LabBook_04_03_16

Claire Green

Monday

On monday I continued to look at the RNA seq data. Unfortunately I lost the first version of the script I wrote. I have the results, but had to rewrite the code and now I'm getting completely different results (unsurprisingly)

This is what I have so far:

```
##RNA-Seq Gene Expression Analysis using Limma##

analysis.name<-"RAV1MAR" #Label analysis
setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Data/GeneExpressionAnalysis/RNA-seq/Ravits/
# Counts <- read.table(file = 'GSE67196_Petrucelli2015_ALS_genes.rawcount.txt', header = TRUE)
#
# write.csv(x = Counts, file = "counts_petrucelli.csv")

Counts <- read.csv(file = "GSE76220_ALS_LCM_RPKM.csv", header = TRUE)
# Counts[,1] <- NULL #do twice
# Counts[,1] <- NULL #do twice

Counts[Counts == 0] <- NA
# Counts[Counts<30] <- NA
Counts <- na.omit(Counts)
# Counts<-subset(Counts, subset=(GeneID !="NA")) #if no gene symbol, discount

# Countszero <-subset(Counts, subset=(row !=0))
# Countszero <- apply(Counts, 1, function(row) all(row !="NA"))
# Counts <- Counts[Countszero,]

library(limma)
library(edgeR)

# Counts <- Counts[!duplicated(Counts[,1]),]
# rownames(Counts)<-Counts[,1]

Counts[,1] <- NULL
Counts <- Counts[,1:21]
# Counts <- data.matrix(Counts)


# dge <- DGEList(counts=Counts)
# dge <- calcNormFactors(dge)

Treat<-factor(rep(c("Control", "Patient"),c(9,12)), levels=c("Control", "Patient"))
design<-model.matrix(~Treat)
rownames(design)<-colnames(Counts)
design
```

```
# v <- voom(dge,design,plot=TRUE)

fit <- lmFit(Counts,design)
fit <- eBayes(fit)
result<-topTable(fit, coef="TreatPatient", adjust="BH", number=nrow(Counts)) #"BH" adjust for multiple
genesort <- result[order(result$adj.P.Val),]
genesort[,7]<-rownames(genesort)


setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Data/GeneExpressionAnalysis/RNA-seq/16_3_1b/
write.csv(genesort, file=paste(analysis.name, "rankeduniqueresult.csv", sep=""), sep="\t", row.names=TRU

topgene <- genesort[1:1000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_1000", sep = ""))
topgene <- genesort[1:2000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_2000", sep = ""))
topgene <- genesort[1:3000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_3000", sep = ""))
topgene <- genesort[1:4000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_4000", sep = ""))
topgene <- genesort[1:5000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_5000", sep = ""))
```

I think the problem might be that I am getting rid of duplicates too early. It's difficult because if I do not get rid of them then I cannot make the gene IDs the row names (as assigning row names does not allow duplicate names). I will have to work out how I managed to do it the last time. It involved taking out the gene names and merging by row number.

