment, which in turn contains further elements subordinated to it, describing among others the 'Protocol', the 'mzData', the 'GelFreeIdentifications' and the 'TwoDimensionalIdentifications' elements of PRIDE XML.

For the instrument description of mzData an own object 'mzData_instrumentImpl' was defined together with a JAXB (<http://docs.oracle.com/javase/6/docs/api/javax/xml/bind/JAXB.html>, Java Architecture for XML Binding) unmarshaller. This unmarshaller allows reading in the information describing the used instruments from preconfigured XML files contained in the /instruments folder into main memory. This template can be used for defining own instrument files, describing the MS instrument of the user. These instrument files are loaded by ProCon and then integrated into the generated PRIDE XML file. By using these instrument files the application is held flexible for the incorporation of information describing emerging new and future instrument developments.

Also other information not contained in ProteinScape® 1.3 can be added during the conversion process by using a file based approach, for instance for adding the bibliographic references: the user can edit the references.properties template in the /config folder and add all his

bibliographic references there. This .properties file is read in and the information from there is added to the generated PRIDE XML file.

2.1.3. Conversion of ProteinScape® 2.1 search results into mzIdentML

For conversion of the ProteinScape® 2.1 [36,37] search results we developed a Java API for accessing the stored results.

Furthermore, there is information about possible Unimod (<http://www.unimod.org/obo/unimod.obo>) modifications and for instance data about protocols and parameters used by the ProteinExtractor algorithm [36,37] for combining the search engine scores of Mascot [38], Phenix [39], Sequest [32,40], ProteinSolver [41], ProFound [42], MS-Fit [43] and Sonar [44] into a meta-score for the peptide identifications. With respect to protein inference, ProteinExtractor combines the results of the search engines Mascot, Phenix, Sequest and ProteinSolver into a meta-score [41].

The tree control of the PS 2.1 client software shows the data and methods supported by PS 2.1. The data is hierarchically organized into projects and each project can include several samples. Each sample includes different examination methods e.g. fractions, digestions, gels, spots, and LC/MS analyses. The identified spectra and search events containing the results of the database searches in turn reside below these examination methods. The inheritance structure of our API classes (Fig. 1) reflects the tree control hierarchy as depicted by PS 2.1. By using the Java API the tree of the selected project is traversed and the user can interactively choose a sample and a single search event belonging to the chosen project (Fig. 2). If the user wants to convert all search events belonging to a gel, he can instead select the radio button “Convert gels” and choose a gel for the selected project–sample combination. Then all search events are automatically listed in the search events table and included in the conversion process.

The corresponding classes of our API that depict different MS examination methods inherit from the ‘AnalyteBaseClass’ and implement the ‘Analytable’ interface (see Fig. 1). All the 8 classes corresponding to the elements of the tree control shown on the left side of the PS 2.1 client software implement the ‘Navigatable’ interface. Such ‘Navigatable’ classes are for instance the different kinds of spectra, compounds (spectra packets) and the so called search events. A PS search event contains the protein and peptide identification results of a single search engine run. These search events are located under the corresponding spectrum in the GUI tree. In addition, under ‘Analyte’ classes there may be search events for the resulting protein result lists resulting from ProteinExtractor [34,35] runs (combining the results of the Mascot [38], Phenix [39], Sequest [32,40] and ProteinSolver [41] algorithms).

The Java class ‘PS2ProgramParams’ contains program parameters and information entered by the user via the ProCon GUI (e.g. the parameter set used for isoelectric point calculation, the location of the output .mzid file and the contact data about the research organization). By using this information the results from ProteinScape® are enriched in order to include data not contained in PS, but which can be included into the resulting mzIdentML file. During conversion the conversion progress is shown to the user by a progress bar.

The class ‘MzIdentMLRefs’ contains hash tables for storing all elements, which are cross-referenced in the mzIdentML file under construction, accessible via their unique IDs. By this a duplicated read from the conversion results is avoided in order to improve the performance. After the whole mzIdentML object is build up, it is ultimately written out in the ‘MzIdentMLParts’ class, by making use of the JAXB marshaller functionality.

During the conversion process the needed information of the CV terms for semantic annotation of the mzIdentML data [13] and the NEWT (NEW Taxonomy database) [45] taxonomy identifiers used are

