Label Free Quantitive (LFQ) of nanoLC ESI DDI-MS data using MaxQuant

16 January, 2019

Contents

General information

- **Project number** :
- **Experiment number**:
- **Project tier**: Collaboration
- **Client**:
- **Reported by**:

Samples

- **Number of samples**:
- **Sample details**:
- **Goal of experiment**:

Sample Preparation

- **Sample prep SOP**:
- **Prepared by**:
- **Protease**:
- **iRT peptide**:

Mass Spectrometry acquisition

- **Nano LC System**:
- **Mass spectrometer**:
- **Analytical column**:
- **Trap column**:
- **Acquisition method**:
- **Analysis date**:
- **Analysis time (in hours)**:

Data analysis

- **LFQ generation**: MaxQuant vXXX
- **Search engine**: Andromeda (implemented in MaxQuant)
- **Data base**: RS Mouse Swissprot iRT
- **Protein FDR cutoff**: 1%
- **Fixed modification**: Carbamidomethylation
- **Variable modification**: Oxidation @ M Acetylation @ Protein N-terminus
- **Comments**: None
- **Statistical analysis**: In-house generated script

Method details

The raw data files were analyzed using MaxQuant to obtain protein identifications and their respective label-free quantification values using in-house standard parameters. Of note, the data were normalization based on the assumption that the majority of proteins do not change between the different conditions. Statistical analysis was performed using an in-house generated R script based on the ProteinGroup.txt file. First, contaminant proteins, reverse sequences and proteins identified "only by site" were filtered out. In addition, proteins that have been only identified by a single peptide and proteins not identified/quantified consistantly in same condition have been removed as well. The LFQ data was converted to log2 scale, samples were grouped by conditions and missing values were imputed using the 'Missing not At Random' (MNAR) method, which uses random draws from a left-shifted Gaussian distribution of 1.8 StDev (standard deviation) apart with a width of 0.3. Protein-wise linear models combined with empirical Bayes statistics were used for the differential expression analyses. The *limma* package from R Bioconductor was used to generate a list of differentially expressed proteins for each pair-wise comparison. A cutoff of the *adjusted p-value* of 0.05 (Benjamini-Hochberg method) along with a |log2 fold change| of 1 has been applied to determine significantly regulated proteins in each pairwise comparison.

Quick summary of parameters used:

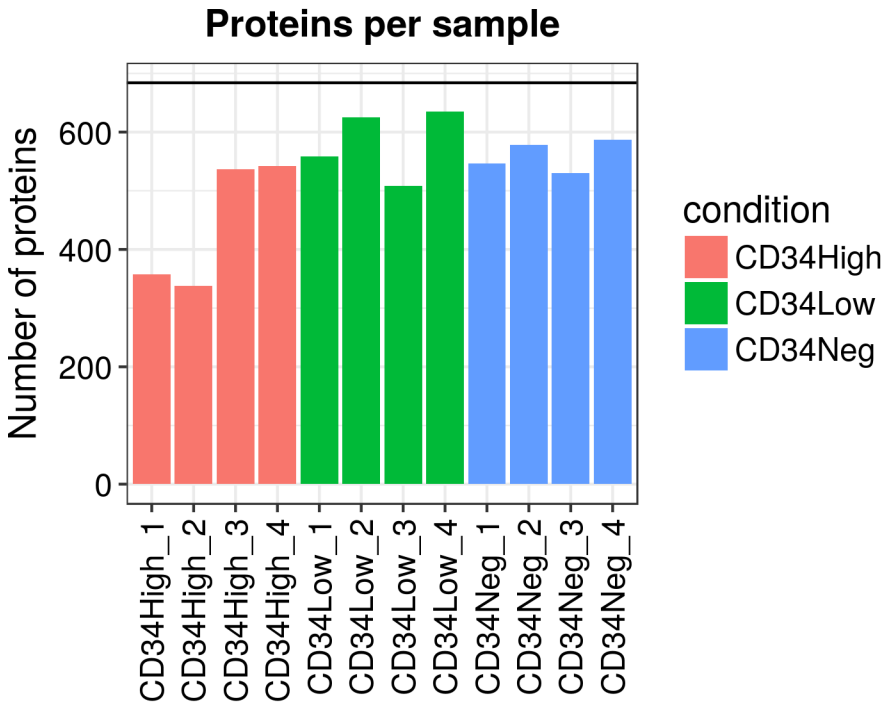
- Tested pairwise comparisons = CD34High_vs_CD34Low, CD34High_vs_CD34Neg, CD34Low_vs_CD34Neg
- Adjusted *p-value* cutoff <= 0.05
- Log fold change cutoff >= 1

