Published in final edited form as:
Curr Protoc Bioinformatics. ; 13(1323): 13.23.1–13.23.23. doi:10.1002/0471250953.bi1323s44.

PepArML: A Meta-Search Peptide Identification Platform

Nathan J. Edwards[*]
Department of Biochemistry and Molecular & Cellular Biology Georgetown University Medical Center

Abstract

The PepArML meta-search peptide identification platform provides a unified search interface to seven search engines; a robust cluster, grid, and cloud computing scheduler for large-scale searches; and an unsupervised, model-free, machine-learning-based result combiner, which selects the best peptide identification for each spectrum, estimates false-discovery rates, and outputs pepXML format identifications. The meta-search platform supports Mascot; Tandem with native, k-score, and s-score scoring; OMSSA; MyriMatch; and InsPecT with MS-GF spectral probability scores — reformatting spectral data and constructing search configurations for each search engine on the fly. The combiner selects the best peptide identification for each spectrum based on search engine results and features that model enzymatic digestion, retention time, precursor isotope clusters, mass accuracy, and proteotypic peptide properties, requiring no prior knowledge of feature utility or weighting. The PepArML meta-search peptide identification platform often identifies 2–3 times more spectra than individual search engines at 10% FDR.

Keywords

Proteomics; Mass-Spectrometry; Machine-Learning; Cloud-Computing

INTRODUCTION

The PepArML (Peptide identification Arbiter by Machine Learning) meta-search peptide identification platform provides a unified search interface to seven search engines; a robust cluster, grid, and cloud computing scheduler for large-scale searches; and an unsupervised, model-free machine-learning-based results combiner. The machine-learning based results combiner was first presented at US HUPO 2008, where it was shown to successfully combine search engine results from Mascot, Tandem, and OMSSA. With the publication of the PepArML manuscript (Edwards *et al*., 2009) it became clear that executing target and decoy searches of large bottom-up LC-MS/MS datasets on multiple search engines was highly error prone, even for expert informaticians, leading to inconsistencies, bad configurations, and failed searches. Multiple search engines also increase the issues of scale, since each spectrum must be analyzed multiple times. These issues led to the development of the PepArML meta-search platform, which automatically configures and executes search engines on heterogeneous compute resources using a simple unified search interface,

[*] Phone: (202) 687-7042 Fax: (202) 687-0057 nje5@georgetown.edu.

managing search jobs to ensure successful completion. Currently, PepArML supports seven search engines: Mascot (Perkins *et al*., 1999); Tandem (Craig and Beavis, 2004) with native, k-score, and s-score (MacLean *et al*., 2006) scoring; OMSSA (Geer *et al*., 2004); MyriMatch (Tabb *et al*., 2007; UNIT 13.17); and InsPecT (Tanner *et al*., 2005) with MS-GF (Kim *et al*., 2008) spectral probability scores. Recent additions to PepArML include spectrum, peptide, and sample preparation features combined with search engines' results to increase the model's ability to discriminate correct peptide identifications from incorrect ones. Importantly, the unsupervised model-free training procedure requires no *a priori* knowledge of the performance, utility, or appropriate weighting of the additional features for a particular dataset. The PepArML meta-search platform often identifies 2–3 times more spectra than individual search engines at false-discovery-rate (FDR) 10%.

We present five basic protocols that together represent a complete PepArML analysis: upload tandem mass-spectra (Basic Protocol 1), configure and submit the search (Basic Protocol 2), monitor and manage the search jobs (Basic Protocol 3), optionally run search jobs in the cloud (Basic Protocol 4), and combine the search results (Basic Protocol 5). We also provide an alternative protocol for batch upload of many, large, or vendor format spectra datafiles (Alternate Protocol 1). A support protocol describing how to register and login is also included (Support Protocol 1).

PepArML can be accessed from the Edwards lab at Georgetown University: http://edwardslab.bmcb.georgetown.edu/PepArML (**Error! Reference source not found.**Figure 1 and Table 1).

We use **bold-face** to refer to any user-interface item's title or label, including PepArML tabs, menu items, or text-entry labels. We use *italics* to refer to example spectra files or user-supplied names. In particular, we use the spectrum file *17mix-test2.mzXML* (see Table 1) from the now defunct Sashimi project repository, to provide an example analysis called *Tutorial*, carried out by *Jane Doe* with username *jdoe*.

BASIC PROTOCOL 1

UPLOAD TANDEM MASS-SPECTRA

The PepArML meta-search engine requires that tandem mass-spectra be uploaded to the PepArML server to conduct the peptide identification analysis.

Necessary Resources—A modern web-browser, such as Internet Explorer, Firefox, Chrome, or Safari is required. Users must register and login to PepArML (Support Protocol 1). To follow the example analysis, download the example spectra datafile *17mix-test2.mzXML.gz* (see Table 1).

1. Navigate to the spectra repository by clicking on the **Spectra** tab. The **Spectra** tab will be highlighted and the table header will read Spectra: /users/*jdoe* to indicate the home folder of *Jane Doe*.

2. Create a new folder to hold the spectral data. Enter an appropriate folder name to identify the dataset (e.g. *Tutorial*) in the **New Folder** text entry field near the

bottom of the page and click **Create**. The spectra repository will initially be empty, but will ultimately contain folders for each tandem mass-spectrometry dataset analyzed using PepArML.

3. Navigate into the dataset's folder by clicking on the folder name, *Tutorial*. The table header will now indicate the current folder: Spectra: /users/*jdoe*/*Tutorial*, which will be empty. To navigate back to the home folder, click on *jdoe* in the table header.

4. Upload a tandem mass-spectrometry datafile. Click the **Browse** button of the **Spectra Upload** interface at the bottom of the page. Select the spectra datafile, *17mix-test2.mzXML.gz*, from the local filesystem and click **Upload**. The **Name** text entry field is used when the spectra datafile filename contains problematic symbols, but this is not needed for *17mix-test2*, so the **Name** field is left blank. PepArML accepts most open spectra datafile formats in common usage for peptide identification, including mzXML, mzData, and mgf, and can work directly with uncompressed, gzip (extension .gz), or bzip2 (extension .bz2) compressed files. For many, large, or vendor format RAW spectra datafiles see Alternative Protocol 1.

5. The **Spectra Upload** interface will show the progress of the datafile upload, indicating the percent complete, the upload rate, and the estimated time remaining (**Error! Reference source not found.**Figure 2). The progress information will change to **Done** when the upload is complete.

6. Once the upload is complete, the PepArML server will check the spectra datafile format to ensure it can be understood. If the datafile format is valid and matches the file extension, a count of MS and MS/MS spectra is shown (**Error! Reference source not found.**Figure 3). If the spectral format is invalid or does not match the file extension, **!!ERROR!!** is displayed. The troubleshooting section discusses some of the common reasons for invalid spectra datafiles and provides suggested resolutions.

7. Further spectra datafiles can be uploaded as soon as the previous upload is complete, even if previously uploaded spectra datafiles have not yet appeared in the repository.

8. Return to the spectra repository home folder by clicking on *jdoe* in the table header. The *Tutorial* folder row of the spectra repository will show the number of spectra datafiles, the total size of the spectra datafiles, and the total number of MS and MS/MS spectra in the dataset (**Error! Reference source not found.**Figure 4).

ALTERNATE PROTOCOL 1

BATCH UPLOAD OF MANY, LARGE, OR VENDOR FORMAT SPECTRA DATAFILES

Uploading individual tandem mass-spectra datafiles using Basic Protocol 1 can be burdensome, particularly if spectra datafiles are in a non-open, vendor format. Furthermore, the spectra repository upload interface of Basic Protocol 1 is limited to 500MB per spectral file. The PepArML batch uploader creates open format peak-lists from vendor format spectral data-files and uploads the resulting spectra datafiles to the PepArML server in a robust manner suitable for many and large files.

Necessary Resources—The PepArML batch uploader must be downloaded (see Table 1) from the Edwards lab and installed. If vendor format conversion and peak-picking/peak-detection/centroiding using the ProteoWizard tools (Kessner *et al*. 2008) is required, then the uploader must be run on Windows computers and may require instrument vendor software to be installed. Users must register for PepArML (Supporting Protocol 1). To follow the example analysis, download the example spectra datafile *17mix-test2.mzXML.gz* (see Table 1).

1. Start the PepArML-Batch-Upload software.
2. Click the **Upload files** Browse button and select one or more spectra datafiles, such as *17mix-test2.mzXML.gz*, for upload. Many vendor format datafiles are supported, in addition to open-format spectra datafiles.
3. The destination folder for the spectra datafiles, *Tutorial*, should be specified as **Folder**. The folder need not be created on the PepArML server before upload.
4. The **User** and **Password** fields should correspond to the PepArML username and password.
5. The options in the **Advanced** tab can be left at their default values. The advanced options may, rarely, need to be changed if the network is particularly slow or unreliable, or to adjust the parameters for vendor format conversion or peak picking.
6. Click the **OK** button to begin the conversion (if needed), compression, and upload of the spectra datafiles (**Error! Reference source not found.**Figure 5).
7. [*Figure 5 near here]
8. The PepArML batch upload log window will indicate the progress of the upload(s).
9. Once the upload is complete, the spectra repository folder, *Tutorial*, under the **Spectra** tab on the PepArML server will show the number of spectra datafiles, the total size of the spectra datafiles, and the total number of MS and MS/MS spectra in the dataset.

SUPPORT PROTOCOL 1

REGISTRATION AND LOGIN

Users must register and login before using PepArML.

Necessary Resources—A modern web-browser, such as Internet Explorer, Firefox, Chrome, or Safari is required. A valid email address is required for registration.

