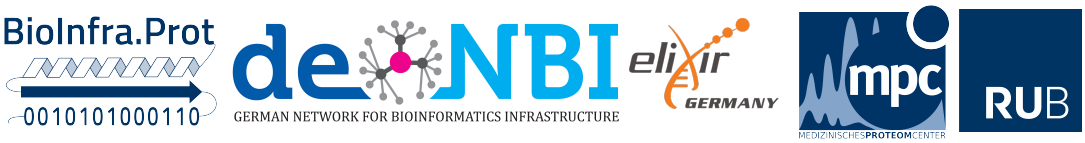
Differential analysis of quantitative proteomics data using R

Exercises

Michael Turewicz, Karin Schork

November 2, 2020



Contents

*Optional advanced exercise.

1 Hands-on session Ia

General hints:

- have a look at the slides to solve the exercises
- write down your code in an editor (e.g. the upper left window of RStudio, see slide 23)
- also look at the help pages of the mentioned R functions

Exercise 1.1: Pocket calculator

Open R and use it as *"pocket calculator"* for the following computational tasks:
$10 + 5$,
$1/2$,
$(3 + 5) * 2$,
$3 + 5 * 2$,
$3 + 5^2$
and $100/0$.

Exercise 1.2: Variables

Generate a variable x with value 2 and one variable y with value 5. Then compute the following expressions: $x + y$, $log_{10}(x \cdot y)$ and y^x.

Choose some of the functions mentioned in the script and use `?` or `??` to learn more about them. Terminate the R session correctly via `q()` without saving the workspace.

Exercise 1.3: Vectors

Create a vector x containing the numbers 5, 2, 4, 6 and 43.

Set the third element of x to 10.

Replace the last element of the vector by its negative value and print the modified vector x into the console using the function `print()`.

Create the vector $y = (4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5)$ in a *"clever"* way and print the vector y into the console.

Exercise 1.4: Matrices

Create the following matrices:

$$X = \begin{bmatrix} 3 & 8 \\ 6 & 2 \end{bmatrix} \qquad Y = \begin{bmatrix} 6 & 3 \\ 9 & 2 \\ 3 & 7 \end{bmatrix}$$

Index the matrices to the following elements, rows and columns of X and Y:

- the upper right element of X

- the lower left element of Y
- the first row of X
- the second column of Y

Exercise 1.5: Data frames

Use the function `data.frame()` to create a table with the following content:

Type	Diameter	Height	Age
Oak	57	15.7	29
Beech	38	12.1	18
Birch	23	8.5	10
Chestnut	63	17.3	27
Beech	41	15.1	21

First, index the data in order to obtain the diameters of the trees. Then, index the data in order to obtain the information on beeches. Finally, index the data in order to find trees that are 20 years or older.

2 Hands-on session Ib

Exercise 1.6: t-test

The following table lists the body height of six men and six women in cm. Create two vectors named `men` and `women` with the respective values. Then conduct a t-test (function `t.test()`) on the data to investigate the difference between men and women. Finally plot a box plot using the command

```
boxplot(men, women, names = c("men", "women"))
```

to visualize this difference.

men	186	176	182	173	190	181
women	172	168	178	166	175	170

Exercise 1.7: ANOVA*

The following table gives body weight in kg of different dogs. Investigate the difference between breeds (T terrier, S sausage dog, P pekinese) by means of ANOVA. Pass the data as a `data.frame()` to the function `aov()`. In case of rejection conduct the post-hoc test *"Tukey's honest significant difference"* using the function `TukeyHSD()`.

breed	T	T	T	S	S	S	P	P	P
weight	7.6	8.1	8.7	5.7	5.6	4.8	6.2	5.9	6.5

3 Hands-on session II

After considering the basics of R, the following exercises will be directly related to the differential analysis of your quantitative proteomics data. Thus, in the following you may also imagine a scenario where hundreds or thousands of proteins have been quantified (e.g., with label-free or targeted MS-based proteomics) in a set of samples from two or more groups of interest. E.g., in a preclinical study aiming at the detection of biomarker candidates these groups may be "diseased" and "healthy controls". Now, we will train the skills to detect the best biomarker candidates by computing p-values and fold changes (i.e., the ratios of means of the considered groups) using R.

Exercise 2.1: Data import

The folder *"data"* on the USB drive contains several files that are part of exercise 2.1. Import the files `example_import.txt`, `r_workshop_final_data1.txt`, `r_workshop_final_data1.csv`, `r_workshop_final_data2.txt` and `r_workshop_final_data2.csv`.
(hint: inspect the help pages of the R functions `read.table()` and `read.csv()`, see also slides 42-44).
It is always a good idea to have a short look into the data first (e.g., in order to see whether there is a header and which column separator is used).