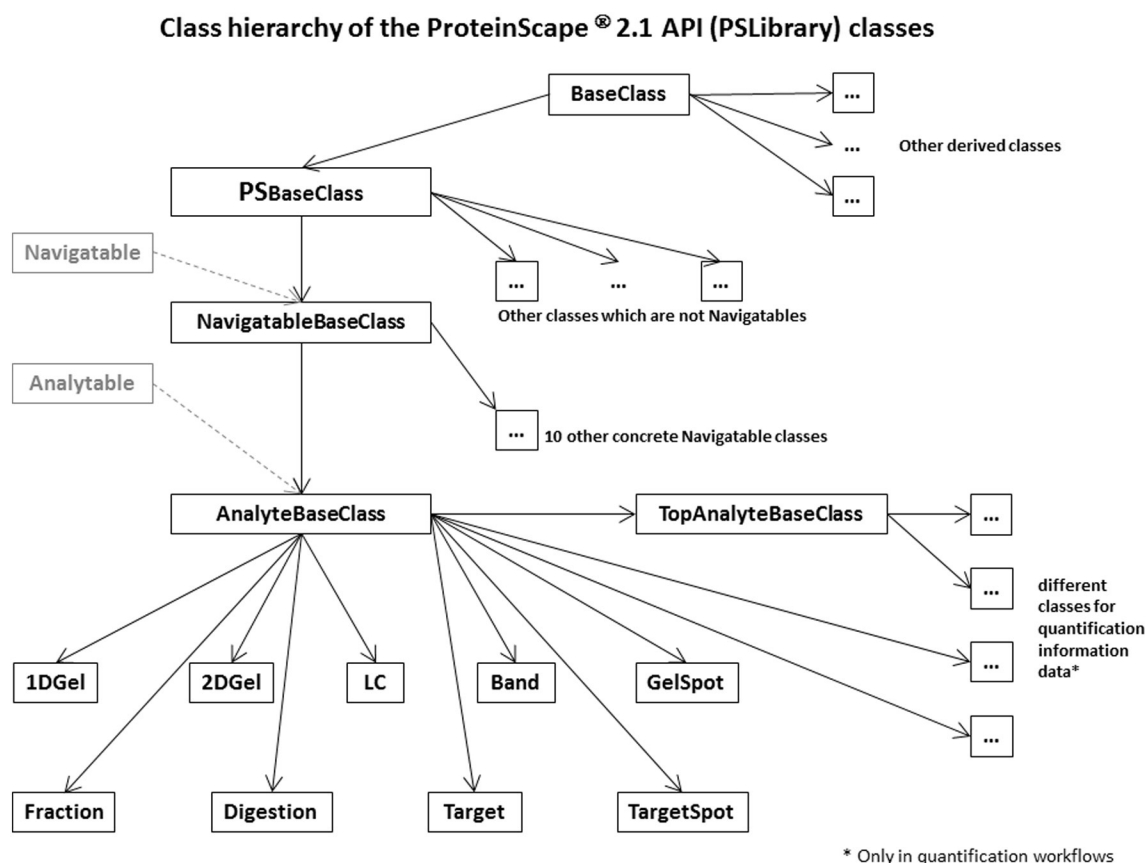


Fig. 1. Class hierarchy of the PSLibrary API classes of ProCon: This API reflects the class hierarchy of the PS 2.1 client software and contains 16 Analytable classes, 24 Navigatable classes and in total counts 226 classes.

