Skeleton Key for RNAseq analysis

Written By: Ciera Martinez

libraries

```r
library(edgeR)
```

Read in YAML guide

```r
library(yaml)
yamls <- yaml.load_file("./de.yml")
```

```r
sample1 <- yamls$sample1
sample2 <- yamls$sample2

sample1
```

```
## [1] "wtaother"
```

```r
sample2
```

```
## [1] "wtcother"
```

Read in Data

Read in raw count data per gene.

```r
counts <- read.delim("../sam2countsResults.tsv",row.names=1)

#check the file
head(counts)
summary(counts)
colnames(counts)
#need to convert NA to 0 counts
counts[is.na(counts)] <- 0
```

Subset per DE expirement

I am going to start by subsetting the particular treatments I am looking at.

```r
colnames(counts)
```

```
##  [1] "tf2ambr1"      "tf2ambr3"      "tf2ambr4"      "tf2ambr6"     
##  [5] "tf2aother1"    "tf2aother2"    "tf2aother4"    "tf2aother7"   
##  [9] "tf2bmbr2"      "tf2bmbr5"      "tf2bmbr6"      "tf2bother1"   
## [13] "tf2bother3"    "tf2bother4"    "tf2bother6"    "tf2cmbr1.4"   
## [17] "tf2cmbr3"      "tf2cmbr6"      "tf2cmbr7"      "tf2cother2"   
## [21] "tf2cother5"    "tf2cother6"    "tf2cother7"    "wtambr2"      
## [25] "wtambr4"       "wtambr5"       "wtaother1"     "wtaother5"    
## [29] "wtaother6"     "wtaother7"     "wtaother8"     "wtbmbr2"      
## [33] "wtbmbr3"       "wtbmbr6"       "wtbmbr8"       "wtbother1.4"  
## [37] "wtbother3"     "wtbother5"     "wtbother8"     "wtcmbr10"     
## [41] "wtcmbr1.4.6"   "wtcmbr2"       "wtcmbr3"       "wtcmbr7"      
## [45] "wtcmbr9"       "wtcother1.3.4" "wtcother2"     "wtcother6"
```

```r
counts1 <- counts[,grep(sample1, colnames(counts), value = TRUE)]
count1Len <- length(colnames(counts1)) #used in to specify library group in next step.

counts2 <- counts[,grep(sample2, colnames(counts), value = TRUE)]
count2Len <- length(colnames(counts2)) #used to specify library group in next step.

counts <- cbind(counts1, counts2)

head(counts)
```

```
##                    wtaother1 wtaother5 wtaother6 wtaother7 wtaother8
## Solyc00g005040.2.1         1         1         1         0         2
## Solyc00g005050.2.1        17        16         9         2         3
## Solyc00g005060.1.1         0         0         0         0         2
## Solyc00g005070.1.1         8         6         5         5         6
## Solyc00g005080.1.1        18        37         6        10         7
## Solyc00g005150.1.1         2         5         0         0         2
##                    wtcother1.3.4 wtcother2 wtcother6
## Solyc00g005040.2.1             0         0        12
## Solyc00g005050.2.1             2         6        37
## Solyc00g005060.1.1            13         0         0
## Solyc00g005070.1.1           169         6        24
## Solyc00g005080.1.1            11        26        35
## Solyc00g005150.1.1             2         1         5
```

Add column specifying library Group

Make a vector called group that will be used to make a new column named group to identify library region type.

```r
group <- c(rep(sample1, count1Len), rep(sample2, count2Len))
d <- DGEList(counts=counts,group=group)
```

```r
d$samples
```

```
##                 group lib.size norm.factors
## wtaother1     wtaother   929017            1
## wtaother5     wtaother  1555921            1
## wtaother6     wtaother   498294            1
## wtaother7     wtaother   479003            1
## wtaother8     wtaother   510148            1
## wtcother1.3.4 wtcother   197345            1
## wtcother2     wtcother   319043            1
## wtcother6     wtcother  1525172            1
```

```
cpm.d <- cpm(d)
d <- d[rowSums(cpm.d>5)>=3,] #change to 5
d <- estimateCommonDisp(d,verbose=T)
```

```
## Disp = 0.287 , BCV = 0.5358
```

```
d <- calcNormFactors(d)
d <- estimateCommonDisp(d)

DEtest <- exactTest(d,pair=c(sample1,sample2))
head(DEtest$table)
```

```
##                         logFC logCPM    PValue
## Solyc00g005050.2.1  0.76915  4.057 2.788e-01
## Solyc00g005070.1.1  5.38653  7.065 1.184e-16
## Solyc00g005080.1.1  1.60087  5.056 8.115e-03
## Solyc00g005160.1.1  1.70235  3.295 3.466e-02
## Solyc00g005440.1.1  0.28819  4.802 7.631e-01
## Solyc00g005840.2.1 -0.03526  4.886 9.759e-01
```

```
results <- topTags(DEtest, n=Inf)
head(results)
```

```
## Comparison of groups:  wtcother-wtaother
##                     logFC logCPM    PValue       FDR
## Solyc01g022780.1.1 8.302  8.536 1.702e-29 2.668e-25
## Solyc10g050260.1.1 7.476 10.450 7.060e-29 5.535e-25
## Solyc07g039270.2.1 7.552  9.782 1.321e-28 6.906e-25
## Solyc10g036800.1.1 7.390  9.791 3.543e-28 1.389e-24
## Solyc10g052420.1.1 7.406  9.823 4.887e-28 1.532e-24
## Solyc11g020560.1.1 7.162 11.675 6.253e-28 1.634e-24
```

```
dim(results$table)
```

```
## [1] 15678     4
```

```
sum(results$table$FDR<.05) # How many are DE genes?
```

