Ocular disease mechanisms elucidated by genetics of human fetal retinal pigment epithelium gene expression

Lab Journal Theme07 - Gene Expression Analysis

Lisa Hu
414264
Bio-Informatica
Hanzehogeschool Groningen, ILST
Marcel Kempenaar
8 March 2022

Contents

1 Loading the Data

```r
#' Setup chunk
knitr::opts_chunk$set(cache = TRUE)
knitr::opts_chunk$set(echo = TRUE)

# Load packages
packages <- c("pander", "dplyr", "affy", "knitr", "ggplot2", "DESeq2", "pheatmap",
              "PoiClaClu", "scales", "apeglm", "EnhancedVolcano", "crayon")
invisible(lapply(packages, library, character.only = T))
require(httr)
require(jsonlite)
```

For decompressing the data, run the code chunks in the Rmd file that deem fit for your situation:

- If you downloaded the data from the official site: Decompress the data and run the Rscript `data_loading.R`.
- If you want to use the dataset delivered with the project: Run the `decompress-dataset` code chunk.

```r
#' Decompress the complete dataset
#' Use this chunk if you did not download the data from the site and want to use
#'  the delivered gzipped dataset

## Set the count.file variable to the full path of the gene file
count.file <- ""
system(paste("gzip -d", count.file))
```

After decompressing the data, the data can be read:

```r
## Read the dataset
dataset <- read.table("./gene_count.txt", sep = "\t", header = TRUE)
## Set rownames of the dataset to first column
row.names(dataset) <- dataset$Gene
## Remove the Gene column
dataset <- dataset[-1]
## Remove the first 5 rows (these rows are not genes)
dataset <- dataset[!(rownames(dataset) %in% c("__no_feature", "__ambiguous",
                     "__alignment_not_unique", "__too_low_aQual", "__not_aligned")),]

## Column indices for dataset
glucose.data <- seq(1, 48, 2)
galactose.data <- seq(2, 49, 2)
groups <- factor(rep(1:2, times=24), labels = c("Glucose", "Galactose"))
col.ordered <- c(colnames(dataset[glucose.data]), colnames(dataset[galactose.data]))

## Colors for the two sample groups (red = galactose, blue = glucose)
group.cols <- hue_pal()(2)
```

2 Exploratory Data Analysis

2.1 Data Sample

```
pander(dataset[0:5, 0:4], split.tables = 64)
```

Table 1: Table continues below

	X1_glucose	X1_galactose	X2_glucose	X2_galactose
__alignment_not_unique	0	0	0	0
__ambiguous	73052	71663	90130	114748
__no_feature	6143654	3901459	4560099	10675855
__not_aligned	0	0	0	0
__too_low_aQual	0	0	0	0

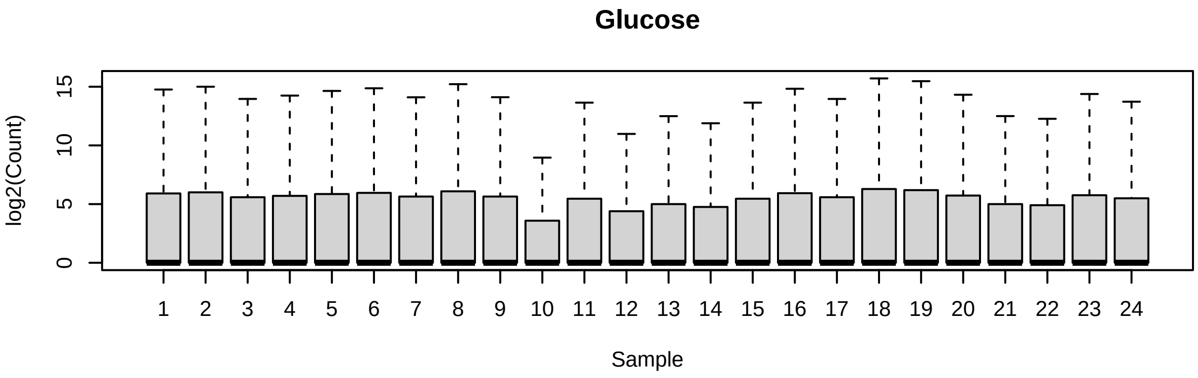
```
pander(summary(dataset[,0:6]), split.tables = 64)
```

