PepQuant2

User Guide

Table of Contents

Introduction

Overview

PepQuant2 is a quantification tool for LC-MS/MS proteomics data. PepQuant2 utilizes peptide identifications from tandem mass spectra search engines in order to interrogate MS1 spectra for theoretical isotopic patterns and quantify peptides/proteins based on their MS1 intensities. PepQuant supports both label-free quantification as well as SILAC labels (lys6, lys8, arg6, and arg10 for MaxQuant output). PepQuant currently supports MS2 identifications in MaxQuant (msms.txt) and pepXML formats with limited mod support (phosphorylation, acetylation, and oxidation) and StatQuest without mod support.

Requirements

PepQuant2 requires a Linux distribution with Libxml2.

Acknowledgments

PepQuant2 is the latest version of a series of proteomics quantification tools going by the name of PepQuant. The original PepQuant was developed as a Java application in the Emili Lab by Jian Liu.

The current implementation incorporates the MINPACK-1 Least Squares Fitting Library. For more information please refer to: http://www.physics.wisc.edu/~craigm/idl/cmpfit.html.

Quick Start Guide

Installation

PepQuant2 is written in C. In order to compile PepQuant simply open a terminal in the main PepQuant directory and run 'make'.

```
cd ./PepQuant2
make
```

This should produce an executable pepquant2.

Running

PepQuant2 comes with default settings for most program parameters. Two parameters that must be specified include the MS2 identifications to use and the FASTA file used for the MS2 searches. PepQuant should be run from the directory containing the mzXML files for the data set.

PepQuant2 currently supports three different input formats with different command line parameters used to specify each one.

Table 1 – PepQuant2 input formats

Input	Command Line Argument	Description
MaxQuant	-Z path_to_msms.txt_file	Running maxQuant produces a directory of different result files. The msms.txt file is usually located in "./combined/txt/msms.txt"
pepXML	-Y path_to_dir_of_pepXML_files	PepXML is a file type supported by many search engines directly or indirectly by conversion. Place all pepXML files in a single directory (only those you want to search) and specify that directory as your argument. (only .pepXML and not .pep.xml or .pepxml will be detected)
StatQuest	-X path_to_StatQuest_directory integer	StatQuest is an in house tool for filtering SEQUEST results. Simply specify the directory containing statQuest results and an integer between 1 and 99 representing the confidence filter used.

The FASTA file is specified using the -F argument followed by the path to the fasta file. Therefore one can run PepQuant2 as follows:

```
path_to_pepQuant/pepquant2 -Z /path/to/msms.txt -F /path/to/fasta_file
```

If pepquant2 is added to the PATH then the path_to_pepQuant can be omitted. While running pepQuant will display what step it is currently on.

Results

PepQuant2 produces a few files after completion. A table of files and a short description of each is included as Appendix 1. Two important files are quant.txt and protQuant.txt. These two files are a table of peptide intensities across all runs and a table of protein intensities across all runs respectively. They are best viewed using a spreadsheet application such as Microsoft Excel or LibreOffice Calc.

In-depth Description

Reading results and isotopic pattern generation

PepQuant2 supports MS2 peptide identifications coming from MaxQuant, StatQuest, and any search engine whose results can be converted to pepXML format. PepQuant does not perform any filtering on MS2 identifications and it is therefore important to apply any and all MS2 filtering steps prior to proceeding to PepQuant2. All of the currently supported input formats identify the data searched i.e. the .raw or .mzXML files used for the MS2 searches. PepQuant will convert .raw to .mzXML but cannot compensate for any other name changes. Therefore it is up to the user to ensure that the names of mzXMLs in the current directory and those reported by the search engine results are the same.

As PepQuant2 reads MS2 identifications it compiles a set of peptides identified. For each of these peptides it determines a theoretical isotopic pattern, keeping track of the top n most intense isotopes, were n is either specified by the user or set to 4 by default. The isotopes reported are actually binned around the most intense isotopes with bin size determined by the ppmCutOff parameter, which is set by the user or 10ppm by default. If PepQuant is unable to generate an isotopic pattern for a given peptide (maybe it has X as an amino acid) it will ignore that peptide in all subsequent steps.

The isotopic pattern generation step supports multi-threading. PepQuant2 will use n threads, were n is user specified or 1 by default. If PepQuant is the only application running feel free to set n to the number of logical processors on the machine.

