Technical Report: Automated Analysis of High-throughput Mass Spectrometry Quantification Data using the `HiQuAnT`

Kenneth Bryan

February 25, 2016

Contents

1 Introduction:

HiQuAnT automates the post-quantification analysis of Mass Spectrometry (MS) generated proteomics data especially those involving additional experimental variables (e.g. labels, replicates, time points and treatments). HiQuAnT eliminates the laborious, manual processing of large protein quantification datasets typical of spreadsheet-based applications and reduces the potential for data handling variation and human error. HiQuAnT imports a plain text quantification file (e.g. MaxQuant proteinGroups.txt) and supports the user in defining and implementing a workflow, encompassing data filtering, replicate grouping, label reversal, data transformation and significance testing. Once parameters HiQuAnT have been configured they may be saved as a config file for later use or may be used to support a one touch command line mode. HiQuAnT also supports preliminary identification of outlying replicates and subsequent interactive result visualization. With HiQuAnT the user can rapidly optimize workflow parameters and generally gain a greater level of control over the analysis of large protein quantification datasets. HiQuAnT results may be visualized directly via an interactive heatmap that can be exported to high quality PDF. Results can also be exported to Cytoscape (.xgmml) or Gephi (.gexf) graph formats supporting large datasets that can be interpreted in this manner (e.g. interactome studies).

2 Step-by-Step Guide to HiQuAnT

2.1 Downloading HiQuAnT

HiQuAnT software and several test datasets can be downloaded from the supporting website (http://hiquant.primesdb.eu/). Video tutorials have also been posted on this page demonstrating some of the aspect of HiQuAnT. HiQuAnT can be downloaded as Mac OS application and Windows executable. These application wrap the Java JAR file and still require that the Java Runtime Environment (JRE) is installed on the host operating system. To check if Java is installed, open the terminal (MacOS, Linux) or the command line (Windows) and type 'java -version'. If the Java Runtime Environment is installed then information akin to that shown below should be seen. If this command is not recognised then the JRE needs to be installed from: http://www.oracle.com/technetwork/java/javase/downloads/.

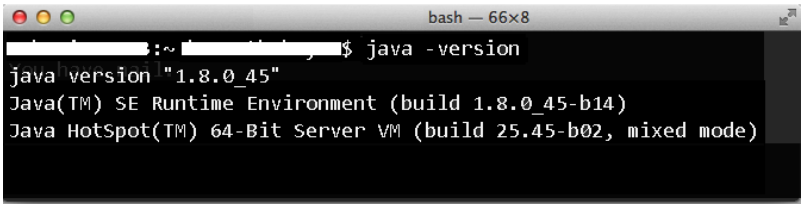


Figure 1: The Java Runtime Environment is required to run HiQuAnT. To check if this is installed type 'java -version' on the command line.

2.2 Running HiQuAnT

The MacOS and Windows downloads contain a simple system specific application file (executed via the mouse pointer with a *double click* or *right click*>*open*) to run the HiQuAnT application. A third option is to download the Runnable JAR file that will run on any system with Java (Java

Runtime Environment (JRE)) installed. This version also gives added options for advanced users, such as setting JVM parameters (e.g. assigning more RAM for larger datasets) and also allows access to the Command Line version of `HiQuAnT` (see Section 2.10) This can be run via the command line (Windows) or terminal (MacOS, Linux) with the following command:

```
(1) "java -jar hiquant.jar"
```

Larger inout datasets may require more memory. which can be added using the "-Xmx" parmeter, for example.

```
(2) "java -jar -Xmx1000m hiquant.jar" (more RAM)
```

Finally this is also the way to run the 'Command Line' version of `HiQuAnT`

```
(3) "java -jar hiquant.jar myfile.config"
(4) "java -jar hiquant.jar myfile.config -s" (silent mode)
```

See section 2.10, for further details.

2.3 Overview of `HiQuAnT's` Graphical User Interface

When the `HiQuAnT` application is run a graphical user interface (GUI) will appear on screen, containing a panel or 'tab' for each of the three main steps (i.e. *Input*, *Set Parameters* and *Analysis*) to guide the user through the selection of pipeline parameter values. Example values are provided for most parameters in labelled (e.g. SILAC) and label-free (LFQ) settings. Every parameter also has a 'popup' explanation, accessed by hovering over the parameter name or via the help menu.

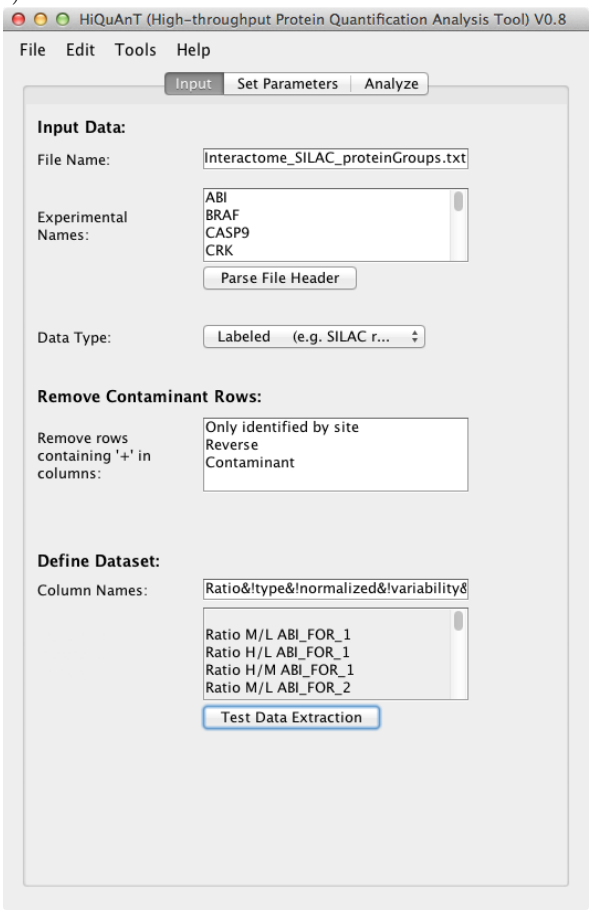
Parameter settings can be also tested prior to full pipeline execution, see 'Test Data Extraction' in section 2.4 and 'Validate Samples' in section 2.5.5. Once the analysis is complete a report containing an overview of results and output files appears on the screen, see Figure 6(b). Result may also be visualised directly via an in-built heatmap option, see section 2.6.3 or , where appropriatem, exported as a experiment vs protein graph file compatible with *Cytoscape* or *Gephi* network analysis applications, see section 2.6.4. Once the parameter settings for a particular dataset have been tried and tested they can be saved as a configuration file, see section 2.9, and re-used again via the GUI mode or with the command line version, see section 2.10.

