Analysis of gene expression

Job Maathuis

9-3-2022

Contents

1. Exploratory Data Analysis **2**

1.1 Loading in the data . . . 2

1.2 Data structure . . . 2

1.3 Data visualisation . . . 4

1.3.1 Boxplot . . . 4

1.3.2 Density plot . . . 5

1.3.3 Barplot . . . 6

1.3.4 Heatmap . . . 7

1.3.5 Multi-dimensional scaling . . . 9

2. Manual pre-processing **10**

3 Discovering differentialy expressed genes (DEGs) **11**

3.1 DESeq2 . . . 11

3.2 EdgeR . . . 14

3.3 Comparison . . . 15

4 Data Analysis and Visualization **15**

3.1 Venndiagram . . . 16

4.1 Volcano plot . . . 17

3.1.1 DESeq2 . . . 17

3.1.2 EdgeR . . . 18

3.1.3 Comparison . . . 19

3.2 Clustering . . . 20

3.2.1 DESeq2 . . . 20

3.2.2 EdgeR . . . 21

3.2.3 Comparison . . . 21

3.3 PCA . . . 22

3.3.1 DESeq2 22
3.3.1 EdgeR 22

4 Differentialy Expressed Genes **23**
4.1 RGS16 23
4.2 MKI67 23

1. Exploratory Data Analysis

1.1 Loading in the data

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)
# shortening patient names by deleting the middle part
patient_codes[,2] <- gsub('-.*-', '-', patient_codes[,2])
colnames(data) <- patient_codes[,2][patient_codes[,1] == colnames(data)]
```

1.2 Data structure

To get an overview of the structure of the dataset the following code is used

```
library(pander)
pander(head(data, 5), caption = "Raw count data of GSE181032")
```

Table 1: Raw count data of GSE181032 (continued 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

	m42-TR2	m43-TR2	s10-TR2	s11-TR2	m29-TR3	m42-TR3
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

```
pander(dim(data))
```

16282 and *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
```

As can be seen above, the data consists out of 16.282 rows, corresponding to the genes, and 15 columns, which are the different samples. The structure of the samples is as follows:

- 3 patients with a mild disease severity (m29, m42, m43), each having 3 technical repeats
- 2 patients with a severe disease severity (s10, s11), each having 3 technical repeats

To distinguish between these groups, some variables are made below

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

# severe patients
```

```
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 value of each column are obtained, respectively.

```
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. In a biological context this means that most of the genes that are analysed are 'off' and only a few genes are being expressed at time of the RNA-seq analysis.

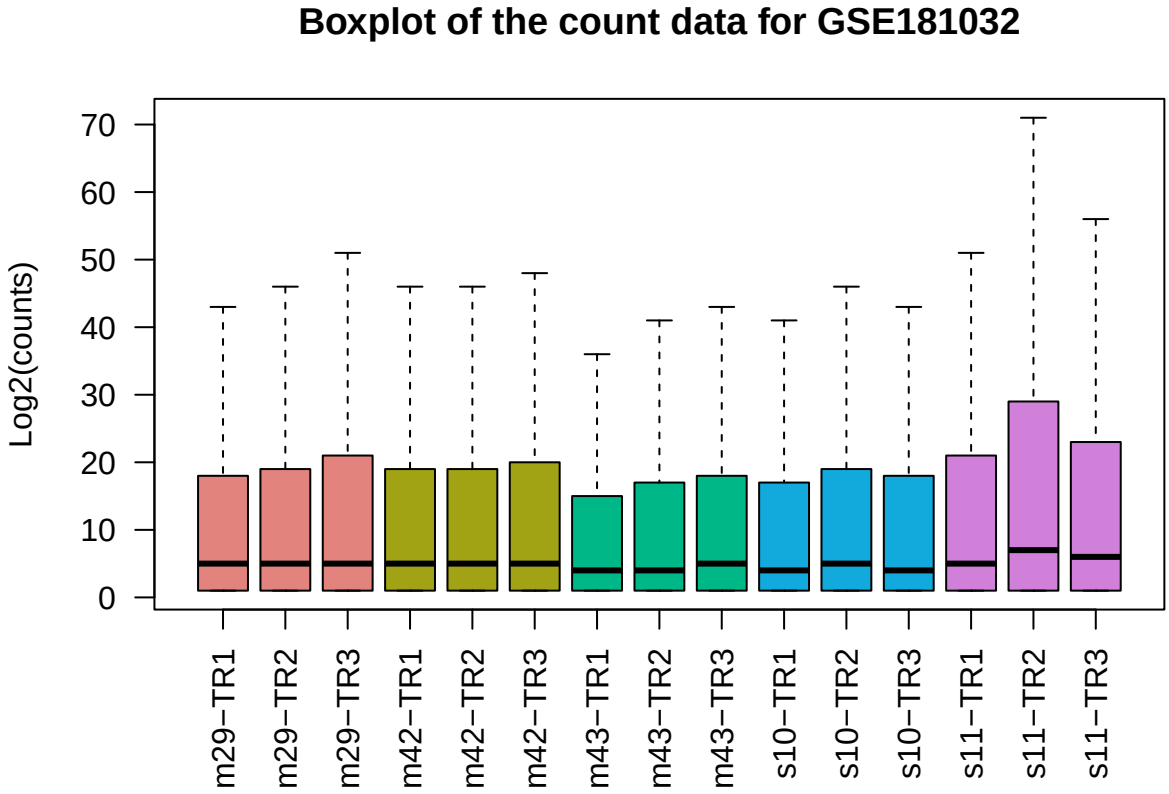
1.3 Data visualisation

1.3.1 Boxplot

Next a boxplot can be made to visualise the structure of the data.

```
library(scales)
colors <- hue_pal(c=70)(5)
```

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

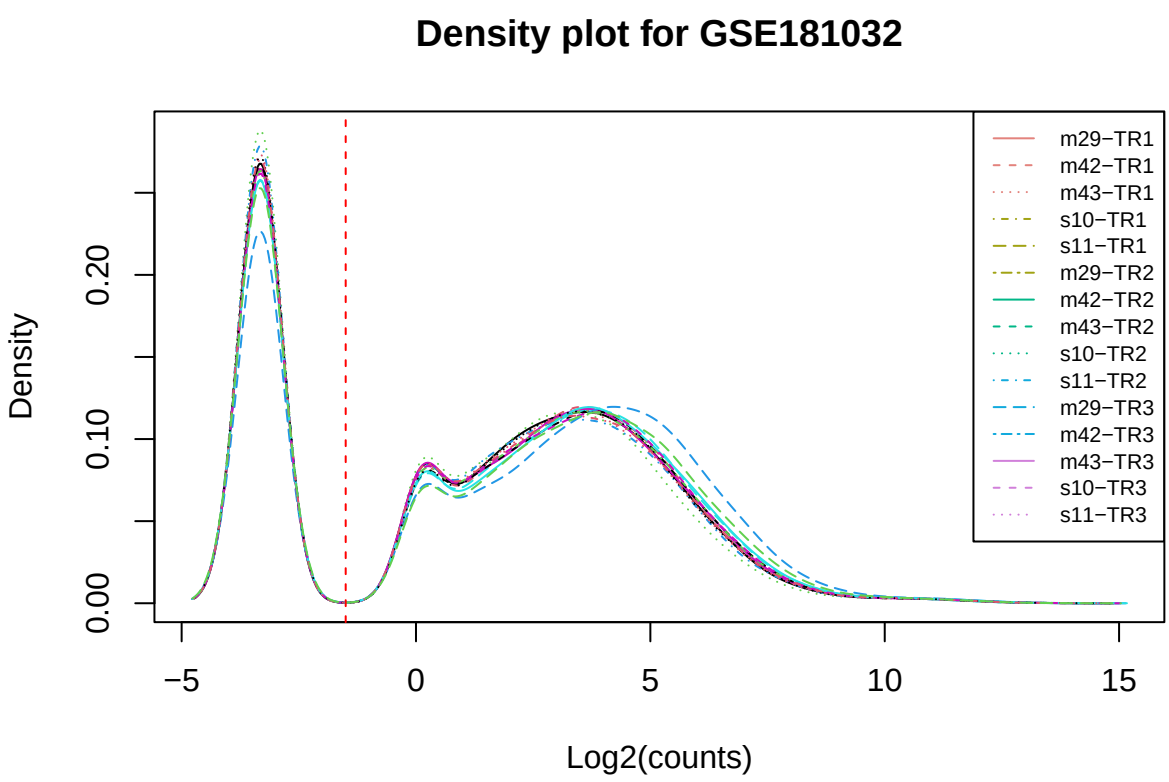


For every repeat of each patient the same same distrubution can be observed. This is also the case if the patients are compared. A somewhat larger deviation can be seen in the s11-TR2 data. The large differences between the maximum values and the other values reflect the summary data in the previous chapter.

1.3.2 Density plot

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

