phpMs: A PHP-based mass spectrometry utilities library

Andrew Collins[1,*], Andrew R Jones[1]

[1]Department of Functional and Comparative Genomics, Institute of Integrated Biology, University of Liverpool, Liverpool, United Kingdom

[*]To whom correspondence should be addressed. Email: acollins@liv.ac.uk

Present Address: Department of Functional and Comparative Genomics, Institute of Integrative Biology, University of Liverpool, Biosciences building, Crown Street, Liverpool, L69 7ZB, United Kingdom

Keywords

phpMs, PHP, proteomics, library, bioinformatics, software, mass spectrometry

Abstract

The recent establishment of cloud computing, high throughput networking, and more versatile web standards and browsers has led to a renewed interest in web-based applications. While traditionally big data has been the domain of optimised desktop and server applications, it is now possible to store vast amounts of data and perform the necessary calculations offsite in cloud storage and computing providers, with the results visualised in a high-quality cross-platform interface via a web browser. There are number of emerging platforms for cloud based mass spectrometry data analysis, however, there is limited pre-existing code accessible to web developers, especially for those that are constrained to a shared hosting environment where Java and C applications are often forbidden from use by the hosting provider.

To remedy this, we provide an open source mass spectrometry library for one of the most commonly used web development languages, PHP. Our new library, phpMs, provides objects for storing and manipulating spectra and identification data as well as utilities for file reading, file writing, calculations, peptide fragmentation and protein digestion, as well as a software interface for controlling search engines. We provide a working demonstration of some of the capabilities at http://pgb.liv.ac.uk/phpMs.

Introduction

Proteomics is a dynamic field, with complex and fast changing requirements for bioinformatics analysis. In tandem with the evolution of bioinformatics approaches, has been the development of data standards, driven by the Proteomics Standards Initiative (PSI). Relevant PSI standards include mzML for raw and processed mass spectrometry (MS) data[1], mzIdentML for peptide and protein identification results[2] and mzTab for a simple summary of quantitative (and identification) data amongst others[3]. Two new PSI formats have also recently been developed for proteogenomics (the use of proteomic data for genome annotation), proBed and proBAM, extending popular genomics formats, for displaying peptides aligned onto chromosomes[4]. Bioinformatics tools for proteomics are written in a range of programming languages, including Java, C++, Python, R, Perl and so forth. In several cases, groups have created application programming interfaces (APIs) to make it easier for developers to incorporate support for data standards in their tools including Java APIs: jmzML[5], jmzIdentML[6], jmzTab[7] and in python: pymzML. Beyond APIs purely for data standards, informatics groups have created libraries containing analysis routines for processing data and constructing pipelines, including the Trans Proteomics Pipeline[8], OpenMS[9], mzidLib[10], ProteoWizard[11] and Pyteomics[12]. Most of these libraries have been used within applications on the desktop or in some cases running as command-line applications on the cloud. Within the cloud environment, tools such as Galaxy[13] are proving popular for creating web-based pipeline executors with basic interfaces. Deployment of pipelines on cloud-based architectures is increasingly being performed via the use of *Containers* (wrappers that include all executables and dependencies), such as Docker (https://www.docker.com/). Various other web-based software suites also offer some proteomics-related utilities, including Protein Prospector[14] and ExPASy (https://www.expasy.org/proteomics).

While there is a wide range of APIs for data standards, and libraries to assist developers or proteome scientists to construct analysis workflows, we believe that there is a current lack of appropriate tools for web developers wishing to build prototypes or rapidly construct browser-enabled applications for proteomics. Cloud-based technologies such as Galaxy and/or Docker are highly effective for executing command-line applications, but do not easily assist the development of web-based data visualisation or prototyping. PHP Hypertext Pre-processor (PHP) is one of the most commonly used languages for developing web applications with many high profile, and consequently high throughput, users such as Facebook, Wikipedia, Tumblr and many more. Its popularity, owed to its extensive native library, ease of use, and supportive community, has ensured it can be expected to be available on most managed shared web hosting services.

While most languages can be utilised for web development, including higher level languages like C/C++, these are generally not available in shared hosting environments, due to the difficulty and cost of managing these types of applications, thus use of these types of languages are often restricted to virtual private server/cloud based environments where more control is available at higher cost. This unfortunately leaves many bioinformatics software unable to use managed shared hosts that limit their users to the commonly deployed LAMP (Linux, Apache, MySQL and PHP) stack.

For writing new web applications, PHP is one of the simplest languages to write, since it has a simple C-like syntax, an extensive native library, and allows in-line blocks of pure HTML to be inserted straightforwardly. For these reasons, PHP is often seen as a highly suited language for rapid prototyping of new ideas. To our knowledge, no other group has developed a PHP-based library for assisting proteomics data analysis. While complex statistical processing is not ideally suited to working in PHP due to a lack of native libraries for advanced data modelling, PHP is perfectly well suited for file manipulations, web-based visualisation and simpler mathematical or text-based tasks. As a result, we have created phpMs, as a new library to assist in the rapid development of web-based tools and data visualisations. Given that our group has already developed a Java-based library for MS data manipulation[10], phpMs is not intended for re-implementing complex algorithms for data processing. Instead, we intend that it will be preferred over the more complex Java libraries for bioinformaticians, including those without formal software engineering training, wishing to develop new web-based tools and utilities.

