PROTEOMICS

# Supporting Information for Proteomics

DOI 10.1002/pmic.201000650

Damian Fermin, Venkatesha Basrur, Anastasia K. Yocum and Alexey I. Nesvizhskii

## Abacus: A computational tool for extracting and pre-processing spectral count data for label-free quantitative proteomic analysis

# ABACUS
# Manual & Tutorial

Damian Fermin
dfermin@umich.edu

## Assumptions

This manual covers the usage of Abacus to obtain spectral counts. This manual does not cover using PeptideProphet or ProteinProphet. The assumption is that you know how to use these tools and/or have already generated the necessary files.

The following files are required to use Abacus:

1. PeptideProphet (pepXML) files
2. ProteinProphet (protXML) files
3. Protein database in FASTA format

Optionally, if you would like to report gene identifiers in the output you need a tab-delimited file that maps geneIDs to the proteinIDs found in the FASTA file. If you are using the either the REFSEQ or Uniprot protein databases, tools to download this mapping for you are available at: http://www.nesvilab.org/geneMappers or http://dfermin.com/software. Included in each distribution is the source code for you to edit and view.

## Sample data

You can download a sample data set on which to test Abacus at this link: ######. The sample data is from a MudPIT experiment that analyzed the expressed prostatic secretions of 9 prostate cancer patients [1]. The sample data consists of PeptideProphet and ProteinProphet for just 2 of the patients. The zip file is approximately 22MB in size and contains the following files:

| File | Description |
|---|---|
| interact-prostate_sec_1.pep.xml | PeptideProphet results file, sample #1 |
| interact-prostate_sec_2.pep.xml | PeptideProphet results file, sample #2 |
| interact-prostate_sec_1.prot.xml | ProteinProphet results file, sample #1 |
| interact-prostate_sec_2.prot.xml | ProteinProphet results file, sample #2 |
| interact-COMBINED.prot.xml | ProteinProphet results file, samples #1 and #2 |
| uniprot_Hs_rel2010-09.fa | FASTA file used for search |
| gene2uniprot.txt | Gene-to-Protein Mapping file |
| tutorial_output.xls | Excel file, the expected results of running Abacus |

| | |
|---|---|
| | on this data set using the default parameters. |

# Pre-installation

Abacus requires JAVA so please make sure you have the latest version of the JAVA Runtime Environment (JRE) installed on your computer. You can download it for free at http://www.java.com/en/download/.

# Running Abacus

## For the impatient

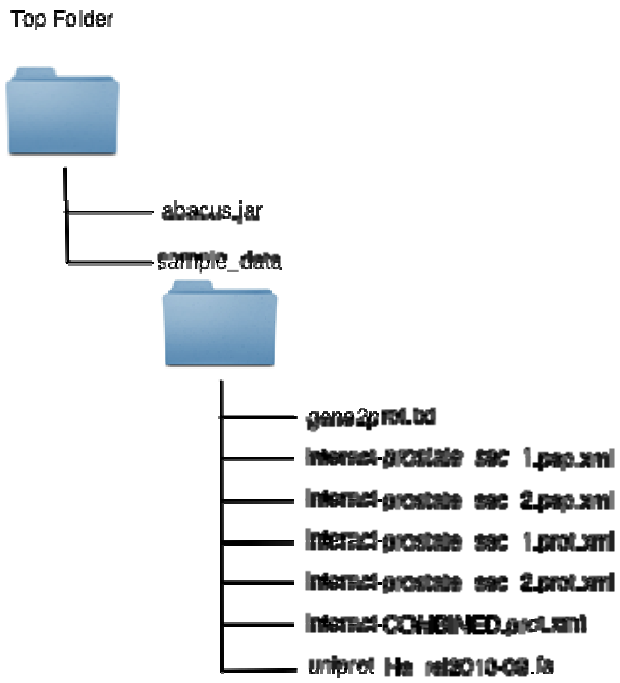
Open a command prompt or terminal in the folder that contains the Abacus.jar file. Double-click on the Abacus icon. However, if you are using Linux, type the following in the terminal:

```
java -Xmx1g -jar abacus.jar
```

This will bring up the main menu. Have fun.