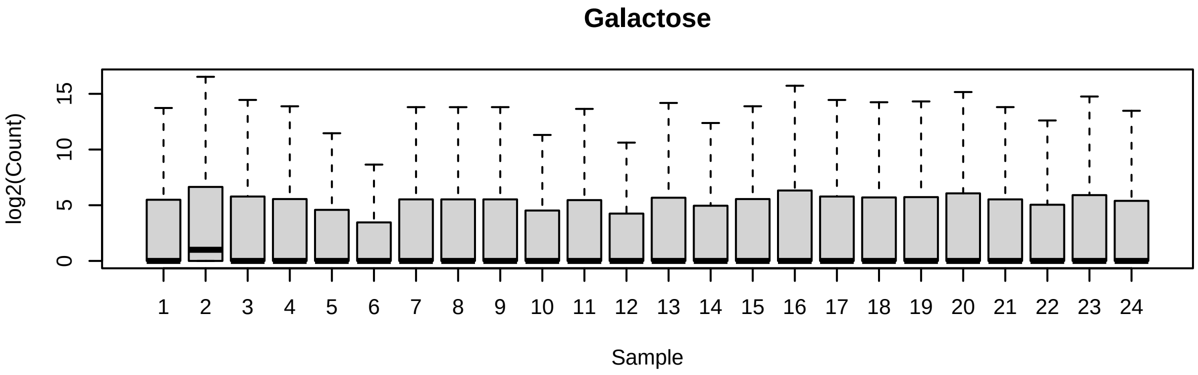
Table 3: Table continues below

X1_glucose	X1_galactose	X2_glucose	X2_galactose	X3_glucose	X3_galactose
Min. : 0	Min. : 0	Min. : 0	Min. : 0	Min. : 0	Min. : 0
1st Qu.: 0	1st Qu.: 0	1st Qu.: 0	1st Qu.: 0	1st Qu.: 0	1st Qu.: 0
Median : 0	Median : 0	Median : 0	Median : 1	Median : 0	Median : 0
Mean : 719	Mean : 549	Mean : 750	Mean : 1147	Mean : 622	Mean : 679
3rd Qu.: 60	3rd Qu.: 44	3rd Qu.: 63	3rd Qu.: 99	3rd Qu.: 47	3rd Qu.: 54
Max. :6143654	Max. :3901459	Max. :4560099	Max. :10675855	Max. :5017129	Max. :5650847

2.2 Boxplots

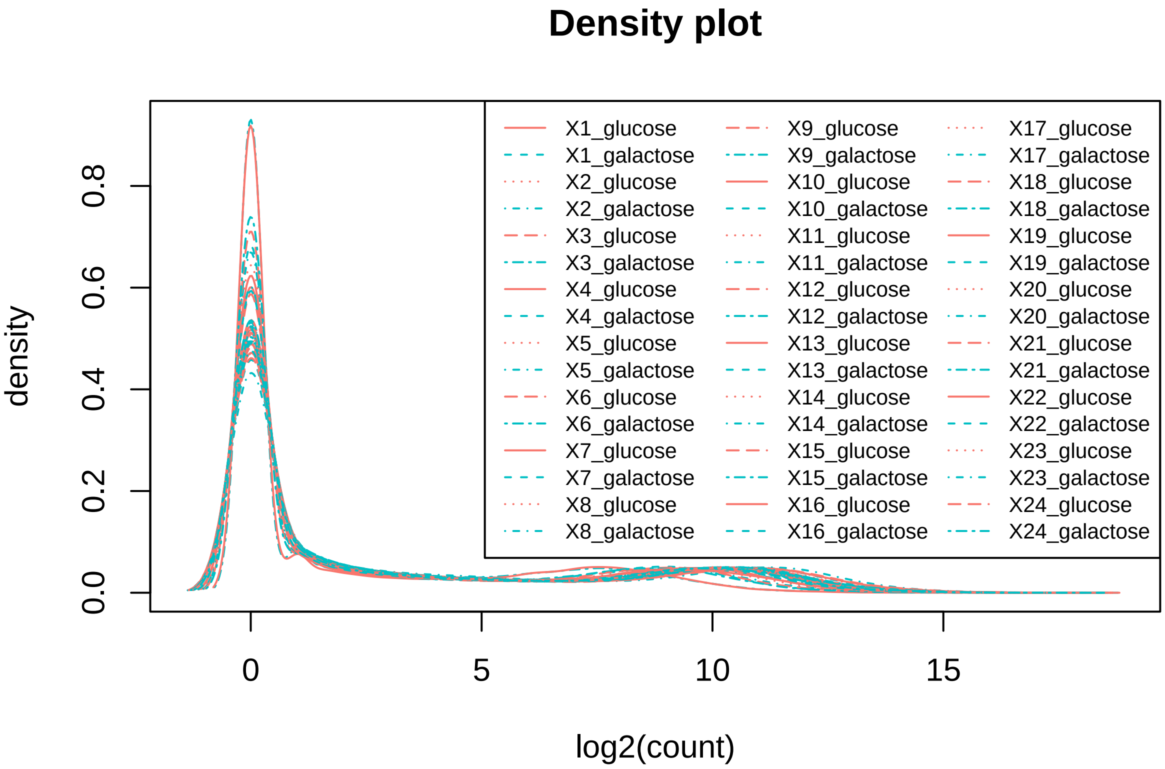
```r
layout(matrix(c(1,1,1,2,2,2), nrow = 6, ncol = 1, byrow = T))
## Glucose plot
boxplot(log2(dataset[glucose.data]+1), main = "Glucose", names = seq(1, 24),
        xlab = "Sample", ylab = "log2(Count)", outline = FALSE)
## Galactose plot
boxplot(log2(dataset[galactose.data]+1), main = "Galactose", names = seq(1, 24),
        xlab = "Sample", ylab = "log2(Count)", outline = FALSE)
```





