HW7 Solution CS6220-Data Mining

Problem 1

(a)

```
#Load the data
load('hw7.Rdata')
ls()
```

```
## [1] "countsTableFull"   "countsTableSubset"
```

```
summary(countsTableFull)
```

```
##        N8                N33              N51                T8          
##  Min.   :     2.0   Min.   :     1   Min.   :      5   Min.   :     0.0  
##  1st Qu.:   118.0   1st Qu.:   141   1st Qu.:    308   1st Qu.:    90.0  
##  Median :   253.0   Median :   289   Median :    659   Median :   220.0  
##  Mean   :   740.7   Mean   :  1491   Mean   :   2003   Mean   :   681.8  
##  3rd Qu.:   550.0   3rd Qu.:   613   3rd Qu.:   1435   3rd Qu.:   505.0  
##  Max.   :393801.0   Max.   :581364   Max.   :1675945   Max.   :330105.0  
##       T33              T51        
##  Min.   :     0   Min.   :     0  
##  1st Qu.:   172   1st Qu.:   219  
##  Median :   378   Median :   486  
##  Mean   :  1324   Mean   :  1412  
##  3rd Qu.:   828   3rd Qu.:  1090  
##  Max.   :365430   Max.   :633871  
```

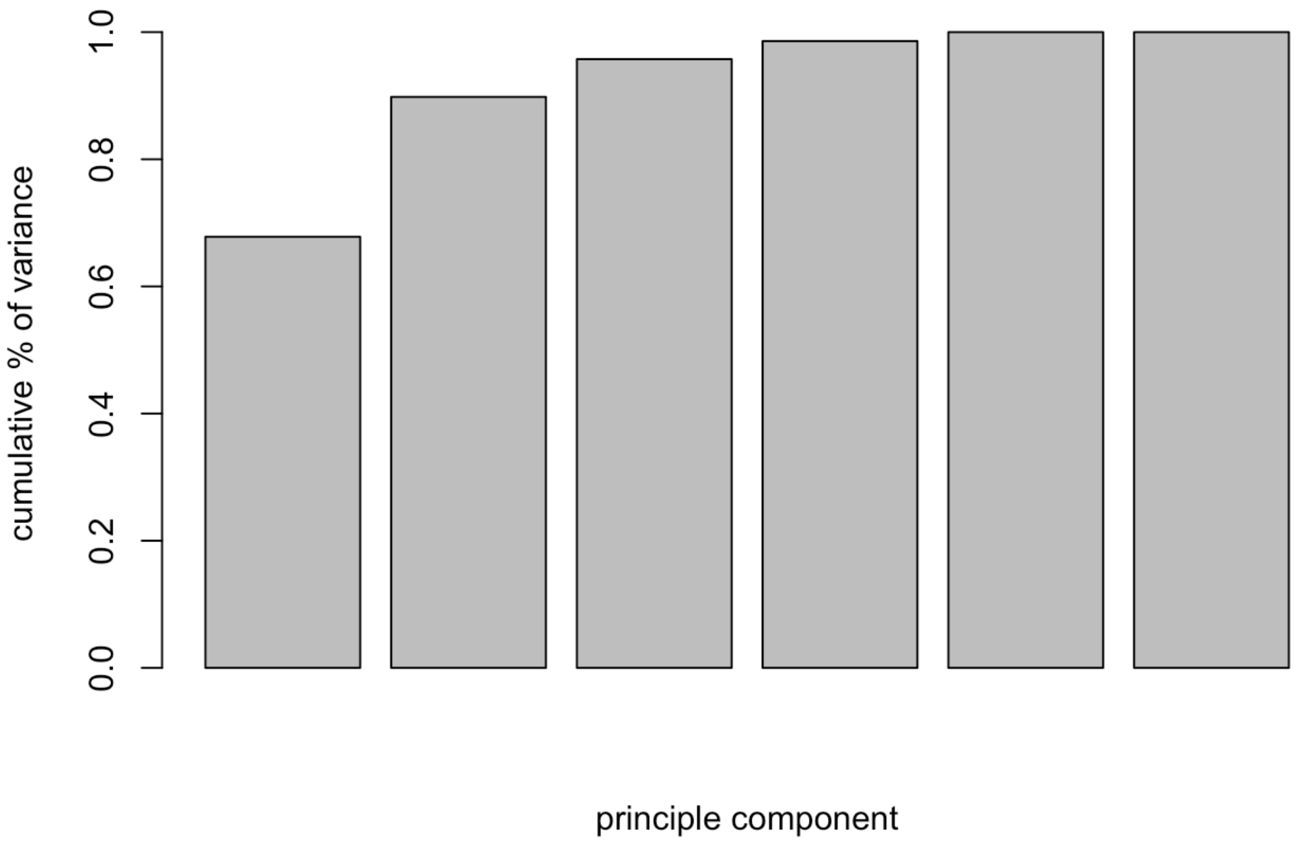
```
head(countsTableFull)
```

```
##              N8   N33   N51  T8  T33  T51
## NM_000014  2242  2285 15121 261  597 1991
## NM_144670 11731 13308  6944 912 3071 1160
## NM_017436   162   111   751 296  362  182
## NM_015665   199   215   512  81  344  342
## NM_023928   470   573   690 710 1112  728
## NM_024666   298   332   856 203  790  909
```