```
library(affy)
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 = 0.7, col=rep(colors, each=3))
abline(v=-1.5, lwd=1, col='red', lty=2)
```

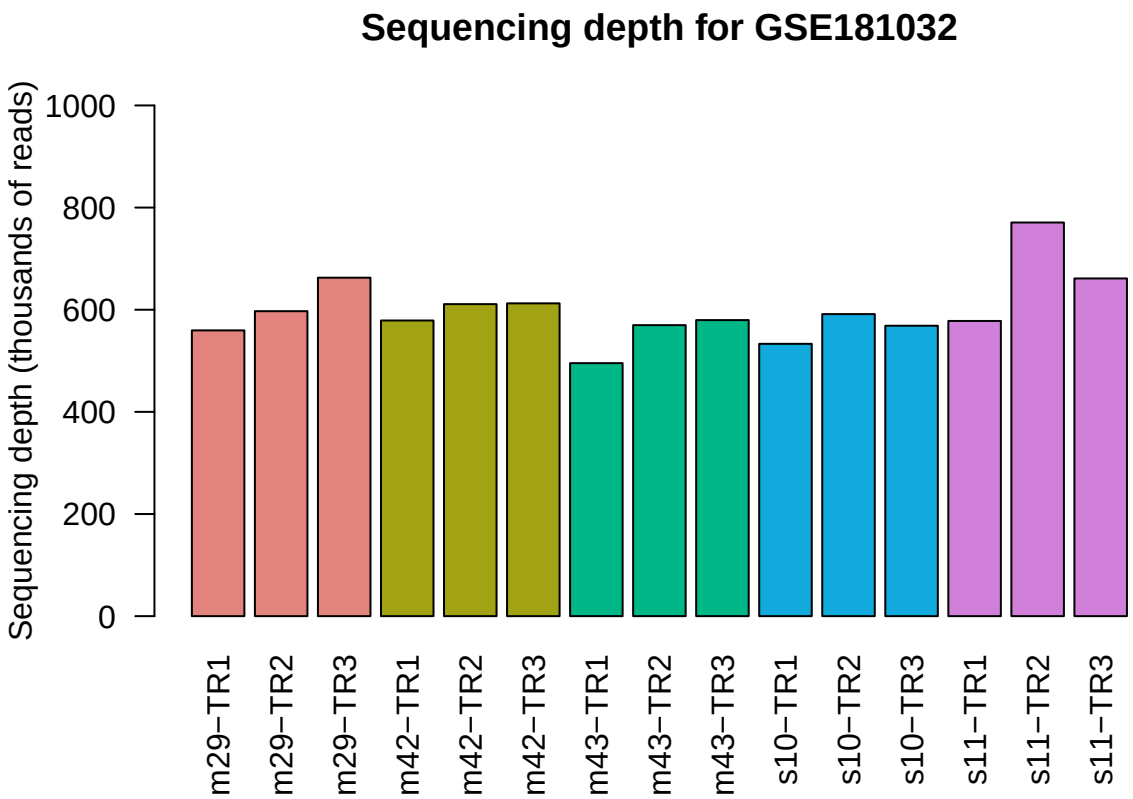


All of the lines follow the same distrubution, where only m29-TR3 is a bit variated from the other data. This variation is not much, so this will probably not cause any big problems. The large peak at the left of the red stippled line is caused by all of the zero's in the dataset.

1.3.3 Barplot

The amount of reads, also known as the read depth, can differ widely between all of the measured samples. To get an overview of the differences in read depth a barplot is made.

```
barplot(colSums(data[,c(m29, m42, m43, s10, s11)]) / 1e3, las = 2,
        ylab = 'Sequencing depth (thousands of reads)',
        main = 'Sequencing depth for GSE181032',
        col=rep(colors, each=3), ylim = c(0, 1000))
```



In the barplot not much deviation can be seen between samples or within the samples. Only patient s11 has some larger deviation between the minimum and maximum sequencing depth, which may cause some minor problems. For this reason the dataset is being normalized. Normalization is achieved by using a variance stabilizing transformation (vst) form the DESeq2 library. When the data is normalized sample distances can be calculated.

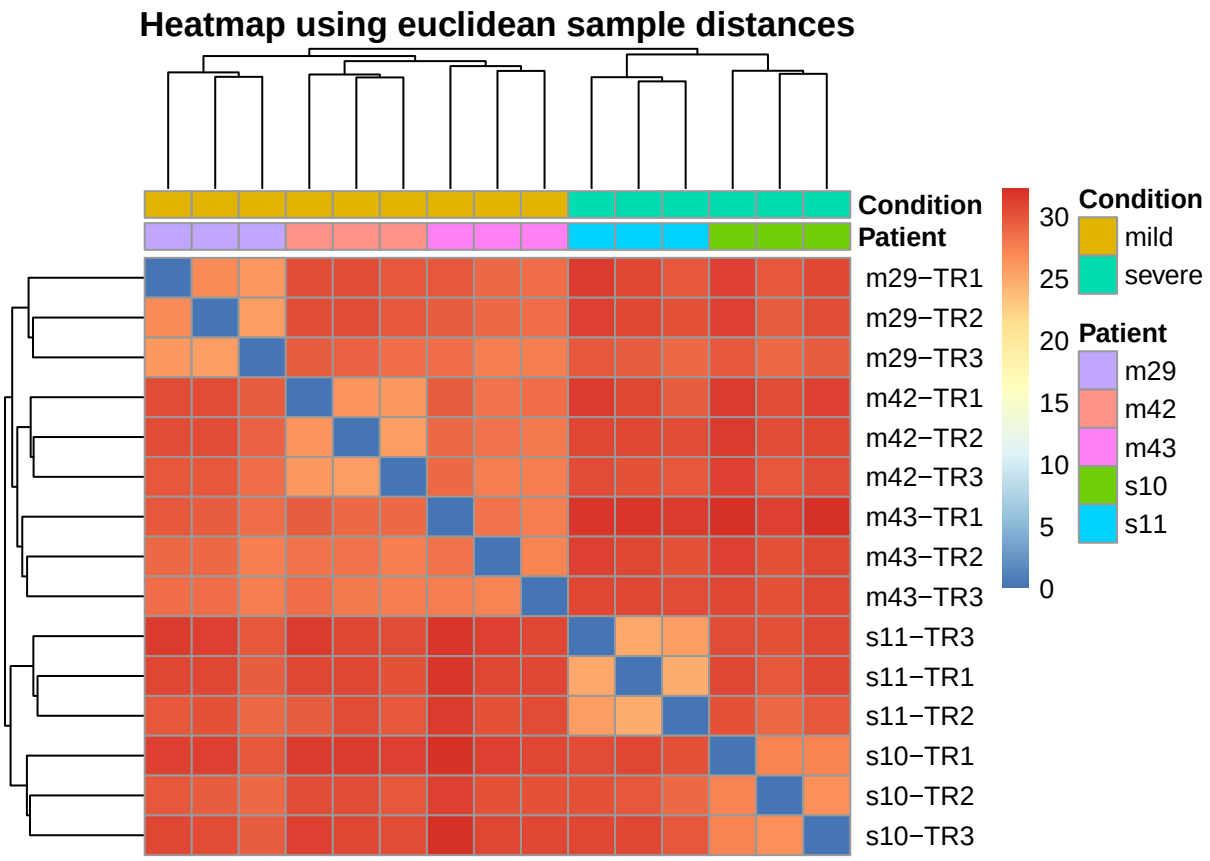
```
library("DESeq2")
# create a DESeq dataset
ddsMat <- DESeqDataSetFromMatrix(countData = data,
                                 colData = data.frame(samples = names(data)),
                                 design = ~ 1)
# normalization
rld.dds <- vst(ddsMat)
# obtain normalized data
rld <- assay(rld.dds)
# transposes and calculate distances
sampledists <- dist(t(rld))
```

1.3.4 Heatmap

Using the calculated euclidean sample distances a heatmap can be created. The clustering is based on condition and patient.