2.3 Density Plots

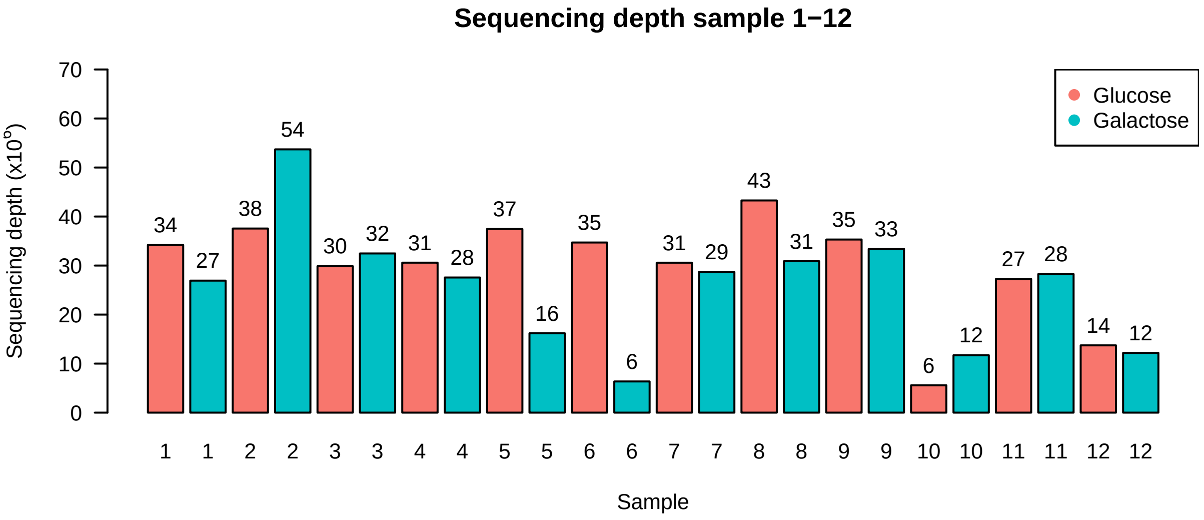
```
plotDensity(log2(dataset+1), main = "Density plot", col = group.cols,
            lty = 1:48, xlab = "log2(count)")
legend("topright", names(dataset), lty = 1:48, col = group.cols,
       cex = 0.7, ncol = 3)
```



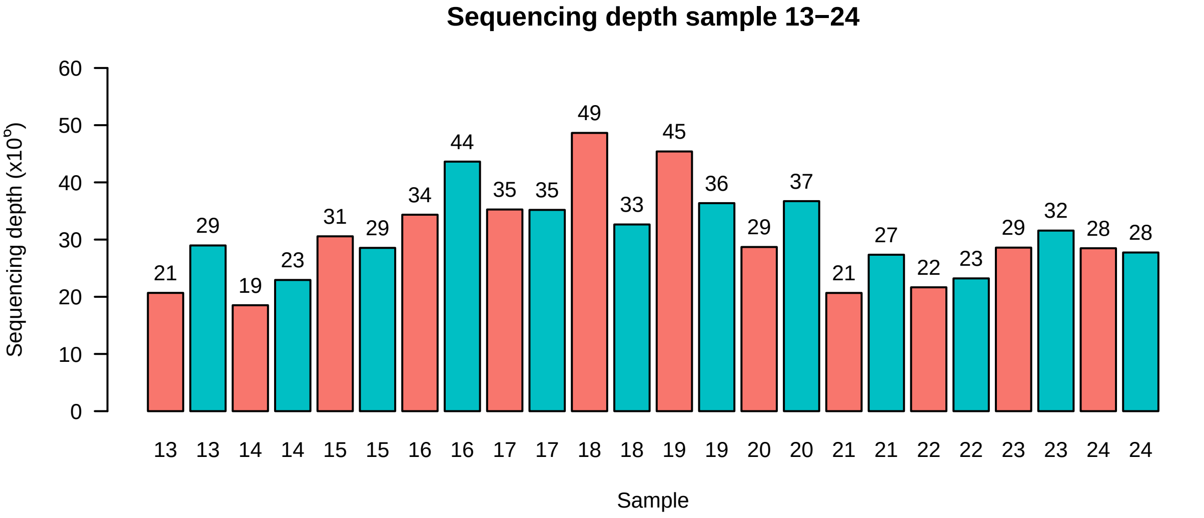
2.4 Barplots

```
layout(matrix(c(1,1,1,2,2,2), nrow = 6, ncol = 1, byrow = T))
## Barplot of first half of the data
x1 <- barplot(colSums(dataset[1:24]/ 1e6), main = "Sequencing depth sample 1-12",
              xlab = "Sample", ylab = expression("Sequencing depth (x10"^6*")"),
              ylim = c(0, 70), las = 2, col = group.cols, xaxt = 'n')
text(x = x1, y = colSums(dataset[1:24]/ 1e6),
     label = round(colSums(dataset[1:24]/ 1e6),0), pos = 3)
axis(1, at = x1, labels = rep(1:12, each = 2), tick = FALSE, cex = 0.6)
legend("topright", c("Glucose", "Galactose"), col = group.cols, pch = 19)

## Rest of the data
x2 <- barplot(colSums(dataset[25:48]/ 1e6), main = "Sequencing depth sample 13-24",
              xlab = "Sample", ylab = expression("Sequencing depth (x10"^6*")"),
              ylim = c(0, 60), las = 2, col = group.cols, xaxt = 'n')
text(x = x2, y = colSums(dataset[25:48]/ 1e6),
     label = round(colSums(dataset[25:48]/ 1e6), 0), pos = 3)
axis(1, at = x1, labels = rep(13:24, each = 2), tick = FALSE, cex = 0.6)
```



Sequencing depth sample 1−12



Sequencing depth sample 13−24

3 Normalization

```
ddsMat <- DESeqDataSetFromMatrix(countData = round(dataset),
                                 colData = data.frame(samples = names(dataset)),
                                                    design = ~ 1)
rld.dds <- vst(ddsMat)
rld <- assay(rld.dds)
sampledists <- dist(t(rld))
```