2.4 The Input tab

2.4.1 Input File:

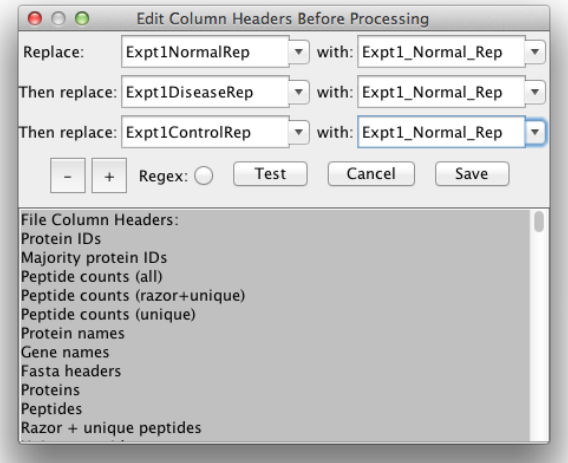
MaxQuant Input File
`HiQuAnT` input format is compatible with a multiple experiment '*proteinGroups.txt*' file output from *MaxQuant* in which sets of data columns represent each experiment. Each set of experimental columns may in turn contain multiple experimental conditions (control, time point, growth conditions) and biological or technical replicates. To allow automation of analysis for such potentially large data files a consistent column heading format must be observed (in addition to the initial MaxQuant terms in the data column header), specifically *Expt1_Cond1_Replicate1* where the separator character may be '_', or ' '(white space). These separators consequently cannot be contained within any of the column descriptor strings/words (i.e.*Expt1*, *Cond1* or *Replicate1*). For *MaxQuant* format `HiQuAnT` reads the column names by parsing the strings after the standard 'Peptides ' string in the Peptides columns at the start of the '*proteinGroups.txt*' file. Maintaining

(a)



Parameter	Value
File Name:	Data file with one or more experiments to be analyzed (e.g. 'proteinGroups.txt'). Enter full path or use 'File > Open...'
Experimental Names:	Names of experiments in the file to be analysed. This may be automated for *MaxQuant* output if correct format is used, see section 2.4.1. Limited column header formatting is provided via the via *Edit> Edit Column Names...* menu option, see panel (b) in this Figure, newline Note: one value per line.
Data Type:	Select SILAC (ratio data) or LFQ (abundance data). Selecting LFQ activates a pop-up, see panel (c) in this Figure, that allows the user to specify which data groups are to be compared (e.g. cond1/ctrl, cond1/cond1,.. etc.) and updates the 'Set Parameters' tab .
Remove Contaminant Rows:	List of column names in which to check for '+' character and remove corresponding flagged rows. Note: one value per line.
Define Dataset:	Defines the exact columns to be selected as the dataset (for each of the experiments listed in 'Experimental Names:') by specifying strings/words that must and must not be present (preceded by a '+' and '-' respectively) in the column name. Note: values must be separated by semi-colon (';') e.g. '`+Ratio;-normalized`'.
Test Data Extraction:	Once above parameters are selected the 'Test Data Extraction' button can be used to test the parsing by printing the names of the selected columns.

(b)



(c)

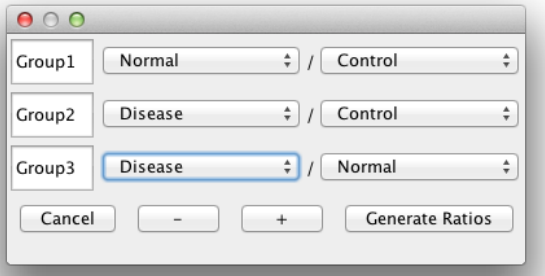


Figure 2: Step 1 - `HiQuAnT` Input Tab

this column heading format allows HiQuAnT to 'understand' the meaning of each data columns and consequently perform processing operations across related columns without the need for tedious manual point-and-click annotation, typical of post-quantification analysis applications, such as *Perseus*. HiQuAnT may handle 'ratio' data derived from labelled SILAC analysis or label free (LFQ) abundance values.

Generic Input File
HiQuAnT is also compatible with the importation of a generic text file, containing labelled or label-free quantification data, that has a basic row/column format and contains column heading labels that conform to the specifications listed above. There must also be at least one row label column. In this case, as the format is unknown, the 'Parse File Header' function in the HiQuAnT GUI will parse the first strings across all column header names, remove duplicates and populate the 'Experimental Names:' text area. The second string/word in the column header will also be used to inform the ratio generation popup that appears when Unlabelled/ LFQ is selected in the 'Data Type:' option, Figure2 (c). The MS quantification file to be analysed (e.g. 'proteinGroups.txt') must first be selected by choosing *File> Open...* and navigating to the file locations. For manual entry the full path (or path relative to the location of the HiQuAnT jar file) must be typed. This file path is also stored when the the configuration file is generated, see section 2.9.

PSI mzTab Format:
HiQuAnT also has a feature to convert from the PSI mzTab format. This is accessible via *Tools>Convert mzTab Files...* When this option is chosen the user must navigate to the folder containing the mzTab files and select this folder (not individual files). HiQuAnT will then convert any file ending with '.mztab' to a generic text file.

Merge Input Files:
Results from experiments may be fragmented i.e. contained across multiple results files (e.g. several proteinGroups.txt files) this may be due experiments being carried out at different times or a lack of computational resources (e.g. HPC Cluster) to process raw MS files in parallel. HiQuAnT also has a feature to merge several inout files in to a single inout file so that they may be processed together in HiQuAnT. This is accessible via *Tools>Merge multiple quantification files...* Again the user must navigate to the filer containing the files to be merged. All files ending in ".txt" in this folder will be merged into a single file. The user must choose the column containing the row label (Merge Key) which will be used to cross reference and merged the rows in each file, see Figure 3.

