Protein significance analysis for mass spectrometry-based proteomics

Meena Choi (choi67@purdue.edu)

http://msstats.org

February 4, 2014

Contents

1 Statistical relative protein quantification for LC-MS, SRM and DIA

MSstats is an open-source R-based package for statistical relative quantification of peptides and proteins in mass spectrometry-based proteomics experiments.

Applicability

`MSstats` 2.0 is applicable to multiple types of sample preparation, including label-free workflows, workflows that use stable isotope labeled reference proteins and peptides, and workflows that use fractionation. It is applicable to global Liquid Chromatography coupled with Mass Spectrometry (LC-MS), targeted Selected Reaction Monitoring (SRM) and Data-Independent Acquisition (DIA or SWATH-MS). It is applicable to experiments that make arbitrary complex comparisons of experimental conditions or times.

`MSstats` 2.0 is currently not applicable to experiments that compare multiple metabolically labeled endogenous samples within a same run. It is not applicable to experiments with iTRAQ labeling. These experiments will be supported in the future.

Statistical functionalities

`MSstats` 2.0 performs three analysis steps. The first step, *data processing and visualization*, transforms and normalizes the intensities of the peaks, and generates workflow-specific and customizable numeric summaries for data visualization and quality control.

The second step, *statistical modeling and inference*, automatically detects the experimental design (e.g. group comparison or time course, presence of labeled reference peptides or proteins) from the data. It then reflects the experimental design, the type of spectral acquisition strategy, and the scope of conclusions (e.g. restricted to the subjects, or expanded to the underlying populations), and fits an appropriate linear mixed model by means of `lm` and `lmer` functionalities in R. The model is used to detect differentially abundant proteins or peptides, or to summarize the protein or peptide abundance in a single biological replicate or condition (that can be used, e.g. as input to clustering or classification).

The third step, *statistical experimental design*, views the dataset being analyzed as a pilot study, utilizes its variance components, and calculates the minimal number of replicates necessary to achieve a pre-specified statistical power.

Interoperability with existing computational tools

`MSstats` takes as input data in a tabular CVS format, which can be produced by any spectral processing tool such as SuperHirn, MaxQuant, Progenesis, MultiQuant, OpenMS or OpenSWATH.

For statistics experts, `MSstats` 2.0 satisfies the interoperability requirements of Bioconductor, and takes as input data in the `MSnSet` format (**?**). The command line-based workflow is partitioned into a series of independent steps, that facilitate the development and testing of alternative statistical approaches. `MSstats` 2.0 complies with the maintenance and documentation requirements of Bioconductor.

Finally, MSstats 2.0 is available as an external tool within Skyline (**?**). The external tool support within Skyline manages MSstats installation, point-and-click execution, parameter collection in Windows forms and output display. Skyline manages the annotations of the experimental design, and the processing of raw data. It outputs a custom report, that is fed as a single stream input into MSstats. This design buffers proteomics users from the details of the R implementation, while enabling rigorous statistical modeling.

Availability

MSstats 2.0 is available under the Artistic-2.0 license at msstats.org. MSstats as an external tool is available at http://proteome.gs.washington.edu/software/Skyline/tools.html. MSstats 2.0 is currently also under evaluation by Bioconductor (http://www.bioconductor.org).

Changes from MSstats 1.0 and SRMstats

For special cases of some experimental workflows, the underlying statistical methodology was previously described, and implemented in R-based packages MSstats 1.0 and SRMstats (**???**).

MSstats 2.0 supersedes MSstats 1.0 and SRMstats, in that it implements all the analysis steps that are available in these packages. In addition, it extends the methodology and the implementation, as follows.

- Unlike MSstats 1.0 (limited to label-free shotgun LC-MS experiments) and SRMstats (limited to SRM experiments), MSstats 2.0 integrates the statistical analysis steps across two sample preparation workflows (label-free, and using labeled reference proteins or peptides), and three spectral acquisition strategies (global LC-MS, targeted SRM and data-Independent DIA or SWATH-MS). The integration enables a greater flexibility of statistical modeling for each dataset.
- MSstats includes new statistical capabilities:
  - *Data processing:* quantification of between-run interferences, custom imputation of missing values by low-intensity signals, custom removal of spectral features.
  - *Data visualization:* more flexible plots of the protein profiles using ggplot2 functionalities in R, in particular displaying pre-specified proteins, customizing axis range, label and angle.
  - *Fitting linear mixed effects models:* fit appropriate linear models for specialized circumstances (e.g. proteins with a single replicate in a condition, proteins with a single feature, proteins with various patterns of missing intensities in groups or runs). For label-free experiments, fit appropriate models for experiments with and without technical replicates. Model the unequal variance between features using iterative weight least squares.
  - *Diagnostics of the quality of model fit:* residual plots and normal quantile-quantile plots across features of a protein, and separately for each feature of a protein to detect deviations from the model assumptions such as unequal variance.

– *Calculation of the sample size:* support of multiple modeling options in the analysis of the future experiment, such as expanded or restricted scope of biological replicates, and experiments with or without systematic interferences.

– *Summarization of protein abundance in a subject or in a condition on a relative scale:* support of label-free experiments and experiments with labeled reference proteins or peptides. Support of multiple output formats (long format and data matrix).

- MSstats 2.0 facilitates the interoperability with existing computational tools. In addition to taking as input a table in a CSV format, it can now be used as an external tool with Skyline by researchers who are unfamiliar with R. It also supports input in the `MSnSet` format, and partitions the analysis into a series of separate well-defined steps for interoperability with Bioconductor.

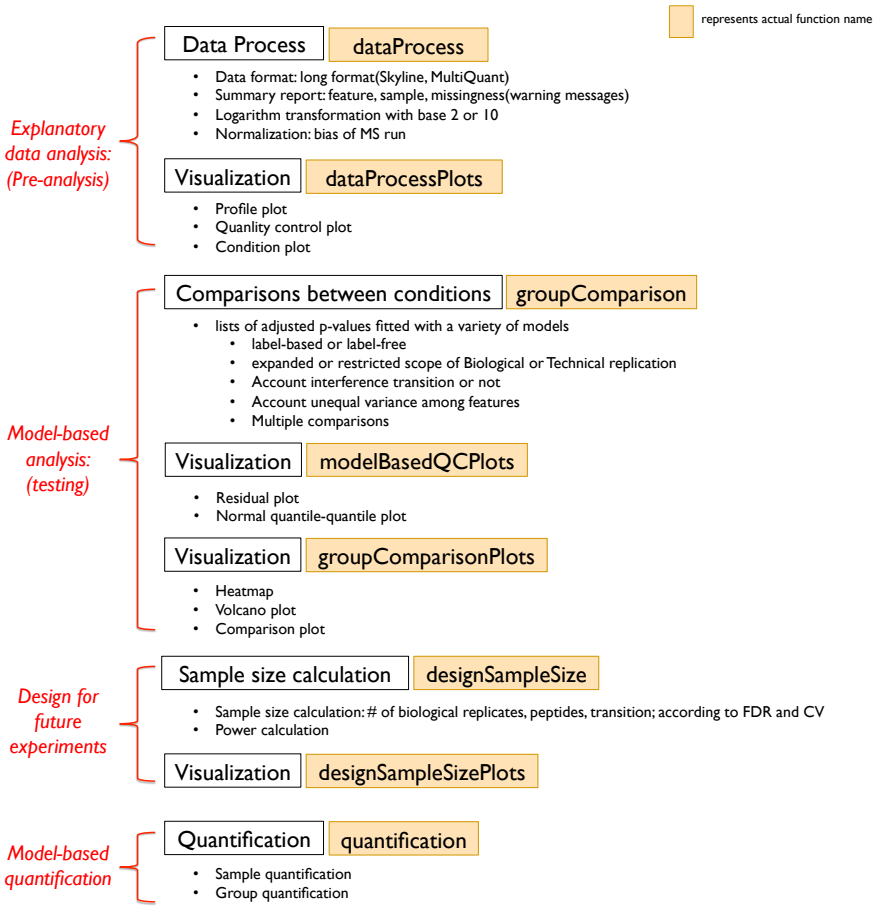


Figure 1: Overview of workflow in `MSstats`

To get started with this package in R, first load the package and then visit the help section of MSstats-package first by the following code.

```
> library(MSstats)
```

```
> ?MSstats
```

2 Allowable data formats

2.1 SRM with stable isotope labeled reference peptides

(1) **10-column format**

The preferred structure of data for use in `MSstats` is "long" format with 10 columns such as the report from other software for identifying peaks. The purpose of MSstats is for statistical analysis after peak identification and quantitation. Therefore, input for MSstats would be output of other software(such as Skyline, MaxQuant, mProphet, openSWATH and so on) for reading raw files such as mzXML, wiff files, or other types of files and identifying peaks. That software generates the report as .csv file. The raw data is required to contain variable of ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity. The variable names should be fixed but are not case-sensitive. Using MSstats report format in Skyline, this required input data will be automatically generated.

(a) `ProteinName`: This column needs information about Protein ID. Statistical analysis for each unique labeling in this column will be done. If you want peptide-level analysis, you can use peptide ID in this column.

(b) `PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge`: The combination of these 4 columns will be used as feature information. If the information of one or more columns is not available for the original raw data, please retain the column variables and type in fixed value (especially LC-MS data, See below LC-MS example input.). For example, the original raw data does not contain the information of `ProductCharge`, we retain the column `ProductCharge` and type in 0 for all transitions in `RawData`.

(c) `IsotopeLabelType`: This column need to say whether this measurement is based on the endogenous peptides (use 'L') or reference(labeled synthetic) peptides (use 'H').

(d) `Condition`: For case-control experiments, this should include case(disease) or control information. For time-course experiments, this should have time points (T1,T2,...) for each measurement. If you have combination of case-control and time-course experiments, this should also combination of these two, such as Disease_T1, Disease_T2, Control_T1,and Control_T2.

(e) `BioReplicate`: This should label with unique patient ID (i.e., same patients should label with the same ID).

`MSstats` does not require that technical replicates be specified in the data. They should be labeled the same as the corresponding biological replicate. `MSstats` detects the presence of technical replicates and accounts for them in the model-based analysis.

