Assignment 1 — Total mark: 20

Due by 12:00pm by email [redacted] *on Sunday, April 14th, 2013. Read the instructions carefully. The assignment will be submitted in the form of a .doc or .docx file. You will give the* `R` *code for each question and comment/interpret it. Comments on each question should not exceed 200 words and the total document should not exceed 10 pages. When including a figure, do not forget to add a succinct legend mentioning exercise number, question number and type of plot.*

Exercise 1 *Box-and-wiskers plot of persons of Golub et al. (1999) data* *[Total mark: 2].*

1. *Use* `boxplot(golub100)` *to produce a box-and-whiskers plot for each column (patient). Make a screen shot to save it in a word processor. Describe what you see. Are the medians of similar size? Is the inter quartile range more or less equal. Are there outliers?* *[1pt]*

```
> load('Golub.RData')
> boxplot(golub100, col = 'gray')
```

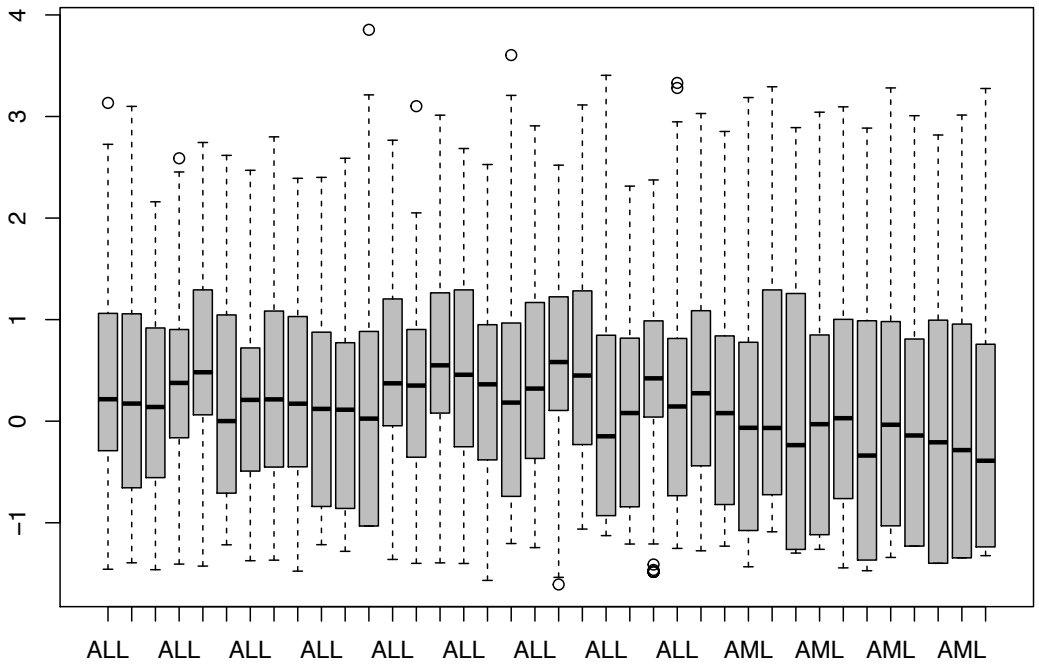

Figure 1: Golub100 data. Boxplot of the samples.

Figure 1 displays the boxplots of the expression of the 100 genes in the Golub study for each patient. The medians vary across patients, and the IQR seems to be larger in the AML patients. We observe few outliers values for some of the samples.