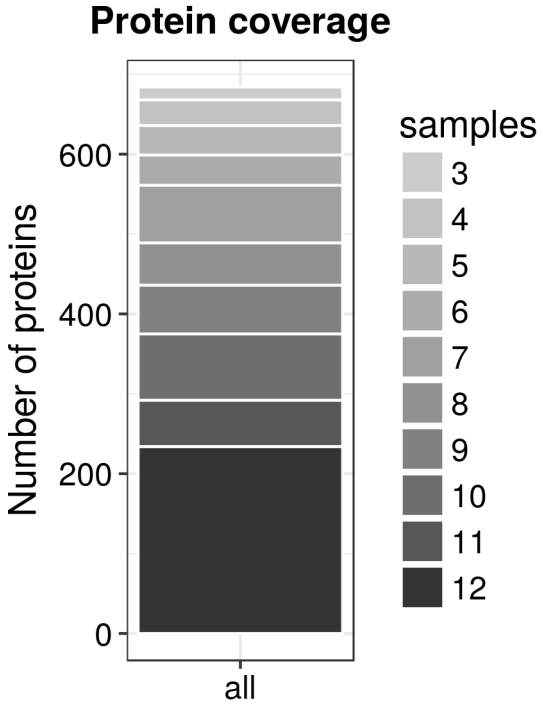
Results

MaxQuant result output contains proteins groups of which *684* proteins were reproducibly quantified.

130 proteins differ significantly between samples.

