Bioinformatics Lab 4

Understanding and basic analysis of Golub dataset in R

The gene expression data collected by Golub et al. (1999) are among the classical in bioinformatics. A selection of the set is called golub and is contained in the multtest package, which is part of Bioconductor. The data consist of gene expression values of 3051 genes (rows) from 38 leukemia patients. Twenty seven patients are diagnosed as acute lymphoblastic leukemia (ALL) and eleven as acute myeloid leukemia (AML). The tumor class is given by the numeric vector golub.cl, where ALL is indicated by 0 and AML by 1. The gene names are collected in the matrix golub.gnames of which the columns correspond to the gene index, ID, and Name, respectively. We shall first concentrate on expression values of a gene with manufacturer name "M92287_at", which is known in biology as "CCND3 Cyclin D3". The expression values of this gene are collected in row 1042 of golub. To load the data and to obtain relevant information from row 1042 of golub.gnames, use the following.

```
>if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
>BiocManager::install(version = "3.10")
>install.packages("multtest")
> library(multtest);
>data(golub)
> golub.gnames[1042,]
[1] "2354" "CCND3 Cyclin D3" "M92287_at"
```

The data are stored in a matrix called golub. The number of rows and columns can be obtained by the functions nrow and ncol, respectively.

```
> nrow(golub)
 [1] 3051
> ncol(golub)
[1] 38
```

So the matrix has 3051 rows and 38 columns, see also dim(golub). Each data element has a row and a column index. Recall that the first index refers to rows and the second to columns. Hence, the second value from row 1042 can be printed to the screen as follows.

```
> golub[1042,2]
[1] 1.52405
```

So 1.52405 is the expression value of gene CCND3 Cyclin D3 from patient number 2. The values of the first column can be printed to the screen by the following.

```
> golub[,1]
```

To save space the output is not shown. We may now print the expression values of gene CCND3 Cyclin D3 (row 1042) to the screen.

```
> golub[1042,]
```

```
[1] 2.10892 1.52405 1.96403 2.33597 1.85111 1.99391 2.06597 1.81649

[9] 2.17622 1.80861 2.44562 1.90496 2.76610 1.32551 2.59385 1.92776

[17] 1.10546 1.27645 1.83051 1.78352 0.45827 2.18119 2.31428 1.99927

[25] 1.36844 2.37351 1.83485 0.88941 1.45014 0.42904 0.82667 0.63637

[33] 1.02250 0.12758 -0.74333 0.73784 0.49470 1.12058
```

To print the expression values of gene CCND3 Cyclin D3 to the screen only for the ALL patients, we have to refer to the first twenty seven elements of row 1042. A possibility to do so is by the following.

```
> golub[1042,1:27]
```

However, for the work ahead it is much more convenient to construct a factor indicating the tumor class of the patients. This will turn out useful e.g. for separating the tumor groups in various visualization procedures. The factor will be called gol.fac and is constructed from the vector golub.cl, as follows.

```
> gol.fac <- factor(golub.cl, levels=0:1, labels = c("ALL","AML"))
```

In the sequel this factor will be used frequently. Obviously, the labels correspond to the two tumor classes. The evaluation of gol.fac=="ALL" returns TRUE for the first twenty seven values and FALSE for the remaining eleven.

This is useful as a column index for selecting the expression values of the ALL patients. The expression values of gene CCND3 Cyclin D3 from the ALL patients can now be printed to the screen, as follows.

```
> golub[1042,gol.fac=="ALL"]
```

For many types of computations it is very useful to combine a factor with the apply functionality. For instance, to compute the mean gene expression over the ALL patients for each of the genes, we may use the following.

```
> meanALL <- apply(golub[,gol.fac=="ALL"], 1, mean)
```

The specification golub[,gol.fac=="ALL"] selects the matrix with gene expressions corresponding to the ALL patients. The 3051 means are assigned to the vector meanALL.

After reading the classical article by Golub et al. (1999), which is strongly recommended, one becomes easily interested in the properties of certain genes. For instance, gene CD33 plays an important role in distinguishing lymphoid from myeloid lineage cells. To perform computations on the expressions of this gene we need to know its row index. This can obtained by the grep function.

```
> grep("CD33",golub.gnames[,2])

[1] 808
```

Hence, the expression values of antigen CD33 are available at golub[808,] and further information on it by golub.gnames[808,].

Questions:

1. Some questions to orientate yourself.

(a) Use the function class to find the class to which the following objects belong: golub, golub[1,1],golub.cl, golub.gnames, apply, exp, gol.fac, plot, ALL.

(b) What is the meaning of the following abbreviations: rm, sum, prod, seq, sd, nrow.

(c) For what purpose are the following functions useful: grep, apply, gl, library, source, setwd, history, str.

2. Create the matrix gendat as follows:

```
>gene1 <- c(1.00,1.50,1.25)

> gene2 <- c(1.35,1.55,1.00)

> gene3 <- c(-1.10,-1.50,-1.25)

> gene4 <- c(-1.20,-1.30,-1.00)

> rowcolnames <- list(c("gene1","gene2","gene3","gene4"), + c("Eric","Peter","Anna"))

> gendat <- matrix(c(gene1,gene2,gene3,gene4), nrow=4, ncol=3, + byrow=TRUE, dimnames = rowcolnames)
```

Print the gendat:

```
> gendat

Eric Peter Anna

gene1 1.00 1.50 1.25

gene2 1.35 1.55 1.30

gene3 -1.10 -1.50 -1.25

gene4 -1.20 -1.30 -1.00
```

Consider the data in the matrix gendat, constructed above. Its small size has the advantage that you can check your computations even by a pocket calculator.

(a) Use apply to compute the standard deviation of the persons.

(b) Use apply to compute the standard deviation of the genes.

(c) Order the matrix according to the gene standard deviations.

(d) Which gene has the largest standard deviation?

3. Computations on gene means of the Golub data.

(a) Use apply to compute the mean gene expression value.

(b) Order the data matrix according to the gene means.

(c) Give the names of the three genes with the largest mean expression value.

(d) Give the biological names of these genes.

4. Computations on gene standard deviations of the Golub data.

(a) Use apply to compute the standard deviation per gene

(b) Select the expression values of the genes with standard deviation larger than two.

(c) How many genes have this property?

5. Oncogenes in Golub data.

(a) How many oncogenes are there in the dataset? Hint: Use grep.

(b) Find the biological names of the three oncogenes with the largest mean expression value for the ALL patients.

(c) Do the same for the AML patients.

(d) Write the gene probe ID and the gene names of the ten genes with largest mean gene expression value to a csv file.

6. Constructing a factor. Construct factors that correspond to the following setting.

(a) An experiment with two conditions each with four measurements.

(b) Five conditions each with three measurements.

(c) Three conditions each with five measurements.

7. Gene means for B1 patients. Load the ALL data from the ALL library and use str and openVignette() for a further orientation.

(a) Use exprs(ALL[,ALL$BT=="B1"] to extract the gene expressions from the patients in disease stage B1. Compute the mean gene expressions over these patients.

(b) Give the gene identifiers of the three genes with the largest mean.