2. *Compute the mean and medians of the patients. What do you observe?* *[1pt]*

```
> # mean:
> apply(golub100, 2, mean)

      ALL       ALL       ALL       ALL       ALL       ALL       ALL
0.4261627 0.2253569 0.1766965 0.4213194 0.5929503 0.1648194 0.1815467
      ALL       ALL       ALL       ALL       ALL       ALL       ALL
0.2926318 0.2687476 0.1248357 0.1138486 0.2002946 0.5082512 0.3035016
      ALL       ALL       ALL       ALL       ALL       ALL       ALL
0.6145734 0.5228583 0.2627021 0.1840416 0.4142964 0.5908791 0.5293863
      ALL       ALL       ALL       ALL       ALL       ALL       AML
0.0641669 0.1091314 0.4654401 0.2025647 0.3750564 0.1395655 -0.0010639
      AML       AML       AML       AML       AML       AML       AML
0.2444376 0.0608007 0.0947361 0.1770030 -0.0657025 0.1297310 -0.0392008
      AML       AML       AML
0.0422152 0.0404711 -0.0601693

> #median
> apply(golub100, 2, median)

     ALL      ALL      ALL      ALL      ALL      ALL      ALL      ALL
0.216310 0.173445 0.139625 0.376730 0.481665 0.001035 0.209645 0.214745
     ALL      ALL      ALL      ALL      ALL      ALL      ALL      ALL
0.172190 0.120940 0.112955 0.025350 0.372840 0.350760 0.550050 0.457205
     ALL      ALL      ALL      ALL      ALL      ALL      ALL      ALL
0.363085 0.183000 0.321160 0.582105 0.449425 -0.148790 0.080170 0.421970
     ALL      ALL      ALL      AML      AML      AML      AML      AML
0.144730 0.275215 0.079355 -0.065285 -0.066760 -0.235745 -0.029640 0.029405
     AML      AML      AML      AML      AML      AML
-0.338460 -0.035130 -0.140770 -0.207795 -0.284710 -0.389435

> # to visualise mean and median:
> par(mfrow = c(2,1))
> boxplot(apply(golub100, 2, mean)~gol.factor, col = 'lightskyblue', main = 'Mean of patients')
> boxplot(apply(golub100, 2, median)~gol.factor, col = 'cyan', main = 'Median of patients')
> par(mfrow=c(1,1))
```

```
      ALL       ALL       ALL       ALL       ALL       ALL       ALL
0.4261627 0.2253569 0.1766965 0.4213194 0.5929503 0.1648194 0.1815467
      ALL       ALL       ALL       ALL       ALL       ALL       ALL
0.2926318 0.2687476 0.1248357 0.1138486 0.2002946 0.5082512 0.3035016
      ALL       ALL       ALL       ALL       ALL       ALL       ALL
0.6145734 0.5228583 0.2627021 0.1840416 0.4142964 0.5908791 0.5293863
      ALL       ALL       ALL       ALL       ALL       ALL       AML
0.0641669 0.1091314 0.4654401 0.2025647 0.3750564 0.1395655 -0.0010639
      AML       AML       AML       AML       AML       AML       AML
0.2444376 0.0608007 0.0947361 0.1770030 -0.0657025 0.1297310 -0.0392008
      AML       AML       AML
0.0422152 0.0404711 -0.0601693
```

```
     ALL       ALL       ALL       ALL       ALL       ALL       ALL       ALL
0.216310  0.173445  0.139625  0.376730  0.481665  0.001035  0.209645  0.214745
     ALL       ALL       ALL       ALL       ALL       ALL       ALL       ALL
0.172190  0.120940  0.112955  0.025350  0.372840  0.350760  0.550050  0.457205
     ALL       ALL       ALL       ALL       ALL       ALL       ALL       ALL
0.363085  0.183000  0.321160  0.582105  0.449425 -0.148790  0.080170  0.421970
     ALL       ALL       ALL       AML       AML       AML       AML       AML
0.144730  0.275215  0.079355 -0.065285 -0.066760 -0.235745 -0.029640  0.029405
     AML       AML       AML       AML       AML       AML
-0.338460 -0.035130 -0.140770 -0.207795 -0.284710 -0.389435
```

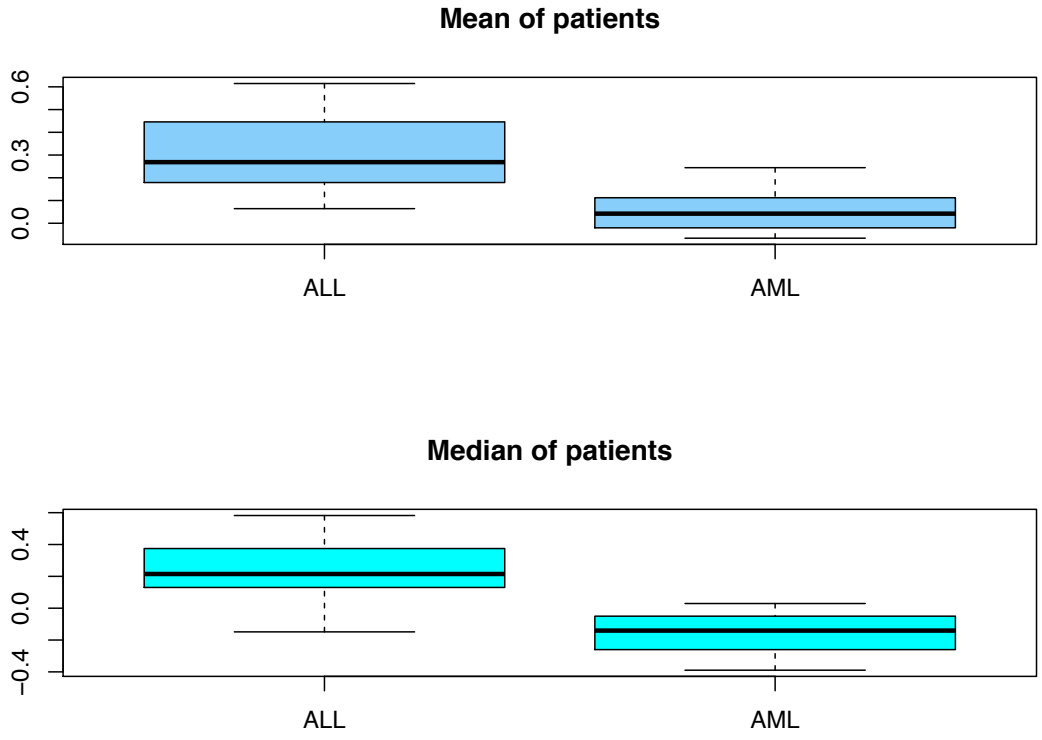

Figure 2: Golub100 data subset. Boxplot of the patients' means.

The means vary across patients, with a lower mean for the AML patients, as shown in the boxplot of the means in Figure 2 (additional plots). Similarly, the median is also lower for the AML patients.