In phpMs, we have incorporated support for mzIdentML processing and visualisation, and MGF format for MS peak lists. We have added routines for performing *in silico* enzymatic digestion of protein sequences within FASTA files, for simulating peptide fragmentation, and a basic interfaces for calling open source search engines, such as MS-GF+[15]. The library also contains a variety of other common utilities for manipulating proteomics identification data. To demonstrate the potential utility of the library, we have deployed several tools on an open access server, accessible from http://pgb.liv.ac.uk/phpMs. All code within phpMs is open source (https://github.com/PGB-LIV/php-ms) and released under the permissive Apache2 licence.

Materials and Methods

phpMs has been written in the PHP scripting language, and developed to be compatible with instances running PHP 5.4 or later. The library can be used on any platform on which PHP is available, currently Windows, Linux or Mac. Specifically, with regards to Linux, PHP is generally widely available via a package manager (such as *yum* or *apt-get*) on all major Linux distributions where similar languages such as Perl or Python might be found. The library itself does not require a web server environment and can be used purely in command line interface (CLI) applications if required. However, the library can be fully utilisable by a web application if preferred as per the phpMs Demo Suite.

The demo suite that is also provided has been built to run on any PHP (5.4 or later) enabled web server such as the commonly available Apache HTTP Server or Microsoft IIS. While PHP 5.4 has been opted for due to it being one of the most widely distributed versions of PHP, this version does lack the recent PHP 7.0 additions of scalar type declarations. However, despite this the code base has been written defensively to require that developers ensure the correct data types are passed to methods. This provides a benefit to developers in two ways; firstly, it reduces the performance overhead of having data in the wrong type (e.g. integer as string) since PHP will auto-cast if possible on each operation, and secondly it also prevents accidental input that can generate valid output without error.

Results

phpMs provides various classes that can be integrated by a developer into their project. The general core of phpMs provides necessary objects that allow for the storage and manipulation of data, such as identification based objects including peptides and proteins, or spectra based objects such as precursor and fragment ions.

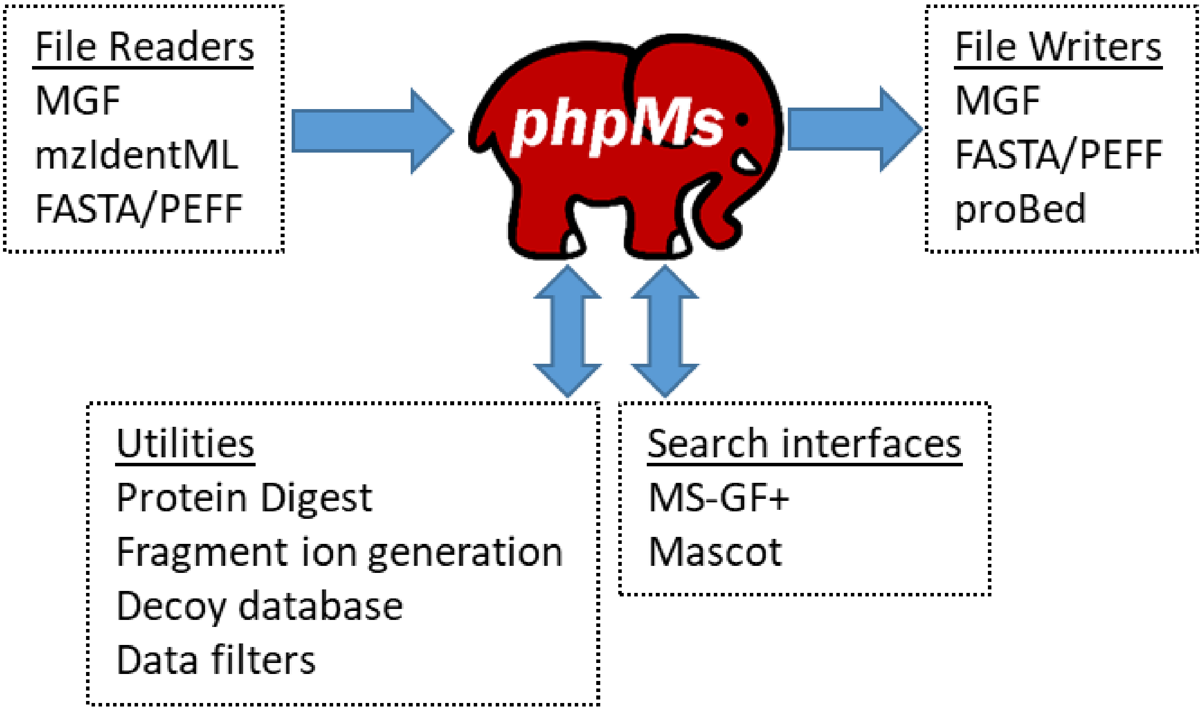


Figure 1: The available tools within phpMs and possible workflow. Note, in addition to the existing file writers, output can be directly written as text or HTML.