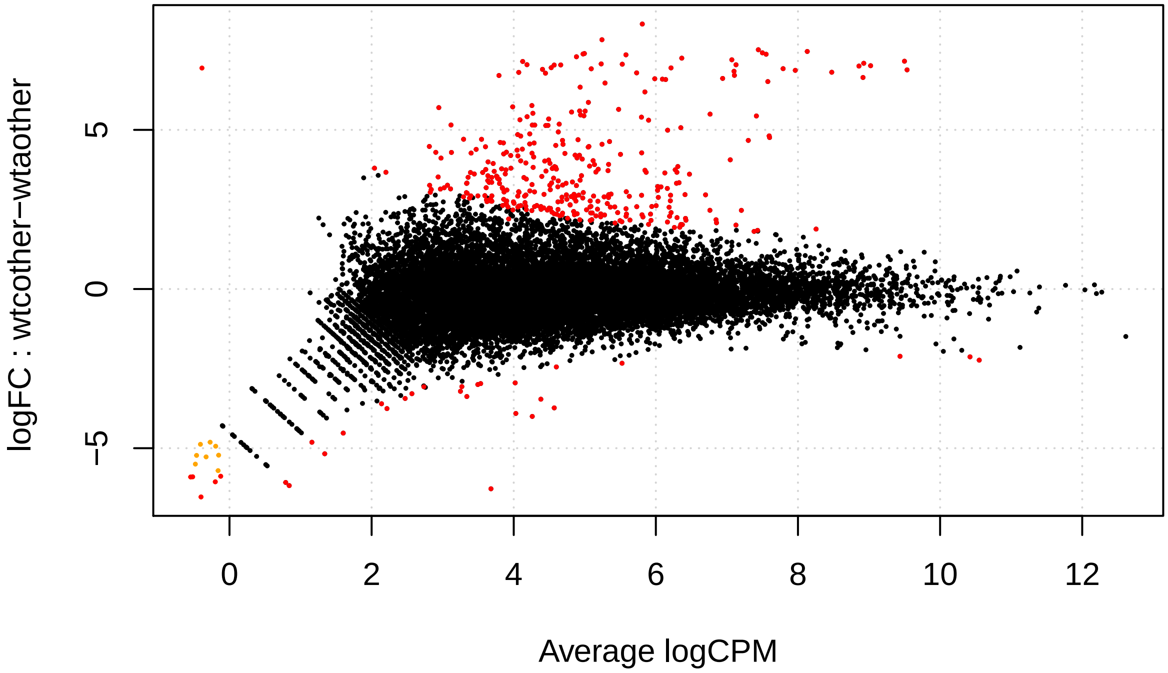
```
## [1] 378
```

```
summary(decideTestsDGE(DEtest,p.value=.05))
```

```
##    [,1]
## -1    31
## 0  15300
## 1    347
```

```
sig.genes <- rownames(results$table[results$table$FDR<0.05,]) # outputs just significant gene names

plotSmear(d,de.tags=sig.genes)
```



Subset by all the ones with a significant score

```
results.sig <- subset(results$table, results$table$FDR < 0.05)
```

What are the genes that are misexpressed? For this we need to add some annotation

Essentially we are merging two annotations files to 1.) only sig genes 2.) all genes

```
annotation1<- read.delim("../ITAG2.3_all_Arabidopsis_ITAG_annotations.tsv", header=FALSE)  #Changed to
colnames(annotation1) <- c("ITAG", "SGN_annotation")
annotation2<- read.delim ("../ITAG2.3_all_Arabidopsis_annotated.tsv")
annotation <- merge(annotation1,annotation2, by = "ITAG")

#Making the only significant gene table
results.sig$ITAG <- rownames(results.sig)  #change row.names to ITAG for merging
results.sig.annotated <- merge(results.sig,annotation,by = "ITAG") #This is merging to only sig genes

#Making all table

results$table$ITAG <- rownames(results$table)
results.all.annotated <- merge(results$table, annotation,by = "ITAG")
```

Write table with results

```
write.table(results.all.annotated, file=paste(sample1,"_",sample2,"_","DE_all.txt",sep=""),sep="\t",row
write.table(results.sig.annotated, file=paste(sample1,"_",sample2,"_","DE_sig.txt",sep=""),sep="\t",row
```

```r
library(rmarkdown)
render("skeletonDE.Rmd", "pdf_document", output_file = paste(sample1,"_",sample2,"_","DE.pdf",sep=""))
```