Lab Journal - Gene Expression Analysis

Noise exposures causing hearing loss generate proteotoxic stress and activate the proteostasis network

Student: Vincent Talen

Student number: 389015

Klas: BFV2

Class: Bio-Informatica

Institute: Institute for Life Science & Technology

Teacher: Marcel Kempenaar

Date: 2022-03-25

Contents

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(cache = TRUE)

# Load all packages/libraries
packages <- c("affy", "pander", "scales", "BiocParallel", "DESeq2")
invisible(lapply(packages, library, character.only = TRUE))
remove(packages)
register(MulticoreParam(6))
```

1 Exploratory Data Analysis

1.1 Preparing the data

To start this project out the data set will be loaded into the memory so the statistics can be performed on it. The first few lines will be shown to show what the contents of the data frame are like and dim() is used to show the length of the data set.

```
# Load in the data set into a data frame
myData <- read.table("GSE160639_RawReadCount.txt", header = T, sep = "\t")
# Change the int row names to the gene_id column and delete the gene_id column afterwards
row.names(myData) <- myData$gene_id
myData <- myData[-1]
# Show the first few lines of the data frame
head(myData)
```

```
##                    A1_70dB A2_70dB A3_70dB A4_70dB B1_94dB B2_94dB B3_94dB
## ENSMUSG00000064351  807021  844885  746507  600313  455027  512010  325579
## ENSMUSG00000069919  366445  395997  269558  284141  302984  427023  410969
## ENSMUSG00000052305  243346  341576  226918  233215  249790  301111  480944
## ENSMUSG00000015090  352912  323278  255015  234177  318835  364719  366617
## ENSMUSG00000064370  381672  361431  312361  262815  222349  253774  159885
## ENSMUSG00000001506  312965  305261  233933  193533  256679  279963  202360
##                    B4_94dB C1_105dB C2_105dB C4_105dB C5_105dB
## ENSMUSG00000064351  416803   745783   521567   444325   761621
## ENSMUSG00000069919  408831   390024   343923   474316   401705
## ENSMUSG00000052305  634849   320317   371479   321232   328205
## ENSMUSG00000015090  381912   307553   251633   333419   258235
## ENSMUSG00000064370  191874   360531   267048   265633   310993
## ENSMUSG00000001506  216174   340531   230902   196198   222483
```

```
# Print the amount of genes and columns
cat("Amount of genes:", dim(myData)[1], "\tColumns in dataframe", dim(myData)[2])
```

```
## Amount of genes: 35496   Columns in dataframe 12
```

By using the str() function the structure of the dataframe/dataset can be seen. Every column consists only of int values, so only the count data.

```
str(myData)
```

For later use the column indices are assigned to variables. This is so later on mistakes won't be made accidentally selecting a wrong column and to simply make it easier to use.

```
group.names <- colnames(myData)
SPL_70dB <- 1:4
SPL_94dB <- 5:8
SPL_105dB <- 9:12
```

1.2 Early Statistics

Summary

To get to know the data set more the data will be visualized in multiple ways. Seeing how everything is grouped and divided gives us a deeper understanding of what the quality of the data set is and what types of statistics will have to be performed.

First a simple summary performing 6-number-statistic on the data columns will be done, this gives a summary overview of each replication for all groups.

