Analysis of gene expression

Job Maathuis

9-3-2022

Exploratory Data Analysis

The data consists of gene expression data which is available as raw count data in a single text file. For this reason the read.table() function can be used and no merging is needed. The column names were changed by using the 'patient_codes.csv'.

```
setwd('C:/Users/jobma/Documents/School/Bio-informatica/Jaar_2/Kwartaal_3/practicum/')
data <- read.table('counts.txt', header = T, row.names = 1)
patient_codes <- read.csv('patient_codes.csv', header=F)
patient_codes[,2] <- gsub('-.*-', '-', patient_codes[,2])  # removes the middle part
colnames(data) <- patient_codes[,2][patient_codes[,1] == colnames(data)]
name_dataset <- 'GSE181032'
```

The dataset has the following structure

```
library(pander)
pander(head(data, 5))
```

Table 1: Table continues below

	m29-TR1	m42-TR1	m43-TR1	s10-TR1	s11-TR1	m29-TR2
A1BG	0	0	2	0	1	0
A2M	1	2	1	1	0	0
A2ML1	0	0	0	0	0	1
A4GALT	0	0	0	0	0	0
AAAS	8	6	4	9	8	3

Table 2: Table continues below

	m42-TR2	m43-TR2	s10-TR2	s11-TR2	m29-TR3	m42-TR3
A1BG	0	0	0	0	0	0
A2M	1	0	1	1	3	1
A2ML1	0	0	0	0	0	0
A4GALT	0	0	0	0	0	0
AAAS	12	10	10	19	16	15

	m43-TR3	s10-TR3	s11-TR3
A1BG	1	1	0
A2M	1	0	0
A2ML1	0	0	0
A4GALT	0	0	0
AAAS	15	9	12

```
print(dim(data))
```

```
## [1] 16282    15
```

```
print(str(data))
```

```
## 'data.frame':    16282 obs. of  15 variables:
##  $ m29-TR1: int  0 1 0 0 8 2 0 0 21 163 ...
##  $ m42-TR1: int  0 2 0 0 6 1 0 0 12 177 ...
##  $ m43-TR1: int  2 1 0 0 4 0 0 0 9 159 ...
##  $ s10-TR1: int  0 1 0 0 9 1 0 0 13 163 ...
##  $ s11-TR1: int  1 0 0 0 8 1 0 0 18 133 ...
##  $ m29-TR2: int  0 0 1 0 3 3 0 0 18 145 ...
##  $ m42-TR2: int  0 1 0 0 12 3 0 0 14 180 ...
##  $ m43-TR2: int  0 0 0 0 10 1 0 0 17 151 ...
##  $ s10-TR2: int  0 1 0 0 10 1 0 0 19 179 ...
##  $ s11-TR2: int  0 1 0 0 19 2 0 0 30 238 ...
##  $ m29-TR3: int  0 3 0 0 16 0 0 0 20 159 ...
##  $ m42-TR3: int  0 1 0 0 15 1 0 0 19 173 ...
##  $ m43-TR3: int  1 1 0 0 15 1 0 0 17 142 ...
##  $ s10-TR3: int  1 0 0 0 9 3 0 0 12 171 ...
##  $ s11-TR3: int  0 0 0 0 12 0 0 0 23 176 ...
## NULL
```

The dataset can be divided, some variables can be made to distinguish between groups

```
m29 <- c(1, 6, 11)
m42 <- c(2, 7, 12)
m43 <- c(3, 8, 13)
m.all <- c(m29, m42, m43)

s10 <- c(4, 9, 14)
s11 <- c(5, 10, 15)
s.all <- c(s10, s11)
```

To get a quick look at the data the summary() function is used. With this function the minimum, first quartile, median, mean, third quartile and the maximum values are obtained.