1. Access PepArML (see Table 1) by URL or google search for PepArML.

2. Click on **Register** in the top-right corner.
3. Provide **User Name** (*jdoe*), (janedoe@university.edu), **Display Name** (*Jane Doe*), and a **Password**. Click the **Create Account** button.
4. Check the email account for a new user email from the PepArML server.
5. Click on the link in the new user email to verify your PepArML registration.
6. Click the **Login** link at the top-right of the page, use the login box at the top-right of the PepArML homepage, or click on the **login** link provided on the verification page.
7. Enter **User Name** and **Password** as directed (**Error! Reference source not found.**Figure 1).
8. Upon successful login, the PepArML homepage is displayed. The **Display Name** (*Jane Doe*) and **Logout** should show at the top-right of the page.

BASIC PROTOCOL 2

CONFIGURE AND INITIATE THE SEARCH

Having uploaded the spectra, the parameters of the peptide identification search can be configured. The desired search engines, mass-spectrometer, proteolytic enzyme, post-translational modifications, and protein sequence database must be specified. The example spectra, *17mix-test2*, represents a tryptic digest of 17 standard proteins, acquired by LC-MS/MS on a Waters Q-Tof Ultima mass-spectrometer. The spectra are to be submitted to all seven PepArML search engines: Tandem with native scoring (tandem), k-score (kscore), and s-score (sscore) plugin scoring, Mascot (mascot), OMSSA (omssa), MyriMatch (myrimatch), and Inspect+MSGF scoring (inspect). Search parameters include Waters Q-Tof instrument, tryptic digestion, typical fixed and variable modifications, semi-tryptic peptides, and the SwissProt protein sequence database.

Necessary Resources—A modern web-browser, such as Internet Explorer, Firefox, Chrome, or Safari is required. Users must register and login to PepArML (Support Protocol 1). Spectra must already have been uploaded to the PepArML server (Basic Protocol 1 or Alternative Protocol 1).

1. Access the spectra repository by clicking on the **Spectra** tab.
2. Click on the row corresponding to the folder, *Tutorial*, containing the uploaded spectra, but not on the name of the folder.
3. Select **Search** from the popup menu. This operation selects all of a folder's spectra for analysis (**Error! Reference source not found.**Figure 4).
4. When the **Search** tab appears, check that the **Spectra** field has the name of the folder, /users/*jdoe*/*Tutorial*.

Peptide Identification Search Parameters

5. **Search Engines**: Each search engine can be selected using the corresponding check-box. By default, all search engines are checked. Note that not all search engines support all search parameter choices, and that the search engines vary in speed and identification sensitivity. Mascot, in particular, is licenced only for a specific number of CPU cores, which can result in its search jobs taking longer to complete.

6. **Instrument**: Select an appropriate mass-spectrometer from the list. This search parameter captures the essential properties of the MS/MS spectra, from fragment ion mass-tolerance to maximum appropriate precursor charge-state (particularly important for MALDI spectra). Alternative fragmentation modes, such as ETD, are considered part of the instrument definition too. New instruments can be added, per user, by the PepArML administrator. For the example analysis, select **Waters Q-Tof** from the **Instrument** menu.

7. **Proteolytic Agent**: Select an appropriate proteolytic agent from the list. A variety of proteolytic enzymes and chemistries are supported by PepArML. Select **None** for intact protein or native peptide analyses. For the example analysis, the default, **Trypsin**, is an appropriate selection.

8. **Fixed Modifications**: (Multi-)select appropriate fixed modifications from the list. Fixed modifications are applied to every instance of an amino-acid from the protein sequence database. **Carbamidomethyl (C)** is selected by default, and is appropriate for the example analysis. The selected modifications are shown to the right.

9. **Variable Modifications**: (Multi-)select appropriate variable modifications from the list. Variable modifications indicate that the modified amino-acid should be considered in addition to the unmodified (fixed) form. **Oxidation (M)**, **Gln→pyro-Glu (N-term Q)**, **Glu→pyro-Glu (N-term E)**, and **Pyro-carbamidomethyl (N-term C)** are selected by default, which is appropriate for the example analysis. The selected modifications are shown to the right. Note that selecting a large number of variable modifications may incur a significant increase in search times.

10. **Sequence Database**: Select an appropriate protein sequence database from the list. The size and release or version of each sequence database is also shown, for reference. New sequence databases can be added, per user, by the PepArML administrator. For the example analysis, the **SwissProt (248MB – Release 2013_06)** protein sequence database should be selected.

11. **Peptide Candidate Selection**: The peptide candidate selection option specifies how peptide sequences should be matched with precursor m/z values. The standard options represent a precursor mass-tolerance of 2 Da, one missed cleavage, and charge states as indicated in the spectra datafile. Specificity of proteolytic cleavage is indicated in the candidate selection name: **Specific**, **Semispecific**, **Nonspecific**. Select **Specific** for intact protein analysis, and

Nonspecific for native peptide analysis. New candidate selection options can be added, per user, by the PepArML administrator. For the example analysis, the default, **Semispecific**, is an appropriate selection.

12. **Spectra**: The folder or spectra datafile to be analyzed. Usually this is set as indicated in Step 3 above, but if desired, can be changed when configuring the search. Both folders and spectra datafiles are valid here. Incomplete pathnames bring up a list of the folders and datafiles in the spectra repository.

PepArML Search Configuration Options

13. **Search Number**: Each spectra datafile may be analyzed multiple times using different parameters. This option specifies the search number to be assigned to the search. By default, searches are numbered sequentially from one. An asterisk indicates that the search number has already by used.

14. **Search Chunk Size**: Each spectra datafile is divided (chunked) into smaller pieces to better distribute the work over a heterogeneous collection of computers. Faster computers will complete more search chunks than slower computers. If the search chunk size is too small, the PepArML scheduler spends too much time managing the search jobs. If the search chunk size is too large, the search job is more likely to fail. Ideally, each search job should take several minutes. For the example analysis, the default, **200**, is an appropriate selection.

15. **Scheduler Priority**: Usually, the PepArML scheduler will run a user's search jobs in the order they are submitted. To run a new search jobs before previously queued search jobs, they can be given a high priority. Note that the scheduler priority only affects the user's search jobs. For the example analysis, the default, **0 (Normal)**, is an appropriate selection.

16. **Internal PepArML Decoy?** Usually, the PepArML scheduler will search the spectra against the target protein sequence database, a reversed decoy protein sequence database, and a randomized decoy protein sequence database. The peptide identifications from the randomized decoy protein sequence database are used internally by the PepArML result combiner to calibrate the heuristic and the machine-learning prediction confidence values. In the absence of the randomized decoy reuslts, the PepArML result combiner can calibrate the machine-learning prediction confidence values using the target peptide identifications, but this approach less reliable. The additional decoy search results are only used by the heuristic and PepArML combiners. For the example analysis, the default, **Yes (2 Decoy Searches)**, is an appropriate selection.

17. Click the **Search** button to submit the search. **Error! Reference source not found.**Figure 6 shows the search parameters for the example analysis.

BASIC PROTOCOL 3

MONITOR AND MANAGE THE SEARCH JOBS

A single PepArML peptide-identification analysis, consisting of many spectra datafiles, target and decoy protein sequence databases, multiple search engines, is decomposed into many search jobs. For an LS-MS/MS dataset *S* consisting of spectra datafiles *f* containing n_f MS/MS spectra; search chunk size parameter n_C; number of decoy databases $n_D \in \{1,2\}$; and number of search engines n_E, the number of search jobs n_J is

$$n_J = n_E\,(1+n_D)\sum_{f\in S}\lceil\frac{n_f}{n_c}\rceil.$$

For the example analysis using the parameters suggested in Basic Protocol 2, the analysis of 1058 MS/MS spectra is decomposed into 126 search jobs with corresponding result files in the results repository (**Results** tab, **Error! Reference source not found.**Figure 7). Once created, the search jobs are in one of four states: **Queued**, **Running**, **Error**, or **Done**. **Queued** jobs are waiting to run; **Running** jobs are currently assigned to a computational resource for execution; **Error** jobs have failed in some way; and **Done** jobs have completed. Failed jobs under the **Error** tab may have suffered an explicit failure in the execution of the job (denoted **Error**); the computational resource may have stopped sending regular heartbeat messages (denoted **Crashed**); or the user may have requested the search job be terminated (denoted **Terminating**, then **Killed**). Jobs in the **Error** and **Crashed** state are automatically requeued by the PepArML server up to three times. The PepArML server automatically schedules search jobs on the available compute resources subject to fair allocation of the resources between users.

Result files are created as empty files and populated when the corresponding search job completes, at which time the PepArML server checks the validity of the results and indicates the filesize, the number of spectra represented, and the number of peptide-spectrum-matches (PSMs). Rarely, the search job will be marked **Done** even though the PSMs cannot be parsed from the result file – in this case, the filesize is shown as non-zero, but the number of spectra and PSMs are left blank. We discuss how to monitor and manage the search jobs to successful completion in the following protocol.

Necessary Resources—A modern web-browser, such as Internet Explorer, Firefox, Chrome, or Safari is required. Users must register and login to PepArML (Support Protocol 1). Spectra must already have been uploaded to the PepArML server (Basic Protocol 1 or Alternative Protocol 1) and a peptide identification search configured and submitted (Basic Protocol 2).

1. Access the pending search job queue by clicking on the **Queue** tab. Note that it can sometimes take a minute or two for the first search jobs to appear, after submitting a PepArML search, depending on the load on the PepArML server.

2. Check individual search jobs by clicking on the job id hyperlink. The single job page contains the PepArML search configuration based on the parameters set in

Basic Protocol 2, the state history of the job, and any error messages generated when the job executes. The individual search job view can be accessed from any of the search job tabs. Click the web-browser back button or click on the **Queue** tab to navigate back to the list of pending search jobs.

3. Check whether the creation of search jobs is ongoing or complete by sorting by descending job id. Click on the **Queue** tab to access the list of pending search jobs, then click on the **Id** header of the job table and select **Sort** → **Set Key: Decreasing**. As the table refreshes periodically, new search jobs will be observed at the top of the table until all search jobs have been queued.