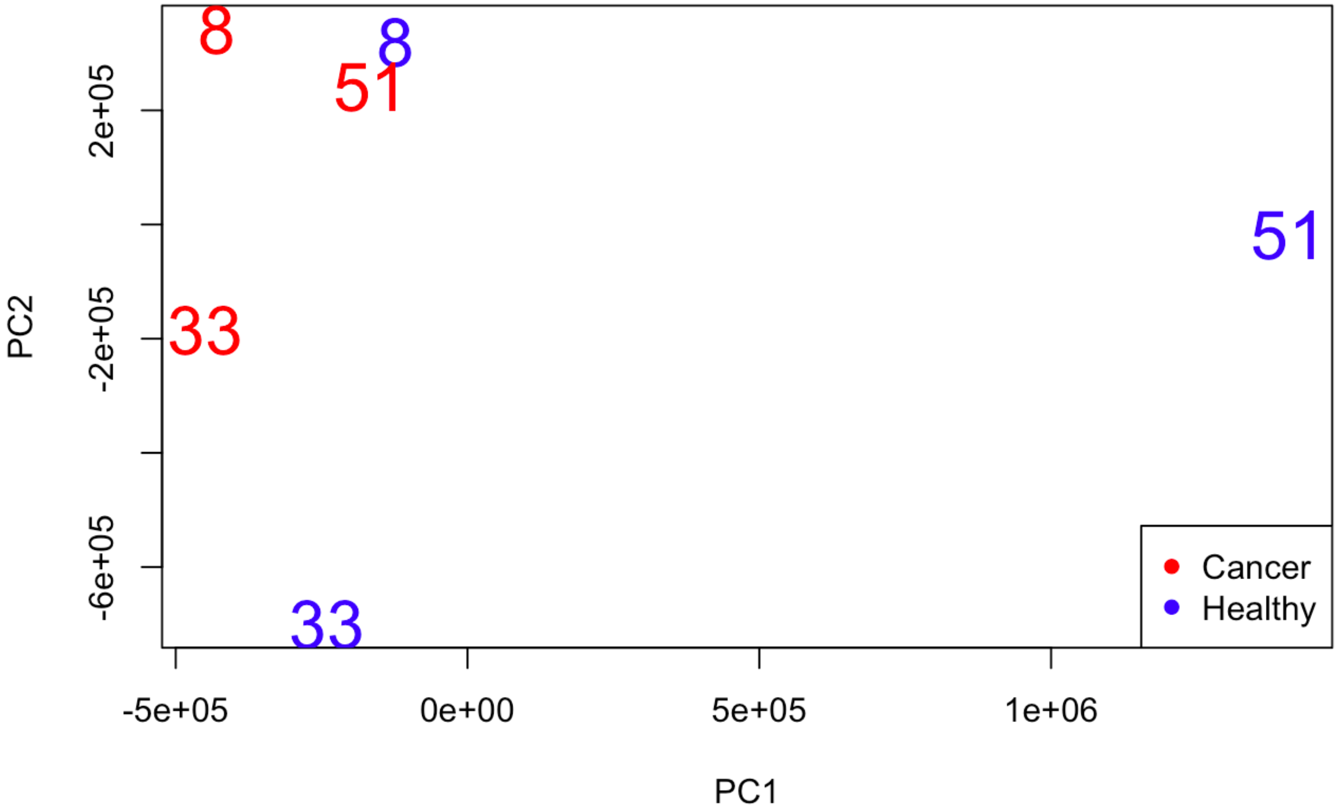
```
#basic Principle Component Analysis without standardization
fullPCA1 <- prcomp(x=t(countsTableFull), center=TRUE, scale.=FALSE)
summary(fullPCA1)
```

```
## Importance of components:
##                              PC1       PC2       PC3       PC4       PC5
## Standard deviation     7.025e+05 4.001e+05 2.080e+05 1.437e+05 1.013e+05
## Proportion of Variance 6.781e-01 2.200e-01 5.943e-02 2.839e-02 1.410e-02
## Cumulative Proportion  6.781e-01 8.981e-01 9.575e-01 9.859e-01 1.000e+00
##                              PC6
## Standard deviation     9.594e-10
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```
barplot( cumsum( fullPCA1$sdev^2/sum(fullPCA1$sdev^2) ) , xlab="principle componen
t", ylab="cumulative % of variance" )
```



```
diseaseStatus <- c(rep("Normal", 3), rep("Cancer",3))
biol.rep <- c(rep(c(8, 33, 51), 2))
myColor <- rep(NA, 6)
myColor[diseaseStatus == "Cancer"] <- "red"
myColor[diseaseStatus == "Normal"] <- "blue"
plot(fullPCA1$x[,1:2], pch=NA, cex=2)
text(fullPCA1$x[,1], fullPCA1$x[,2], biol.rep, col=myColor, cex=2)
legend("bottomright", pch=16, col=c("red", "blue"), c("Cancer", "Healthy"))
```



i. There are 6 PCs, which is the number of the observations (i.e., patien samples). The number of components are limited to the minimum of the number of columns and the number of variables. 6-1=5 components are ‘informative’ (i.e., they represent potentially useful reduction in dimensionality).

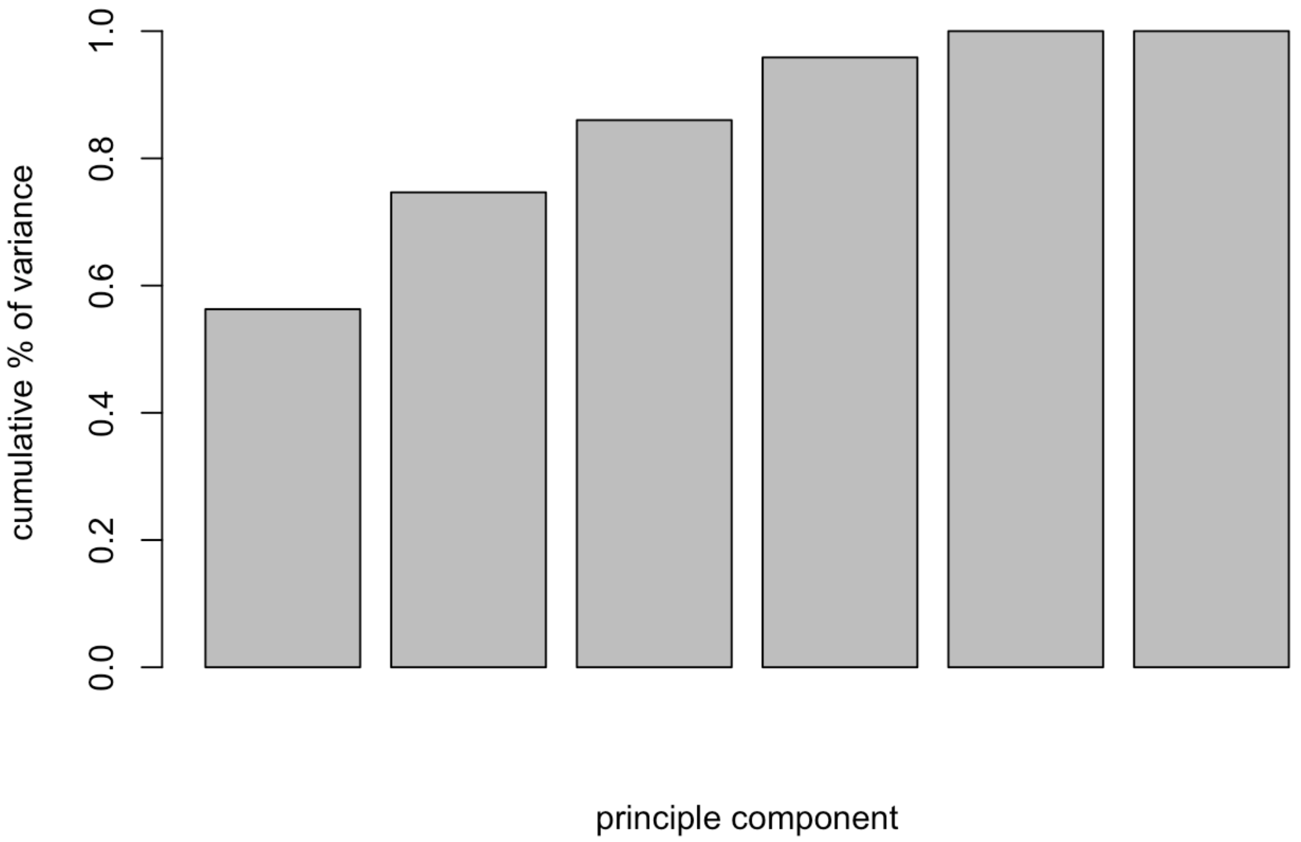
ii. A desirable plot should show a clear distinction between the two clusters of data in at least one of the dimensions of the plotted PCs. Additionaly, the members of a cluster should not be considerably separated from each other. On the PCA plot, the library N51 is seprated from the rest, indicating potential problems with the quality of the measurements.