```
# Disable printing 'table continues' lines between split sections of the table
panderOptions("table.continues", "")
# Pretty print the summary of the data frame
pander::pander(summary(myData), split.tables = 69)
```

A1_70dB	A2_70dB	A3_70dB	A4_70dB
Min. : 0.0	Min. : 0	Min. : 0.0	Min. : 0.0
1st Qu.: 1.0	1st Qu.: 0	1st Qu.: 0.0	1st Qu.: 0.0
Median : 21.0	Median : 20	Median : 16.0	Median : 16.0
Mean : 1161.2	Mean : 1192	Mean : 973.6	Mean : 877.3
3rd Qu.: 909.2	3rd Qu.: 915	3rd Qu.: 749.2	3rd Qu.: 690.0
Max. :807021.0	Max. :844885	Max. :746507.0	Max. :600313.0

B1_94dB	B2_94dB	B3_94dB	B4_94dB
Min. : 0.0	Min. : 0	Min. : 0	Min. : 0
1st Qu.: 0.0	1st Qu.: 0	1st Qu.: 0	1st Qu.: 0
Median : 16.0	Median : 17	Median : 15	Median : 18
Mean : 941.7	Mean : 1058	Mean : 910	Mean : 1091
3rd Qu.: 730.0	3rd Qu.: 817	3rd Qu.: 692	3rd Qu.: 813
Max. :455027.0	Max. :512010	Max. :480944	Max. :634849

C1_105dB	C2_105dB	C4_105dB	C5_105dB
Min. : 0	Min. : 0.0	Min. : 0.0	Min. : 0
1st Qu.: 0	1st Qu.: 0.0	1st Qu.: 0.0	1st Qu.: 0
Median : 18	Median : 15.0	Median : 16.0	Median : 17
Mean : 1134	Mean : 913.6	Mean : 903.1	Mean : 1045

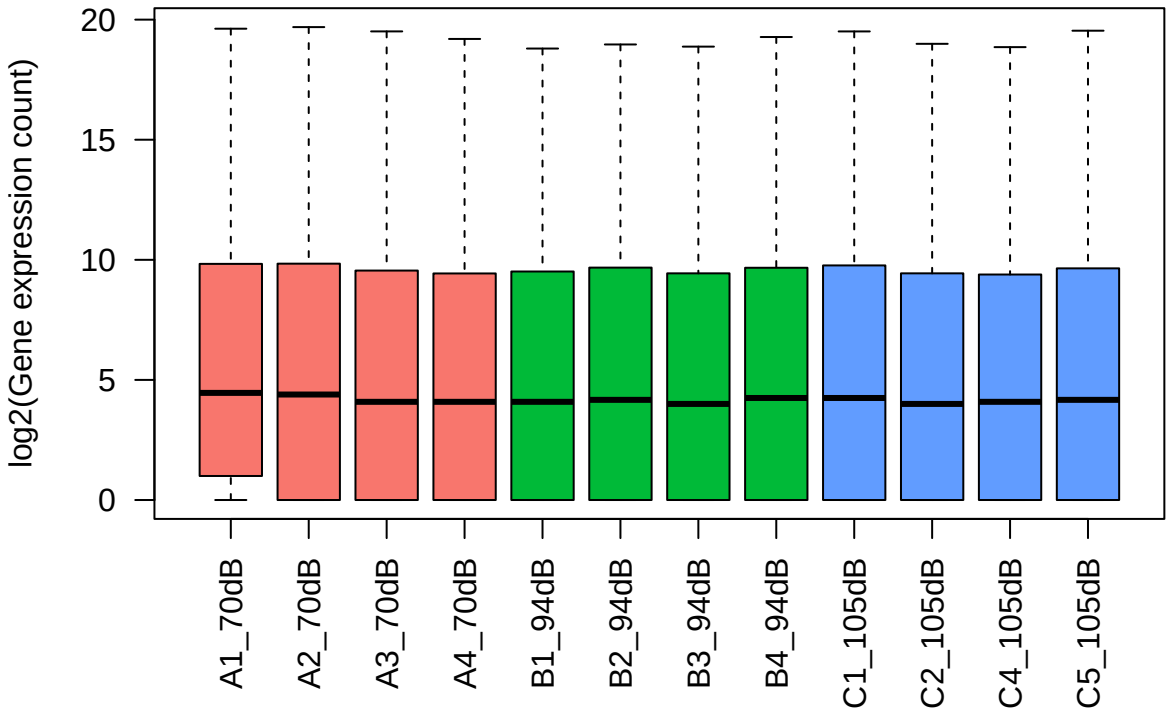
C1_105dB	C2_105dB	C4_105dB	C5_105dB
3rd Qu.: 870 Max. :745783	3rd Qu.: 692.0 Max. :521567.0	3rd Qu.: 669.0 Max. :474316.0	3rd Qu.: 799 Max. :761621

At first glance of the summary it can be seen that the expression at the lowest (70) dB genes seem to be expressed the most and declining whilst the SPL increases. It also looks like there is a noticeable difference between each replication.

Boxplot

Now a boxplot will be made for each data column, visualizing this will help quickly spot irregularities after which further detailed statistics need to be performed to do any quality control. To increase the visibility a log2 transformation will be done to show changes more informative.