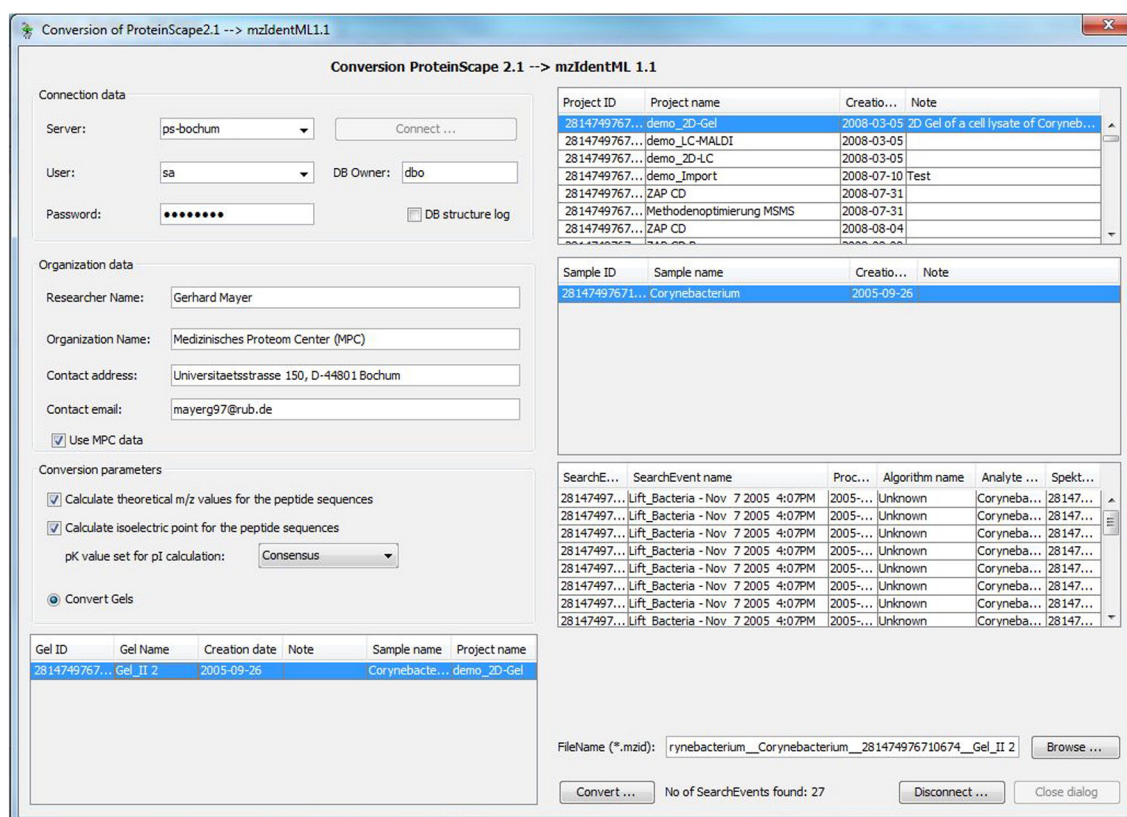


Fig. 2. The GUI of the converter is implemented in the class 'PS_mzIdentML_Dialog'. After entering the connection information, the organization data and the conversion parameters on the left side one can connect to the information stored in ProteinScape® 2.1 and then first select a project, then a sample and at last a search event, i.e. the identification result of a search event run on the right side. Then a file name for the location of the resulting .mzid file is proposed, which the user can alter if needed. After that the user can start the conversion process.

obtained from the corresponding .obo files via the internet. If there is no internet access to the ontology resource available, the CV terms are acquired instead by parsing in the needed information from local ontology files contained in the resources folder. To improve performance we use hash tables to accelerate the access to already read-in CV term information, so that the ontology files must be read in only once.

The isoelectric point calculation is done by iteratively solving the Henderson–Hasselbalch equation [46]. The algorithm calculates the partial charges for each of the charged residues (C, D, E, H, K, R, and Y) and the terminal carboxyl and amino group and adds them up. The isoelectric point is defined as the pH value at which this total charge is zero. Due to the monotone dependence of the charges on the pH value one can determine the zero point by an efficient linear interpolation algorithm, so that the result is found after about 7–8 iterations. Due to the dependency of the calculated pI value from the used pKa value set we also compute an averaged pI over the different pKa value sets as a consensus pI value. This algorithm is also used in the standalone Peptide Property Calculator, which is integrated into the Tools menu of ProCon and which allows one to calculate the isoelectric point (pI value) and the molecular weight for an entered (uncharged) peptide.

2.1.4. Conversion of ProteomeDiscoverer® 1.x (x = 1, 2, 3, 4) search results into mzIdentML

The ProteomeDiscoverer® (PD) result files are given as SQLite (<http://www.sqlite.org>) flat file databases [47].

The PD converter GUI allows one to specify information not contained in PD e.g. the researchers' affiliation information and the sample name (Fig. 3) and to set parameters for the conversion. Analogous to the ProteinScape® converter the PD converter contains two classes 'PDProgrammParameters' and 'ConversionProgress' for exchanging parameters resp. conversion progress information with the GUI.

Since result files from ProteomeDiscoverer® versions 1.1 and 1.2 contain no protein inference information, e.g. no information about the assignment of the proteins to the protein ambiguity groups, one has to specify also the .prot.xml file in order to obtain the missing information. After entering the .msf file, ProCon reads in the PD version and disables the input field for the .prot.xml file for versions 1.3 and above. For reading in the .prot.xml file the converter contains the class 'ProtXMLHandler' in the package de.mpc.PD.XML, which implements a SAX (Simple API for XML, <http://docs.oracle.com/javase/tutorial/jaxp/sax/>) parser for retrieval of information contained in the .prot.xml file. Besides that the PDLibrary.jar implements also a parser for .pep.xml files in the class 'PepXMLHandler'. Because the PD result files may sometimes contain XML files stored inside text fields. Therefore the PDLibrary.jar contains also XML Handlers ('FDRSettingsHandler', 'PDWorkflowHandler', and 'ResultFilterSetHandler') for reading these embedded XML files via JAXB. For instance the 'PDWorkflow' is an XML description of the used ProteomeDiscoverer workflow.

