Exploratory Data Analysis: Which genes and their expression levels may be associated with developing familial alzheimer's disease?

Orfeas Gkourlias

2022-09-25

Codebook

```
codebook <- read_delim(file = "codebook.txt",
                       delim = ";",
                       show_col_types=FALSE)

knitr::kable(codebook)
```

column	description
ID_REF	Probe ID
GSM701543	Parkinson's sample 1 RNA Concentration
GSM701543	Parkinson's sample 2 RNA Concentration
GSM701544	Alzheimer's sample 1 RNA Concentration
GSM701545	Alzheimer's sample 2 RNA Concentration
no.mutation	Means of Parkinson's RNA Concentration
mutation	Means of Parkinson's RNA Concentration
log.2.fold.change	Log2 Fold Change Values between mutation and no.mutation
fold.change	Fold Change Values between mutation and no.mutation

Introduction

The aim of this project is to find links between certain gene expressions and familial Alzheimer's disease, using machine learning. To be more specific for the sake of data analysis: The mutation being observed is a presenilin 2 mutation, using patient-specific induced pluripotent stem cells (iPSC) to facilitate expression of the mutant type. Four different expression profiles were collected, using the Affymetrix Human Genome U133 Plus 2.0 Array. When looking at the names of columns, genes and differing values of expression, it's important to consider those are are all Affymetrix standards, which may need to be converted to further down the line. For example: Converting the gene IDs to ensembl IDs.

Initial Data and Variables

Let's first take a look at the provided .csv file, it's structure and first entries.

```
raw.df = read.csv("../data/GSE28379.csv")
head(raw.df, 5)
```

```
##      ID_REF GSM701542 GSM701543 GSM701544 GSM701545 no.mutation mutation
## 1 1007_s_at 615.52540 739.77800 720.90040 735.84750   677.65170 728.3740
## 2   1053_at 319.87120 654.39166 319.87140 319.87150   487.13143 319.8714
## 3    117_at  20.04304  32.15144  14.41752  24.94408    26.09724  19.6808
## 4    121_at 239.84415 171.02960 137.31161 176.75978   205.43687 157.0357
## 5 1255_g_at 155.14342 335.75186 177.99786 128.04279   245.44764 153.0203
##   log.2.fold.change fold.change
## 1         0.1041354   1.0748500
## 2        -0.6068188   0.6566430
## 3        -0.4071086   0.7541332
## 4        -0.3876026   0.7643988
## 5        -0.6816920   0.6234337
```

ID_Ref.

This column indicates the probe ID's, as sequenced by the Affymetrix Human Genome U133 Plus 2.0 Array. This is athe result of the sequencing technique. These are probe ID's, which don't represent a lot by themselves. They can, however, be used to find the ensembl ID's and gene symbols, which will be attempted below with Bioconductor:

```
genes <- select(hgu133plus2.db, c(raw.df[,1]), c("SYMBOL","ENTREZID", "GENENAME", "ONTOLOGY", "PATH"))
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
na.list <- genes[is.na(genes$SYMBOL),] # Storing the genes which were not detected for whatever reason.
colnames(genes)[1] <- "ID_REF" # Renaming  the column so the following merge works.
main <- merge(raw.df, genes, by=c("ID_REF")) # Merging the two dataframes together into a new one.
main <- main[!(main$ID_REF == "!series_matrix_table_end"),] # Removing an indicator row.
head(main[,c(1,10,11,12)], 5)
```

```
##      ID_REF SYMBOL ENTREZID                                    GENENAME
## 2 1007_s_at   DDR1      780 discoidin domain receptor tyrosine kinase 1
## 3 1007_s_at   DDR1      780 discoidin domain receptor tyrosine kinase 1
## 4 1007_s_at   DDR1      780 discoidin domain receptor tyrosine kinase 1
## 5   1053_at   RFC2     5982              replication factor C subunit 2
## 6   1053_at   RFC2     5982              replication factor C subunit 2
```

To summarize what has just been done: A bioconductor database was used to find the corresponding gene for every probe. It's important to note that for yet unknown reasons some probes were not recognized. The proper database was used, as can be seen by the database name.