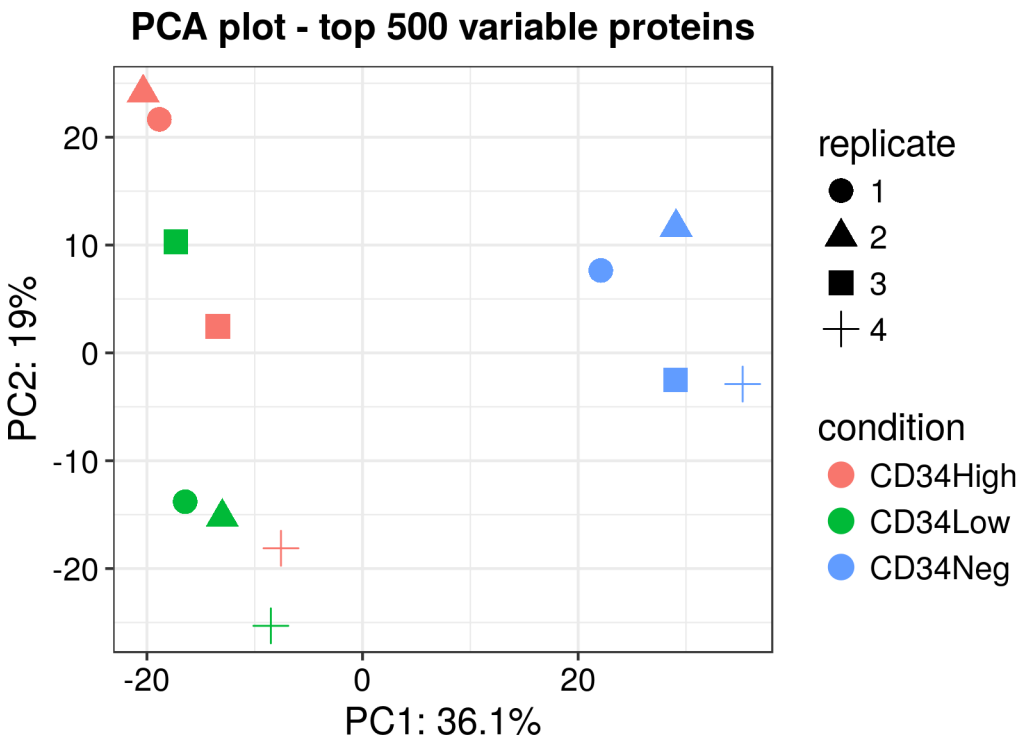
Number of identified proteins



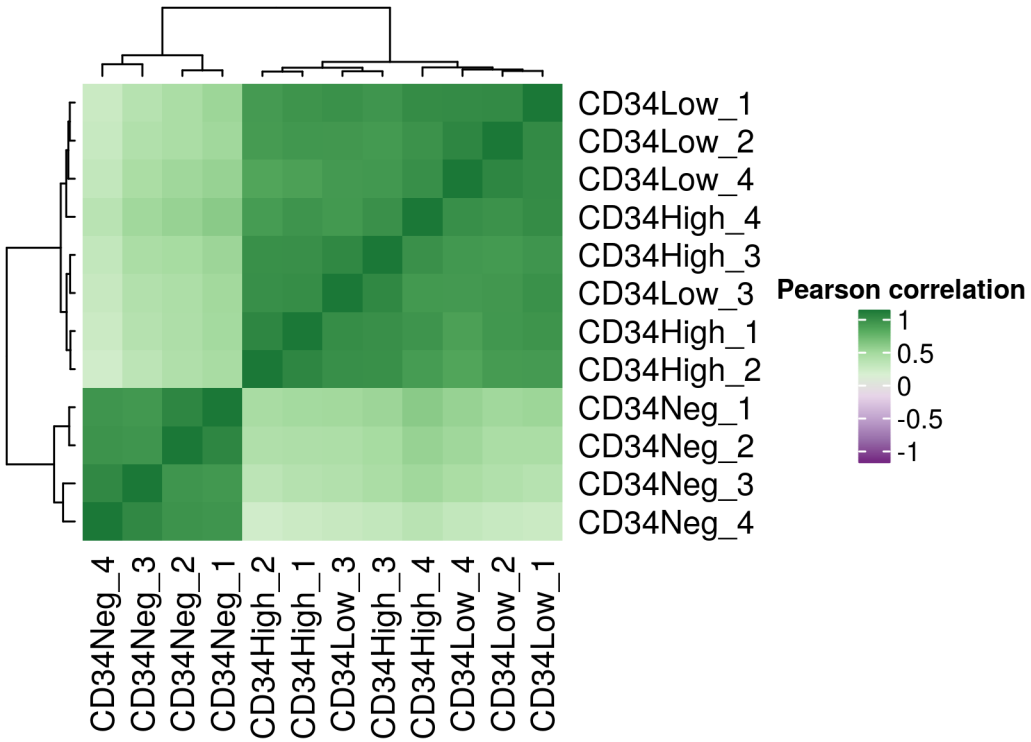


Exploratory Analysis