```
# Create a color pallet variable to use in graphs
myColors = rep(hue_pal()(3), each = 4)
# Plot the boxplot with log2(1) scale.
boxplot(log2(myData + 1), ylab = "log2(Gene expression count)",
        outline = FALSE, col = myColors, las = 2)
```

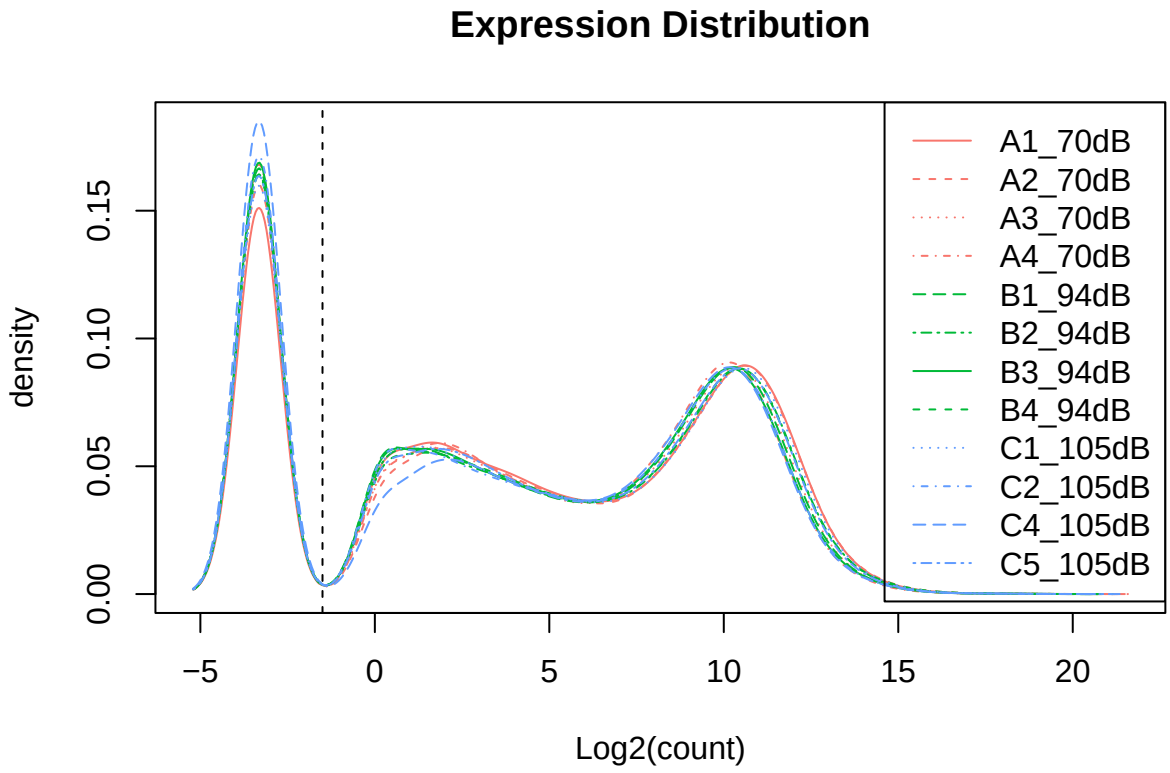


Density plot

Another useful way of visualizing the data set is using a density plot, it shows a distribution of the log2-transformed count data for all samples which makes spotting problems easier.

```
# Plot a density plot with log2(0.1) scale and
# with a different color and line type for each replication
plotDensity(log2(myData + 0.1), col=myColors,
            lty=c(1:ncol(myData)), xlab='Log2(count)',
            main='Expression Distribution')

# Add a legend and a vertical line
legend('topright', names(myData), lty=c(1:ncol(myData)), col=myColors)
abline(v=-1.5, lwd=1, col='black', lty=2)
```



Looking at the density plot of the expression distribution it looks like all replications have a relatively similar amount of reads sequenced since all the peaks lie together.