(f) `Run`: This means MS experiment measuremnts. If the same individual(patient) repetitively measured in the same Condition, `BioReplicate` values are the same,

but `Run` values are different based on repetition. If all the transitions from one sample is not measured in a single MS run, but split into multiple MS runs due to the limited transitions per run, please also specify this information.

(g) `Intensity`: This is required to be original signal without any log transformation and can be specified as the peak of height or the peak of area under curve. Any other quantitative representative of abundance can be used instead.

An example of a dataset in SRM is shown in Figure 2.

	A	B	C	D	E	F	G	H	I	J
1	ProteinName	PeptideSequence	PrecursorCharge	FragmentIon	ProductCharge	IsotopeLabelType	Condition	BioReplicate	Run	Intensity
2	ACEA	EILGHEIFFDWELP	3	y3	0	H	1	ReplA	1	66472.3847
3	ACEA	EILGHEIFFDWELP	3	y3	0	L	1	ReplA	1	5764.16228
4	ACEA	EILGHEIFFDWELP	3	y4	0	H	1	ReplA	1	101005.166
5	ACEA	EILGHEIFFDWELP	3	y4	0	L	1	ReplA	1	61.65238
6	ACEA	EILGHEIFFDWELP	3	y5	0	H	1	ReplA	1	90055.4993
7	ACEA	EILGHEIFFDWELP	3	y5	0	L	1	ReplA	1	472.691803
8	ACEA	TDSEAATLISSTID	2	y10	0	H	1	ReplA	1	43506.5425
9	ACEA	TDSEAATLISSTID	2	y10	0	L	1	ReplA	1	217.203553
10	ACEA	TDSEAATLISSTID	2	y7	0	H	1	ReplA	1	68023.0377
11	ACEA	TDSEAATLISSTID	2	y7	0	L	1	ReplA	1	725.284308
12	ACEA	TDSEAATLISSTID	2	y8	0	H	1	ReplA	1	68276.0489
13	ACEA	TDSEAATLISSTID	2	y8	0	L	1	ReplA	1	243.658527

Figure 2: Example dataset from SRM experiment (as a .CSV file) in "long" format. Each row corresponds to a single intensity. (See section 3 in details.)

(2) **How to assign `Condition, BioReplicate, Run` columns from design of experiment**

`Condition, BioReplicate, Run` columns are based on your design of experiments. `Condition` means Disease groups, different time points or the combination of groups and time points. `BioReplicate` needs sample or patient ID. `Run` means MS measurement run.

(a) **Case-control experiment**
Let's assume that there are two groups, disease and control, and each has 2 individuals (biological replicates), which are total 4 individual. Also we repeat to measure each individual twice as technical replicates. Then the possible values in these 3 columns are in Table 1.

Condition	BioReplicate	Run
Disease	sample1	1
Disease	sample1	2
Disease	sample2	3
Disease	sample2	4
Control	sample3	5
Control	sample3	6
Control	sample4	7
Control	sample4	8

Table 1: Possible assignment for design of experiment information in case-control

If there is no technical replicate, `BioReplicate` and `Run` will be the same. In case-control, `BioReplicate` is nested in `Condition` and `Run` is the combination of biological replicates and technical replicates.

(b) **Time-course experiment**
In time-cour experiment, the same individual is repetitively measured across conditions (time points). Let's assume that there are four individuals (biological replicates). We measure each individual at two time points, time1 and time2, repetively. Then the values in these 3 columns are one of rows in Table 2.

Condition	BioReplicate	Run
Time1	sample1	1
Time2	sample1	2
Time1	sample2	3
Time2	sample2	4
Time1	sample3	5
Time2	sample3	6
Time1	sample4	7
Time2	sample4	8

Table 2: Possible assignment for design of experiment information in time-course

If there is no technical replicate, `Run` column is the combination of `Condition` and `BioReplicate`. Then if there are technical replicates, `Run` column is the combination of `Condition`, `BioReplicate`, and technical replicates.

(3) *MSnSet* **format**

`MSstats` also allows data to be in the format of `MSnSet`, suggested general format on the proteomics in `MSstats` package. A MSnSet contains several components, of which the most commonly accessed are the assayData, phenoData, and the featureData components. The assayData is a matrix of intensities, where each row corresponds to the unit of analysis, a peptide feature; the columns correspond to sample ids. The phenoData contains columns that describe the biological samples, conditions in the experiment. The featureData contains columns describing the peptide features, such as the name or id of the underlying protein and information of features.

For data stored as an expression set, group labels information are required. If more than one variable is listed in group argument, a concatenated variable is created based on the variables. The remaining information (peptide feature ids, biological replicate ids, and abundance) can be extracted from the rows and columns of featureData and phenoData or the users can assign them based on their design of experiments.

2.2 Label-free LC-MS

(1) **10-column format**

For label-free LC-MS, the required input is 10-column format, which is the same as for SRM in section 2.1 (1). Only difference between expeirments in input format of MSstats is how to assign feature information. Feature information will consist of 4 columns,

`PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge`. If any value of 4 columns is different, it will make different feature ID. The datasets from LC-MS experiment do not have multiple fragments per peptide. Therefore `PrecursorCharge, FragmentIon, ProductCharge` columns might not be available. You can retain the column variables with any fixed value (such as NA or 0). In example data set, peptide feature ID is used in `FragmentIon` column to distinguish feature.(Figure 3)

ProteinName	PeptideSequence	PrecursorCharge	FragmentIon	ProductCharge	IsotopeLabelType	Condition	BioReplicate	Run	Intensity
bovine	S.PVDIDTK_5	NA	5	NA	L	C1	1	1	2636791.5
bovine	S.PVDIDTK_5	NA	5	NA	L	C1	1	2	1992418.5
bovine	S.PVDIDTK_5	NA	5	NA	L	C1	1	3	1982146.38
bovine	S.PVDIDTK_5	NA	5	NA	L	C2	1	4	5019594
bovine	S.PVDIDTK_5	NA	5	NA	L	C2	1	5	4560467.5
bovine	S.PVDIDTK_5	NA	5	NA	L	C2	1	6	3627848.75
bovine	S.PVDIDTK_5	NA	5	NA	L	C5	1	13	145511.83
bovine	S.PVDIDTK_5	NA	5	NA	L	C5	1	14	291829.69
bovine	S.PVDIDTK_5	NA	5	NA	L	C6	1	16	786667.38
bovine	S.PVDIDTK_5	NA	5	NA	L	C6	1	17	705295.31
bovine	S.PVDIDTK_5	NA	5	NA	L	C6	1	18	453448.78
bovine	S.PVDIDTK_5	NA	5	NA	L	C3	1	7	NA

Figure 3: Example dataset from LC-MS shortgun experiment.(See section 4 in details.)

(2) **How to assign `Condition, BioReplicate, Run` columns from design of experiment**

For label-free LC-MS, it is the same as for SRM in section 2.1 (2).

(3) *MSnSet* **format**

If there are all required information in *MSnSet*, it is the same as for SRM in section 2.1 (3).

2.3 Label-free DIA

(1) **10-column format**

For label-free DIA, the required input is 10-column format, which is the same as for SRM in section 2.1 (1). The values of required columns for DIA can be extracted from output of other software after identifing and quantifying peaks, such as openSWATH. Each row in required input for `MSstats` corresponds to single intensity value. Only difference between expeirments in input format of MSstats is how to assign feature information. For example, openSWATH provides the information about 4 columns, `PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge`. In this case, the format of input is the same as SRM experiment. If some variables are not available or you want to use feature ID itself, use feature ID values in `FragmentIon` or `ProductCharge`.

In example data set, feature ID (or transition ID) is used in `ProductCharge` column to distinguish feature.(Figure 4)

(2) **How to assign `Condition, BioReplicate, Run` columns from design of experiment**

For label-free DIA, it is the same as for SRM in section 2.1 (2).

ProteinName	PeptideSequence	PrecursorCharge	FragmentIon	ProductCharge	IsotopeLabelType	Condition	BioReplicate	Run	Intensity
350748	TPPAAVLLK	2	y7	109401	L	2	1	3	257486
350748	TPPAAVLLK	2	y7	109401	L	2	2	4	141159
350748	TPPAAVLLK	2	y7	109401	L	1	1	1	452908
350748	TPPAAVLLK	2	y7	109401	L	1	2	2	348222
515084	NIC[160]VNAIAPGFIESDMTGVLPEK	3	y3	7717	L	2	1	3	12753
515084	NIC[160]VNAIAPGFIESDMTGVLPEK	3	y3	7717	L	2	2	4	12857
515084	NIC[160]VNAIAPGFIESDMTGVLPEK	3	y3	7717	L	1	1	1	89652
515084	NIC[160]VNAIAPGFIESDMTGVLPEK	3	y3	7717	L	1	2	2	76724
515084	MVNEAIESLGSIDVLVNNAGITNDK	3	y9	57971	L	2	1	3	2052
515084	MVNEAIESLGSIDVLVNNAGITNDK	3	y9	57971	L	2	2	4	1050
515084	MVNEAIESLGSIDVLVNNAGITNDK	3	y9	57971	L	1	1	1	10772
515084	MVNEAIESLGSIDVLVNNAGITNDK	3	y9	57971	L	1	2	2	10516

Figure 4: Example dataset from DIA experiment.(See section 5 in details.)

(3) *MSnSet* **format**

If there are all required information in *MSnSet*, it is the same as for SRM in section 2.1 (3).

3 Example workflow with label-based SRM: time-course investigation of *S. Cerevisiae*

3.1 Experimental design

The example dataset is from a label-based SRM experiment of a time course yeast study. This is a partial data set obtained from a published study (**?**). The experiment targeted 45 proteins in the glycolysis/gluconeogenesis/TCA cycle/glyoxylate cycle network, which spans the range of protein abundance from less than 128 to 10E6 copies per cell. Three biological replicates were analyzed at ten time points (T1-T10), while yeasts transited through exponential growth in a glucose-rich medium (T1-T4), diauxic shift (T5-T6), post-diauxic phase (T7-T9), and stationary phase (T10). Prior to trypsinization, the samples were mixed with an equal amount of proteins from the same N15-labeled yeast sample, which was used as a reference. Each sample was profiled in a single mass spectrometry run, where each protein was represented by up to two peptides and each peptide by up to three transitions. The goal of this study is to detect significantly change in protein abundance across time points. Transcriptional activity under the same experimental conditions has been previously investigated by (DeRisi et. al., 1997). Genes coding for 29 of the proteins are differentially expressed between conditions similar to those represented by T7 and T1 and could be treated as external sources to validate the proteomics analysis. In this exampled data set, two of the targeted proteins are selected and validated with gene expression study: Protein IDHC (gene name IDP2) is differentially expressed in time point 1 and time point 7, whereas, Protein PMG2 (gene name GPM2) is not. The protein names are based on Swiss Prot Name. This dataset is stored in the package in a data structure `RawData`.