Exercise 2 *Hypothesis testing 1/2* *[Total mark: 5].*

1. *Gene selection. We are interested in the following list of candidate genes for the Golub study (you will use the* **golub100** *data from the practical):*

```
list.gene = c(
  "LYZ Lysozyme", "CTSD Cathepsin D (lysosomal aspartyl protease)",
  "Clone 23721 mRNA sequence", "Neuromedin B mRNA",
  "DHPS Deoxyhypusine synthase",
  "GB DEF = (lambda) DNA for immunoglobin light chain",
  "Leukotriene C4 synthase (LTC4S) gene", "KIAA0102 gene" ,
  "Non-lens beta gamma-crystallin like protein (AIM1) mRNA, partial cds",
  "CD24 signal transducer mRNA and 3' region")
```

```
> # create list of genes
> list.gene = c(
+   "LYZ Lysozyme", "CTSD Cathepsin D (lysosomal aspartyl protease)",
+   "Clone 23721 mRNA sequence", "Neuromedin B mRNA",
+   "DHPS Deoxyhypusine synthase",
+   "GB DEF = (lambda) DNA for immunoglobin light chain",
+   "Leukotriene C4 synthase (LTC4S) gene", "KIAA0102 gene" ,
+   "Non-lens beta gamma-crystallin like protein (AIM1) mRNA, partial cds",
+   "CD24 signal transducer mRNA and 3' region")
```

   (a) *For each of these genes, perform a two-sample t-test values for which the ALL mean is greater than the AML mean (test with unequal variances). You will formulate the null and alternative hypotheses, the test statistic used (and the distribution of the test statistic, including the number of degrees of freedom), the rejection region and draw your conclusion. (Advice: you can create a vector called* **p.value** *which will store the p-values associated to the tests, and name* **p.value** *using the function* **names**). *[2pt]*

```
> # version 1:do a loop
> p.value = vector(length = length(list.gene))
> for(k in 1:length(list.gene)){
+   p.value[k] = t.test(golub100[list.gene[k],]~gol.factor, var.equal = FALSE)$p.value
+ }
> # version 2: create a function and use apply
> my.function = function(x){
+   t.test(x~gol.factor, var.equal = FALSE)$p.value
+ }
> p.value = apply(golub100[list.gene,], 1, my.function)
> names(p.value) = list.gene
```

   For each gene, let μ_{ALL} *(*μ_{AML}*) be the true mean of the expression level of the ALL (AML) patients*m_{ALL} *(*m_{AML}*) be the sample mean of the expression level of the ALL (AML) patients. The null hypothesis is* $H_0 : \mu_{ALL} = \mu_{ALL}$ *and the alternative hypothesis is* $H_1 : \mu_{ALL} \neq \mu_{ALL}$.
   Under H_0*, the test statistics used is*

$$T = \frac{m_{ALL} - m_{ALL}}{\sqrt{s^2_{ALL}/n_{ALL} + s^2_{AML}/n_{AML}}}$$

   where $n_{ALL} = 27$*(*$n_{AML} = 11$*) is the number of patients with ALL (AML) and* s_{ALL}*(*s_{AML}*) is the sample variance of that given gene for ALL (AML) patients.*
   We have T τ_{36}.
   We reject H_0 *if* $|T| > T_{table}(0.975, 36) =$ *qt(0.975, 36)* $= 2.028094$*, or if the p-value associated to this test is lower than 5%. In that case, we will conclude that this given gene is differentially expressed between the ALL and AML conditions.*
   The p-values for all the genes of interest are given in Table 1. We can see that all genes have a very low p-value < 0.01.

	p-value
LYZ Lysozyme	0.0000003933
CTSD Cathepsin D (lysosomal aspartyl protease)	0.0000033941
Clone 23721 mRNA sequence	0.0000087099
Neuromedin B mRNA	0.0000007561
DHPS Deoxyhypusine synthase	0.0000001038
GB DEF = (lambda) DNA for immunoglobin light chain	0.0000030673
Leukotriene C4 synthase (LTC4S) gene	0.0000016540
KIAA0102 gene	0.0000046124
Non-lens beta gamma-crystallin like protein (AIM1) mRNA, partial cds	0.0000081013
CD24 signal transducer mRNA and 3' region	0.0000103915

Table 1: p-values of the genes of interest (t-test, unequal variances)

(b) Report amongst these genes those that have a p-value < 0.01. We will refer to these genes as 'differentially expressed' genes. *[1pt]*

```
> which(p.value < 0.01)
```

```
                                                    LYZ Lysozyme
                                                               1
                                  CTSD Cathepsin D (lysosomal aspartyl protease)
                                                               2
                                                 Clone 23721 mRNA sequence
                                                               3
                                                     Neuromedin B mRNA
                                                               4
                                                DHPS Deoxyhypusine synthase
                                                               5
                                 GB DEF = (lambda) DNA for immunoglobin light chain
                                                               6
                                              Leukotriene C4 synthase (LTC4S) gene
                                                               7
                                                         KIAA0102 gene
                                                               8
Non-lens beta gamma-crystallin like protein (AIM1) mRNA, partial cds
                                                               9
                                          CD24 signal transducer mRNA and 3' region
                                                              10
```

```
> DE.gene = names(which(p.value < 0.01))
```

(c) Illustrate these results by plotting these differentially expressed genes using boxplots. Interpret the boxplots. *[2pt]*

```
> par(mfrow=c(4,4)) # divide window into 4 rows, 4 columns
> for(k in 1:length(DE.gene)){
+   boxplot(golub100[DE.gene[k],]~gol.factor, col = 'plum', main = paste(DE.gene[k]),
+           cex.main = 0.5)
+ }
> par(mfrow=c(1,1))
```