The current ProteomeDiscoverer® (PD) converter has the advantage that it works with the versions 1.1, 1.2, 1.3 and 1.4 of PD, although the information about ProteinDetectionLists are not contained in the PD 1.1 and 1.2 result files. Regarding conversions of very large PD 1.1 and 1.2 result files, resulting e.g. from the use of large search databases, the conversion process can be very slow, because the matching of the information from the .prot.xml file with the information contained in the PD result file is computationally very demanding: internally there are sequences matched with SQL string matching procedures in order to find the peptides in the identified proteins. This technique shows a very poor performance. Another drawback is that in the .prot.xml file the matches of decoys are not reported, so that the resulting mzIdentML files from PD versions 1.1 and 1.2 remains incomplete with respect to matching so called decoy sequences. This decoy information is used to control the rate of incorrect peptide and protein identifications [48] by

Fig. 3. The GUI of the ProteomeDiscoverer® converter asks the user first for the locations of the input and output files, the users contact data and the conversion parameters. After that the conversion into the mzIdentML file can be started by pressing the 'OK' button.

taking matches between spectra and shuffled, random or reverse decoy sequences [49] into account, which can be used to filter the results according to a specified false discovery rate. In order to prevent these disadvantages it is strongly recommended to use version 1.3 of PD or above.

Because the mzIdentML format makes extensive use of internal references to avoid information redundancy, the converter for PD versions 1.3 and later internally uses a 'MzIdentMLRefs' class for all cross-referenced elements of the assembled mzIdentML file analogously to the PS converter. This class eases the administration of the references inside the mzIdentML file under construction so that the number of result files accesses is kept at a minimum, what enhances the performance of the converter. The marshalling of the created mzIdentML file via JAXB is again implemented in the 'MzIdentMLParts' class.

There are currently no plans to support also conversions for the output obtained from the older PD version 1.0, but the support for newly upcoming versions of PD would be of course desirable and is planned as far as the general data result file format is not changed and remains non-proprietary and therefore accessible.

3. Comparison with the M2Lite converter

There is another converter (M2Lite, <https://bitbucket.org/paiyetan/m2lite>) for ProteomeDiscoverer.msf files already published. The publication in [50] mentions one big advantage of M2Lite over ProCon: much better performance and a lower memory footprint. We converted an example file (test_completeSet_MK.msf, see the Supplementary material) with both converters in order to compare the two converter programs according their runtime behavior, their outcome and the

ease of handling for the end user. We used ProCon 0.9.571 and the latest version of M2Lite3 (M2Lite3.2015.01.27.zip) for this comparison.

In fact, ProCon was much slower than M2Lite3 (15:29 min for ProCon against 4:39 min for M2Lite on an Intel Core™2Duo PC E8600 @3.33 GHz with 8 GB RAM under Windows 7, 64 Bit with Java 1.7.0_75). This means the runtime factor of 3.3 (resp. 3.5 after replacing the outdated sqLitejdbc-v056.jar of M2Lite3 by the newest sqLite-jdbc-3.8.7.jar) clearly favors M2Lite. In [50] it is mentioned that M2Lite was designed to retrieve the information via delegation to R, which is much more performant for SQL calls than the Java JDBC. Furthermore M2Lite uses a special swap directory for minimizing the memory footprint.

On the other hand several shortcomings of M2Lite compared with ProCon became obvious. We got the test_completeSet_MK.mzid output file from the M2Lite conversion and compared it with the corresponding output from ProCon. One drawback of M2Lite is for example that it can convert only PD 1.3 and 1.4 files. Disadvantageous is also the requirement to have the .fasta file at hand. The PRIDE database [5] at the moment contains lot of .msf files, but mostly the used .fasta file is not stored there. M2Lite does not process results obtained from multiple search engines and supports only search databases in the deprecated IPIv387 (International Protein Index) and the REFSEQ [51] formats. In addition it doesn't calculate the isoelectric point information for the peptides and at the moment does not support the generation of a ProteinDetectionList. Another drawback of M2Lite is that it is much more difficult to configure for the end user and has no GUI interface available. A summary of all identified shortcomings of M2Lite compared to ProCon is given in Table 1.

4. Results and discussion

Besides the ongoing enhancements concerning stability and performance improvements, a couple of additional functionalities which can be added in future to the ProCon program are conceivable.

Because both PD and PS support also quantification workflows, another important feature of future ProCon versions would be the integration of conversions into the mzQuantML [14] and/or mzTab [52] formats. Also the integration of a converter for spectral counting [53, 54] information represented in Excel files into mzQuantML is a possible feature to be considered for integration into ProCon in the future. At the moment a prototypical converter for such a CSV (Comma Separated Values) or TSV (Tab Separated Values) spectral counting file into mzQuantML conversion is available at the <https://code.google.com/p/tsv-or-csv-mzquantml-converter> web site.

Another point currently not well addressed is performance optimization of the used SQL queries, which can drastically reduce the runtime of a conversion, especially for the conversion of very large .msf files, which e.g. result from the use of large search databases. We made some performance statistics of the ProteomeDiscoverer converter in order to see how the conversion time depends on different factors e.g. file size, number of spectra, number of peptides and number of proteins. As shown in the file ProCon_paper_bar_charts.xlsx in the Supplementary material, there is no clear evidence for a single factor to which the run time can be attributed. The time for conversion without including the ProteinDetectionList is nearly the same as with ProteinDetectionList. The reason for this behavior is that our algorithm was optimized for generating a complete .mzid file including the ProteinDetectionList. In case the user doesn't want to include this list in the output file, then only the list output is omitted.

