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

```
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")
invisible(lapply(packages, library, character.only = T))
```

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.

```
#' 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:

```
## 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]

## Indices for dataset
galactose.data <- seq(2, 49, 2)
glucose.data <- seq(1, 48, 2)

## A new column with all the means of each row for both groups
dataset$Glucose_mean <- rowMeans(dataset[glucose.data])
dataset$Galactose_mean <- rowMeans(dataset[galactose.data])
```

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
___alignment_not_unique	0	0
___ambiguous	73052	71663
___no_feature	6143654	3901459
___not_aligned	0	0
___too_low_aQual	0	0

	X2_glucose	X2_galactose
___alignment_not_unique	0	0
___ambiguous	90130	114748
___no_feature	4560099	10675855
___not_aligned	0	0
___too_low_aQual	0	0

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

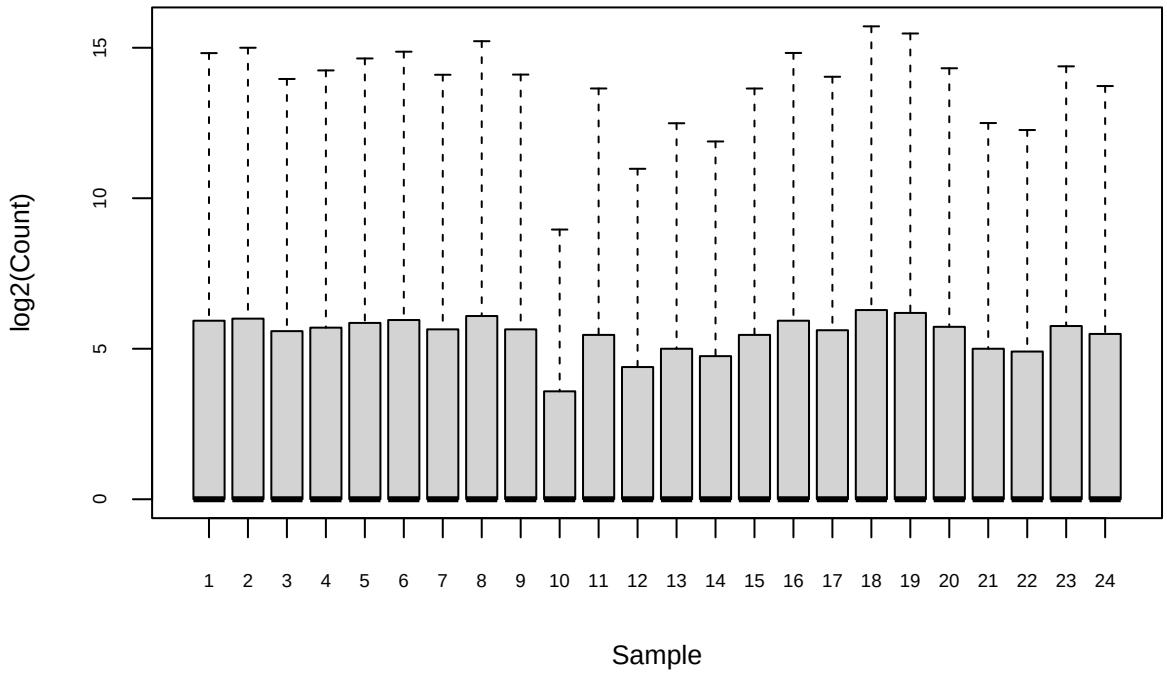
Table 3: Table continues below

X1_glucose	X1_galactose	X2_glucose
Min. : 0	Min. : 0	Min. : 0
1st Qu.: 0	1st Qu.: 0	1st Qu.: 0
Median : 0	Median : 0	Median : 0
Mean : 719	Mean : 549	Mean : 750
3rd Qu.: 60	3rd Qu.: 44	3rd Qu.: 63
Max. :6143654	Max. :3901459	Max. :4560099

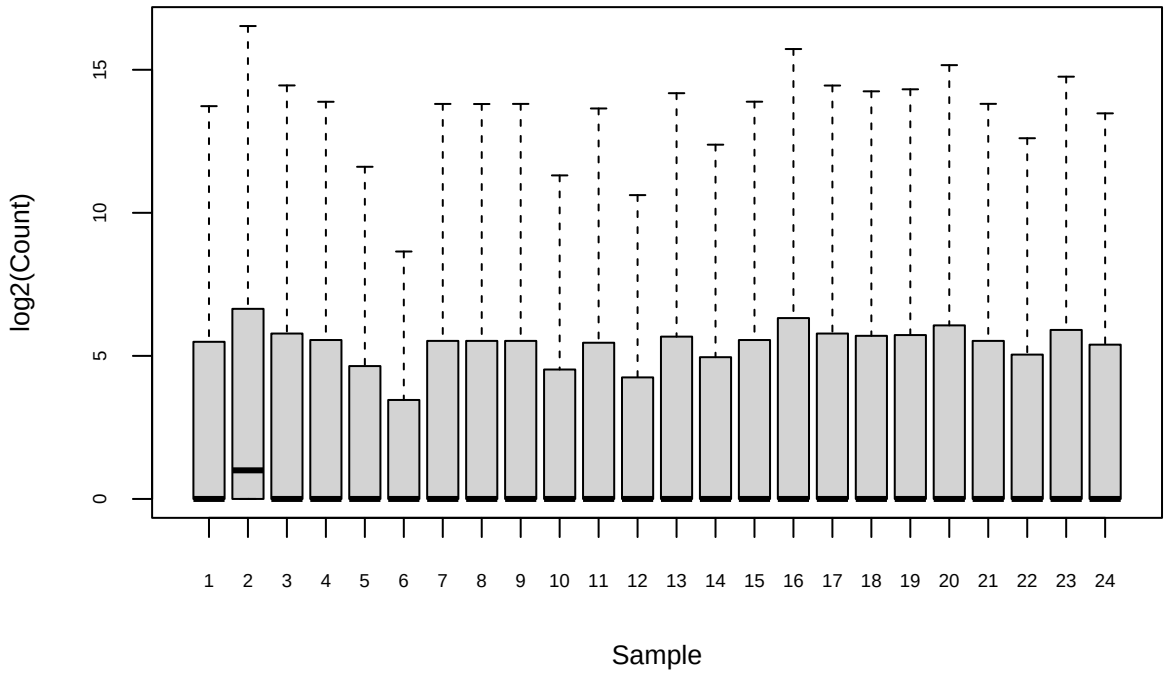
X2_galactose	X3_glucose	X3_galactose
Min. : 0	Min. : 0	Min. : 0
1st Qu.: 0	1st Qu.: 0	1st Qu.: 0
Median : 1	Median : 0	Median : 0
Mean : 1147	Mean : 622	Mean : 679