```
library(pheatmap)

# Convert the sample distances into a matrix for creating a heatmap
```

```
sampleDistMatrix <- as.matrix(sampledists)

# create annotation, which represents the design of the data
annotation <- data.frame(Patient = factor(rep(1:5, times = 3),
                                          labels = c("m29", "m42", "m43", "s10", "s11")),
                         Condition = factor(rep(c(rep("mild", times = 3),
                                                  rep("severe", times = 2)), times = 3),
                                            levels = c('mild', 'severe')),
                         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 (
rownames(annotation) <- names(data)

# Create heatmap
pheatmap(sampleDistMatrix, show_colnames = FALSE,
         annotation_col = annotation[,c(1,2)],
         clustering_distance_rows = sampledists,
         clustering_distance_cols = sampledists,
         main = "Heatmap using euclidean sample distances")
```

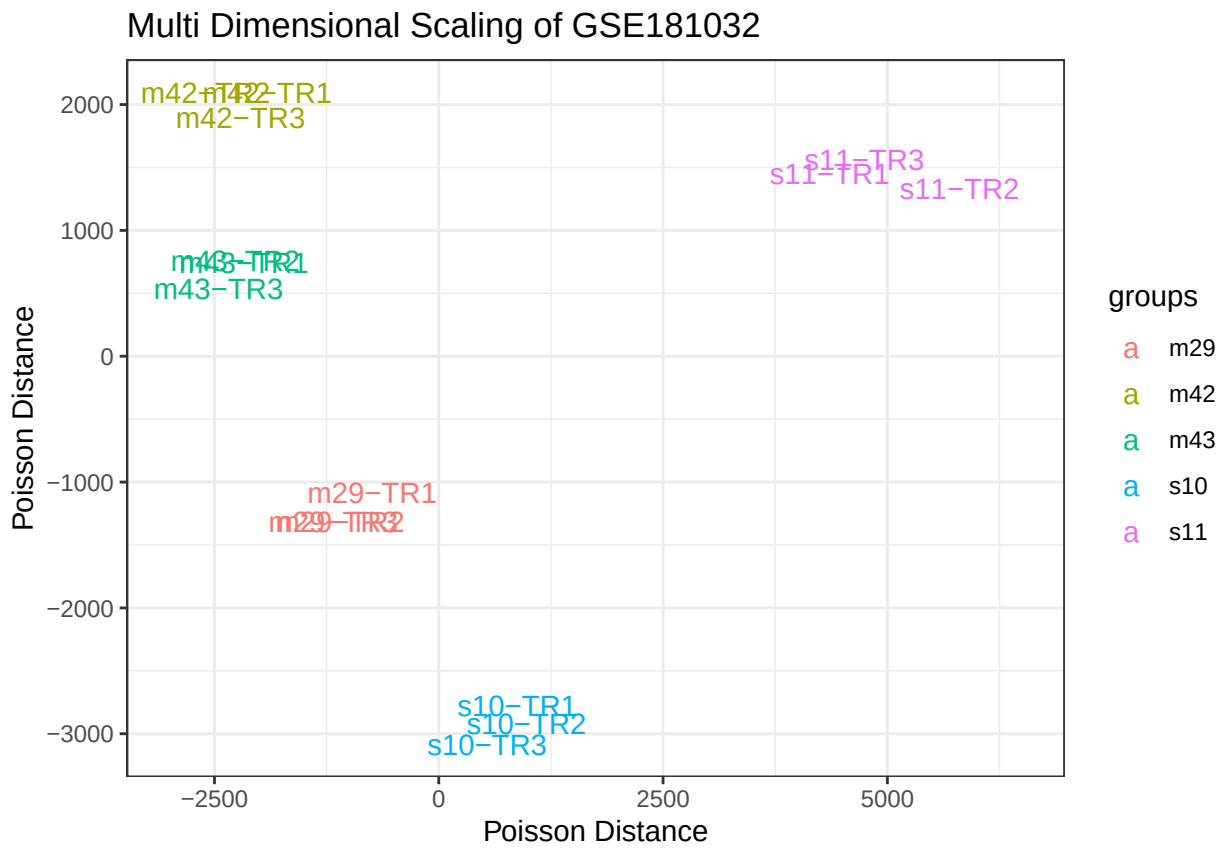


In the heatmap a clustering can be seen. The mild and severe patients are divided. The mild patients are orange colored and the severe patients show a dark red color. This color difference, and therefore the overal euclidean distance, is not very large. Furthermore, each patient except for m43 can be distinguished by a light shaded box clustering all of the repeats together.

1.3.5 Multi-dimensional scaling

With the use of multidimensional scaling a visual representation can be made of the dissimilarities of the samples. These dissimilarities are based upon Poission distances. It is expected that the repeats of each patient are close to each other because they show less dissimilarities when compared to other samples. However, if this is not the case the data may need to be cleaned or outliers needs to be removed.

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

dds <- assay(ddsMat)
# Calculate distances using Poisson distances
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))

# Rename col names
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 of GSE181032') +
  labs(x = "Poisson Distance", y = "Poisson Distance") +
  xlim(-3000, 6500) +
  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 not 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-axis values)

2. Manual pre-processing

Later on we libraries are used in order to normalize the data and filter out low count genes. This is essential because these low count genes may disturb the statistical test that these libraries apply. However, in this chapter the normalization and filtering will be done manually in order to get some insight into what is going during these steps.

First, the data is normalized using log2 transformation on the fragments per million mapped fragments (FPM) data. Low count genes are then filtered out based on the criterium that the total FPM of a gene is higher or equal to 3. If this is not the case, the whole gene will be removed from the dataset.

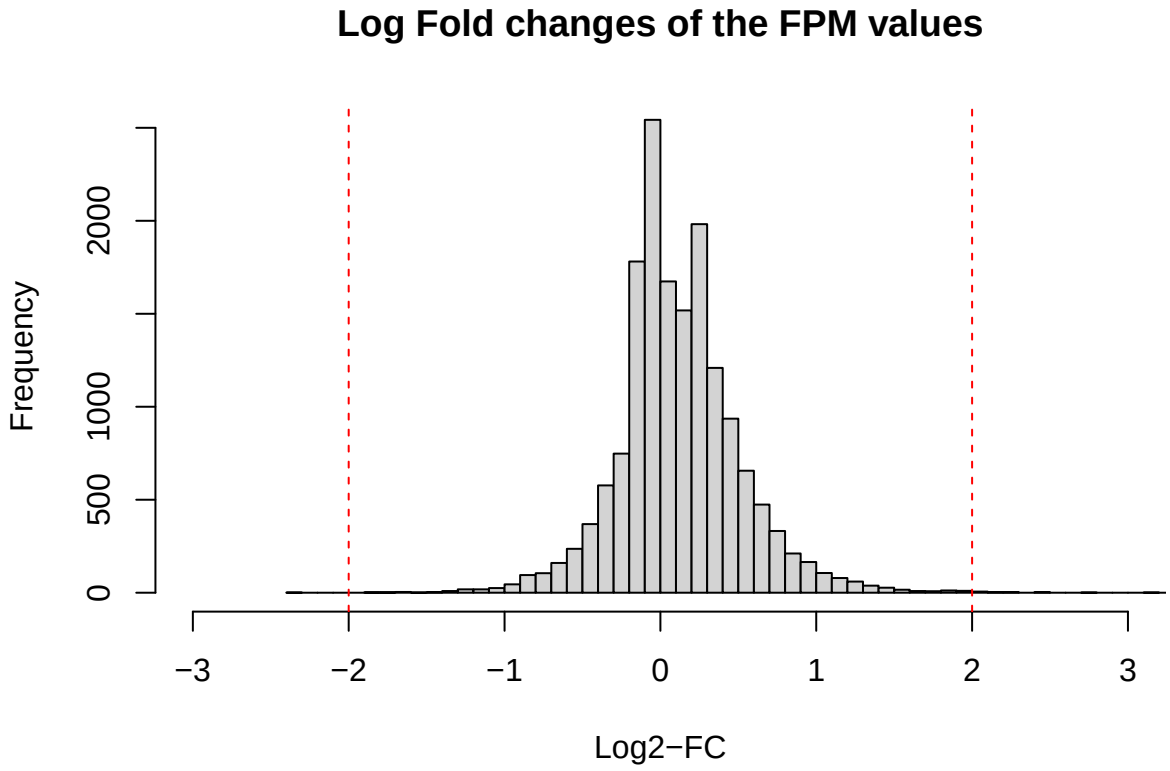
```
# Perform a naive FPM4 normalization
data.fpm <- log2( (data / (colSums(data) / 1e6)) + 1 )


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
```

This process results in the filtering out of 3326 genes which is 20.43 % of the total genes

Using these FPM values the log fold changes can be calculated between the mild and severe groups. Since log values are used the fod change can be obtained by by subtracting the mean of each group for each gene. With the log fold changes a histogram can be made to see the distribution.

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

hist(data.fpm$lfc, breaks=80, xlab = "Log2-FC",
     main = "Log Fold changes of the FPM values",
     xlim = c(-3, 3))
abline(v=-2, col = 'red', lty=2)
abline(v=2, col = 'red', lty=2)
```



As can be seen in the histogram, most of the Log Fold changes are within the -2 and 2 boundaries. These are the boundaries used later on to determine the differential expressed genes (DEG's). So seeing this histogram it is not expected that there are a lot of DEG's when comparing the mild and severe patients.

