Introduction R and exploratory data analysis

Lab work

In this first practical, you will explore R and RMarkDown and use it to perform an initial analysis of a microarray dataset. You will work in RStudio. Begin by starting it up and having a look at the various subwindows.

When you start working with new languages and environments it can help to keep a "cheat sheet" handy. For R, ggplot2, RStudio and RMarkDown, good cheat sheets can be found here.

R

Maths and variables

Use the console to answer the following questions.

a. Calculate the modulus of 13 and the difference between 25 and the product of 3 and 6.

b. Calculate 2^10 + 2^6.

c. Create two variables x and y and assign them the values π and e. Calculate x to the power of y, rounded to the first decimal.

Vectors

d. Create a vector v containing the first five prime numbers. What is range(v)?

e. Set the sixth element of v to 0/0. What is sum(v), and why?

f. How do you select only the valid numbers in v?

Matrices and packages

g. Install the "magic" library and load it, so it can be used.

h. Generate a 3x3 magic square M (hint: ?magic).

i. What is the maximum value of the square of M?

Loops, functions and scripts

j. Write a loop that generates a vector f containing the first 10 Fibonacci numbers (hint: see Wikipedia).

k. Write a function is.odd(v) that returns only the odd numbers in a vector v.

l. Combine the answers to the two questions above into a script that outputs what percentage of the first 50 Fibonacci numbers is odd.

RMarkDown

m. Write an RMarkDown report that outputs a magic square of size 5, without echoing the R code. Give the report a descriptive header.

n. Now add the code for your function is.odd(v), without running it.

o. Finally, add scatterplots of the built-in iris dataset, using plot(iris), and generate a Word file.

Optional

p. Generate a vector v containing 4 4's, 8 8's and 12 12's, in that order, using rep().

q. Calculate the sum of $(2^i/i^2 + 3^i/i^3)$ for i = 10...20, using a for loop.

r. Calculate the sum of $(2^i/i^2 + 3^i/i^3)$ for i = 10...20, using a single command sequence.

s. Create a vector l containing the labels "Sample 1" to "Sample 72", using paste().

t. Use runif() and a rounding function to generate a vector r of 10,000 integers uniformly drawn in the range 1...3. Verify your answer using table().

u. Create a 5x5 magic square matrix M as above and calculate the row sums and the column sums using the apply() function.

v. Solve $Mx = b$ for x, where M is a 3x3 magic square matrix and b = c(1,2,3), using solve(). Verify your outcome.

Summary statistics

In this second part, you will work with a small, real-world microarray dataset. It contains 4 Affymetrix HG-U133A microarray samples, 2 each taken in 2 human T-cell development stages: single positive CD4+CD-8 and single positive CD4-CD8+ (for an overview of T-cell development, see Fig. 1 in http://www.nature.com/nri/journal/v2/n5/full/nri798.html). CDx+ means a cell expresses a T-cell receptor CDx, CDx- means it does not.

Dataset characteristics

a. Download and unzip the data for block 1 from Blackboard. The microarray data is stored in "SP48.txt". Inspect it using a text editor (Notepad or Word) and check what separator is used and whether it contains a header. What is the correct call to read.table() in R? Use it to read in the data.

b. Use View(), class(), dim(), ncol(), nrow() and names() to inspect the data. What is the number of rows of the dataset?

c. What do the rows correspond to? Hint: have a look at http://www.affymetrix.com/support/help/faqs/mouse_430/faq_8.jsp.

d. What is the number of columns?

e. You can check your answers using str(), which summarizes the aspects of a dataset. Now inspect the first and last few rows of the data using head() and tail(), and get an overview using summary(). What do you think the range of the expression data is?

f. Estimate how many unique genes the dataset contains?

g. Which gene is represented by most probesets?

h. How many NAs are there?

<u>Visualization</u>

i. Use table() to learn how many immunity-related genes there are in the data and create a bar plot.

j. Create histograms of the four microarray samples. What do they tell you about the intensity distribution?

k. Now create histograms of the log2() of the microarray data, using ggplot2. What do these tell you?

l. Are densities more informative?

<u>Data inspection</u>

m. Plot densities for both immune genes and others, using ggplot(data=data, aes(x=log(SP4plus.1))) + geom_density() + facet_wrap(~Type). Repeat for the other arrays. Do you see a difference between the distributions of these two types of genes?

n. Use ggplot() to make a scatterplot of the expression values measured in the two CD4+CD8- samples. Do the same for the CD4-CD8+ samples. What do you notice?

In order to visualize some aspects of the data, it is sometimes necessary to reshape your data or calculate intermediate results yourself. There is no standard recipe, but in general you can use the functions as.matrix(), as.numeric() and data.frame() to convert selected rows and colums to a matrix, matrix of numbers or a data frame, respectively.

To plot the average expression of genes in the CD4+CD8- samples against those in the CD4-CD8+ samples, you can do the following:

```
> # Calculate the mean CD4+CD8- expression
> mn4 <- rowMeans(data[,4:5])
> # Calculate the mean CD4-CD8+ expression
> mn8 <- rowMeans(data[,6:7])
> # Concatenate into data frame
> mndata <- data.frame(SP4plus=mn4,SP8plus=mn8)
```

o. Use the code above and ggplot() to make a scatterplot of the average expression values in the CD4+CD8- samples vs. those in the CD4-CD8+ samples. What do you see?

p. Modify the R code above so that mndata also contains the Type column found in the original data, and use this to colour the points according to the gene type. Can you spot a difference between the two sets of genes? *Optional: retrieve the names of the two immune-related genes that stand out.*

q. Find the two genes coding for the receptors, CD4 and CD8A and plot their expression. Does it concur with your ideas?

r. Create boxplots of the four microarrays using boxplot(log2(data[,4:7])). What potential problem do you notice?

s. What could cause this potential problem and how could you correct for it?

<u>Optional</u>

t. The data you were supplied with was already pre-processed: probe values were combined into overall expression values for probesets BioConductor library contains numerous packages to work directly with the raw data, in this case Affymetrix CEL files.

Install the affyQCReport package and its dependencies:

```
> source("https://bioconductor.org/biocLite.R")
> biocLite("affyQCReport")
```

Then generate an Affymetrix quality report using:

```
> library(affyQCReport)
> QCReport(ReadAffy(celfile.path="<MYFOLDER>"))
```

where you replace by the name of the folder that contains the CEL files ("SP4+_1na.CEL" ..., "SP8+_2na.CEL"). This function places a file AffyQCReport.pdf in the same folder. Open it - does it corroborate your findings above?