GSM

The GSM columns indicate the different samples used in the paper this data is derived from. The values under these columns represent sequencing concentration, which is the result of a normalisation algorithm called MAS5.0. This algorithm is also developed by Affymetrix. These values are also not log transformed. A dedicated column was made for that.

The first two samples, GSM701542 and GSM701543 are iPSC sequences derived from Sparadic Parkinson's disease patients. The latter two, GSM701544 and GSM701545 are iPSC sequences from familial Alzheimer's disease (FAD) patients. The referenced paper aimed to compare the two conditions and their gene expressions.

No mutation & Mutation

The first column, no mutation, signifies the average of the first two non mutated parkinson's samples. The second column, mutation, shows the average of the two FAD mutant type samples.

Log2 Fold Change

These are subtracted 2log fold change values, showing which of the two averages are up regulated and down regulated. In case of a positive number, the mutation type samples are up regulated and the non-mutant types are down regulated. The reverse is true in case of a negative number.

Fold change

The ratio between the mutation and no mutation values. Mutation being divided by no mutation in this case.

Data variance & spread.

The original paper aimed to compare two groups and their expressions. By looking at the calculated Log2FC data, it'll be possible to see how much the two groups differ. Let's first look at the most significant differently expressing genes.

```
main <- main[order(main$log.2.fold.change),] # Reordering the DF by log2FC
head(main[,c(10,6,7,8)], 10) #Showing the 10 most significant down regulated genes
```

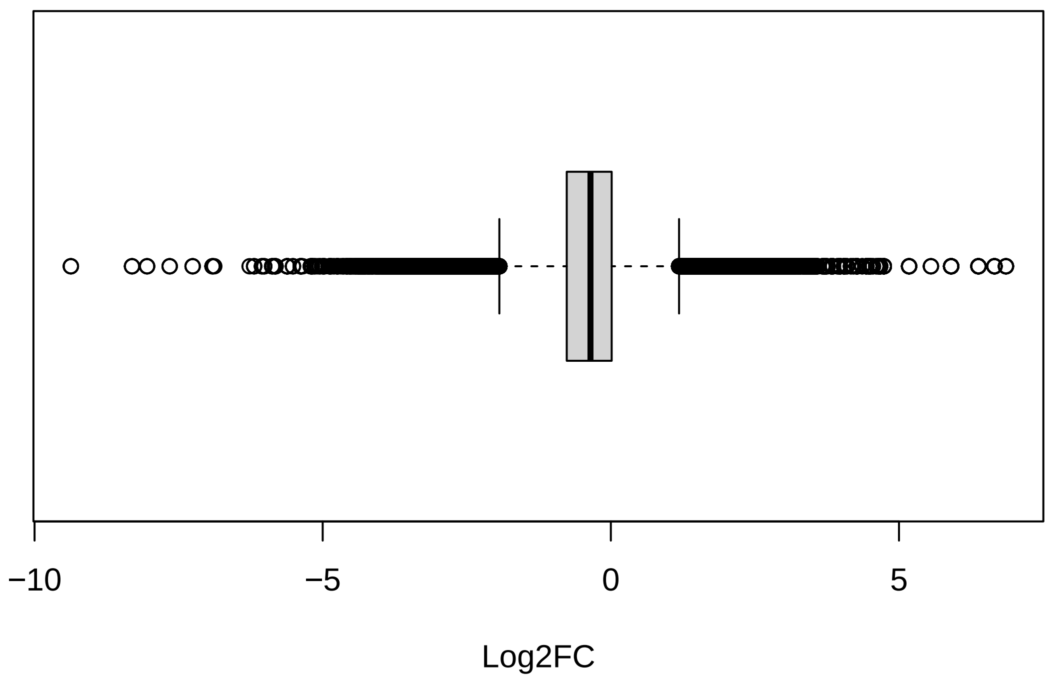
```
##       SYMBOL no.mutation  mutation log.2.fold.change
## 34889 RPS4Y1  3377.05241 5.1017696         -9.370551
## 34890 RPS4Y1  3377.05241 5.1017696         -9.370551
## 34891 RPS4Y1  3377.05241 5.1017696         -9.370551
## 54303  DDX3Y  1358.47492 4.2837637         -8.308893
## 54304  DDX3Y  1358.47492 4.2837637         -8.308893
## 54305  DDX3Y  1358.47492 4.2837637         -8.308893
## 50657 EIF1AY   217.13481 0.8207200         -8.047485
## 50658 EIF1AY   217.13481 0.8207200         -8.047485
## 50659 EIF1AY   217.13481 0.8207200         -8.047485
## 89983 ZNF257    45.97356 0.2280066         -7.655585
```

```
tail(main[,c(10,6,7,8)], 10) #Showing the 10 most significant up regulated genes
```

```
##       SYMBOL no.mutation  mutation log.2.fold.change
## 94100  CASP1   1.0538908 106.37357           6.65727
## 94101  CASP1   1.0538908 106.37357           6.65727
## 94102  CASP1   1.0538908 106.37357           6.65727
## 94103  CASP1   1.0538908 106.37357           6.65727
## 56957   BMP5   0.4355889  50.47331           6.85641
## 56958   BMP5   0.4355889  50.47331           6.85641
## 56959   BMP5   0.4355889  50.47331           6.85641
## 56960   BMP5   0.4355889  50.47331           6.85641
## 56961   BMP5   0.4355889  50.47331           6.85641
## 56962   BMP5   0.4355889  50.47331           6.85641
```