To know about example dataset, `RawData`, visit the help section of RawData first by the following code.

```
> ?RawData
```

3.2 Reading the data

Labeled reference peptide SRM data from two proteins identified in yeast study is stored in "long" format as a data.frame labeled `RawData` in `MSstats` ; the data can be accessed once the package has been installed and loaded in `R`.

```
> head(RawData)
```

```
    ProteinName PeptideSequence PrecursorCharge FragmentIon ProductCharge
243        IDHC   ATDVIVPEEGELR               2          y7             NA
244        IDHC   ATDVIVPEEGELR               2          y7             NA
245        IDHC   ATDVIVPEEGELR               2          y8             NA
246        IDHC   ATDVIVPEEGELR               2          y8             NA
247        IDHC   ATDVIVPEEGELR               2          y9             NA
248        IDHC   ATDVIVPEEGELR               2          y9             NA
    IsotopeLabelType Condition BioReplicate Run   Intensity
243                H         1        ReplA   1 84361.08350
244                L         1        ReplA   1   215.13526
245                H         1        ReplA   1 29778.10188
246                L         1        ReplA   1    98.02134
247                H         1        ReplA   1 17921.29255
248                L         1        ReplA   1    60.47029
```

3.3 Pre-processing data and quality control of MS runs

(1) **Data processing steps and options**

All data need to pre-processing prior to statistical analysis. Possible steps may include:

- logarithm transformation with base 2 (default) or 10 of the intensities
- normalization to remove systematic bias between MS run after logarithm transformation. For label-based experiments, constant normalization is performed based on reference signals across runs among all proteins. For label-free experiments, constant normalization based on endogenous signals across runs among all proteins is performed. Therefore users need to be careful for label-free experiments. In this case, normalization with standard protein or any other normalization has to be done prior to use of the package. If normalization=FALSE, no normalization is performed.
- calculation of between-run-interference score (based on the correlation between mean of peptide by run and intensity). It detects interference for further statistical model.

After pre-processing data, the summary of experimental design will be showed. When a transition is missing completely in a condition or a MS run, a warning message is sent to the console notifying the user of the missing transitions. The quantitative data after data pre-processing and quality control of MS runs not only contain the same variable from the raw data, but with addition variables for statistical model fitting and group comparison. For examples, Variable ABUNDANCE represents the final measurement, which could be normalized or not depending on the options you specified in `dataProcess`. Default option in `dataProcess` is with log2 transformation and normalization.

To get started with this function, visit the help section of dataProcess first by the following code.

```
> ?dataProcess

> QuantData<-dataProcess(RawData)

  Summary of Features :
                         count
# of Protein                 2
# of Peptides/Protein      2-2
# of Transitions/Peptide   3-3

  Summary of Samples :
                            1 2 3 4 5 6 7 8 9 10
# of MS runs                3 3 3 3 3 3 3 3 3  3
# of Biological Replicates 3 3 3 3 3 3 3 3 3  3
# of Technical Replicates  0 0 0 0 0 0 0 0 0  0

> head(QuantData)

   PROTEIN             PEPTIDE TRANSITION                    FEATURE LABEL
1     IDHC     ATDVIVPEEGELR_2      y7_NA      ATDVIVPEEGELR_2_y7_NA     H
3     IDHC     ATDVIVPEEGELR_2      y8_NA      ATDVIVPEEGELR_2_y8_NA     H
5     IDHC     ATDVIVPEEGELR_2      y9_NA      ATDVIVPEEGELR_2_y9_NA     H
7     IDHC DQTNDQVTVDSATATLK_2     y10_NA DQTNDQVTVDSATATLK_2_y10_NA     H
9     IDHC DQTNDQVTVDSATATLK_2     y11_NA DQTNDQVTVDSATATLK_2_y11_NA     H
11    IDHC DQTNDQVTVDSATATLK_2      y8_NA  DQTNDQVTVDSATATLK_2_y8_NA     H
   GROUP_ORIGINAL SUBJECT_ORIGINAL RUN GROUP SUBJECT SUBJECT_NESTED INTENSITY
1               1            ReplA   1     0       0            0.0 84361.083
3               1            ReplA   1     0       0            0.0 29778.102
5               1            ReplA   1     0       0            0.0 17921.293
7               1            ReplA   1     0       0            0.0  4481.229
9               1            ReplA   1     0       0            0.0  1871.042
11              1            ReplA   1     0       0            0.0  2640.060
   ABUNDANCE SuggestToFilter
1   15.84764               0
3   14.34531               0
5   13.61273               0
7   11.61303               0
9   10.35297               0
11  10.84970               0
```

(2) **Visualization for explanatory data analysis**

To illustrate the quantitative data after data-preprocessing and quality control of MS runs, dataProcessPlots takes the quantitative data from function (`dataProcess`) as input and can generates three types of figures.

- Profile Plot (Figure 6): identify the potential sources of variation of each protein. X-axis is run. Y-axis is log-intensities of transitions. Reference/endogenous signals are in the left/right panel. Line colors indicate peptides and line types indicate transitions.

- QC Plot (Figure 5) : illustrate the systematic bias between MS runs. After normalization, the reference signals for all proteins should be stable across MS runs. X-axis is run. Y-axis is log-intensities of transition. Reference/endogenous signals are in the left/right panel. The pdf file contains (1) QC plot for all proteins and (2) QC plots for each protein separately.
- Condition Plot (Figure 7): illustrate the systematic difference between conditions. X-axis is condition. Y-axis is log ratio of endogenous over reference. For label-free, Y-axis is log intensity of endogenous. If scale is TRUE, the levels of conditions is scaled according to its actual values at x-axis. Red points indicate the mean of log ratio for each condition. If interval is "SE", blue error bars indicate the confidence interval with 0.95 significant level for each condition. If interval is "SD", blue error bars indicate the standard deviation for each condition.The interval is not related with model-based analysis.

There are several options such as size of label, axis, file and so on. For example, `which.Protein` can be assigned as protein list to draw plots. and `address` is the name of folder that will store pdf file for plots. If `address`=FALSE, plot will be just showed in window.

To get started with this function and information about detailed options, visit the help section of dataProcessPlots first by the following code.

```
> ?dataProcessPlots
```

(a) Quality control plot

```
> dataProcessPlots(data=QuantData,type="QCPlot")
```

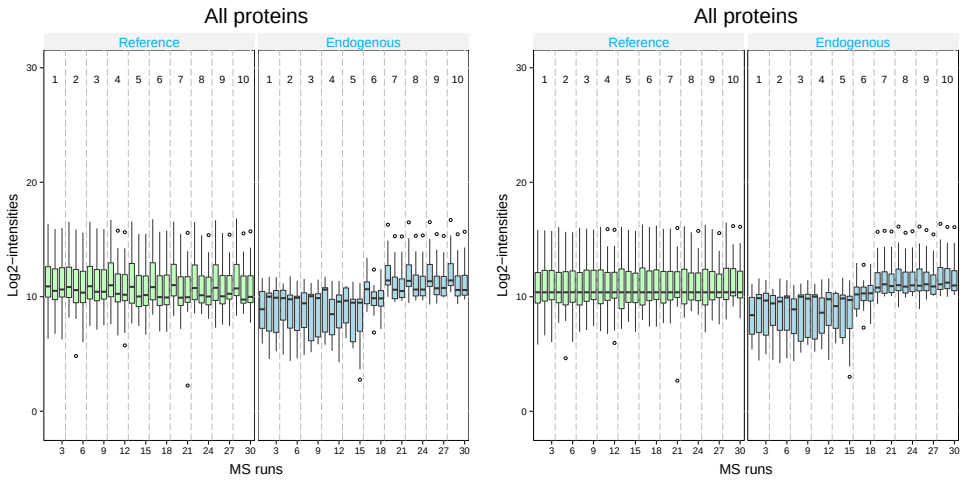


Figure 5: All protein together in QCplots before(left panel) and after(right panel) normalization

(b) Profile plot

```
> dataProcessPlots(data=QuantData,type="ProfilePlot")
```

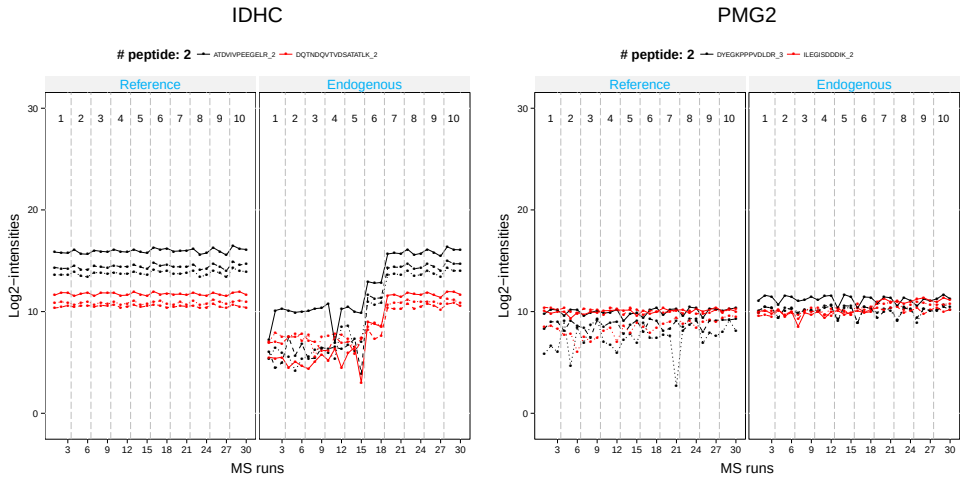


Figure 6: Profile plots for Protein IDHC and PMG2 after normalization. Quantitative profiles of reference and endogenous transitions in label-based SRM experiments. X-axis: run. Y-axis: log-intensities of transitions. Line types indicate transitions, and colors indicate peptides.