4. Check whether the search jobs are being scheduled to run. Click on the **Running** tab to access the list of running search jobs (**Error! Reference source not found.**Figure 8). It may take several minutes for compute resources to free up for your jobs, depending on the number of other PepArML searches running, but over time, you should see your jobs being scheduled and running on the available compute resources. Check that jobs corresponding to each search engine take an appropriate length of time.

5. Check whether any search jobs are consistently failing. Click on the **Error** tab to access the list of search jobs with an error. Sometimes the specified search options are incompatible with a specific search engine, or a specific compute resource is having trouble, such as a disk filling up. Check a few jobs' error messages by clicking on the job id hyperlink to access the single job page. Back on the **Error** tab, you can requeue crashed, killed, or error state jobs by clicking on a row and selecting **Requeue** from the popup menu and choosing **One, Page**, or **Search**. One requeues only the job corresponding to the clicked row; Page requeues all jobs on the current page of the table; while Search requeues all jobs from the search corresponding to the clicked row.

6. Check whether the search jobs are being completed. Click on the **Done** tab to access the list of running search jobs. Check that jobs corresponding to each search engine take an appropriate length of time.

7. Check whether the result files are being created. Click on the **Results** tab to show the results repository. The row corresponding to the spectra repository folder for the study, *Tutorial*, will also be present in the results repository. The folder will show the number of files (one per search job), the number that are empty (incomplete search jobs), and the percent of the files that are empty (**Error! Reference source not found.**Figure 7). This provides some notion of progress for your search. Once all search jobs complete and all files in a folder are non-empty, the PepArML combiner will be automatically configured and scheduled. Rarely, a job will complete without returning a result or will return a corrupted result file. In this case, the PepArML combiner will never be scheduled, or it will fail on the corrupted result file. Problematic results can be easily found by navigating into the result folder by clicking on its name, *Tutorial*, clicking the **MS/MS Spectra** header, and selecting **Sort** → **Set Key: Increasing**. Result files with no MS/MS Spectra are either empty or non-empty and corrupt. Check whether a search job is available in the system for such result files – click on the row and select **Find Job** in the popup menu. If no job is found or the job is marked **Done**, then a new search job can be created and queued for the empty or corrupt result file. Click the back-button to return to the table of result files, click on the row of the problematic result file, and select **Requeue** → **One** from the popup menu. Do not create multiple pending search jobs corresponding to the same result file.

ALTERNATE PROTOCOL 2

RUN SEARCH JOBS IN THE CLOUD

The Edwards lab compute cluster provides compute resources for the PepArML server, providing about 80 CPUs for PepArML search jobs. However, for large or urgent searches, or when many users are running searches at the same time, the freely provided compute resources may not be sufficient to complete the search in a reasonable time. PepArML is designed to support the use of remote compute resources in the cloud or in a university high-performance-computing center. We show here how to use Amazon Web Services (AWS) to boost the throughput of PepArML searches using only a credit card.

Necessary Resources—A modern web-browser, such as Internet Explorer, Firefox, Chrome, or Safari is required. Users must register for PepArML (Support Protocol 1). Spectra must already have been uploaded to the PepArML server (Basic Protocol 1 or Alternative Protocol 1) and a peptide identification analysis configured and submitted (Basic Protocol 2). Users should have verified that the search jobs are being scheduled and are completing successfully (Basic Protocol 3). Finally, users must have signed up for an EC2 capable account with Amazon Web Services at http://aws.amazon.com.

1. Login to AWS, navigate to the AWS Management Console, access the EC2 service, and select region “US East (N. Virginia)” if necessary.
2. Click “AMIs”, change the filter to “Public Images,” and search for “PepArML” using the search box. This should return a single PepArML Worker image. The current version is “PepArML Worker 1.6.6” with ID ami-fb9c5a92. Right-click the image and select “Spot Request” from the popup menu (**Error! Reference source not found.**Figure 9).
3. Select the C1 High-CPU Extra Large (c1.xlarge) instance type. The current price per instance hour is generally about 7 cents – so I generally bid 10 cents per instance hour. Initially, request just a few (spot) instances. Note that 10 instances provide approximately the same compute resource as the current Edwards lab cluster and at this price point, cost less about $17 a day (**Error! Reference source not found.**Figure 10).
4. Provide your PepArML username and password in the User Data field, on one line, separated by a space (**Error! Reference source not found.**Figure 11).
5. All other options can be left at their default values.

6. Verify that the PepArML scheduler is allocating jobs to the AWS instance and that the jobs are completing successfully. On the PepArML site, click on the **Running** tab to check running jobs (**Error! Reference source not found.**Figure 8). Amazon instances typically appear at the top of the table. It may take a few minutes for the instance to be started and the first search jobs to be scheduled and complete. The most common reason a running AWS instance fails to run search jobs is an incorrect PepArML username or password. If, after a few minutes, the AWS console shows the instance running, but no search jobs are scheduled on it, terminate the instance, and make another request, paying close attention to username and password entry or seek assistance from the PepArML administrator.

BASIC PROTOCOL 5

COMBINE SEARCH RESULTS USING PEPARML COMBINER

Once the search results are complete and all result files are populated, the PepArML combiner is automatically run to determine the best peptide-spectrum-match for each spectrum, estimate false-discovery-rates, and format the results as pepXML and other formats. However, in some circumstances, it may be desirable to run the PepArML combiner manually. First, if multiple search instances with different search numbers are being executed on the same spectra, the combiner will not be run until all results files in the folder are populated. If all the result files from one of the search instances are complete, its results may be combined by running the PepArML combiner manually. Second, if one of the search engines performs so poorly that it is preferable to remove its results from the analysis, the combiner can be manually configured to ignore the search engine's results. Lastly, when only some of the spectra datafiles in a folder should be considered, the combiner can be manually configured for this too.

Necessary Resources—A modern web-browser, such as Internet Explorer, Firefox, Chrome, or Safari is required. Users must register and login to PepArML (Support Protocol 1). Spectra must already have been uploaded to the PepArML server (Basic Protocol 1 or Alternative Protocol 1) and a peptide identification search configured and submitted (Basic Protocol 2). Finally, search jobs must have completed and the corresponding result files populated (Basic Protocol 3 and, optionally, Basic Protocol 4).

1. Navigate to the result repository by clicking on the **Results** tab. The **Results** tab will be highlighted and the table header will read **Results:** /users/*jdoe* to indicate the home folder of *Jane Doe*.

2. Click on the row corresponding to the folder, *Tutorial*, containing the result files to be combined, but not on the name of the folder.

3. Select **Combine** from the popup menu. This operation selects a folder's results files for combining.

4. When the **Combine** tab appears, check that the **Results Name** field has the name of the folder, /users/*jdoe/Tutorial*.

Combiner Parameters

5. **Results Name**: The folder of search results to be combined. Usually this is set as indicated in Step 3 above, but if desired, can be changed manually. To specify that the result files from only some of the spectra be considered, the Results Name may specify a partial filename and the wildcard symbol "*" after the folder, such as /users/*jdoe/Tutorial/17mix-**.

6. **Results Number**: Since multiple searches may be conducted in a single folder the Result Number of the desired search's results should be specified.

7. **Results From**: Select which search engines' results from the results folder and search instance to combine. If a search engine is known to perform poorly, based on the automatic combiner analysis, it can sometimes be useful to configure a new combiner analysis without the problematic search engine's results.

8. **Combiner Methods**: After merging the PSMs from each search engine, the PepArML combiner can use any of three methods to select the best PSM per spectrum: a search engine's primary score metric, a search engine voting and FDR-based heuristic, and the PepArML machine-learning based predictor. Usually, all three methods are used.

9. **Combiner Number**: Just as multiple searches may be carried out on a folder of spectra datafiles, multiple combiner runs may be carried out on a folder's results. The automatically configured combiner instantiation will use the same combiner number as the search instance.

10. **Decoy Results**: PepArML can be configured to use only one of the decoy results, even if two decoy searches were carried out, by changing this option. See Step 16 of Basic Protocol 2 for more on this. Generally, if two sets of decoy results are available, it is better to use the internal decoy; and if only one set of decoy results is available, there is no choice to be made. In most cases, this option should be left at **Auto**.

11. **Scheduler Priority**: Usually, the PepArML scheduler will run a user's combiner jobs in the order they are requested. However, just as for search jobs, if a new combiner job should be scheduled before already queued combiner jobs, it can be given a higher priority. Note that combiner jobs are preferentially scheduled before search jobs. In most cases, this option should be left at **0 (Normal)**.

12. Click the **Combine** button to run the combiner.

13. Rarely, the combiner analysis will fail due to missing, empty, or corrupt result files. In this case, the combiner job will appear in the **Error** tab and the error message will give some indication of the issue. Usually, following Step 7 of Basic Protocol 3 will uncover the problematic result file that needs to be recomputed. Once the problematic result file is correctly populated, the combiner job itself, under the **Error** tab, can be requeued.

14. An email is sent to the user's email address once the combiner analysis is complete. For most PepArML analyses, no user intervention is required after Basic Protocol 2 is complete until this email is received. The email contains a link to download the results of the combiner analysis. Alternatively, the PepArML result file can be downloaded from the result repository (**Error! Reference source not found.**Figure 12).

GUIDELINES FOR UNDERSTANDING RESULTS

PepArML result files, such as *Tutorial.peparml.1.zip*, are zip files containing many different files and results, including a comparison of the performance of individual search engines, the heuristic, and PepArML combiners; spectrum based and protein based results for individual search engines, the heuristic and PepArML combiners in comma-separated-value (CSV) and XML formats; an evaluation of the relative importance of PSM features in each PepArML training iteration; and file(s) recording the parameters of the search and the association between (PepArML) protein accessions and protein descriptions from the protein sequence database FASTA file.

