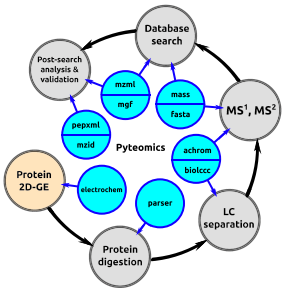
J. Am. Soc. Mass Spectrom. (2013)
DOI: 10.1007/s13361-012-0516-6

APPLICATION NOTE

Pyteomics—a Python Framework for Exploratory Data Analysis and Rapid Software Prototyping in Proteomics

Anton A. Goloborodko,[1,2,3] Lev I. Levitsky,[2,3] Mark V. Ivanov,[2,3] Mikhail V. Gorshkov[2,3]

[1]Department of Physics, Massachusetts Institute of Technology, Boston, MA 02139, USA
[2]Institute for Energy Problems of Chemical Physics, Russian Academy of Sciences, Moscow 119334, Russia
[3]Moscow Institute of Physics and Technology (State University), Dolgoprudny, Moscow region 141700, Russia



Abstract. Pyteomics is a cross-platform, open-source Python library providing a rich set of tools for MS-based proteomics. It provides modules for reading LC-MS/MS data, search engine output, protein sequence databases, theoretical prediction of retention times, electrochemical properties of polypeptides, mass and *m/z* calculations, and sequence parsing. Pyteomics is available under Apache license; release versions are available at the Python Package Index http://pypi.python.org/pyteomics, the source code repository at http://hg.theorchromo.ru/pyteomics, documentation at http://packages.python.org/pyteomics. Pyteomics.biolccc documentation is available at http://packages.python.org/pyteomics.biolccc/. Questions on installation and usage can be addressed to pyteomics mailing list: pyteomics@googlegroups.com

Key words: Data processing, Bioinformatics, Proteomics

Received: 19 June 2012/Revised: 3 October 2012/Accepted: 8 October 2012

Introduction

Modern mass spectrometry-based proteomics deals with large datasets and complex analytical workflows and, therefore, depends vitally on the software support. Traditionally, proteomics software was written in statically-typed languages such as C/C++ and Java [1–3] (see http://www.ms-utils.org for reference). While showing high performance, they are relatively slow in terms of development speed. In scientific programming and data analysis, this disadvantage is especially noticeable: very often scientific software is developed in the "exploratory" mode where the set of specifications is worked out along with the code itself.

The scripting languages represent another alternative. Their unique flexibility and ease of development make them highly attractive for development of scientific applications [4, 5]. However, there are very few proteomic projects written in dynamic general-purpose languages [6]. This situation is changing now. Particularly, Python is gaining popularity in the proteomic community in recent years [7–9]. This trend can be explained by the unique combination of its properties: high speed of development, interactivity, enormous choice of high-quality libraries, including packages for numeric calculations, statistics and plotting; relatively painless parallelization and low-level optimization. These qualities of Python are already appreciated by scientists from many other scientific fields [10].

In this communication, we present a set of easy-to-use annotated Python tools for both researchers and software developers in the field of proteomics, and, specifically, LC-MS/MS data mining.

Methods

Pyteomics is designed as a toolbox that assists bioinformaticians in developing their own proteomic projects in Python. Scripts for exploratory reproducible data analysis are forming one class of such projects: in fact, most modules in Pyteomics initially appeared as byproducts of such research projects. Another type of tasks targeted by Pyteomics is rapid software prototyping. While the performance of Python may be an order or two less than that of low-level languages, this difference is compensated by increased speed of development at the stage of prototyping. Once the final architecture of a program is designed, the performance gap can be bridged by rewriting the critical sections of code in Cython [11] or C++.

We did not aim to create a set of solutions for a particular problem in proteomics (e.g., peptide-spectrum matching or data management). Instead, we wanted to design a set of reusable generic components allowing accomplishment of a

Authors Goloborodko and Levitsky contributed equally to this work.

Correspondence to: Mikhail V. Gorshkov; *e-mail:* gorshkov@chph.ras.ru

Published online: 05 January 2013

large variety of specialized tasks. The modular, functional design of the library is intended to facilitate both borrowing of the code from the library and integrating the code base of other projects.

In our opinion, a software project should not be characterized only by the number of functions it contains. API documentation and tutorials, unit testing, packaging and distribution, code management—all these components comprise the "good programming practices" often omitted in scientific programming [12]. In Pyteomics, we have paid special attention to these issues, trying to make its usage as easy as possible. In this regard, the choice of Python is straightforward because it provides well-developed solutions addressing these problems.