3 Discovering differentialy expressed genes (DEGs)

3.1 DESeq2

The DESeq2 package can be used for differential analysis of high-dimensional count data. It will apply a normalization (which was done manually in the previous chapter) on the data and pefrom a differential

expression analysis. The technical repeats are also combined using the collapseReplicates function.

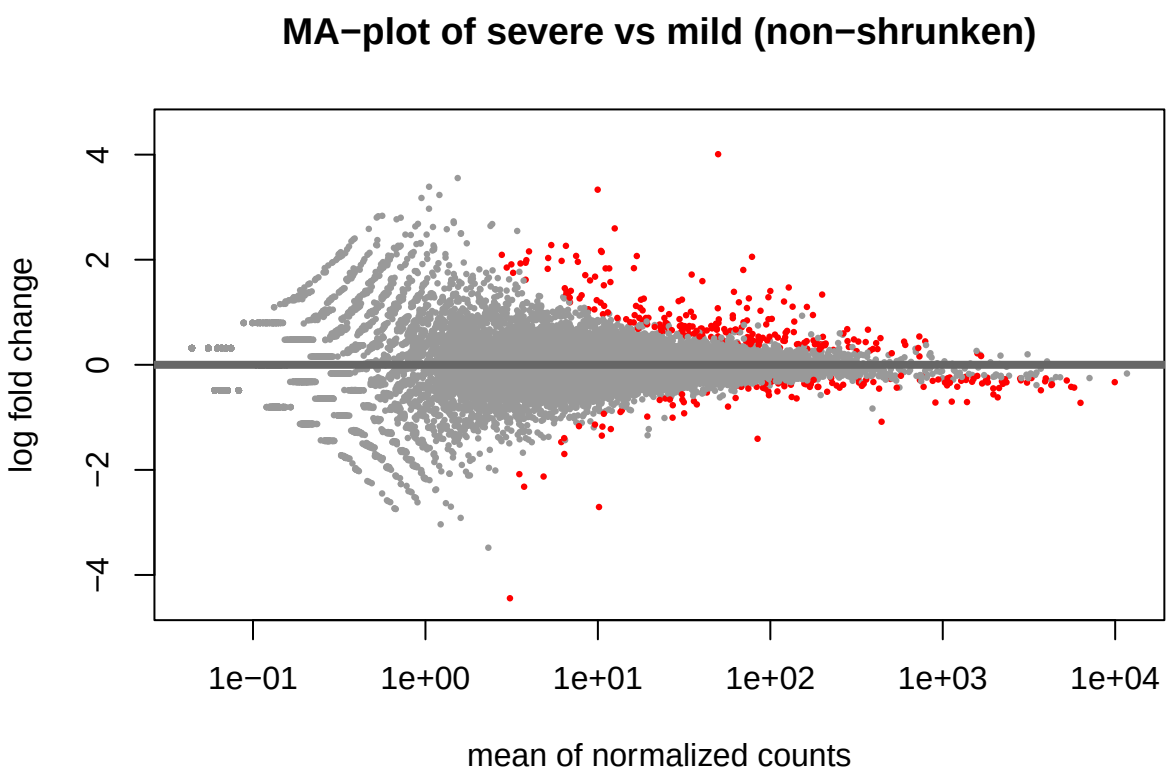
```
dds <- DESeqDataSetFromMatrix(countData = data,
                                colData = annotation,
                                design = ~ Condition)
dds <- DESeq(dds, betaPrior = FALSE)
dds.collapse <- collapseReplicates(dds, groupby = annotation$Patient)
res <- results(dds.collapse)
pander(as.data.frame(head(res)), caption = "DESeq2 results")
```

Table 4: DESeq2 results

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
A1BG	0.3677	-0.1646	1.521	-0.1082	0.9138	NA
A2M	0.8517	-1.319	1.077	-1.225	0.2207	NA
A2ML1	0.06701	-0.4852	3.135	-0.1548	0.877	NA
A4GALT	0	NA	NA	NA	NA	NA
AAAS	9.928	-0.0104	0.3084	-0.03373	0.9731	0.9948
AACS	1.311	-0.1606	0.8086	-0.1986	0.8426	NA

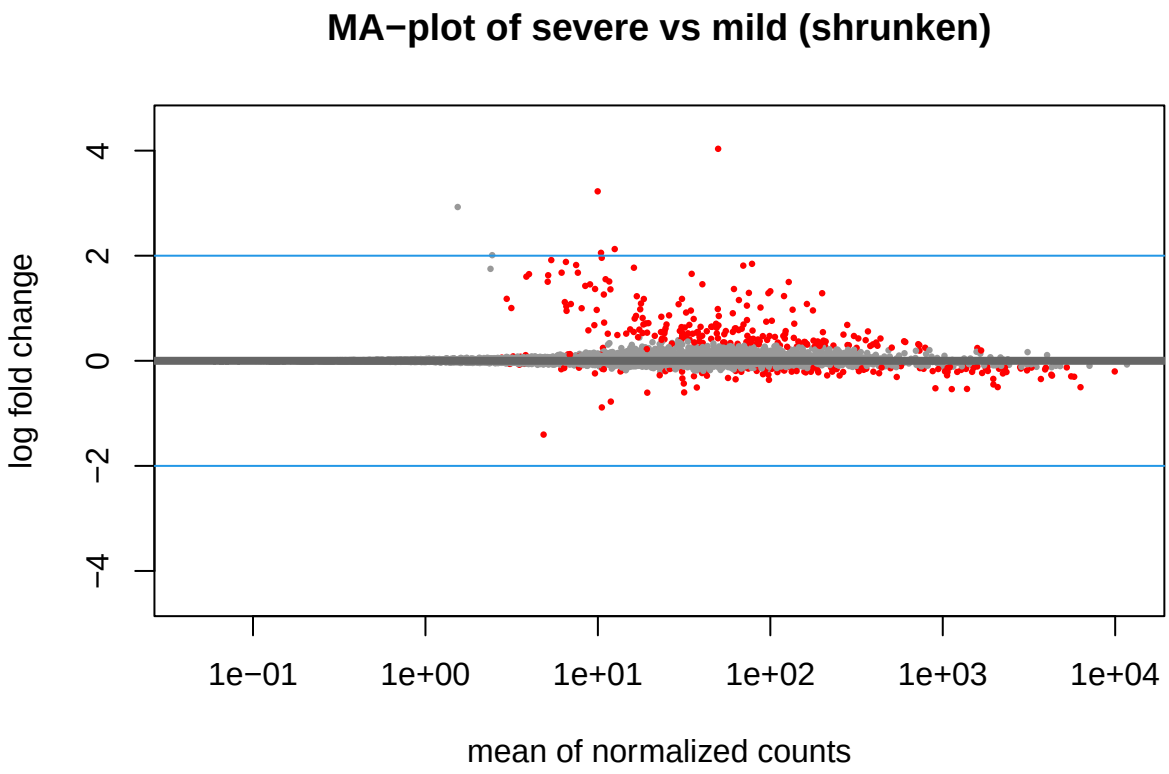
The results that are obtained from this DESeqDataSet object (see table above) are plotted in an MA-plot. This plot visualizes the differences between the log transformed data of the mild and severe groups, better known as the log fold change.

```
DESeq2::plotMA(res, ylim = c(-4.5, 4.5),
       main = "MA-plot of severe vs mild (non-shrunken)", alpha = 0.05, colSig = "red")
```



The red-coloured dots are significant genes (p.value < 0.05) and the grey dots are insignificant genes. On the left side of the MA-plot a lot of noise can be seen. This is due to the low count genes. In order to remove this noise the log fold changes can be shrunken. This is done by the following code-block.

```
res <- lfcShrink(dds.collapse, coef = 2, type = "apeglm")
DESeq2::plotMA(res,  main = "MA-plot of severe vs mild (shrunken)",
               ylim = c(-4.5, 4.5), alpha = 0.05, colSig = "red")
abline(h=c(-2, 2), col=4)
```



