Skeleton Key for RNAseq analysis

Written By: Ciera Martinez

See README.md for more detailed instructions of how to use script

Analysis

libraries

```
library(edgeR)
library(yaml)
```

Read in YAML guide

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

This part assigns your YMAL to a object in R. This will be used throughout the script to specify which sample types you are comparing.

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

sample1
```

```
## [1] "tf2ambr"
```

```
sample2
```

```
## [1] "tf2cmbr"
```

Read in Data

Read in raw count data per gene.

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

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

Subset DE expirement

Start by subsetting the particular treatments which are being compared.

```
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)
```

```
##                    tf2ambr1 tf2ambr3 tf2ambr4 tf2ambr6 tf2cmbr1.4 tf2cmbr3
## Solyc00g005040.2.1       12        0        3       12          0        6
## Solyc00g005050.2.1       33        1       14       17          1       34
## Solyc00g005060.1.1        1        5        1        1          0        1
## Solyc00g005070.1.1       14       22       23        5         23       11
## Solyc00g005080.1.1       19        2       25       32         22        7
## Solyc00g005150.1.1        3        0        0        4          1        3
##                    tf2cmbr6 tf2cmbr7
## Solyc00g005040.2.1        8        4
## Solyc00g005050.2.1       17       12
## Solyc00g005060.1.1        0        0
## Solyc00g005070.1.1        8        9
## Solyc00g005080.1.1        8       12
## Solyc00g005150.1.1        0        0
```

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)
```

Check to see if the group column matches your sample name and they are appropriate.

```
d$samples
```

```
##              group lib.size norm.factors
## tf2ambr1   tf2ambr  1313540            1
## tf2ambr3   tf2ambr    91726            1
## tf2ambr4   tf2ambr  1438416            1
## tf2ambr6   tf2ambr  1088653            1
## tf2cmbr1.4 tf2cmbr   443572            1
## tf2cmbr3   tf2cmbr  1337575            1
## tf2cmbr6   tf2cmbr   790129            1
## tf2cmbr7   tf2cmbr   832907            1
```

Differential expression using edgeR

Make sure there is full understanding on each edgeR command being used. The manual is amazing so read it *before* running the DE analysis below edgeR manual.

```
cpm.d <- cpm(d) #counts per mutant
d <- d[rowSums(cpm.d>5)>=3,] #This might be a line to adjust. It is removing genes with low counts.
d <- estimateCommonDisp(d,verbose=T)
```

```
## Disp = 0.4389 , BCV = 0.6625
```

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

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

```
##                       logFC logCPM  PValue
## Solyc00g005040.2.1 -0.43370  3.325 0.73762
## Solyc00g005050.2.1 -0.02528  4.014 0.96560
## Solyc00g005070.1.1 -1.34481  5.420 0.07163
## Solyc00g005080.1.1  0.03302  4.655 0.97402
## Solyc00g005440.1.1 -0.88019  4.922 0.21466
## Solyc00g005840.2.1 -0.10207  4.774 0.89713
```

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

```
## Comparison of groups:  tf2cmbr-tf2ambr
##                     logFC logCPM    PValue       FDR
## Solyc08g078890.1.1 -7.234  6.245 2.772e-13 4.121e-09
## Solyc07g044980.2.1  5.366  7.995 6.893e-11 5.123e-07
## Solyc07g045410.1.1 -5.325  7.378 1.942e-10 9.620e-07
## Solyc05g041220.1.1  5.795  6.220 4.955e-10 1.841e-06
## Solyc02g071980.2.1  5.021  7.447 6.218e-10 1.848e-06
## Solyc02g023990.2.1  5.142  6.761 1.356e-09 2.962e-06
```

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

```
## [1] 14864     4
```

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

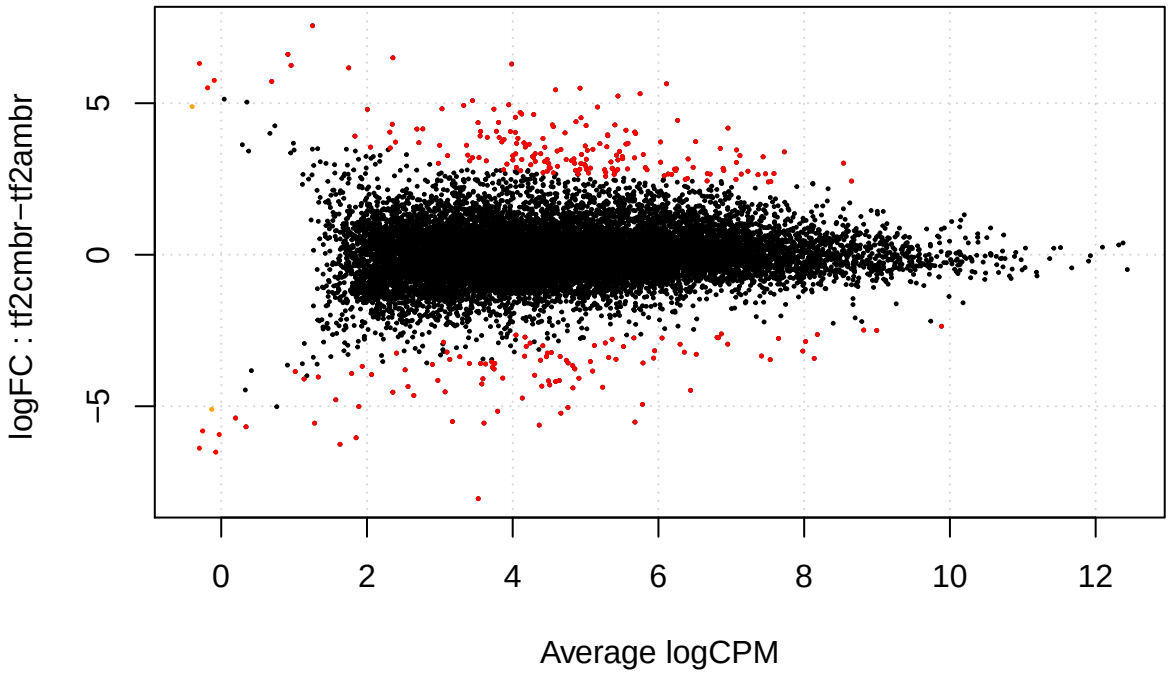
```
## [1] 282
```

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

```
##    [,1]
## -1   104
## 0  14582
## 1    178
```

```r
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 genes with a significant FDR score.

```r
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("../requisiteData/ITAG2.3_all_Arabidopsis_ITAG_annotations.tsv", header=FALSE)
colnames(annotation1) <- c("ITAG", "SGN_annotation")
annotation2<- read.delim ("../requisiteData/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
```

Now run the script below for a full `knitr` report of what was run and leave this report in the folder that the analysis was done with output files.

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