Module Contents

mass: an interface for calculation of masses of peptides and proteins, modified meptides, meptide ions, and isotopic distributions. The module offers tools for calculation of masses (and *m/z*) of specific ion types and for specific isotopic states,

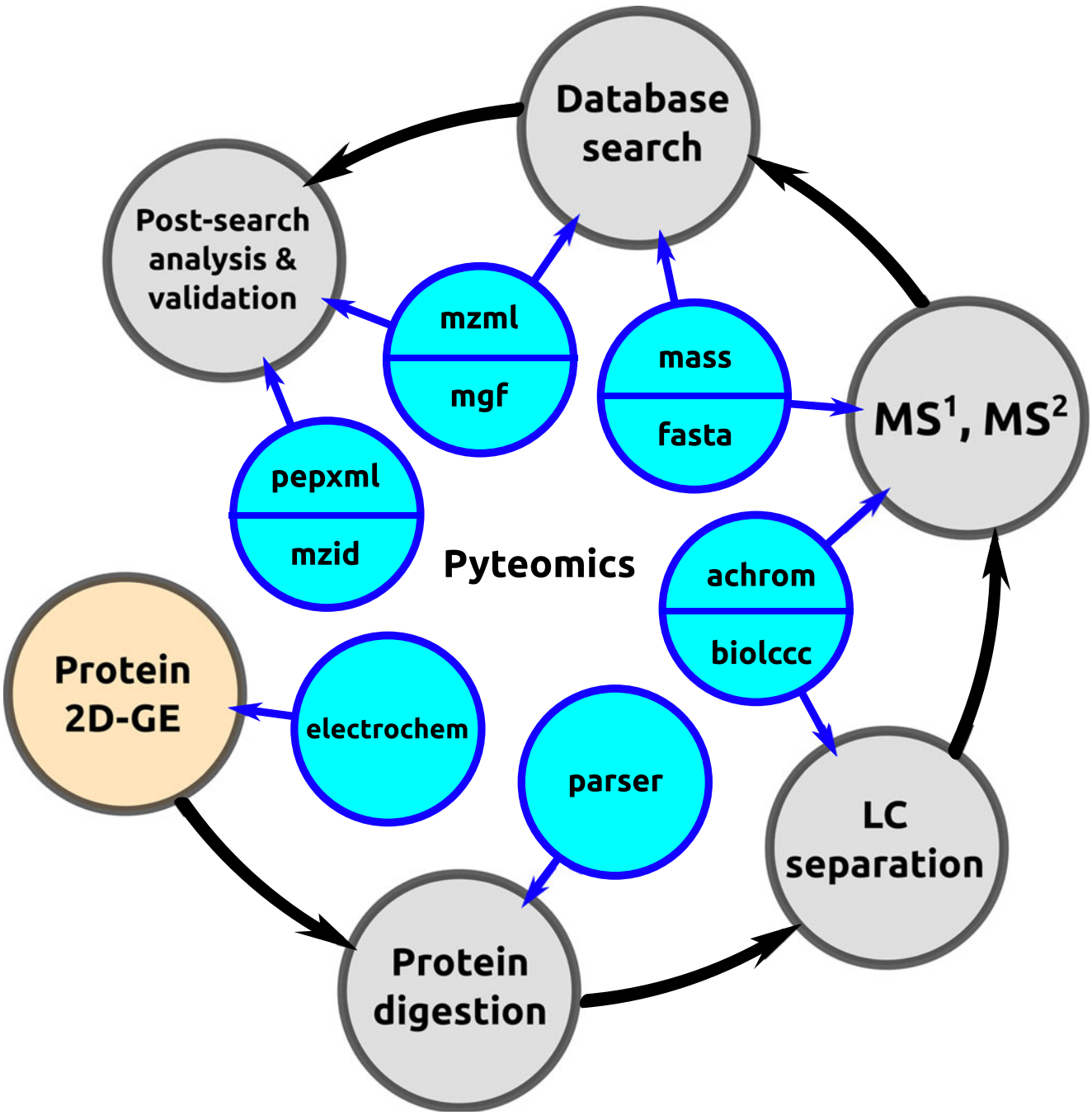


Figure 1. Functionality of pyteomics modules (inner circles) covers all stages of a typical proteomics experiment (outer circles) for data processing in bottom-up, top-down, and middle-down approaches

estimation of isotopic state abundances, and determining the most probable isotopic state of a molecule.

