Bioinformatics Advance Access published February 16, 2011

Proteomics to go: Proteomatic enables the user-friendly creation of versatile MS/MS data evaluation workflows

Michael Specht*, Sebastian Kuhlgert, Christian Fufezan and Michael Hippler
Institute of Plant Biology and Biotechnology, University of Münster, Germany

Associate Editor: Prof. John Quackenbush

ABSTRACT

Summary:

We present Proteomatic, an operating system-independent and user-friendly platform that enables the construction and execution of MS/MS data evaluation pipelines using free and commercial software. Required external programs such as for peptide identification are downloaded automatically in the case of free software. Due to a strict separation of functionality and presentation, and support for multiple scripting languages, new processing steps can be added easily.

Availability and Implementation:

Proteomatic is implemented in C++/Qt, scripts are implemented in Ruby, Python and PHP. All source code is released under the LGPL. Source code and installers for Windows, Mac OS X, and Linux are freely available at http://www.proteomatic.org.

Contact: michael.specht@uni-muenster.de

1 INTRODUCTION

Mass spectrometry has evolved as a powerful tool for the high-throughput analysis of complex protein mixtures, producing immense amounts of data (Aebersold and Mann, 2003). Dedicated software is essential for the identification of peptides and proteins from tandem mass spectra (MS/MS). In addition to commercial software, the increasing availability of free tools for different purposes allows for manifold alterations in the choice of individual programs and their arrangement in an MS/MS data evaluation pipeline. Most programs are controlled via the command-line interface (CLI), which is necessary in order for the program to be included into an automated pipeline. On the other hand, this mode of interaction makes the program less accessible to users. Some programs are delivered with a dedicated graphical user interface (GUI) which facilitates changing parameters and running the program. However, in order to create an automated processing pipeline in which multiple programs are chained together, CLI tools must be used and programming knowledge is required.

Here, we report on Proteomatic, a versatile and user-friendly platform for the construction of MS/MS data processing pipelines. Although alternatives exist (Keller *et al.*, 2005; Kohlbacher *et al.*, 2007), we chose to design a system which implements a strict separation of functionality and the GUI, where new processing steps can be provided using various programming languages. Proteomatic enables the incorporation of programs such as BLAST (Altschul *et al.*, 1990), PEAKS (Ma *et al.*, 2003), or OMSSA (Geer *et al.*, 2004) and provides a GUI to adjust the parameters of each program.

In contrast to other workflow management systems like Taverna (Oinn *et al.*, 2004) and Galaxy (Goecks *et al.*, 2010) Proteomatic operates in a decentralized fashion. Web servers are not required, and all programs are executed locally on the user's machine. Whenever freely available external programs are required Proteomatic will download and unpack the appropriate packages automatically, thus facilitating the application of such programs.

2 METHODS

On the conceptual level, albeit transparent to the user, Proteomatic is split into three distinct parts: (i) program descriptions (CLI tools atlas), (ii) processing scripts (Proteomatic scripts), and (iii) the Proteomatic GUI.

The separation of functionality from the GUI is achieved through the use of the external program descriptions which provide all necessary information to automatically construct a GUI for a certain external program and to allow its incorporation into a pipeline.

2.1 CLI tools atlas

Information about various free and commercial mass spectrometry-related programs is stored as YAML-formatted descriptions. These descriptions contain information about parameters, input/output files and download locations in the case of free software. Possible parameter types include integer and real numbers, strings, text fields, drop-down boxes, and boolean flags.

2.2 Proteomatic scripts

The Proteomatic scripts implement all functionality available in Proteomatic. Features such as automatic software downloading and output file tracking are provided by a framework implemented in Ruby. Scripts implemented in other languages implicitly access the same functionality through an 'any language hub' which acts as an abstraction layer between the Ruby framework and scripting languages other than Ruby. As available for external programs, a YAML-formatted description also exists for every Proteomatic script. If a script acts as a wrapper around an external program, its description may reflect the external program's parameters by including its description from the CLI tools atlas.

The implementation of a Proteomatic script is straightforward, regardless of the actual scripting language used (Ruby, Python and PHP are currently supported). In general, a Proteomatic script defines a subclass of `ProteomaticScript` and implements the virtual `run()` method. The underlying framework collects user-provided input files and parameters, as well as the requested output files and makes this information available in the three instance variables `input`, `param`, and `output`. A complete developer's documentation can be found on the website.

*To whom correspondence should be addressed.