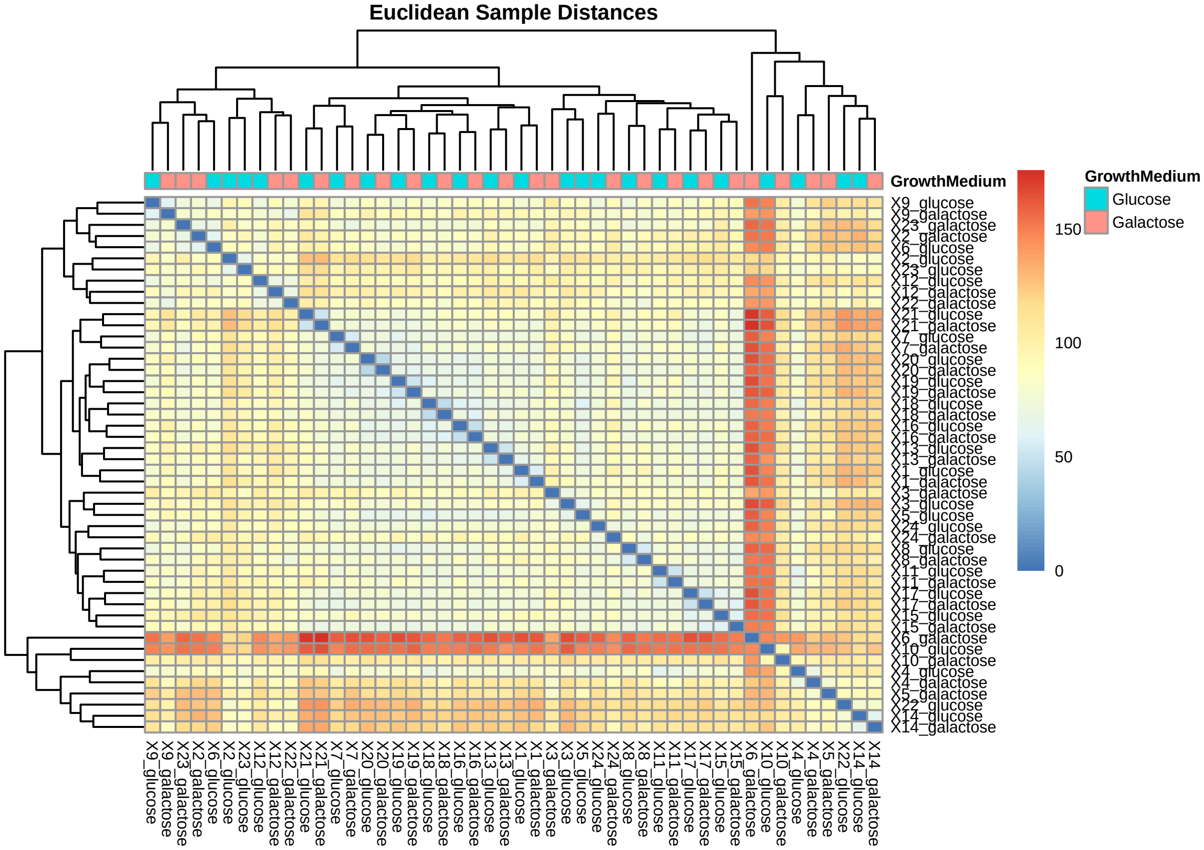
3.1 Heatmaps

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

annotation <- data.frame(GrowthMedium = groups)

rownames(annotation) <- names(dataset)

pheatmap(distMatrix, show_colnames = T,
         annotation_col = annotation,
         clustering_distance_rows = sampledists,
         clustering_distance_cols = sampledists,
         main = "Euclidean Sample Distances", fontsize= 6)
```



A plot without the clustering and ordered groups:

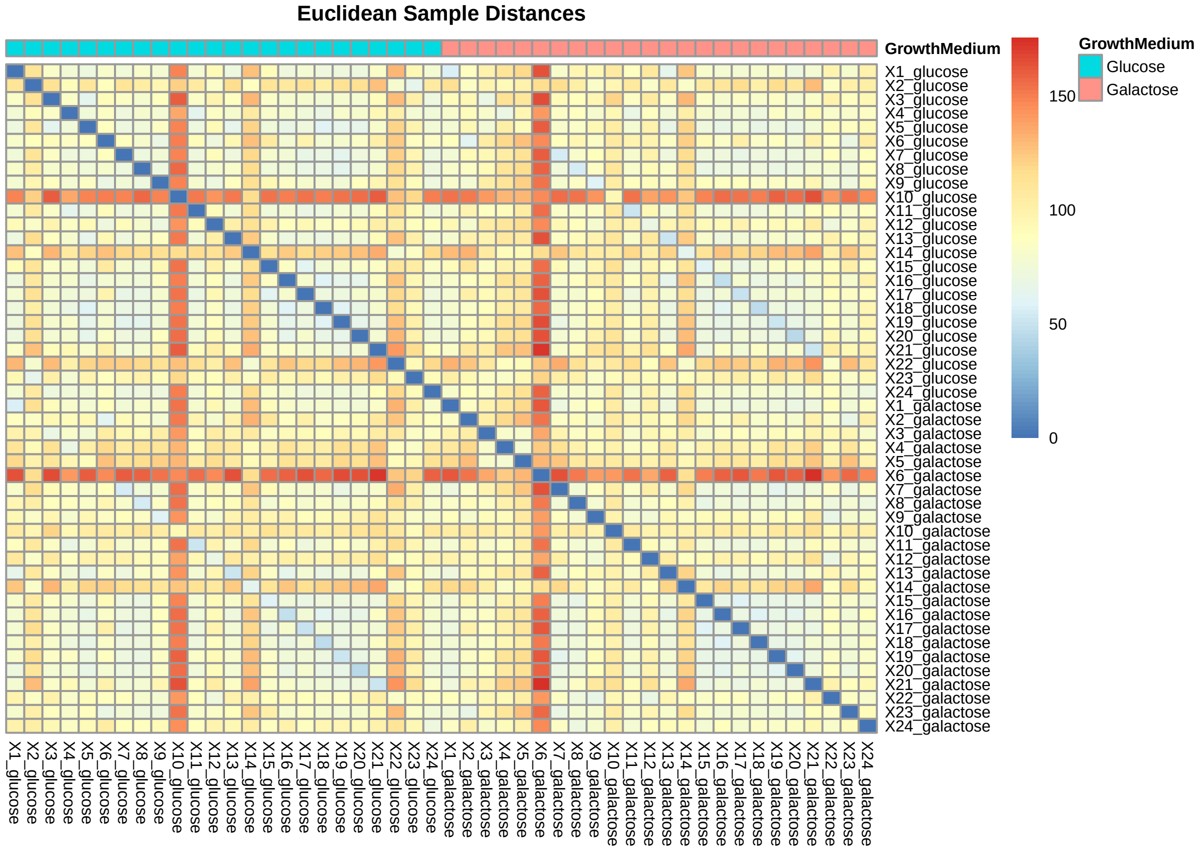
```
rld.ord <- rld[,col.ordered]
sampledists.ord <- dist(t(rld.ord))