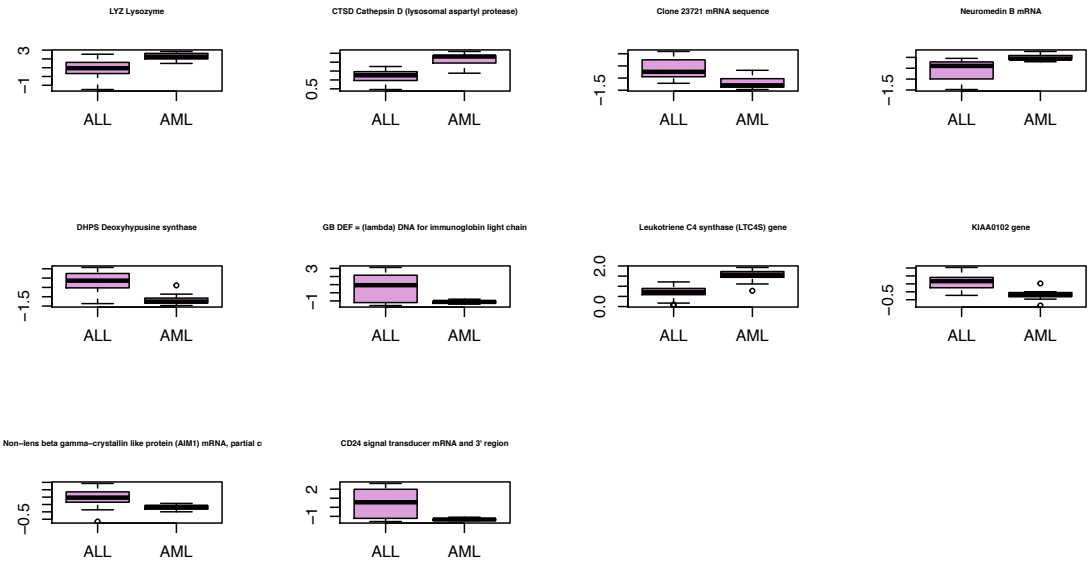

Figure 3: Golub100 data. Boxplot of the DE genes with respect to the patients' tumour status.

The boxplots displayed in Figure 3 illustrate the pattern of each of these genes with respect to the patients' tumour status. Four of these genes are over expressed in AML and the rest are overexpressed in ALL.

Exercise 3 *Hypothesis testing 2/2* *[Total mark: 4].*

1. *Gene CD33. Use the* `grep()` *function to find the index of the important gene CD33 among the rownames of* `golub100`*. For each test below, formulate the null and alternative hypotheses, the test statistic used (its value and the distribution of the test statistic, including the number of degrees of freedom), the rejection region and draw your conclusion:*

```
> index.CD33 = grep('CD33', rownames(golub100))
> # check:
> rownames(golub100)[index.CD33]

[1] "CD33 CD33 antigen (differentiation antigen)"

> # This is the expression of this gene:
> CD133 = golub100[index.CD33,]
```

   (a) *Test the normality of the ALL and AML expression values*[1]. *[1pt]*

```
> # test for ALL
> shapiro.test(CD133[gol.factor=="ALL"])

        Shapiro-Wilk normality test

data:  CD133[gol.factor == "ALL"]
W = 0.9696, p-value = 0.592

> # test for AML
> shapiro.test(CD133[gol.factor=="AML"])
```

[1]For this particular question, only formulate the null and alternative hypotheses, the value of the test statistic, the rejection region and draw your conclusion)

```
        Shapiro-Wilk normality test

data:  CD133[gol.factor == "AML"]
W = 0.9121, p-value = 0.2583
```

We use the same notations as above (Exercise 2), $\boldsymbol{x}_{ALL}$ denotes the expression levels of CD133 for all the ALL patients, and $\boldsymbol{x}_{AML}$ denotes the expression levels of CD133 for all the AML patients. $\sigma^2_{ALL}(\sigma^2_{AML})$ is the true variance of the expression level of CD133 for class ALL (AML).
For the ALL class, we have $H_0 : \boldsymbol{x}_{ALL} \sim \mathcal{N}(\mu_{ALL}, \sigma^2_{ALL})$, and $H_1 : \boldsymbol{x}_{AML}$ does not follow a $\mathcal{N}(\mu_{AML}, \sigma^2_{AML})$ (similarly for the AML class). According to the shapiro tests above, the test stastistics is $W = 0.9696$ for the ALL group and $W = 0.9121$ for the AML group. We reject H_0 if the p-value associated to this test is < 0.05. For both classes, we do not reject H_0 and we can therefore make the assumption that the expression levels of CD133 follows a Normal distribution.

(b) *Test for the equality of variances.* *[1pt]*

```
> var.test(CD133 ~ gol.factor, alternative = 'two.sided')

        F test to compare two variances

data:  CD133 by gol.factor
F = 0.4605, num df = 26, denom df = 10, p-value = 0.1095
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.137670 1.192365
sample estimates:
ratio of variances
         0.4604523
```

We use the same notations as in exercise 2 and question a).
$H_0 : \sigma^2_{ALL} = \sigma^2_{AML}$ vs. $H_1 : \sigma_{ALL} \neq \sigma_{AML}$. We use an F statistics to compare the variance between the two groups:

$$F = \frac{s^2_{ALL}}{s^2_{AML}} \sim \mathcal{F}_{(n_{ALL}-1),(n_{AML}-1)} = \mathcal{F}_{26,10}$$

We reject H_0 if $F >$ qf(0.975, 26, 10) = 3.34 or, alternatively, if the p-value associated with this test is < 0.05. According to the F test performed in R, we have $F = 0.4605$ and the p-value = 0.1095, so we do not reject H_0 and a test with unequal variances must be performed to compare the means.