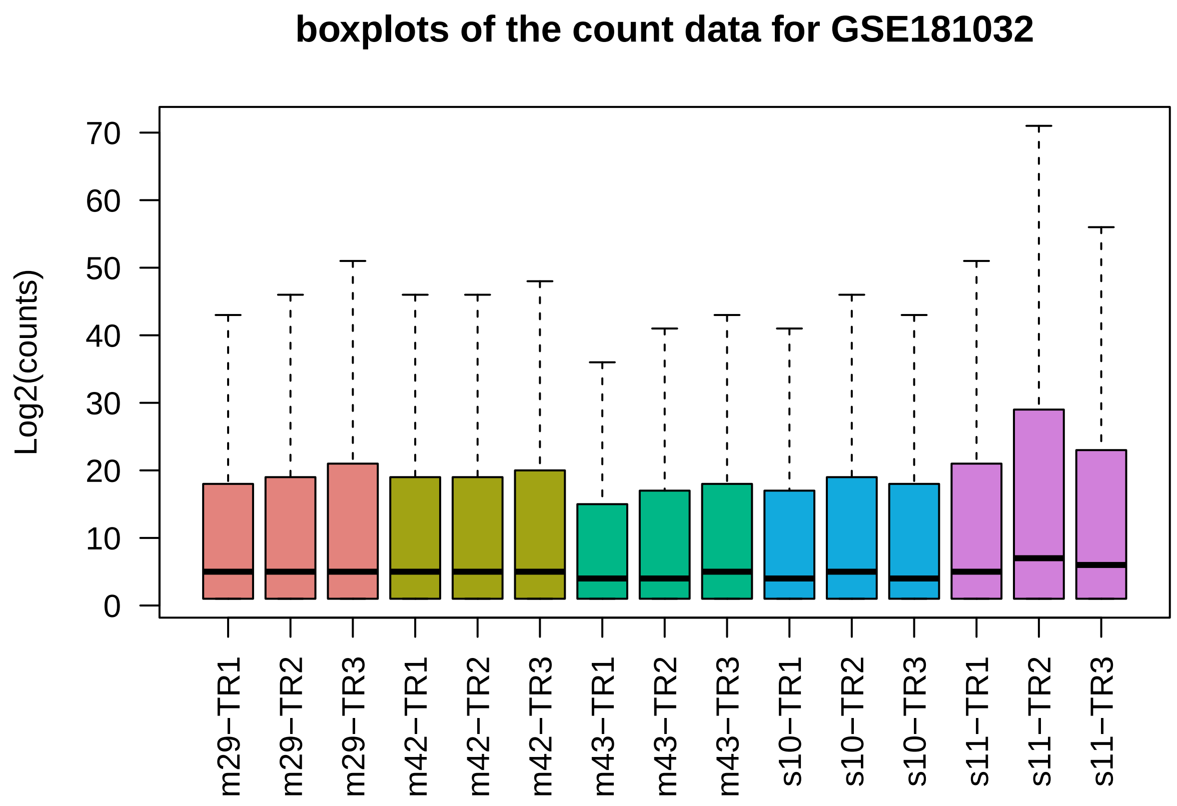
```
summary(data)
```

```
##      m29-TR1            m42-TR1            m43-TR1            s10-TR1
##  Min.   :    0.00   Min.   :    0.00   Min.   :    0.00   Min.   :    0.00
##  1st Qu.:    0.00   1st Qu.:    0.00   1st Qu.:    0.00   1st Qu.:    0.00
##  Median :    4.00   Median :    4.00   Median :    3.00   Median :    3.00
##  Mean   :   34.36   Mean   :   35.55   Mean   :   30.43   Mean   :   32.75
##  3rd Qu.:   17.00   3rd Qu.:   18.00   3rd Qu.:   14.00   3rd Qu.:   16.00
##  Max.   :11538.00   Max.   :11394.00   Max.   :10072.00   Max.   :11121.00
##      s11-TR1            m29-TR2            m42-TR2            m43-TR2
##  Min.   :    0.0   Min.   :    0.00   Min.   :    0.00   Min.   :    0.00
##  1st Qu.:    0.0   1st Qu.:    0.00   1st Qu.:    0.00   1st Qu.:    0.00
##  Median :    4.0   Median :    4.00   Median :    4.00   Median :    3.00
##  Mean   :   35.5   Mean   :   36.68   Mean   :   37.52   Mean   :   34.99
##  3rd Qu.:   20.0   3rd Qu.:   18.00   3rd Qu.:   18.00   3rd Qu.:   16.00
##  Max.   :10663.0   Max.   :12369.00   Max.   :12465.00   Max.   :11447.00
##      s10-TR2            s11-TR2            m29-TR3            m42-TR3
##  Min.   :    0.00   Min.   :    0.00   Min.   :    0.0   Min.   :    0.00
##  1st Qu.:    0.00   1st Qu.:    0.00   1st Qu.:    0.0   1st Qu.:    0.00
##  Median :    4.00   Median :    6.00   Median :    4.0   Median :    4.00
##  Mean   :   36.32   Mean   :   47.34   Mean   :   40.7   Mean   :   37.62
##  3rd Qu.:   18.00   3rd Qu.:   28.00   3rd Qu.:   20.0   3rd Qu.:   19.00
##  Max.   :12091.00   Max.   :13122.00   Max.   :13664.0   Max.   :12547.00
##      m43-TR3            s10-TR3            s11-TR3
##  Min.   :    0.0   Min.   :    0.00   Min.   :    0.00
##  1st Qu.:    0.0   1st Qu.:    0.00   1st Qu.:    0.00
##  Median :    4.0   Median :    3.00   Median :    5.00
##  Mean   :   35.6   Mean   :   34.92   Mean   :   40.61
##  3rd Qu.:   17.0   3rd Qu.:   17.00   3rd Qu.:   22.00
##  Max.   :12015.0   Max.   :11651.00   Max.   :12333.00
```