After the results of the different search engines are merged, based on spectrum identifiers, the combiner must select at most one PSM per spectrum, and then evaluate the selected PSMs to estimate their spectral false-discovery-rates. For each search engine, its primary score is used to choose the best PSM per spectrum and to evaluate the selected PSMs. For Mascot, OMSSA, and Tandem with native, k-score, and s-score scoring, the primary score is the E-value. For MyriMatch, the primary score is the so-called mvh score, while for Inspect, the MSGF spectral probability of the PSM is used as the primary score. Spectral false-discovery-rates are estimated using the Gygi method (Peng *et al.*, 2003; Elias and Gygi, 2007) and the reversed protein sequence database decoy. The voting and FDR heuristic selects PSMs per spectrum based on the number of agreeing search engines, with a penalty for the number of decoy identifications from the internal decoy search. The PepArML combiner uses a heuristic unsupervised training procedure to learn which PSMs are true and which are not, and produces a confidence value between zero and one for the target and decoy PSMs. While the heuristic and PepArML combiners use the extra set of decoy results internally, the estimation of FDR for the selected PSMs is carried out using the same method used for the individual search engines. Since the evaluation of each combiner is carried out in an identical manner, the number of spectra at specific FDR thresholds can be compared.

The performance of each combiner method can be seen in the file stats.csv, which lists the number of spectra with identifications at 1%, 5%, and 10% FDR. Table 1 shows these values for the example analysis of *17mix-test2*. A similar comparison is presented in the image file fdrcurves.png, which shows the *q*-value curve of the spectrum and peptide FDR values versus the number of identified spectra and peptides (**Error! Reference source not found.**Figure 13).

For each combiner method, the peptide identifications are output as a CSV format file and a pepXML file, with names *method*-pred-efdr.csv and *method*.pep.xml. In addition, two protein centric files are generated - a CSV file of the peptide identifications grouped by protein and the output of Protein Prophet (Nesvizhskii et al., 2003) run on the pepXML file, with names *method*-prot.csv and *method*.prot.xml. For example, the PepArML combiner analysis results are in the files peparml-pred-efdr.csv, peparml.pep.xml, peparml-prot.csv, and peparml.prot.xml.

The CSV format peptide identifications list the selected PSMs, one per spectrum. The first few columns provide the spectral identifiers; experimental and theoretical precursor m/z and charge values; the identified peptide sequence and its modifications; the protein accessions associated with the peptide; and the primary metrics of identification performance - nagree, the number of search engines with this identification, and estfdr, the estimate of the spectrum FDR for all identifications with this primary score or better. The next columns provide the various features from each search engine. Feature suffixes indicate the search engine and search number responsible for the feature, with Mascot indicated by “m”, Tandem with native scoring by “t”, k-score scoring by “k”, s-score scoring by “s”, OMSSA by “o”, MyriMatch by “y”, and Inspect by “i”. Search engine-based feature names are derived directly from each search engine's terminology, where possible. Missing values are shown as blank. Following the search engine features are the features available for all peptide-spectrum-matches (PSMs) including digest specificity features, isotope cluster features, proteotypic peptide features (Mallick et al., 2006), and retention-time modeling features. A full list of the PSM features and their definitions can be found on the PepArML support web-site (see Table 1).

The CSV format peptide identification result files can readily manipulated by loading into Excel, selecting all rows (Ctrl-A), and using the AutoFilter feature (Data → Filter) to select rows by various criteria. Peptide identifications can be filtered at 1% FDR using the AutoFilter pull-down menu on the estfdr column and selecting Number Filters → Less Than Or Equal To → 0.01. The pepXML format peptide identifications do not capture all of the PSM features, providing the search engines' primary scores for each PSM, the number of agreeing search engines (nagree) and the estimated FDR of the identification (estfdr). In addition, the Peptide Prophet (Keller et al., 2002) probability in the pepXML file is populated with (1-estfdr), to aid with downstream analysis using Trans-Proteomic-Pipeline tools.

The CSV format protein identification files, *method*-prot.csv, are similar to the peptide identification result files, but are grouped by identified proteins. Only proteins identified by at least two distinct peptides with spectral FDR at most 10% are shown. Protein metrics, with respect to 10% FDR filtered peptides, include number of distinct peptides, number of non-overlapping peptides, and % coverage. All PSMs for a protein are shown, regardless of their FDR and PSM features are similar to those for the CSV format peptide identifications. Proteins are ordered by decreasing distinct peptides, at 10% FDR. Proteins with the same or a subset of the peptide identifications are placed in a protein group with their containing proteins. Within a protein group, the protein with the most distinct peptides at 10% FDR is marked with “>>”, proteins with the same peptide identifications are marked with “=>”, and proteins with a strict subset of the peptide identifications are marked with “->”. The first protein group, top to bottom, to use a peptide's identifications marks them with “*” in the “first” column. Protein groups with only a few low-quality peptide identifications marked as first are usually artifacts. Peptide start and end positions, relative to the protein group's lead protein, are also shown here, and each protein's peptides are sorted by start and end amino-acid. Excel Auto-Filter can be applied to a single protein group's peptide identifications by selecting the corresponding rows. Note that one-hit-wonder peptide assignments are omitted from the protein report by the two-distinct peptides at 10% FDR criteria.

The protXML format protein identification files, *method*.prot.xml, are generated by Protein Prophet run on the pepXML format peptide identifications using an estimated FDR based Peptide Prophet probability score of (1-estfdr) filtered at “probability” 0.9.

The relative importance of PSM features in each PepArML iteration can be evaluated by checking the infogain.png figure, shown for the example analysis in **Error! Reference source not found.**Figure 14. A full list of the PSM features and their definitions can be found on the PepArML support web-site (see Table 1). The figure displays the information gain of each feature for each iteration of the unsupervised PepArML training heuristic. The green dot indicates the first iteration and the red dot indicates the iteration used in the final results. Since the initial heuristic training labels are heavily influenced by agreeing search engines and strong search engine scores, we expect that the primary scores for each search engine and the nagree feature to have the highest information gain. Usually, the information gain of these features is in the 0.2–0.4 range. For datasets with high resolution precursor ion measurements, the information gain of massdiff will also generally be high.

Finally, the searchparams.ini specifies the search parameters used in the search, and the proteins.txt file provides a mapping from the accessions PepArML reports to the full definition line in the protein sequence database FASTA file.

Distinguishing Features of PepArML Peptide Identifications

The peptide identifications assigned by PepArML provide a number of additional features not normally computed by sequence database search engines, which have utility not only for the machine-learning-based determination of correct peptide identifications, but also for downstream data-processing and presentation pipelines. First, the number of agreeing search engines is output for all combiner methods, in addition to the estimated FDR of the PSMs. Since none of the combiner methods explicitly constrain the extent of search engine agreement, setting a high threshold on the number of agreeing search engines can boost the specificity of peptide identification assignments, beyond that captured by the estimated FDR value. Second, in support of quantitation workflows, PepArML extracts and reports iTRAQ and TMT reporter ion intensities when search configurations specify these modifications.

Third, PepArML provides a number of features that evaluate the experimental precursor ion with respect to the theoretical isotope cluster of the assigned peptide. The “c13massdiff” feature provides the absolute value of the mass distance to the closest theoretical isotope cluster peak; the “c13peak” feature provides the peak number of the closest theoretical isotope cluster peak, with 0 representing the monoisotopic peak and 1 representing the single ^{13}C peak; the “c13relint” feature provides the theoretical relative abundance of the closest theoretical isotope cluster peak; and the “icscore” provides a χ^2 goodness-of-fit match score for the precursor's experimental and theoretical isotope clusters.

Fourth, PepArML models experimental retention times and provides features for the predicted retention time, “rtpred,” and the difference between the predicted and experimental retention time, “rtdelta,” for all PSMs. PepArML fits a linear regression model to the experimental retention-time for initial high-confidence PSMs based on theoretical peptide physiochemical values and amino-acid compositions. The retention time modeling features and the experimental retention time, “retention_time,” often provide an important orthogonal signal for determining correct peptide identifications in the machine-learning model. **Error! Reference source not found.**Figure 14 shows that for the example analysis, the “rtdelta” feature has a strong information gain value, suggesting it may help distinguish correct peptide identifications.

Lastly, PepArML computes a post-translational modification site-localization score for selected PSMs with modifications. The “siteprob” feature evaluates all identified modification forms of the selected PSM's peptide, assessing the PepArML prediction confidence associated with each form, and distributing the confidence to each modification site. Scores of 100 indicate confidently placed modifications, with a score of 50 indicating equal likelihood for two potential sites, often observed for a single phosphorylation on adjacent residues. Scores of 0 indicate that the site was modified in a PSM, but that the PepArML prediction confidence of the PSM was very low.

Recognizing Heuristic Training Failure

The unsupervised heuristic machine-learning training procedure used by the PepArML combiner can fail in two ways. First, the heuristic construction of the initial training set may fail to identify any proteins with at least two distinct peptides that all search engines agree on. In this case, the iterative training procedure cannot get started and the PepArML peptide and protein identification files will be empty. This is an unusual condition, and usually reflects a difficult or small dataset or incorrect search parameters. Usually, the results from individual search engines or the voting and FDR based heuristic will suggest a reason for the lack of agreement. The second failure mode is when the initial set of confidently identified proteins is too large and PepArML machine-learning model considers too many PSMs to be correct. In this case the infogain.png plot will show very low information gain values for normally important features, such as nagree and search engine primary scores, or high information gain values for normally irrelevant features. For this failure, the fdrcurves.png plot will not show the usual leveling-off behavior and the number of spectra will continue to rise rapidly with FDR. This second failure mode is rare but typically occurs for very large datasets for which two distinct, unanimous peptides is not a strong enough criteria to ensure that only high-confidence proteins are used to infer training labels.

COMMENTARY

Background Information

The PepArML meta-search peptide identification platform is based on three major technologies: a peptide identification search engine configuration and execution abstraction layer; a web-based user-interface, search job scheduler, and data transfer manager for local and remote compute resources; and an unsupervised, model-free, machine-learning based results combiner. We briefly describe these components to provide context.