Searching MS1 spectra for theoretical patterns

PepQuant2 then sequentially opens every mzXML file and searches all MS1 spectra in the mzXML for all theoretical isotopic patterns. PepQuant2 does this for charge states 1+ through 4+ by default, though the upper charge state can be set by the user. As seen in the figure on the left, for every theoretical peak(blue) every observed peak(magenta) falling within ppmCutOff of the theoretical peak is summed to determine the observed isotopic profile. If all isotopic states show at least some intensity, the observed and theoretical profiles are sufficiently correlated (0.99 by default, can be user specified) and the total observed intensities are above a cutoff (1E6 by default, can be user specified) the peptide is recorded along with the correlation and intensities values. At least one charge state must pass the above criteria for the peptide to be recorded. For charge states not meeting this criteria, both correlation and intensity values are recorded as zero.

Since the mzXML files contain information concerning retention time for their respective spectra, PepQuant2 utilizes this moment to record retention time and all other pertinent information.

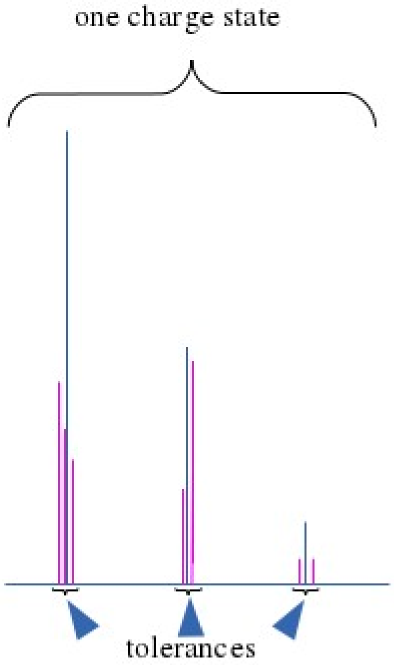


Figure 1: Matching an isotopic profile

Retention time calibration

At this point PepQuant has a series of MS2 identifications and their respective retention times as well as MS1 identifications and their respective retention times. In order to generate an MS2 retention time table, for every (peptide, run) pair we take the retention time to be the weighted centroid of all MS2 identifications for that given pair. For MS1 we use the apical retention time for a given (peptide, run) pair. However, for the apical retention time to be considered valid it requires that at least one other peak is identified within peakWindow (user defined, default 30 seconds) of the apical retention time.

After the tables are created there will undoubtedly be a lot of holes (retention time = 0). To fill these holes we first determine the median MS1 and MS2 retention times for every peptide using the two retention time tables. The median is taken using only positive retention time values. We then align every run to the medians by fitting the linear equation y=-(a)x + b by least squares. Here y would be the median retention time of a peptide and x would be the run specific retention time. Again, we only use positive values for the alignment. We perform the alignment once for the MS1 table and once for the MS2 table.

Once the alignment parameters are determined we can use them to plug holes. We do this by building up our retention time table using most trusted retention times first. With this in mind we start with our MS2 retention time table. For entries with non-zero retention times, replace them with the MS1 retention time only if the two values are within alignWindow (90 by default, can be user specified) seconds of each other. For zero entries, fill them by applying the previously determined run specific alignment parameters to the peptide median. Use the MS1 if the MS1 and MS2 median are within alignWindow of each other, otherwise use the MS2 median. Ideally after plugging holes, a retention time of zero should suggest we do not expect that peptide to be present in that given run.

Quantification

Peptide quantification is then carried out by taking the sum of all MS1 peaks found that fall within quantWindow (default 150 seconds, can be user specified) of the calibrated retention time for a given (peptide, run) pair. After the peptide quantification is complete, the FASTA file specified is read and peptides are mapped to proteins. If a peptide maps to more than one protein it is not unique and is not used for protein quantification. Once peptide to protein mapping is complete the protein quantification results are also printed.

Label-free vs SILAC

PepQuant2 was originally designed for label-free quantification. Support for SILAC labeling has been added. Labeled amino acids need to be specified at run time via the command line arguments (Appendix 2). PepQuant will pool labeled and unlabeled MS2 identifications and use the common pool in determining MS2 retention times for both heavy and light peptides. Separate isotopic profiles will be generated for both labeled and unlabeled peptides and they will be treated as separate peptides in all respects except for the MS2 retention times. In output produced by PepQuant, labeled version of peptides are prefixed with '*' and unlabeled peptides are labeled with the default '_'.