3rd Qu.: 99	3rd Qu.: 47	3rd Qu.: 54
Max. :10675855	Max. :5017129	Max. :5650847

2.2 Plots for insight

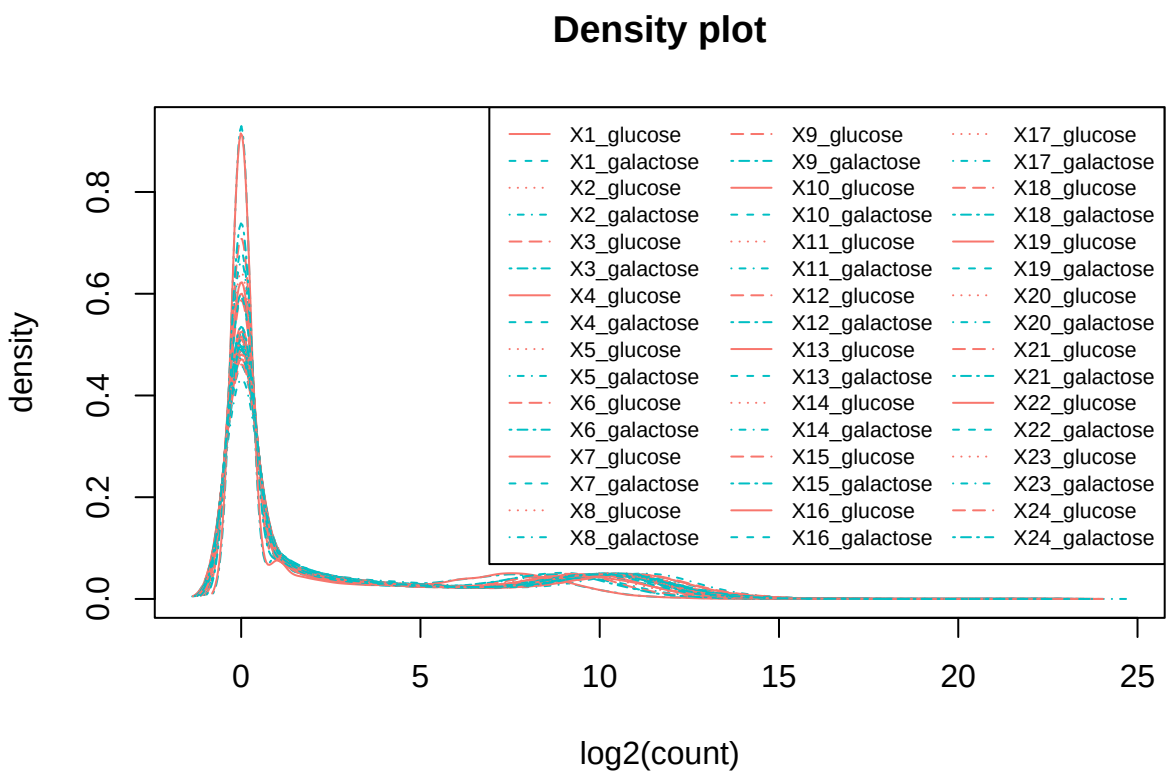
```
boxplot(log2(dataset[glucose.data]+1), main = "Glucose", names = seq(1, 24),
        xlab = "Sample", ylab = "log2(Count)", outline = FALSE,
        cex.axis = 0.6, cex.lab = 0.8)
```




```
boxplot(log2(dataset[galactose.data]+1), main = "Galactose", names = seq(1, 24),
        xlab = "Sample", ylab = "log2(Count)", outline = FALSE,
        cex.axis = 0.6, cex.lab = 0.8)
```

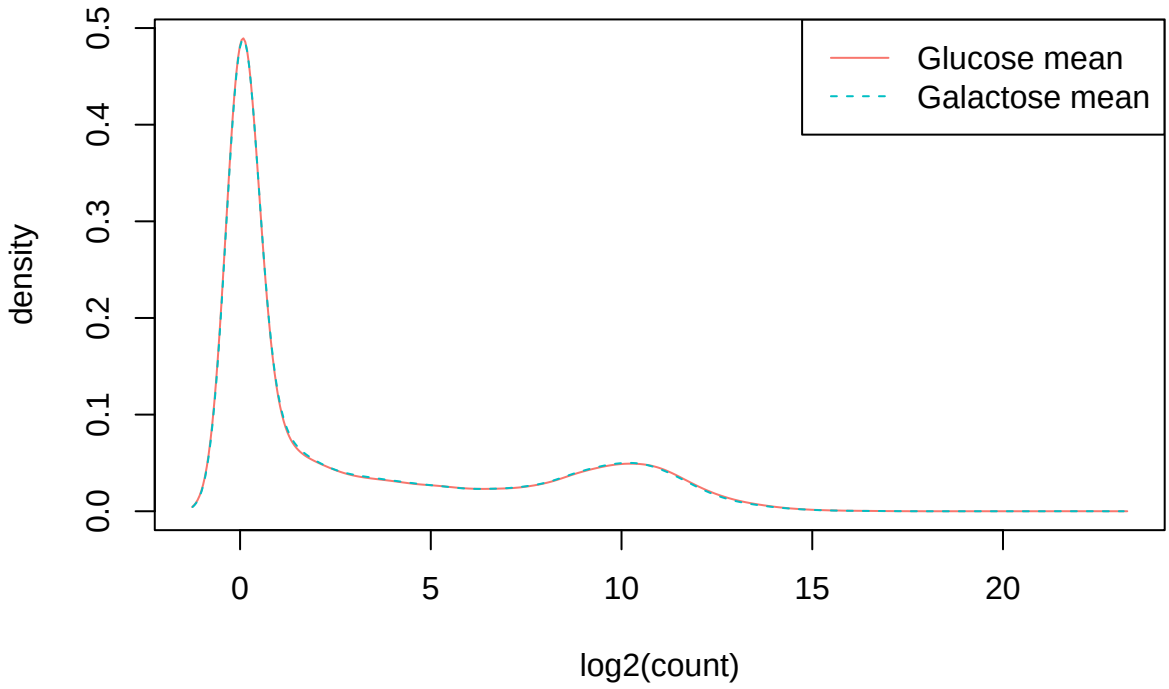


```
## Colors for the two groups (red = galactose, blue = glucose)
group.cols <- hue_pal()(2)
plotDensity(log2(dataset[1:48]+1), main = "Density plot", col = group.cols,