The search engine abstraction layer automatically constructs search engine configuration files and input spectra datafiles for each search engine from a simple unified peptide identification configuration. This abstraction significantly increases the reliability of search engine execution and consistency of peptide identification results, but it necessarily does not use unique search engine capabilities – the resulting peptide identifications, unique to a single search engine, would not be consistent with the principle of search engine consensus increasing peptide identification confidence. Even so, the abstraction layer cannot make the search engines entirely consistent in their interpretation of parameters, as search engines support for specific types of modifications and mass-error units varies. Nevertheless, the automated construction of configuration files and input spectra datafiles ensures that the peptide identification searches are carried out as reliably and consistently as possible.

The web-based user-interface, search job scheduler, and data-transfer manager not only enables remote access to the PepArML meta-search tool, but also makes it possible to utilize compute resources, such as cloud or university high-performance computing centers, remote to the user *and* the PepArML server. PepArML defines a permission structure which enables users to provision their own compute resources to carry out search jobs, without publicly sharing details of spectra or search configuration, and which scales to hundreds of simultaneous search jobs. The scheduler reserves user-provided compute resources for the users' jobs only, but shares the centralized resources fairly between all users. Critical for the success of this approach is the realization that such a distributed computing model must account for heterogeneous compute resources of different speeds and the potential for search jobs or compute resources to fail unexpectedly. At scale, failed search jobs become common and discovering and redoing a few failed searches amongst thousands of successful jobs becomes a significant burden. The PepArML infrastructure handles most failures automatically, and provides easy ways to find and redo failed searches when necessary.

The PepArML results combiner uses a supervised machine-learning technique, Random Forrest (Breiman, 2001), to learn the properties of and then predict the correct peptide identifications in the merged peptide identification results of multiple search engines. The Random Forrest technique is termed model-free because it does not require any prior knowledge of the distribution or covariance of its features to be applied successfully. This is important for PepArML because it allows us to add a variety of potentially discriminating PSM features without concern for a formal model. Rather than relying on the machine-learning technique to generalize from prescribed training datasets, we use an automated training heuristic to determine high-confidence protein identifications, labeling their peptide identifications correct and rest incorrect, before carrying out supervised learning. After training, the prediction confidence is used to determine a refined set of high-confidence proteins, and the process iterated (**Error! Reference source not found.**Figure 15). This unsupervised training heuristic is quite powerful, as allows the PepArML combiner to adapt to the specific characteristics of each dataset, using each PSM feature to the extent it is useful. Consequently, the PepArML combiner is able to discover properties of LC-MS/MS datasets, such as high resolution precursor measurements or predictable retention times that help distinguish correct peptide identifications.

Finally, the PepArML results combiner estimates the statistical significance of its peptide identifications using same procedure regardless of the search engines, making it possible to compare individual search engines and the PepArML combiner for peptide identification performance on a level playing field.

While this meta-search model for peptide identification search has significant advantages, not all applications of peptide identification searches match its strengths. PepArML is not designed to support the "blind" identification of post-translational modifications by large scale enumeration, refinement searches, or proteogenomics, as it cannot take advantage of the special features in some search engines that facilitate these analyses. Similarly, PepArML will not perform well on datasets with few identifications or identifications that do not tend to group by protein sequence. Such datasets make it difficult to select a high confidence protein set to seed the unsupervised PepArML training heuristic.

Critical Parameters and Troubleshooting

Spectra Datafiles—PepArML requires spectra datafiles of centroided peak-lists in one of the common open formats and will flag spectra with **!!ERROR!!** if they cannot be interpreted successfully. In addition to issues such as file truncation and corruption, PepArML requires that the format of the datafile correspond with the file extension, including compression, and that spectrum scan numbers correctly and uniquely identify each spectrum in the datafile. For the XML formats, this is straightforward, but for MGF files, in particular, there is no convention for how this information should be represented. PepArML uses a number of rules to extract scan numbers from MGF file TITLE fields, but this can be an unreliable process.

PepArML assumes that each spectra datafile corresponds to a single acquisition and that the spectra in each file have distinct scan numbers. Copies of a spectrum with different charge-states are permitted, but otherwise repeated scan numbers are not permitted. Do not merge many spectral datafiles into one monolithic MGF file, as this generally results in repeated scan numbers.

All things being equal, the optimal PepArML spectra datafiles are those generated from the vendor format spectra by the batch upload tool (Alternative Protocol 1) – compressed mzXML format, with centroided MS *and* MS/MS spectra with retention times. The MS spectra enable the precursor isotope cluster to be evaluated and the retention times can be difficult to extract reliably from MGF format spectra datafiles.

Study Layout—All spectra datafiles for a particular study should be grouped in a single folder. Since there is some randomness in the machine-learning-based combiner training algorithm, it is preferable to ensure that all datafiles in a particular study be analyzed together by the PepArML combiner, rather than introducing an additional source of variability. When configuring a PepArML search, specify the study folder, rather than individual datafiles, as this ensure all datafiles are searched consistently.

Search Parameters—It is important to remember that PepArML uses traditional search engines behind the scenes and doesn't just analyze each spectrum once – with seven search engines and two decoys, each spectrum is searched at least 21 times! As such, the consequences of expensive parameter choices, such as many variable modifications or nonspecific digests, are magnified for PepArML searches. Users are encouraged to choose search parameters judiciously to avoid these issues.

Heuristic Training Failure—The unsupervised PepArML training heuristic can fail for a variety of reasons. While this can be detected as described in the GUIDELINES FOR UNDERSTANDING RESULTS section, determining the root cause of the failure will often suggest a remedy. In order to get started, PepArML training requires at least two unanimous peptide identifications representing distinct peptides on the same protein with strong primary search engine scores. Sometimes, when the dataset is of poor quality or the search parameters poorly chosen, none of the search engines make many high-quality peptide identifications, in this case, the issue cannot be resolved at the combiner stage of the process. On the other hand, peptide identification unanimity can be just as easily compromised when one or two search engines perform poorly, and in this case, the training failure can often be resolved by excluding the problematic search engines, identified using stats.csv, from the combining step. Basic Protocol 5 describes how to manually configure and run the PepArML combiner, and in particular, Step 7 describes the selection of the search engines' results for combining.

ACKNOWLEDGEMENT

The development of the PepArML meta-search platform was supported NIH Grant R01 CA126189.

LITERATURE CITED

Breiman L. Random forests. Machine Learning. 2001; 45(1):5–32.

Craig R, Beavis RC. TANDEM: matching proteins with tandem mass spectra. Bioinformatics. 2004; 20(9):1466–1467. [PubMed: 14976030]

Edwards N, Wu X, Tseng C-W. An unsupervised, Model-Free, Machine-Learning combiner for peptide identifications from tandem mass spectra. Clinical Proteomics. 2009; 5(1)

Elias JE, Gygi SP. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. Nature methods. 2007; 4(3):207–214. [PubMed: 17327847]

Geer LY, Markey SP, Kowalak JA, Wagner L, Xu M, Maynard DM, Yang X, Shi W, Bryant SH. Open mass spectrometry search algorithm. J. Proteome Res. 2004; 3(5):958–964. [PubMed: 15473683]

Keller A, Nesvizhskii AI, Kolker E, Aebersold R. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. Anal. Chem. 2002; 74(20):5383–5392. [PubMed: 12403597]

Kessner D, Chambers M, Burke R, Agus D, Mallick P. ProteoWizard: open source software for rapid proteomics tools development. Bioinformatics. 2008; 24(21):2534–2536. [PubMed: 18606607]

Kim S, Gupta N, Pevzner PA. Spectral probabilities and generating functions of tandem mass spectra: A strike against decoy databases. Journal of Proteome Research. 2008; 7(8):3354–3363. [PubMed: 18597511]

MacLean B, Eng JK, Beavis RC, McIntosh M. General framework for developing and evaluating database scoring algorithms using the TANDEM search engine. Bioinformatics. 2006; 22(22): 2830–2832. [PubMed: 16877754]

Mallick P, Schirle M, Chen SS, Flory MR, Lee H, Martin D, Ranish J, Raught B, Schmitt R, Werner T, Kuster B, Aebersold R. Computational prediction of proteotypic peptides for quantitative proteomics. Nature Biotechnology. 2006; 25(1):125–131.

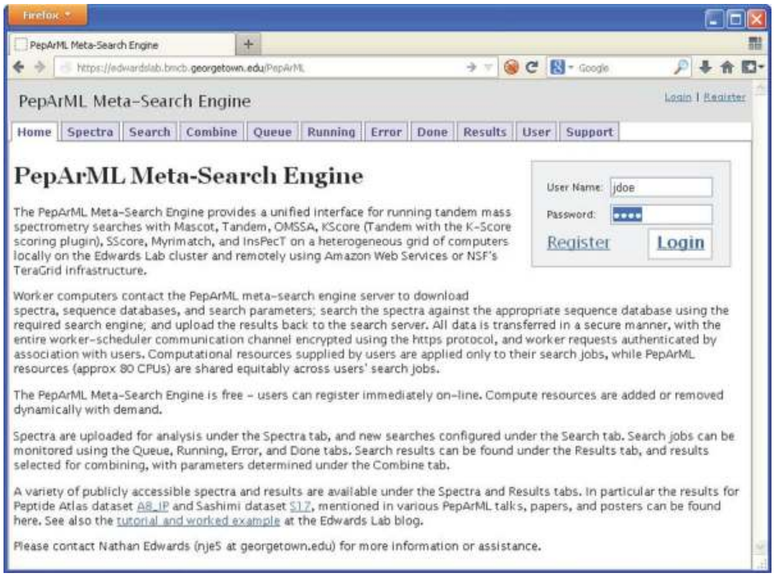
Nesvizhskii AI, Keller A, Kolker E, Aebersold R. A statistical model for identifying proteins by tandem mass spectrometry. Anal. Chem. 2003; 75(17):4646–4658. [PubMed: 14632076]

Peng J, Elias JE, Thoreen CC, Licklider LJ, Gygi SP. Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (LC/LC–MS/MS) for Large-Scale protein analysis: the yeast proteome. Journal of Proteome Research. 2003; 2(1):43–50. [PubMed: 12643542]

Perkins DN, Pappin DJ, Creasy DM, Cottrell JS. Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis. 1999; 20(18):3551–3567. [PubMed: 10612281]

Tabb DL, Fernando CG, Chambers MC. MyriMatch: Highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis. J. Proteome Res. 2007; 6(2):654–661. [PubMed: 17269722]

Tanner S, Shu H, Frank A, Wang LC, Zandi E, Mumby M, Pevzner PA, Bafna V. InsPecT: identification of post translationally modified peptides from tandem mass spectra. Anal. Chem. 2005; 77(14):4626–4639. [PubMed: 16013882]



User Name: jdoe

PepArML Meta-Search Engine

The PepArML Meta-Search Engine provides a unified interface for running tandem mass spectrometry searches with Mascot, Tandem, OMSSA, KScore (Tandem with the K-Score scoring plugin), SScore, Myrimatch, and InsPecT on a heterogeneous grid of computers locally on the Edwards Lab cluster and remotely using Amazon Web Services or NSF's TeraGrid infrastructure.