When working with labeled data sets more of the outputted tables become relevant. The protQuant.txt table will show the intensity of proteins derived from labeled and unlabeled peptides (unlabeled peptides may or may not have an associated heavy version). The protHquant.txt will have protein intensities derived from labeled peptides only, protLQuant.txt will be the same except for unlabeled peptides (only peptides with an associated labeled peptide). The protHLratio.txt file contains the values in protHquant.txt divided by their corresponding entries in protLquant.txt.

Labeled data has only been tried for MaxQuant results. Since MS2 identifications are pooled, the spectral counts tables produced will not be accurate for these data sets.

Appendix 1: Files Generated by PepQuant

Table 2 – PepQuant2 output files

Filename	Description
searchResultssMS2.txt	A list of MS2 identifications as reported by the search engine used. The list appears in the following format: peptide sequence data file scan number and retention time
searchResults.txt	A list of MS1 identifications as detected by PepQuant. The list uses a similar format to that used in searchResultssMS2.txt however in addition to scan number and retention time information Four pairs of values seperated by '|' are also included. Each pair is has the form |correlation – intensity|. The four pairs are for different charge states with the leftmost being the highest charge state and the rightmost being for the lowest.
rt.txt	A table of calibrated retention times used for every (peptide, run) pair.
quant.txt	A table of peptide quantification results. Includes an intensity value for every (peptide, run) pair.
protSpectra.txt	A table of protein spectral counts where the values are for the sum of spectral counts of unique peptides only.
protQuant.txt	A table of protein quantification results. Protein quantification results are derived from unique peptides only. Includes an intensity value for every (protein, run) pair. In the case of a labeled data set, the intensities in this file are a sum of all types of intensities.
protLQuant.txt	A table of protein quantification results. Protein quantification results are derived from unique peptides only. Includes an intensity value for every (protein, run) pair. In the case of a labeled data set, the intensities in this file are for light versions of proteins only. In the case of label-free this should be a table of zeroes.
protHQuant.txt	A table of protein quantification results. Protein quantification results are derived from unique peptides only. Includes an intensity value for every (protein, run) pair. In the case of a labeled data set, the intensities in this file are for heavy versions of proteins only. In the case of label-free this should be a table of zeroes.
protHLration/txt	The values of protHQuant.txt divided by their corresponding values in protLQuant.txt.
pepSpectra.txt	A table of ms2 spectral counts as reported by the search engine used for each (peptide, run) pair.
ms2rt.txt	A table of retention times based on the weighted centroid of MS2 identifications.
ms2params.txt	A table of alignment parameters for MS2 retention times as determined by least squares. The columns specify the run, b, and a for the linear equation y = -(a)x + b. Alignment is against the values in median.txt.
ms1rt.txt	A table of retention times based on the apical retention time discovered in MS1 spectra.
ms1params.txt	A table of alignment parameters for ms1 retention times as determined by least squares. The columns specify the run, b, and a for the linear equation y = -(a)x + b. Alignment is against the values in median.txt.
median.txt	A table of MS1 and MS2 median retention times for every peptide. The medians are taken for positive values only in ms1rt.txt and ms2rt.txt.
coverage.txt	A FASTA like file where for every amino acid in a protein sequence the number of times it was mapped to by a unique peptide is shown.
peptides.txt	A table of theoretical isotopic patterns [mz, relative intensity] for every peptide. Table assume no charge. In the case of label-free all peptides will be begin with '_'. In the case of a labeled data set, heavy peptides will start with '*'.

* Most important files are highlighted in yellow.

Appendix 2: Command line arguments

To run supply options to pepquant2, separate arguments with spaces. Both uppercase and lowercase arguments can be used.

Example: `pepquant2 -Z /path/to/msms.txt -F /path/to/FASTA -T 8`

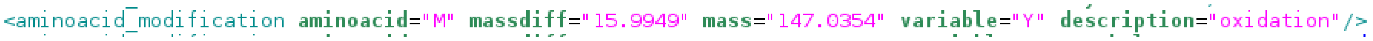
Table 3 – PepQuant2 command line arguments