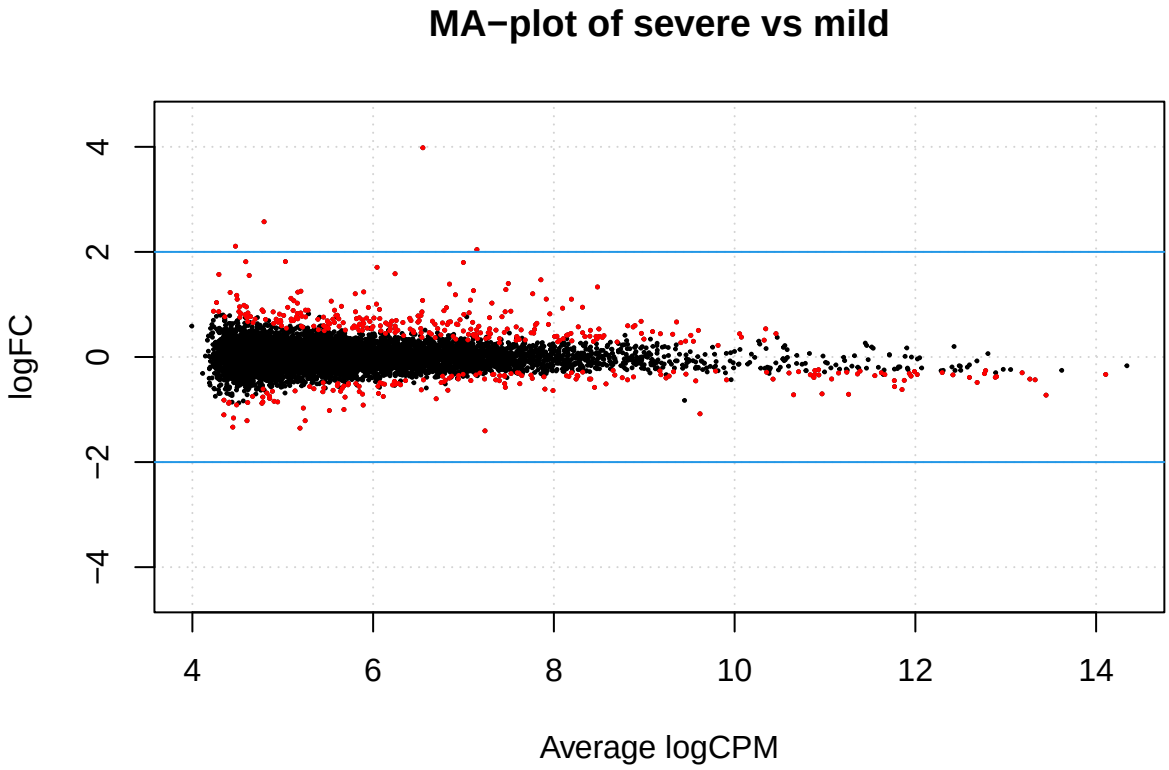
The noise has now successfully been removed, which results in a cleaner dataset. The structure and the amount of the significant points (red- coloured) is the same, only closer to 0. The amount of significant genes is fairly large, even at low log fold changes. In contrast, there only a few of significant genes having a log fold change above 2 or below -2. The log fold changes will be the biggest discriminating factor in selecting DEG's.

3.2 EdgeR

EdgeR is a similar technique as DESeq2 and will also be used to discover DEG's. The results of each technique will be compared.

```
library(edgeR)
dge <- DGEList(counts = data, group = annotation$Condition)


design <- model.matrix(~ annotation$Condition)
keep <- filterByExpr(dge, design)
dge <- dge[keep, , keep.lib.sizes=FALSE]

dge <- calcNormFactors(dge)
dge <- estimateDisp(dge, design)
et <- exactTest(dge)

# variable which is later used to create the volcano plot
res.edger <- topTags(et, n = Inf)
```

```
deGenes <- decideTestsDGE(et, p=0.05, lfc=0)
deGenes <- rownames(et)[as.logical(deGenes)]
plotSmear(et, de.tags=deGenes, ylim = c(-4.5, 4.5), main = "MA-plot of severe vs mild")
abline(h=c(-2, 2), col=4)
```



In het MA-plot obtained by edgeR the significant genes are coloured red. A lot of genes are significant, even with low log fold changes. However, only a few are above the log fold change of 2.

3.3 Comparison

When comparing the MA-plot obtained from DESeq2 and edgeR a lot of similarities can be observed. In both plots no downregulated genes (log fold change below -2) can be seen. In addition, in each plot four upregulated genes are observed (log fold change above 2). With both methods there are also a lot significant genes which are below the log fold change tershold. In the following chapter volcano plots are made and the upregulated genes will be annotated and compared (DESeq2 vs edgeR).

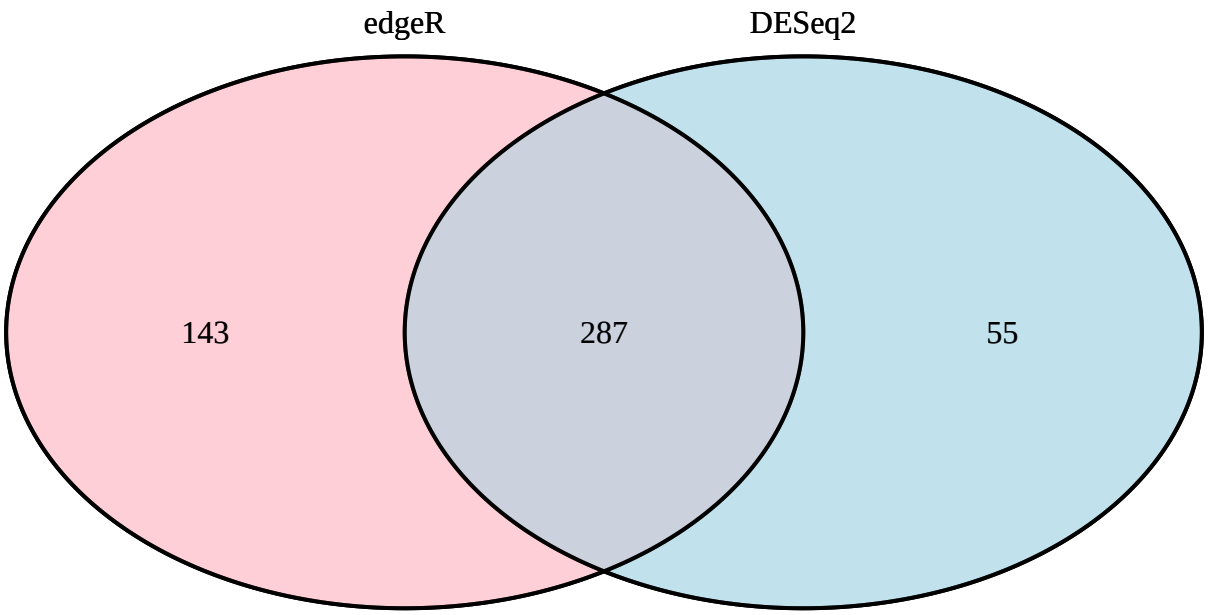
4 Data Analysis and Visualization

In this chapter the data obtained from DESeq2 and edgeR will be analysed and visualised. Data visualisation is achieved by making a venndigram, a volcano plot, a heatmap and a principal components analysis (PCA) plot for each technique (DESeq2 and edgeR). Moreover, the plots from each techniques will be compared.

3.1 Venndiagram

To see the similarities in significant genes a venndiagram will be made. THis illustrates the amount of genes that are found to be significint in both techniques and the amount of genes that are only significint in one of the two techniques.

```
library(VennDiagram)
deseq.degs <- row.names(res[which(res$padj < 0.05),])
edger.degs <- row.names(res.edger$table[which(res.edger$table$FDR < 0.05),])

venn.plot <- draw.pairwise.venn(length(deseq.degs),
                                length(edger.degs),
                                # Calculate the intersection of the two sets
                                length(intersect(deseq.degs, edger.degs)),
                                category = c("DESeq2", "edgeR"), scaled = F,
                                fill = c("light blue", "pink"), alpha = rep(0.5, 2),
                                cat.pos = c(0, 0))

# Actually plot the plot
grid.draw(venn.plot)
```