(c) Condition plot

```
> dataProcessPlots(data=QuantData,type="ConditionPlot")
```

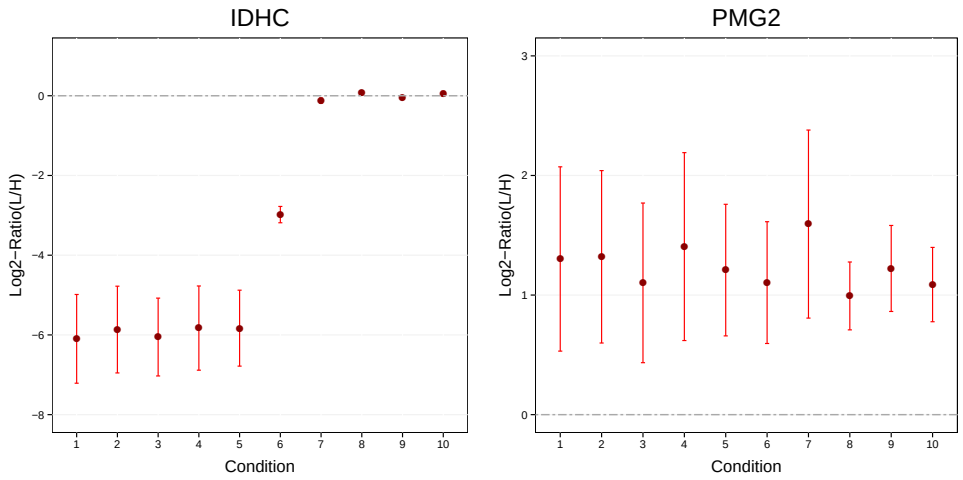


Figure 7: Condition plots for Protein IDHC and PMG2. Illustrate the systematic difference between conditions. X-axis is condition. Y-axis is log ratio of endogenous over reference. Red points indicate the mean of log ratio for each condition. Blue error bars indicate the confidence interval with 0.95 significant level for each condition.

3.4 Model-based inference

3.4.1 Setting up a linear mixed effects model

A statistical model has the capability to formally characterize the sources of variation of all measurements that pertain to a protein, and to distinguish the systematic patterns of differen-

tial abundance from noise. A statistical model describes the relationship between a *response* variable and a set of variables that have been observed along with the response. In proteomic experiments, categorical variables, whose values are denoted by labels, may include peptide features, conditions under which replicates are observed, e.g., disease state, treatment, or time, and biological replicates.

`MSstats` tests for significant changes in protein abundance across conditions based on a family of linear mixed-effects models in LC-MS, SRM, DIA experiment. Experimental design of case-control study (patients are not repeatedly measured) or time course study (patients are repeatedly measured) is automatically determined based on proper statistical model. Other choices of model specification include :

1. labeling technique : TRUE(default) represent the labeled reference peptide study. FALSE represent label-free study.

2. scope of inference for biological replicates(variable SUBJECT) or technical replicates(variable RUN): restricted scope, which limited conclusion from the model to observed biological unites or technical units, or expanded scope, which expands conclusion from the model to the population of biological units or technical units. The underlying model fitting functions are `lm` and `lmer` for fixed-effects models and mixed-effects models, respectively.

3. interference: TRUE(default) means data contain interference transitions and need additional model interaction to address the interference. FALSE means data contain no interference transitions and no need additional model interaction to address the interference.

4. unequal variance between features: If the unequal variation of error for different peptide features is detected, then a possible solution is to account for the unequal error variation by means of a procedure called iteratively re-weighted least squares. featureVar=TRUE performs an iterative fitting procedure, in which features are weighted inversely proportionally to the variation in their intensities, so that feature with large variation are given less importance in the estimation of parameters in the model.

5. In general, we recommend to keep interaction(`interference`=TRUE) and to use constant variance among features(`featureVar`=FALSE) for SRM experiment.

To get started with this function, visit the help section of groupComparison first by the following code.

```
> ?groupComparison
```

Also comparison of interest can be assigned as `contrast.matrix`. Based on the levels of conditions, specify 1 or -1 to the conditions of interests and 0 otherwise. The levels of conditions are sorted alphabetically. Command `levels(QuantData$GROUP_ORIGINAL)` can illustrate the actual order of the levels of conditions. When a feature is missing completely in a condition or a MS run, a warning message is sent to the console notifying the user of the missing feature. Additional filtering or imputing process is required before model fitting.

For example, we attempt to compare differential abundance between time 1 and 7 in a set of targeted proteins. In statistical terms, we test $H_0 : L = \mu_{\mathrm{T7}} - \mu_{\mathrm{T1}} = 0$ against the

alternative $H_a : L = \mu_{T7} - \mu_{T1} \neq 0$, where μ_{T1} is the mean abundance of the protein in the time 1. In this label-based SRM experiment, we recommend the fitted model with expanded scope of technical replication (i.e. `labeled`=TRUE, `scopeOfTechReplication`="expanded" as default). Finally, the `groupComparison` function is used to simultaneously estimate the log fold change L and the corresponding standard error of the estimate, and to test H_0.

```
> levels(QuantData$GROUP_ORIGINAL)

 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"

> comparison<-matrix(c(-1,0,0,0,0,0,1,0,0,0),nrow=1)
> row.names(comparison)<-"T7-T1"

> testResultOneComparison<-groupComparison(contrast.matrix=comparison, data=QuantData)
> testResultOneComparison$ComparisonResult

  Protein Label    log2FC        SE    Tvalue  DF    pvalue adj.pvalue
1    IDHC T7-T1 6.0227709 0.1438273 41.875016 100 0.0000000  0.0000000
2    PMG2 T7-T1 0.3044394 0.2242354  1.357677 100 0.1776219  0.1776219
```

The testing result contains variable of Protein, Comparison(Label), log2 fold change(logFC), standard error (SE), T values (Tvalue), degree of freedom (DF), raw p-values (pvalue), adjusted p-values based on Benjamini and Hochberg method to collect multiple testing issue and further control false discovery rate (adj.pvalue). The positive logFC values indicates up-regulated and the negative log FC values indicates down-regulated in time 7. The small adjusted p-value means that the regulation is statistically significant.

Also multiple companions between groups are possible. This is a testing result of multiple comparisons of `QuantData` based on the intensity-based linear model with expanded scope. The comparison is time 1 vs time 3 (T3-T1), time 1 vs time 7 (T7-T1), time 1 vs time 9 (T9-T1). This result dataset is stored in the package in a data structure `testResultMultiComparisons`.

```
> comparison1<-matrix(c(-1,0,1,0,0,0,0,0,0,0),nrow=1)
> comparison2<-matrix(c(-1,0,0,0,0,0,1,0,0,0),nrow=1)
> comparison3<-matrix(c(-1,0,0,0,0,0,0,0,1,0),nrow=1)
> comparison<-rbind(comparison1,comparison2, comparison3)
> row.names(comparison)<-c("T3-T1","T7-T1","T9-T1")
> testResultMultiComparisons<-groupComparison(contrast.matrix=comparison,data=QuantData)
> testResultMultiComparisons$ComparisonResult

  Protein Label     log2FC        SE     Tvalue  DF    pvalue adj.pvalue
1    IDHC T3-T1  0.1052223 0.1438273  0.7315877 100 0.4661312  0.4661312
4    PMG2 T3-T1 -0.1830632 0.2242354 -0.8163883 100 0.4162186  0.4661312
2    IDHC T7-T1  6.0227709 0.1438273 41.8750159 100 0.0000000  0.0000000
5    PMG2 T7-T1  0.3044394 0.2242354  1.3576775 100 0.1776219  0.1776219
3    IDHC T9-T1  6.1204163 0.1438273 42.5539234 100 0.0000000  0.0000000
6    PMG2 T9-T1  0.0718434 0.2242354  0.3203927 100 0.7493392  0.7493392
```

3.4.2 Verifying the assumption of the model

Results based on statistical models are accurate as long as the assumptions of the model are met. The model assumes that the measurement errors are normally distributed with mean

0 and constant variance σ^2_{Error}. The assumption of a constant variance can be checked by examining the residuals from the model, i.e., the deviations of the predicted intensities from the actual intensities. If a plot of residuals against predicted values shows a random scatter, then the assumption is appropriate. Residual plots can be produced for all proteins in the dataset by using the `modelBasedQCPlots` function with `type="ResidualPlots"` in `MSstats` . Option, `which.Protein` can be assigned as protein list to draw plots and `address` is the name of folder that will store pdf file for plots. If `address`=FALSE, plot will be just showed in window.

To check whether the errors are well approximated by a normal distribution, the `modelBasedQCPlots` function with `type="QQPlots"` in `MSstats` can be used. This function produces a normal quantile-quantile plot for each protein. If points fall approximately along a straight line, then the assumption is appropriate for that protein. Only large deviations from the line are problematic.

The input of this function is "ModelQC" in the testing results from function (`groupComparison`). The example result is `testResultMultiComparisons`

To get started with this function, visit the help section of modelBasedQCPlots first by the following code.

```
> ?modelBasedQCPlots
```

```
> modelBasedQCPlots(data=testResultMultiComparisons$ModelQC, type="ResidualPlots",
+                   which.Protein="PMG2", address=FALSE)
```

```
> modelBasedQCPlots(data=testResultMultiComparisons$ModelQC,type="QQPlots",
+                   which.Protein="PMG2", address=FALSE)
```

In practice, it is common for the variation of error to be unequal for different peptide features. If this is diagnosed from the residual plots, then a possible solution is to account for the unequal error variation by means of a procedure called *iteratively re-weighted least squares*. This procedure is implemented in `MSstats` by means of the `featureVar` argument in `groupComparison`. `featureVar` = TRUE performs an iterative fitting procedure, in which features are weighted inversely proportional to the variation in their intensities, so that features with large variation are given less importance in the estimation of parameters in the model.