(c) *Test for the equality of the means by an appropriate t-test.* *[1pt]*

```
> t.test(CD133 ~ gol.factor, var.equal = FALSE)

        Welch Two Sample t-test

data:  CD133 by gol.factor
t = -6.7878, df = 13.915, p-value = 9.048e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.6254199 -0.8445301
sample estimates:
mean in group ALL mean in group AML
       -0.8812041         0.3537709
```

See answers from exercise 1, question a) for null and alternative hypothesis, test statistics, distribution, degrees of freedom and rejection region.
We have $T = -6.7878$ with approximation of degrees of freedom for a Welsh's test of 13.915. We have `qt(0.975, 36) = 2.02` and the p-value is $9.048e^6$, therefore, we reject H_0 and there is a significant difference between the means of the ALL and the AML group in the expression levels of CD33.

(d) *Is the experimental effect strong?* *[1pt]*
Yes, given the low value of the p-value obtained for the t-test, we can conclude that there is a strong difference in the expression levels between the two groups.

Exercise 4 *Clustering and visualisation* *[Total mark: 9].*

1. *Gene selection. We are still interested in the list of the 10 candidate genes from Exercise 2 for the Golub study.*

   Create a new data frame from the `golub100` *data set so that the data set subset contains only these genes of interest.*

```
> # create the data frame with the list of genes
> data.sub = golub100[list.gene, ]
> dim(data.sub)

[1] 10 38
```

2. *Hierarchical clustering. Output the heatmap, using the* $1-$*correlation distance and the average linkage. Comment on the clusters.* *[2pt]*

```
> # compute correlations
> genes.cor <- cor(t(data.sub), use="pairwise.complete.obs", method="pearson")
> samples.cor <- cor(data.sub, use="pairwise.complete.obs", method="pearson")
> # compte distances and average linkage
> genes.cor.dist <- as.dist(1-genes.cor)
> genes.tree <- hclust(genes.cor.dist, method='average')
> samples.cor.dist <- as.dist(1-samples.cor)
> samples.tree <- hclust(samples.cor.dist, method='average')

> # plot heatmap
> library("RColorBrewer")
> # 1. set the colors for the gene expression data
> col <- colorRampPalette(brewer.pal(10, "RdBu"))(256)
> col <- rev(col) # reverse colors as indicated above
> # 2. set the colors for the samples
> patientcolors <- ifelse(gol.factor =='AML', 'red', 'blue')
> heatmap(data.sub,scale="row", col=col, Rowv=as.dendrogram(genes.tree),
+ Colv=as.dendrogram(samples.tree), main="Golub data.sub", ColSideColors=patientcolors)
```

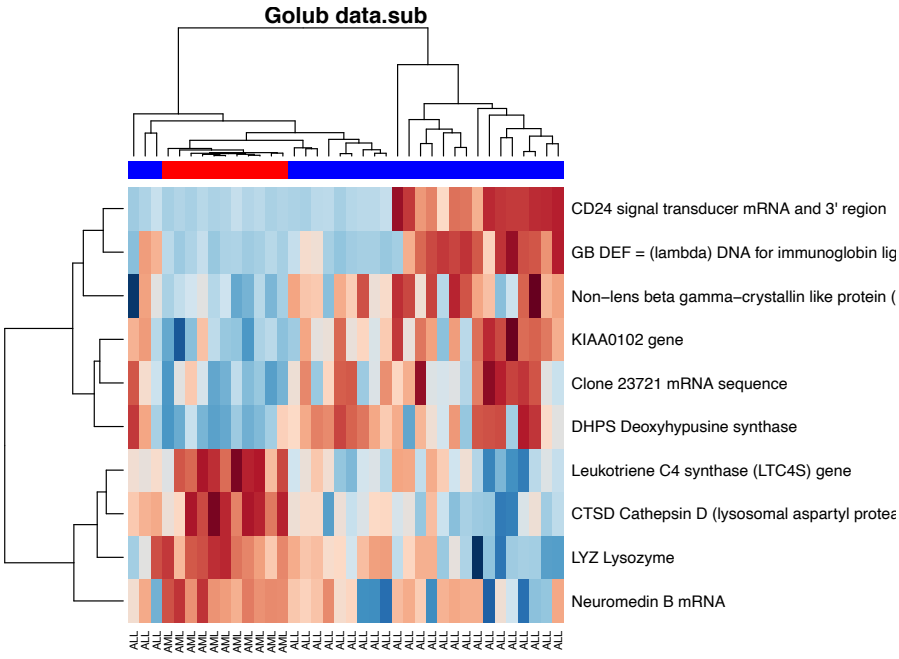

Figure 4: Golub data subset. Hierarchical clustering with 1-correlation distance and average linkage

From the heatmap from Figure 4, we can see that the AML are regrouped with some of the ALL samples. A cluster of the top 6 genes are overexpressed in some of the ALL samples and underexpressed in the mix of AML + ALL samples

(on the left hand side). Globally, we can say that we are not observing a nice clustering of the data, which is unexpected, given the low p-values obtained in exercise 1.