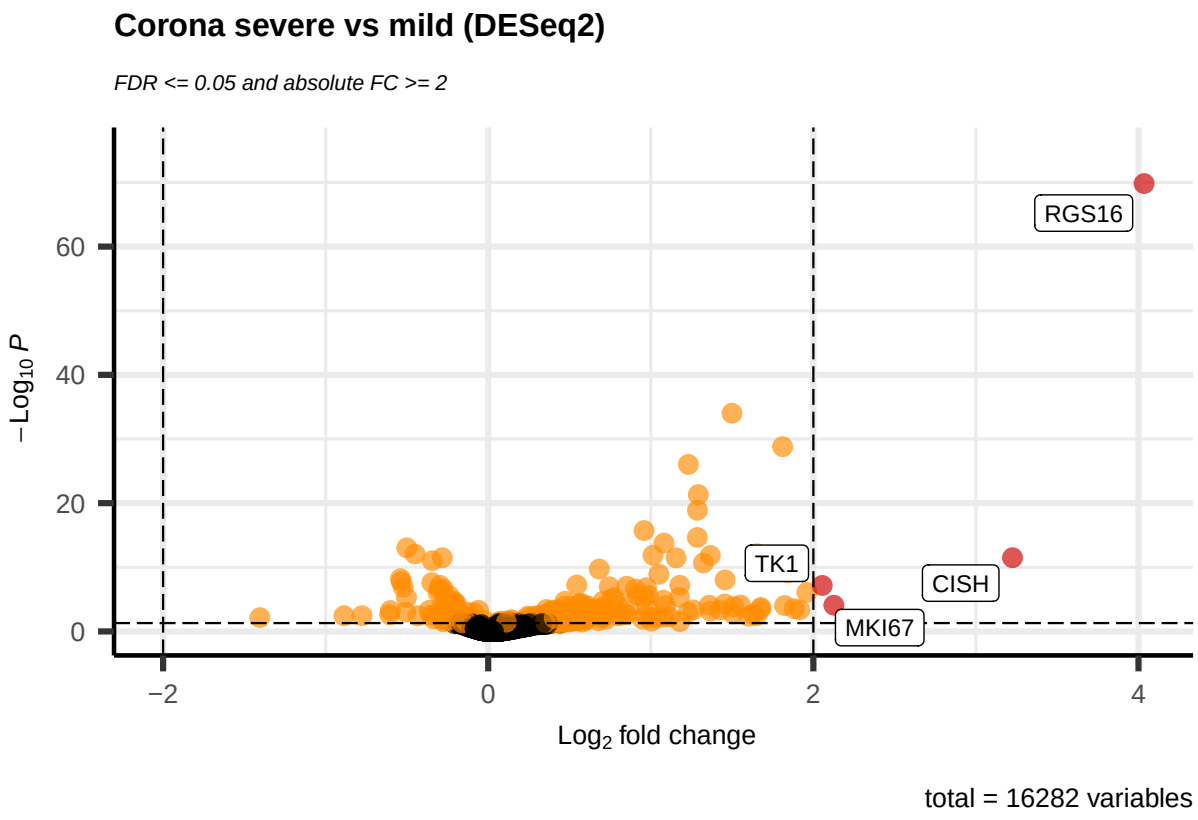
According to the venndiagram 287 significant genes are found by both DESeq2 and edgeR, which is a relatively large amount. DESeq2 found 55 significant genes that were not found by edgeR and edgeR found 143 significant gens not found by DESeq2. In short, there will probably be some similarities in DEG's when the results of DESeq2 and edgeR are compared.

4.1 Volcano plot

First, the data will be visualised using a volcano plot. A volcano plot is a type of a scatterplot in which the log fold change is plotted against he adjusted p-value. The EnhancedVolcano library is used for this.

3.1.1 DESeq2

```
library(EnhancedVolcano)

# vulcano plot
EnhancedVolcano(res, x = 'log2FoldChange', y = 'padj',
                lab=rownames(res),
                title = "Corona severe vs mild (DESeq2)",
                subtitle = bquote(italic('FDR <= 0.05 and absolute FC >= 2')),
                # Change text and icon sizes
                labSize = 3, pointSize = 3, axisLabSize=10, titleLabSize=12,
                subtitleLabSize=8, captionLabSize=10,
                xlim = c(min(res$log2FoldChange) + 0.5, max(res$log2FoldChange) + 0.5, na.rm = T),
                # Disable legend
                legendPosition = "none",
                # Set cutoffs
                pCutoff = 0.05, FCcutoff = 2,
                # make fancy boxes and better colours
                boxedLabels = T,
                drawConnectors = T,
                widthConnectors = 1.0,
                min.segment.length =  1,
                col = c("black", "green", "darkorange", "red3"),
                colAlpha = 2/3)
```



As expected no downregulated genes are observed and only four upregulated genes. The DEG's that are oberved are:

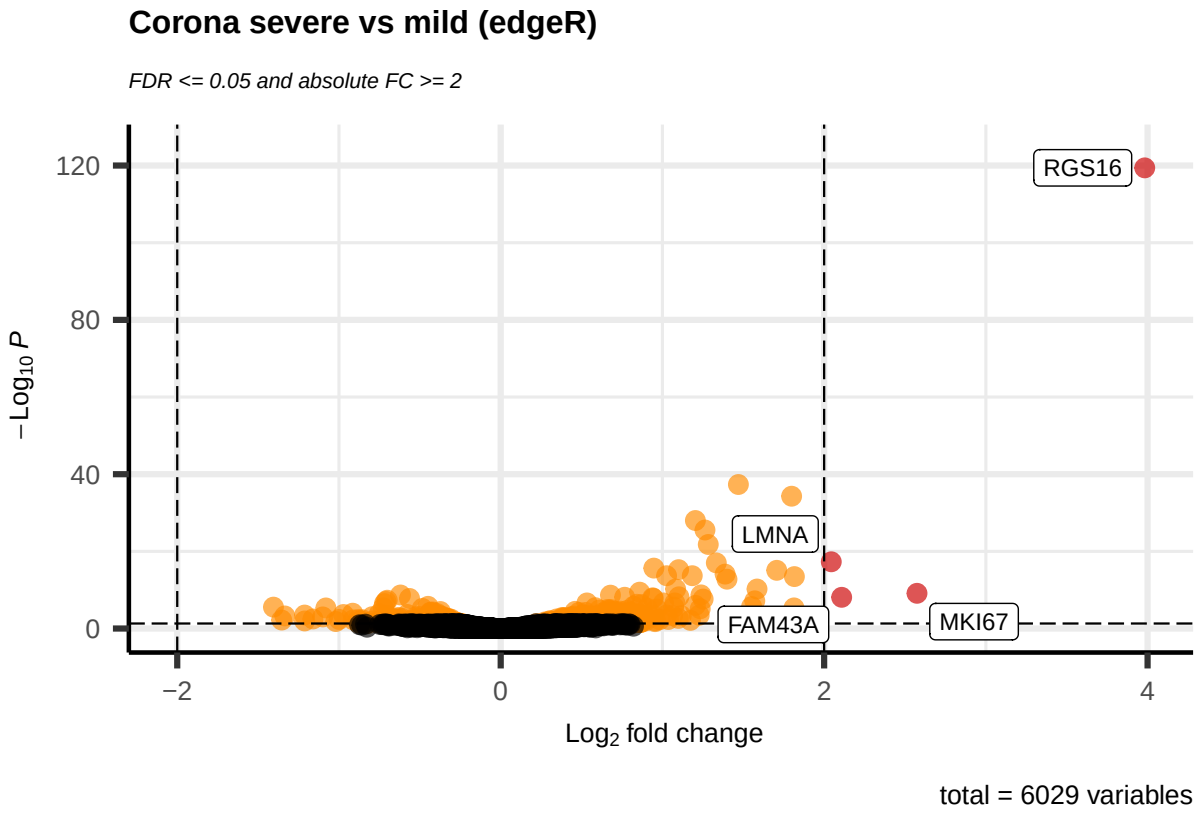
- TK1
- MKI67
- CISH
- RGS16

RGS16 is the clearly the most significant and upregulated gene when compared to the other DEG's. To see if the same results are obtained by edgeR the same volcano plot is used on the edgeR results.

3.1.2 EdgeR

```
EnhancedVolcano(res.edger$table, x = 'logFC', y = 'FDR',
                lab=rownames(res.edger$table),
                title = "Corona severe vs mild (edgeR)",
                subtitle = bquote(italic('FDR <= 0.05 and absolute FC >= 2')),
                # Change text and icon sizes
                labSize = 3, pointSize = 3, axisLabSize=10, titleLabSize=12,
                subtitleLabSize=8, captionLabSize=10,
                xlim = c(min(res$log2FoldChange) + 0.5, max(res$log2FoldChange) + 0.5, na.rm = T),
                # Disable legend
                legendPosition = "none",
                # Set cutoffs
```

```
                pCutoff = 0.05, FCcutoff = 2,
                # make fancy boxes and better colours
                boxedLabels = T,
                drawConnectors = T,
                widthConnectors = 1.0,
                min.segment.length =  2,
                col = c("black", "green", "darkorange", "red3"),
                colAlpha = 2/3)
```



In the volcano plot above, again, no downregulated genes are observed when the severe patients are compared against the severe patients. Furthermore, four upregulated genes can be noticed:

- LMNA
- FAM43A
- MKI67
- RGS16

RGS16 is the most significant and upregulated DEG.