Edit Column Header Labels:
Simple editing (find and replace) of column header labels is also provided by HiQuAnT via *Edit> Edit Column Names...*, see Figure 2(b). This allows, for example, accepted separators (see section 2.4.1), to be inserted around specified strings/words in the column headers to allow conversion to HiQuAnT compliant format. Upon saving the editing column names a new input file is generated (e.g. 'myfile_Edited.txt') within the same working directory and the value of 'File Name:' parameter field is updated. This feature only allows insertion of separators and does not allow for reordering of column heading strings, which must be ordered as outlined in section 2.4.1.

2.4.2 Experiment Names:

The experiments to be analysed in the input file must then be entered in the 'Experimental Names:' text area. This may be done manually (simply enter the first string/word of the

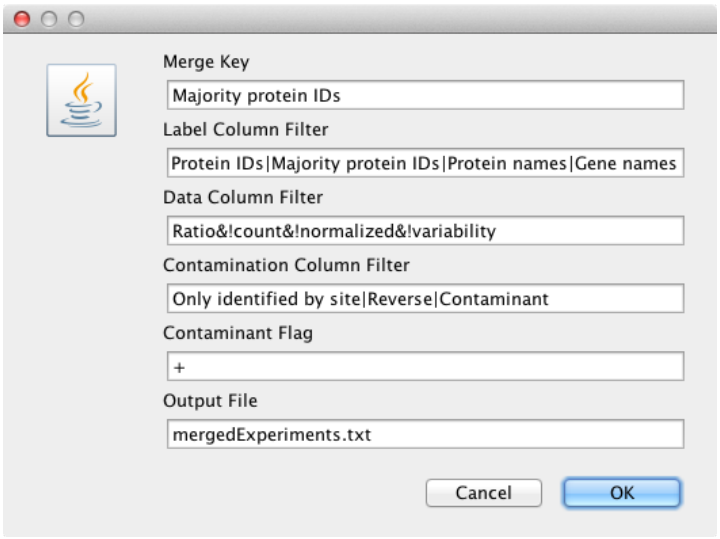


Figure 3: The options for marging multiple protein quantification files in to a single input that can be processed by `HiQuAnT`.

column header name(s) that represent the experiment name) or may be automated for large multi-experiment quantification files, that may contain 10's of 100's of experiments (e.g. high-throughput bait interaction studies where each bait is an experiment). If the input file is a *MaxQuant* 'proteinGroups.txt' file, and column names conform to the format outlined in section 2.4.1 the 'Parse File Header' button can be used to automate this process. Note:In the case where Label-Free data is being analyzed that males use of a single control experiment, the name of the control experiement(e.g. 'Ctrl') must also be entered here. See Figure 2(a).

2.4.3 Data Type:

The 'Data Type:' is set to SILAC by default. On selection of the LFQ format a pop-up appears to allow the user to define which condition group will be compared compared, see Figure2(c). This feature automatically extracts the second string/word in the column header name (again this also relies on the providing the correct column header format outline in section 2.4.1). The default values for other parameters are also updated depending on this selection.

Quantification Data: Labelled quantification data, such as those derived by the SILAC (Stable Isotope Labelling by Amino Acids in Cell Culture) protocol, may have two or three different labelling isotopes (e.g. High [H], Medium [M], Low [L]) that may be used to distinguish different experimental growth conditions. This also allows improved accuracy as samples can be run together (e.g. disease/normal) in the same MS run (and subsequently deconvoluted by quantification software applications such as MaxQuant) to avoid systematic error between runs. MaxQuant outputs a '*proteinGroups.txt*' text file that contains pre-computed relative isotope abundance measurements or 'ratio' data in data columns. Data columns header names will also have the relevant isotope labels (e.g. H/L or M/L). These strings may be used to partially or fully define which columns need to be inverted (due to reverse labelling) or the columns that are part of the same experimental condition/grouping, see subsection 2.5.4 and Figure 2.5.

Label Free Quantification Data: Label Free Quantification (LFQ) results, such as those exported from MaxQuant as a '*proteinGroups.txt*' text file, contain protein abundance measurements (as opposed to pre-computed abundance ratios for labelled data) for each experimental condition. Upon selecting 'LFQ' in the 'Data Type:' option in the 'Input tab', see Figure2 (a), the user is then prompted to define the experimental conditions/sub-groups, see Figure2 (c), that should be compared to address the particular research question, this in turn instructs `HiQuAnT` how to generate the 'ratio' data. `HiQuAnT` also populates the 'Define Experimental Groups: -columns ' parameter setting in the 'Set Parameters' tab based on this information. If there is only one control file for multiple experiment files (as is often the case) the user must select option ending with '(One Control Experiment)'. Note: For this option to be the control experiment must be entered in the 'Experiment Names:' field.

2.4.4 Remove Contaminant Rows:

The 'Remove Contaminant Rows:' parameter lists the columns under which to search for the the '+' flag (standard MaxQuant flag). If this text area is left blank or if these column labels do not exist in the file then no action is taken.

2.4.5 Define Dataset and Test Data Extraction

The 'Define Dataset: -Column Names:' parameter enables precise definition of the data column to extract from the input file for analysis. The user can list arbitrary strings/words that *must* be present (pre-fix with a '&' and interpret as logical 'AND') and strings/words that *must not* be present (prefix with a '&!' and interpret as logical 'AND NOT') in the headers of the data columns.