By taking a glance at the tables above, it seems that down regulation consists of more extreme values than up regulation. While this may be telling of how expression is affected by the mutation in general, it's not enough on its own to draw any conclusions yet. Let's further explore the log2fc values by creating a boxplot.

```
boxplot(main$log.2.fold.change, horizontal = TRUE, xlab = "Log2FC", title = "Boxplot of Log2FC")
```



As can be seen from the boxplot, there appears to be great variance in the log2FC values. This may be further examined by looking at a summary.

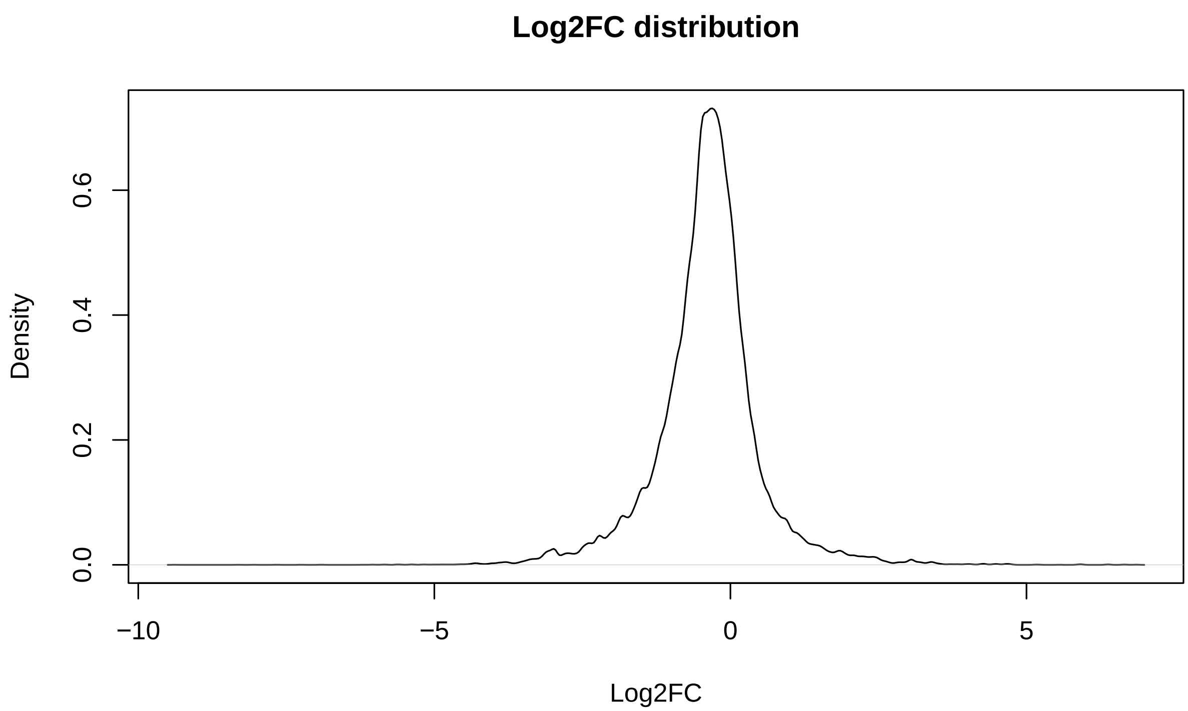
```
summary(main$log.2.fold.change)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -9.37055 -0.76497 -0.35343 -0.38615  0.01466  6.85641
```