PCA plot



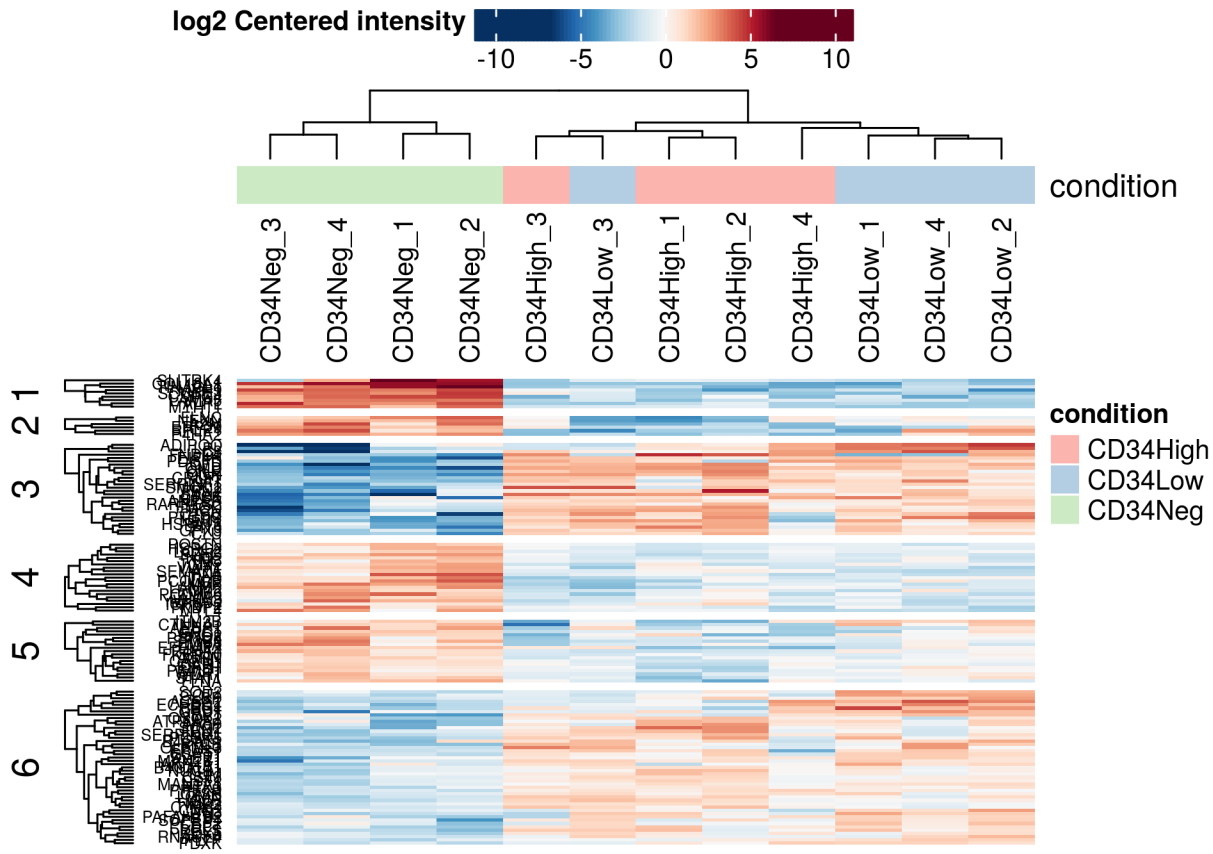
Sample Correlation matrix



Differential Expression Analysis

