LabBook_16_09_2016

Claire Green

Wednesday & Thursday

I'm trying to establish the expression data for each of the 178 DEGs but it was more complicated than I thought. Because I used both ensembl IDs and HGNC symbols, I have to run the script twice merging with each identifier. I then have to double check that all the genes have the right expression data and no duplicates. What I noticed was that one of the genes - HMOX2 has two alleles each with different ensembl IDs. I'm not sure whether to count them both or not, so I will have to ask. I will leave them in for now.

SCRIPT FOR ENSEMBL ID-BASED MERGE

```
#Selecting DEGS from expression matrix

#Load list of interesting genes
#setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Data/GeneExpressionAnalysis/Microarray/")
setwd(dir = "/Users/clairegreen/Desktop/")
Genelist <- read.csv("overlap_ens2hgnc.csv", header = TRUE)

#load dataset
setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Code/Results/GeneExpression/DEG_Test2/")
exprs <- read.csv("C9rankeduniqueresult.csv")

#Make gene symbol row names
#rownames(exprs) <- exprs$Ensembl
exprspat <- exprs[,52:59]
exprspat[,(length(exprspat)+1)] <- exprs$Ensembl

#Make gene symbol a column
# exprspat <- cbind(exprspat, exprs$Gene.Symbol)
# colnames(exprspat)[length(exprspat)] <- "Gene.Symbol"

#Merge by interesting gene names with expression to form matrix
patgene <- merge(Genelist, exprspat, by.x = "ensembl_gene_id", by.y = "V9")
#patgene <- patgene[!duplicated(patgene[,11]),]
# rownames(patgene) <- patgene$V1
# patgene[,1] <- NULL

setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Code/Results/GeneExpression/Pathways_to_TDP
write.csv(patgene, file = "C9m_DEG_Ens_Exprs.csv")
```

SCRIPT FOR HGNC ID-BASED MERGE

```
#Selecting DEGS from expression matrix

#Load list of interesting genes
```

```
#setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Data/GeneExpressionAnalysis/Microarray/")
setwd(dir = "/Users/clairegreen/Desktop/")
Genelist <- read.csv("overlap_ens2hgnc.csv", header = TRUE)

#load dataset
setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Code/Results/GeneExpression/DEG_Test2/")
exprs <- read.csv("C9rankeduniqueresult.csv")

#Make gene symbol row names
#rownames(exprs) <- exprs$Ensembl
exprspat <- exprs[,52:59]
exprspat[,(length(exprspat)+1)] <- exprs$Gene.Symbol

#Make gene symbol a column
# exprspat <- cbind(exprspat, exprs$Gene.Symbol)
# colnames(exprspat)[length(exprspat)] <- "Gene.Symbol"

#Merge by interesting gene names with expression to form matrix
patgene <- merge(Genelist, exprspat, by.x = "hgnc_symbol", by.y = "V9")
#patgene <- patgene[!duplicated(patgene[,11]),]
# rownames(patgene) <- patgene$V1
# patgene[,1] <- NULL

setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Code/Results/GeneExpression/Pathways_to_TDP
write.csv(patgene, file = "C9m_DEG_Ens_hgnc.csv")
```

Then I had to, by hand, copy and paste the corresponding expression values for each of the 178 DEGs (or 177 as I decided to remove the HMOX2 that didn't have a corresponding ensembl ID in the RNA-seq dataset)

Next, I took the curated expression matrices and ran cor.test on each. I used Spearman because it isn't affected by the range of variables, which is important as the range in microarray is much less than in RNA-seq. It also means that outliers don't screw up the correlation so much.

The CHMP2B data came out super weird. Because there are only three values I don't think it has the power to be able to calculate a reliable Spearman's Rho. Statistically I should remove it as it's not a clean calculation but at the same time it's removing a genetic background that is important. I will continue analysis but I am sceptical that it will produce anything interesting.

Once I had a Spearman's Rho value for each gene pair, I had to rank it using the following script

```
setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Code/Results/GeneExpression/Pathways_to_TDP
C9mR <- read.csv("C9m_CorResults.csv")

rankC9 <- C9mR[order(abs(C9mR[,2]), decreasing = TRUE),]
rankC9$rank <- seq.int(nrow(rankC9))
```

This is okay however I come across two problems - first of all, the cor.test function produces a lot of values that are the same. As the readout only takes into account up to 7 decimal places, I can't tell if it's just something to do with the powering making it the same or whether it's actually different numbers but rounding causes them to be similar at 7 decimal places. Also, some gene pairs have a value of 1. That's weird. I don't think that should really happen.