Judging by the box plot and summary table, the genes seem to be mostly down regulated.

To further look at the distribution of the data, a density plot can also be used.

```
plot(density(main$log.2.fold.change), main = "Log2FC distribution", xlab = "Log2FC")
```



Judging by the box plots and density plot, the data seems to be roughly normally distributed. This is useful when looking at statistical significance in analyzing correlations. The lower and higher tail end genes will be stored for these ends.

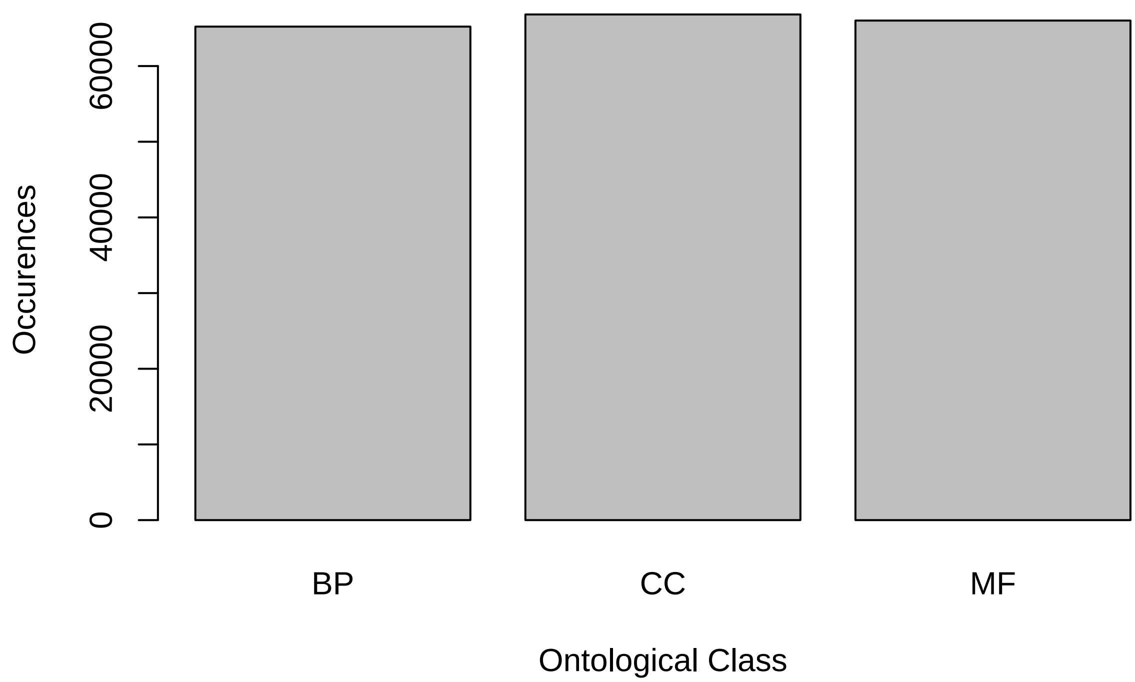
```
high <- main[pnorm(main$log.2.fold.change, mean = mean(main$log.2.fold.change),
                   sd = sd(main$log.2.fold.change)) >= 0.975,]
low <- main[pnorm(main$log.2.fold.change, mean = mean(main$log.2.fold.change),
                  sd = sd(main$log.2.fold.change)) <= 0.025,]
```

Correlations

Since the data in the file are all individual reads, there will not be any variables that affect each other, besides the obvious calculation of fold changes and means. That being said: There could be a connection between up/down regulation of certain genetic regions. Taking RPS4Y1, a gene encoding for a ribosomal sub unit, as an example: Could there be a trend in the mutation affecting down/up regulation of specifically ribosomal genes? Questions like these are not suited for EDA, but the addition of gene symbols, KEGG pathway, ontology and entrezID's columns in this document will help in answering such questions, which will eventually mean answering the research question.