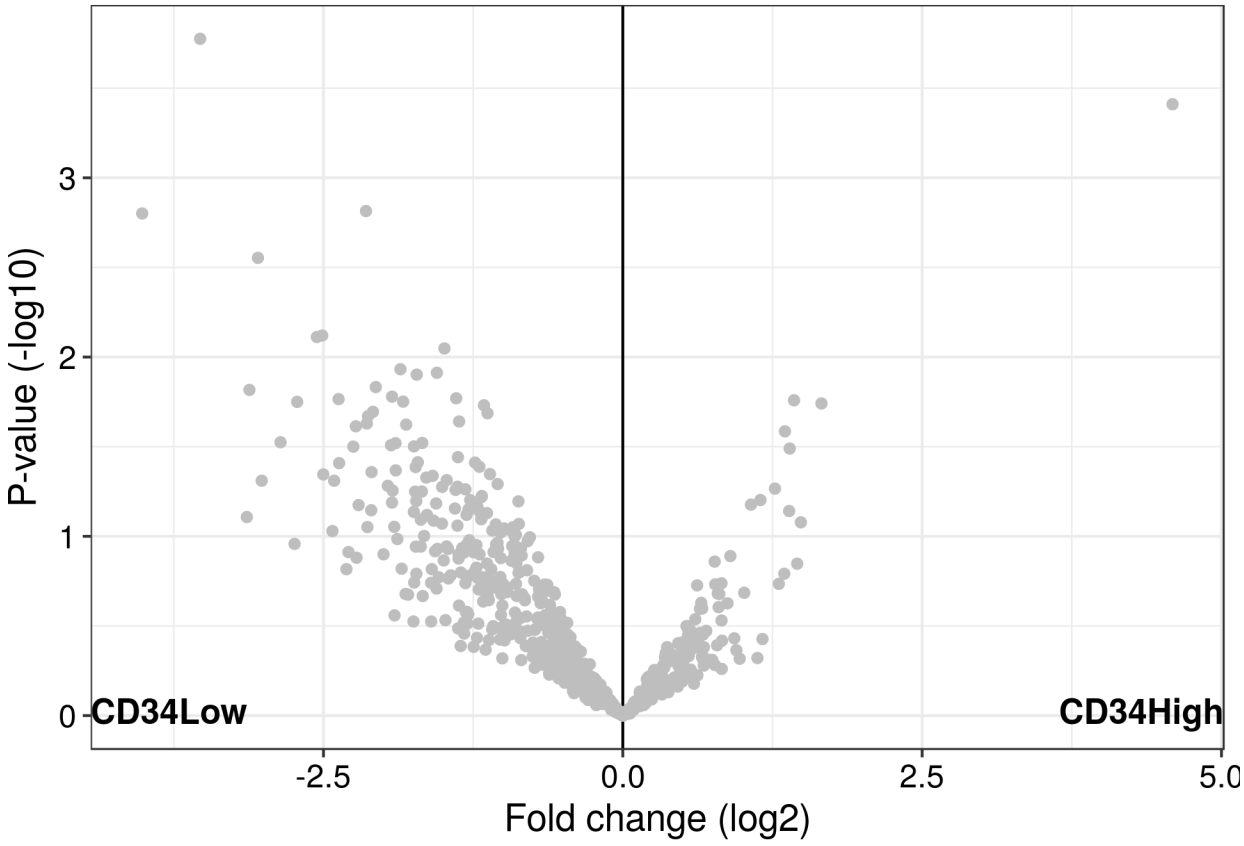
Heatmap

A plot representing an overview of all significant proteins (rows) in all samples (columns).