3. *Now, display an heatmap with the Euclidian distance and the Ward linkage. Comment on the differences with the heatmap obtained above.* *[1pt]*

```
> # compute distances and average linkage
> genes.cor.dist <- dist(data.sub)
> genes.tree <- hclust(genes.cor.dist, method='ward')
> samples.cor.dist <- dist(t(data.sub))
> samples.tree <- hclust(samples.cor.dist, method='ward')

> # plot heatmap
> heatmap(data.sub, scale="row", col=col,
+ Rowv=as.dendrogram(genes.tree), Colv=as.dendrogram(samples.tree),
+ main="Golub data.sub", ColSideColors=patientcolors)
```

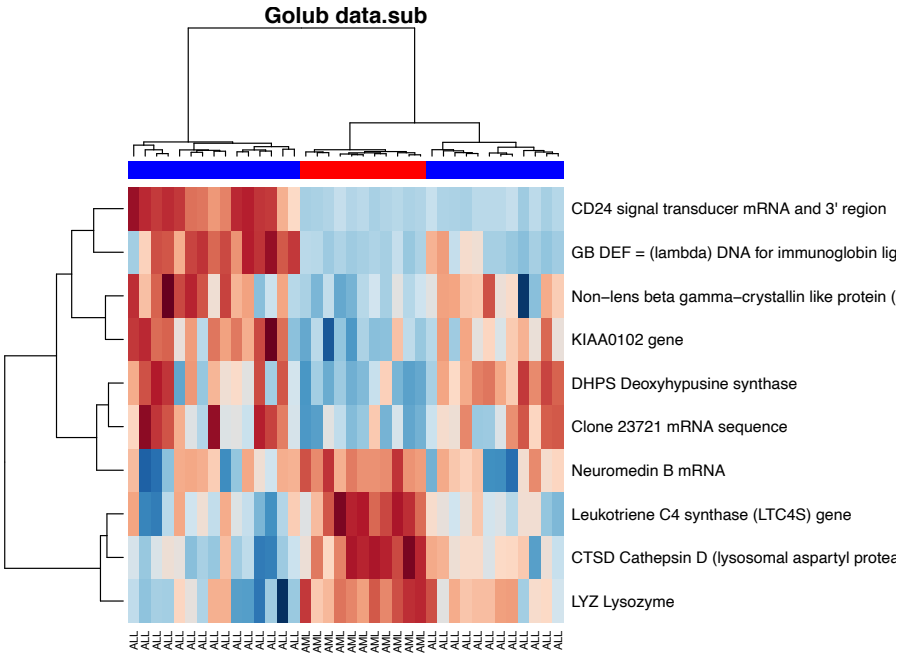

Figure 5: Golub data subset. Hierarchical clustering with Euclidian distance and 'Ward' linkage

The heatmap from Figure 5 is pretty similar to the one obtained above. The AML samples are still mixed with some of the ALL samples. However, this time a cluster of the top 7 genes are overexpressed in some of the ALL samples and underexpressed in the mix of AML +ALL samples (on the right hand side).
We can say that none of these clusterings give very good results.
In the remaining of the exercise, you will transpose the data and check that the number of rows of the data frame is 38 (the number of patients).

```
> #transpose the data
> data.pca = t(data.sub)
> dim(data.pca)

[1] 38 10
```

4. *Principal Component Analysis. Apply a PCA on the data frame (use the arguments center and scale).* *[1pt]*

```
> golub.pca = prcomp(data.pca,center = T, scale. = T)
```

```
 [1] 0.555848711 0.135643443 0.098779061 0.053622140 0.050017011 0.036373304
 [7] 0.025682752 0.022803862 0.012982085 0.008247632
[1] 0.6914922
```

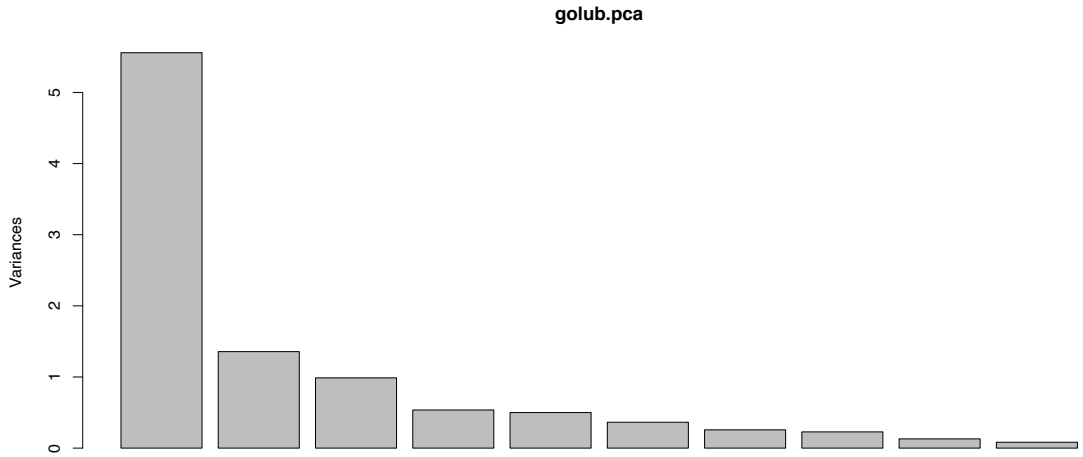

Figure 6: Golub data subset.PCA and screeplot of the amount of variance explained.