(b)

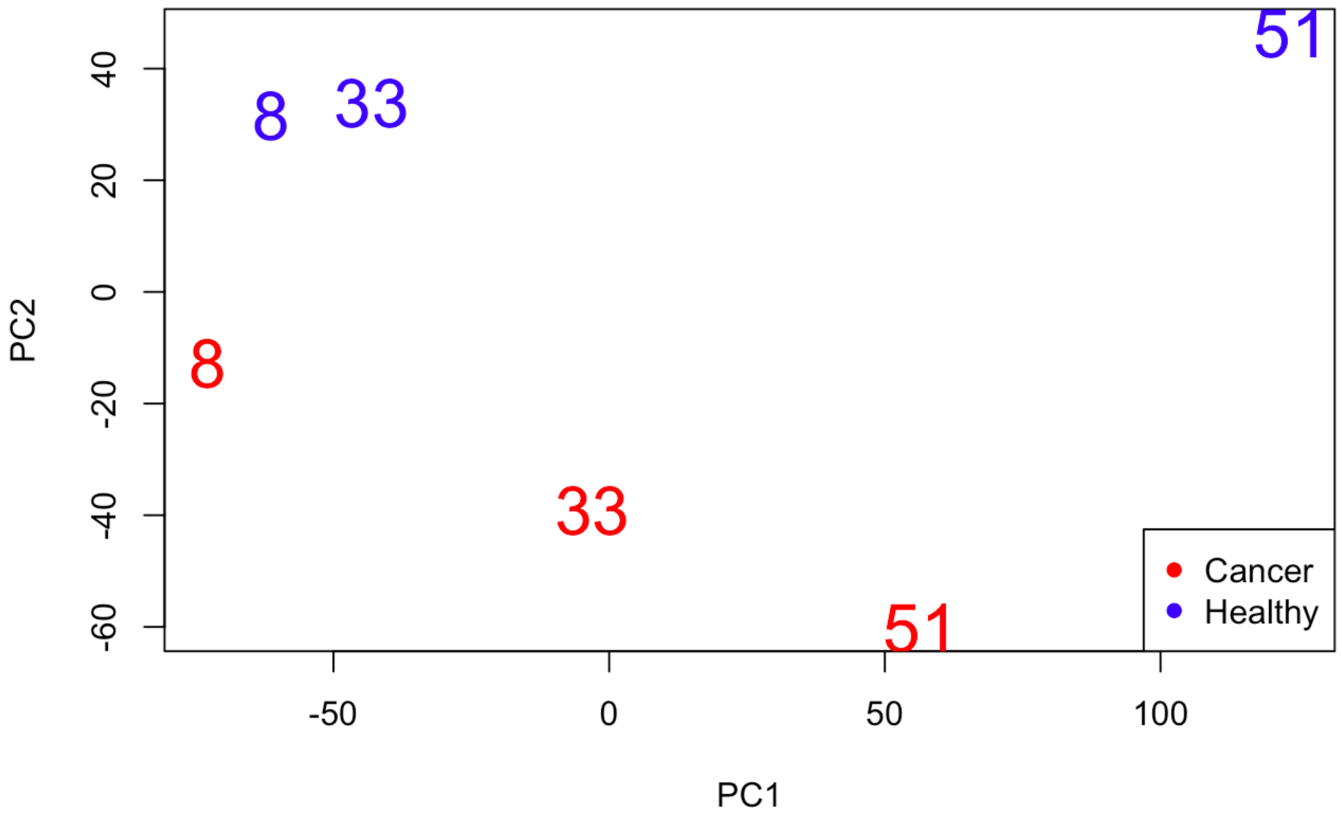
```
fullPCA2 <- prcomp(x=t(countsTableFull), center=TRUE, scale.=TRUE)
summary(fullPCA2)
```

```
## Importance of components:
##                            PC1     PC2     PC3      PC4      PC5       PC6
## Standard deviation     76.7068 43.8169 34.4531 32.09464 20.78593 9.907e-14
## Proportion of Variance  0.5629  0.1837  0.1136  0.09854  0.04133 0.000e+00
## Cumulative Proportion   0.5629  0.7466  0.8601  0.95867  1.00000 1.000e+00
```

```
barplot( cumsum( fullPCA2$sdev^2/sum(fullPCA2$sdev^2) ) , xlab="principle componen
t", ylab="cumulative % of variance" )
```



```
plot(fullPCA2$x[,1:2], pch=NA, cex=2)
text(fullPCA2$x[,1], fullPCA2$x[,2], biol.rep, col=myColor, cex=2)
legend("bottomright", pch=16, col=c("red", "blue"), c("Cancer", "Healthy"))
```



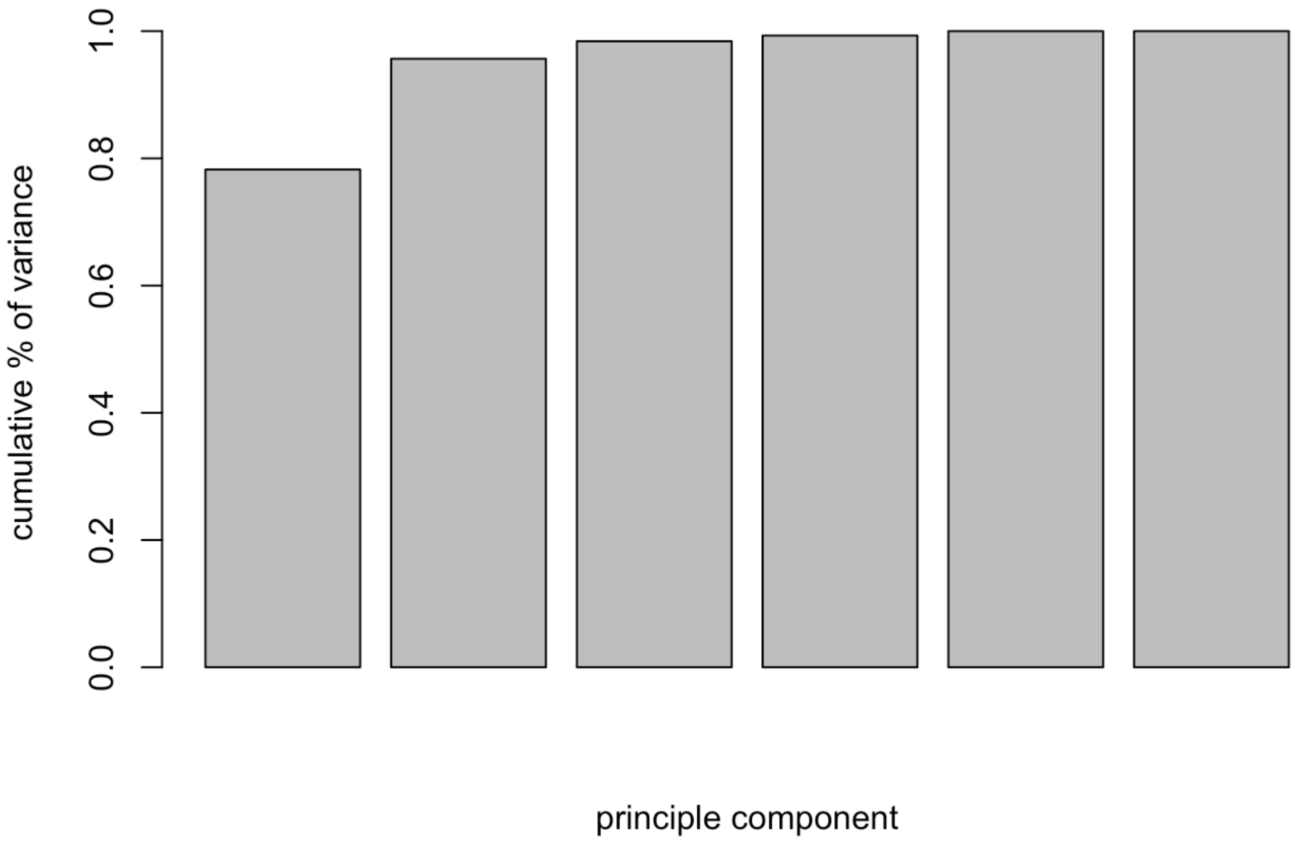
Scaling and centering the data improve the sepration of clusters in the score plots. However, the percentage of variation explained by the first two Principal Components decreased with standardization. Because it eliminated a relatively strong and systematic source of variation incorporated into the first few components.

