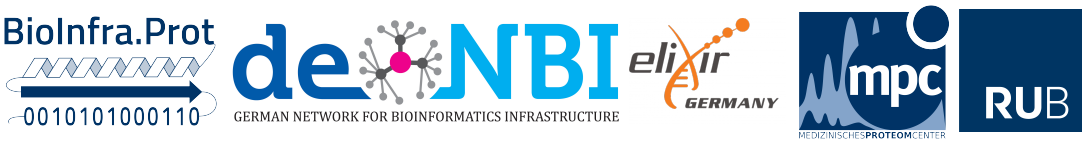
Advanced analysis of quantitative proteomics data using R

Exercises

Michael Turewicz, Karin Schork

November 9, 2020



Contents

1 Hands-on session I: R packages and data preprocessing

General hints:

- look at the slides to solve the exercises
- write down your code in an editor (e.g. the upper left window of RStudio)
- also look at the help pages of the mentioned R functions

Exercise 1.1: Installing and using R packages

Please start your R console and set your current R repositories to "CRAN" and "BioC software". Then, choose a location near your current whereabouts as your CRAN as well as your Bioconductor mirror. Install the following packages, that you will need in the following exercises:

- openxlsx (for reading and writing xlsx files)
- limma (for normalization)
- affy (for MA-Plots)
- gplots (for heatmap)
- pROC (for ROC curves)

Please check whether you have correctly installed the packages by loading them and calling their help pages.

Exercise 1.2: Missing values

In the following hands-on sessions we will use a dataset containing 19 samples of Hepatocellular Carcinoma (HCC) and 19 samples of corresponding nontumorous tissue. For more information on the study see

Wael Naboulsi et al. Quantitative Tissue Proteomics Analysis Reveals Versican as Potential Biomarker for Early-Stage Hepatocellular Carcinoma Journal of Proteome Research 2016 15 (1), 38-47 DOI: 10.1021/acs.jproteome.5b00420

Import the dataset `HCC_19vs19_raw_abundances.xlsx` (unnormalized data) into R using the function `read.xlsx` from the R package `openxlsx`. Alternatively you can import the csv file.

Write the first 9 columns from the data set (containing protein accessions and other information) in a different dataframe for later use and delete them from the original dataset. Overwrite all zero values in the remaining dataframe with `NA`.

Answer the following questions:

- How many missing values are there in the dataset in total?
- How many proteins have no missing values? Create a barplot showing how many proteins exists with different numbers of missing values (hint: use the function `table()`).
- Which sample has the most missing values and how many? Create a barplot showing the number of missing values for each sample.

Hint: the funtions `rowSums()` and `colSums()` may be useful.

Exercise 1.3: Normalization

Use the function `normalizeBetweenArrays` from the `limma` package to normalize the dataset from Exercise 1.2. Apply a log2-transformation before normalizing. Create three new datasets, containing the median, quantile and LOESS normalized data.

Compare the three normalization methods with the non-normalized data by creating boxplots and MA-plots. For the MA-plots, use the function `MAPlots` given in the file `MA_Plots.R` (the R package `affy` needs to be installed). The file for this can be found in the exercise folder of the material. Look at the script for information the arguments.

As the normalization was conducted on log2-transformed data, the argument `log` of the function must be set to `FALSE`.

Which is the most suitable normalization method for this dataset?

2 Hands-on session II: Clustering

Exercise 2.1: Clustering

Perform hierarchical clustering on the quantile normalized dataset using the function `hclust`. Compare different combinations of the distances and linkage methods by looking at the corresponding dendrograms. How does only using the 20 most differential proteins (by means of p-value) effect the clustering results?

As linkage methods test `"average"`, `"complete"` and `"single"`. As distances compare `"euclidean"`, `"manhattan"` and your self-defined correlation-based distance. For an arbitrary dataset x latter can be defined as follows:

```
> as.dist((1-cor(x, use = "pairwise.complete.obs"))/2)
```

Hint: Read the help pages of the functions `dist`, `as.dist` and `hclust`! Don't forget to check whether you have to transpose your data matrix first (using the function `t`)!

Exercise 3.2: Heatmaps

Draw a heatmap for the given dataset that has been quantile normalized using the function `heatmap.2` from the package `gplots`. Remove proteins with missing values from the dataset. Your heatmap shall include both row and column dendrograms as

well as a color key. Optimize the arguments trace, scale, cexRow and cexCol. Is there any effect when instead of all proteins only the 20 most differential hits (by means of p-value) are used?

3 Hands-on session III: PCA and ROC

Exercise 3.1: PCA

Compute a principal component analysis on the quantile normalized data set from Exercise 1.2.

To do this, first delete all rows (proteins) containing missing values using `na.omit()`. Second, transpose the dataset using `t()`, as the variables (proteins) need to be in columns and observations (samples) in rows. Then, apply `prcomp` to the dataset.

Create a plot of the first two principal components.

Exercise 3.2: ROC

In the paper corresponding to the dataset amongst others the following proteins were found significantly regulated between Hepatocellular carcinoma and healthy tissue.

- ATP-dependent RNA helicase (DDX39) (O00148)
- Fibulin-5 (FBLN5) (Q9UBX5)
- Myristoylated alanine-rich C-kinase substrate (MARCKS) (P29966)
- Serpin H1 (SERPINH1) (P50454)

Create ROC-curves for these proteins, showing the best threshold and the AUC on the normalized dataset.

4 Hands-on session IV: Writing own R functions

Exercise 4.1: Simple function

Write a short R function that converts a given temperature value in Fahrenheit to Celsius (substract 32, multiply by 5, then divide by 9) and test it.

Exercise 4.2: Function for ROC curves

Re-use the code from exercise 3.2 and write a function that automatically plots the ROC curve for a given protein accession. Use `stop()` to give an error if the defined accession is not present in the dataset. Return the ROC-object obtained by ROC() in the end.

Add arguments to the function (with appropriate default values) that allows costumization of the plot, e.g. turning on/off the printing of the AUC or best cutoffs.