(a) Will two components be enough to explain most of the variance in the data? (give some numerical figures). [1pt]

```
> plot(golub.pca)
> # percentage of variance
> golub.pca$sdev^2/sum(golub.pca$sdev^2)
 [1] 0.555848711 0.135643443 0.098779061 0.053622140 0.050017011 0.036373304
 [7] 0.025682752 0.022803862 0.012982085 0.008247632
> #cumulative percentage of variance
> sum(golub.pca$sdev[1:2]^2)/sum(golub.pca$sdev^2)
[1] 0.6914922
```

With two components, the PCA model can explain 69% of the total variance, which is enough for summarizing most of the information.

(b) Output the sample plot on the first two components. The samples (patients) should appear on this plot. Comment. [1pt]

```
> # Setting up the colors for the 3 clusters on the plot:
> my.color.vector <- ifelse(gol.factor == 'AML', 'green', 'blue')
> # Plotting the PC scores:
> par(pty="s")
> plot(golub.pca$x[,1], golub.pca$x[,2], ylim=range(golub.pca$x[,1]),
+      xlab="PC 1", ylab="PC 2", type ='n', lwd=2)
> text(golub.pca$x[,1], golub.pca$x[,2], labels=colnames(data.sub), cex=0.7, lwd=2,
+      col=my.color.vector)
```

Figure 7 shows a nice grouping of the samples on the first principal component. Given the screeplot from Figure 6, we can see that most of the variance is explained on the first component anyway. The meaning of the second component is not that obvious (but the second component is useful for representation purposes).

(c) Comment on the biplot obtained. By plotting the boxplots on some chosen genes, explain the characteristics of the genes of interest with respect to how they are located on the biplot: are they overexpressed / underexpressed in some biological conditions? [2pt]
We can observe two types of genes on the biplot (Figure 8). Those which arrows are pointing towards the ALL samples on the left hand side, and those pointing towards the AML samples on the right hand side. Additional boxplots (Figure 9) show that the direction of the arrows indicate an overexpression of the gene in the sample groups it is pointing to.

```
> biplot(golub.pca, cex = c(1,0.8))
```

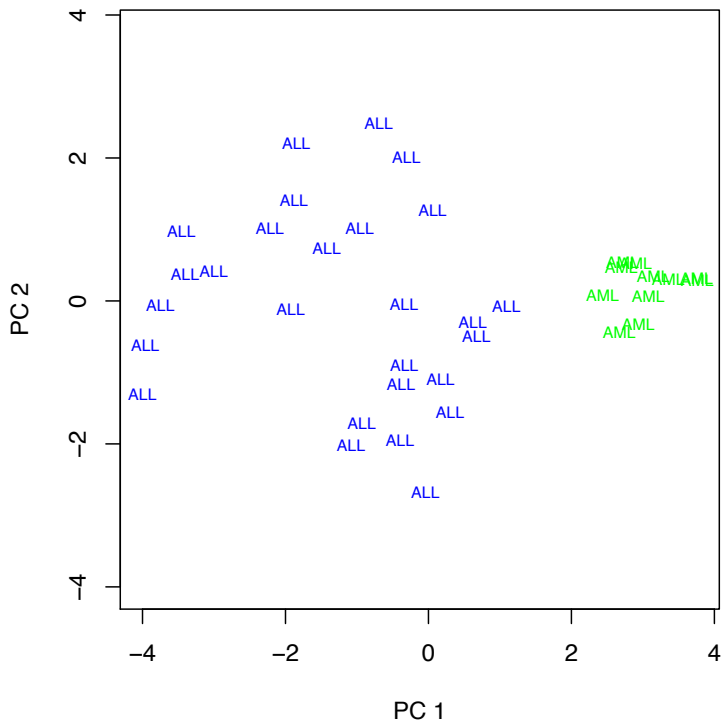

Figure 7: Golub data subset.PCA sample plot on the first two principal components.

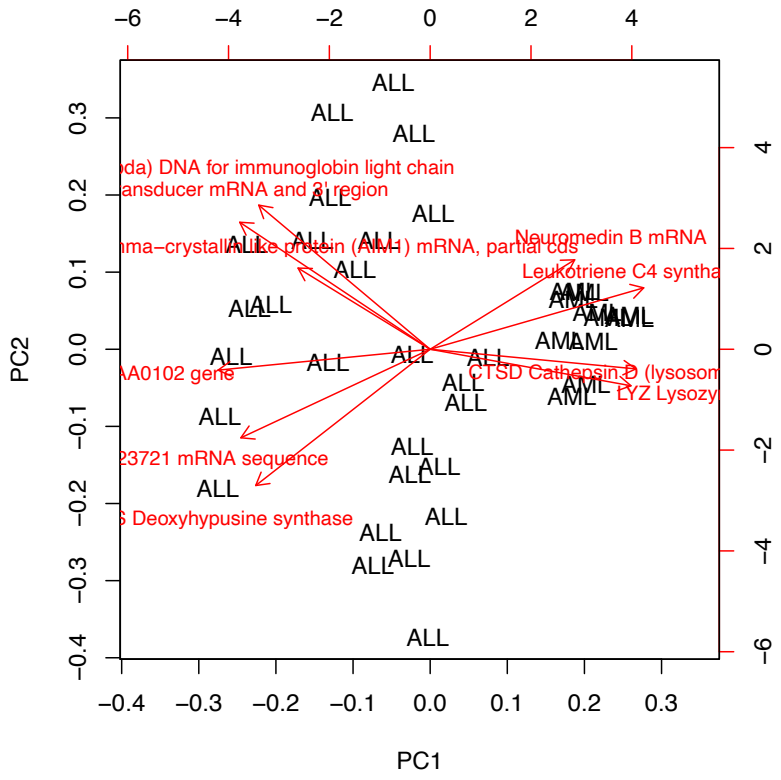

Figure 8: Golub data subset.PCA biplot on the first two principal components.