In the data above a large difference between the maximum values and the other values can be observed. This is probably due to the fact that the data contains a lot of zeros or low values.

Next a boxplot can be made to visualise the data.

```
library(scales)
colors <- hue_pal(c=70)(5)
boxplot((data[,c(m29, m42, m43, s10, s11)] + 1),
        main = 'boxplots of the count data for GSE181032',
        ylab = 'Log2(counts)', outline = F, las = 2,
        col=rep(colors, each=3))
```



For every patient the repeats shows the same distrubution, this is also the case if the patients are compared. A somewhat larger deviation can be seen in s11-TR2 data.

The distrubtion of the count data can also be visualised using a density plot.

```
library(affy)
```

```
## Loading required package: BiocGenerics
```

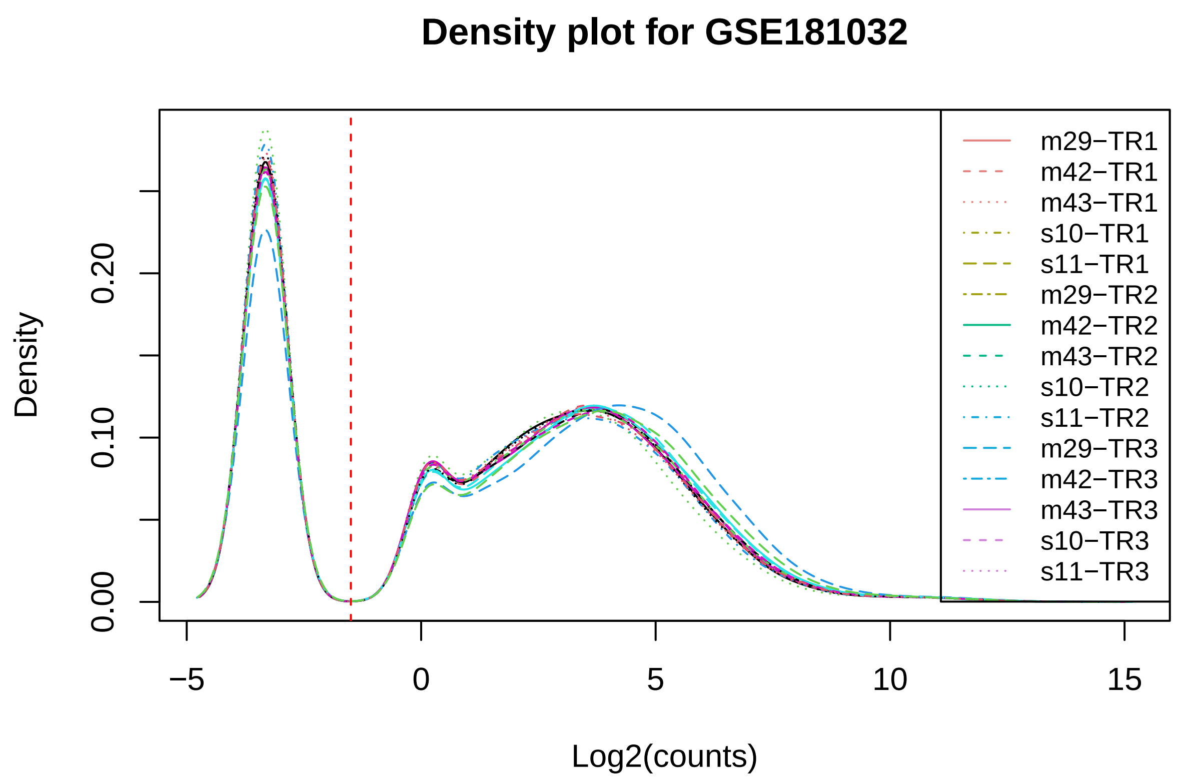
```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
```

