Skeleton Key for RNAseq analysis

Written By: Ciera Martinez

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

Run the `render()` function below and everything will be run with report at end.

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

Analysis (Aucutal start)

libraries

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

Read in YAML guide

This reads in the information in the `de.yml` file which has the two names of the samples you are interested in comparing.

```
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] "wtaother"
```

```
sample2
```

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

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. This might need to be modified depending on the naming of your samples. In my case each sample is named by sample and rep number, so the script is identifying any sample with the sample name given in the `de.yml` file.

```
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)
```

```
##                    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.

```
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
## 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
```

Differential expression using edgeR

Make sure there is full understanding of each edgeR command being used. The manual is amazing so read it *before* running the DE analysis below edgeR manual. There are many options and they must be set to be appropriate for your analysis.

```
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 = 0.287 , 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?
```

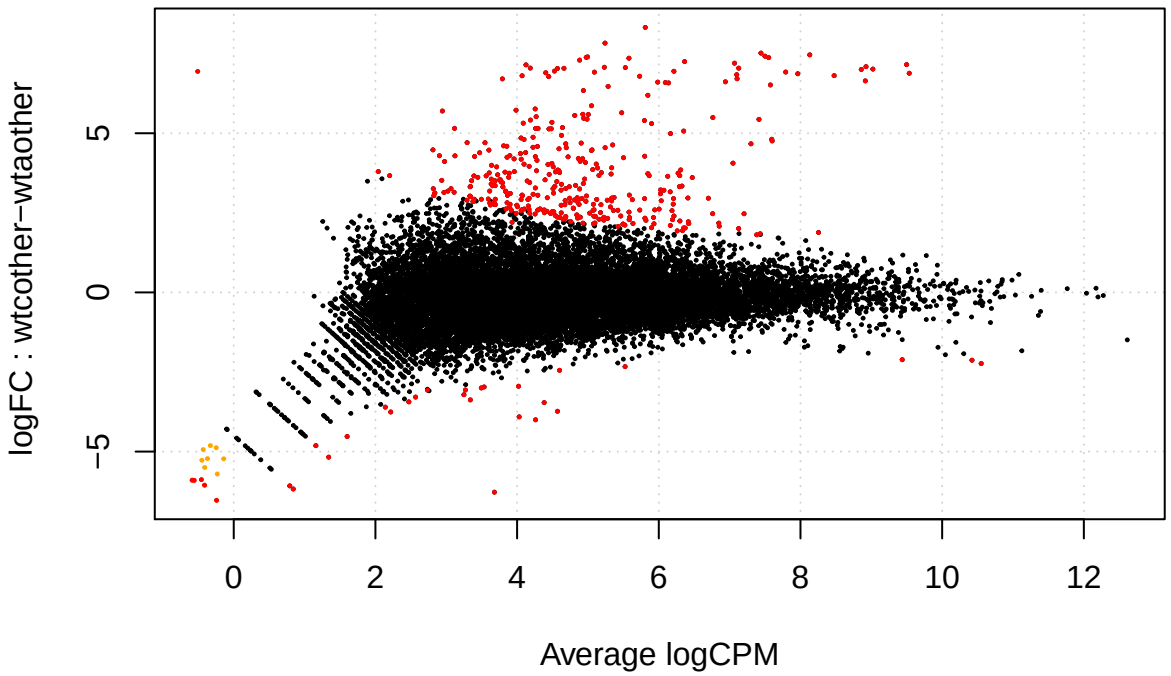
```
## [1] 378
```

```
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 all the genes with a significant FDR score less than .05.

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

dim(results.sig)

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")
head(annotation)
```

```
##                 ITAG
## 1 Solyc00g005000.2.1
## 2 Solyc00g005040.2.1
## 3 Solyc00g005050.2.1
## 4 Solyc00g005080.1.1
## 5 Solyc00g005900.1.1
## 6 Solyc00g006490.2.1
##
## 1                                                    Aspartic proteinase nepenthesin I (AHRD V1 **--
## 2                                                      Potassium channel (AHRD V1 ***- D0EM91_9ROSI)
## 3
## 4
## 5 Oxygen-evolving enhancer protein 1, chloroplastic (AHRD V1 ***- PSBO_SOLTU); contains Interpro doma
## 6    Serine/threonine-protein phosphatase 6 regulatory subunit 3 (AHRD V1 **-- SAPS3_HUMAN); contains
##         AGI symbol
## 1 AT3G20015   <NA>
## 2 AT5G46240   KAT1
## 3 AT5G11680   <NA>
## 4 ATCG01280 YCF2.2
## 5 AT5G66570  MSP-1
## 6 AT1G07990   <NA>
##
## 1
## 2
## 3
## 4
## 5
## 6 SIT4 phosphatase-associated family protein; similar to SIT4 phosphatase-associated family protein
##   X..identity alignment.length e.value bit.score percent.query.align
## 1       63.76              447  7e-148       520               89.94
## 2       66.02              103   2e-37       150               85.71
## 3       76.96              204   1e-88       322               98.98
## 4       91.25               80   2e-38       153               79.80
## 5       69.62               79   4e-26       112               78.79
## 6       61.92              856   0e+00       979               99.77
```

```
#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", all.x=TRUE) #This is merging only sig
```

```
#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.