3.4.3 Testing for protein-level differential abundance between conditions

To summarize the results of fold changes and adjusted p-values for differentially abundant proteins, `groupComparisonPlots` takes testing results from function (`groupComparison`) as input : (1) volcano plot (specify "VolcanoPlot" in option type) for each comparison separately; (2) heatmap (specify "Heatmap" in option type) for multiple comparisons ; (3) comparison plot (specify "ComparisonPlot" in option type) for multiple comparisons per protein.

To get started with this function, visit the help section of groupComparisonPlots first by the following code.

```
> ?groupComparisonPlots
```

- Volcano plot : illustrate actual log-fold changes and adjusted p-values for each comparison separately with all proteins. The x-axis is the log fold change. The base of logarithm

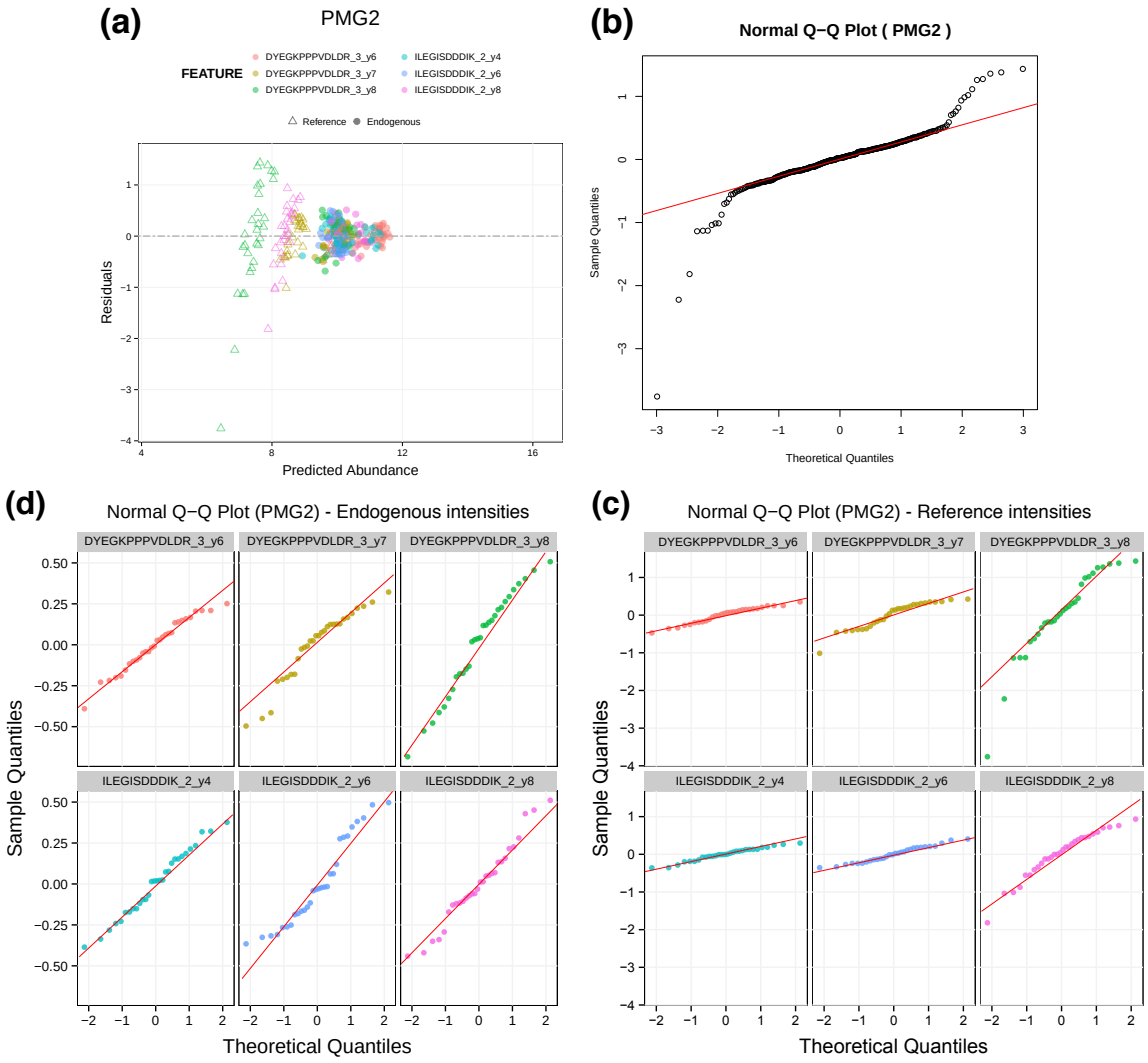


Figure 8: (a) Residual plot (upper left panel) and (b) normal quantile-quantile plot (upper right panel) for Protein PMG2. The unequal variance between feature is detected for Protein PMG2. The pattern in the qqplot is due to deviations from the assumption of constant variance, and not necessarily from the assumption of Normality. (c) and (d) The normal quantile-quantile plot per feature (bottom panel) illustrates that. It shows that (1) since the slope of the Q-Q plot is related to the variance of the distribution of the errors, the different slopes of the lines offer another indication of unequal variance, (2) the assumptions of Normality of each separate feature is more plausible. The features with lower intensities have a larger variance, and are more likely to deviate from the Normality assumption.

transformation is the same as specified in "logTrans" from `dataProcess`. The y-axis is the negative log2 adjusted p-values. The horizontal dashed line represents the FDR cutoff. The points below the FDR cutoff line are non-significantly abundant proteins (colored in black). The points above the FDR cutoff line are significantly abundant proteins (colored in red/blue for up-/down-regulated). If fold change cutoff is specified (FCcutoff

= specific value), the points above the FDR cutoff line but within the FC cutoff line are non-significantly abundant proteins (colored in black).

- Heatmap : illustrate up-/down-regulated proteins for multiple comparisons with all proteins. Each column represents each comparison of interest. Each row represents each protein. Color red/blue represents proteins in that specific comparison are significantly up-regulated/down-regulated proteins with FDR cutoff and/or FC cutoff. The color scheme shows the evidences of significance. The darker color it is, the stronger evidence of significance it has. Color gold represents proteins are not significantly different in abundance.

- Comparison plot : illustrate log-fold change and its variation of multiple comparisons for single protein. X-axis is comparison of interest. Y-axis is the log fold change. The red points are the estimated log fold change from the model. The blue error bars are the confidence interval with 0.95 significant level for log fold change. This interval is only based on the standard error, which is estimated from the model.

The input of this function is "ComparisonResult" in the testing results from function (`groupComparison`). The example result is `testResultMultiComparisons` based on the testing results of comparison of time1 and time 7.

(1) Volcano plot

```
> head(testResultMultiComparisons$ComparisonResult)
```

```
  Protein Label     log2FC        SE     Tvalue  DF    pvalue adj.pvalue
1    IDHC T3-T1  0.1052223 0.1438273  0.7315877 100 0.4661312  0.4661312
4    PMG2 T3-T1 -0.1830632 0.2242354 -0.8163883 100 0.4162186  0.4661312
2    IDHC T7-T1  6.0227709 0.1438273 41.8750159 100 0.0000000  0.0000000
5    PMG2 T7-T1  0.3044394 0.2242354  1.3576775 100 0.1776219  0.1776219
3    IDHC T9-T1  6.1204163 0.1438273 42.5539234 100 0.0000000  0.0000000
6    PMG2 T9-T1  0.0718434 0.2242354  0.3203927 100 0.7493392  0.7493392
```

1. Default : sig=0.05 / No FCcutoff / No ylimUp / with Protein name

```
> groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult,
+                      type="VolcanoPlot",which.Comparison=c("T7-T1"), address=FALSE)
```

2. FCcutoff=70 / ylimUp=100 / without Protein name

Note : FCcutoff=70 is due to demonstration purpose.

```
> groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult,
+                      type="VolcanoPlot",FCcutoff=70, ylimUp=100,
+                      ProteinName=FALSE, which.Comparison=c("T7-T1"), address=FALSE)
```

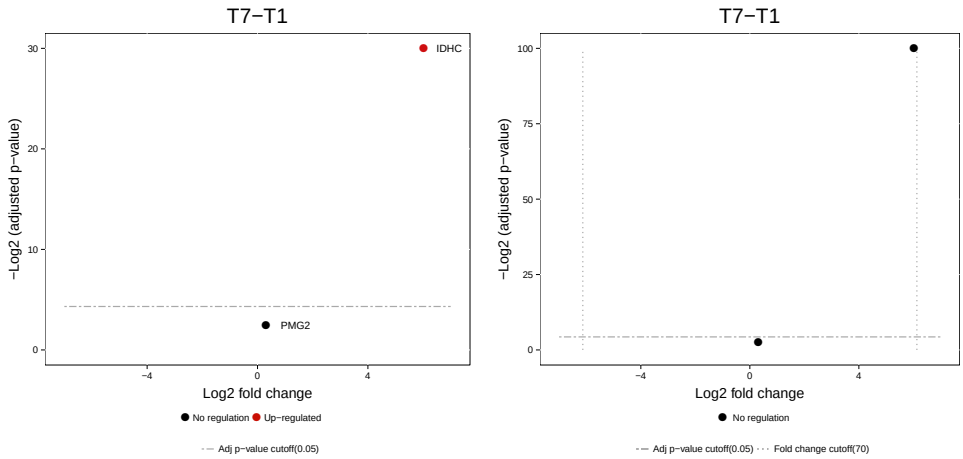


Figure 9: Volcano plot of a single comparison. The x-axis is the log fold change of two conditions. The y-axis is the negative log2 adjusted p-values. The dashed line represents the FDR cutoff of 0.05(default setup `sig`=0.05). The points below the cutoff line are the non-significantly abundant proteins (colored in black). Left volcano plot is the result for default option. Right volcano plot is the result for setting up `FCcutoff`=70 and upper limit of y-axis is 100 without protein names. The setup of FCcutoff=70 is for demonstrate purpose onlythat protein IDHC is not significant after applying fold change cutoff.