distMatrix.ord <- as.matrix(sampledists.ord)

annotation.ord <- data.frame(GrowthMedium = factor(rep(1:2, each = 24),
                                               labels = c("Glucose", "Galactose")))

rownames(annotation.ord) <- col.ordered

pheatmap(distMatrix.ord, show_colnames = TRUE,
         annotation_col = annotation.ord, cluster_rows = FALSE, cluster_cols = FALSE,
         main = "Euclidean Sample Distances", fontsize= 6)
```

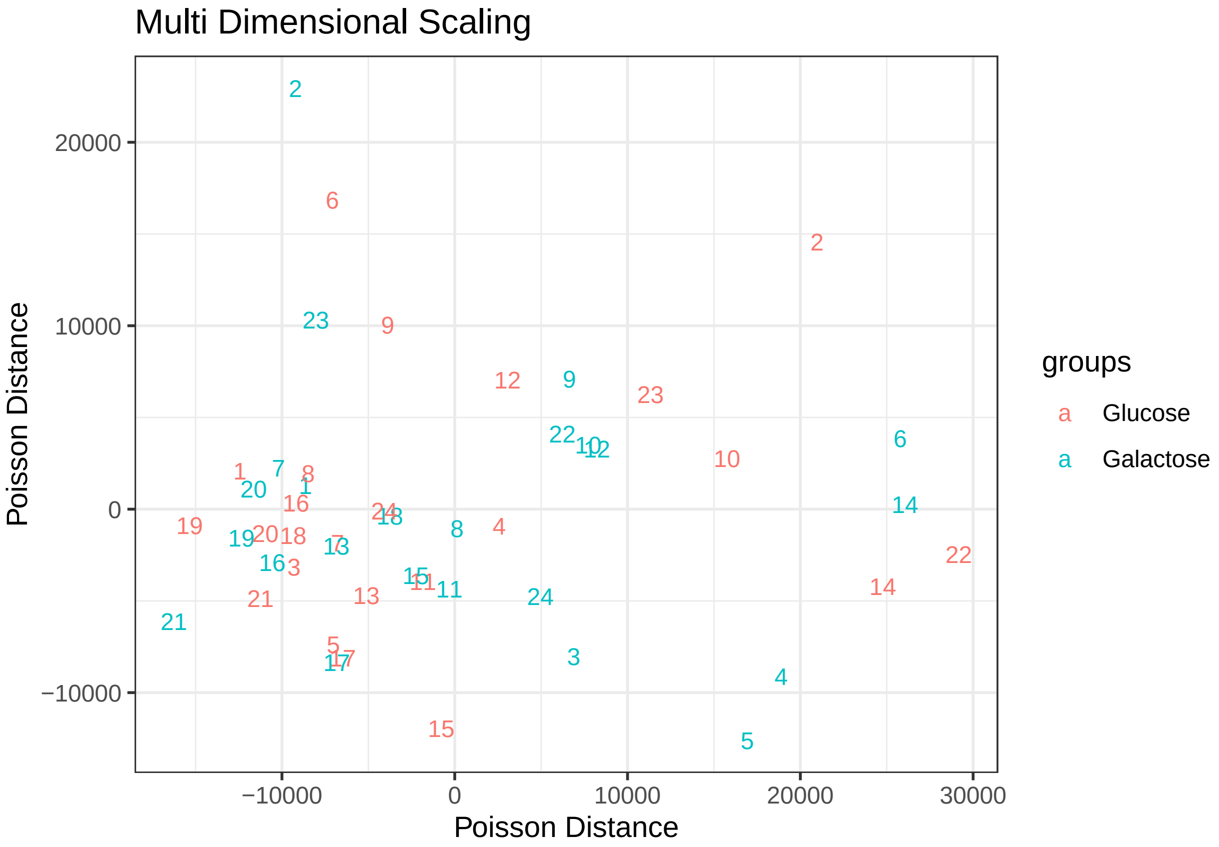


3.2 Multi Dimensional Scaling

Since the colours confirm the sample group, the sample names were simplified to only the number indicating the different samples.