Ratio M/L MEK1_Cond1_2	Ratio M/L normalized MEK1_Cond1_2	Ratio M/L variability [%] MEK1_Cond1_2	Ratio M/L count MEK1_Cond1_2
0.84447	0.92702	9.6244	11
NaN	NaN	NaN	1
0.69816	0.8946	8.1652	2
NaN	NaN	NaN	1
0.60368	0.77748	19.828	6
0.71761	0.94976	14.344	7
1.0001	1.1321	4.265	4
0.89809	1.178	33.414	4
0.91097	1.1077	5.6969	2
0.79462	0.88733	36.874	5
1.1334	1.1578	0.62504	2
0.72564	0.91964	2.1656	3
NaN	NaN	NaN	1
NaN	NaN	NaN	0
NaN	NaN	NaN	0
0.50444	0.60151	13.182	6
NaN	NaN	NaN	1
0.90811	0.99493	14.347	7
0.46808	0.6097	20.575	11
0.39875	0.56052	26.008	6
0.53119	0.62271	60.969	5
0.45672	0.56358	25.041	8
0.46592	0.53464	22.066	16
NaN	NaN	NaN	0
0.80628	0.91618	30.075	8

Figure 4: `HiQuAnT` allows precise definition of the input file data columns to processed via the 'Define Dataset > Column Names:' parameter, see section 2.5.4.

For example, a *MaxQuant* 'proteinGroups.txt' file may contain *'Ratio...', 'Ratio ... normalised,' 'Ratio ...variability[%]', 'Ratio ...count data', 'Ratio ...type'* columns, see Figure 4. To select only the 'Ratio...' column to analyze simply enter : `'&Ratio&!normalised&!variability&!count'`. (Note: Upper and lower case is respected so '&!count' will not match with '&!Count' for example). This list may need to be edited for different versions of *MaxQuant 'proteinGroups.txt' files* (note also that the case of some letters in the input file column names may also change) or replaced with a custom list when a generic text input file imported provided the column header names comply format specified in section 2.4.1. `HiQuAnT` allows this step to be tested within the GUI before proceeding by clicking the 'Test Data Extraction' button. This prints the header names

of the data columns that will be extracted, as well ask the the total experiment count and total column count, in the grey output panel below, see Figure 2(a). Do not proceed until this output matches exactly the columns that you intend to analyse.

2.5 The Set Parameters tab

Once the data rows and columns to be analysed are selected, transformation and data column grouping options can be set in the 'Set Parameters tab'. Example values are provided for both SILAC and LFQ pipelines in the *MaxQuant* 'proteinGroups.txt' data format. These can also be used as a guide for the generic text input file format.

2.5.1 Zero replacement and Column Normalization

In each case the resultant ratio values are log_2 transformed prior to further statistical analysis and visualization. The first parameter 'Zero replacement' sets the value should be used to replace '0' to avoid the $log_2(0)$ error. The 'Normalization' parameter sets the type of data column standardization prior to log_2 transformation, for example, it is typical to divide ratio values by the median column value to allow cross run (cross column) comparisons.

2.5.2 Label Reversals

Labelled quantification data (e.g. SILAC) may contain with label reversals to reduce label bias, therefore the 'ratio' values in these data columns must inverted. before analysis. As with to with the 'Input tab' these columns can be generically defined by strings/words entered in the 'Columns to Invert' parameter. This allows the automation of this process and is essential to enable efficient analysis for high-throughput quantification datasets. For example, to invert all columns whose column name contains 'High/Low' AND 'Forward' enter '`High/Low;Forward`'. Additional definitions (equivalent of logical 'OR') must be placed on a new line, see Figure 5.

2.5.3 The 'Minimum Replicates:' threshold and 'Replicate Grouping' method

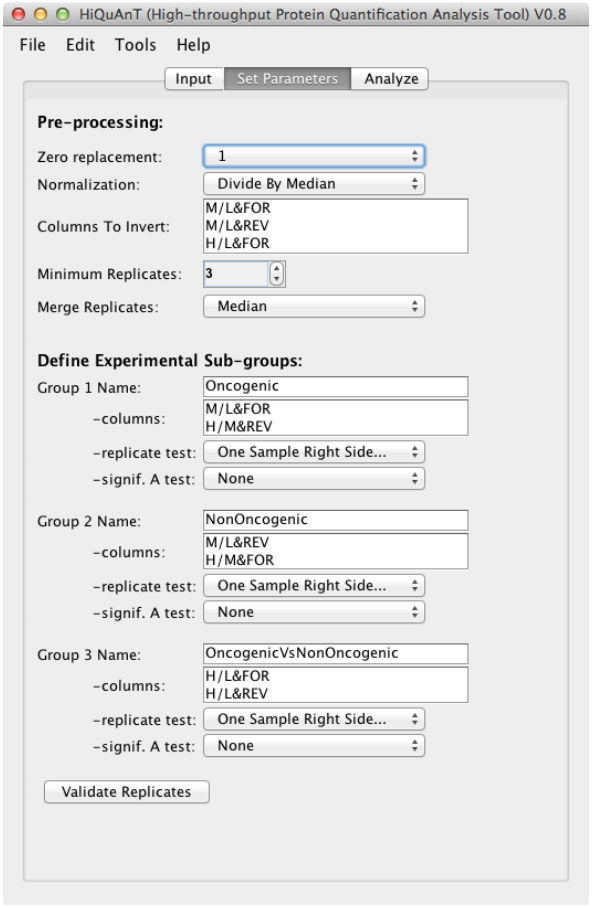
The 'Minimum Replicates:' parameter specifies the minimum number of values that a row/peptide group must have within a sample group in order to be valid. If this number falls below this threshold then 'NaN' will replace this values when the sample group replicates means/medians are calculated. 'Replicate Grouping' defines how to consolidate replicates for a sample group, currently 'Mean' and 'Median' grouping can be applied.

2.5.4 Defining Experimental Groups and Statistical Tests

`HiQuAnT` allows generic definition of experimental sub-groups (e.g. normal, disease, control, ...) using strings/words that appear in the column header names for the sub-group. This enables automated grouping data columns, necessary for efficient processing high-throughput quantification datasets. The 'Group Name', corresponding column definition and particular statistical tests required can be defined for between 1- 3 experimental sub-groups (e.g. normal/control, disease/control, normal/disease).