(c)

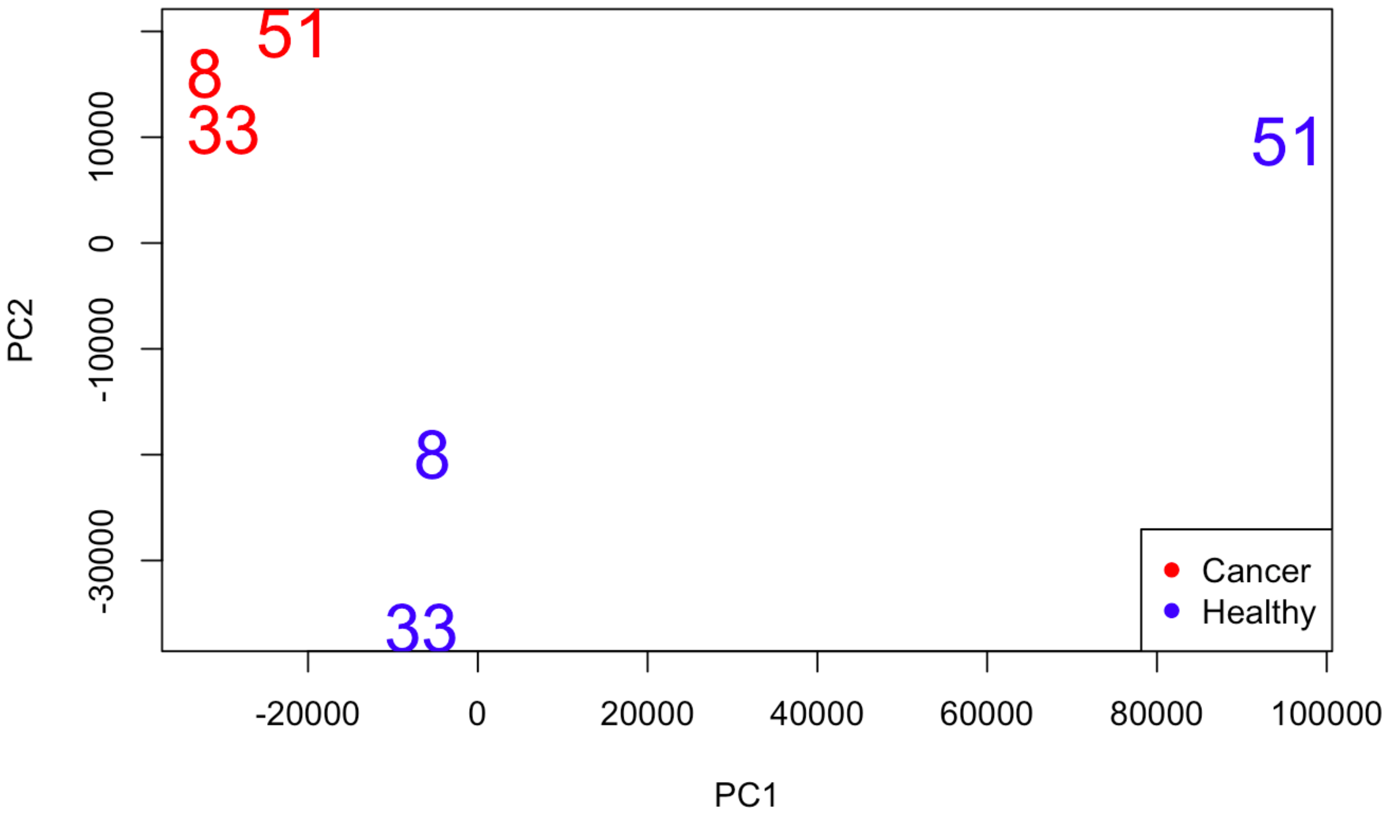
```
subPCA1 <- prcomp(x=t(countsTableSubset), center=TRUE, scale.=FALSE)
summary(subPCA1)
```

```
## Importance of components:
##                              PC1       PC2       PC3       PC4      PC5
## Standard deviation     4.815e+04 2.271e+04 9.040e+03 5.141e+03 4.53e+03
## Proportion of Variance 7.825e-01 1.741e-01 2.758e-02 8.920e-03 6.93e-03
## Cumulative Proportion  7.825e-01 9.566e-01 9.841e-01 9.931e-01 1.00e+00
##                              PC6
## Standard deviation     1.172e-11
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```
barplot( cumsum( subPCA1$sdev^2/sum(subPCA1$sdev^2) ) , xlab="principle componen
t", ylab="cumulative % of variance" )
```



