Skeleton Key for RNAseq analysis

Written By: Ciera Martinez

libraries

```
library(edgeR)
```

Read in YAML guide

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

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

sample1
```

```
## [1] "wtambr"
```

```
sample2
```

```
## [1] "wtcmbr"
```

Read in Data

Read in raw count data per gene.

```
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.

```
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"
```

```
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)
```

```
##                   wtambr2 wtambr4 wtambr5 wtcmbr10 wtcmbr1.4.6 wtcmbr2
## Solyc00g005040.2.1       0       2       8        0           9       3
## Solyc00g005050.2.1       0       6       6        5          38      21
## Solyc00g005060.1.1       0       0       1        1           3       0
## Solyc00g005070.1.1      24       3       9        5          12       7
## Solyc00g005080.1.1       9      15      19        0           7      19
## Solyc00g005150.1.1       0       1       2        0           1       3
##                   wtcmbr3 wtcmbr7 wtcmbr9
## Solyc00g005040.2.1       1       0       0
## Solyc00g005050.2.1      11       4       7
## Solyc00g005060.1.1       0       1       0
## Solyc00g005070.1.1       4       6       1
## Solyc00g005080.1.1      45       4       7
## Solyc00g005150.1.1       3       2       1
```

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.

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

```
d$samples
```

```
##              group lib.size norm.factors
## wtambr2     wtambr   395165            1
## wtambr4     wtambr   792542            1
## wtambr5     wtambr   632686            1
```

```
## wtcmbr10    wtcmbr   459717            1
## wtcmbr1.4.6 wtcmbr  1158809            1
## wtcmbr2     wtcmbr  1130695            1
## wtcmbr3     wtcmbr  1560130            1
## wtcmbr7     wtcmbr   374882            1
## wtcmbr9     wtcmbr   386974            1
```

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

```
## Disp = 0.3224 , BCV = 0.5678
```

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

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

```
##                       logFC logCPM    PValue
## Solyc00g005050.2.1  1.0894  3.923 9.420e-02
## Solyc00g005070.1.1 -2.5693  4.285 7.602e-05
## Solyc00g005080.1.1 -1.2091  4.430 3.876e-02
## Solyc00g005440.1.1  0.4826  4.733 4.314e-01
## Solyc00g005840.2.1 -0.4685  5.029 3.929e-01
## Solyc00g006470.1.1 -3.2989 11.483 4.224e-10
```

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

```
## Comparison of groups:  wtcmbr-wtambr
##                      logFC logCPM    PValue       FDR
## Solyc00g071180.2.1 -9.509  6.404 1.839e-30 2.795e-26
## Solyc08g079850.1.1 -6.631  9.008 3.066e-28 2.330e-24
## Solyc05g008070.2.1 -7.259  6.506 5.864e-25 2.971e-21
## Solyc03g062850.1.1 -6.607  6.771 1.328e-24 5.046e-21
## Solyc09g091110.2.1 -5.832  7.745 5.003e-23 1.521e-19
## Solyc11g020560.1.1 -5.606  9.471 1.099e-22 2.785e-19
```

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

```
## [1] 15199     4
```

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

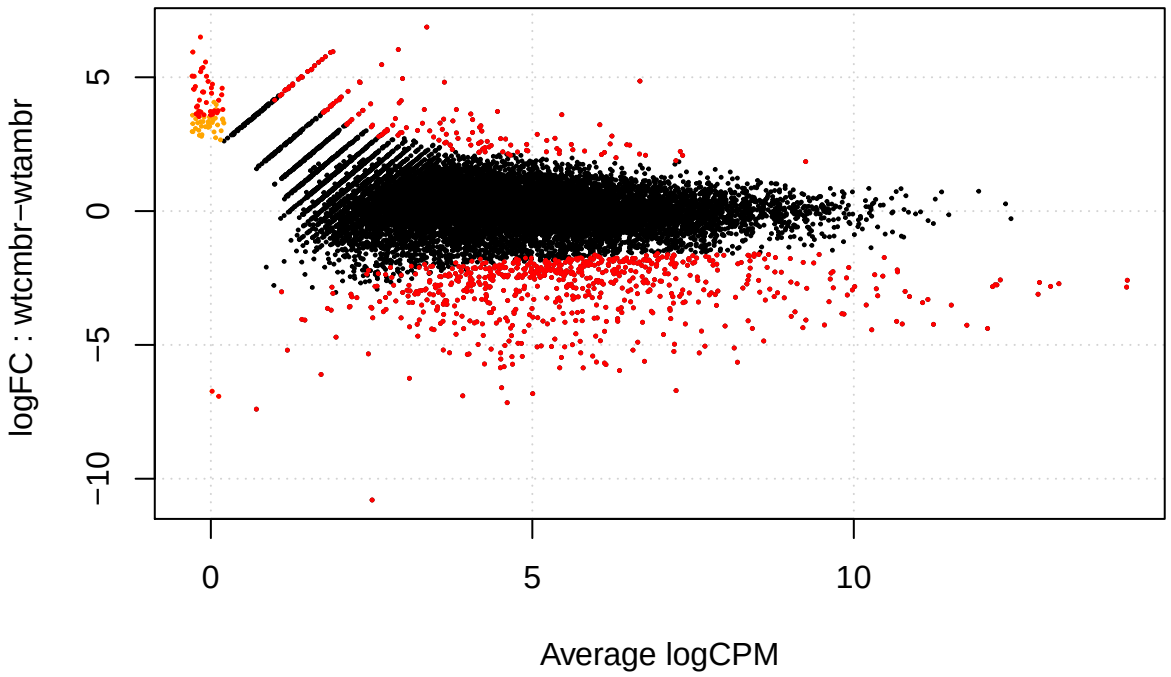
```
## [1] 880
```

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

```
##    [,1]
## -1   694
## 0  14319
## 1    186
```

```
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
```

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