Tuesday

This was my second attempt at re-creating my initial analysis. For this I had to do as I said above, and remove the gene symbols for the analysis but re-merge by rowname at the end, and *then* remove the duplicates.

```
##RNA-Seq Gene Expression Analysis using Limma##

analysis.name<-"PET" #Label analysis
setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Data/GeneExpressionAnalysis/RNA-seq/Petrucel
# Counts <- read.table(file = 'GSE67196_Petrucelli2015_ALS_genes.rawcount.txt', header = TRUE)
#
# write.csv(x = Counts, file = "counts_petrucelli.csv")

Counts <- read.csv(file = "FCX_Petrucelli.csv", header = TRUE)

Counts[Counts == 0] <- NA
# Counts[Counts<30] <- NA
Counts <- na.omit(Counts)
# Counts<-subset(Counts, subset=(GeneID !="NA")) #if no gene symbol, discount

# Countszero <-subset(Counts, subset=(row !=0))
# Countszero <- apply(Counts, 1, function(row) all(row !="NA"))
# Counts <- Counts[Countszero,]
```

```
library(limma)
library(edgeR)

# rownames(Counts)<-Counts[,1]

# Counts[,1] <- NULL
Countnum <- Counts[,2:28]
# Counts <- data.matrix(Counts)

#DGElist
dge <- DGEList(counts=Countnum)
dge <- calcNormFactors(dge)

#Design
Treat<-factor(rep(c("Control", "Patient"),c(9,18)), levels=c("Control", "Patient"))
design<-model.matrix(~Treat)
rownames(design)<-colnames(Countnum)
design

#Voom transformation
v <- voom(dge,design,plot=TRUE)

#Limma fitting
fit <- lmFit(v,design)
fit <- eBayes(fit)
result<-topTable(fit, coef="TreatPatient", adjust="BH", number=nrow(Countnum)) #"BH" adjust for multipl
result <- merge(result, Counts, by="row.names", all=TRUE)
result <- result[,2:8]

uniqueresult <- result[!duplicated(result[,7]),]
genesort <- uniqueresult[order(uniqueresult$adj.P.Val),]

setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Data/GeneExpressionAnalysis/RNA-seq/16_3_2a
write.csv(genesort, file=paste(analysis.name, "rankeduniqueresult.csv", sep=""), sep="\t", row.names=TR

topgene <- genesort[1:1000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_1000", sep = ""))
topgene <- genesort[1:2000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_2000", sep = ""))
topgene <- genesort[1:3000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_3000", sep = ""))
topgene <- genesort[1:4000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_4000", sep = ""))
topgene <- genesort[1:5000,]
write.csv(x = topgene, file = paste(analysis.name,"_ap_5000", sep = ""))
```

I compared this list with the previous list, and there is about 25-30% overlap between the two, which isn't great. Here are the Top ~200 consensus genes from each list:

First Attempt	Second Attempt
S100B	MEST
TNFAIP8	SULT1A1
SDC2	ZNF551

First Attempt	Second Attempt
ZFHX4	COL4A2
HLA-DOA	ZNF219
IL11RA	FMOD
TNFAIP1	MYH9
TIPARP	JAG1
C15orf27	FSTL1
ITGB8	ECEL1
HLA-DMA	RXRB
ISM1	ATG16L2
C6orf192	RNF157
DYNLRB2	ADAMTS1
HNRNPF	CCDC92
BIRC3	COL14A1
EFHD1	COL4A1
DUSP1	HIST2H2BF
TSPAN9	IQGAP2
EPS8L2	ITGA6
FOXF1	RRP1
MUS81	ZNF808
NVL	GRIN2B
TNFRSF10D	ABCC9
HSPA2	SYT17
GRN	CHMP2A
KCNA3	PLAGL1
NENF	HSP90B1
C3	LMBR1L
PRSS16	MTMR4
TCFL5	NUDT14
C10orf128	PEBP1
ZNF219	PLA2G4A
C3orf33	SCD5
CPSF4	TNC
CXorf57	XPA
EDN1	ZNF471
ELFN1	CD44
FAM46C	CRY1
FLJ33630	FBLN2
FMO2	ARHGAP10
ASB6	NID1
NLGN1	CCDC85B
CHCHD6	MYL6B
B3GAT1	NR2F2
C4orf29	PPP2CB
TCERG1L	C7
GJB5	INTS3
IRGQ	UBL5
RPS9	EIF2C3
CRYAB	AOC3
AIRE	ITPK1
FAM84B	RAB27A
MRPL50	SPARC
PMVK	OSMR