We tested the conversion of some large files on both the dual core desktop PC with 8 GByte RAM and a virtual dual core machine with comparable compute power with 32 GByte RAM. According to our tests the conversion performance was not memory-bound, even when the memory usage on the virtual machine was beyond the 8 GByte limit. So we conclude that the conversion performance is not primarily memory bound.

An option to optimize the runtime performance and minimize the memory footprint of ProCon would be surely an approach making use of R and a swap directory in analogy to the M2Lite strategy [50], but this would mean that the ease of use would be lost due to additional configuration required by the end user.

A medium-term goal is to make ProCon available as a KNIME [31] (Konstanz Information Miner) workflow system (<http://www.knime.org>) node, so that it can easily integrated into proteomics workflows.

Table 1
Identified shortcomings of M2Lite [50] compared with ProCon.

M2Lite is more complicated to use for end-users, because:
– It needs the search database (.fasta file)
– It has no Graphical User Interface (GUI)
– It requires the R statistics system http://www.r-project.org (and the RSQLite package, http://cran.r-project.org/web/packages/RSQLite/RSQLite.pdf) and is therefore more complicated to configure
M2Lite missing features/functionality:
– Supports only the databases in IPIv387 (deprecated, http://www.ebi.ac.uk/IPI) and REFSEQ [51] format, whereas ProCon supports .fasta files from any database
– Supports only PD versions 1.3 and 1.4
– Does not support multiple search engines
– No calculated isoelectric point in <SpectrumIdentificationItem> elements
– No amino acid sequences in <DBSequence> elements
– Less CV annotations; often uses <userParam> elements instead of <cvParam> elements
– The <ProteinDetectionList> element is missing (still commented out in the source code)
– References not the latest version of the psi-ms.obo ontology
– No platform independent newline support (only Linux style \n)

A first step towards this direction was already done by integrating batch processing support into ProCon. In this batch mode the special command line parameter `-conv` specifies the converter program and must be one of the values PD1x, PS13, PS21, or SEQO for selecting the appropriate converter program. The other information normally specified in the GUI, can be specified via command-line parameters. The names of these command line parameters are specified in the ProCon user manual. There are mandatory and optional command line parameters and if an optional parameter is not specified, then the given default value is used. The GUI of ProCon is made invisible in batch mode and the values of the command line parameters are set programmatically in this invisible GUI and then the conversion is started, so both the GUI mode and the batch mode can share the same code for the conversion. If an error occurs, the error message is handled by an overwritten MessageBox, which in case of batch mode logs the message to the console instead of showing the MessageBox. In contrast to ProCon M2Lite works only in batch mode and has no GUI capabilities [50].

Another future plan is to convert partial ProteomeXchange submissions of ProteomeDiscoverer.msf files already contained in the PRIDE [5] database into complete submissions. For this we started cooperating with the PRIDE team from the EBI.

5. Conclusions

ProCon is a conversion tool which complements other converters e.g. PRIDE Converter 2 [17] and ProteoWizard [25]. But these tools don't support the conversion of ProteomeDiscoverer® and ProteinScape® result at the moment and also the M2Lite converter [50] has still its weaknesses. ProCon makes the conversion of proteomics data of PD and PS into the standard formats mzIdentML [11] resp. PRIDE XML easy even for laboratories without access to specialized bioinformatics expertise. Thereby ProCon can assist in preparing data submission to public repositories as recommended in the paper publication guidelines of major proteomics journals (http://www.mcponline.org/site/misc/ParisReport_Final.xhtml) [55,56]. By supporting the conversion of data into such standard formats, ProCon can help to ease the public and free community data access to proteomics data stored in public repositories, connected with a plethora of benefits for the scientific community. For instance this allows the reproduction of results for critical assessment, the reanalysis of data with new methods, algorithms and/or software and addressing new upcoming research questions, so that comparisons of the results of different methods/algorithms are possible. Thus meta-analyses of proteomics data and the integration of these data with data from other 'omics' fields (transcriptomics, metabolomics, interactomics, glycomics, lipidomics, ...) [57] are alleviated and it can be expected that this will lead to new insights regarding signaling cascades, functional disease mechanisms and the identification of new biomarkers and/or therapeutic drug targets [58].

Conflict of interest

All authors declare that they have no financial/commercial conflicts of interest.