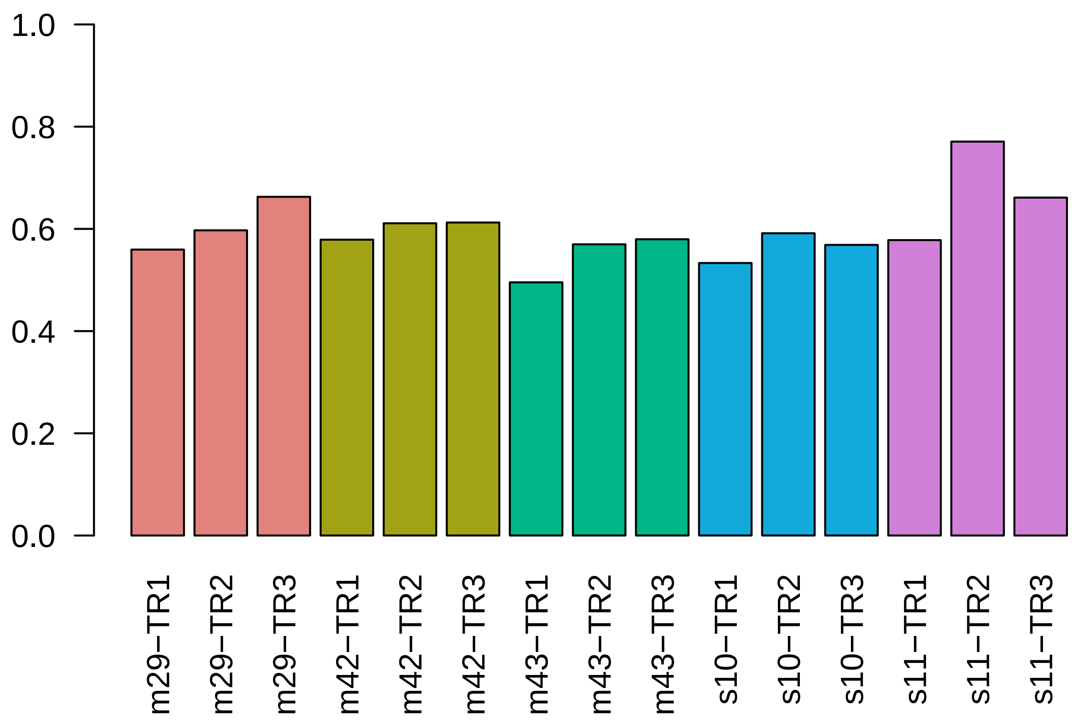
```
## Loading required package: Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
plotDensity(log2(data + 0.1),  main = 'Density plot for GSE181032',
       xlab = 'Log2(counts)', ylab = 'Density')
legend('topright', names(data), lty=c(1:ncol(data)),
       cex = 0.8, col=rep(colors, each=3))
abline(v=-1.5, lwd=1, col='red', lty=2)
```