## For the more patient (Tutorial)

This tutorial will assume you are using the sample data set mentioned above. Once you have downloaded and unzipped the archive you should have a new folder called "sample_data". Ideally, you want your folder structure to look something like this for the tutorial to work smoothly:

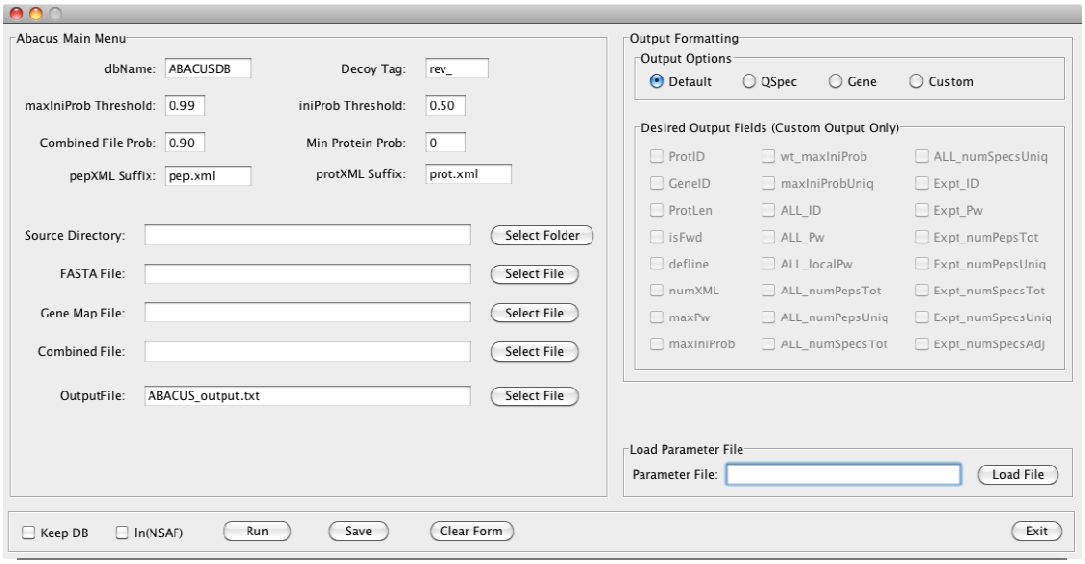



## Run Abacus

From the “Top Folder”, double click on the Abacus icon. If you are using Linux, type the following command at the terminal:

```
java –Xmx1g -jar abacus.jar
```

This starts up the graphical user interface (GUI) component of the program.




Each field in this menu is explained in the next section. Once you have made your selections on this menu you can either:

- Run Abacus
- Save the parameters to a file
- Exit the program

For the sample data set, use the default parameters given by Abacus for now. Then:

1. Select the folder that contains the sample data.
2. Select the FASTA file.
3. Select the gene2prot.txt file.
4. Select the COMBINED file.
5. Select a location to save the output file.

After you have done 1-5 above, click Run. A new window should pop up. This is the Abacus console. It will report what is going on in the program. If an error occurs, it will be printed to this console. You will not be able to close this window until Abacus finishes.

After Abacus has finished running, you will find the ABACUS_output.txt in the output path you specified. You can compare your results to the sample output provided for the tutorial (tutorial_output.xls). Assuming you used the default parameters, the files should be identical.

You can save this test run to a text file by clicking on Save. Abacus parameter files can be loaded using the "Load File" button. Alternatively, you can run Abacus with a particular parameter file from the command line using this command:

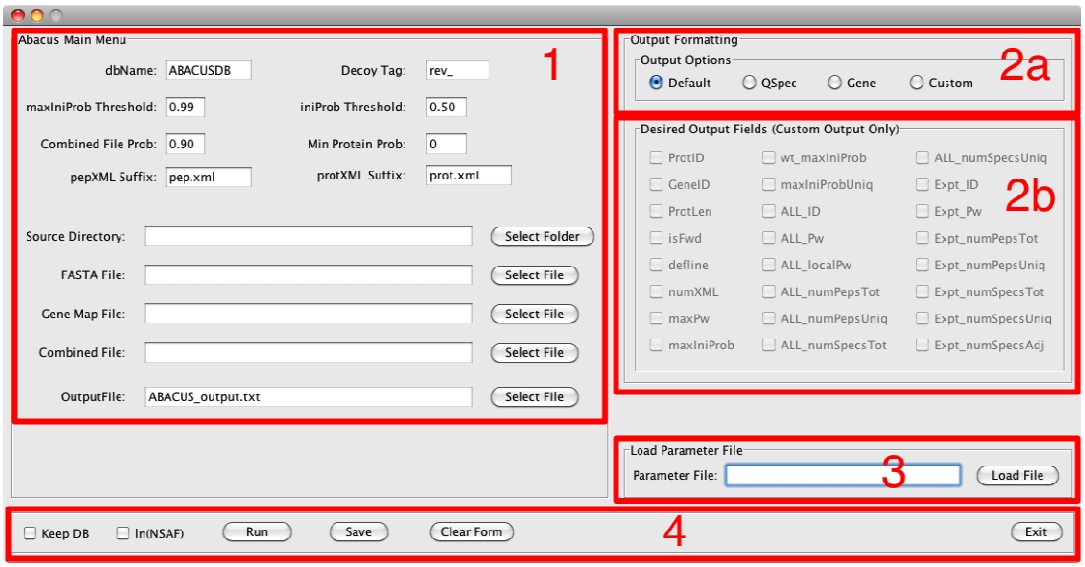
```
java -Xmx1g -jar abacus.jar -p <PARAMETER_FILE_NAME>
```

When you run Abacus from the command line as shown above, the GUI does not appear and the program starts running immediately. All job information is printed to the command line. *In general, it is faster to run Abacus from the command line.*

The parameter file once generated can be edited manually with any text editor (vi, WordPad, etc.). Each field in the file is documented to help you understand what it is and does.

## The GUI Options

Each region of the GUI is explained below. The sections of the screen are broken up and numbered based upon the image below:



## 1. Main Menu

| Main Menu Options | Description |
|---|---|
| dbName | The name to be given to the database: this is trivial and only needs to be changed if you want to save the database for further analysis. (More on this later) |
| Decoy Tag | If your search database contains decoy proteins, include the first few characters of the protein identifiers. If you didn't use decoys, you can ignore this field. |
| maxIniProb Threshold | In order to be considered as a candidate protein by Abacus, a protein must have associated with it at least one peptide with a probability above this threshold in the COMBINED file. |
| iniProb Threshold | Only peptides in the pepXML files with probabilities above this threshold will be used for spectral counting. |
| Combined File Prob. | The minimum probability score a protein needs in the COMBINED file in order to be retained by Abacus as a candidate protein. |
| Min Protein Prob. | The minimum probability a protein needs in the individual files in order to be reported by Abacus as being present in the individual files. |
| pepXML Suffix | The file extension used for pepXML files. |
| protXML Suffix | The file extension used for protXML files. |
| Source Directory | The path to where the XML files are on the computer |
| FASTA File | The path to the FASTA file to be used |
| Gene Map File | The path to the text file that maps gene ID's to protein ID's (optional) |
| Combined File | The path to the COMBINED file. |
| Output File | The path to the output file that Abacus will make. A default is provided based upon the output options. |

## 2. Output Options

The output Options panel is broken up into two parts. Part 'A' is about choosing the output file type. Part 'B' is only relevant if you are generating a custom output file.

### 2a. Output Options

| Output Options | Description |
|---|---|
| Default | The output file produced contains all of the fields normally reported by Abacus. This is the most verbose output format. |
| QSpec | The output file that is generated is formatted for submission to QSpec [2]. |
| Gene | **By selecting this output option, Abacus runs in gene-centric mode.** All of the analysis is carried out at the gene level. The output file mirrors that of the Default output option, but is formatted differently to focus on genes. **This option can only be used if a Gene Map file is provided (see section 1 above).** |
| Custom | The output file contains only the fields selected by the users. The available fields are described in 2b below. |