First Attempt	Second Attempt
FAM82A2	MPZL2
PLEKHB1	PROS1
TNFRSF1A	TRIM25
C1orf88	C22orf32
GLI3	AEBP1
APOL1	EIF4E1B
ATP5EP2	NAMPT
CH25H	CKB
TCEAL8	ZDHHC18
BMP7	IL1R1
PLA2G7	TNFAIP3
COL3A1	LHX4
RNF149	EPB41
RHOB	SEMA4F
ANKRD40	C1orf61
PODN	PLXND1
NEIL1	DNASE2B
FOXF2	CCDC80
ZC3H6	SLC39A14
NAMPT	HNRNPAB
ACAA2	SIK1
CD58	THBD
TMEM55B	TTC32
MSRA	H1F0
GPRC5C	MRPL16
BBX	JPH2
CKB	CASC2
SV2A	KCTD12
MED19	ERN1
AKIRIN2	NHLRC3
GSTO2	CRYBA2
TNFRSF18	DMWD
ZNF570	ANKRD44
LIPE	PEX16
MTUS1	TNFRSF1A
ALDH1A1	TOMM7
CETN3	CA8
CCDC24	FAT4
ZNF440	NBAS
C1orf61	PRKG1
TPT1	RCN1
RBM17	ADAMTS9
IGFBP7	CDK5
KLF2	ELFN1
ANO3	GPR108
TINAGL1	KLF6
TNC	MDP1
FN1	PCDH18
EXTL1	GNPTG
GPR146	LAMC1
TPM2	GABPB2
TIMP2	MXI1

First Attempt	Second Attempt
NCF2	ADCY3
TNFRSF1B	AQP1
ZNF471	PDZD11
ANKRD10	OGFRL1
HIST2H2BF	CNN3
ID3	ZNF439
ARHGAP29	RPL38
MEGF10	TCEAL3
BBS1	DUSP1
TCF12	MRTO4
KIF1C	DLG5
RASGRP3	ZFPL1
PAPSS2	ADPRH
DAAM1	ANGPT1
ZFP36L2	GLIS3
UACA	PDPN
PDK3	SLC20A2
TJP2	EPHA2
FAM104A	ANKS3
ARMC10	MED29
GNAI3	ACBD7
KDELC1	ZFP82
GPR124	CCDC24
THBS2	FGD5
NACA2	KLHL18
TNFAIP3	ARHGAP29
CTDSP1	RPL37A
TMTC4	C1S
RFTN2	RPL36AL
RGS1	FAU
CXCL1	FLJ33630
ASXL3	ANXA1
EIF4E2	C9orf69
SYT17	FTH1
MTMR4	RAD9B
TOB1	NFKBIZ
ZIC3	PHLDB2
C2	SNHG10
CDC42EP2	RPS29
AIF1	SLC2A1
FUT1	SPCS1
LASP1	AFAP1L1
CP	FOS
TMEM109	EMP1
ZNF230	PAK7
RPN1	OSBPL11
BMPR1B	SF3B14
CBX5	TCFL5
CSRP2	DGAT2
CTSL1	MYH7B
LIX1	STK32A
NFATC4	

First Attempt	Second Attempt
NME4	
RGS5	
MTMR10	
PLEKHA7	
FOXN3	
HLA-DRA	
TWSG1	
ZNF439	
CRABP1	
GLUL	
NID1	
FBXO32	
DAAM2	
TRMT61B	
H1F0	
LPIN3	
ZNF799	
AFAP1L1	
ITGB5	
PIK3AP1	
TMPRSS5	
C1orf162	
EHD4	
HLA-DPA1	
LIMD1	
MYH7B	
S1PR5	
ME3	
TP53INP2	
ZNF528	
CCDC92	
ENPP2	
ARHGAP24	
RSPO2	
IL13RA1	
SYF2	
C11orf30	
DSN1	
ARPC1B	
NEGR1	
ARF4	

I then decided to compared the overlap of these lists with the list generated from the microarray analysis. As it turned out, there was very similar consensus for both lists. Subsequently, I decided that as there was no way of me regenerating the first list, I would carry on with the second. I also spent more time perfecting the method for the second list so I am more confident of its process.

Below is a table showing the consensus genes at different threshold combinations. As you increase the microarray threshold, new genes are coloured blue. As you increase the RNA seq threshold, new genes are in red. New genes to both are in purple. Black shows genes previously identified in both, and thus are genes that appear as consensus soonest.