Worker computers contact the PepArML meta-search engine server to download spectra, sequence databases, and search parameters; search the spectra against the appropriate sequence database using the required search engine; and upload the results back to the search server. All data is transferred in a secure manner, with the entire worker-scheduler communication channel encrypted using the https protocol, and worker requests authenticated by association with users. Computational resources supplied by users are applied only to their search jobs, while PepArML resources (approx 80 CPUs) are shared equitably across users' search jobs.

The PepArML Meta-Search Engine is free – users can register immediately on-line. Compute resources are added or removed dynamically with demand.

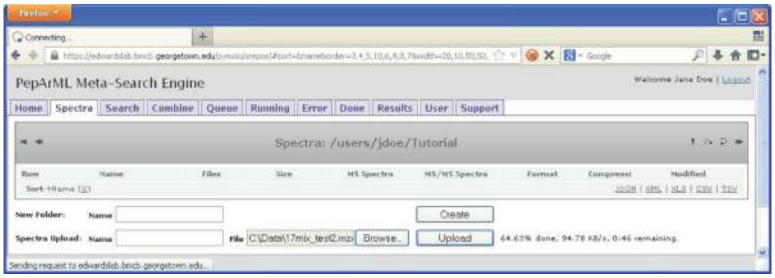
Spectra are uploaded for analysis under the Spectra tab, and new searches configured under the Search tab. Search jobs can be monitored using the Queue, Running, Error, and Done tabs. Search results can be found under the Results tab, and results selected for combining, with parameters determined under the Combine tab.

A variety of publicly accessible spectra and results are available under the Spectra and Results tabs. In particular the results for Peptide Atlas dataset A8_IP and Sashimi dataset S17, mentioned in various PepArML talks, papers, and posters can be found here. See also the tutorial and worked example at the Edwards Lab blog.

Please contact Nathan Edwards (nje5 at georgetown.edu) for more information or assistance.

Figure 1.

PepArML homepage.



PepArML Meta-Search Engine

Welcome Jane Doe | Logout

Spectra: /users/jdoe/Tutorial

Sort: +Name (X)

New Folder: Name

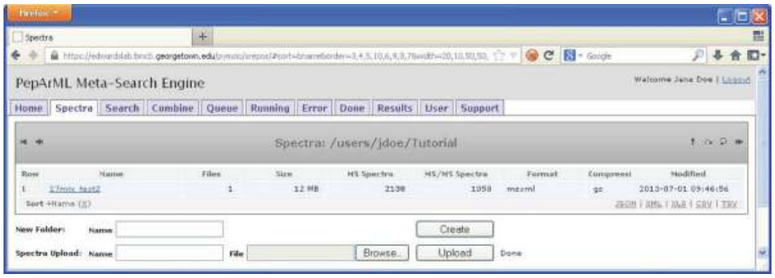
Spectra Upload: Name File C:\Data\17mix_test2.mzx

64.63% done, 94.78 KB/s, 0:46 remaining.

Sending request to edwardslab.bmcb.georgetown.edu...

Figure 2.

Uploading *17mix-test2.mzxml.gz* to the *Tutorial* folder of the spectra repository.



Spectra: /users/jdoe/Tutorial

Row	Name	Files	Size	MS Spectra	MS/MS Spectra	Format	Compressi	Modified
1	17mix-test2	1	12 MB	2138	1058	mzxml	gz	2013-07-01 09:46:56

Sort +Name (X)

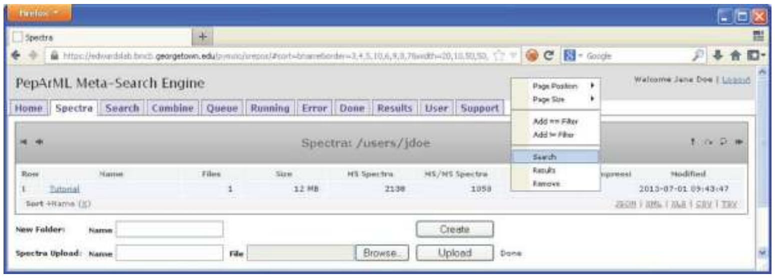
JSON | XML | XLS | CSV | TSV

New Folder: Name

Spectra Upload: Name File

Figure 3.

Completed upload of datafile *17mix-test2.mzxml.gz* to the *Tutorial* folder of the spectra repository.



PepArML Meta-Search Engine

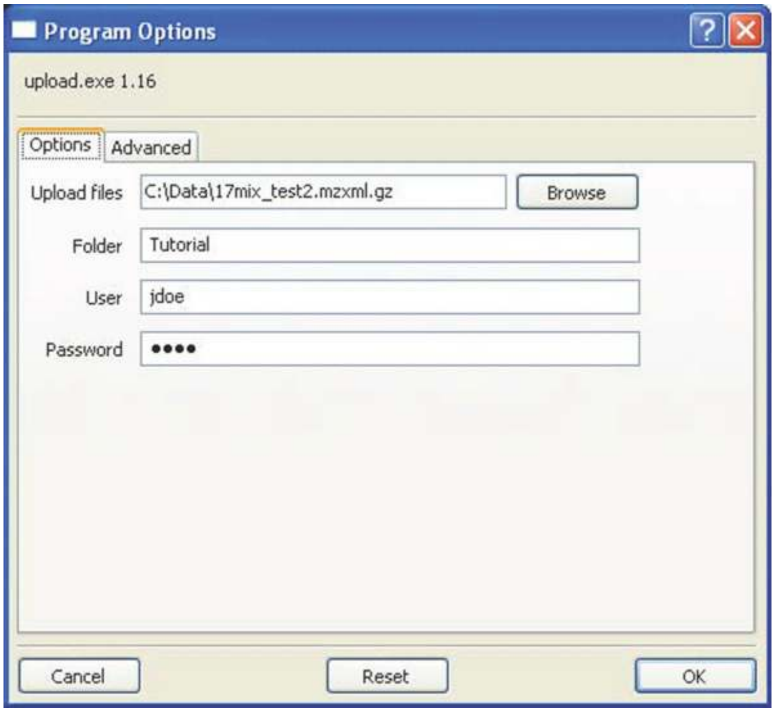
Spectra: /users/jdoe

Row	Name	Files	Size	MS Spectra	MS/MS Spectra	pressed	Modified
1	Tutorial	1	12 MB	2138	1058		2013-07-01 09:43:47

Sort +Name (X)

Figure 4.

Tutorial folder of spectra repository populated with spectra *17mix-test2* and selection of Search from the popup menu.



Program Options

upload.exe 1.16

Upload files C:\Data\17mix_test2.mzxml.gz

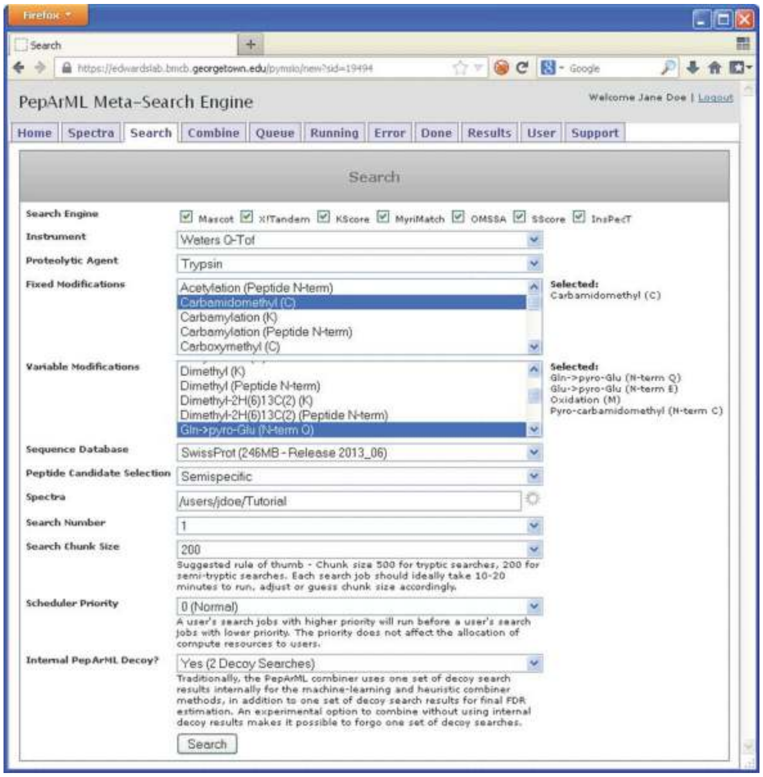
Folder Tutorial

User jdoe

Password ●●●●

Figure 5.