For labelled datasets the 'ratio' data columns are grouped and statistically evaluated via a 'One Sample T-test' (right/left /two sided) or a Significance A test (Q-function, Outlier test). For label-free data the '-columns' definition automatically set after the 'LFQ' value is selected in the 'Data Type:' and the user defines the data column to be compared. In this case, a Two sample T-test (right/left/two sided) is carried it on the abundance data and a Significance A test

(a)



Parameter	Value
Zero Replacement:	Value to replace '0' with when log transforming data.
Normalization:	Data column standardisation method, to enable comparison of data columns (e.g. samples under different experimental conditions).
Columns To Invert:	Which columns ratios to invert (to, for example, undo label reversals for SILAC)
Minimum Replicates:	The minimum number of replicates (for a particular sample) needed to validate measurements.
Replicates Grouping:	The method to consolidate replicate values into a single representative value for each sample group.
Define Experimental Groups:	Set Name and define columns that represent each sample group such as 'Disease/Control' , 'Normal/Control' etc. Specific statistical tests can also be selected for each group. To generate ratios and autofill these parameters the 'LFQ' option must first be selected in the Input tab 'Data Type' parameter.
Validate Samples:	Click this button to generate interactive box plots of all replicates in each group for each experiment, see panel (b). Any outlying replicates may also be selected and removed at this stage, see selected replicate (red) in panel (b).

(b)

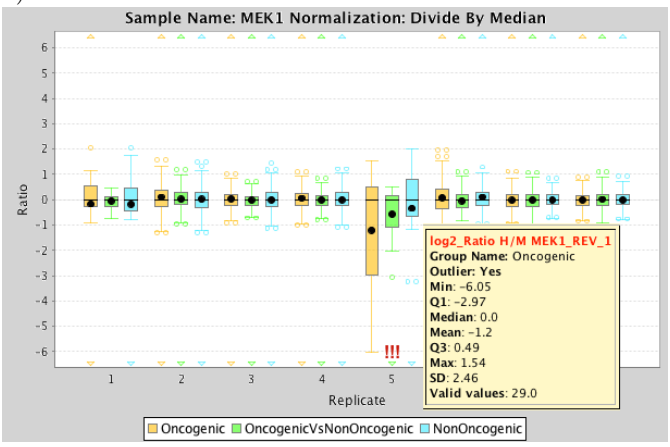


Figure 5: Step 2 - `HiQuAnT` Set Parameters Tab (b) Replicates can be visualised via the interactive boxplot feature and to aid identification and removal of outlying replicates.

can be performed after 'ratio' generation (see section), log_2 transformation and group averaging. (Note: the $log2$ transformation is always applied by where statistics are calculated on 'ratio' data (e.g. One Sample T-test of labelled data or Significance A test on Label-Free data ratios.))

2.5.5 Visualization and Validation of Replicates

When parameters have been selected the data may be visually validated via interactive box plots. In the case of labelled data a box plot is constructed showing the log_2 transformed ratios for each replicate for each group. Each experiment is shown in a separate box plot. For label-free data the abundance values for each replicate are plotted. Box plot visualisation aids identification of outlying replicates. HiQuant suggests possible outliers by flagging samples (with three red exclamation marks - '!!!') whose mean lies greater than 2 standard deviation from the grand mean (mean of all replicates). Replicates can be selected and excluded from the analysis by right clicking on the corresponding replicates(s) and clicking the 'Exclude Selected Replicates' button. Box plots can be exported to publication quality PDF format by clicking the 'Export To PDF' button for use within supplementary material.

2.6 The Analyze tab

2.6.1 Set Output Parameters

The analysis pipeline is executed on the 'Analyze' tab, see Figure 6. Here the fold change and statistical cut-off thresholds can be set and the row label the columns to append to the output file can be selected. Additional value columns can also be appended such as the log_{10}(sum of the Intensity values) for each row. The sum of the intensity values is often used to estimate abundance levels for labelled data. Lastly, a label for the output file name can also be set.

2.6.2 Set Output Files

Once the analysis pipeline has completed a report window opens outlining the summary of the analysis results, see Figure 6(b). These result are the list of differentially expressed/abundant proteins for all experiments that were processed from the the input. The result from each experiment is output to a separate ".txt" output file (e.g. one output file for each bait-prey experiment) `HiQuAnT` outputs one or all of he following 4 result files, depending on the options selected on the 'Analyze Tab', a Log file and Report File (plain text form of the on screen report):

Pre-processed replicates: All replicates, transformed & standardized
- e.g. 'Result_Expt1_allReps_SILAC_142915030txt', see Figure 7(a)

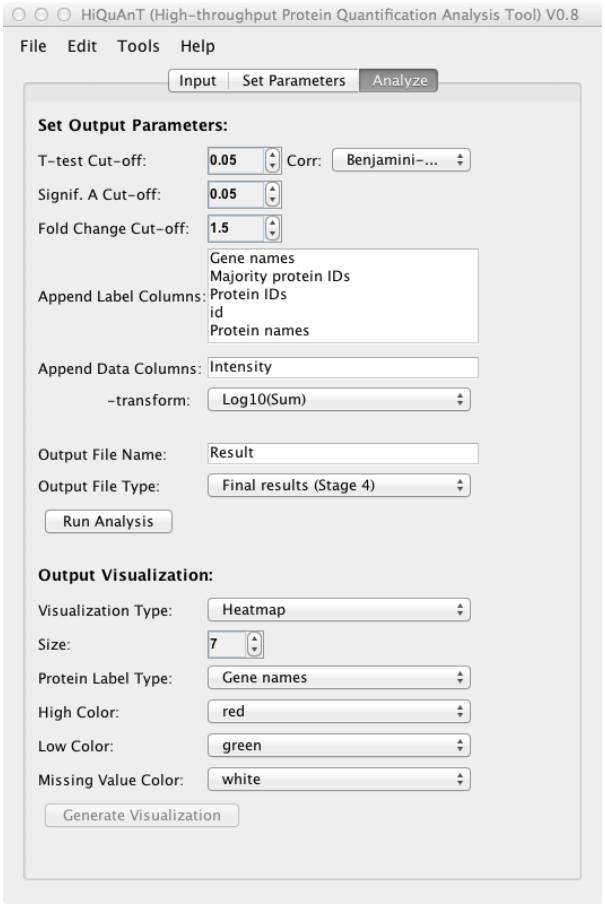
Minimal replicates: All replicates, transformed & standardized passing the replicate threshold
- e.g. 'Result_Expt1_min_3_reps_SILAC_14291503.txt', see Figure 7(b)