After assigning it to the variable `dat1` inspect the imported file `r_workshop_final_data1.txt` using the functions `print()`, `dim()`, `head()` and `tail()`. In order to use `dat1` for the following exercises transform it into a matrix using the command `dat1 <- as.matrix(dat1)`.

Exercise 2.2: Descriptive statistics

Please use row 581 from `dat1` (from exercise 2.1) and compute characteristics like mean, variance, and extrema for group *"D"* and group *"C"*. To this end, create a vector of group membership using the command `groups <- substr(colnames(dat1), 1, 1)`. Then save the values of group *"D"* (i.e., `dat1[581, groups == "D"]`) in a vector `d` and the last five values (i.e., `dat1[581, groups == "C"]`) in a vector `c`.

Additionally, check the output of the function `summary()`. Finally draw a boxplot of the log2-transformed values using `boxplot(log2(d), log2(c))`.

Exercise 2.3: Create plots

By using the function `rnorm(n, mean, sd)` one can draw `n` random numbers from a normal distribution with mean `mean` and standard deviation `sd`. Create a vector x containing 50 random numbers from a normal distribution with mean 10 and standard deviation 1. Then create a vector y containing 50 random numbers from a normal distribution with mean 0 and standard deviation 0.5. Sum the vectors and store them in a vector z. Create a histogram as well as a boxplot of the vector x. Create a scatterplot

of vector z versus vector x. Create a barplot of the first 10 entries of y. (Hint: For the basic plot functions take a look at slides 54 and 57.)

Exercise 2.4: Graphics parameters

Recreate the scatter plot of exercise 2.3. Add an appropriate title and label the axes with "Variable x" and "Variable z". Instead of the default black circles use red triangles. Finally, save the graphic as png file into the results file of your USB drive or on the computer.

Hints: Take a look at slide 55 and the example on slide 58. You can get a list of predefined colours using the command `colors()`. Possible plot symbols are shown on the help page of the function argument `pch`. How to save a graphic is explained on slide 59.

Exercise 2.5: For-loop

Use a for-loop to calculate and print the first ten square numbers $(1^2, 2^2, ..., 10^2)$. Save the calculated numbers in a vector called `square`.

Exercise 2.6: Calculation of p-values and fold changes

In this exercise we will use the dataset `dat1` that you read into `R` in exercise 2.1. Make sure that you have transformed `dat1` into a matrix (`is.matrix(dat1)` should give the result `TRUE`).

Construct a for-loop which iterates over all 1000 rows of `dat1` and computes row-wise p-values (t-test) and mean ratios (fold changes) between the two groups. The t-test should be performed on the log2-transformed values (function `log2()`), while the mean ratios should be calculated on the untransformed data. Create a vector of p-values (`p`) and a vector of mean ratios (`fc`) that contain the p-values resp. mean ratios for all rows. Finally, to manage the multiple testing problem adjust the p-values using the function `p.adjust()` with the method `fdr`.

Hints: Take a look at slides 46-47. Construct the for-loop in a text editor (e.g. the upper left window of RStudio) before you test it in the R console.

Exercise 2.7: Simple volcano plot

Draw a simple volcano plot using the results from exercise 2.6. To this end, compute the `-log10` of the (not adjusted) p-values and the `log2` of the mean ratios. Then plot the resulting log-mean-ratios vs the resulting log-p-values. Also add a plot title and axis labels.

Exercise 2.8: Data export

Combine the results of exercise 2.6 with the original data `dat1` in order to obtain the summary of a simple differential analysis. To this end, use the function `cbind()` to

combine `dat1` with the vectors containing the p-values, adjusted p-values and mean ratios. Inspect the resulting table via `head()`. Then export the resulting table using the function `write.table()` as txt file into the results folder of your USB drive or on the computer. Finally, open the resulting file in excel.

Exercise 2.9: Improved volcano plot*

Draw an improved volcano plot (like on slide 51) with the following features:

- A horizontal dashed line indicates the p-value threshold of 0.05.
- Two vertical dashed lines indicate the respective mean ratio threshold on each side of the volcano plot.
- All differentially expressed proteins (e.g., p-value < 0.05 and mean ratio > 2 or $< \frac{1}{2}$) are highlighted in a specific color.
- The proteins that are significant after the multiple testing correction are highlighted in another color.

Hints: For drawing the lines the function `abline()` is useful. You can use the function `which()` (see slide 19) to select the differential and very best features.

Exercise 2.10: Abundance plot*

Plot the abundance profiles of a few (e.g., 2 or 4) interesting differentially expressed proteins (like on slide 52). Try to include some numeric information of the differential analysis in the plots (e.g., the adjusted p-values).
Hint: You can use the function `paste0()` to print the value of an object to use it for example in the plot title or within the function `text()`.