The core has been written to encourage good practice by enforcing strict data typing reducing the potential for memory usage concerns and processing inefficiencies due to mistyped data, a critical concern when building highly responsive web applications. Further the core has been kept simple, yet flexible, allowing for an application developer to easily extend the provided classes should they need to in their own classes with their own methods and fields.

The following sections describe classes provided that allow for the manipulation or analysis of data stored within the core, such as the import and export of data, or the calculation of properties on peptides and protein objects.

File Formats

phpMs can import data into its core objects from a number of common file formats used in proteomic identification workflows. With regards to peak lists i.e. precursor and fragmentation data, phpMs is capable of parsing Mascot Generic Format (MGF) data. Sequence data can be parsed directly from FASTA files, by default phpMs will automatically recognise the type of FASTA file and extract UniProt/KB meta-data in relation to each protein within the file. The PSI is currently working on an extension to the FASTA format, called PEFF (PSI Extended Fasta Format, https://github.com/HUPO-PSI/PEFF). PEFF adds machine readable meta-data to the headers for records within a FASTA file for modifications that have been observed on proteins, as well as sequence variants. In theory, this will allow future search engines to search for known modifications and variants automatically. To assist in the finalisation of the PEFF standard, we have added a PEFF reader to phpMs, which is able to process the extra data present within each FASTA record.

Further, phpMs can extract identification data from the mzIdentML format, including support for the recently released mzIdentML 1.2 update[16]. Our mzIdentML reader is able to extract meta-data from the protocol in the file and peptide and protein identification scores or statistics. We reduce the complexity of working with mzIdentML data in phpMs by not requiring the developer to directly interact with API calls to parts of the mzIdentML file, instead our parser handles the reading of the file and the extraction of data into the phpMs core objects, from there the developer is able to interact with any data that was able to be pulled into the core. The advantage of this is it allows phpMs, and consequently the developers who uses the library, to seamlessly support multiple identification formats in the future. A developer would thus only have to focus on working with the data content, regardless of the format, rather than how to extract the data from each format.

Finally, phpMs is also capable of writing MGF, generic FASTA and PEFF formatted files. We have added support for conversion of mzIdentML files containing proteogenomics data (i.e. chromosomal locations) into proBed 1.0 format, available in the core library and in the demo suite. The proBed files can be uploaded onto genome browsers, such as Ensembl[17] or the UCSC browser[18] for data visualisation. The tool can convert a 100MB mzIdentML to proBed in < 30 seconds running on a standard server, demonstrating the speed with which PHP code is able to run.

Digestion

Support is available for trypsin and a further 20 other enzymes as documented from the HUPO-PSI's controlled vocabulary PSI-MS[19]. For each enzyme, it is possible to instruct phpMs to perform n-terminal methionine excision and to account for a user-provided number of missed cleavage occurrences. In addition to the existing enzymes, a generic regular expression class is also provided, further allowing for any currently unsupported enzymes to be trivially added if the digestion rule can be expressed as a Perl-compatible regular expression. The code within phpMs could thus be used by another developer wishing to perform on-the-fly *in silico* digestion of protein sequences into peptides (e.g. paired with the FASTA reader), and render the results to HTML for web presentation.

Fragmentation

phpMs can generate b, y, c, z, a and x fragmentation ions from both an unmodified and modified sequence. Further through the use of a factory class it is possible to automatically generate the correct fragmentation ions for a particular fragmentation mode such as ETD or CID. Neutral losses such as ammonia or water loss are by default not automatically generated, however, these can be generated by specifying a modification.

Filtering

Generally, it is expected that users will want to reduce the amount of data they analyse and visualise such as limiting by the length, charge, retention time, or mass of a peptide or spectra. These types of filters are natively supported and developed to process either individual records or to process large arrays of data. As per other areas of the phpMs library, interfaces and abstract classes are available to allow for downstream developers to extend the filtering capabilities to fit their own requirements.