Replicate medians: Experimental sub-group (e.g. Normal, Disease, Control) Median values
- e.g. 'Result_Expt1_Median_min_3_reps_SILAC_14291503.txt', see Figure 7(c)

Final Results: Group Median values that pass the statistical and fold change thresholds
- e.g. 'Result_Expt1_Median_min3Reps_0-05_0-05_1-0_SILAC_14291503.txt', see Figure 7(d)

Report File: Table summary of all significantly changing peptide groups for each experiment, see Figure 6(b)

(a)



Parameter	Value
T-test Cut-off:	Minimum T-test significance value in Output File Stage 4, select '1' to include all, see section 2.6.1 & Figure 7(d).
Corr:	Apply a 'Correction for Multiple Comparisons' (e.g. Benjamini-Hochberg).
Signif. A Cut-off:	Minimum Significance A value to be taken in Output File 4, select '1' to include all, see section 2.6.1 & Figure 7(d).
Fold Change Cut-off:	Minimum Fold Change ($log_2(ratio)$) value to be taken in Output File 4, select '1' to include all, see section 2.6.1 & Figure 7(d).
Append Label Columns:	Label columns to be copied from input file and appended to beginning of all output files.
Append Data Columns:	Optional data columns to be copied from input file and appended to end of output files 3 and 4. Columns such as 'Intensity' columns may need to be processed and summarized via the '-transform' option.
Append Label Columns:	Label columns to be copied from input file and appended to beginning of all output files.
Output File Name:	String to append to beginning of output file name.
Run Analysis:	Execute pipeline and output report, see panel (b) and result result files, see section 2.6.1 and Figure 7(d).
Visualization Type:	Type of out put visualisation (v0.751) currently heatmap visualisation implemented to visualise $log_2(Ratio)$ values for SILAC and LFQ across each sample group for each experiment.
Protein Label Type:	Select label from those listed in 'Append Label Columns:' to annotate the heat map rows
Cell Color:	Colors to represent high , low and missing values.
Visualization Results:	Execute visualisation button. Only active once 'Run Analysis' has completed. Disabled again if tab is changed.

(b)

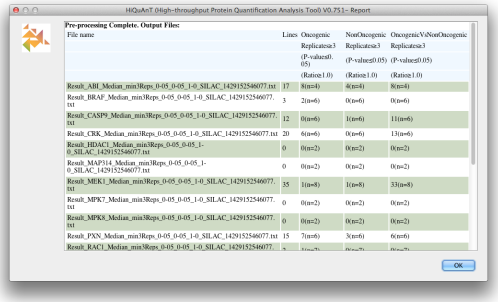
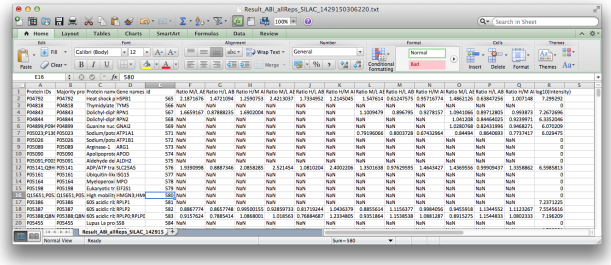


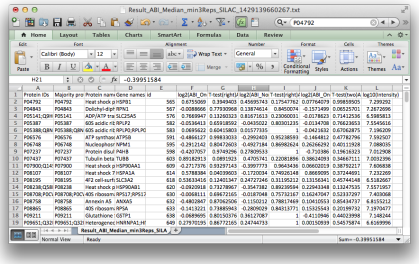
Figure 6: Step 3 - `HiQuAnT` Analyze Tab

Log File: Record of processing steps for GUI mode

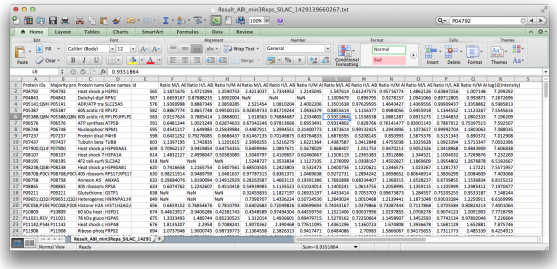
(a)



(c)



(b)



(d)

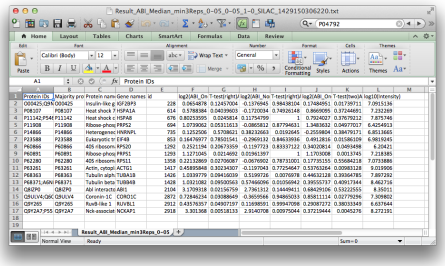


Figure 7: `HiQuAnT` output files for each experiment. All replicates, transformed & standardized(a), all replicates, transformed & standardized that pass the replicate threshold (b), experimental sub-group (e.g. Normal, Disease, Control) median values(c) and experimental sub-group values that pass the statistical and fold change thresholds.

2.6.3 Output Heatmap Visualization:

Pipeline results may then be visualised via the 'Output Visualization' feature that currently supports the interactive heatmap visualisation, see Figure 8. This visualises the results in Output File 4 (the $log_2(Ratio)$ column for each sub-group) and allows the user to see selected values (via the 'mouse-over' feature). There is also an option to hierarchically cluster rows based on different similarities (e.g. Pearson, Spearman, Cosine). There is also two options specifically for sparse datasets (such as those that would be produced when there is little overlap between baits in a multiple bait-prey dataset) that are based on the counts of the co-presence of entries regardless of direction (binary) and taking the direction of the protein change into account (directed binary). The User may also search for proteins/peptides of interest across row labels via the search option. If the list of protein names is two or greater in size or a subset of the heatmap is visualized. The heatmap may also be exported to to a publication quality PDF document or for may be used as supporting supplementary material.