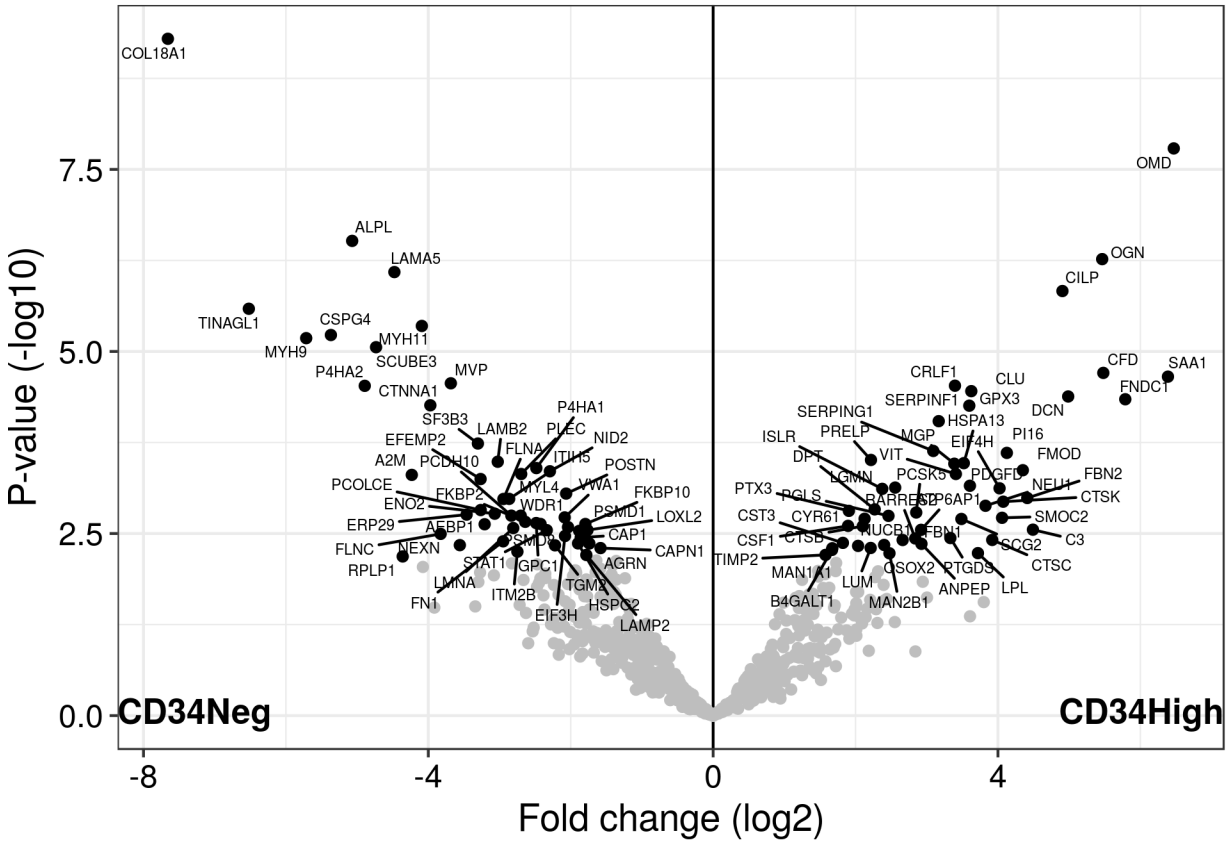


Volcano Plots

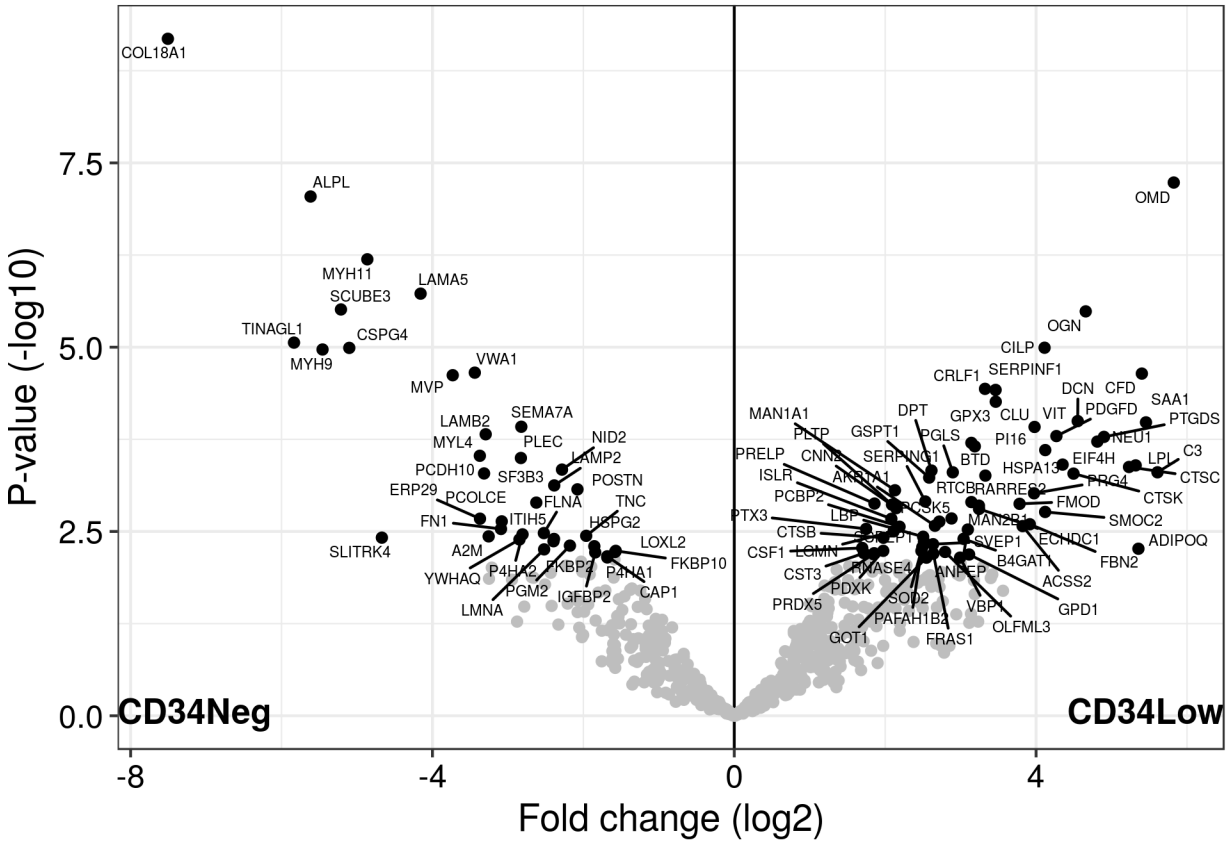
```
## [1] "volcano_plot_CD34High_vs_CD34Low"
```




```
## [1] "volcano_plot_CD34High_vs_CD34Neg"
```




```
## [1] "volcano_plot_CD34Low_vs_CD34Neg"
```