All of the lines follow the same distrubution, where only m29-TR3 is a bit variated from the other data. This variation is not much, so this will probably not cause any big problems.

```
barplot(colSums(data[,c(m29, m42, m43, s10, s11)]) / 1e6, las = 2,
        col=rep(colors, each=3), ylim = c(0, 1))
```



```
library("DESeq2")

(ddsMat <- DESeqDataSetFromMatrix(countData = data,
                                  colData = data.frame(samples = names(data)),
                                  design = ~ 1))
```

```
## class: DESeqDataSet
## dim: 16282 15
## metadata(1): version
## assays(1): counts
## rownames(16282): A1BG A2M ... ZYX ZZEF1
## rowData names(0):
## colnames(15): m29-TR1 m42-TR1 ...  s10-TR3 s11-TR3
## colData names(1): samples
```

```
# DESeq2 will construct a SummarizedExperiment object and combine this
# into a 'DESeqDataSet' object. The 'design' argument usually indicates the
# experimental design using the condition(s) names as a 'factor', for now we use just '~ 1'

# Perform normalization
rld.dds <- vst(ddsMat)
# 'Extract' normalized values
rld <- assay(rld.dds)
# transposes and calculate distances
sampledists <- dist(t(rld))
```

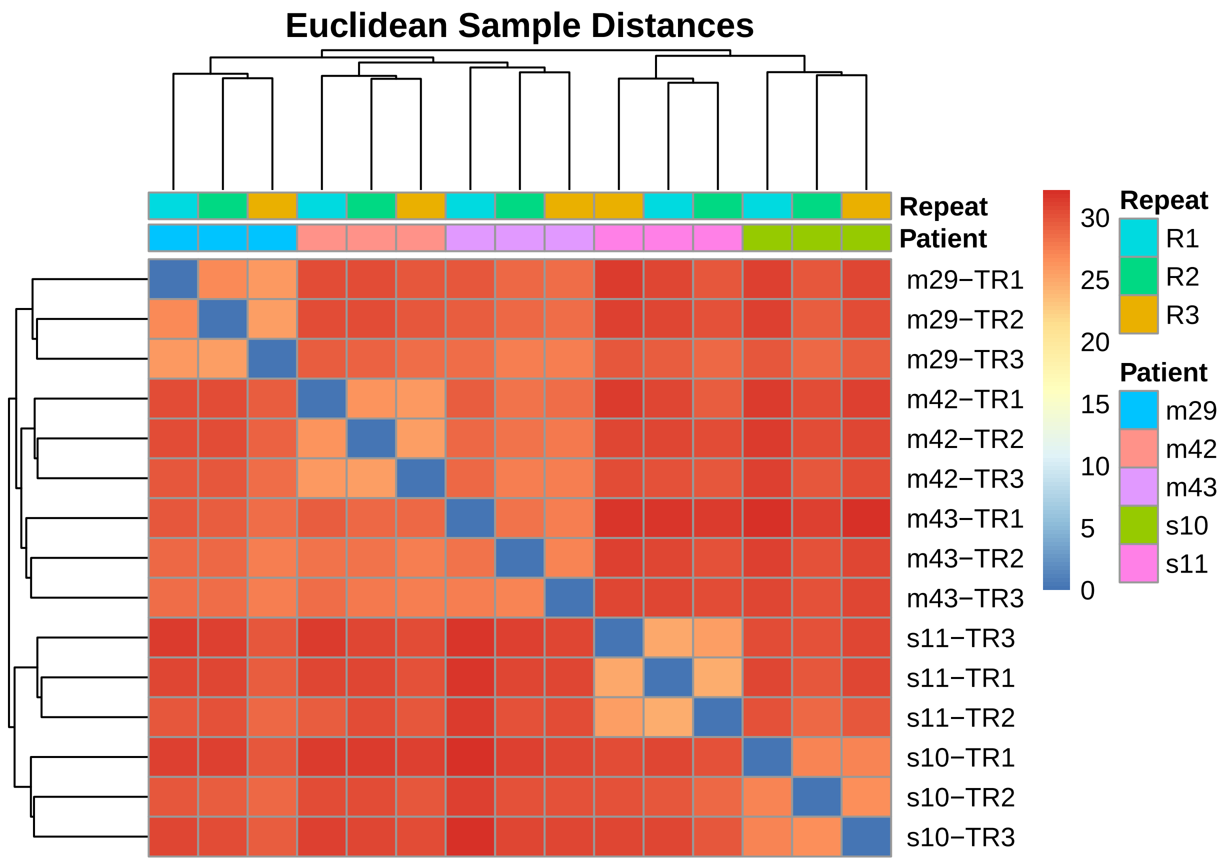
```
# We use the 'pheatmap' library (install with install.packages('pheatmap'))
library(pheatmap)

# Convert the 'dist' object into a matrix for creating a heatmap
sampleDistMatrix <- as.matrix(sampledists)

# The annotation is an extra layer that will be plotted above the heatmap columns
# indicating the cell type
annotation <- data.frame(Patient = factor(rep(1:5, times = 3),
                                          labels = c("m29", "m42", "m43", "s10", "s11")),
                         Repeat = factor(rep(rep(1:3, times = 1), each = 5),
                                         labels = c("R1", "R2", "R3")))

# Set the rownames of the annotation dataframe to the sample names (required)
rownames(annotation) <- names(data)

pheatmap(sampleDistMatrix, show_colnames = FALSE,
         annotation_col = annotation,
         clustering_distance_rows = sampledists,
         clustering_distance_cols = sampledists,
         main = "Euclidean Sample Distances")
```



