Exploratory Data Analysis

Orfeas Gkourlias

2022-09-23

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"))
```

```
## '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   1053_at   RFC2     5982                 replication factor C subunit 2
## 4    117_at  HSPA6     3310 heat shock protein family A (Hsp70) member 6
## 5    121_at   PAX8     7849                                  paired box 8
## 6 1255_g_at GUCA1A     2978                 guanylate cyclase activator 1A
```

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 dis ease (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 varaince

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
```

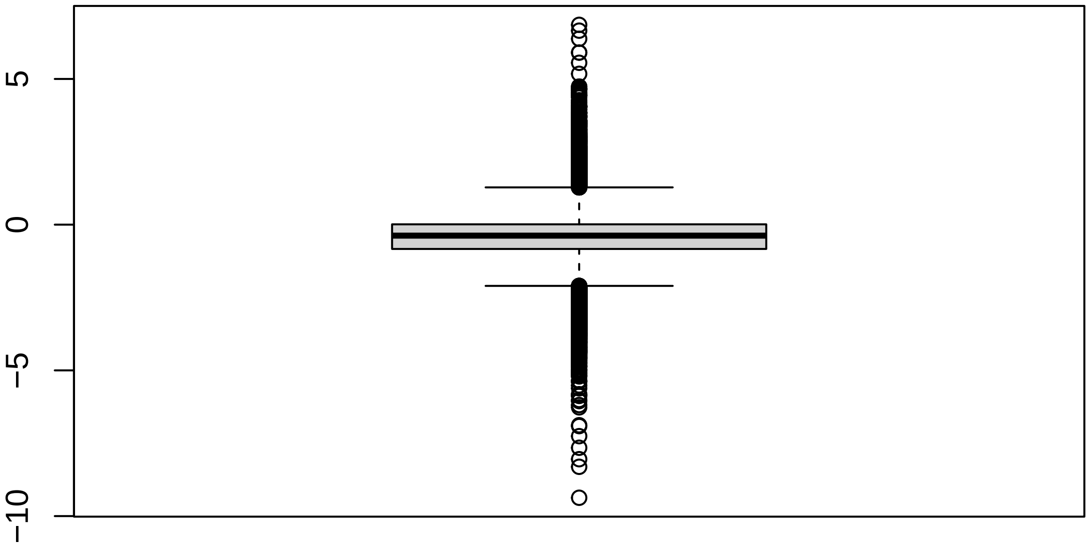
```
##       SYMBOL no.mutation   mutation log.2.fold.change
## 11737 RPS4Y1  3377.05241  5.1017696         -9.370551
## 14897  DDX3Y  1358.47492  4.2837637         -8.308893
## 14288 EIF1AY   217.13481  0.8207200         -8.047485
## 21020 ZNF257    45.97356  0.2280066         -7.655585
## 14287 EIF1AY   770.43858  5.0370598         -7.256954
## 40891    HRK   326.14995  2.7076090         -6.912372
## 44064 TXLNGY   123.20178  1.0447735         -6.881689
## 51508   <NA>    80.66129  1.0460687         -6.268827
## 14898  DDX3Y   741.18944 10.1538220         -6.189748
## 39797  USP9Y   292.10337  4.3945396         -6.054623
```

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

```
##          SYMBOL no.mutation   mutation log.2.fold.change
## 21331       HGF   0.7918953   20.19333          4.672425
## 332      CARD16   9.8492504  262.60704          4.736748
## 333       CASP1   9.8492504  262.60704          4.736748
## 21719     CASP1   1.1523522   41.64598          5.175524
## 20067      CD69   0.5375389   25.26361          5.554548
## 37774     ITGB6   1.2899955   77.32014          5.905406
## 37775 LINC02478   1.2899955   77.32014          5.905406
## 14353      MMP1  17.0247395 1416.90241          6.378964
## 21720     CASP1   1.0538908  106.37357          6.657270
## 15347      BMP5   0.4355889   50.47331          6.856410
```

By taking a glance at the tables above, it seems that down regulation consits 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 it's own to draw any conclusions yet. Let's further explore the log2fc values by creating a boxplot.

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


5
0
−5
−10