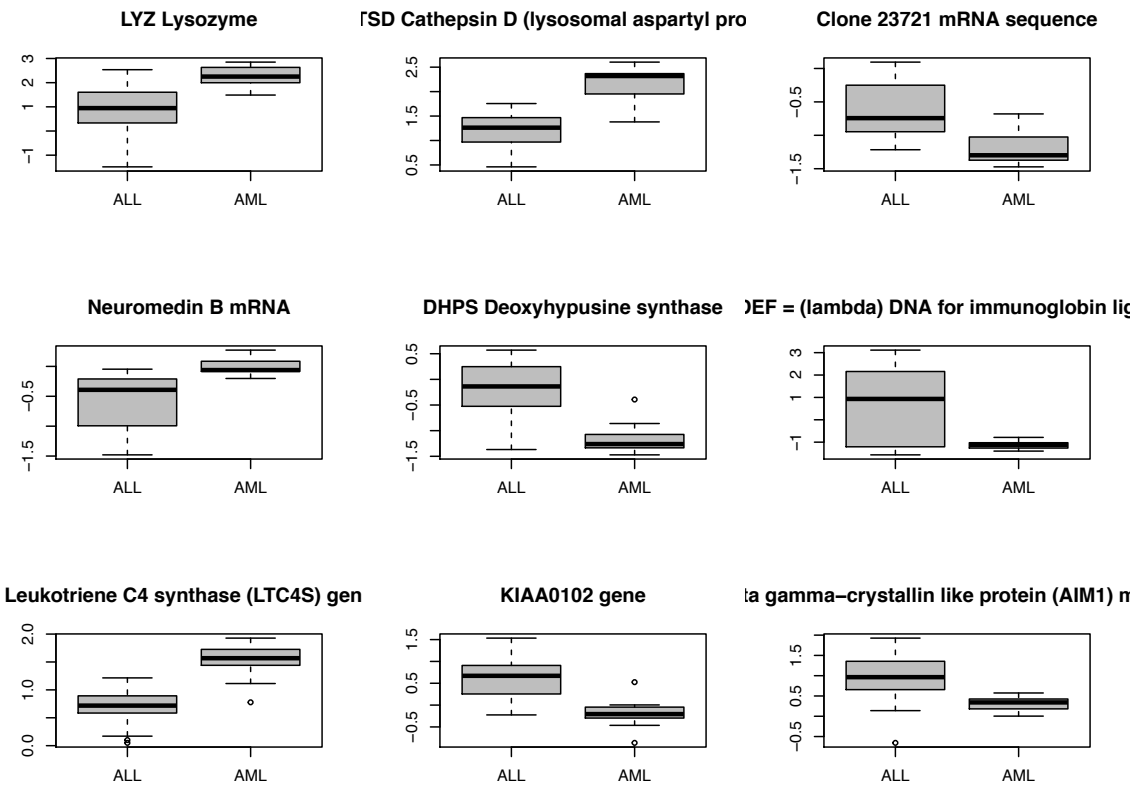

Figure 9: Golub data subset.Boxplots of the genes of interest with respect to the tumour class.

```
> par(mfrow=c(3,3))
> for(i in 1:9){
+   boxplot(data.pca[,i]~gol.factor, main = paste(colnames(data.pca)[i]), col = 'grey')
+ }
```

(d) Compare the clustering of the genes observed on the PCA biplot to the K-means clustering with $k = 2$. *[1pt]*

```
> kmeans.res = kmeans(data.sub, centers =2)
> #kmeans.res$cluster  # outputs the classification
```

We can compare the clustering output from K-means (see Table 2) to the clustering of the PCA on the first component by looking at the sign of the first column of `golub.pca$rotation` *and using the* `ifelse()` *function as follows to store the result.*

```
> which(golub.pca$rotation[,1] >0)
> which(golub.pca$rotation[,1] <0)
> cluster.pca = vector(length = length(list.gene))
> names(cluster.pca) = list.gene
> cluster.pca = ifelse(golub.pca$rotation[,1] >= 0, 2, 1)
```

We can see that two genes differ in their attribution to the clusters between K-means and PCA first component: 'Neuromedin B mRNA' and 'CD24 signal transducer mRNA and 3' region'.

	cluster.kmeans	cluster.pca
LYZ Lysozyme	2	2
CTSD Cathepsin D (lysosomal aspartyl protease)	2	2
Clone 23721 mRNA sequence	1	1
Neuromedin B mRNA	1	2
DHPS Deoxyhypusine synthase	1	1
GB DEF = (lambda) DNA for immunoglobin light chain	1	1
Leukotriene C4 synthase (LTC4S) gene	2	2
KIAA0102 gene	1	1
Non-lens beta gamma-crystallin like protein (AIM1) mRNA, partial cds	2	1
CD24 signal transducer mRNA and 3' region	1	1

Table 2: Clustering outputs from k-means with k = 2 and with PCA on the first principal component