FAQ

I am overwhelmed. How do I identify interesting candidate proteins?

In most cases, interesting candidate proteins are those proteins that are significantly up- or down-regulated between 2 conditions. In the volcano blot these proteins are located in the left and right upper quadrant.

Why do we need filter on missing values?

The dataset contains proteins which are not quantified in all replicates. Some proteins are even only quantified in a single replicate.

This leaves our dataset with missing values, which need to be imputed (see below). However, this should not be done for proteins that contain too many missing values. Therefore, we filtered out proteins that contain too many missing values.

What exactly means that missing values have been imputed? And why do I get 2 data matrices (before and after imputation)?

If a peptide (and thus protein) has been identified in a certain sample, MaxQuant tries to find the corresponding mass peak in the other samples and if a corresponding peak can be found, MaxQuant provides a quantitative value for this peak (basically the area under the curve). However, if MaxQuant cannot find the corresponding peak, it will return a **zero** as quantitative value. Of course, this **might** mean that the peptide (and protein) is absent in this sample (and those proteins are often the most important proteins!!). However, it **can also mean** that the peak is not quantifiable (for example due to isobaric contaminants or abnormal peak shapes

etc...). **So, the most important thing to understand here is that a LFQ value of zero does not necessarily mean that the peptide/protein is absent!!**

All LFQ values of zero are defined as missing values. The problem is now that the log2 of zero is not defined, so after calculating the log2 of all LFQ values (which is a default step during data analysis), all zeros will result in an "N/A" value. And it is of course impossible to do a Student's t-test (or nearly any statistical test) if the matrix contains N/A values.

A common way of dealing with this problem is to impute the missing values based on a normal distribution. In lay terms, a histogram distribution of all peptide intensities in a sample is calculated and the N/A values are replaced with real values located at the lower end of this distribution. This procedure has its pros and cons, which is why we provide both the unimputed matrix (containing N/A values) and the imputed matrix (where the N/A values are replaced based on a normal intensity distribution)

What are the pros of imputation?
* We are able to provide statistical information of every protein identified across the samples.

What are the cons?
* Values are basically made up, i.e. a protein, which is actually absent in a sample, will get suddenly a real quantitative value.

How does this influence the results?
* The results will not be hugely different for proteins that are reasonably abundant. The worst thing that happens to those proteins is that the fold-change gets (slightly) altered.