		RNA Seq				
		1000	2000	3000	4000	5000
Microarray	1000	0	0	0	0	0
	2000	0	0	0	0	0
	3000	0	0	TARDBP	PFDN1 TARDBP TARS	PFDN1 TARDBP TARS
	4000	0	0	TARDBP DCN TCF4	PFDN1 TARDBP TARS DCN TCF4 ETS2	PFDN1 TARDBP TARS DCN TCF4 ETS2 RPLP2 CST3
	5000	JAG1	JAG1 SPARC KCTD12 ANXA1	COX6A1 JAG1 SERBP1 SYNM SPARC KCTD12 HBB ANXA1 TARDBP BGN RAB40B DCN TCF4 GBAS PLEKHB1 ACTN1	COX6A1 JAG1 PFDN1 SERBP1 SYNM SPARC KCTD12 HBB ANXA1 TARDBP TARS BGN RAB40B DCN TCF4 GBAS PLEKHB1 ACTN1 ETS2	COX6A1 JAG1 PFDN1 SERBP1 SYNM SPARC KCTD12 HBB ANXA1 TARDBP TARS BGN RAB40B DCN TCF4 GBAS PLEKHB1 ACTN1 ETS2 RGS2 RPLP2 CST3

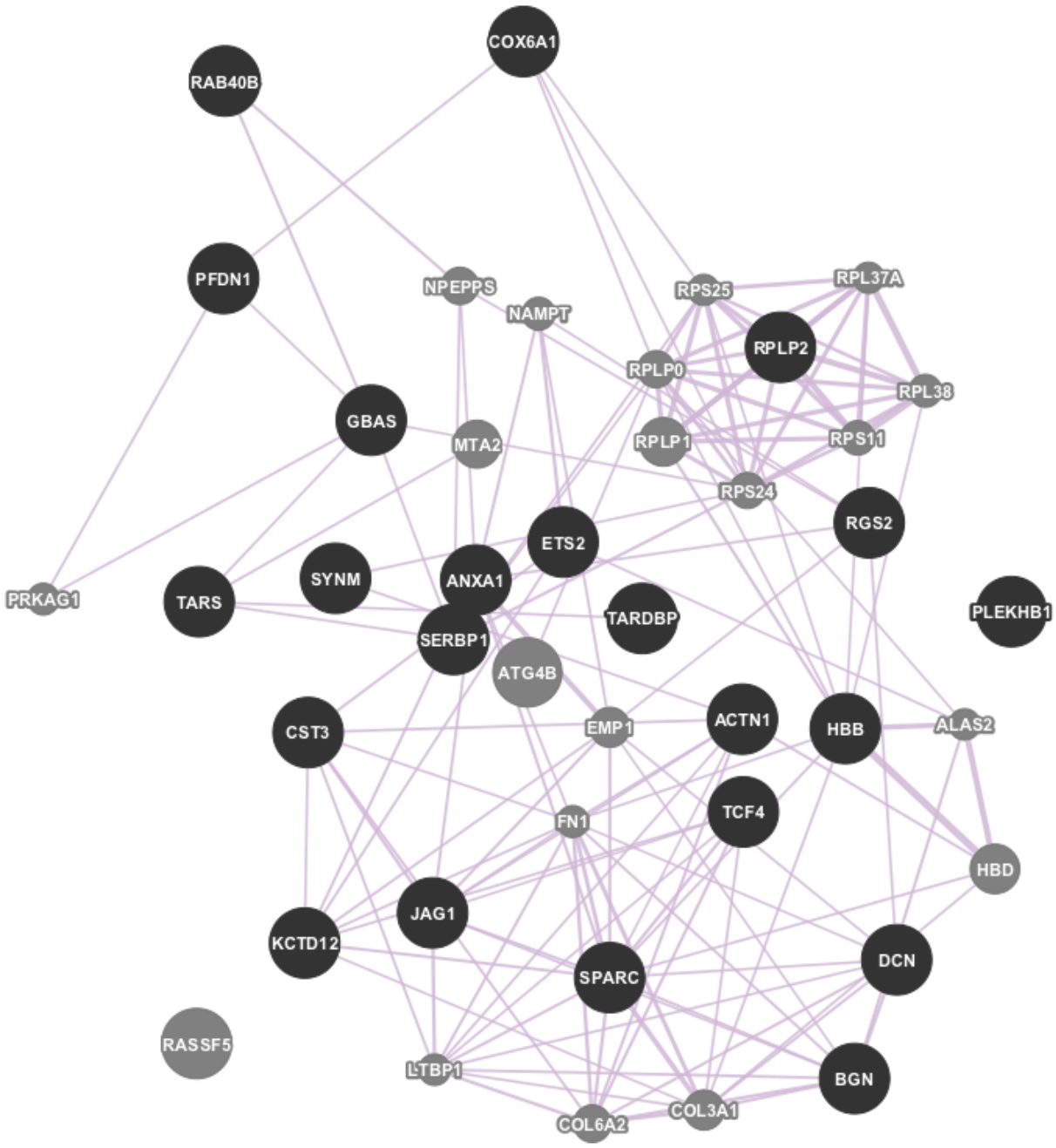
The first thing to note is that TARDBP is in this list.