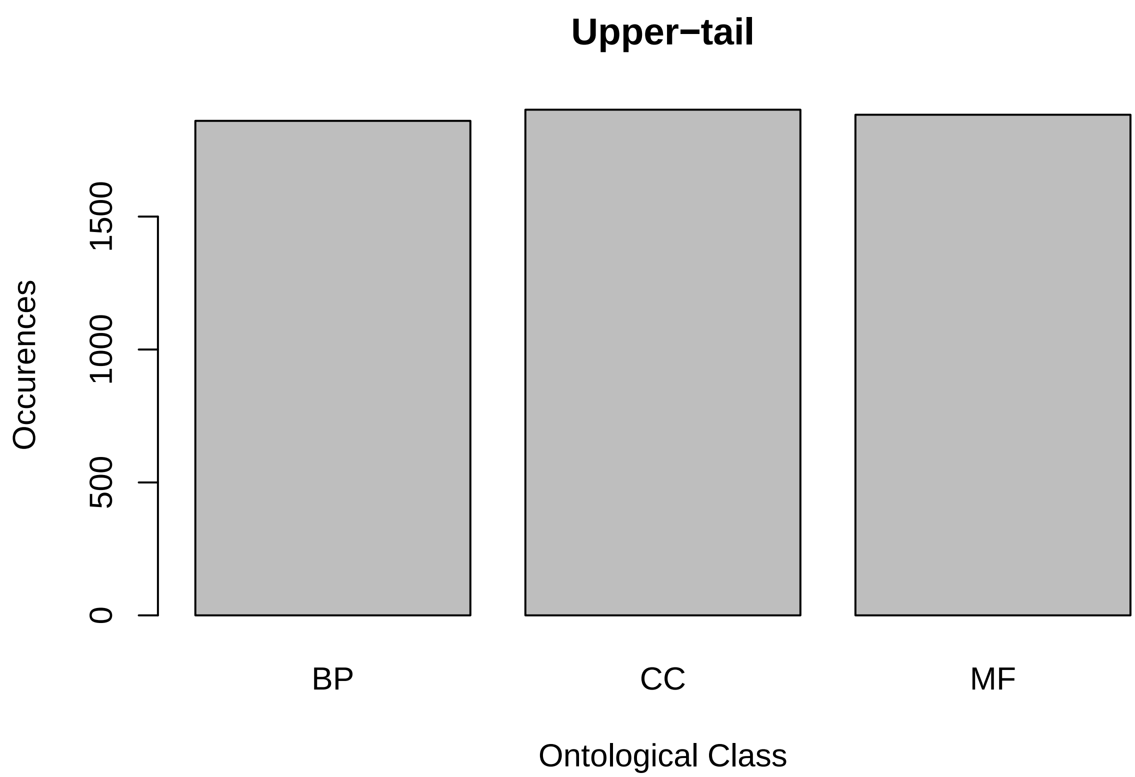
Plotting the different KEGG IDs and ontological classifications can however offer insight into which pathways and classes are most prevalent.

```
barplot(table(main$ONTOLOGY), xlab = "Ontological Class", ylab = "Occurences")
```