- However, the results can and will change quite dramatically for proteins that are low abundant.
- An example: let's assume we have 2 conditions (WT and mutant) with 3 replicates each and a protein has been identified with 3 "real values" in the WT samples and 3x N/A values in the mutant sample. And let's assume that the normal intensity distribution has its lowest value around 20 and the highest value around 30 (most peptides center around 25).
- **If the 3 "real values" are relatively high (let's say 25)**, the fold-change based on the unimputed matrix is very high (basically it is infinite: present in WT vs absent in the mutant). Even if we impute and the 3x N/A values get a low value assigned to it (let's say 20), the protein is still up-regulated in the WT samples.
- **However, if the 3 "real values" are relatively low (let's say 20.5)**, the fold-change based on the unimputed matrix is still very high (still infinite: present in WT vs absent in the mutant). But if we impute now and the 3x N/A values get values around 20 assigned to them, the protein is suddenly hardly mis-regulated anymore (despite the fact that it might be even absent in the mutant sample).

This is the reason why we provide both the unimputed as well as imputed data matrix. See 8.1 how to deal with both matrices.

Which quantitative value is directly (and linearly) proportional to the actual protein abundance? Is it the log2-transformed intensity or the untransformed intensity?

The untransformed intensities are directly proportional to the actual protein abundance. The log2 "fold-changes (FC)" that are given by the "Student's T-test Differences" columns are mathematically defined as:

Difference $= log2FC = log2(intensity(condition1)/intensity(condition2)) =$

$log2(intensity(condition1)) - log2(intensity(condition2))$

Assuming I have identified candidate proteins, what should I do next?

This depends on mostly you, but LFQ should be considered as a "first hint" of what is going on between the different conditions. MBPF recommends doing the following:

- If you are only interested in a few candidate proteins (for example after pull-down experiments), try to verify the results through different approaches (for example Western blotting if you have an antibody etc) or repeat the LFQ experiment to further confirm the validity of the results. More advanced targeted mass spectrometric methods such as parallel or multiple reaction monitoring (PRM or MRM) can also be used to further verify LFQ results.
- If you are interested in a more global approach, i.e. if you are trying to understand the global changes on a systems biology level, we recommend conducting pathway analyses to identify pathways and/or groups of proteins involved in distinct biochemical mechanisms.

Detailed explanation of the attached files

Principal component analysis (PCA)

A PCA is a technique used to emphasize variation and bring out strong patterns in a dataset. In brief, the more similar 2 samples are, the closer they cluster together. Of course, this means that biological replicates (and in particular technical replicates) should cluster tightly together. For further information, here are a few links, which explain the principals of PCAs:

http://ordination.okstate.edu/PCA.htm

http://setosa.io/ev/principal-component-analysis/

Sample Correlation plot

A correlation matrix is plotted as a heatmap and visualize the Pearson correlation coefficients between the different samples.

Pairwise correlation plots

Another way of visualizing how samples are related to each other (i.e among replicates or among condition) is a scatterplot. The plot shows pair-wise comparison of all samples. The lower half represents a scatterplot of protein intensities in two samples (gray points) and the blue line is a *loess* regression line. The diagonal line shows the distribution of (log2) protein intensitites in the given sample and the upper part of matrix shows the actual correlation values.

Heatmap

The heatmap representation gives an overview of all significant proteins (rows) in all samples (columns). This visualization allows the identfication of general trends such as if one sample or replicate is highly different compared to the others and might be considerd as an outlier. Additionally, the hierarchical clustering of samples (columns) indicates how related the different samples are and hierarchicalclustering of proteins (rows) identifies similarly behaving proteins.

Volcano Plot (for each comparison one Plot)

A volcano blot is merely a graphical visualization by plotting the "Fold Change (Log2)" on the x-axis versus the –log10 of the "adjusted *p-value*" on the y-axis. Interesting candidate proteins are located in the left and right upper quadrant.