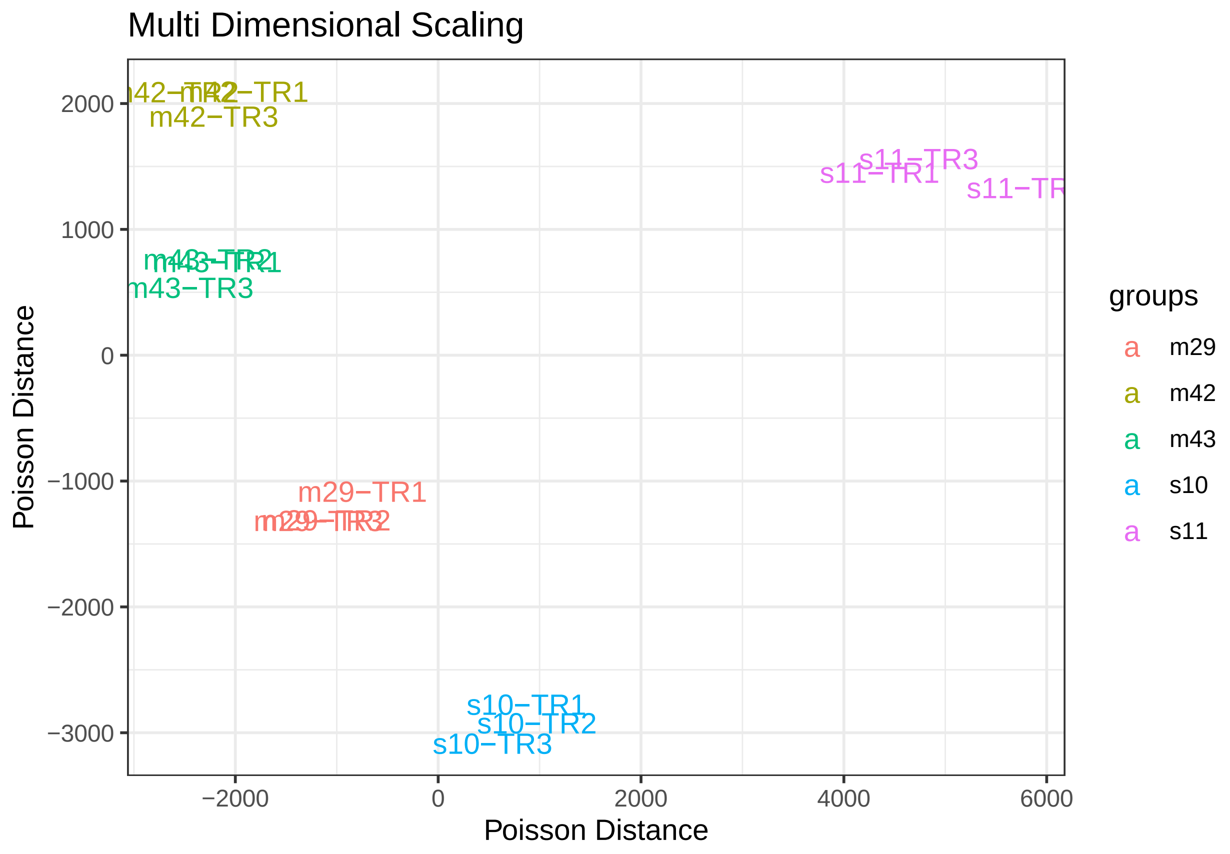
```
library('PoiClaClu')
library('ggplot2')
```

```r
# Use the raw (not r-log transformed!) counts
dds <- assay(ddsMat)
poisd <- PoissonDistance(t(dds))
# Extract the matrix with distances
samplePoisDistMatrix <- as.matrix(poisd$dd)
# Calculate the MDS and get the X- and Y-coordinates
mdsPoisData <- data.frame( cmdscale(samplePoisDistMatrix) )

# And set some better readable names for the columns
names(mdsPoisData) <- c('x_coord', 'y_coord')

# Separate the annotation factor (as the variable name is used as label)
groups <- factor(rep(1:5, times=3),
                 labels = c("m29", "m42", "m43", "s10", "s11"))
coldata <- names(data)

# Create the plot using ggplot
ggplot(mdsPoisData, aes(x_coord, y_coord, color = groups, label = coldata)) +
  geom_text(size = 4) +
  ggtitle('Multi Dimensional Scaling') +
  labs(x = "Poisson Distance", y = "Poisson Distance") +
  theme_bw()
```