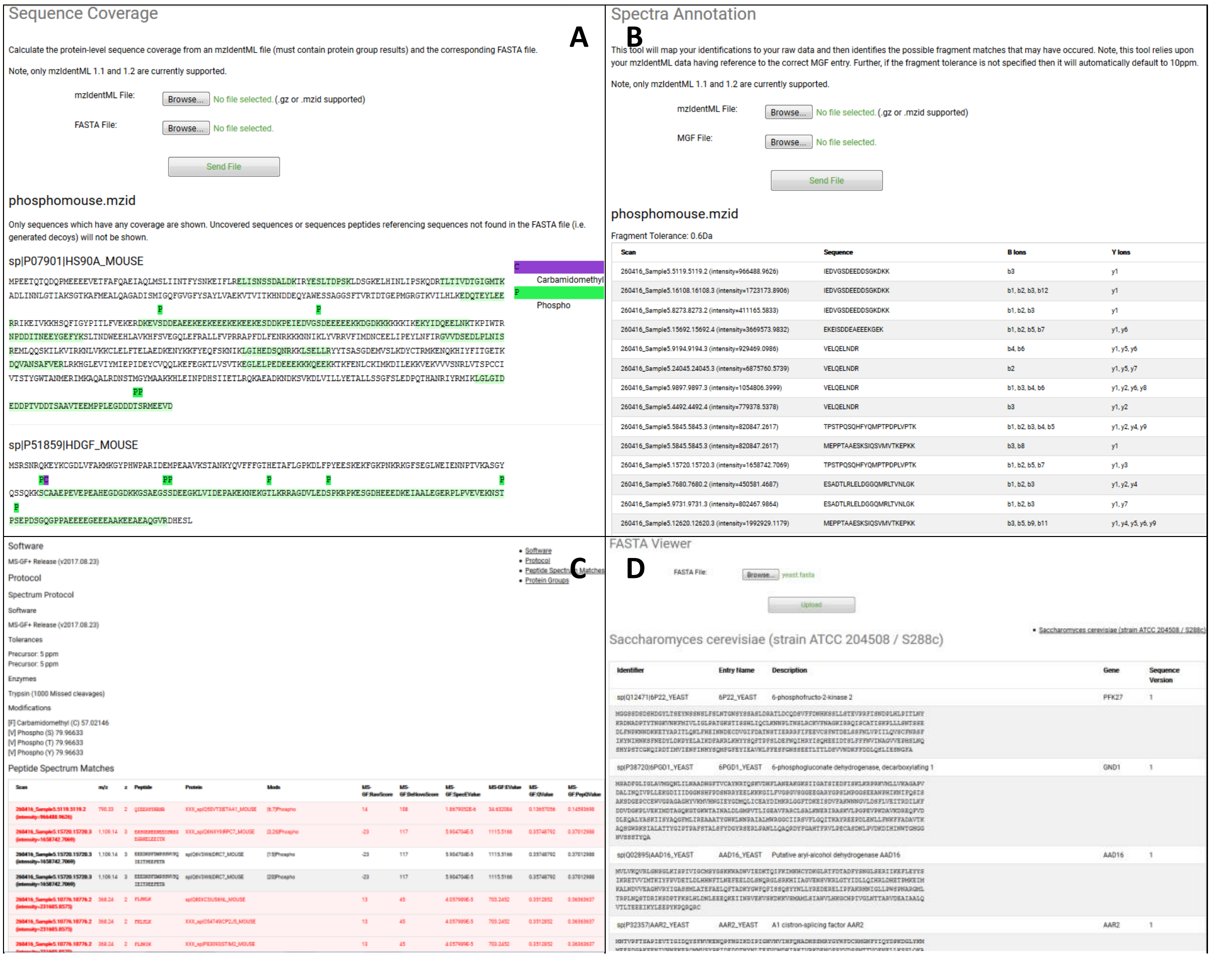
Search Engines

A simple proof of concept for search engines has been included in the 1.0 release. Currently an API has been built that will allow for programmatically sending search requests, and retrieving results from search engines. Currently only Mascot and MS-GF+ are natively supported by the library. The demo suite includes an MS-GF+ tool that allows for the running of simplified searches (to reduce load on the demo server) that will be visualised in the demo suite's mzIdentML viewer once the search is complete. This library and demo can easily be extended to support more tools and more complex data files allowing for a purely web based interface to what are traditionally CLI tools.

Conclusion & Future Perspectives

We anticipate that the new library will support or enhance the development of new web based analytical and visualisation platforms. Since the library provides many functionalities out of the box such as supporting standard formats, filtering, and calculations, it will reduce the workload for a future project allowing the team responsible to focus purely on developing the specifics of their application rather than infrastructure.

While this library is still under active development, it is expected that as it matures, and the needs of downstream developers become more apparent, future revisions will be produced with more tools. phpMs will be continuously developed to support features such as additional common file formats, new calculations and third-party tool integrations. In the immediate future, we plan to support additional file formats to further make phpMs accessible to a wider number of users, for example including enhanced support for new PSI formats proBed / proBAM and PEFF. Given that working with XML formats can be challenging for some developers, we will also prioritise improving the range of supported routines for mzIdentML and mzML.



Sequence Coverage

Calculate the protein-level sequence coverage from an mzIdentML file (must contain protein group results) and the corresponding FASTA file.

Note, only mzIdentML 1.1 and 1.2 are currently supported.

mzIdentML File: Browse... No file selected. (.gz or .mzid supported)

FASTA File: Browse... No file selected.

Send File

phosphomouse.mzid

Only sequences which have any coverage are shown. Uncovered sequences or sequences peptides referencing sequences not found in the FASTA file (i.e. generated decoys) will not be shown.