Batch upload of *17mix-test2.mzxml.gz* to the *Tutorial* folder of the spectra repository.



PepArML Meta-Search Engine

Welcome Jane Doe | Logout

Search

Search Engine ☑ Mascot ☑ X!Tandem ☑ KScore ☑ MyriMatch ☑ OMSSA ☑ SScore ☑ InsPecT

Instrument: Waters Q-Tof

Proteolytic Agent: Trypsin

Fixed Modifications:
Acetylation (Peptide N-term)
Carbamidomethyl (C)
Carbamylation (K)
Carbamylation (Peptide N-term)
Carboxymethyl (C)

Selected:
Carbamidomethyl (C)

Variable Modifications:
Dimethyl (K)
Dimethyl (Peptide N-term)
Dimethyl-2H(6)13C(2) (K)
Dimethyl-2H(6)13C(2) (Peptide N-term)
Gln->pyro-Glu (N-term Q)

Selected:
Gln->pyro-Glu (N-term Q)
Glu->pyro-Glu (N-term E)
Oxidation (M)
Pyro-carbamidomethyl (N-term C)

Sequence Database: SwissProt (246MB - Release 2013_06)

Peptide Candidate Selection: Semispecific

Spectra: /users/jdoe/Tutorial

Search Number: 1

Search Chunk Size: 200

Suggested rule of thumb - Chunk size 500 for tryptic searches, 200 for semi-tryptic searches. Each search job should ideally take 10-20 minutes to run, adjust or guess chunk size accordingly.

Scheduler Priority: 0 (Normal)

A user's search jobs with higher priority will run before a user's search jobs with lower priority. The priority does not affect the allocation of compute resources to users.

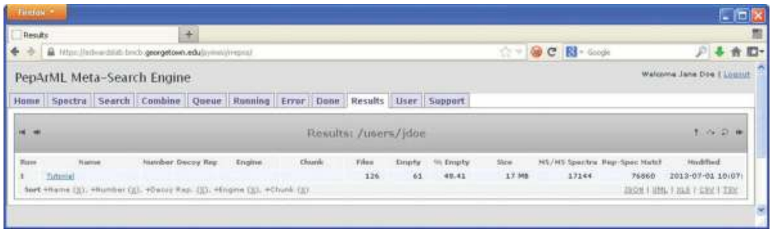
Internal PepArML Decoy?: Yes (2 Decoy Searches)

Traditionally, the PepArML combiner uses one set of decoy search results internally for the machine-learning and heuristic combiner methods, in addition to one set of decoy search results for final FDR estimation. An experimental option to combine without using internal decoy results makes it possible to forgo one set of decoy searches.

Search

Figure 6.

Search parameters for the example analysis of *17mix-test2*.



PepArML Meta-Search Engine

Welcome Jane Doe | Logout

Results: /users/jdoe

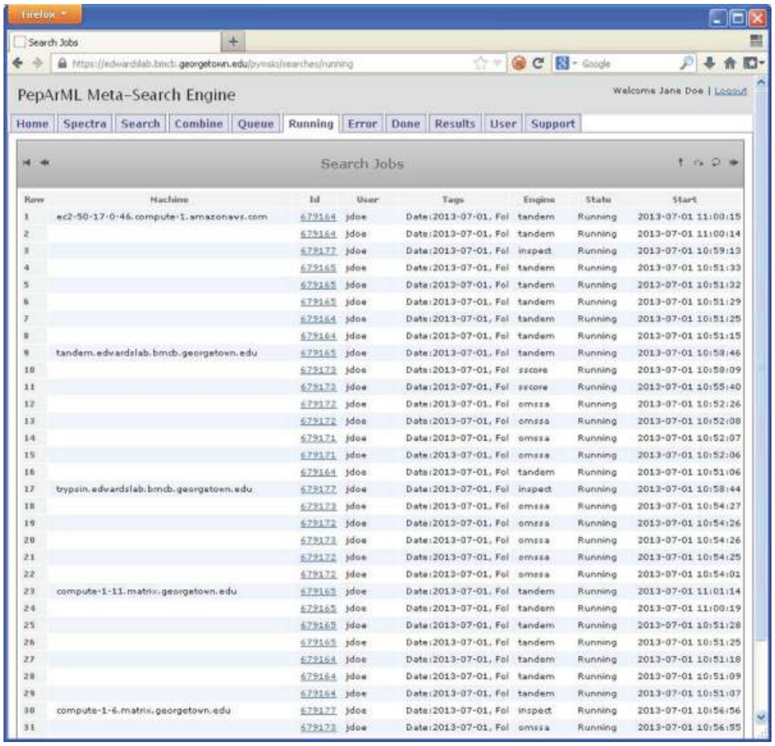
Row	Name	Number	Decoy Rep	Engine	Chunk	Files	Empty	% Empty	Size	MS/MS Spectra	Pep-Spec Match	Modified
1	Tutorial					126	61	48.41	17 MB	17144	76860	2013-07-01 10:07:

Sort +Name (X), +Number (X), +Decoy Rep. (X), +Engine (X), +Chunk (X)

JSON | XML | XLS | CSV | TSV

Figure 7.

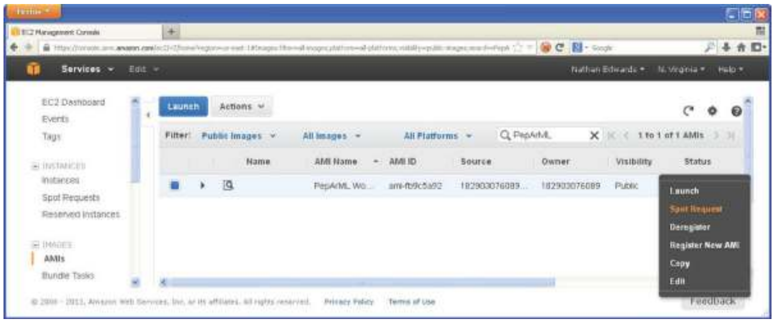
Tutorial folder of results repository showing progress of the example analysis.


Firefox
Search Jobs
+
https://edwardslab.bmcb.georgetown.edu/pymsio/searches/running
Google
PepArML Meta-Search Engine
Welcome Jane Doe | Logout
Home
Spectra
Search
Combine
Queue
Running
Error
Done
Results
User
Support
Search Jobs

Row	Machine	Id	User	Tags	Engine	State	Start
1	ec2-50-17-0-46.compute-1.amazonaws.com	679164	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 11:00:15
2		679164	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 11:00:14
3		679177	jdoe	Date:2013-07-01, Fol	inspect	Running	2013-07-01 10:59:13
4		679165	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:33
5		679165	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:32
6		679165	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:29
7		679164	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:25
8		679164	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:15
9	tandem.edwardslab.bmcb.georgetown.edu	679165	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:58:46
10		679173	jdoe	Date:2013-07-01, Fol	sscore	Running	2013-07-01 10:58:09
11		679173	jdoe	Date:2013-07-01, Fol	sscore	Running	2013-07-01 10:55:40
12		679172	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:52:26
13		679172	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:52:08
14		679171	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:52:07
15		679171	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:52:06
16		679164	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:06
17	trypsin.edwardslab.bmcb.georgetown.edu	679177	jdoe	Date:2013-07-01, Fol	inspect	Running	2013-07-01 10:58:44
18		679173	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:54:27
19		679172	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:54:26
20		679173	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:54:26
21		679172	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:54:25
22		679172	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:54:01
23	compute-1-11.matrix.georgetown.edu	679165	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 11:01:14
24		679165	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 11:00:19
25		679165	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:28
26		679165	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:25
27		679164	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:18
28		679164	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:09
29		679164	jdoe	Date:2013-07-01, Fol	tandem	Running	2013-07-01 10:51:07
30	compute-1-6.matrix.georgetown.edu	679177	jdoe	Date:2013-07-01, Fol	inspect	Running	2013-07-01 10:56:56
31		679173	jdoe	Date:2013-07-01, Fol	omssa	Running	2013-07-01 10:56:55

Figure 8.

Example analysis search jobs running on the Edwards lab cluster (edwardslab.bmcb.georgetown.edu), Amazon Web Services (amazonaws.com), and Georgetown HPC computing resources (maxtrix.georgetown.edu).

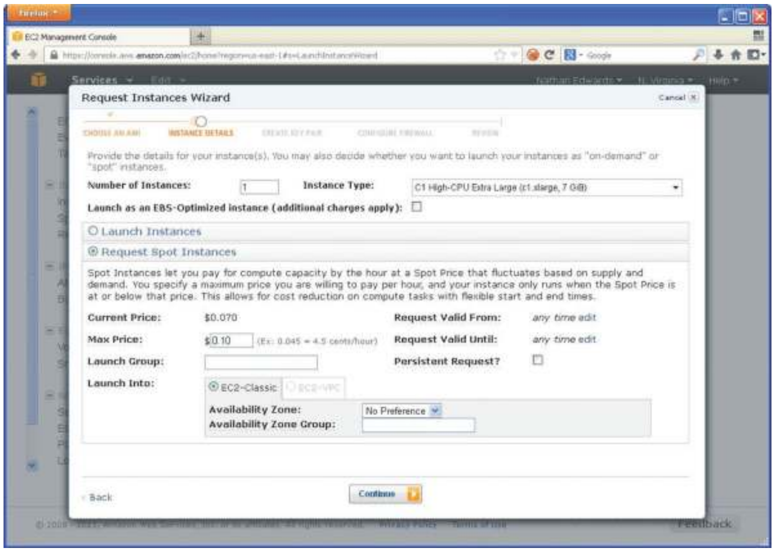


Filter: Public Images All images All Platforms PepArML 1 to 1 of 1 AMIs

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status
	PepArML Wo...	ami-fb9c5a92	182903076089...	182903076089	Public	

Figure 9.

