LabBook_13_05_2016

Claire Green

Monday

On Monday I spent most of the day sorting out stuff for seattle, and also preparing the presentation for the meeting with the GTC developer group

Tuesday

In the morning I looked at performing enrichment on my genes, and also on an expanded list from Genemania. Unfortunately there was no enrichment for ALS and AD genes or John's subnetwork 28, no matter if the list was expanded or not.

In the afternoon I called in on the developer's meeting and gave my presentation. Everyone was very welcoming and I think they took my suggestions into consideration. Afterwards, I contacted John about Win and my conversation about using mutation frequency as a measure of how important a gene is in disease. John and I ended up having a long discussion and he suggested I do the following

Download the ExAC data from the Broad Institute. This data measures the constraint of a gene i.e. how resistant it is to mutation. The theory is that if the gene is less likely to mutate, it means it is more important biologically as life would not be viable without it.

First of all, I looked for enrichment of all my genes in the total list. The list is approximately 18,000 genes. Using a fisher's excact test, 42 of my 45 genes are present:

```
setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Data/GeneExpressionAnalysis")

G <- read.table(file = "allgenes.txt")
g <- G$V1

setwd("/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Code/Results/GeneExpression")
#Load ExAC Data
Exac.All <- read.table(file = "fordist_cleaned_exac_r03_march16_z_pli_rec_null_data.txt", header = TRUE
exacgenes <- Exac.All$gene

#Load file with all genes
library(hgu133plus2.db)
```

```
## Loading required package: AnnotationDbi

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, as.vector, cbind,
##     colnames, do.call, duplicated, eval, evalq, Filter, Find, get,
##     grep, grepl, intersect, is.unsorted, lapply, lengths, Map,
##     mapply, match, mget, order, paste, pmax, pmax.int, pmin,
##     pmin.int, Position, rank, rbind, Reduce, rownames, sapply,
##     setdiff, sort, table, tapply, union, unique, unlist, unsplit
```

```
## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Loading required package: IRanges
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: org.Hs.eg.db
```

```
## Loading required package: DBI
```

```
##
```

```
##
```

```r
sym <- hgu133plus2SYMBOL
sym1 <- mappedkeys(sym)
sym2 <- as.list (sym[c(sym1)])
sym3 <- data.frame (sym2)
sym.probes <- names (sym2)
sym.genes <- sym3[1,]
sym.genes <- t(sym.genes)

allgenes <- sym.genes[!duplicated(sym.genes),]

y <- g
snp <- exacgenes
```

```r
#How many test geneset genes contain snps
x.in <- length (which(y %in% snp))
#how many do not
x.out <- length(y) - x.in
#total number of snp genes
tot.in <- length(snp)
#total number of all genes
tot.out <- length(allgenes)-length(tot.in)


#create count matrix
counts <- matrix (nrow=2, ncol=2)
counts [1,] <- c(x.in, tot.in)
counts [2,] <- c(x.out, tot.out)

#Conduct fisher's exact test for count data
a5 <-fisher.test (counts)
enrich <- a5$p
print(enrich)
```

```
## [1] 5.410963e-11
```

This shows that my DEGs are highly enriched in the list of constrained genes. I then looked for enrichment of the list John had generated which was all genes with PLI>=0.95 which indicates they are the top 5% of constrained genes. This list was approximately 2500 in length.

```r
setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Data/GeneExpressionAnalysis")

G <- read.table(file = "allgenes.txt")
g <- G$V1

setwd("/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Code/Results/GeneExpression")
#Load ExAC Data
setwd(dir = "/Users/clairegreen/Documents/PhD/TDP-43/TDP-43_Code/Results/GeneExpression")
L <- read.table(file = "exac.pli.0.95.txt")
l <- L$V1

y <- g
snp <- l

#How many test geneset genes contain snps
x.in <- length (which(y %in% snp))
#how many do not
x.out <- length(y) - x.in
#total number of snp genes
tot.in <- length(snp)
#total number of all genes
tot.out <- length(allgenes)-length(tot.in)


#create count matrix
counts <- matrix (nrow=2, ncol=2)
counts [1,] <- c(x.in, tot.in)
counts [2,] <- c(x.out, tot.out)

#Conduct fisher's exact test for count data
a5 <-fisher.test (counts)
enrich <- a5$p
print(enrich)
```

```
## [1] 0.001323247
```