All of the technical repeats from each patient are clustered together. Only s11-TR2 is somewhat seperated from the other repeats of the s11 patient, but this distance is still small. For these reasons, the data does noet have to be cleaned since there are no clear outliers visible. Furthermore, all of the mild patients are on the left, while the severe patience are a bit shifted to the right (all positive x_coord values)

Manual pre-processing

Later on we will use libraries in order to normalise the data and calculate the fold-changes of each gene. In order to get some insight into the data, this is done manually. First, the data is normalised to fragments per million mapped fragments

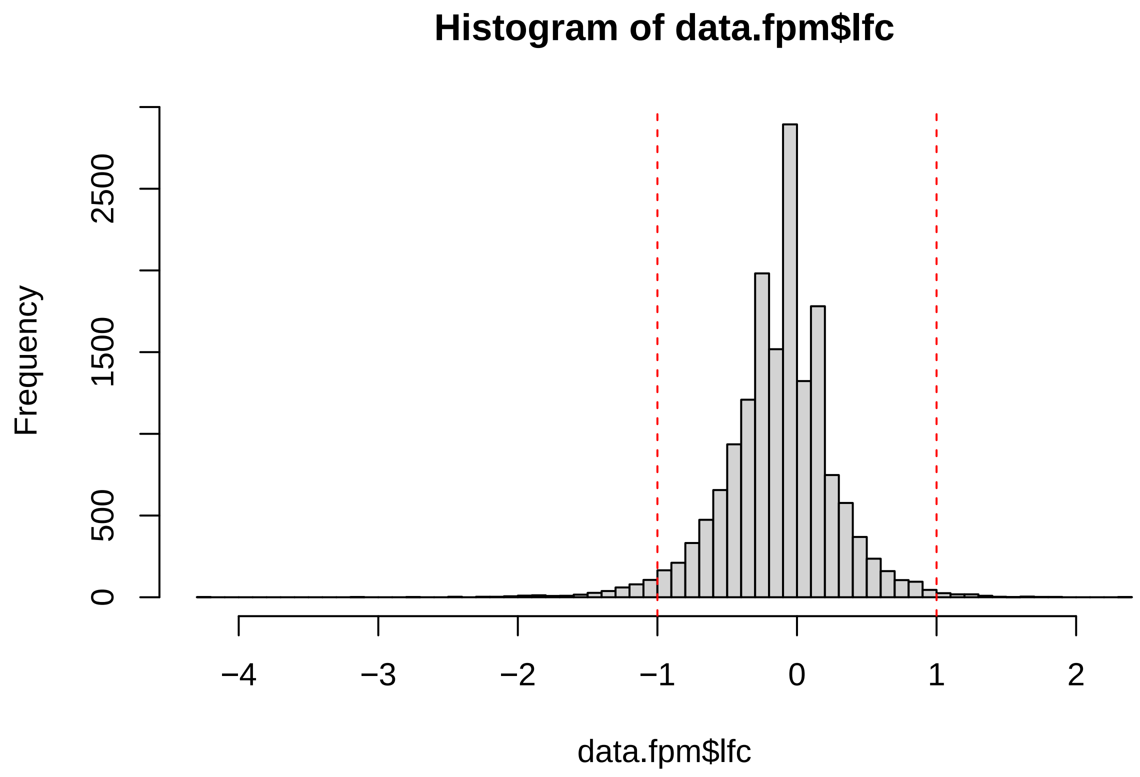
```
# Perform a naive FPM4 normalization
# Note: log transformation includes a pseudocount of 11
data.fpm <- log2( (data / (colSums(data) / 1e6)) + 1 )


data.fpm.filtered <- data.fpm[rowSums(data.fpm) >= 3,]
cat(nrow(data.fpm) - nrow(data.fpm.filtered), "genes have been filtered out")
```

```
## 3326 genes have been filtered out
```

```
data.fpm$m.mean <- rowMeans(data.fpm[m.all])
data.fpm$s.mean <- rowMeans(data.fpm[s.all])
data.fpm$lfc <- data.fpm$m.mean - data.fpm$s.mean

hist(data.fpm$lfc, breaks=80)
abline(v=-1, col = 'red', lty=2)
abline(v=1, col = 'red', lty=2)
```



Discovering Differentialy Expressed Genes

```
#data[nrow(data) + 1,] <- factor(rep(c('mild', 'mild', 'mild', 'severe', 'severe'), times # = 3))

(ddsMat <- DESeqDataSetFromMatrix(countData = data,
                                  colData = annotation,
                                  design = ~ Patient))
```

```
## class: DESeqDataSet
## dim: 16282 15
## metadata(1): version
## assays(1): counts
## rownames(16282): A1BG A2M ... ZYX ZZEF1
## rowData names(0):
## colnames(15): m29-TR1 m42-TR1 ...  s10-TR3 s11-TR3
## colData names(2): Patient Repeat
```

```
(ddsCollapse <- collapseReplicates(ddsMat, groupby = ddsMat$Patient))
```

```
## class: DESeqDataSet
## dim: 16282 5
## metadata(1): version
## assays(1): counts
## rownames(16282): A1BG A2M ... ZYX ZZEF1
## rowData names(0):
## colnames(5): m29 m42 m43 s10 s11
## colData names(2): Patient Repeat
```

```
(ddsCollapse <- DESeq(ddsMat,  betaPrior=FALSE))
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## class: DESeqDataSet
## dim: 16282 15
## metadata(1): version
## assays(4): counts mu H cooks
## rownames(16282): A1BG A2M ... ZYX ZZEF1
## rowData names(34): baseMean baseVar ... deviance maxCooks
## colnames(15): m29-TR1 m42-TR1 ...  s10-TR3 s11-TR3
## colData names(3): Patient Repeat sizeFactor
```