sp|P07901|HS90A_MOUSE

```
MPEETQTQDQPMEEEEVETFAFQAEIAQLMSLIINTFYSNKEIFLRELISNSSDALDKIRYESLTDPSKLDSGKELHINLIPSKQDRTLTIVDTGIGMTK
ADLINNLGTIAKSGTKAFMEALQAGADISMIGQFGVGFYSAYLVAEKVTVITKHNDDEQYAWESSAGGSFTVRTDTGEPMGRGTKVILHLKEDQTEYLEE
                                     P                                      P
RRIKEIVKKHSQFIGYPITLFVEKERDKEVSDDEAEEKEEKEEEKEKEEKESDDKPEIEDVGSDEEEEEKKDGDKKKKKKIKEKYIDQEELNKTKPIWTR
NPDDITNEEYGEFYKSLTNDWEEHLAVKHFSVEGQLEFRALLFVPRRAPFDLFENRKKKNNIKLYVRRVFIMDNCEELIPEYLNFIRGVVDSEDLPLNIS
REMLQQSKILKVIRKNLVKKCLELFTELAEDKENYKKFYEQFSKNIKLGIHEDSQNRKKLSELLRYYTSASGDEMVSLKDYCTRMKENQKHIYFITGETK
DQVANSAFVERLRKHGLEVIYMIEPIDEYCVQQLKEFEGKTLVSVTKEGLELPEDEEEKKKQEEKKTKFENLCKIMKDILEKKVEKVVVSNRLVTSPCCI
VTSTYGWTANMERIMKAQALRDNSTMGYMAAKKHLEINPDHSIIETLRQKAEADKNDKSVKDLVILLYETALLSSGFSLEDPQTHANRIYRMIKLGLGID
                              PP
EDDPTVDDTSAAVTEEMPPLEGDDDTSRMEEVD
```

C Carbamidomethyl

P Phospho

sp|P51859|HDGF_MOUSE

```
MSRSNRQKEYKCGDLVFAKMKGYPHWPARIDEMPEAAVKSTANKYQVFFFGTHETAFLGPKDLFPYEESKEKFGKPNKRKGFSEGLWEIENNPTVKASGY
      PC                        PP                    P           P                                 P
QSSQKKSCAAEPEVEPEAHEGDGDKKGSAEGSSDEEGKLVIDEPAKEKNEKGTLKRRAGDVLEDSPKRPKESGDHEEEDKEIAALEGERPLPVEVEKNST
 P
PSEPDSGQGPPAEEEEGEEEAAKEEAEAQGVRDHESL
```

Spectra Annotation

This tool will map your identifications to your raw data and then identifies the possible fragment matches that may have occurred. Note, this tool relies upon your mzIdentML data having reference to the correct MGF entry. Further, if the fragment tolerance is not specified then it will automatically default to 10ppm.

Note, only mzIdentML 1.1 and 1.2 are currently supported.

mzIdentML File: Browse... No file selected. (.gz or .mzid supported)

MGF File: Browse... No file selected.

Send File

phosphomouse.mzid

Fragment Tolerance: 0.6Da

Scan	Sequence	B Ions	Y Ions
260416_Sample5.5119.5119.2 (intensity=966488.9626)	IEDVGSDEEDDSGKDKK	b3	y1
260416_Sample5.16108.16108.3 (intensity=1723173.8906)	IEDVGSDEEDDSGKDKK	b1, b2, b3, b12	y1
260416_Sample5.8273.8273.2 (intensity=411165.5833)	IEDVGSDEEDDSGKDKK	b1, b2, b3	y1
260416_Sample5.15692.15692.4 (intensity=3669573.9832)	EKEISDDEAEEEKGEK	b1, b2, b5, b7	y1, y6
260416_Sample5.9194.9194.3 (intensity=929469.0986)	VELQELNDR	b4, b6	y1, y5, y6
260416_Sample5.24045.24045.3 (intensity=6875760.5739)	VELQELNDR	b2	y1, y5, y7
260416_Sample5.9897.9897.3 (intensity=1054806.3999)	VELQELNDR	b1, b3, b4, b6	y1, y2, y6, y8
260416_Sample5.4492.4492.4 (intensity=779378.5378)	VELQELNDR	b3	y1, y2
260416_Sample5.5845.5845.3 (intensity=820847.2617)	TPSTPQSQHFYQMPTPDPLVPTK	b1, b2, b3, b4, b5	y1, y2, y4, y9
260416_Sample5.5845.5845.3 (intensity=820847.2617)	MEPPTAAESKSIQSVMVTKEPKK	b3, b8	y1
260416_Sample5.15720.15720.3 (intensity=1658742.7069)	TPSTPQSQHFYQMPTPDPLVPTK	b1, b2, b5, b7	y1, y3
260416_Sample5.7680.7680.2 (intensity=450581.4687)	ESADTLRLELDGGQMRLTVNLGK	b1, b2, b3	y1, y2, y4
260416_Sample5.9731.9731.3 (intensity=802467.9864)	ESADTLRLELDGGQMRLTVNLGK	b1, b2, b3	y1, y7
260416_Sample5.12620.12620.3 (intensity=1992929.1179)	MEPPTAAESKSIQSVMVTKEPKK	b3, b5, b9, b11	y1, y4, y5, y6, y9

Software

MS-GF+ Release (v2017.08.23)

Protocol

Spectrum Protocol

Software

MS-GF+ Release (v2017.08.23)

Tolerances

Precursor: 5 ppm
Precursor: 5 ppm

Enzymes

Trypsin (1000 Missed cleavages)

Modifications

[F] Carbamidomethyl (C) 57.02146
[V] Phospho (S) 79.96633
[V] Phospho (T) 79.96633
[V] Phospho (Y) 79.96633