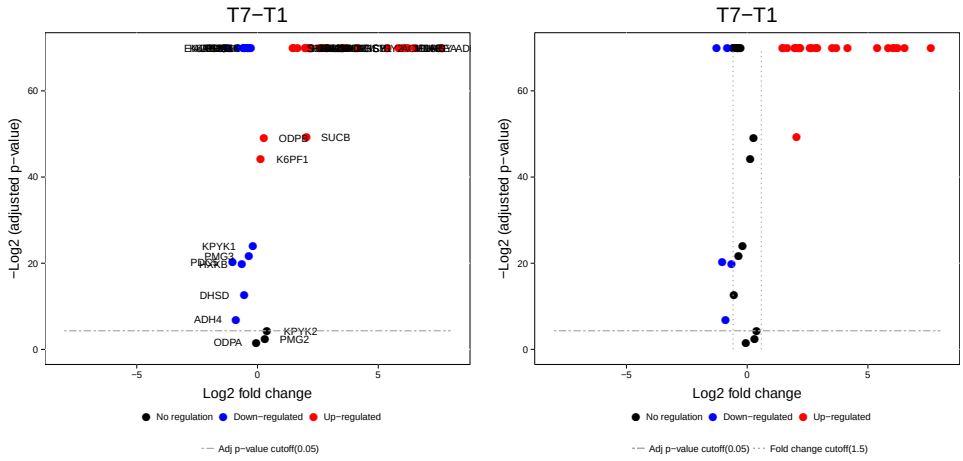


Figure 10: Volcano plots of a single comparison for 45 proteins from the example time courses yeast study with all 45 proteins. Left volcano plot is the result for default option. Right volcano plot is the result for setting up `FCcutoff`=1.5 and upper limit of y-axis is 60 without protein names.

(2) Heatmap

1. Default : sig=0.05 / No FCcutoff

```
> groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult,
+                      type="Heatmap",address=FALSE)
```

2. FCcutoff=70

```
> groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult,
+                      type="Heatmap",FCcutoff=70,address=FALSE)
```

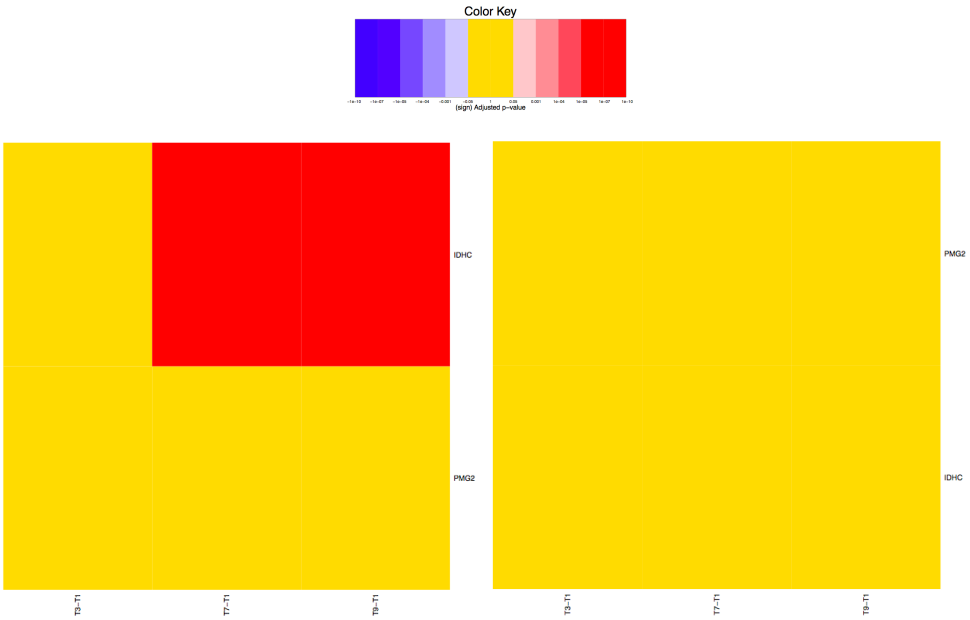


Figure 11: Heatmap of multiple comparisons. The column represents different comparisons. The row represents each protein. Color red/blue represents proteins in that specific comparison are significantly up-regulated/down-regulated proteins based on adjusted p-values with the FDR cutoff (specified in the option `sig`) and/or with the fold change cutoff (specified in the option `FCcutoff`). Color gold represents proteins are not significantly different in abundance. The color scheme shows the evidences of significance. The darker color it is, the stronger evidence of significance it has. The color key shows the cutoff of adjusted p-value for color scheme. Left Heatmap is applied with FDR cutoff is 0.05 and no fold change cutoff. Right Heatmap is applied with both FDR cutoff is 0.05 and FC cutoff is 70. Note that specifying the FC cutoff = 70 is for demonstration purpose, so that protein IDHC becomes insignificant after both FDR and fold change cutoff.

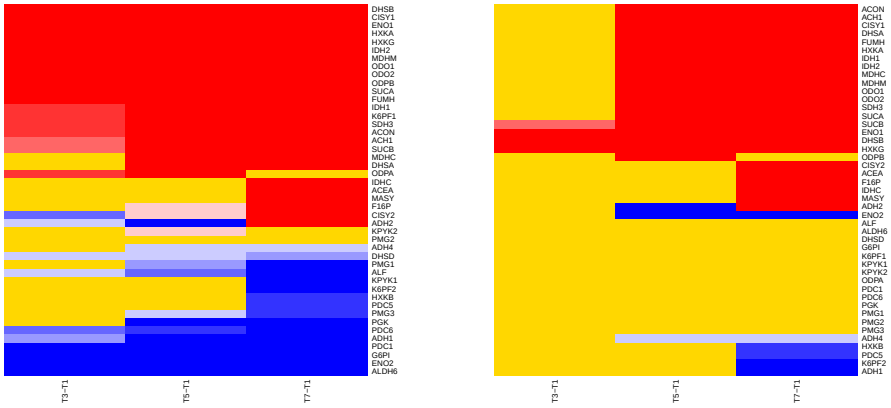


Figure 12: Heatmap of multiple comparisons for the 45 proteins from the example time courses yeast study with all 45 proteins, which is the same result of Figure 10. Left heatmp is the result for default option using FDR cutoff=0.05 only. Right heatmap is the result under both FDR cutoff=0.05 and FCcutoff=1.5

(3) Comparison plot

```
> groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult,
+                      type="ComparisonPlot")
```

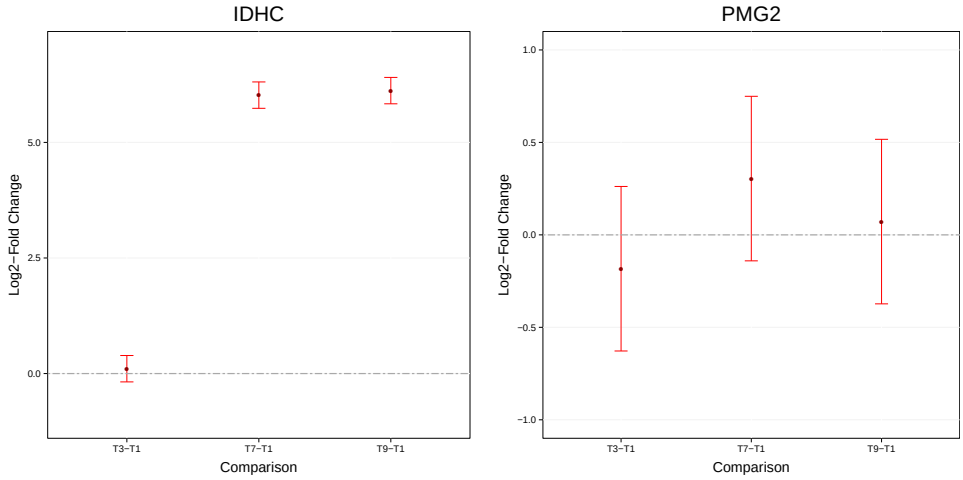


Figure 13: Comparison plots for multiple comparisons. The x-axis is the comparisons which the user designed. The y-axis is the log fold change for each comparison. The red points mean the estimated log fold change. The blue error bars mean the confidence interval with 0.95 significant level for log fold change. The horizontal line at log fold change=0 (the log fold change of one comparison is located around this line) means that there is no significant difference for this comparison.

3.5 Sample size calculation for a future experiment

To calculate sample size for future experiments, the function fits the model and uses variance components. The underlying model fitting with intensity-based linear model with technical MS run replication. Input in the example is `QuantData`. Estimated sample size is rounded to 0 decimal. Four options of the calculation:

1. number of biological replicates per condition
2. number of peptides per protein
3. number of transitions per peptide
4. power is a pre-specified statistical power which defined as the probability of detecting a true fold change. Users should input the average of power that is expected. Otherwise, the function calculate the power for this category,

It can only obtain either one of the categories of the sample size calculation (numSample, numPep, numTran) or power calculation at the same time. Other three values are need to be inputed.

To get started with this function, visit the help section of designSampleSize first by the following code.

```
> ?designSampleSize
```

`desiredFC`, the range of a desired fold change which includes the lower and upper values of the desired fold change, and FDR, a pre-specified false discovery ratio (FDR) to control the overall false positive, are required. Two options for fitting model are possible, choice of scope of biological replication and choice of interference data. For scope of biological replication, "restricted"(default) represents restricted scope of biological replication to the selected individuals. "expanded" represents expanded scope of biological replication to the whole population. For interference, TRUE(default) means data contain interference transitions and need additional model interaction to address the interference. FALSE means data contain no interference transitions and no need additional model interaction to address the interference.