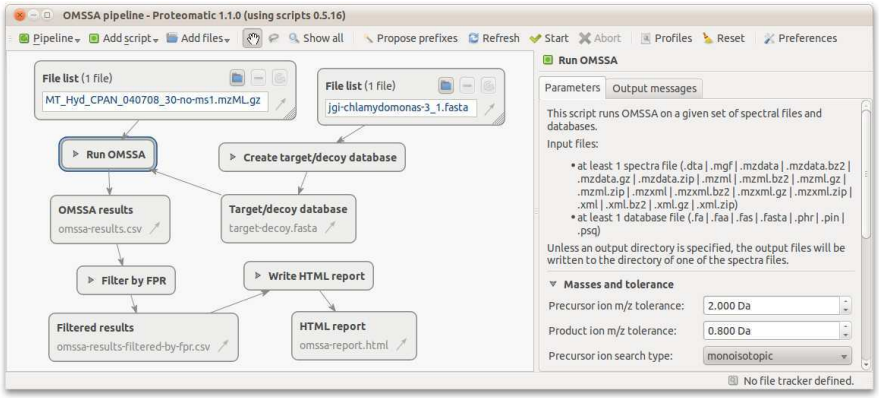

Figure 1. Proteomatic screenshot demonstrating a simple MS/MS data processing pipeline. The pipeline shown implements a protein identification pipeline, using a target/decoy approach in conjunction with OMSSA. The processing pipeline can be seen on the left hand side of the window, composed of existing input files (blue font), yet to be created output files (gray font) and scripts in between. The right hand side of the window contains the user-adjustable parameters of the *Run OMSSA* script. Once a pipeline has been constructed, it can be saved and re-run on a different set of input files at a later time.

2.3 Proteomatic GUI

The Proteomatic GUI is implemented as a C++/Qt application, enabling seamless integration with Windows, Mac OS X, or Linux desktops. The application itself does not provide any MS/MS data evaluation functionality but acts as a user interface layer on top of the Proteomatic scripts.

Users may choose various processing steps from a menu. Every script is depicted as a box on a canvas and its parameters can be modified in the right-hand pane (see Fig. 1). Files can be added to the canvas and specified as input files to a script by connecting both boxes via an arrow. By connecting the output files of one script to another script, increasingly complex pipelines can be constructed. Once a pipeline has been created, it can be executed by clicking the *Start* button.

Although the Proteomatic GUI does not provide any means to inspect result files, output files can be opened via a double-click, thereby delegating the handling of the output file to underlying operating system.

3 RESULTS

Proteomatic contains more than 70 scripts for various purposes, e.g. peptide and protein identification at a user-defined FPR (false positive rate) using a target/decoy approach (Elias and Gygi, 2007; Käll *et al.*, 2008). In addition, protein groups can be determined, thus reducing the amount of peptides matching to multiple sequences (Nesvizhskii and Aebersold, 2005). Proteomatic can also be used for peptide and protein quantitation using a novel quantitation tool, qTrace[1], as demonstrated in (Terashima *et al.*, 2010). Detailed documentation of Proteomatic and all available scripts can be found on the Proteomatic website.

4 CONCLUSIONS

Proteomatic provides a high-throughput data evaluation platform for protein identification, using a variety of freely available programs which are downloaded automatically when required, thus providing a straightforward system to evaluate large MS/MS data sets.

Through the use of scripting languages, existing functionality can easily be adjusted and new processing steps can be added using Ruby, Python, PHP, or potentially any other operating system-independent scripting language (see Supplemental Material for a detailed example). The storage of program descriptions and source code for Proteomatic scripts and the GUI in separate, publicly accessible Git repositories facilitates the enhancement of the system. We hope that the variety of supported scripting languages and the straightforward deployment to the Proteomatic GUI encourages community contributions and fuels the development of novel MS/MS data evaluation tools.

[1] manuscript in preparation

ACKNOWLEDGEMENT

Funding: This work was supported by Deutsche Forschungsgemeinschaft [FU780/2-1 to C.F.]; and Bundesministerium für Bildung und Forschung [0315265C to M.H.].

REFERENCES

Aebersold, R. and Mann, M. (2003). Mass spectrometry-based proteomics. *Nature*, **422**(6928), 198–207.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, **215**(3), 403–410.

Elias, J. E. and Gygi, S. P. (2007). Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat Meth*, **4**(3), 207–214.

Geer, L. Y., Markey, S. P., Kowalak, J. A., Wagner, L., Xu, M., Maynard, D. M., Yang, X., Shi, W., and Bryant, S. H. (2004). Open mass spectrometry search algorithm. *J Proteome Res*, **3**(5), 958–964.

Goecks, J., Nekrutenko, A., Taylor, J., and Team, T. G. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol*, **11**(8), R86.

Käll, L., Storey, J. D., MacCoss, M. J., and Noble, W. S. (2008). Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J Proteome Res*, **7**(1), 29–34.

Keller, A., Eng, J., Zhang, N., Li, X., and Aebersold, R. (2005). A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Mol Syst Biol*, **1**:2005.0017.

Kohlbacher, O., Reinert, K., Gröpl, C., Lange, E., Pfeifer, N., Schulz-Trieglaff, O., and Sturm, M. (2007). TOPP – the OpenMS proteomics pipeline. *Bioinformatics*, **23**(2), e191–197.

Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., and Lajoie, G. (2003). PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid Commun Mass Sp*, **17**(20), 2337–2342.

Nesvizhskii, A. I. and Aebersold, R. (2005). Interpretation of shotgun proteomic data: the protein inference problem. *Mol Cell Proteomics*, **4**(10), 1419–1440.

Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., and Li, P. (2004). Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, **20**(17), 3045 –3054.

Terashima, M., Specht, M., Naumann, B., and Hippler, M. (2010). Characterizing the anaerobic response of Chlamydomonas reinhardtii by quantitative proteomics. *Mol Cell Proteomics*, **9**(7), 1514–1532.