3.1.3 Comparison

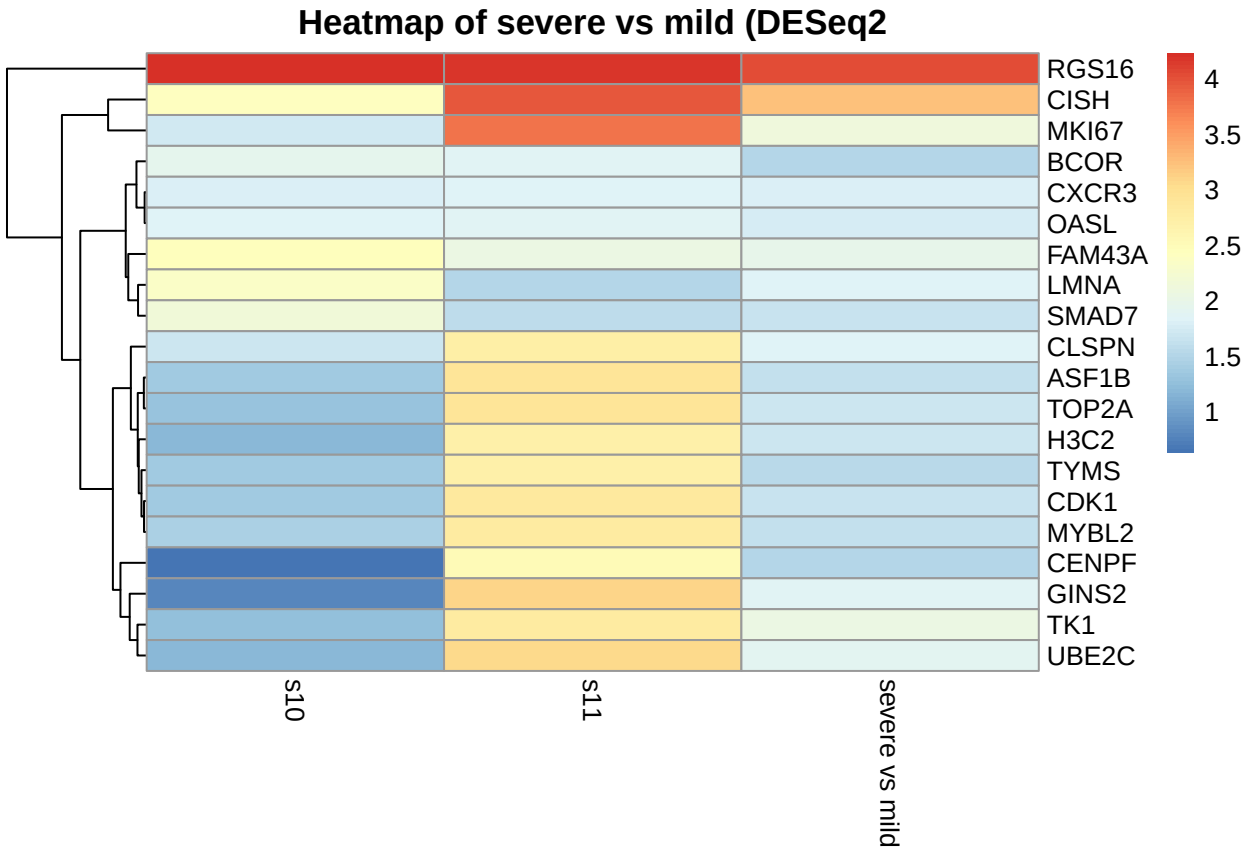
When comparing the volcano plots of DESeq2 and edgeR a similar structure can be seen. Both plots do not show any downregulated genes. Moreover, the top DEG in both cases is RGS16. Additionally, both methods annotated MKI67 as a DEG. The other two genes of DESeq2 (TK1 and CISH) are not observed by edgeR and the two genes LMNA and FAM43A are not observed by DESeq2.

3.2 Clustering

Clustering and a heatmap visualisation will be obtained using the `pheatmap` library. For better visualisation the log fold change criterium is set to 1.5. This ensures enough genes are shown in the heatmap, instead of the four DEG's found by DESeq2 and edgeR.

3.2.1 DESeq2

```
res.omit <- na.omit(res)
sig.genes <- res.omit[res.omit$padj <= 0.05 & abs(res.omit$log2FoldChange) > 1.5,]

expr <- data.frame(rowMeans(data.fpm[s10]) - data.fpm$m.mean,
                   rowMeans(data.fpm[s11]) - data.fpm$m.mean,
                   res$log2FoldChange)
colnames(expr) <- c("s10", "s11", "severe vs mild")

pheatmap(expr[rownames(expr) %in% rownames(sig.genes),],
         cluster_cols = F,
         main = "Heatmap of severe vs mild (DESeq2")
```

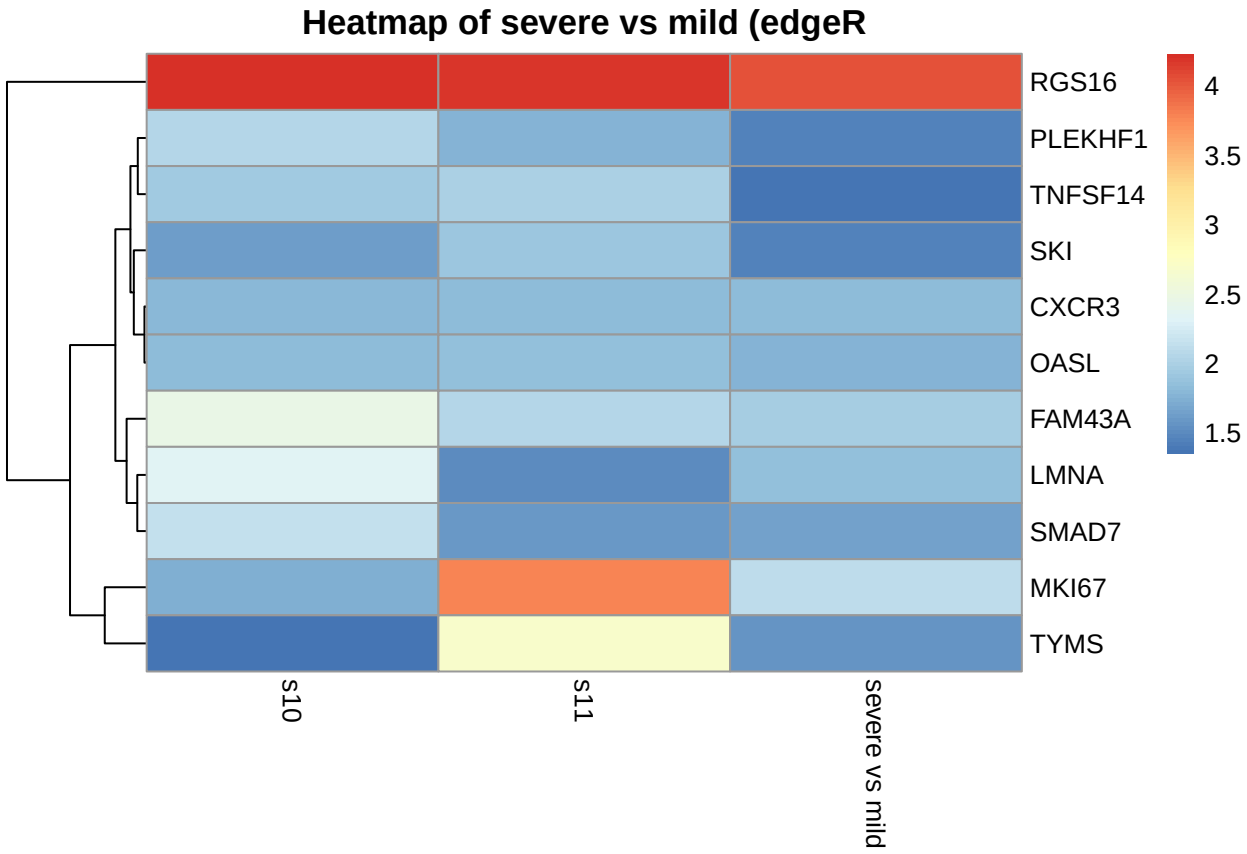


As expected the RGS16 gene stands out because of its high fold change value. Furthermore, CISH and MKI67 are also clearly clustered. The TK1 DEG is less visible by the heatmap and clustering. This is probably due to the lower log fold change and lower significance (higher p-value). This was visualised by the volcano plot in which TK1 was on the edge of the 2 log fold change. When comparing the s11 patient

with the s10 patient a lot of dissimilarities can be seen, especially at the lower end of the heatmap. the s11 patient shows a higher log fold change at most of the genes.