```
dds <- assay(ddsMat)
poisd <- PoissonDistance(t(dds), type="deseq")
## Extract matrix with distances
poisDistMatrix <- as.matrix(poisd$dd)
## Calculate MDS for X- and Y- coordinates
mdsPoisData <- data.frame(cmdscale(poisDistMatrix))
## Readable names
names(mdsPoisData) <- c("x_coord", "y_coord")
## Annotation label
coldata <- rep(1:24, each=2)

ggplot(mdsPoisData, aes(x_coord, y_coord, color = groups, label = coldata)) +
        geom_text(size = 3) +
        ggtitle("Multi Dimensional Scaling") +
        labs(x = "Poisson Distance", y = "Poisson Distance") +
        theme_bw()
```



4 Discovering Differentially Expressed Genes (DEGs)

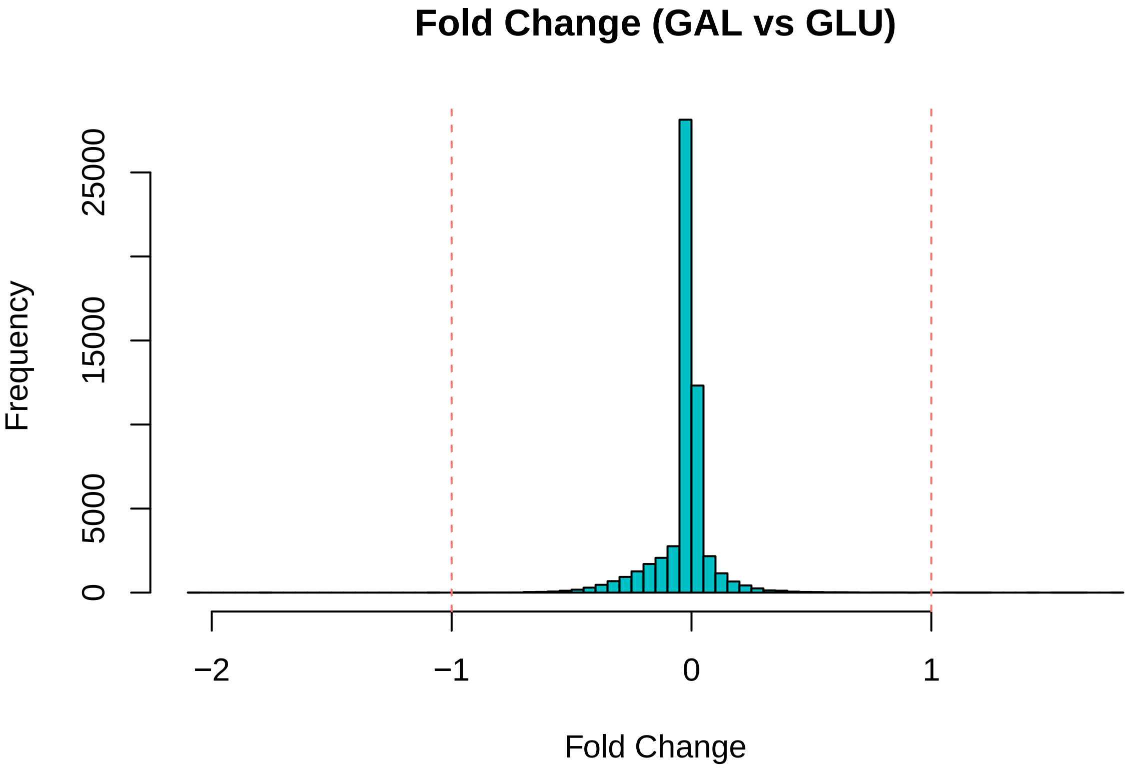
4.1 The Fold Change Value

```
fpm <- log2( (dataset / (colSums(dataset) / 1e6)) + 1 )

## New columns for average fpm
fpm$avg_glu <- rowMeans(fpm[glucose.data])
fpm$avg_gal <- rowMeans(fpm[galactose.data])

## Calculate FC
fpm$fc_galglu <- fpm$avg_gal - fpm$avg_glu

## Create histogram of the fold changes
hist(fpm$fc_galglu, main = "Fold Change (GAL vs GLU)",
     col=group.cols[2], breaks=100, xlab = "Fold Change")
## Vertical ablines
abline(v = c(1,-1), lty = 2, col = group.cols[1])
```



4.2 DESeq2 Analysis

Genes with average read count below 10 and with zero counts in more than 20% of samples (10 samples) were considered not expressed and filtered.

```r
## Create design frame
design <- data.frame(groups, row.names = colnames(dataset))
## Create new DDS object with correct design
dds <- DESeqDataSetFromMatrix(countData = dataset,
                                 colData = design, design = ~ groups)
## Keep genes with more than 1 count
dds <- dds[ rowSums(counts(dds)) > 1, ]
dds <- DESeq(dds)
dds
```

```
## class: DESeqDataSet
## dim: 36435 48
## metadata(1): version
## assays(6): counts mu ... replaceCounts replaceCooks
## rownames(36435): ENSG00000000005.5 ENSG00000000419.8 ...
##   ENSGR0000002586.13 ENSGR0000169100.8
## rowData names(23): baseMean baseVar ... maxCooks replace
## colnames(48): X1_glucose X1_galactose ... X24_glucose X24_galactose
## colData names(3): groups sizeFactor replaceable
```

```r
## Results
dds.res <- results(dds, alpha=0.05)