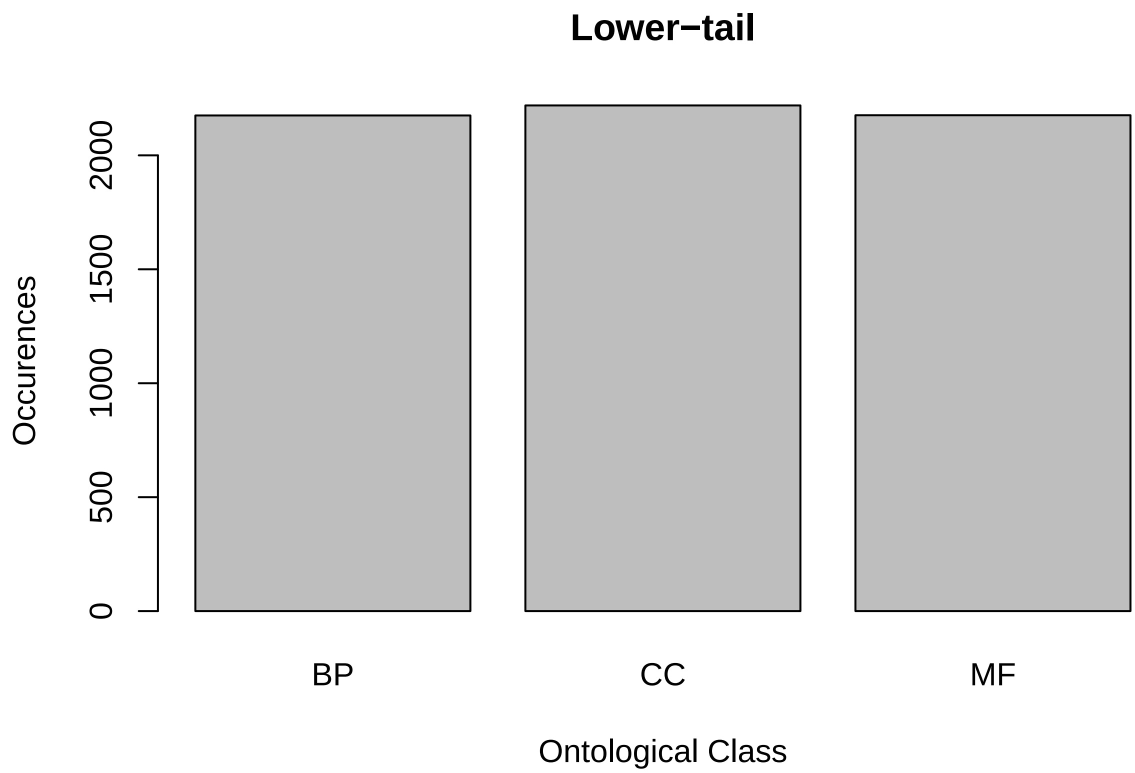


The differing classes appear to be close in the amount of occurrences. Looking at the values in the lower and upper 2.5% may net different results.

```r
barplot(table(high$ONTOLOGY), xlab = "Ontological Class", ylab = "Occurences", main ="Upper-tail")
```




```r
barplot(table(low$ONTOLOGY), xlab = "Ontological Class", ylab = "Occurences", main ="Lower-tail")
```



The bar plots show that the lower-end and upper-end tails are also equally divided.

It may be possible to find distinctions between different classes by looking at the Log2FC values for every different class.

```
BP <- subset(main, ONTOLOGY == "BP")
CC <- subset(main, ONTOLOGY == "CC")
MF <- subset(main, ONTOLOGY == "MF")

summary(BP$log.2.fold.change)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -9.3706 -0.7440 -0.3463 -0.3775  0.0141  6.8564
```