### 2b. Desired Output Fields (Custom Output Only)

| Field | Description |
|---|---|
| ProtID | The protein identifier. |
| GeneID | The gene identifier. This can only be reported if a Gene Map file is given. |
| ProtLen | The length of the protein in amino acid residues. |
| isFwd | Indicates if a protein or gene is a real or decoy entry. '1' means the entry is real (i.e. "is Forward"); '0' means the entry is a decoy. |
| defline | The description of the protein or gene. |

| | |
|---|---|
| numXML | The number of protXML files in which the protein was identified. |
| maxPw | The maximum ProteinProphet probability observed for the entry among the individual protXML files. |
| maxIniProb | The maximum PeptideProphet probability observed for the entry in the combined file. |
| wt_maxIniProb | For the maxIniProb value observed, this field indicates its degeneracy in the experiment. This is the "weight" (wt) of the peptide as reported in the protXML files. A peptide with a weight of 1 is unique to a single protein. A peptide with a weight of 0.5 is shared between two proteins. |
| maxIniProbUniq | The maxIniProb value observed among the peptides that are *unique* (i.e. wt > 0.9) to the protein. |
| ALL_ID | The protein identifier in the combined file for the protein entry. This is formatted as <NUMBER>-<LETTER> to coincide with the labeling schema of the protXML file. |
| ALL_Pw | The ProteinProphet probability assigned to the protein group in the COMBINED file. |
| ALL_localPw | The ProteinProphet probability assigned to the specific protein identifier reported in the Abacus output. This value is from the COMBINED file. |
| ALL_numPepsTot | The total number of peptides reported in the COMBINED file for the protein. |
| ALL_numPepsUniq | The number of *unique* peptides reported in the COMBINED file for the protein. |
| ALL_numSpecsTot | The total number of spectra reported in the COMBINED file for the protein. |
| ALL_numSpecsUniq | The number of unique spectra reported for the protein in the COMBINED file. Unique spectra are recorded from peptides that are unique to the protein (i.e. wt > 0.9). |
| Expt_ID | The protein identifier for the protein in a particular experiment protXML file. This is formatted as <NUMBER>-<LETTER> to coincide with the labeling schema of the protXML file. |
| Expt_Pw | The ProteinProphet probability assigned to the specific protein identifier reported in the Abacus output within the given experiment's protXML file. |
| Expt_numPepsTot | The total number of spectra reported for the protein in the experiment's protXML file. |
| Expt_numPepsUniq | The number of *unique* peptides reported for the protein in the experiment's protXML file. |
| Expt_numSpecsTot | The total number of spectra reported for the protein in the experiment's protXML file |
| Expt_numSpecsUniq | The number of unique spectra reported for the protein in the experiment's protXML file. |
| Expt_numSpecsAdj | The *adjusted spectral count* for the protein in the experiment's protXML file. |

There are 7 experimental (Expt) fields. If you had N protXML files, there will be $N \times 7$ experimental fields in your Abacus output.

## 3. Load Parameter File

This field allows you to load a parameter file into the GUI. All of the data contained in the parameter file will be used to fill out the various parts of the GUI. Once loaded, you can change the options as you see fit.

### 4. Management Bar

| Field | Description |
|---|---|
| Keep DB | By default, Abacus does not retain the temporary files it uses to generate the final output. By checking this box, the HyperSQL that is created by Abacus for data processing will not be deleted when Abacus is done. |
| ln(NSAF) | This option adds one additional column to the output file for every spectral count fields (total, unique, and adjusted). This is the Normalized Spectral Abundance Factor as described here [3]. This is discussed later on. |
| Run | Run Abacus. |
| Save | Save the current options to a parameter file for later use. |
| Clear form | Restore the Main Menu back to its default settings. |
| Exit | Quit Abacus. |

## Normalized Spectral Abundance Factor (NSAF)

As mentioned above, Abacus can report NSAF values for all 3 spectral count fields. The actual value reported by Abacus is calculated using this method:

$$(NSAF)_k = \left[ \left( \frac{(SpC/L)_k}{\sum_{i=1}^{N} (SpC/L)_i} \right) \right]$$

Optionally, NSAF is log-transformed.