3.5.1 Minimal number of biological replicates per condition

```
> designSampleSize(data=QuantData,numSample=TRUE,numPep=3,numTran=4,power=0.8,
+                  desiredFC=c(1.25,1.75),FDR=0.05)

   desiredFC numSample numPep numTran  FDR power    CV
1      1.250        15      3       4 0.05   0.8 0.004
2      1.275        12      3       4 0.05   0.8 0.005
3      1.300        11      3       4 0.05   0.8 0.006
4      1.325         9      3       4 0.05   0.8 0.007
5      1.350         8      3       4 0.05   0.8 0.007
6      1.375         7      3       4 0.05   0.8 0.008
7      1.400         6      3       4 0.05   0.8 0.009
8      1.425         6      3       4 0.05   0.8 0.009
9      1.450         5      3       4 0.05   0.8 0.011
10     1.475         5      3       4 0.05   0.8 0.011
```

```
11     1.500         4      3       4 0.05   0.8 0.013
12     1.525         4      3       4 0.05   0.8 0.013
13     1.550         4      3       4 0.05   0.8 0.013
14     1.575         4      3       4 0.05   0.8 0.013
15     1.600         3      3       4 0.05   0.8 0.017
16     1.625         3      3       4 0.05   0.8 0.016
17     1.650         3      3       4 0.05   0.8 0.016
18     1.675         3      3       4 0.05   0.8 0.016
19     1.700         3      3       4 0.05   0.8 0.016
20     1.725         2      3       4 0.05   0.8 0.023
21     1.750         2      3       4 0.05   0.8 0.023
```

3.5.2 Power calculation

```
> designSampleSize(data=QuantData,numSample=2,numPep=3,numTran=4,power=TRUE,
+                  desiredFC=c(1.25,1.75),FDR=0.05)

   desiredFC numSample numPep numTran  FDR power
1      1.250         2      3       4 0.05  0.01
2      1.275         2      3       4 0.05  0.01
3      1.300         2      3       4 0.05  0.01
4      1.325         2      3       4 0.05  0.01
5      1.350         2      3       4 0.05  0.01
6      1.375         2      3       4 0.05  0.02
7      1.400         2      3       4 0.05  0.03
8      1.425         2      3       4 0.05  0.05
9      1.450         2      3       4 0.05  0.08
10     1.475         2      3       4 0.05  0.12
11     1.500         2      3       4 0.05  0.16
12     1.525         2      3       4 0.05  0.21
13     1.550         2      3       4 0.05  0.26
14     1.575         2      3       4 0.05  0.32
15     1.600         2      3       4 0.05  0.38
16     1.625         2      3       4 0.05  0.43
17     1.650         2      3       4 0.05  0.49
18     1.675         2      3       4 0.05  0.54
19     1.700         2      3       4 0.05  0.59
20     1.725         2      3       4 0.05  0.64
21     1.750         2      3       4 0.05  0.69
```

3.5.3 Visualization for sample size calculation

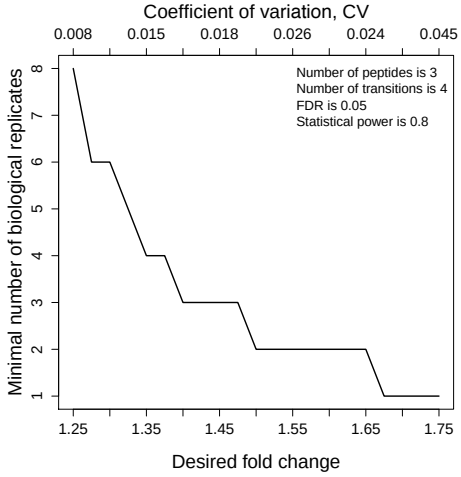
To illustrate the relationship of desired fold change and the calculated minimal number sample size which are (1) number of biological replicates per condition, (2) number of peptides per protein, (3) number of transitions per peptide, and (4) power. The input is the result from function (`designSampleSize`).

To get started with this function, visit the help section of designSampleSizePlots first by the following code.

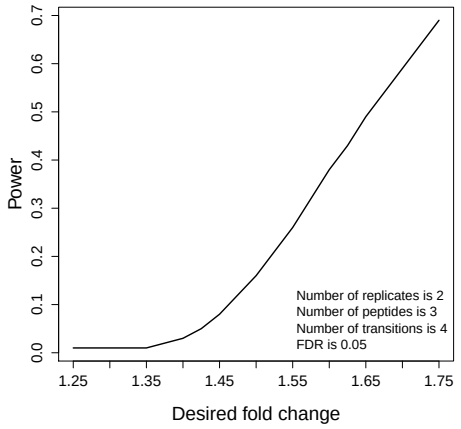
```
> ?designSampleSizePlots
```

```
> # Minimal number of biological replicates per condition
> result.sample<-designSampleSize(data=QuantData,numSample=TRUE,numPep=3,numTran=4,power=0.8,
```

```
+                                    desiredFC=c(1.25,1.75),FDR=0.05)
> designSampleSizePlots(data=result.sample)
```




```
> # Power
> result.power<-designSampleSize(data=QuantData,numSample=2,numPep=3,numTran=4,power=TRUE,
+                                desiredFC=c(1.25,1.75),FDR=0.05)
> designSampleSizePlots(data=result.power)
```



3.6 Quantification of protein abundance in individual samples or in conditions

Downstream analysis, e.g., clustering or classification of individual samples based on protein profiles, sometimes requires a single value of a protein in each biological replicate. That is, it is sometimes of interest to have a table where columns are biological replicates and rows are proteins. A summarization step is needed to express the measurements from multiple features into a single value per protein per biological replicate. As in testing for differential abundance, this can be done by using model-based quantities.

In MSstats , the quantification function can be used to produce the estimates of protein abundance for all biological replicates in the study or for all groups. For sample quantification, the label of each biological sample is a combination of the corresponding group and the sample ID. The same model with groupComparison will be used. However, if there is only one transition in a certain protein, the estimate of variation is NA. Therefore, the result may be unreliable. The quantification for light is the endogenous transition quantification.The quantification for reference (heavy) is the average among all reference intensities. The quantification of ratio between endogenous and reference intensity would be the quantification of light minus the reference quantification. Two formats of output are supported. "long" for long format which has the columns named Protein, Condition, LonIntensities (and BioReplicate if sample quantification). "matrix" for data matrix format which has the rows for Protein and the columns, which are Groups(or Conditions) for group quantification or the combinations of BioReplicate and Condition (labeled by "BioReplicate"_"Condition") for sample quantification. Default is "matrix".

The input of this function is the quantitative data from function (dataProcess). The example data is QuantData.

To get started with this function, visit the help section of quantification first by the following code.

```
> ?quantification
```

Consider quantitative data (i.e. QuantData) from a yeast study with ten time points of interests, three biological replicates, and no technical replicates which is a time-course experiment. Sample quantification shows model-based estimation of protein abundance in each biological replicate within each time point. Group quantification shows model-based estimation of protein abundance in each time point.

```
> #  (1): Sample quantification
> subQuant<-quantification(QuantData)
> head(subQuant)
```

```
       ReplA_1  ReplA_2   ReplA_3   ReplA_4  ReplA_5  ReplA_6  ReplA_7  ReplA_8
IDHC  6.482258 7.144148  6.753797  6.891645  7.51246 9.955367 12.76964 12.94651
PMG2 10.392707 9.959031 10.078554 10.520689 10.27148 9.909567 10.39021 10.14813
      ReplA_9 ReplA_10   ReplB_1  ReplB_2   ReplB_3  ReplB_4   ReplB_5  ReplB_6
IDHC 12.81644 12.94724  6.959746  6.84807  6.586199  6.86521  7.096384  9.86911
PMG2 10.49330 10.26594 10.400018 10.72289 10.514320 10.23663 10.388234 10.38962
      ReplB_7  ReplB_8  ReplB_9 ReplB_10   ReplC_1   ReplC_2   ReplC_3
IDHC 12.77337 12.95800 12.88595 12.95984  6.916612  7.065317  7.154622
PMG2 10.55574 10.03953 10.28494 10.13845 10.433715 10.598824 10.034071
       ReplC_4   ReplC_5   ReplC_6  ReplC_7  ReplC_8  ReplC_9 ReplC_10
IDHC  7.411565  6.554255  9.884292 12.74810 12.96304 12.82765 12.90613
PMG2 10.779834 10.287874 10.334494 11.15592 10.11160 10.21011 10.17977
          Ref
IDHC 12.884638
PMG2  9.107043
```

```
> # (2): Group quantification
> groupQuant<-quantification(QuantData, type="Group")
> head(groupQuant)
```

```
             1         2        3        4         5         6        7
IDHC  6.786205  7.019178  6.83154  7.05614  7.054366  9.902923 12.76371
PMG2 10.408813 10.426913 10.20898 10.51238 10.315861 10.211228 10.70063
            8        9       10       Ref
IDHC 12.95585 12.84335 12.93774 12.884638
PMG2 10.09976 10.32945 10.19472  9.107043
```

4 Example workflow with label-free LC-MS: controlled spike-in experiment

The workflow is the same as SRM experiment except some options for analysis and inference. The summary of result is shown below. See details from *user tutorial* in msstats.org

4.1 Experimental design

?, has previously described the experiment with the sample with 6 standard Proteins, (horse myoglobin, bovine carbonic anhydrase, horse Cytochrome c, chicken lysozyme, yeast alcohol dehydrogenase, rabbit aldolase A) Each protein was represented by 7-21 peptides and each peptide was represented by 1-5 transition. Using latin square design of experiment frame, each sample has different dilution for each protein. Each of the LC-MS runs is analyzed in 3 technical replicate. There are total 18 injection runs. The data set is available in website (http://prottools.ethz.ch/muellelu/web/Latin_Square_Data.php) and in msstats.org.

4.2 Pre-processing data and quality control of MS runs

The default normalization method for label-free MS experiment in MSstats is the constant normalization with all endogenous intensities along all proteins. However, if there are small number of features in each run, compared with SRM and DIA experiments, we can not keep the assumption that the distribution of intensities across run is the same. The alternative could be normalization using internal standard peptide. See details in section 3.3.

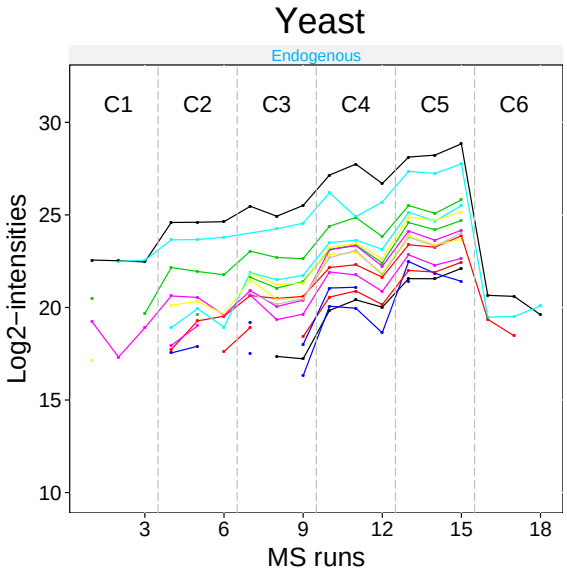


Figure 14: Profile plot for one spiked protein, Alcohol dehydrogenase-Yeast. The profiles show some non-parallel patterns (i.e. interferences). Features with low signals have missing values.