- Software
- Protocol
- Peptide Spectrum Matches
- Protein Groups

Peptide Spectrum Matches

Scan	m/z	z	Peptide	Protein	Mods	MS-GF:RawScore	MS-GF:DeNovoScore	MS-GF:SpecEValue	MS-GF:EValue	MS-GF:QValue	MS-GF:PepQValue
260416_Sample5.5119.5119.2 (intensity=966488.9626)	790.33	2	[illegible]	XXX_sp|Q5SVT3|ETAA1_MOUSE	[6,7]Phospho	14	108	1.8679052E-6	34.632084	0.13657056	0.14593698
260416_Sample5.15720.15720.3 (intensity=1658742.7069)	1,109.14	3	[illegible]	XXX_sp|Q6NXY9|RPC7_MOUSE	[3,26]Phospho	-23	117	5.904704E-5	1115.5166	0.35748792	0.37012988
260416_Sample5.15720.15720.3 (intensity=1658742.7069)	1,109.14	3	[illegible]	sp|Q6V3W6|DRC7_MOUSE	[15]Phospho	-23	117	5.904704E-5	1115.5166	0.35748792	0.37012988
260416_Sample5.15720.15720.3 (intensity=1658742.7069)	1,109.14	3	[illegible]	sp|Q6V3W6|DRC7_MOUSE	[20]Phospho	-23	117	5.904704E-5	1115.5166	0.35748792	0.37012988
260416_Sample5.10776.10776.2 (intensity=231605.8575)	368.24	2	[illegible]	sp|Q80XC3|US6NL_MOUSE		13	45	4.057989E-5	703.2452	0.3512852	0.36363637
260416_Sample5.10776.10776.2 (intensity=231605.8575)	368.24	2	[illegible]	XXX_sp|O54749|CP2J5_MOUSE		13	45	4.057989E-5	703.2452	0.3512852	0.36363637
260416_Sample5.10776.10776.2	368.24	2	[illegible]	XXX_sp|P83093|STIM2_MOUSE		13	45	4.057989E-5	703.2452	0.3512852	0.36363637

FASTA Viewer

FASTA File: Browse... yeast.fasta

Upload

- Saccharomyces cerevisiae (strain ATCC 204508 / S288c)

Saccharomyces cerevisiae (strain ATCC 204508 / S288c)

Identifier	Entry Name	Description	Gene	Sequence Version
sp|Q12471|6P22_YEAST	6P22_YEAST	6-phosphofructo-2-kinase 2	PFK27	1
[illegible]				
sp|P38720|6PGD1_YEAST	6PGD1_YEAST	6-phosphogluconate dehydrogenase, decarboxylating 1	GND1	1
[illegible]				
sp|Q02895|AAD16_YEAST	AAD16_YEAST	Putative aryl-alcohol dehydrogenase AAD16	AAD16	1
[illegible]				
sp|P32357|AAR2_YEAST	AAR2_YEAST	A1 cistron-splicing factor AAR2	AAR2	1
[illegible]				

Figure 2: Examples of the visualisations produced by the phpMs Demo suite: A) protein sequence coverage view from an mzIdentML and FASTA formatted input; B) annotation of fragment ions identified from mzIdentML and MGF input; C) results rendered from an MS-GF+ search, and D) FASTA Viewer.

Accessibility & Other Tools

phpMs 1.0 is now available directly from source or as a Composer project distributed by Packagist. phpMs is compatible with any PHP instance running PHP 5.4 or later. A web server is not necessary for CLI applications, however, for any web based project a PHP enabled web server must be used (e.g. Apache HTTP Server or Microsoft IIS). phpMs source can be downloaded directly from GitHub (https://github.com/PGB-LIV/php-ms) or installed from Packagist (https://packagist.org/packages/pgb-liv/php-ms). The GitHub URL contain links to an API reference document (http://pgb.liv.ac.uk/ci/phpMs/doc) to assist developers with the libraries usage. Additionally, a web based live demonstration of phpMs capabilities can be accessed at: http://pgb.liv.ac.uk/phpMs. The source code for the demo suite is also available on GitHub (https://github.com/PGB-LIV/php-ms-example).

phpMs development is to be continued and supported, and to assist with providing support to potential issues, an issue tracker is available at: https://github.com/PGB-LIV/php-ms/issues.

Funding

The development of phpMs has been supported by funding from BBSRC whose support we gratefully acknowledge [BB/L005239/1].

References

1. Martens, L.; Chambers, M.; Sturm, M.; Kessner, D.; Levander, F.; Shofstahl, J.; Tang, W. H.; Römpp, 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. *Molecular & Cellular Proteomics* **2011,** 10, (1), R110.000133.

2. Jones, A. R.; Eisenacher, M.; Mayer, G.; Kohlbacher, O.; Siepen, J.; Hubbard, S.; Selley, J.; Searle, B.; Shofstahl, J.; Seymour, S.; Julian, R.; Binz, P.-A.; Deutsch, E. W.; Hermjakob, H.; Reisinger, F.; Griss, J.; Vizcaino, J. A.; Chambers, M.; Pizarro, A.; Creasy, D., The mzIdentML data standard for mass spectrometry-based proteomics results. *Molecular & Cellular Proteomics* **2012,** 11, (7), M111.014381.