## Filter out reads below 10
dds.filtered <- dds.res[!dds.res$baseMean < 10,]

## Shrinkage
resultsNames(dds)
```

```
## [1] "Intercept"                   "groups_Galactose_vs_Glucose"
```

```r
lfc.gal <- lfcShrink(dds, coef = "groups_Galactose_vs_Glucose", res = dds.res,
                     type = "apeglm")
lfc.gal
```

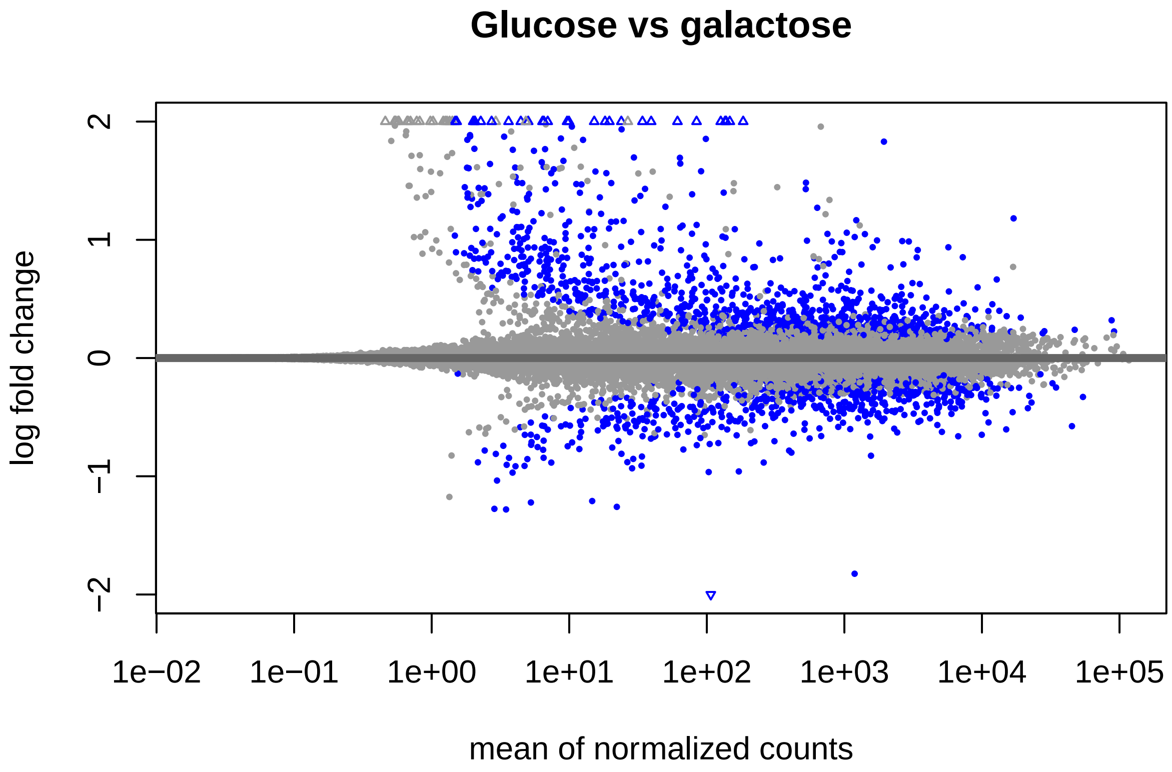
```
## log2 fold change (MAP): groups Galactose vs Glucose
## Wald test p-value: groups Galactose vs Glucose
## DataFrame with 36435 rows and 5 columns
##                       baseMean log2FoldChange     lfcSE      pvalue       padj
##                      <numeric>      <numeric> <numeric>   <numeric>  <numeric>
## ENSG00000000005.5    0.0298412     0.00278716 0.1165906  0.90906754         NA
## ENSG00000000419.8  589.0022929     0.09818758 0.0796091  0.11074211  0.3905760
## ENSG00000000457.9  390.5569233    -0.11394559 0.0450573  0.00596489  0.0673552
## ENSG00000000460.12 111.5560416    -0.08659705 0.0886569  0.17707492  0.4917964
## ENSG00000000938.8    0.3970255     0.01549200 0.1172466  0.28744653         NA
## ...                        ...            ...       ...         ...        ...
## ENSG00000273489.1    0.8747360   -0.002918776  0.114659 0.902217532         NA
## ENSG00000273492.1   63.7033055   -0.419107808  0.152604 0.000355189 0.00897252
## ENSG00000273493.1    1.4199441    0.039972643  0.122618 0.173645787         NA
## ENSGR0000002586.13   0.0933423    0.006372662  0.116774 0.741274602         NA
## ENSGR0000169100.8    0.0336766   -0.000606336  0.116518 0.973989898         NA
```

```
summary(lfc.gal)
```

```
##
## out of 36434 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 1062, 2.9%
## LFC < 0 (down)     : 688, 1.9%
## outliers [1]       : 0, 0%
## low counts [2]     : 13422, 37%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