achrom: peptide retention time prediction using the additive model of peptide separation by liquid chromatography [13]. The model has additional corrections for the length of a peptide and the terminal amino acids. A simple interface allows calibrating the model for a specific chromatographic setup.

biolccc: a software implementation of BioLCCC, a physical model of polypeptide chromatography [14]. BioLCCC not only allows prediction of retention times of peptides, but may also be used to estimate the effect of various parameters of experimental setup on the selectivity of peptide separation.

electrochem: prediction of polypeptide charge and isoelectric point using Henderson-Hasselbalch equation.

mzml, fasta, mgf, pepxml, mzid: parsers for the community standards of proteomic data representation. These modules allow full access to the information stored in the corresponding formats. The retrieved data are converted to the standard Python data types. The structure of the original files is preserved where possible. Additionally, fasta and mgf modules allow generation of FASTA databases (including decoy database generation) and MGF files. The mzml module allows extracting information from files in mzML format [15]. Its design enables parsing large files in a memory-efficient manner, making it easy to process files of several Gb in size on virtually any machine. The parsing time is approximately 10–15 s/Gb on a PC with a 2.4 GHz Intel CPU. The mzid module provides support for the recently introduced mzIdentML format [16].

parser: a technical module that simplifies handling of peptide sequences. It allows in silico cleavage of polypeptide sequences, generation of modified sequences, etc.

Results and Discussion

Pyteomics implements a wide range of methods that cover all stages of typical bottom-up and top-down proteomic workflows as shown schematically in Figure 1: in silico protein digestion, retention time and isoelectric point prediction, mass (and *m/z*) calculation, combined with full access to data stored in protein sequence databases, LC-MS/MS data files, and search engine output. This enables combining experimental data and calculations in all possible ways.

Pyteomics was already used in a number of research projects [17–20]. In the recent study on peptide sequence scrambling [17], pyteomics.mass was used to perform thorough annotation of all fragment peaks in the high-quality databases of CAD and HCD fragmentation spectra. Pyteomics.achrom and pyteomics.parser were applied to evaluate the degree of orthogonality between HILIC and reversed-phased peptide separation techniques [19]. Apart from the published studies, pyteomics is commonly employed in our laboratory and by our collaborators for routine tasks such as proteomic search engine optimization or post-search analysis of peptide-spectrum matches as well as for illustrative/educational purposes.

Conclusions

Pyteomics provides a set of tools for rapid proteomics software development. It implements low-level mass spectrometry abstractions in high-level Python programming language. The advantages of the library are the simplicity of design, diversity of implemented modules, and careful implementation in accordance with good programming practices. The library may find its application niche wherein the code flexibility is the key feature (e.g., in exploratory data analysis and/or software prototyping). The source code of the library, documentation, and installation packages are available for the public under the Apache license, ver. 2.0 (http://www.opensource.org/licenses/Apache-2.0). The license allows copying, distributing, and modifying the work, or using any parts of it (including commercial use) with the condition of attribution to the original authors. Pyteomics features Python 3 support since ver. 1.2.0.

Feature addition is made via source code repository at http://hg.theorchromo.ru/pyteomics, thus allowing accepting patches from the community and complying with the continuous integration paradigm. The directions of further development include optimization of the critical segments of code with Cython, as well as adding support for other file formats and new features resulting from community feedback. We also plan to interact actively with other software projects to provide a consistent and feature-rich Python environment for bioinformaticians.

Acknowledgments

The authors are grateful to their colleagues Dr. Irina Tarasova, Dr. Marina Pridatchenko, Anna Lobas, and Tatiana Perlova from the Institute for Energy Problems of Chemical Physics, as well as Dr. Achim Treumann from the University of Newcastle for useful discussions and suggestions on pyteomics functionality.

This work was supported in part by grants from the European Commission (FP7 project Prot-HiSPRA, #282506) and the Russian Basic Science Foundation (project #11-04-00515).