4.3 Model-based inference

LC-MS experiments are often subject to random interferences, and contain missing values. To express the fact that the interferences are non-systematic, we recommend to remove interaction (`interference`=FALSE). To express the fact that the variances are unequal between the features, we recommend to use feature-specific variance (`featureVar`=TRUE). Also there are three options to handle missing values, (1) remove interaction which means to assume features demonstrate no interference across runs, (2) impute with average minimum intensity across runs, (3) remove features that completely miss in some conditions.

4.3.1 Setting up a linear mixed effects model

First, we assign the comparison of interest. Next,we fit a model that specifies systematic interferences and constant variance. Since this is a label-free experiment, use `labeled`=FALSE. See section 3.4.1 for details regarding specifying scope of conclusions.

4.3.2 Verifying the assumption of the model

Also, in this case the protein has many missing peaks (complete missing in some conditions).The groupComparison function automatically detects this pattern of missing values, and uses the model without interaction. See in section 3.4.2 for other possible options.

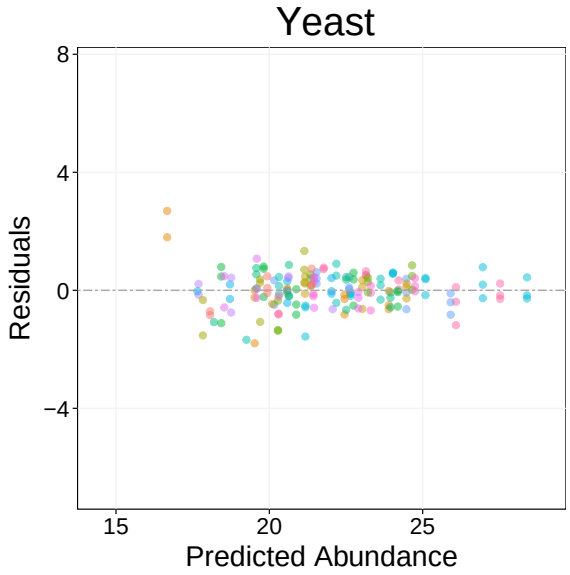


Figure 15: Residual plot for one spiked protein, Alcohol dehydrogenase-Yeast. The profiles shows a larger variance for features with a lower intensity.

4.4 Testing for protein-level differential abundance between conditions

Heatmap summarizes all results of comparison, that `MSstats` detects difference between groups. See details about other types of plot in section 3.5.

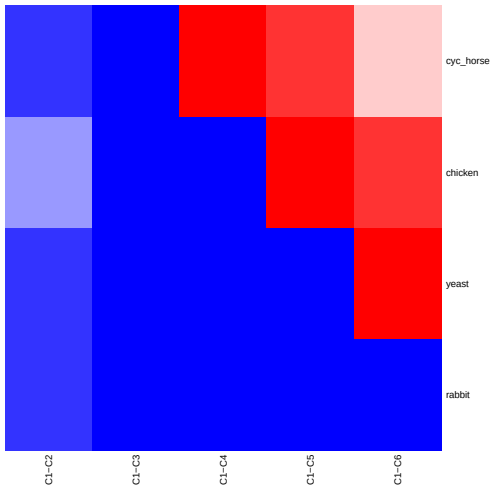


Figure 16: Heatmap of significance testing results of 4 spiked proteins across 6 conditions.

4.5 Sample size calculation for a future experiment

Same as in SRM experiments. See details in section 3.5.

4.6 Quantification of protein abundance in individual samples or in conditions

Same as in SRM experiments. See details in section 3.6.

5 Example workflow with label-free DIA: a group comparison study of *S.Pyogenes*

The workflow is the same as SRM experiment except some options for analysis and inference. The summary of result is shown below. See details from *user tutorial* in msstats.org

5.1 Experimental design

The example dataset is from a label-free DIA experiment of S.pyogenes study. For demonstration purposes, this partial data set consists of two proteins only. Two biological conditions are Strep0 and Strep10 (S. Pyogenes with 0% and 10% of human plasma added). Two biological replicates from each condition were profile with a SWATH-MS-enabled AB SCIEX TripleTOF 5600 System. The identification and quantification of spectral peaks was assisted by a spectral library, and was performed using OpenSWATH software http://proteomics.ethz.ch/openswath.html.

5.2 Pre-processing data and quality control of MS runs

The default normalization method for label-free MS experiment in MSstats is the constant normalization with all endogenous intensities along all proteins. In case of DIA, there are enough number of intensities in each run. Therefore, quantile normalization is also suitable because the assumption for same distribution of intensities for endogenous transition across all MS runs is reasonable. See details in section 4.3.

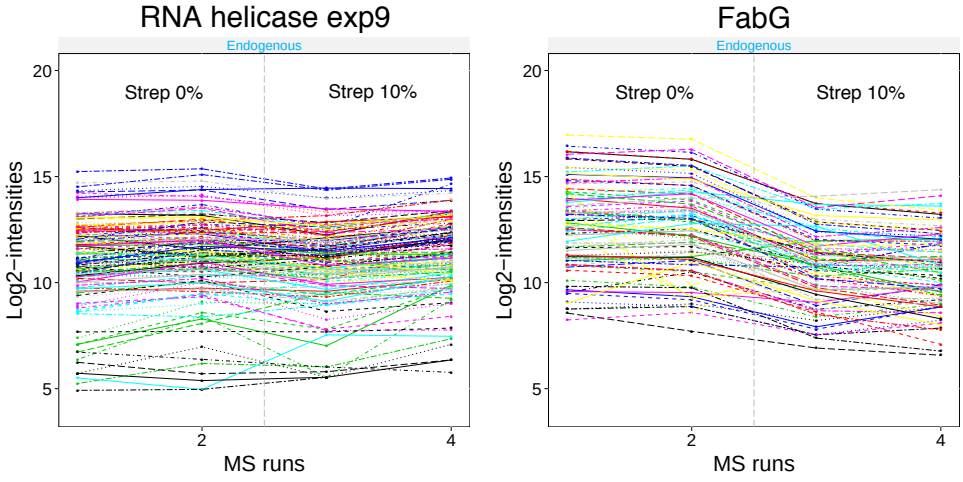


Figure 17: Profile plots for two example proteins, Probable RNA helicase exp9 and FabG in DIA experiment. The proteins have a large number of features. The profiles show some non-parallel patterns (i.e. interferences).

5.3 Model-based inference

DIA experiments are often subject to random interferences, and unequal variance between features. To express the fact that the interferences are non-systematic, we recommend to remove interaction (`interference`=FALSE). To express the fact that the variances are unequal between the features, we recommend to use feature-specific variance (`featureVar`=TRUE).

5.3.1 Setting up a linear mixed effects model

First, we assign the comparison of interest. Next,we fit a model that specifies systematic interferences and constant variance. Since this is a label-free experiment, use `labeled`=FALSE. See section 3.4.1 for details regarding specifying scope of conclusions.

5.3.2 Verifying the assumption of the model

The unequal variance between features is detected from Residual plot and normal quantile-quantile plots.(Figure 18) Therefore to consider the unequal variance is recommended as below. See in section 3.4.2 for other possible options.

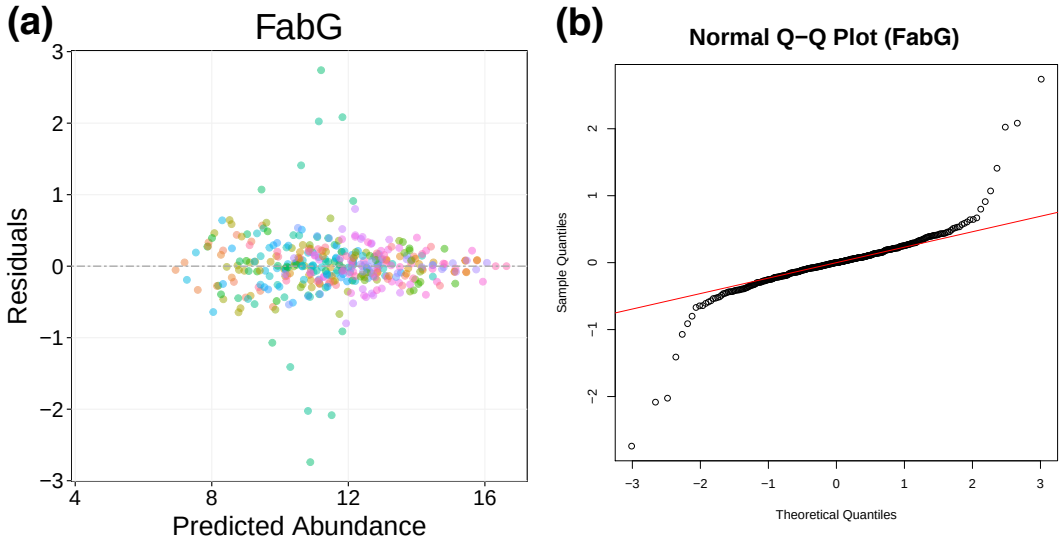


Figure 18: (a) Residual plot (left panel) for FabG protein using the model under the assumption constant variance across features. (b) normal quantile-quantile plot (right panel) shows the pattern in the qqplot that is due to deviations from the assumption of constant variance, and not necessarily from the assumption of Normality.

5.3.3 Testing for protein-level differential abundance between conditions

Based on detection about interference in Figure 17 and unequal variance in Figure 18, we consider them using option `featureVar`=TRUE,`interference`=FALSE as below.

See in section 3.4.3 for other options and visualizations.

5.4 Sample size calculation for a future experiment

Same as in SRM experiments. See details in section 3.5.

5.5 Quantification of protein abundance in individual samples or in conditions

Same as in SRM experiments. See details in section 3.6.