## HyperSQL Database Access

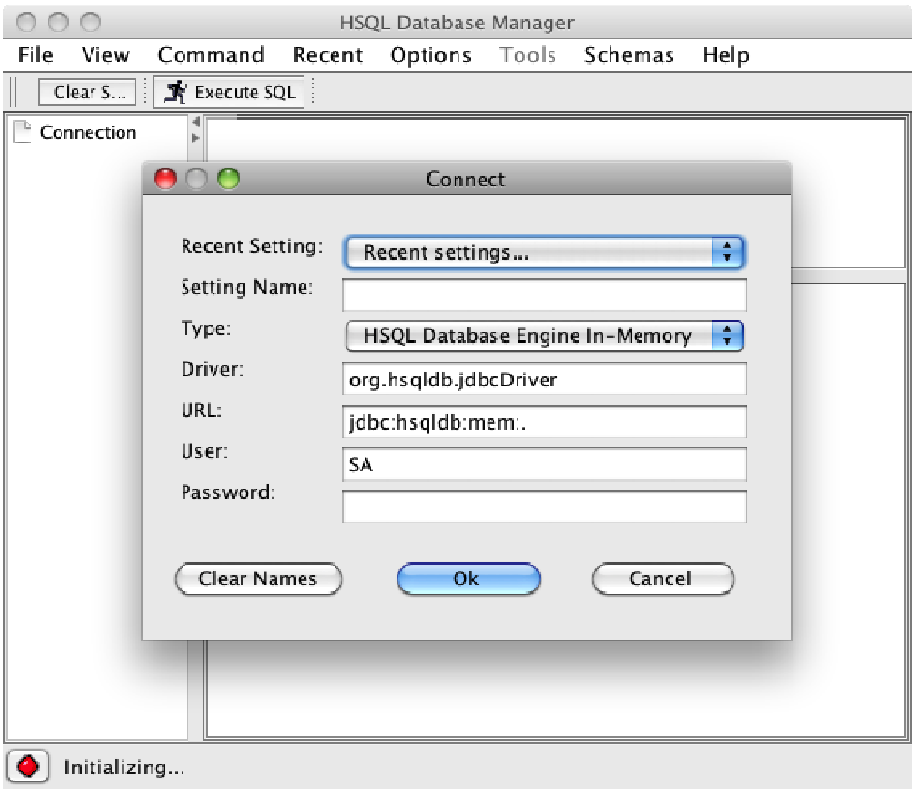
Abacus uses the HyperSQL Database to store and query the information it reports. The user can access the contents of the database directly should the default output be insufficient. To do this, Abacus must be run with the 'Keep DB' option checked. Please note that when keeping the database Abacus takes longer to generate results.

Once Abacus is done, move to the folder where the database files were created by using the command line. This is usually the folder that contains Abacus. The name of the database is whatever you specified in the 'dbName' field.

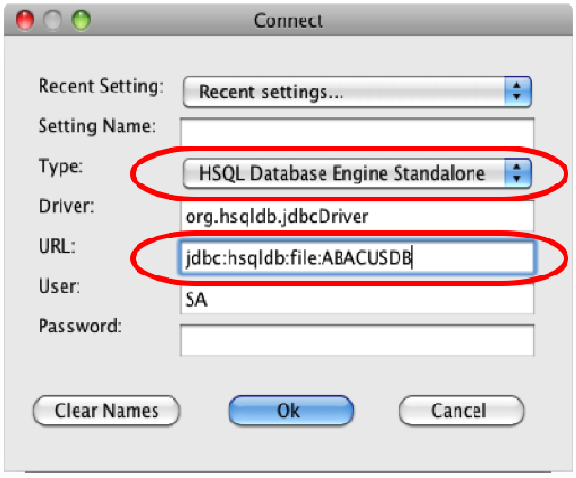
Once in the folder, type the following command from the command line:

```
java -cp ".:./abacus.jar" org.hsqldb.util.DatabaseManagerSwing
```