4.2.1 MA Plot

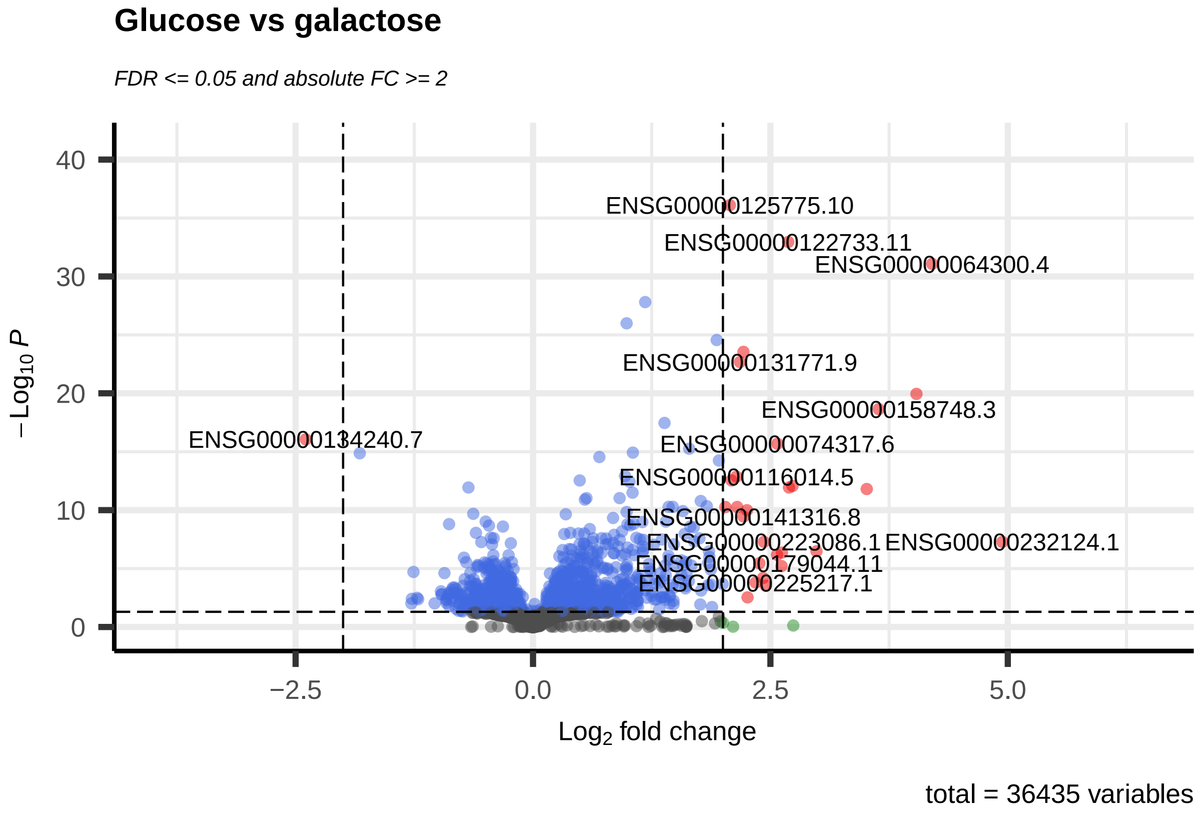
```
DESeq2::plotMA(lfc.gal, main = "Glucose vs galactose", ylim = c(-2, 2))
```



5 Data analysis and Visualization

5.1 Volcano Plot

```
EnhancedVolcano(lfc.gal, x = 'log2FoldChange', y = 'padj', lab=rownames(lfc.gal),
                title = "Glucose vs galactose",
                subtitle = bquote(italic('FDR <= 0.05 and absolute FC >= 2')),
                # Change text and icon sizes
                labSize = 3, pointSize = 1.5, axisLabSize=10, titleLabSize=12,
                subtitleLabSize=8, captionLabSize=10,
                # Disable legend
                legendPosition = "none",
                # Set cutoffs
                pCutoff = 0.05, FCcutoff = 2)
```



The points in red in this plot are important to find out what genes are differentially expressed. To do this, we can subset the data (`lfc.gal`) to filter the red dots:

```
genes <- subset(lfc.gal, lfc.gal[,'log2FoldChange'] > 2 & -log10(lfc.gal[,'padj']) > 0.05
                | lfc.gal[,'log2FoldChange'] < -2)
```

The rownames are gene IDs from the Ensembl database. Using an Ensembl ID to Gene Symbol converter[1], the gene names can be found:

```
## Get the IDs
url <- "https://biotools.fr/human/ensembl_symbol_converter/"
ids <- rownames(genes)
ids_json <- toJSON(ids)

## Create the request
body <- list(api = 1, ids = ids_json)
req <- POST(url, body = body)

output <-  fromJSON( content(req, "text"), flatten = TRUE )
df <- data.frame(unlist(output))
colnames(df) <- "Gene symbol"
pander(df, split.tables = 64)
```

	Gene symbol
ENSG00000064300.4	NGFR
ENSG00000074317.6	SNCB
ENSG00000116014.5	KISS1R
ENSG00000122733.11	PHF24
ENSG00000125775.10	SDCBP2
ENSG00000131771.9	PPP1R1B
ENSG00000134240.7	HMGCS2
ENSG00000141316.8	SPACA3
ENSG00000158014.10	SLC30A2
ENSG00000158748.3	HTR6
ENSG00000172482.4	AGXT
ENSG00000172901.15	LVRN
ENSG00000173110.6	HSPA6
ENSG00000173930.8	SLCO4C1
ENSG00000179044.11	EXOC3L1
ENSG00000185303.11	SFTPA2
ENSG00000186265.5	BTLA
ENSG00000214955.5	AP000317.1
ENSG00000223086.1	RNA5SP155
ENSG00000225217.1	HSPA7
ENSG00000225794.1	AC073321.1
ENSG00000228793.1	AL138881.1
ENSG00000231013.1	AC013275.1
ENSG00000232124.1	AP001057.1
ENSG00000243648.1	AC109454.1
ENSG00000250366.2	TUNAR
ENSG00000254607.2	AP001783.1
ENSG00000264063.1	MIR3687-2
ENSG00000264462.1	MIR3648-2
ENSG00000270066.2	AL356488.2

[1] https://www.biotools.fr/human/ensembl_symbol_converter