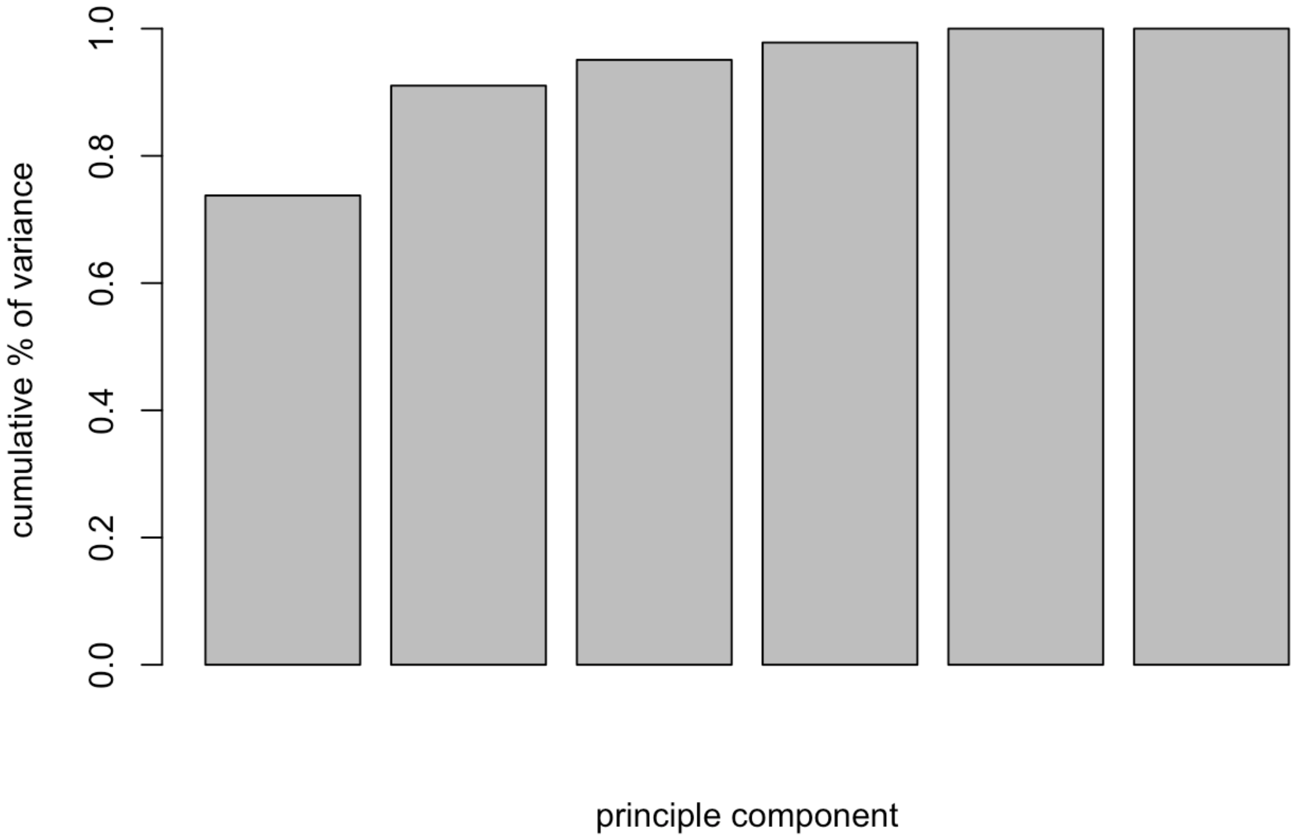
```
plot(subPCA1$x[,1:2], pch=NA, cex=2)
text(subPCA1$x[,1], subPCA1$x[,2], biol.rep, col=myColor, cex=2)
legend("bottomright", pch=16, col=c("red", "blue"), c("Cancer", "Healthy"))
```



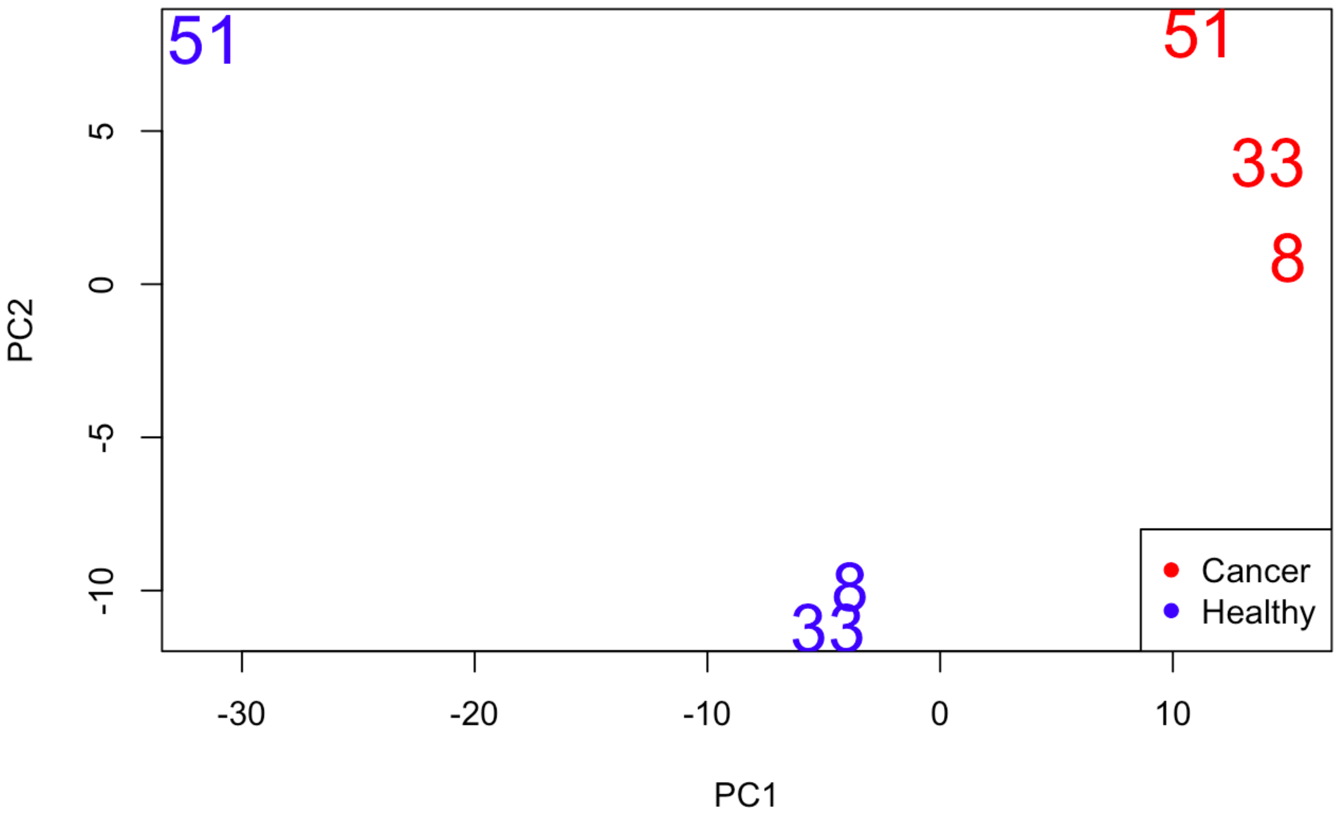
```
subPCA2 <- prcomp(x=t(countsTableSubset), center=TRUE, scale.=TRUE)
summary(subPCA2)
```

```
## Importance of components:
##                            PC1    PC2     PC3    PC4     PC5       PC6
## Standard deviation     17.7897 8.6068 4.17155 3.4163 3.06252 4.401e-15
## Proportion of Variance  0.7377 0.1727 0.04056 0.0272 0.02186 0.000e+00
## Cumulative Proportion   0.7377 0.9104 0.95093 0.9781 1.00000 1.000e+00
```

```
barplot( cumsum( subPCA2$sdev^2/sum(subPCA2$sdev^2) ) , xlab="principle componen
t", ylab="cumulative % of variance" )
```



```
plot(subPCA2$x[,1:2], pch=NA, cex=2)
text(subPCA2$x[,1], subPCA2$x[,2], biol.rep, col=myColor, cex=2)
legend("bottomright", pch=16, col=c("red", "blue"), c("Cancer", "Healthy"))
```



Limiting the descriptors of each sample to the ‘active’ genes was useful. The first two principle components explained a larger proportion of variation, and better reflected the separation between the underlying groups of samples.

The standardization did not have much effect in this particular dataset. This is because all the genes are quantified roughly on a same scale. The standardization would have had a bigger effect on a dataset where the descriptors difffer dramatically in scale.

