# Analysis of GSE25007 to prepare a ranked list of genes by their correlation with age.

*Jason Kennerdell*

*9/19/2017*

**Synopsis:**

Here I am ranking genes according to the degree to which they are correlated with age. This list will be used with the orederedList package in another script to deterimine whether there is significant overlap between genes differentially expressed in *E(z)* mutant brains and genes correlated with age.

**Load needed packages**

```
library(ggplot2); library(limma); library(GEOquery); library(affy); library(gcrma);library(affyPLM); li
library(annotate); library(drosophila2.db)
```

**Download data from GEO**

```
inpath <- "~/Desktop/brain/GSE25007/GSE25007"
# download the raw CEL files:
while(!file.exists(file.path(inpath, "GSE25007_RAW.tar"))){
  eList2 <- getGEOSuppFiles("GSE25007", makeDirectory=F, baseDir=inpath)
  }
```

**Unzip data from GEO**

```
tar <- grep("tar$", list.files(inpath), value=T)
while(!file.exists(file.path(inpath, "GSM614349.CEL"))){
  setwd(inpath)
  untar(file.path(inpath, tar))
  CEL_gz <- grep("CEL.gz$", list.files(inpath), value=TRUE)
  lapply(file.path(inpath, CEL_gz), gunzip, remove=FALSE)
}
CEL <- grep("CEL$", list.files(inpath), value=TRUE)
```

**Calculate correlations and coefficients:**

```
setwd(inpath)
raw.data <- read.affybatch(CEL)
eset <- gcrma(raw.data)
```

```
## Adjusting for optical effect.........Done.
## Computing affinities.Done.
## Adjusting for non-specific binding.........Done.
## Normalizing
```

## Calculating Expression

```
dat <- exprs(eset)

# Calculate p value for each probeset, to be applied by row of the matrix:
getP <- function(x){
        age = c(3,3,3,30,30,30,60,60,60)
        fit <- lm(formula = x ~ age)
        summary(fit)$coefficients[2,4]
}
getBeta <- function(x){
        age = c(3,3,3,30,30,30,60,60,60)
        fit <- lm(formula = x ~ age)
        summary(fit)$coefficients[2,1]
}
sumDat <- data.frame(row.names=rownames(dat))
sumDat$p.value <- apply(dat, MARGIN=1, FUN=getP)
sumDat$beta <- apply(dat, MARGIN=1, FUN=getBeta)
#Remove AFFX probesets
sumDat <- sumDat[!grepl("^AFFX-", rownames(sumDat)),]
#Remove NaN: In some probesets, all values are the same across age, giving pvalue NaN
sumDat <- sumDat[!is.nan(sumDat$p.value),]

sumDat <- sumDat[order(sumDat$p.value),]
```

### Add Ensembl gene names for each probeset

```
probes_1 <- as.character(rownames(sumDat))
sumDat$ensembl <- mapIds(drosophila2.db, keys=probes_1,
                            c("ENSEMBL"), keytype="PROBEID", multiVals="asNA")
```

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

```
sumDat <- sumDat[!is.na(sumDat$ensembl),]
```

### Compress genes into one row per ensembl name

```
sumDat$probeset <- rownames(sumDat)

compressed_sumDat <- ddply(sumDat, .(ensembl),
            summarize, beta=mean(beta), p.value=mean(p.value))

# show how it worked:
sumDat[sumDat$ensembl=="FBgn0000015",]
```

```
##                p.value        beta     ensembl     probeset
## 1627871_at   0.2827806 0.005213608 FBgn0000015   1627871_at
## 1641589_s_at 0.4743941 0.004036759 FBgn0000015 1641589_s_at
## 1632781_s_at 0.7235152 0.002797969 FBgn0000015 1632781_s_at
```

```
compressed_sumDat[compressed_sumDat$ensembl=="FBgn0000015",]
```

```
##       ensembl        beta   p.value
```

```
## 3 FBgn0000015 0.004016112 0.4935633
```

## Add a product to sort the list by:

```
compressed_sumDat$prod <- -log10(compressed_sumDat$p.value)*compressed_sumDat$beta
compressed_sumDat <- compressed_sumDat[order(compressed_sumDat$prod, decreasing=T),]
```

## Prepare a ranked gene list entirely by p value:

```
upgenes <- compressed_sumDat[compressed_sumDat$beta>0,]
upgenes <- upgenes[order(upgenes$p.value),]
head(upgenes)
```

```
##          ensembl       beta      p.value      prod
## 3946 FBgn0031775 0.09066047 9.907021e-08 0.6349911
## 8941 FBgn0040735 0.02357414 5.324881e-07 0.1478968
## 2651 FBgn0029507 0.08291106 5.737657e-07 0.5174699
## 8939 FBgn0040733 0.02273366 4.631188e-06 0.1212683
## 3366 FBgn0030808 0.02275069 6.806960e-06 0.1175539
## 3046 FBgn0030310 0.04808252 7.391507e-06 0.2467242
```

```
tail(upgenes)
```

```
##           ensembl         beta   p.value         prod
## 4430  FBgn0032609 2.421159e-05 0.9979587 2.148625e-08
## 5671  FBgn0034807 7.971699e-07 0.9981328 6.470244e-10
## 7697  FBgn0038233 3.341677e-06 0.9994559 7.898180e-10
## 1862  FBgn0024365 6.094877e-06 0.9995239 1.260559e-09
## 8580  FBgn0039698 3.493345e-06 0.9997324 4.059912e-10
## 11502 FBgn0265267 2.000863e-07 0.9997497 2.175281e-11
```

```
dngenes <- compressed_sumDat[compressed_sumDat$beta<=0,]
dngenes <- dngenes[order(dngenes$p.value, decreasing=T),]
head(dngenes)
```

```
##           ensembl          beta   p.value          prod
## 5010  FBgn0033663 -4.536187e-06 0.9994814 -1.021885e-09
## 10318 FBgn0069354 -8.589052e-06 0.9987560 -4.643088e-09
## 4628  FBgn0032925 -1.456335e-05 0.9986824 -8.338921e-09
## 6177  FBgn0035736 -1.504242e-05 0.9982556 -1.140575e-08
## 490   FBgn0003187 -5.976491e-08 0.9981857 -4.713347e-11
## 7106  FBgn0037270 -1.164075e-05 0.9979578 -1.033485e-08
```

```
tail(dngenes)
```

```
##           ensembl         beta      p.value        prod
## 6124  FBgn0035649 -0.018821410 4.377408e-05 -0.08203844
## 692   FBgn0004242 -0.005928273 3.225493e-05 -0.02662627
## 10122 FBgn0053202 -0.015904473 2.073152e-05 -0.07448651
## 5636  FBgn0034731 -0.015560635 1.480967e-05 -0.07514938
## 8719  FBgn0039937 -0.030295639 1.348605e-05 -0.14754326
## 8700  FBgn0039900 -0.018599584 4.715846e-06 -0.09906957
```

```
lst <- rbind(upgenes, dngenes)
write.csv(file=file.path("~/Desktop/brain/GSE25007",
                         "ranked_aging_correlated_genes_pvalue_only.csv"),
          lst)
```