            lty = 1:48, xlab = "log2(count)")
legend("topright", names(dataset[1:48]), lty = 1:48, col = group.cols,
       cex = 0.7, ncol = 3)
```

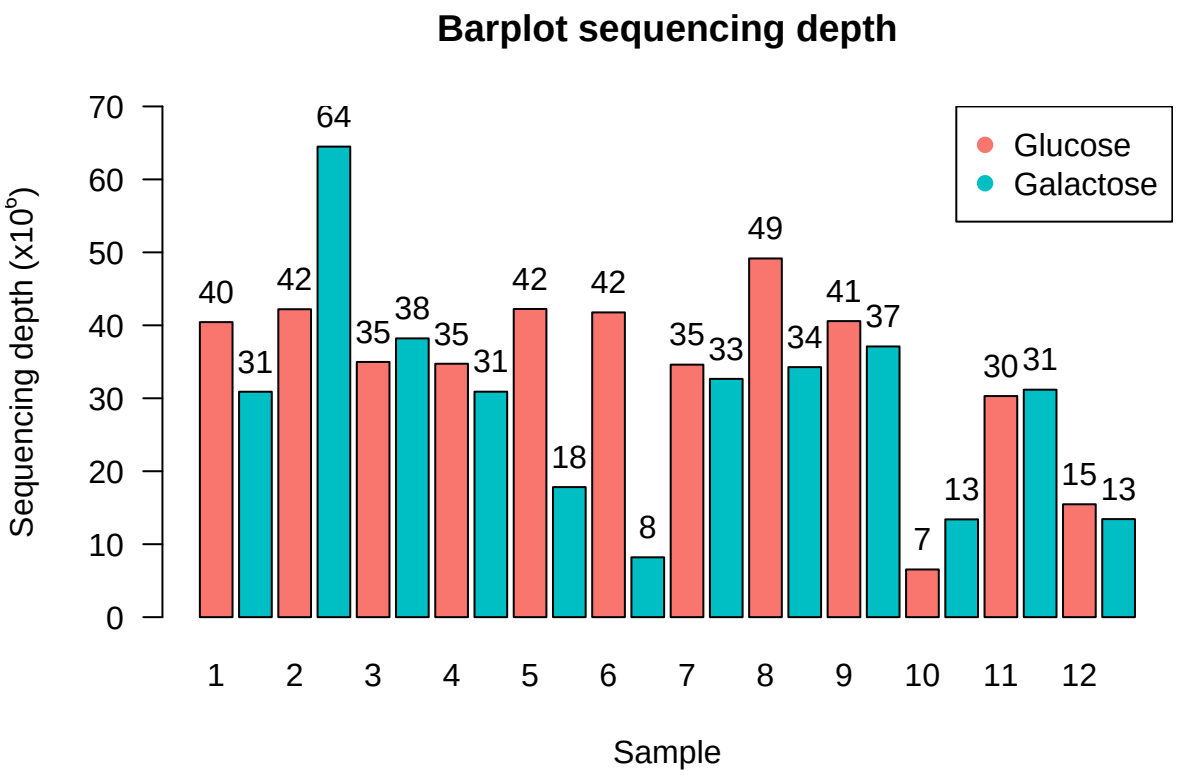


The density plot can be hard to read as is, so by taking the average of each gene per sample group, a more simplified density plot can be created:

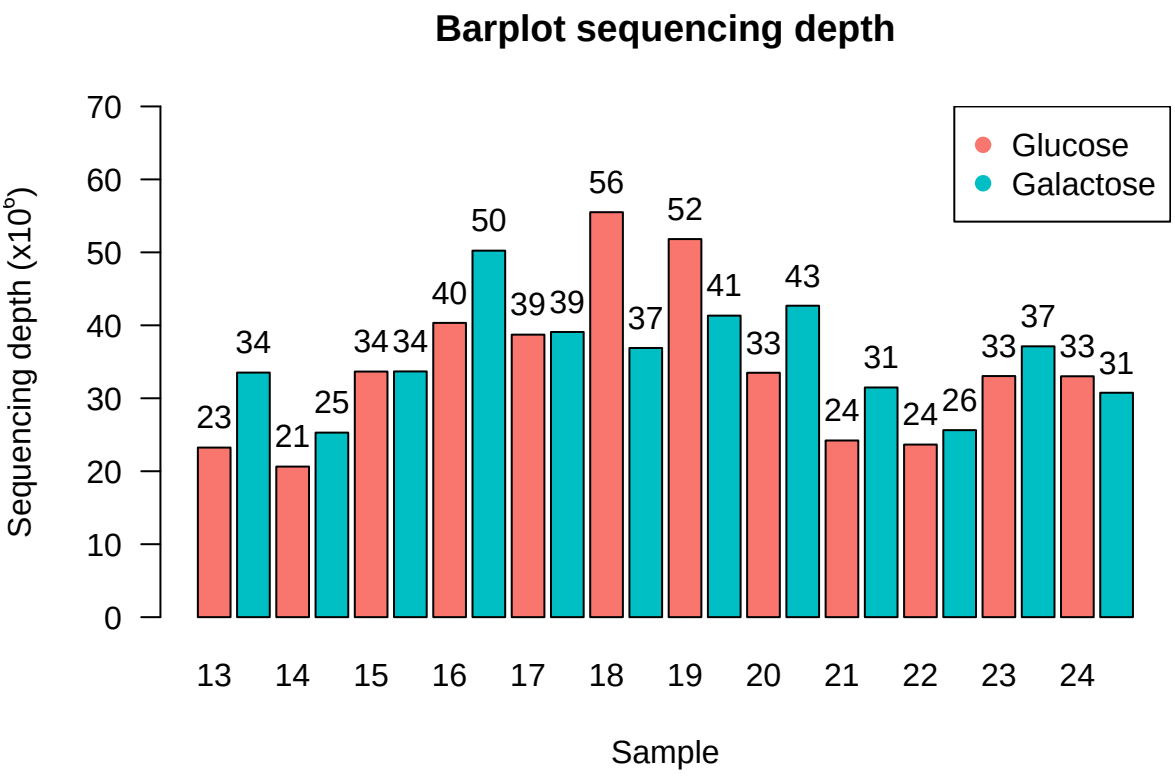
```
plotDensity(log2(dataset[,c("Glucose_mean", "Galactose_mean")]+1), main = "Density plot",
            col = group.cols, lty = c(1,2), xlab = "log2(count)")
legend("topright", c("Glucose mean", "Galactose mean"), lty = c(1,2), col = group.cols)
```




```
## Barplot of first half of the data
x1 <- barplot(colSums(dataset[1:24]/ 1e6), main = "Barplot sequencing depth",