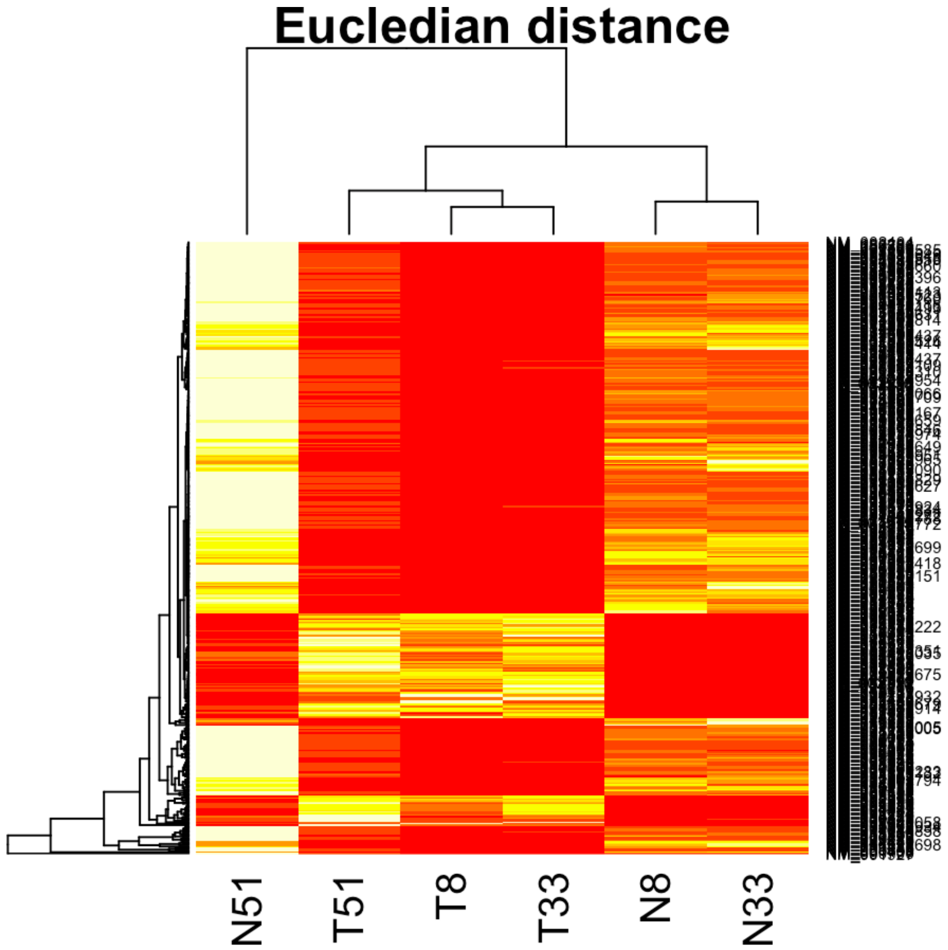
(d)

The biggest difference between tehse results is the use of the variables (i.e., descriptors of each sample). Adding more noisy predictors reduces the effectiveness of the dimension reduction. The standardization did not have much effect because all the genes were quantified roughly on a same scale.

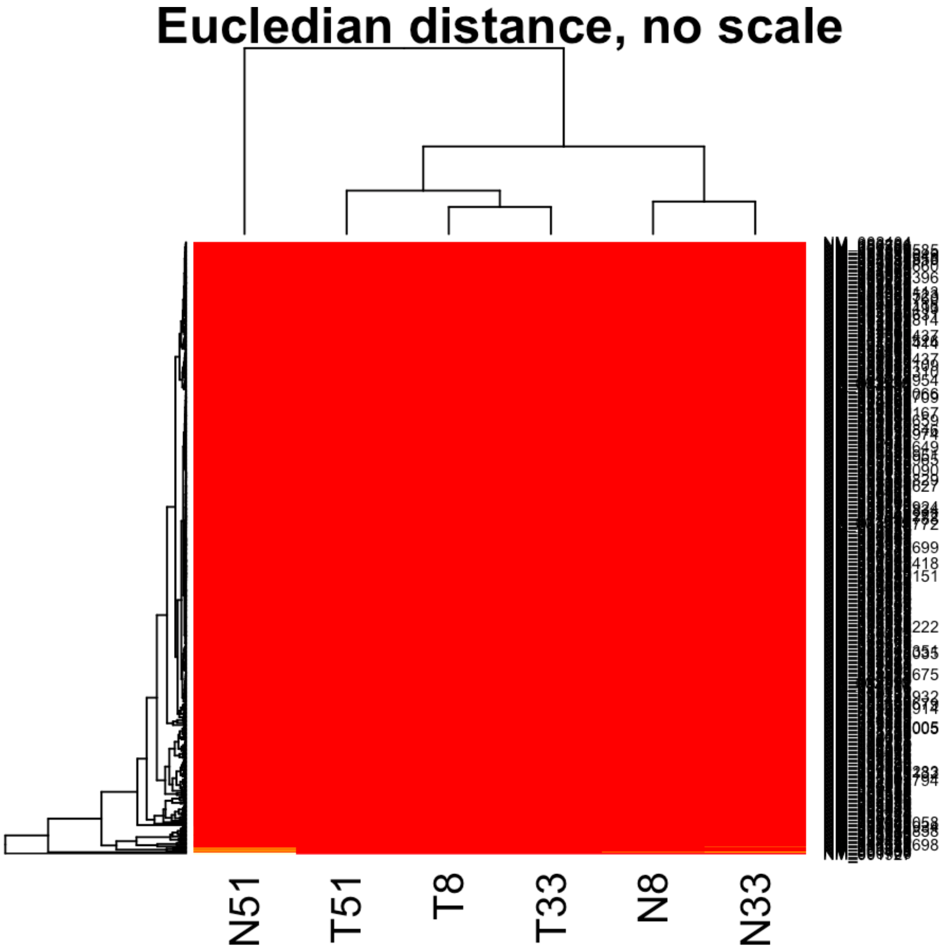
Problem 2

(a)

```
#default setting
heatmap(countsTableSubset, main="Eucledian distance")
```