Also, I can't work out how to rank the values in a way that means pairs with the same number are given the same rank, and then it skips the right amount of ranks in the next different number. Usually that stuff

is easy to google but things like Stack Exchange are only giving methods on how to rank a vector - which doesn't work for me as I need to keep the gene pair names.

I realised that I didn't need to rank it myself as spearman rank does it for you - duh!

What I did find out was that the RNA seq data set was two genes short - therefore around 400 pairs short. This turned out to be for two reasons - one gene had been susceptable to excel's ability to turn gene names into dates. So SEPT11 had become sept-11 and wasn't being matched. The second problem was different gene names for the same gene. In the microarray datasets, the gene was referred to as AHCTF1 whereas in the RNA seq data it was referred to as GPR116. This meant I had to change the original DEG set to swap in this name and then once I had matched up the expression data I could then change the name to AHCTF1 afterwards so the correlation would match up.

When I came to conduct spearman's rank it failed somewhat as it produced the error "Cannot compute exact p-value with ties". After googling this problem, others suggested using Kendall's Tau as it can take ties into account.

I used Kendall's Tau, and then visualised the results as a network in Cytoscape. I used the Edge Weighted Spring-Embedded layout weighted to pvalue to show the clustering of data. As you can see, they did cluster into two groups but this may just be a tissue type effect.

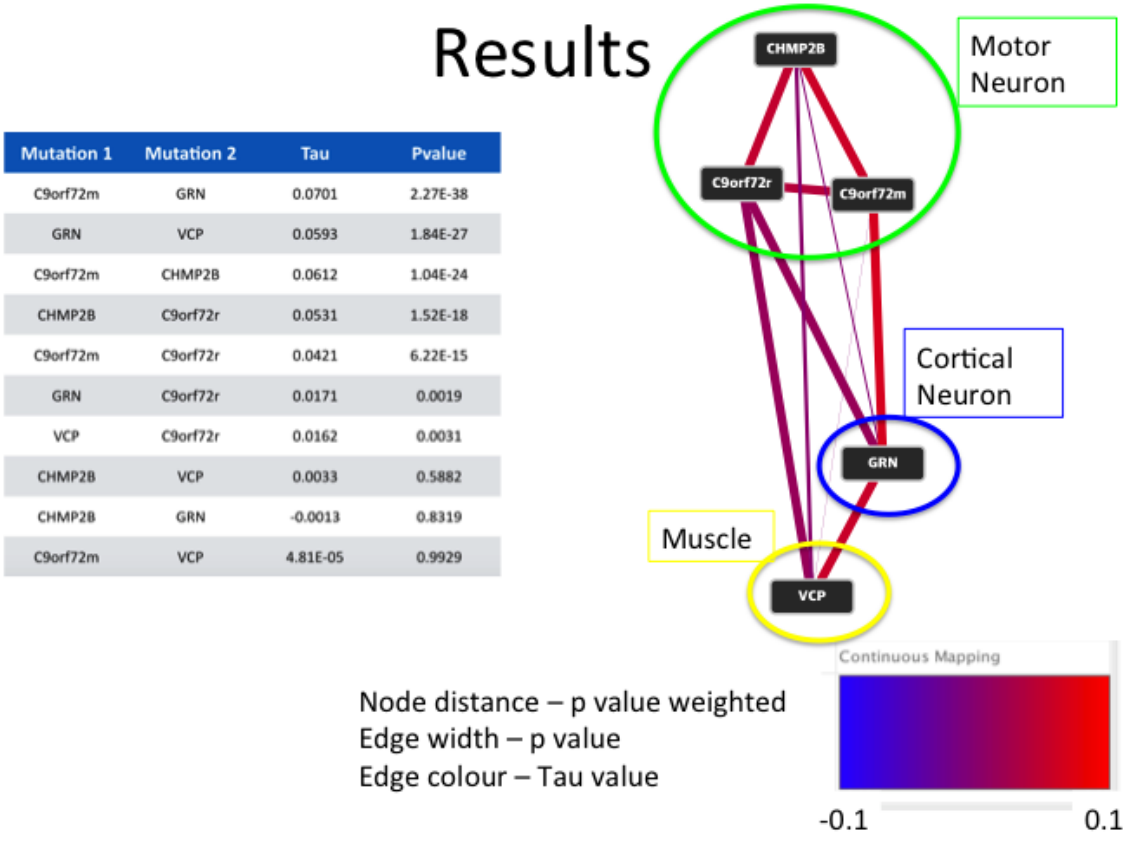

Figure 1: