Bioinformatics
Lab 4

Understanding and basic analysis of Golub dataset in R

Submission by

Nitesh yadav BT18CSE045,
Kalp Pawar BT18CSE037

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.
=> "matrix" "array", "numeric", " numeric", " matrix", "function", "function", "factor", "standardGeneric", "ExpressionSet".

(b) What is the meaning of the following abbreviations: rm, sum, prod, seq, sd, nrow.
=> remove, summation, product, sequence, standard deviation, no. of rows

(c) For what purpose are the following functions useful: grep, apply, gl, library, source, setwd, history, str.
=> Using R studio's help or use the internet search we 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

2. Consider the data in the matrix gendat, constructed above. Its small size has the advantage that you can check your computations even with a pocket calculator.
(a) Use apply to compute the standard deviation of the persons.

```
=>      apply(gendat,2,sd).
```

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

```
=>      apply(gendat,1,sd).
```

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

```
=>      sdVal<- apply(gendat,1,sd)
        o <- order(sdVal,decreasing=TRUE)
        gendat[o,]
```

(d) Which gene has the largest standard deviation?

=> gene2

3. Computations on gene means of the Golub data.
(a) Use apply to compute the mean gene expression value.

```
=>      data(golub, package = "multtest")
> meanOfGolub <- apply(golub,1,mean) # mean of genes
> meanOfGolub
```

```
 [1] -1.129013158 -0.846745526  0.260806053  0.949457632
 [5]  0.475348158  0.555668684  3.058682105  2.956468684
        ………
```

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

```
=>      orderedGolub <- order(meanOfGolub, decreasing = TRUE)
        golub[orderedGolub,]
        > golub[orderedGolub,]
        [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,]  2.64342  1.01416  1.70477  1.63845 -0.36075  1.73451
[2,]  0.41189  1.09121  0.66959  0.80965  0.52493  0.93866
[3,]  0.17642  0.42573  0.60000  0.35192  0.70560  0.40819
[4,]  2.42764  1.34873  1.61846  1.80194  0.81975  2.18509
[5,] -0.18126 -0.70011  0.18340  0.49696 -0.53083 -0.32664
[6,]  0.80633  0.26994  0.49549  0.15222 -0.03737 -0.27141
[7,] -0.33978 -0.12775  0.32201  0.12330 -1.42668  0.59579
………………………………
```

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

```
=> golub.gnames[orderedGolub[1:3],3]
[1] "U43901_rna1_s_at" "M13934_cds2_at"   "X01677_f_at"
```

(d) Give the biological names of these genes.

```
> golub.gnames[orderedGolub[1:3],2]
[1] "37 kD laminin receptor precursor/p40 ribosome associated protein gene"
[2] "RPS14 gene (ribosomal protein S14) extracted from Human ribosomal protein S14 gene"
[3] "GAPD Glyceraldehyde-3-phosphate dehydrogenase"
```

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

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

```
=>      > sdgolub <- apply(golub,1,sd)
        > sdgolub
  [1] 0.5878202 0.5292176 0.4999966 1.7157505 1.7212612
  [6] 1.6166706 0.6755388 0.7738532 0.5100332 0.7507548
 [11] 0.5221723 0.5731508 0.6606204 1.3963950 0.8056142
 [16] 0.3816615 0.6496579 0.5913919 0.5486941 0.5761311
…………………………………………………………………………
```

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

```
> # select genes whose sdgolub > 2
> golubsd <- golub[which(sdgolub > 2)]
```

(c) How many genes have this property?

=> length(golubsd)

i.e 0

5. Oncogenes in Golub data.

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

=>

```
> #5
> length(agrep("^oncogene",golub.gnames[,2]))
[1] 42
```

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

=> gol.fac

```
 [1] ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL
ALL ALL ALL ALL ALL ALL ALL
[27] ALL AML AML AML AML AML AML AML AML AML AML AML
Levels: ALL AML
> rowindex <- agrep("^oncogene",golub.gnames[,2])
> oncogol <- golub[rowindex,]
> oncogolub.gnames <- golub.gnames[rowindex,]
> meangol <- apply(oncogol[,gol.fac=="ALL"],1,mean)
> meangol
 [1] -1.013413333  0.026425926 -0.300583704  0.060526667  1.044555185 -0.283669630  0.812428519 -0.463173333
 [9] -1.230901111 -0.271092963  0.343261481  0.131822222 -0.974125926 -0.160778519 -0.777962963 -0.005456296
[17] -0.241111111 -0.928446296 -0.631110000  1.431738148 -0.141772593  0.369009630 -0.304648148  1.112213333
[25]  0.107616667 -0.977334074  0.431988889 -0.609660370 -0.225234815 -0.467505185 -1.223695185 -0.199179630
[33] -0.243334815  0.914368519  0.298548889  0.975898889 -1.161305185 -1.238866667  0.219340000  0.829462593
[41] -0.569108148  0.928563333
```

(c) Do the same for the AML patients.

=>

```
> meangol <- apply(oncogol[,gol.fac=="AML"],1,mean)
> meangol
 [1] -0.980203636  0.429889091 -0.448765455  0.003861818  0.791785455  1.098498182  1.161930909 -0.424228182
 [9] -1.330860000 -0.164797273  0.712014545  0.115006364 -0.853126364 -0.361431818 -0.528630000  0.173829091
[17] -0.869589091 -0.547760909 -0.466171818  1.404113636 -0.082412727  0.497500000 -0.249169091  0.949267273
[25]  0.129575455  0.306901818  0.939765455  0.059900909 -0.618456364 -1.314710000 -0.704390000 -0.792855455
[33]  0.880079091  1.703262727  1.269604545  1.452010000  0.649089091 -0.658342727  0.210742727  0.925041818
[41] -0.316480000  1.565982727
```

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

=>

```
> x <- golub.gnames[orderedGolub[1:10],c(3,2)]
> colnames(x) <- c("probe ID","gene name")
> write.csv(x,file="D:\\SEM8\\BIOINFOMATICS\\Lab\\Lab4\\golubout.csv")
> read.csv(file = "D:\\SEM8\\BIOINFOMATICS\\Lab\\Lab4\\golubout.csv")
    X             probe.ID
1   1       U43901_rna1_s_at
2   2         M13934_cds2_at
3   3            X01677_f_at
4   4            X03689_s_at
5   5            U06155_s_at
6   6            D49824_s_at
7   7            Z49148_s_at
8   8            X00351_f_at
9   9 AFFX-HSAC07/X00351_M_at
10 10 AFFX-HSAC07/X00351_5_at
```

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

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

```
=>> gl(2,5)
 [1] 1 1 1 1 1 2 2 2 2 2
```

(b) Five conditions each with three measurements.

```
=>> gl(5,3)
 [1] 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5
Levels: 1 2 3 4 5
```

(c) Three conditions each with five measurements.

```
=>> gl(3,5)
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels: 1 2 3
```

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.

```
> library(ALL);
> data(ALL)
> meanB1 <- apply(exprs(ALL[,ALL$BT=="B1"]),1, mean)
> o <- order(meanB1,decreasing=TRUE)
> o
  [1] 12586  1979  1974 10990  6609   311  1558  5962
  [9]  2488  1261  4636  2070  4687  1540  5328  1701
 [17]  4845  1127  4123  3145  1396  1973  2296  2351
```

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

=>

```
> meanB1[o[1:3]]
AFFX-hum_alu_at       31962_at      31957_r_at
      13.41648       13.16671       13.15995
```