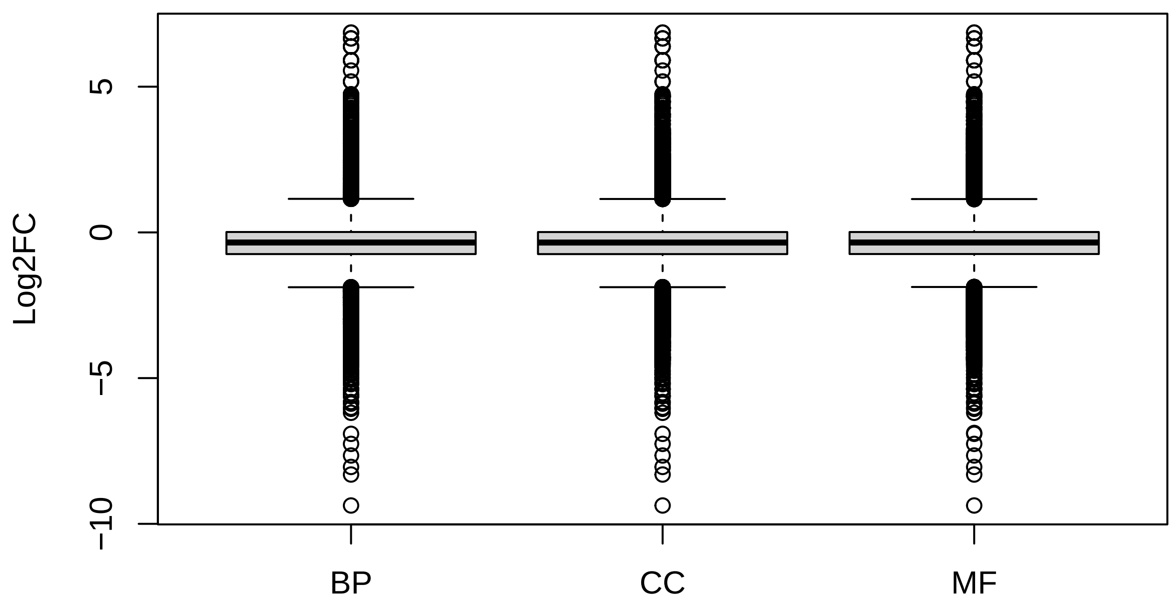
```
summary(CC$log.2.fold.change)
```

```
##     Min.   1st Qu.    Median      Mean   3rd Qu.     Max.
## -9.37055 -0.74445 -0.34809 -0.37869  0.01193  6.85641
```

```
summary(MF$log.2.fold.change)
```

```
##     Min.   1st Qu.    Median      Mean   3rd Qu.     Max.
## -9.37055 -0.74151 -0.34626 -0.37680  0.01279  6.85641
```

```
boxplot(BP$log.2.fold.change, CC$log.2.fold.change, MF$log.2.fold.change,
        names = c("BP", "CC", "MF"), ylab = "Log2FC")
```



Both the summaries and box plots are very similar for every classification. At first glance this indicates that the classes are not a detriment in the fold change values of genes.

To confirm this suspicion, t-test may be performed. Since multiple groups are being observed, an ANOVA test may be used.

```
summary(aov(main$log.2.fold.change~main$ONTOLOGY))
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## main$ONTOLOGY      2      0  0.0602   0.082  0.922
## Residuals     198036 146114  0.7378
## 16126 observations deleted due to missingness
```

As can be seen from the results, the P-Value indicates that there's no significant difference in Log2FC means of ontological classes.

The same may be done for the pathways. The pathway column does have significantly more unique occurrences than the ontological classifications. The ones with the most significant log2FC values will therefore be shown and considered.

```
occur_path_high <- table(high$PATH)
occur_path_high <- occur_path_high[order(occur_path_high, decreasing = TRUE)]
occur_path_high[1:10] # list is around 160 diferent pathway entries long.
```

```
##
## 01100 04060 05200 04080 04510 04020 04810 04062 04512 04610
##   168   120   102    87    81    60    60    54    54    54
```

```
#Showing the 10 most prevelant ones at the higher end.

occur_path_low <- table(low$PATH)
occur_path_low <- occur_path_low[order(occur_path_low, decreasing = TRUE)]
occur_path_low[1:10] # list is around 160 diferent pathway entries long.
```

```
##
## 01100 04080 05200 04010 04020 04060 04650 00980 00982 04630
##   209   105    93    78    78    60    57    56    54    51
```

```
#Showing the 10 most prevelant ones at the lower end.
```

Summary

The initial data offers insight into the differing expression values of genes between a mutant and non-mutant group. By examining the variables and columns provided, multiple facets of the data was analysed. First, there seems to be a great spread in the data when looking at log2FC values. The data is also normally distributed. To find correlations between the variables, notably log2FC and different identifiers, the most significant DEGs were extracted. Analysis of these genes and their complimentary KEGG pathways and ontological identifiers was performed. No significant correlations were found by looking at those two variables and examining the corresponding log2FC values.