3.2.2 EdgeR

```
sig.genes <- res.edger[res.edger$table$FDR <= 0.05 & abs(res.edger$table$logFC) > 1.5,]

expr <- data.frame(rowMeans(data.fpm[s10]) - data.fpm$m.mean,
                   rowMeans(data.fpm[s11]) - data.fpm$m.mean,
                   res$log2FoldChange)
colnames(expr) <- c("s10", "s11", "severe vs mild")

pheatmap(expr[rownames(expr) %in% rownames(sig.genes),],
         cluster_cols = F,
         main = "Heatmap of severe vs mild (edgeR")
```



Again RGS16 can be noticed right away. However, the other DEG's are not as noticable as RGS16. MKI67 shows a light orange color at the s11 patient and a light blue color when severe is compared against mild (higher upregulated than dark blue). More upregulation is observed in the s11 patient when compared to the s10 patient.

3.2.3 Comparison

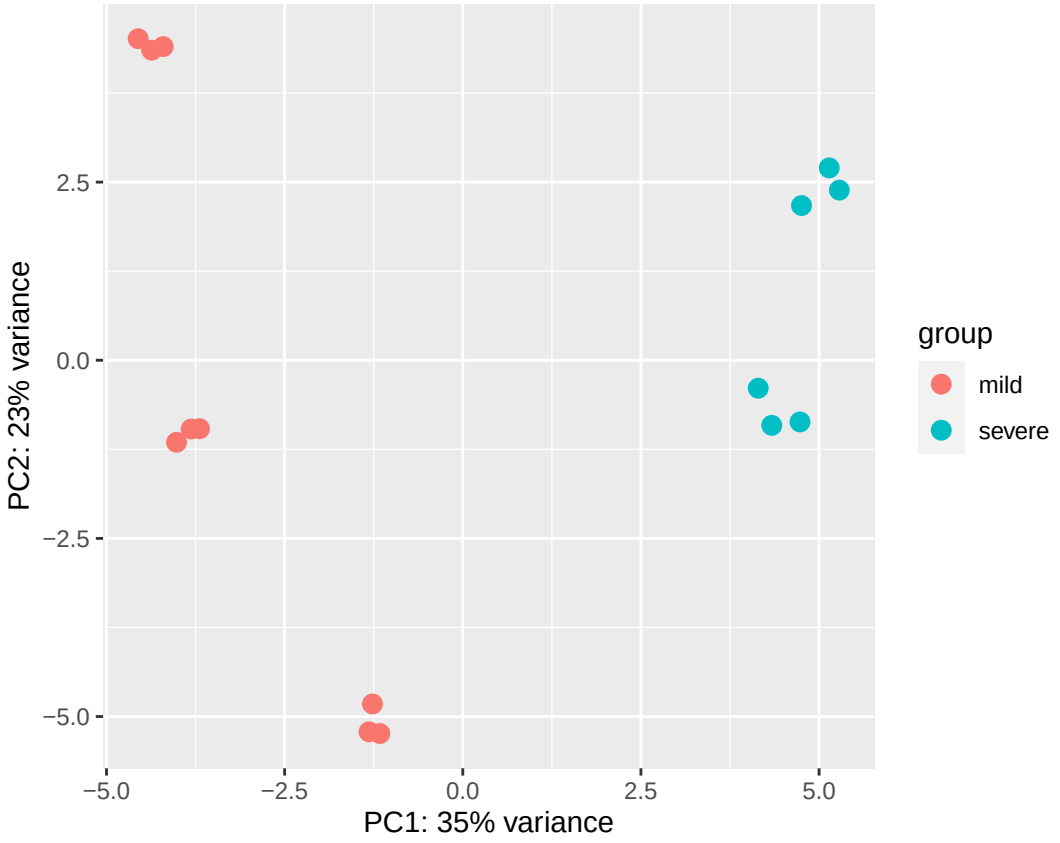
When comparing the heatmaps of DESeq2 and edgeR only one clear similarity can be seen, which is the strongly upregulated RGS16 gene. The DESeq2 and edgeR technique both show a higher log fold change

in most of the genes in s11 when compared to the s10 patient. Furthermore not a lot similarities are seen. EdgeR show a lot less genes compared to DESeq2 (after adjusting the log fold change criterium to 1.5).

3.3 PCA

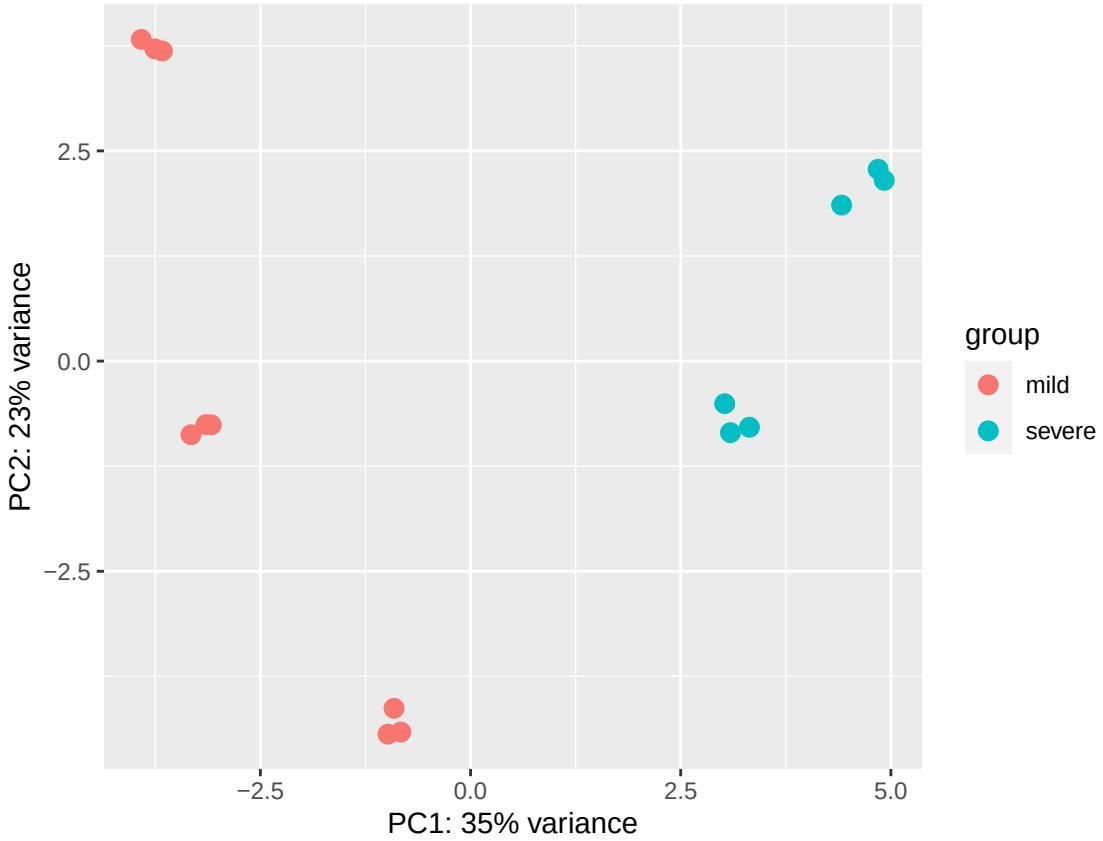
3.3.1 DESeq2

```
rlog_deseq <- rlog(dds)
plotPCA(rlog_deseq, intgroup="Condition")
```



3.3.1 EdgeR

```
edger.dds <- DESeqDataSetFromMatrix(dge$counts, colData = annotation, design = ~ Condition)
rlog_edger <- rlog(edger.dds)
plotPCA(rlog_edger, intgroup="Condition")
```



4 Differentialy Expressed Genes

After comparing the results achieved by DESeq2 and edgeR it is concluded that RGS16 is the most differentially expressed in both cases. Both techniques also annotate the MK167 as a DEG (the second most differentially expressed according to edgeR). For these reasons RGS16 and MK167 are briefly discussed.

4.1 RGS16

The RGS16 (Regulator of G protein Signalling 16) gene indirectly inhibits signal transduction. The gene product of RGS16 increasing the activity of GTPase, which hydrolysis GTP to GDP. GDP in its turn deactivates G proteins. G proteins transduce information from outside of the cell to the inside of the cell. In our case RGS16 is significantly up-regulated in the severe patients and therefore more inhibition of signal transduction will be observed. This may influence the disease severity, since signal transduction is reduced and this might result in a slower recovery. However, this is only speculation and there is no direct evidence that this results in a higher disease severity.

4.2 MKI67

The MKI67 (Marker of Proliferation Ki-67) gene plays in important role for cell proliferation. The protein product of MKI67, also known as Ki067 protein, is present during all phases of the cell cycle and is absent in resting cells. In short, when there is cell proliferation MKI67 is active. This is probably due to ongoing recovery of that patient. In other words, this up-regulated gene does probably not cause the disease severity.