Acknowledgments

The development of ProCon was funded by the European Union projects ProDaC (<http://www.fp6-prodac.eu>, EU FP6 grant LSHG-CT-2006-036814), and ProteomeXchange (<http://www.proteomexchange.org>, EU FP7 grant number 260558), the Deutsche Gesetzliche Unfallversicherung (DGUV) project DGUV-Lunge (617.0 FP 339A), P.U.R.E. (<http://www.pure.rub.de>), a project of Nordrhein-Westfalen, a federal state of Germany, and the de.NBI (<http://www.denbi.de>) project funded by the German Federal Ministry of Education and Research (BMBF), grant number FKZ 031 A 534A. We thank Phil Jones from the EBI (<http://www.ebi.ac.uk>) for making available the PRIDE_core_2.5.4.jar library.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <http://dx.doi.org/10.1016/j.jprote.2015.06.015>.

References

- [1] E.W. Deutsch, File formats commonly used in mass spectrometry proteomics, *Mol. Cell. Proteomics* 11 (2012) 1612–1621.
- [2] F.F. Gonzalez-Galarza, D. Qi, J. Fan, C. Bessant, A.R. Jones, A tutorial for software development in quantitative proteomics using PSI standard formats, *Biochim. Biophys. Acta* 2014 (1844) 88–97.
- [3] Y. Perez-Riverol, E. Alpi, R. Wang, H. Hermjakob, J.A. Vizcaino, Making proteomics data accessible and reusable: current state of proteomics databases and repositories, *Proteomics* 15 (2015) 930–950.
- [4] M. Riddle, J.K. Eng, Proteomics data repositories, *Proteomics* 9 (2009) 4653–4663.
- [5] J.A. Vizcaino, R. Coté, F. Reisinger, H. Barsnes, J.M. Foster, J. Rameseder, et al., The Proteomics Identifications database: 2010 update, *Nucleic Acids Res.* 38 (2010) D736–D742.
- [6] E.W. Deutsch, H. Lam, R. Aebersold, PeptideAtlas: a resource for target selection for emerging targeted proteomics workflows, *EMBO Rep.* 9 (2008) 429–434.
- [7] Y. Perez-Riverol, H. Hermjakob, O. Kohlbacher, L. Martens, D. Creasy, J. Cox, et al., Computational proteomics pitfalls and challenges: HavanaBioinfo 2012 workshop report, *J. Proteomics* 87 (2013) 134–138.
- [8] S. Orchard, Data standardization and sharing—the work of the HUPO-PSI, *Biochim. Biophys. Acta* 1844 (1 Pt A) (2014) 82–87.
- [9] E.W. Deutsch, J.P. Albar, P.A. Binz, M. Eisenacher, A.R. Jones, G. Mayer, et al., Development of data representation standards by the human proteome organization proteomics standards initiative, *J. Am. Med. Inform. Assoc.* 22 (2015) 495–506.
- [10] G. Mayer, A.R. Jones, P.A. Binz, E.W. Deutsch, S. Orchard, L. Montecchi-Palazzi, et al., Controlled vocabularies and ontologies in proteomics: overview, principles and practice, *Biochim. Biophys. Acta* 1844 (2014) 97–107 [Part A].
- [11] G. Mayer, L. Montecchi-Palazzi, D. Ovelheiro, A.R. Jones, P.A. Binz, E.W. Deutsch, et al., The HUPO proteomics standards initiative-mass spectrometry controlled vocabulary, *Database (Oxford)* 2013 (2013) 1–13 (2013:bat009).
- [12] F. Ghali, R. Krishna, P. Lukasse, S. Martinez-Bartolome, F. Reisinger, H. Hermjakob, et al., Tools (Viewer, Library and Validator) that facilitate use of the peptide and protein identification standard format, termed mzIdentML, *Mol. Cell. Proteomics* 12 (2013) 3026–3035.
- [13] A.R. Jones, M. Eisenacher, G. Mayer, O. Kohlbacher, J. Siepen, S.J. Hubbard, et al., The mzIdentML data standard for mass spectrometry-based proteomics results, *Mol. Cell. Proteomics* 11 (2012) (M111 014381).
- [14] M. Walzer, D. Qi, G. Mayer, J. Uszkoreit, M. Eisenacher, T. Sachsenberg, et al., The mzQuantML data standard for mass spectrometry-based quantitative studies in proteomics, *Mol. Cell. Proteomics* 13.10 (2014) 2765–2775.
- [15] L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, et al., mzML—a community standard for mass spectrometry data, *Mol. Cell. Proteomics* 10 (2011) (R110 000133).
- [16] E.W. Deutsch, M. Chambers, S. Neumann, F. Levander, P.A. Binz, J. Shofstahl, et al., TraML—a standard format for exchange of selected reaction monitoring transition lists, *Mol. Cell. Proteomics* 11 (R111) (2012) 015040.