References

1. Kessner, D., Chambers, M., Burke, R., Agus, D., Mallick, P.: ProteoWizard: open source software for rapid proteomics tools development. *Bioinformatics* **24**(21), 2534–2536 (2008)
2. Kohlbacher, O., Reinert, K., Gröpl, C., Lange, E., Pfeifer, N., Schulz-Trieglaff, O., Sturm, M.: TOPP—the OpenMS proteomics pipeline. *Bioinformatics* **23**(2), 191–197 (2007)

3. May, D., Law, W., Fitzgibbon, M., Fang, Q., McIntosh, M.: Software platform for rapidly creating computational tools for mass spectrometry-based proteomics. *J.P.R.* **8**(6), 3212–3217 (2009)
4. Ousterhout, J.K.: Scripting: higher level programming for the 21st century. *Computer* **31**, 23–30 (1998)
5. Loui, R.P.: In praise of scripting: real programming pragmatism. *Computer* **41**, 22–26 (2008)
6. Prince, J.T., Marcotte, E.M.: mspire: mass spectrometry proteomics in Ruby. *Bioinformatics* (Oxford, England) **24**(23), 2796–2797 (2008)
7. Strohalm, M., Kavan, D., Novák, P., Volný, M., Havlícek, V.: mMass 3: a cross-platform software environment for precise analysis of mass spectrometric data. *Anal. Chem.* **82**, 4648–4651 (2010)
8. Specht, M., Kuhlgert, S., Fufezan, C., Hippler, M.: Proteomics to go: proteomatic enables the user-friendly creation of versatile MS/MS data evaluation workflows. *Bioinformatics* (Oxford, England) **27**, 1183–1184 (2011)
9. Bald, T., Barth, J., Niehues, A., Specht, M., Hippler, M.: pymzML–Python module for high throughput bioinformatics on mass spectrometry data. *Bioinformatics* (Oxford, England) **28**(7), 1052–1053 (2012)
10. Perez, F., Granger, B.E., Hunter, J.D.: Python: an ecosystem for scientific computing. *Comput. Sci. Eng.* **13**, 13–21 (2011)
11. Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D.S., Smith, K.: Cython: the best of both worlds. *Comput. Sci. Eng.* **13**, 31–39 (2011)
12. Rother, K., Potrzebowski, W., Puton, T., Rother, M., Wywial, E., Bujnicki, J.M.: A toolbox for developing bioinformatics software. *Brief. Bioinform.* **13**, 244–257 (2012)
13. Meek, J.L.: Prediction of peptide retention times in high-pressure liquid chromatography on the basis of amino acid composition. *Proc. Natl. Acad. Sci. U. S. A.* **77**, 1632–1636 (1980)
14. Gorshkov, A.V., Tarasova, I.A., Evreinov, V.V., Savitski, M.M., Nielsen, M.L., Zubarev, R.A., Gorshkov, M.V.: Liquid chromatography at critical conditions: comprehensive approach to sequence-dependent retention time prediction. *Anal. Chem.* **78**, 7770–7777 (2006)
15. Martens, L., Chambers, M., Sturm, M., Kessner, D., Levander, F., Shofstahl, J., Tang, W.H., Rompp, A., Neumann, S., Pizarro, A.D., Montecchi-Palazzi, L., Tasman, N.,Coleman, M., Reisinger, F., Souda, P., Hermjakob, H., Binz, P-A., Deutsch E.W.: mzML— a community standard for mass spectrometry data. *Mol. Cell. Proteom.* **10**(1) (2011)
16. Eisenacher, M.: mzIdentML: an open community-built standard format for the results of proteomics spectrum identification algorithms. *Methods Mol. Biol.* (Clifton, NJ) **696**, 161–177 (2011)
17. Goloborodko, A.A., Gorshkov, M.V., Good, D.M., Zubarev, R.A.: Sequence scrambling in shotgun proteomics is negligible. *J. Am. Soc. Mass Spectrom.* **22**, 1121–1124 (2011)
18. Levitsky, L., Goloborodko, A.A., Gorshkov, M.V.: The influence of search parameters and mass spectrometry data quality on the search engine performance in shotgun proteomics: a systematic study. Proceedings of the 59th ASMS Conference on Mass Spectrometry and Allied Topics. June 2011, Denver, CO, (2011)
19. Moskovets, E., Goloborodko, A.A., Gorshkov, A.V., Gorshkov, M.V.: Limitation of predictive 2D liquid chromatography in reducing the database search space in shotgun proteomics: in silico studies. *J. Sep. Sci.* **35**, 1771–1778 (2012)
20. Goloborodko, A.A., Mayerhofer, C., Zubarev, A.R., Tarasova, I.A., Gorshkov, A.V., Zubarev, R.A., Gorshkov, M.V.: Empirical approach to false discovery rate estimation in shotgun proteomics. *Rapid Commun. Mass Spectrom.* **24**, 454–462 (2010)