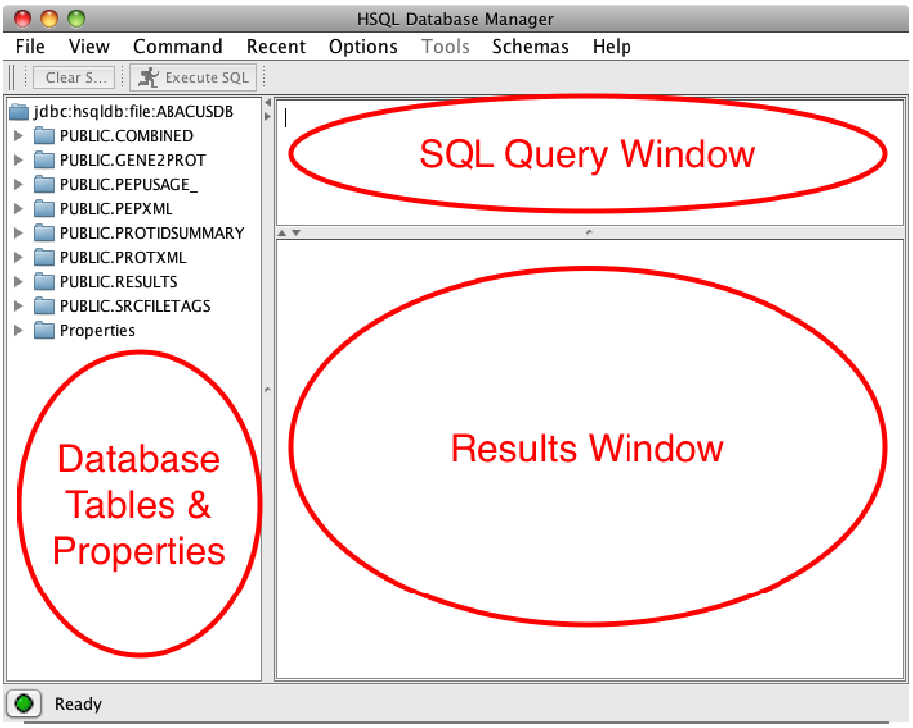
This will produce the following window:




Let's assume you ran through the tutorial above and checked the 'Keep DB' option. To access the database you created as part of the tutorial. Change the Connect menu window to look like this:




Then click on 'Ok'. This should bring you to the main query window that should look something like this:




The various sections of the HyperSQL interface are labeled above. You can obtain a description of each table in the database by clicking on its folder icon in the "Database Tables & Properties" window. If you wish to run your own queries on the database you would type them in SQL in the "SQL Query Window". The results of your query would appear in the "Results Window". For a complete description of the interface, refer to the HyperSQL Database manual available at: http://hsqldb.org/doc/2.0/util-guide/index.html .

The following is a brief description of each of the tables normally generated by Abacus and stored in the HyperSQL database. These tables are specific to the protein-centric output but the table properties are similar for the gene-centric algorithm.

| Table Name | Description |
|---|---|
| COMBINED | Holds the data parsed out of the COMBINED file. |
| GENE2PROT | Holds the mappings between gene IDs and protein IDs. |
| PEPUSAGE_ | Table reports which peptides are mapped to which proteins in each experimental file. It also contains information about the spectral count for each individual peptide, its alpha value. |
| PEPXML | Holds data parsed out of each pepXML file. |
| PROTIDSUMMARY | This table is used to select the representative protein ID from the COMBINED file protein groups. |
| PROTXML | Holds data parsed out of each experimental protXML file. |
| RESULTS | The contents of this table are what Abacus finally outputs . |
| SRCFILETAGS | This file maps pepXML files to their corresponding protXML files. |

## References

[1] Drake, R. R., Elschenbroich, S., Lopez-Perez, O., Kim, Y., *et al.*, In-depth proteomic analyses of direct expressed prostatic secretions. *J Proteome Res* 2010, *9*, 2109-2116.
[2] Choi, H., Fermin, D., Nesvizhskii, A. I., Significance analysis of spectral count data in label-free shotgun proteomics. *Mol Cell Proteomics* 2008, *7*, 2373-2385.
[3] Zhang, Y., Wen, Z., Washburn, M. P., Florens, L., Refinements to label free proteome quantitation: how to deal with peptides shared by multiple proteins. *Anal Chem* 2010, *82*, 2272-2281.