- [17] R.G. Coté, J. Griss, J.A. Dienes, R. Wang, J.C. Wright, H.W. van den Toorn, et al., The PRoteomics IDentification (PRIDE) Converter 2 framework: an improved suite of tools to facilitate data submission to the PRIDE database and the ProteomeXchange consortium, *Mol. Cell. Proteomics* 11 (2012) 1682–1689.
- [18] R.G. Coté, F. Reisinger, L. Martens, jnzML, an open-source Java API for mzML, the PSI standard for MS data, *Proteomics* 10 (2010) 1332–1335.
- [19] K. Helsens, M.Y. Brusniak, E. Deutsch, R.L. Moritz, L. Martens, jTraML: an open source Java API for TraML, the PSI standard for sharing SRM transitions, *J. Proteome Res.* 10 (2011) 5260–5263.
- [20] F. Reisinger, R. Krishna, F. Ghali, D. Ríos, H. Hermjakob, J.A. Vizcaino, et al., jnzIdentML API: a Java interface to the mzIdentML standard for peptide and protein identification data, *Proteomics* 12 (2012) 790–794.
- [21] D. Qi, R. Krishna, A.R. Jones, The jnzQuantML programming interface and validator for the mzQuantML data standard, *Proteomics* 14 (2014) 685–688.
- [22] J. Griss, F. Reisinger, H. Hermjakob, J.A. Vizcaino, jnzReader: a Java parser library to process and visualize multiple text and XML-based mass spectrometry data formats, *Proteomics* 12 (2012) 795–798.
- [23] H. Barsnes, M. Vaudel, N. Colaert, K. Helsens, A. Sickmann, F.S. Berven, et al., compomics-utilities: an open-source Java library for computational proteomics, *BMC Bioinforma.* 12 (2011) 70.
- [24] T. Bald, J. Barth, A. Niehues, M. Specht, M. Hippler, C. Fufezan, pymzML—Python module for high-throughput bioinformatics on mass spectrometry data, *Bioinformatics* 28 (2012) 1052–1053.
- [25] D. Kessner, M. Chambers, R. Burke, D. Agus, P. Mallick, ProteoWizard: open source software for rapid proteomics tools development, *Bioinformatics* 24 (2008) 2534–2536.
- [26] J.A. Vizcaino, E.W. Deutsch, R. Wang, A. Csordas, F. Reisinger, D. Ríos, et al., ProteomeXchange provides globally coordinated proteomics data submission and dissemination, *Nat. Biotechnol.* 32 (2014) 223–226.
- [27] Y. Perez-Riverol, R. Wang, H. Hermjakob, M. Muller, V. Vesada, J.A. Vizcaino, Open source libraries and frameworks for mass spectrometry based proteomics: a developer's perspective, *Biochim. Biophys. Acta* 2014 (1844) 63–76.
- [28] M. Vaudel, J.M. Burkhardt, R.P. Zahedi, E. Oveland, F.S. Berven, A. Sickmann, et al., PeptideShaker enables reanalysis of MS-derived proteomics data sets, *Nat. Biotechnol.* 33 (2015) 22–24.
- [29] R. Wang, A. Fabregat, D. Ríos, D. Ovelheiro, J.M. Foster, R.G. Coté, et al., PRIDE Inspector: a tool to visualize and validate MS proteomics data, *Nat. Biotechnol.* 30 (2012) 135–137.
- [30] M. Eisenacher, L. Martens, T. Hardt, M. Kohl, H. Barsnes, K. Helsens, et al., Getting a grip on proteomics data — Proteomics Data Collection (ProDaC), *Proteomics* 9 (2009) 3928–3933.
- [31] W.A. Warr, Scientific workflow systems: Pipeline Pilot and KNIME, *J. Comput. Aided Mol. Des.* 26 (2012) 801–804.
- [32] J.K. Eng, A.L. McCormack, J.R. Yates, An approach to correlate tandem mass-spectral data of peptides with amino-acid-sequences in a protein database, *J. Am. Soc. Mass Spectrom.* 5 (1994) 976–989.
- [33] J.K. Eng, T.A. Jahan, M.R. Hoopmann, Comet: an open-source MS/MS sequence database search tool, *Proteomics* 13 (2013) 22–24.
- [34] C.M. Bailey, S.M. Sweet, D.L. Cunningham, M. Zeller, J.K. Heath, H.J. Cooper, SLoMo: automated site localization of modifications from ETD/ECD mass spectra, *J. Proteome Res.* 8 (2009) 1965–1971.
- [35] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, et al., The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration, *Nat. Biotechnol.* 25 (2007) 1251–1255.
- [36] H. Thiele, J. Glandorf, P. Hufnagel, G. Korting, M. Bluggel, Managing proteomics data: from generation and data warehousing to central data repository, *J. Proteomics Bioinform.* 01 (2008) 485–507.
- [37] H. Thiele, J. Glandorf, P. Hufnagel, Bioinformatics strategies in life sciences: from data processing and data warehousing to biological knowledge extraction, *J. Integr. Bioinform.* 7 (2010) 141.