Selection of PepArML Worker Amazon Machine Image for spot request.



Request Instances Wizard

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 Instance Type: C1 High-CPU Extra Large (c1.xlarge, 7 GiB)

Launch as an EBS-Optimized instance (additional charges apply): □

○ Launch Instances

● Request Spot Instances

Spot Instances let you pay for compute capacity by the hour at a Spot Price that fluctuates based on supply and demand. You specify a maximum price you are willing to pay per hour, and your instance only runs when the Spot Price is at or below that price. This allows for cost reduction on compute tasks with flexible start and end times.

Current Price: $0.070 Request Valid From: any time edit

Max Price: $0.10 (Ex: 0.045 = 4.5 cents/hour) Request Valid Until: any time edit

Launch Group: Persistent Request? □

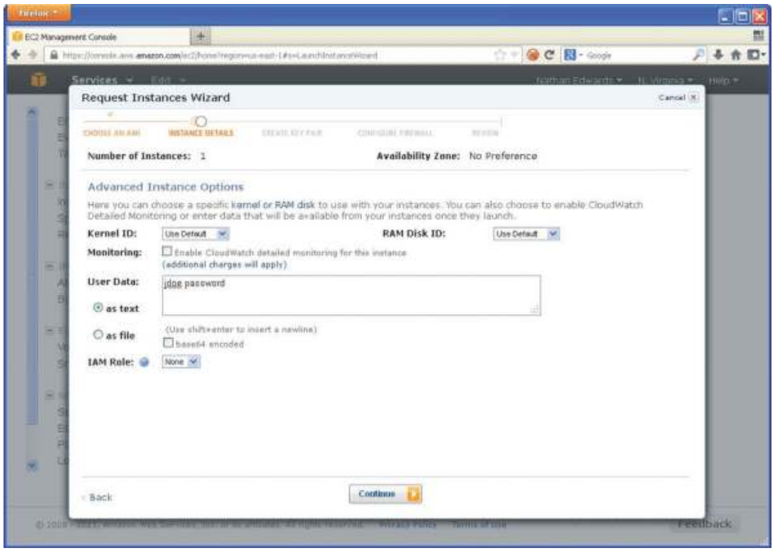
Launch Into: ● EC2-Classic ○ EC2-VPC

Availability Zone: No Preference

Availability Zone Group:

Figure 10.

Setting the Amazon spot request instance type and bid price.



Number of Instances: 1 Availability Zone: No Preference

Advanced Instance Options

Here you can choose a specific kernel or RAM disk to use with your instances. You can also choose to enable CloudWatch Detailed Monitoring or enter data that will be available from your instances once they launch.

Kernel ID: Use Default RAM Disk ID: Use Default

Monitoring: □ Enable CloudWatch detailed monitoring for this instance (additional charges will apply)

User Data: jdoe password

● as text

○ as file

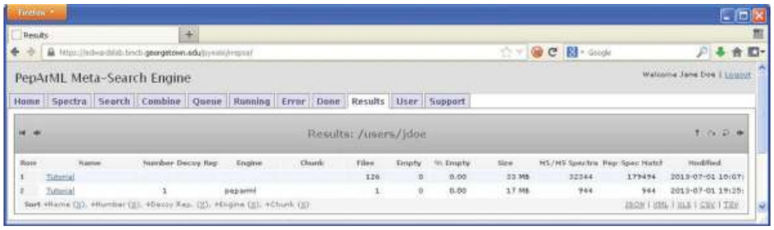
(Use shift+enter to insert a newline)

□ base64 encoded

IAM Role: None

Figure 11.

PepArML username and password in the Amazon spot request User Data field.



PepArML Meta-Search Engine

Welcome Jane Doe | Logout

Results: /users/jdoe

Row	Name	Number Decoy Rep	Engine	Chunk	Files	Empty	% Empty	Size	MS/MS Spectra	Pep Spec Match	Modified
1	Tutorial				126	0	0.00	33 MB	32344	179494	2013-07-01 10:07:
2	Tutorial	1	peparml		1	0	0.00	17 MB	944	944	2013-07-01 19:25:

Sort: +Name (X), +Number (X), +Decoy Rep. (X), +Engine (X), +Chunk (X)

Figure 12.

Completed PepArML analysis for the Tutorial folder.

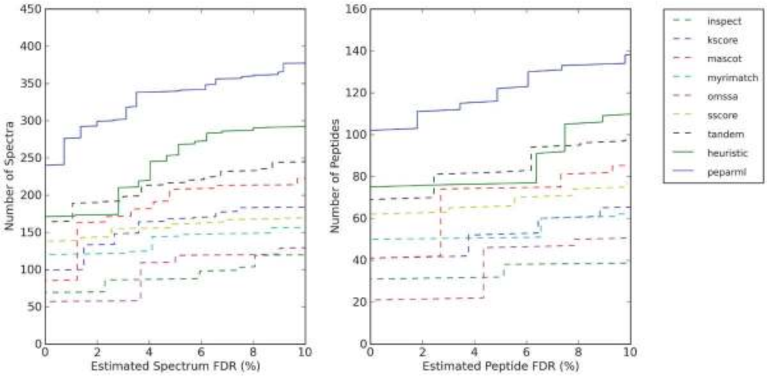


Figure 13.

Evaluation of combiner methods by spectrum and peptide q-values (fdrcurves.png).

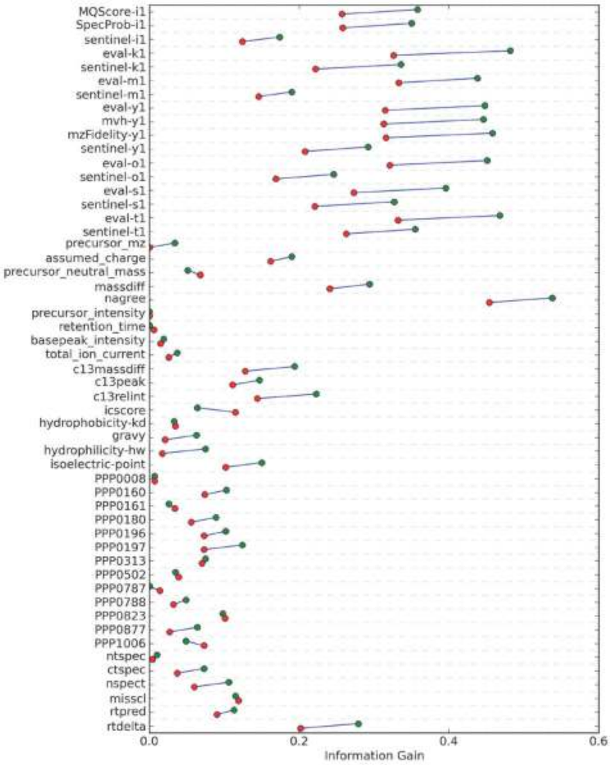


Figure 14.

Information gain of PepArML PSM features for the example anaysis (infogain.png).

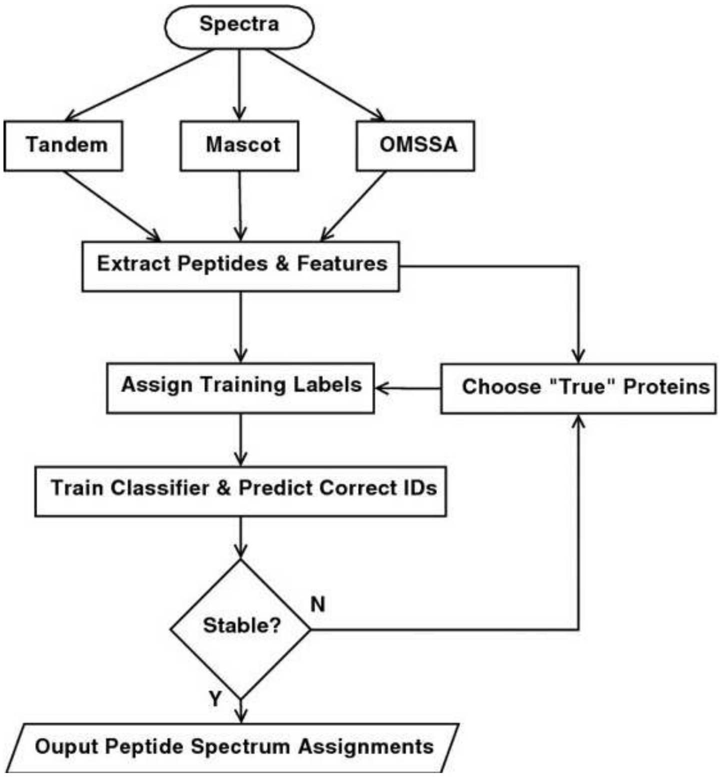

Spectra
Tandem
Mascot
OMSSA
Extract Peptides & Features
Assign Training Labels
Choose "True" Proteins
Train Classifier & Predict Correct IDs
Stable?
N
Y
Ouput Peptide Spectrum Assignments

Figure 15.

Schema for unsupervised PepArML training heuristic (Edwards et al., 2009, used with permission)

Table 1

Table of PepArML redirection keywords – append to http://edwardslab.bmcb.georgetown.edu/ or http://tinyurl.com/.

Page/Data	Keyword
Homepage	PepArML
Batch-Upload	PepArML-Batch-Upload
PSM Features	PepArML-PSM-Features
Example Spectra	PepArML-Example-Spectra
Example Results	PepArML-Example-Results
Support	PepArML-Support

Table 2

Evaluation of combiner methods by identified spectra at various FDR thresholds (stats.csv).

	1% FDR	5% FDR	10% FDR
inspect	69	87	119
kscore	99	168	183
mascot	85	208	222
myrimatch	120	144	156
omssa	57	109	129
sscore	138	161	169
tandem	164	216	244
heuristic	171	253	292
peparml	276	339	377