3. Griss, J.; Jones, A. R.; Sachsenberg, T.; Walzer, M.; Gatto, L.; Hartler, J.; Thallinger, G. G.; Salek, R. M.; Steinbeck, C.; Neuhauser, N.; Cox, J.; Neumann, S.; Fan, J.; Reisinger, F.; Xu, Q. W.; Del Toro, N.; Perez-Riverol, Y.; Ghali, F.; Bandeira, N.; Xenarios, I.; Kohlbacher, O.; Vizcaino, J. A.; Hermjakob, H., The mzTab data exchange format: communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience. *Mol Cell Proteomics* **2014,** 13, (10), 2765-75.

4. Menschaert, G.; Wang, X.; Jones, A. R.; Ghali, F.; Fenyo, D.; Olexiouk, V.; Zhang, B.; Deutsch, E. W.; Ternent, T.; Vizcaino, J. A., The proBAM and proBed standard formats: enabling a seamless integration of genomics and proteomics data. *Genome Biology* **2018,** accepted for publication.

5. Côté, R. G.; Reisinger, F.; Martens, L., jmzML, an open-source Java API for mzML, the PSI standard for MS data. *PROTEOMICS* **2010,** 10, (7), 1332-1335.

6. Reisinger, F.; Krishna, R.; Ghali, F.; Ríos, D.; Hermjakob, H.; Antonio Vizcaíno, J.; Jones, A. R., jmzIdentML API: A Java interface to the mzIdentML standard for peptide and protein identification data. *PROTEOMICS* **2012,** 12, (6), 790-794.

7. Xu, Q.-W.; Griss, J.; Wang, R.; Jones, A. R.; Hermjakob, H.; Vizcaíno, J. A., jmzTab: A Java interface to the mzTab data standard. *PROTEOMICS* **2014,** 14, (11), 1328-1332.

8. Deutsch, E. W.; Mendoza, L.; Shteynberg, D.; Farrah, T.; Lam, H.; Tasman, N.; Sun, Z.; Nilsson, E.; Pratt, B.; Prazen, B.; Eng, J. K.; Martin, D. B.; Nesvizhskii, A. I.; Aebersold, R., A guided tour of the Trans-Proteomic Pipeline. *PROTEOMICS* **2010,** 10, (6), 1150-1159.

9. Sturm, M.; Bertsch, A.; Gropl, C.; Hildebrandt, A.; Hussong, R.; Lange, E.; Pfeifer, N.; Schulz-Trieglaff, O.; Zerck, A.; Reinert, K.; Kohlbacher, O., OpenMS - An open-source software framework for mass spectrometry. *BMC Bioinformatics* **2008,** 9, (1), 163.

10. Ghali, F.; Krishna, R.; Lukasse, P.; Martínez-Bartolomé, S.; Reisinger, F.; Hermjakob, H.; Vizcaíno, J. A.; Jones, A. R., Tools (Viewer, Library and Validator) that Facilitate Use of the Peptide and Protein Identification Standard Format, Termed mzIdentML. *Molecular & Cellular Proteomics* **2013,** 12, (11), 3026-3035.

11. Chambers, M. C.; Maclean, B.; Burke, R.; Amodei, D.; Ruderman, D. L.; Neumann, S.; Gatto, L.; Fischer, B.; Pratt, B.; Egertson, J.; Hoff, K.; Kessner, D.; Tasman, N.; Shulman, N.; Frewen, B.; Baker, T. A.; Brusniak, M.-Y.; Paulse, C.; Creasy, D.; Flashner, L.; Kani, K.; Moulding, C.; Seymour, S. L.; Nuwaysir, L. M.; Lefebvre, B.; Kuhlmann, F.; Roark, J.; Rainer, P.; Detlev, S.; Hemenway, T.; Huhmer, A.; Langridge, J.; Connolly, B.; Chadick, T.; Holly, K.; Eckels, J.; Deutsch, E. W.; Moritz, R. L.; Katz, J. E.; Agus, D. B.; MacCoss, M.; Tabb, D. L.; Mallick, P., A cross-platform toolkit for mass spectrometry and proteomics. *Nat Biotech* **2012,** 30, (10), 918-920.

12. Goloborodko, A. A.; Levitsky, L. I.; Ivanov, M. V.; Gorshkov, M. V., Pyteomics--a Python framework for exploratory data analysis and rapid software prototyping in proteomics. *J Am Soc Mass Spectrom* **2013,** 24, (2), 301-4.

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

14. Chalkley, R. J.; Baker, P. R.; Medzihradszky, K. F.; Lynn, A. J.; Burlingame, A. L., In-depth analysis of tandem mass spectrometry data from disparate instrument types. *Mol Cell Proteomics* **2008,** 7, (12), 2386-98.

15. Kim, S.; Pevzner, P. A., MS-GF+ makes progress towards a universal database search tool for proteomics. *Nat Commun* **2014,** 5, 5277.