- [38] D.N. Perkins, D.J.C. Pappin, D.M. Creasy, J.S. Cottrell, Probability-based protein identification by searching sequence databases using mass spectrometry data, *Electrophoresis* 20 (1999) 3551–3567.
- [39] J. Colinge, A. Masselot, I. Cusin, E. Mahe, A. Niknejad, G. Argoud-Puy, et al., High-performance peptide identification by tandem mass spectrometry allows reliable automatic data processing in proteomics, *Proteomics* 4 (2004) 1977–1984.
- [40] J.K. Eng, B. Fischer, J. Grossmann, M.J. Maccoss, A fast SEQUEST cross correlation algorithm, *J. Proteome Res.* 7 (2008) 4598–4602.
- [41] D.C. Chamrad, G. Korting, K. Stuhler, H.E. Meyer, J. Klose, M. Bluggel, Evaluation of algorithms for protein identification from sequence databases using mass spectrometry data, *Proteomics* 4 (2004) 619–628.
- [42] W.Z. Zhang, B.T. Chait, Profound: an expert system for protein identification using mass spectrometric peptide mapping information, *Anal. Chem.* 72 (2000) 2482–2489.
- [43] K.R. Clauser, P. Baker, A.L. Burlingame, Role of accurate mass measurement (+/–10 ppm) in protein identification strategies employing MS or MS/MS and database searching, *Anal. Chem.* 71 (1999) 2871–2882.
- [44] H.I. Field, D. Fenyo, R.C. Beavis, RADARS, a bioinformatics solution that automates proteome mass spectral analysis, optimises protein identification, and archives data in a relational database, *Proteomics* 2 (2002) 36–47.
- [45] I.Q. Phan, S.F. Pilboud, W. Fleischmann, A. Bairoch, NEWT, a new taxonomy portal, *Nucleic Acids Res.* 31 (2003) 3822–3823.
- [46] G. Henriksson, A.K. Englund, G. Johansson, P. Lundahl, Calculation of the isoelectric points of native proteins with spreading of pKa values, *Electrophoresis* 16 (1995) 1377–1380.
- [47] N. Colaert, H. Barsnes, M. Vaudel, K. Helsens, E. Timmerman, A. Sickmann, et al., Thermo-msf-parser: an open source Java library to parse and visualize Thermo Proteome Discoverer msf files, *J. Proteome Res.* 10 (2011) 3840–3843.
- [48] J.E. Elias, S.P. Gygi, Target-decoy search strategy for mass spectrometry-based proteomics, *Methods Mol. Biol.* 604 (2010) 55–71.
- [49] K.A. Reidegeld, M. Eisenacher, M. Kohl, D. Chamrad, G. Korting, M. Bluggel, et al., An easy-to-use Decoy Database Builder software tool, implementing different decoy strategies for false discovery rate calculation in automated MS/MS protein identifications, *Proteomics* 8 (2008) 1129–1137.
- [50] P. Aiyetan, B. Zhang, L. Chen, Z. Zhang, H. Zhang, M2Lite: an Open-source, Lightweight, Pluggable and Fast Proteome Discoverer MSF to mzIdentML Tool, *J. Bioinforma.* 1 (2014) 40–49.
- [51] K.D. Pruitt, G.R. Brown, S.M. Hiatt, F. Thibaud-Nissen, A. Astashyn, O. Ermolaeva, et al., RefSeq: an update on mammalian reference sequences, *Nucleic Acids Res.* 42 (2014) D756–D763.
- [52] J. Griss, A.R. Jones, T. Sachsenberg, M. Walzer, L. Gatto, J. Hartler, et al., The mzTab data exchange format: communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience, *Mol. Cell. Proteomics* 13 (2014) 2765–2775.
- [53] D.H. Lundgren, S.I. Hwang, L. Wu, D.K. Han, Role of spectral counting in quantitative proteomics, *Expert Rev. Proteomics* 7 (2010) 39–53.
- [54] W. Zhu, J.W. Smith, C.M. Huang, Mass spectrometry-based label-free quantitative proteomics, *J. Biomed. Biotechnol.* 2010 (2010) 840518.
- [55] R.A. Bradshaw, A.L. Burlingame, S. Carr, R. Aebersold, Reporting protein identification data: the next generation of guidelines, *Mol. Cell. Proteomics* 5 (2006) 787–788.
- [56] H. Rodriguez, M. Snyder, M. Uhlen, P. Andrews, R. Beavis, C. Borchers, et al., Recommendations from the 2008 International Summit on Proteomics Data Release and Sharing Policy: the Amsterdam principles, *J. Proteome Res.* 8 (2009) 3689–3692.
- [57] E. Sabido, N. Selevesek, R. Aebersold, Mass spectrometry-based proteomics for systems biology, *Curr. Opin. Biotechnol.* 23 (2012) 591–597.
- [58] Y. Yang, S.J. Adelstein, A.I. Kassir, Target discovery from data mining approaches, *Drug Discov. Today* 17 (2012) S16–S23 (Suppl.).