COX6A1 is a Charcot-Marie-Tooth associated gene. JAG1 is mostly a cancer gene but is loosely associated with nervous system diseases such as MS SYNM is also associated with nervous system diseases, but CTTV states this is mostly through text mining SPARC has been associated with MS, ALS and muscular dystrophy by RNA expression ANXA1 RNA exp for ALS and MS BGN is associated to PD by text mining RAB40B to ALS by RNA exp DCN - ALS by text mining TCF4 PD by RNA exp ACTN1 MS by RNA exp PFDN1 - PD by text mining ETS2 - Alzheimer's by text mining RGS2 - MS & PD by RNA exp RPLP2 - MS by RNA exp CST3 - MS by RNAexp, ALS, Alzheimer's, Creutzfeldt Jacob by text mining

Before I get too far with analysis, I need to perform random permutation tests to ensure my numbers of genes are higher than expected by chance.

.	Top 1000	Top 2000	Top 3000	Top 4000	Top 5000
Number	39	158	382	646	951
AverageRan	33	131	295	525	821
pValue	0.12	0.0062	<.001	<.001	<.001

Top 1000 is not significant however top 2000 and upwards are very significant. The middle row shows the average random consensus as a comparison. Running the genes through GeneMANIA shows that they are somewhat tightly co-expressed, and this co-expression is increased with the addition of 20 other genes.



A problem I feel I have is that the two RNA data sets were not quite analysed the same way as the two groups provided slightly different pre-processed information. The Petrucelli data came as a count matrix however the Ravits data was a matrix of expression values (like in microarrays). This meant that I'm not entirely sure what their criterion were for analysis. I believe that I should generate the count matrix and remove all values below a certain threshold. At the moment I have just removed zero values but since counts can go into the thousands, taking values below 50 seems suspect.I will need to ask what others recommend.

This problem can only be answered by downloading the FastQ files for both and doing the analysis myself. This requires getting familiar with using a bash shell to download the files.

Wednesday/Thursday

It took me a little while to get used to bash scripting and setting up iceberg. This is what I have done so far:

1) To log into iceberg, use the command ssh -X uniusername@iceberg.sheffield.ac.uk
2) Once in iceberg, you have to set up a node you want to use. You have to specify using the following command:

```
**qrsh -P hidelab -l rmem=32G/64G -l mem=32/64G -binding linear:16**
```

So that I didn't have to type this every time, I wrote a bash script to do it for me. To write a bash script you have to open nano and write

```
**#!/bin/bash

  do something
```

and save with the filename ending in .sh

After this, you run the chmod command to set permissions

```
chmod -x filename.sh
```

To download files from a url, you first must create an array of the urls. If the files are all in one place you only need to say the one, but in some cases the urls are in multiple folders. If this is true, the urls must be put into an array. You can do this by saying array=(url1 url2 url3...urlN). Once this is done, you can download them as follows

```
For i in "${array[@]}"
do
wget $i
done
```

The @ sign says to go through each value in the array and conduct wget.

After this, the files should start downloading.

Friday

On Friday we had a visit from Sally John, the VP of Genomics and Computational Biology at Biogen Idec. She was coming to see how we were getting on and how to manage our projects for the next year. Since I am not technically funded by Biogen, it wasn't so much for me, but it was an interesting experience. What I did take from it is that Biogen have some RNA-seq data from a failed study they did on a drug called Dexpramipexole, potentially around 950 samples. Will have to wait and see, as you never know how long it takes for this stuff to happen.