2.6.4 Output Network/Graph Visualization:

Results may also be visualised via a experiment/protein (bipartite) network containing two node types - experiments and quantified proteins - that are connected by edges if a protein is quantified

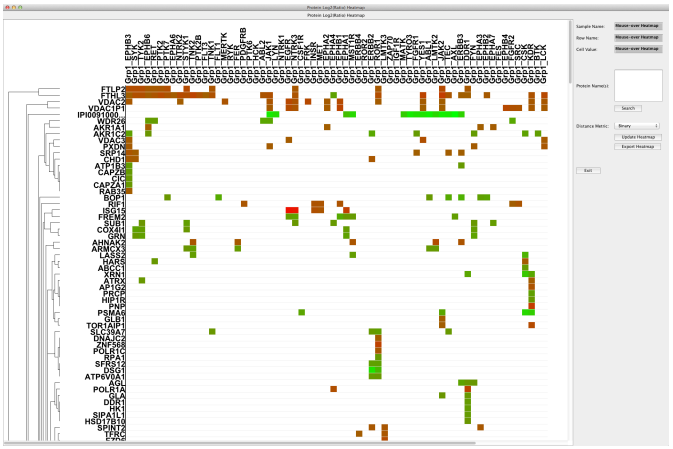


Figure 8: Interactive heat map visualization of output. The user may 'mouse-over' heat map cells to see values and search rows for protein/peptide of interest. The hear map may be exported as a publication quality PDF document.

in an experiment. Such visualisation highlights proteins that may be quantified/expressed across multiple experiments. Importantly information regarding the direction of the fold change between experimental conditions is also contained within the network file and used to colour edges (e.g default colours red for more expressed and green for less expressed). So for example, if an experiment contains diseased vs normal these edges the user may observe how quantified proteins change change over experiments such as time points or treatments. Of course such a visualisation is most useful for affinity-purification (AP) studies where each experiment represents the 'prey' that interact with a particular 'bait' protein. In such cases (if the experiment names contain the bait protein name in the same format as the ids within the input file) the network becomes a uni-partite Protein-Protein Interaction (PPI) network, where edges represent interactions that may occur (normal vs control) or that may vary across experimental conditions (e.g. disease vs normal). This type of visualisation may also be useful for interpreting any set of experiments where the treatment growth condition changes, such as a set of siRNAs knockdowns or time series.

2.7 Ancillary Tools

The Menu contains a Tools option which through which the mzTab conversion and input (see section 2.4). Also located here are further statistical methods and downstream functional analysis tools.

2.7.1 Multiple Group Statistics

Currently this option supports statistical analysis of multiple groups via one-way ANOVA (ANalysis Of VAriance). This can be run per any number of HiQuant experiments/output files (Stage 2 output files containing processed but as yet unmerged replicates) to assess which protein groups differ significantly across three or more experiments or more precisely cases in which proteins groups are unlikely to have been derived from the same 'population' of values for across all experiments. Results may also be corrected for multiple comparisons using the Benjamini-Hochberg

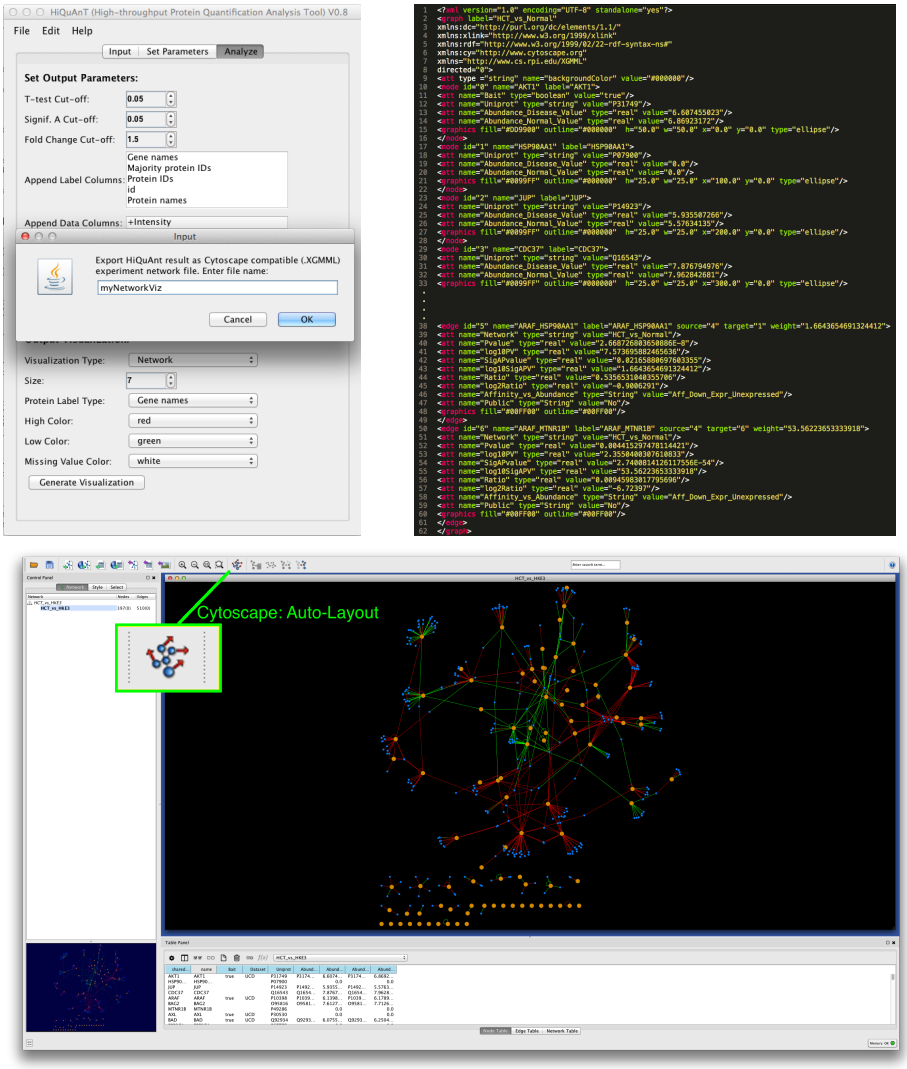


Figure 9: `HiQuAnT` can export a Cytoscape compatible XGMML file, containing protein abundance (edge) fold changes and significance values that can be viewed directly by Cytoscape and laid out with the click of a button (the Atuo Layout button). Alternatively `HiQuAnT` also supports the graph exchange format (.gexf) used by the Gephi Network Analysis and Visualisation application which is particularly useful for producing high quality PDF images.

corrections. Before this analysis can be performed the appropriate 'input' files containing the complete values for all sample replicates must be generated by selecting the 'All Stages' or 'Minimal replicates (Stage 2) on the 'Output File Type:' option on HiQuant's 'Analyze' tab running the analysis. The user must then go to 'Tools>Multiple Group Statistics...' where a dialogue will appear prompting the user to select the files on which to perform the ANOVA and then choose the parameters, including which experiments and sub-groups to select as groups, see Figure 10. Analysis results are output to a plain text tab delimited file.