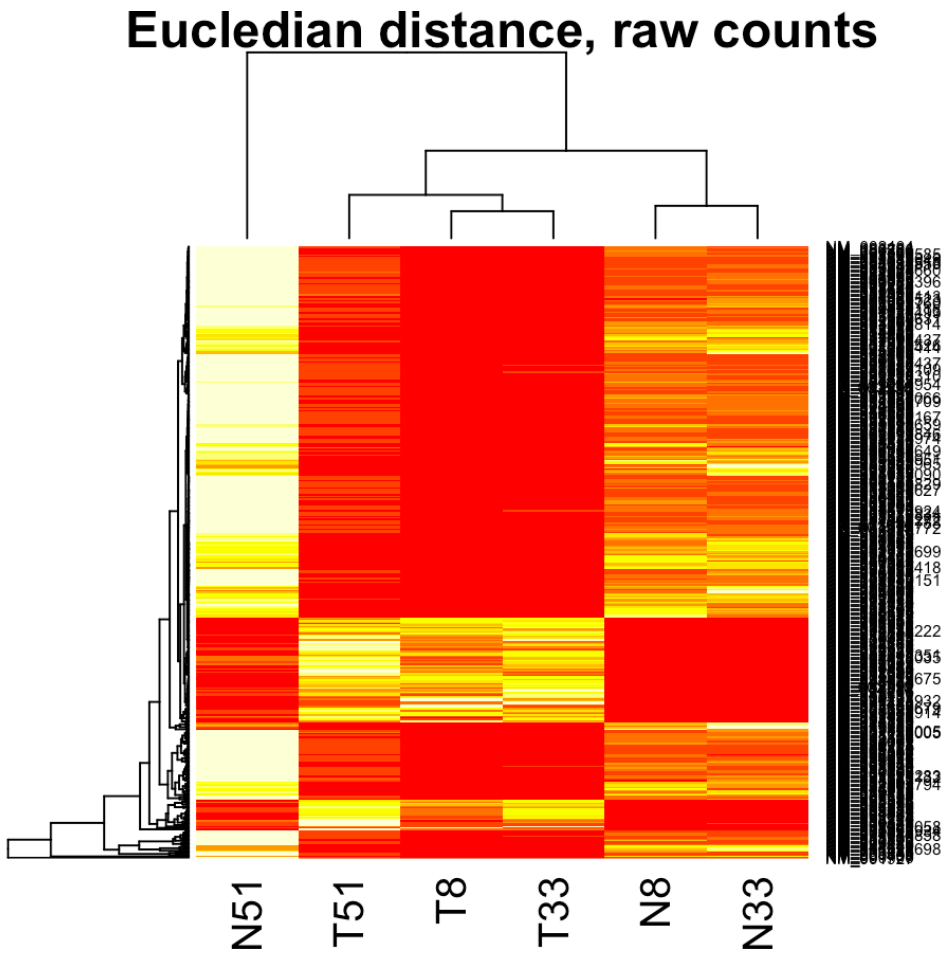
```
#scale="none"
heatmap(countsTableSubset, scale="none", main="Eucledian distance, no scale")
```



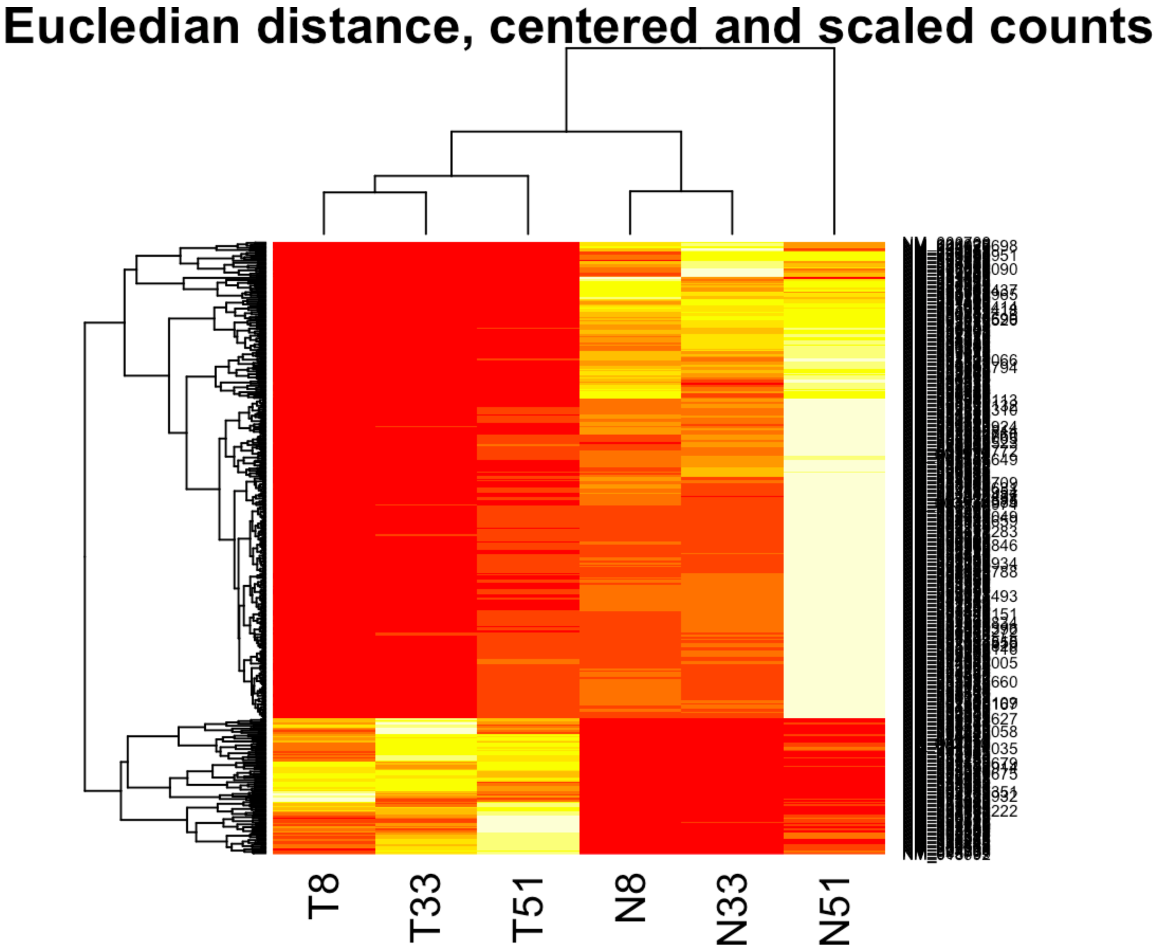
With the scaling option, the color range for each row (here, for each gene) is arranged separately. Without this option, all values of all the variables are represented on the same color scale. When the color is not scaled, extreme values in one variable can dominant the color in all other columns and make the plot less imformative.

(b)