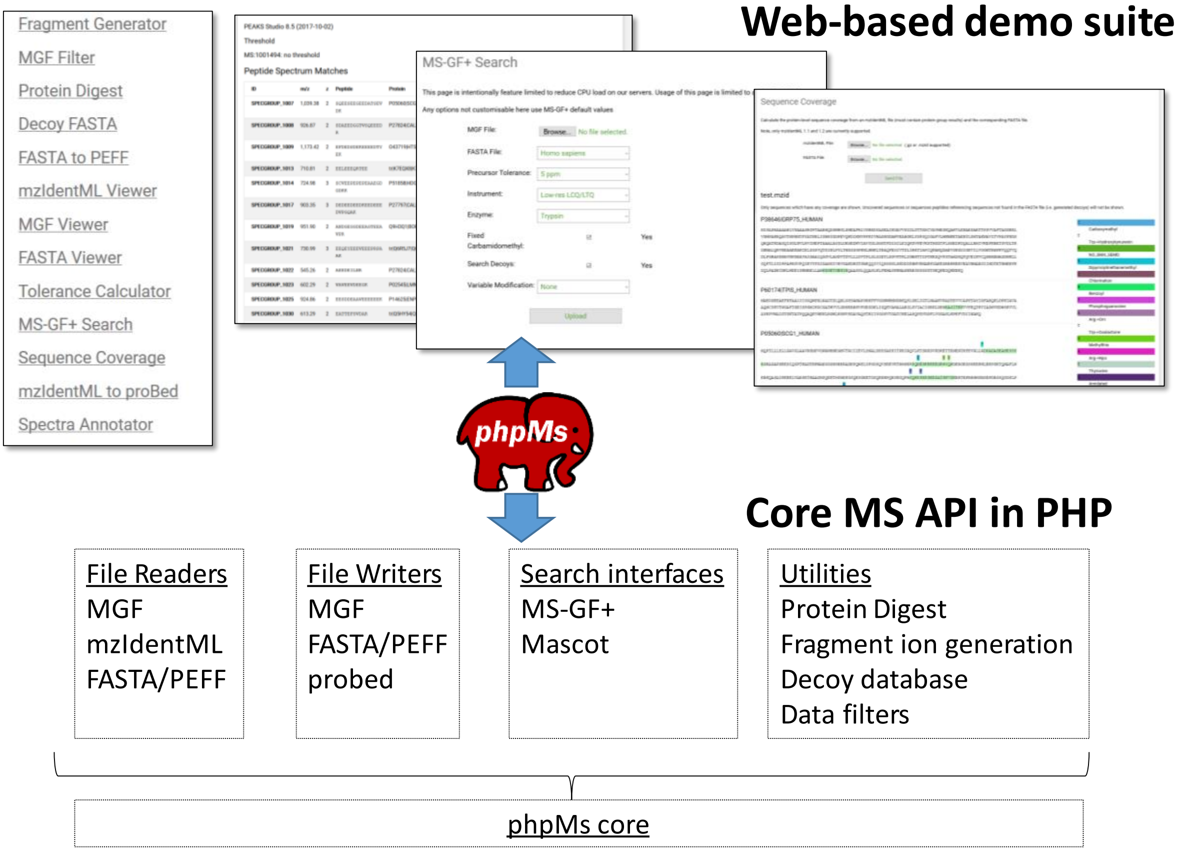
16. Vizcaíno, J. A.; Mayer, G.; Perkins, S.; Barsnes, H.; Vaudel, M.; Perez-Riverol, Y.; Ternent, T.; Uszkoreit, J.; Eisenacher, M.; Fischer, L.; Rappsilber, J.; Netz, E.; Walzer, M.; Kohlbacher, O.; Leitner, A.; Chalkley, R. J.; Ghali, F.; Martínez-Bartolomé, S.; Deutsch, E. W.; Jones, A. R., The mzIdentML Data Standard Version 1.2, Supporting Advances in Proteome Informatics. *Molecular & Cellular Proteomics : MCP* **2017,** 16, (7), 1275-1285.

17. Aken, B. L.; Achuthan, P.; Akanni, W.; Amode, M. R.; Bernsdorff, F.; Bhai, J.; Billis, K.; Carvalho-Silva, D.; Cummins, C.; Clapham, P., Ensembl 2017. *Nucleic acids research* **2016,** 45, (D1), D635-D642.

18. Kent, W. J.; Sugnet, C. W.; Furey, T. S.; Roskin, K. M.; Pringle, T. H.; Zahler, A. M.; Haussler; David, The Human Genome Browser at UCSC. *Genome Research* **2002,** 12, (6), 996-1006.

19. Mayer, G.; Montecchi-Palazzi, L.; Ovelleiro, D.; Jones, A. R.; Binz, P.-A.; Deutsch, E. W.; Chambers, M.; Kallhardt, M.; Levander, F.; Shofstahl, J.; Orchard, S.; Antonio Vizcaíno, J.; Hermjakob, H.; Stephan, C.; Meyer, H. E.; Eisenacher, M., The HUPO proteomics standards initiative- mass spectrometry controlled vocabulary. *Database* **2013,** 2013, bat009.

For TOC only



Web-based demo suite

Core MS API in PHP

File Readers
MGF
mzIdentML
FASTA/PEFF

File Writers
MGF
FASTA/PEFF
probed

Search interfaces
MS-GF+
Mascot

Utilities
Protein Digest
Fragment ion generation
Decoy database
Data filters

phpMs core