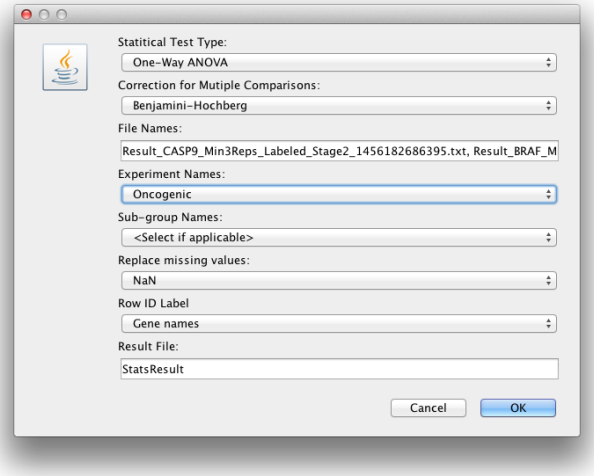


Figure 10: The parameter dialogue for HiQuant's ANOVA analysis.

2.7.2 DAVID Functional Annotation

Once a list of protein/gene names have been generated for a given analysis (e.g. list of prey for each bait/time point or list of gene/proteins changing across disease) it is common to perform further downstream functional enrichment analysis. The functional enrichment analysis tool from the DAVID (Database for Annotation, Visualization and Integrated Discovery) database (https://david.ncifcrf.gov/)is the most widely used online tool to perform functional enrichment analysis. HiQuant provides a feature to support rapid functional enrichment analysis across one or more HiQuant result file (Final result Stage 4 files) by harnessing the power of DAVID database via its RESTful based API (https://david.ncifcrf.gov/content.jsp?file=DAVID_API.html). This feature is accessible via 'Tools>DAVID Functional Annotation...' and supports all query features and parameter values provided by the API, which can be selected via HiQuant's custom search dialogue (see Figure 11). The query maybe be previewed prior to execution and if executed show the results in the system's default web-browser. Importantly this feature allows querying of multiple files in parallel, so will support, for example, simultaneous querying of 20 prey files in a bait-prey experiment. Note: currently the DAVID database enforces querying limits of 400 gene/protein ids and 200 hits per day from a machine (see (https://david.ncifcrf.gov/content.jsp?file=DAVID_API.html for further details.).

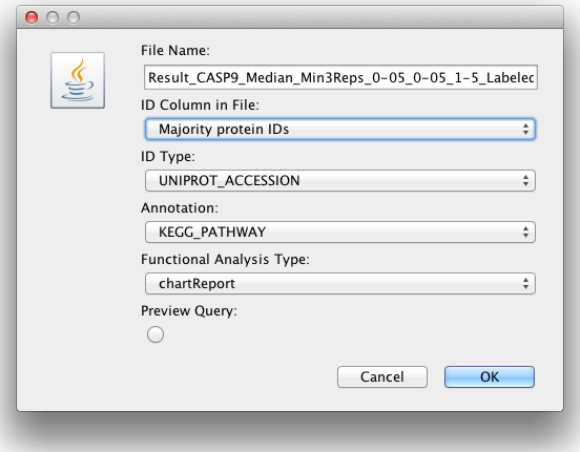


Figure 11: The parameter Dialogue for HiQuant's DAVID based functional enrichment analysis.

2.8 Help Options

A detailed description of every parameter is a available via 'Help > Parameter Descriptions' and additional online support and example files may directly linked to within the GUI application via 'Help > http://hiquant.primesdb.eu/'

2.9 Saving and Loading Analysis Pipeline Parmeters

When the pipeline has been executed the parameter settings may be saved to a configuration file (e.g. 'mySettings.config') via 'File > Generate Configuration File'. The configuration file may be re-loaded to facilitate similar subsequent analysis pipelines via 'File > Load Configuration File' or to enable reproducibility of quantification data analysis. Once this file has been generated the command line mode may be run by including this file as a parameter (and also ensuring the input file is available at the referenced path), see section 2.10.

2.10 Command Line Mode

Once a configuration file has been generated in the GUI mode `HiQuAnT` may be run in command line mode (by-pass the GUI). This mode can be accessed by providing a configuration file the first argument. The configuration file must have a config file type/extension e.g. *my file.config* (as generated via GUI mode) and is a plain text file in which the input file and parameter values may be edited by the user. By default, progress is output but `HiQuAnT` can be run in silent mode by providing a second argument (-s).

```
(1) "java -jar hiquant.jar myfile.config" (command line mode)
(2) "java -jar hiquant.jar myfile.config -s" (silent mode)
```

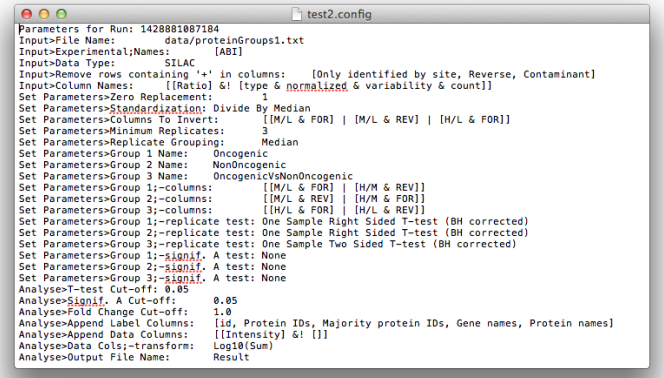


Figure 12: `HiQuAnT` GUI mode enables analysis parameters to be saved in a configuration file (my file.config) for use with later runs. This configuration file can also be used as an argument to run the in command mode, see section 2.10.