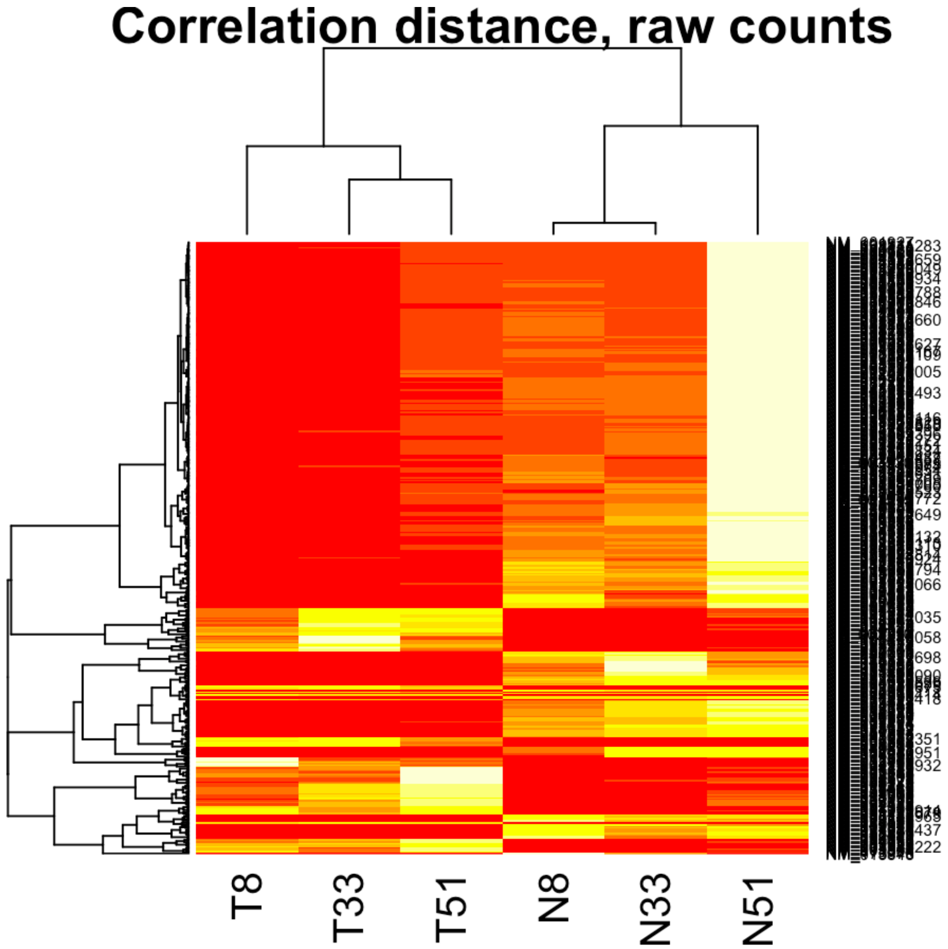
```
suppressMessages(library(bioDist, quietly = T))
centeredScaledData <- t(scale(t(countsTableSubset)))
# Alternatively:
#library(genefilter)
#centeredScaledData = (countsTableSubset - rowMeans(countsTableSubset)) / rowSds(c
ountsTableSubset)

heatmap(countsTableSubset, main="Eucledian distance, raw counts")
```



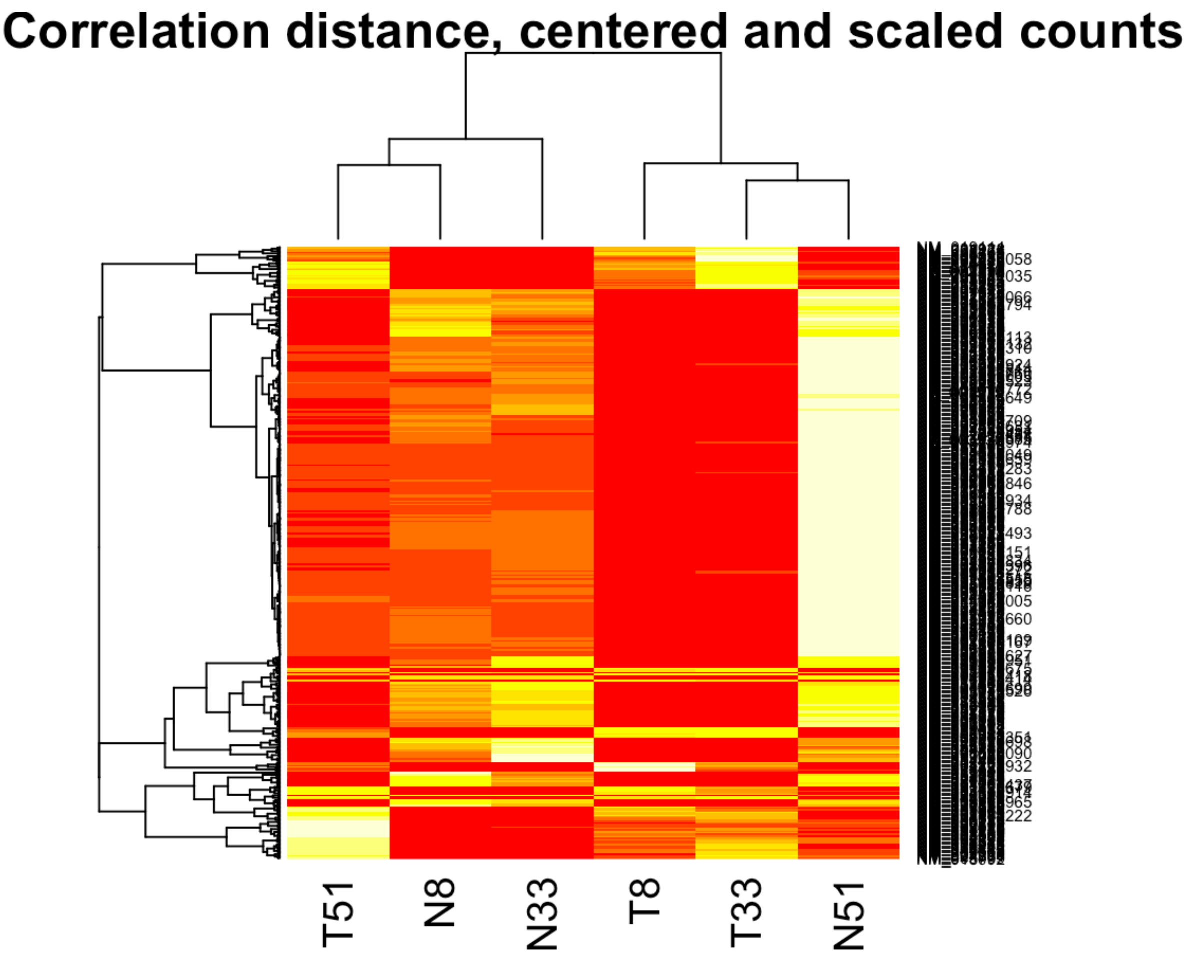
```
heatmap(centeredScaledData, main="Eucledian distance, centered and scaled counts")
```



```
heatmap(countsTableSubset, distfun = cor.dist, main="Correlation distance, raw cou
nts")
```



```
heatmap(centeredScaledData, distfun = cor.dist, main="Correlation distance, center
ed and scaled counts")
```



(c)

Eucledian distance, with centered and scaled predictors, gave best results (the differences in the patterns is most pronounced, and samples of same type are co-clustered).

Problem 3

(a)

```
kmfull <- kmeans(t(countsTableFull), 2)
kmfull$cluster
```

```
##   N8  N33  N51   T8  T33  T51
##    1    1    2    1    1    1
```

```
kmfull2 <- kmeans(scale(t(countsTableFull)), 2)
kmfull2$cluster
```

```
##   N8  N33  N51   T8  T33  T51
##    1    1    2    1    1    2
```

Kmean doesnot show a good clustering result. There is likely a measurement problen with patient 51.

(b)

```
kmsub <- kmeans(t(countsTableSubset), 2)
kmsub$cluster
```

```
##  N8 N33 N51  T8 T33 T51
##   2   2   1   2   2   2
```

```
kmsub2 <- kmeans(scale(t(countsTableSubset)), 2)
kmsub2$cluster
```

```
##  N8 N33 N51  T8 T33 T51
##   1   1   2   1   1   1
```

On the subset dataset, Kmean still couldn’t cluster the data correctly. However, when this subset data is scaled, kmean sould cluster the data correctly.

(c) Scaling the data could be usuful before applying clustering algorithms. However, Kmeans was not good in gneral for this dataset because of its outliers. Depending on the dataset, some other algorithms might work better.