              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 = "Barplot sequencing depth",
              xlab = "Sample", ylab = expression("Sequencing depth (x10"^6*")"),
              ylim = c(0, 70), 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)
legend("topright", c("Glucose", "Galactose"), col = group.cols, pch = 19)
```


Barplot sequencing depth
Sequencing depth (x10^6)
70
60
50
40
30
20
10
0
Glucose
Galactose
23
34
21
25
34
34
40
50
39
39
56
37
52
41
33
43
24
31
24
26
33
37
33
31
13
14
15
16
17
18
19
20
21
22
23
24
Sample

3 Normalization

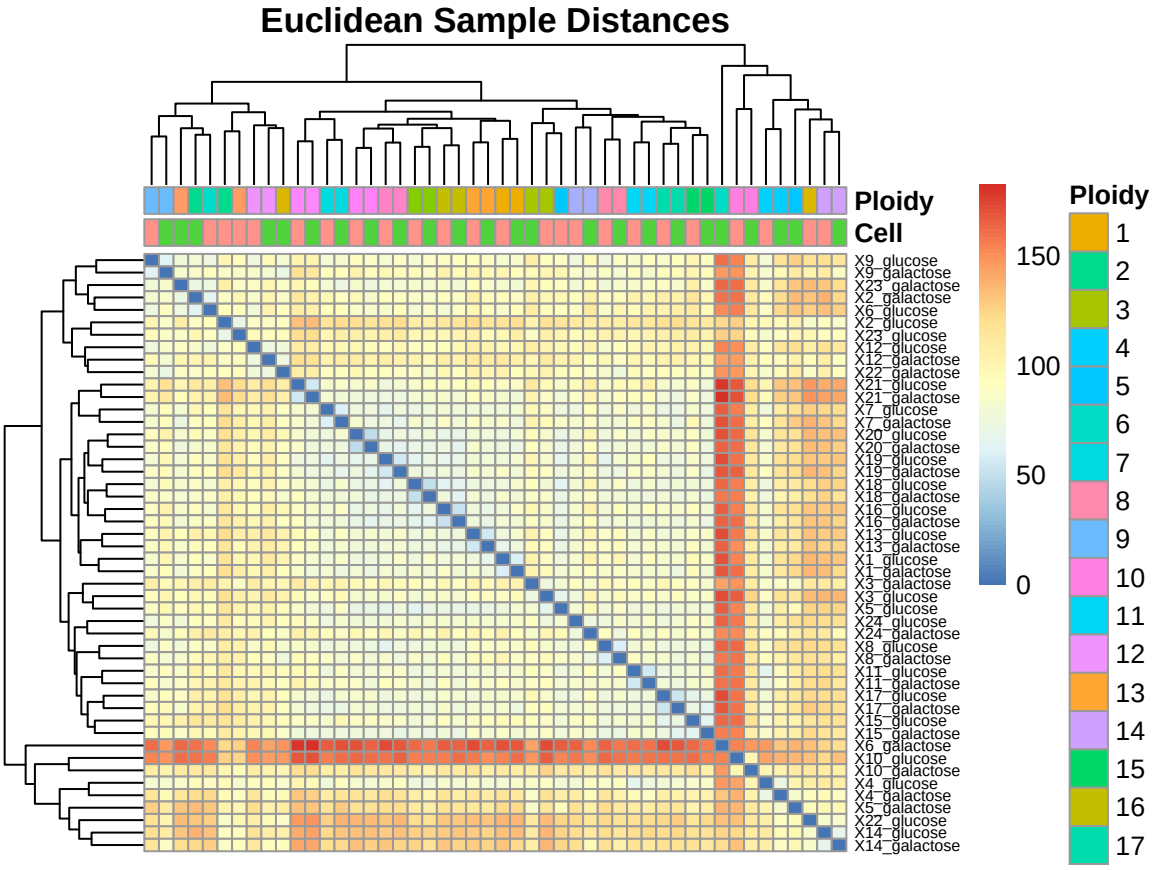
```
ddsMat <- DESeqDataSetFromMatrix(countData = round(dataset[1:48]),
                                 colData = data.frame(samples = names(dataset[1:48])),
                                                      design = ~ 1)
rld.dds <- vst(ddsMat)
rld <- assay(rld.dds)
sampledists <- dist(t(rld))
```

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

annotation <- data.frame(Cell = factor(rep(1:2, times = 24),
                                       labels = c("Glucose", "Galactose")),
                         Ploidy = factor(rep(1:24, each = 2), labels = as.character(1:24)))

rownames(annotation) <- names(dataset[1:48])

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




```
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 labels
groups <- factor(rep(1:2, times=24), labels = c("Glucose", "Galactose"))
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() +
        theme(legend.position = "top")
```