BioInformatcs

*Assignment*

# Some questions to orientate yourself.

(a) matrix, numeric, numeric, matrix, function, function, factor, standardGeneric, ExpressionSet. (b) remove, summation, product, sequence, standard deviation, number of rows, (c) Use R its help or use the internet search key ”r wiki grep” to find the following answers: searching regular expressions, return a vector from a function on the rows or columns of a matrix, generate a factor by specifying the pattern of levels, load addon packages, make R reading input from a file or URL, set the working directory to a certain map, print the last · commands

# gendat

1. apply(gendat,2,sd).
2. apply(gendat,1,sd).
3. To order the data frame according to the gene standard deviations. sdexprsval <- apply(gendat,1,sd)

* <- order(sdexprsval,decreasing=TRUE) gendat[o,]

1. gene1

# Computations on gene means of the Golub data.

1. Computation of mean gene expression values. data(golub, package = "multtest")

meangol <- apply(golub,1,mean)

1. To order the data frame use

* <- order(meangol,decreasing=TRUE) and golub[o,]

1. Give the names of the three genes with the largest mean expression value.

* golub.gnames[o[1:3],3]

[1] "U43901\_rna1\_s\_at" "M13934\_cds2\_at" "X01677\_f\_at"

1. Give their biological names.

* golub.gnames[o[1:3],2]

1. "37 kD laminin receptor precursor/p40 ribosome associated protein
2. "RPS14 gene (ribosomal protein S14) extracted from Human ribosoma
3. "GAPD Glyceraldehyde-3-phosphate dehydrogenase"

# Computations on gene standard deviations of the Golub data.

1. The standard deviation per gene can be computed by sdgol <- apply(golub,1,sd).
2. The gene with standard deviation larger than 0.5 can be selected by golubsd <-
3. sum(sdgol>0.5) gives that the number of genes having sd larger than 0.5 is 1498.

# Oncogenes in Golub data.

1. length(agrep("^oncogene",golub.gnames[,2])) gives 42.
2. By the script below the "Cellular oncogene c-fos is found. data(golub, package="multtest")

rowindex <- agrep("^oncogene",golub.gnames[,2]) oncogol <- golub[rowindex,]

oncogolub.gnames <- golub.gnames[rowindex,]

gol.fac <- factor(golub.cl,levels=0:1, labels= c("ALL","AML")) meangol <- apply(oncogol[,gol.fac=="ALL"],1,mean)

* <- order(meangol,decreasing=TRUE) oncogolub.gnames[o[1:3],2]

[1] "PIM1 Pim-1 oncogene" "JUNB Jun B proto-oncogene"

[3] "Proto-oncogene BCL3 gene"

1. (c)

meangol <- apply(oncogol[,gol.fac=="AML"],1,mean) o <- order(meangol,decreasing=TRUE)

* oncogolub.gnames[o[1:3],2]

[1] "PIM1 Pim-1 oncogene" "JUNB Jun B proto-oncogene"

[3] "Proto-oncogene BCL3 gene" (d)

Writing results to a csv file. Be aware of the correct column separation. x <- oncogolub.gnames[o[1:10],c(3,2)]

colnames(x) <- c("probe ID","gene name") write.csv(x,file="goluboutcsv") write.table(x,file="goluboutnorowname",row.names=FALSE)

# 7. Gene means for B1 patients:

library(ALL); data(ALL)

meanB1 <- apply(exprs(ALL[,ALL$BT=="B1"]),1, mean) o <- order(meanB1,decreasing=TRUE)

meanB1[o[1:3]]