Argument	Status	Description
-A [integer]	Optional Default = 90	The permissible difference between MS2 and MS1 retention times before defaulting to MS2 over MS1.
-C [float]	Optional Default = 0.99	The correlation cutoff for observed isotopic pattern matching to theoretical isotopic pattern
-E [integer]	Optional Default = 30	The maximum time window within which at least one neighbor peak must be found to declare a MS1 retention time the apical retention time
-F [path_to_file]	Required	The path to a FASTA file used by MS2 search engine.
-I [float]	Optional Default = 1E6	The minimal intensity of an observed isotopic pattern for a given charge state for it to be considered a valid hit.
-K [integer]	Optional Default = 0	The label status of lysine. A value of 6 assumes lysine is made using C13 and 8 assumes lysine is made using both C13 and N15.
-M [integer]	Optional Default = 4	The maximum charge to be looked at by PepQuant2 when looking through MS1 spectra from isotopic patterns.
-O	Optional	Specifies whether to turn off modification site localization. More details can be found in appendix 4.
-P [float]	Optional Default = 0.000010	The PPM tolerance divided by 1E6. Used for generation of theoretical isotopic patterns and matching these patterns to observed patterns.
-Q [integer]	Optional Default = 150	The time window around the calibrated retention time used for quantification.
-R [integer]	Optional Default = 0	The label status of arginine. A value of 6 assumes arginine is made using C13 and 10 assumes arginine is made using both C13 and N15.
-S [integer]	Optional Default = 4	The number of isotopic states used in the theoretical isotopic pattern.
-T [integer]	Optional Default = 1	The number of threads to use during isotopic pattern generation and MS1 spectra interrogation.
-X [path_to_dir] int	One and only one is required	Path to a directory of StatQuest reslts and correlation cutoff used.
-Y [path_to_dir]		Path to a directory of pepXML files.
-Z [path_to_file]		Path to MaxQuant msms.txt file.

Appendix 3: Known Limitations

The following is a list of known limitations, given in no meaningful order

- mzXML files must have uncompressed peak lists and use 32 bit precision, and network byte order.
- Cysteine is assumed to be always carbidomethylated.
- Spectral counts for labeled data sets are not accurate since MS2 identifications for labeled and unlabeled versions of the same peptide are pooled and shared between both versions of the same peptide.
- Mod support is limited to acetylation, oxidation, and phosphorylation for MaxQuant
- Mod support for pepXML files is limited to variable mods and depends on their being a description attribute for the mod. Some search engines only produce information concerning the mod mass, in such cases the user may have to add the description attribute manually. An example of a valid aminoacid_modification node with required description is shown below.



```
<aminoacid_modification aminoacid="M" massdiff="15.9949" mass="147.0354" variable="Y" description="oxidation"/>
```

- There is no ModSupport for StatQuest. It should be possible to replace SEQUEST mod symbols with MaxQuant like mod markers: (ac), (ph), and (ox) in the StatQuest files to get it working.
- MaxQuant is the only supported MS2 source for labeled data.
- Setting cutoffs too low can result in a large number of hits that can consume large amounts of memory.

Appendix 4: the -O argument

Normally when PepQuant2 reads in the results from a search engine it treats peptides with the same sequence but different modification site localization as distinct peptides. For example, the peptides KM(ox)AVILMAR and KMAVILM(ox)AR are different, as are KM(ox)SWIT(ph)CSAR and KM(ox)SWITCS(ph)AR. Peptides with the same sequence, same type and number of modifications but different modification sites are indistinguishable in MS1 scans. Furthermore, such peptides will have very similar, if not identical, retention time. This means that the same MS1 intensities will be assigned to all isomers. In order to prevent this from happening PepQuant can treat all peptides differing in only modification site location as the same peptide. This option is turn on with the -o argument (ohh not zero).

In order to ignore different modification sites, PepQuant2 pushes all modifications onto the leftmost unmodified, equivalent amino acid. For example:

KM(ox)AVILMAR would remain unchanged, (ox) already on leftmost M

KMAVILM(ox)AR would become KM(ox)AVILMAR. Note that the two peptides (KM(ox)AVILMAR, KMAVILM(ox)AR) would now be treated as one.

Another example:

KM(ox)SWIT(ph)CSAR would become KM(ox)S(ph)WITCSAR. The (ox) is already on the leftmost M and the (ph) moves to the leftmost S. Normally an amino acid needs to be the same in order to be considered equivalent, Threonine (T) and Serine (S) are exceptions.

KM(ox)SWITCS(ph)AR would become KM(ox)S(ph)WITCSAR. The (ox) is already on the leftmost M and the (ph) moves to the leftmost S. Note that the two peptides (KM(ox)SWIT(ph)CSAR, KM(ox)SWITCS(ph)AR) would now be treated as one.

Peptides that differ in the number of modifcations are still treated as distinct peptides, e.g. KM(ox)SWITCS(ph)AR and KM(ox)SWIT(ph)CS(ph)AR are not the same even after adjustment.