Skeleton Key for RNAseq analysis

Written By: Ciera Martinez

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

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=""))
```

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] "tf2bother"
```

```
sample2
```

```
## [1] "wtbother"
```

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)
```

```
##                   tf2bother1 tf2bother3 tf2bother4 tf2bother6 wtbother1.4
## Solyc00g005040.2.1          6          3          5          2           0
## Solyc00g005050.2.1         46          9         23         22           0
## Solyc00g005060.1.1          0          1          1          1           0
## Solyc00g005070.1.1         25          4         11         11           0
## Solyc00g005080.1.1         52         12         15         12           0
## Solyc00g005150.1.1         11          0          0          1           0
##                   wtbother3 wtbother5 wtbother8
## Solyc00g005040.2.1         8         0         3
## Solyc00g005050.2.1        25         0        14
## Solyc00g005060.1.1         0         0         0
## Solyc00g005070.1.1         6         2         4
## Solyc00g005080.1.1        29         0        11
## Solyc00g005150.1.1         2         0         2
```

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
## tf2bother1  tf2bother  2415227            1
## tf2bother3  tf2bother   626786            1
## tf2bother4  tf2bother  1003586            1
## tf2bother6  tf2bother   854903            1
## wtbother1.4  wtbother     1421            1
## wtbother3    wtbother  1076939            1
## wtbother5    wtbother   200587            1
## wtbother8    wtbother   499487            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 = 0.327 , BCV = 0.5718
```

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

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

```
##                       logFC logCPM    PValue
## Solyc00g005050.2.1 -0.12332  5.171 5.771e-01
## Solyc00g005070.1.1 -0.38788  4.028 7.088e-01
## Solyc00g005080.1.1  0.06034  4.796 7.598e-01
## Solyc00g005440.1.1 -0.64142  5.222 2.673e-01
## Solyc00g005840.2.1  3.21142  7.672 7.769e-06
## Solyc00g005880.1.1  0.45501  3.491 6.026e-01
```

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

```
## Comparison of groups:  wtbother-tf2bother
##                       logFC logCPM    PValue       FDR
## Solyc04g074380.2.1 7.792 12.140 1.382e-20 2.089e-16
## Solyc10g078540.1.1 7.573 13.679 3.602e-20 2.722e-16
## Solyc04g074390.2.1 6.989 10.519 1.278e-17 6.440e-14
## Solyc02g076780.2.1 6.702 10.766 7.274e-17 2.749e-13
## Solyc09g059140.1.1 6.585  8.743 1.108e-15 3.350e-12
## Solyc12g097060.1.1 6.792  8.043 7.946e-15 2.002e-11
```

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

```
## [1] 15118     4
```

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

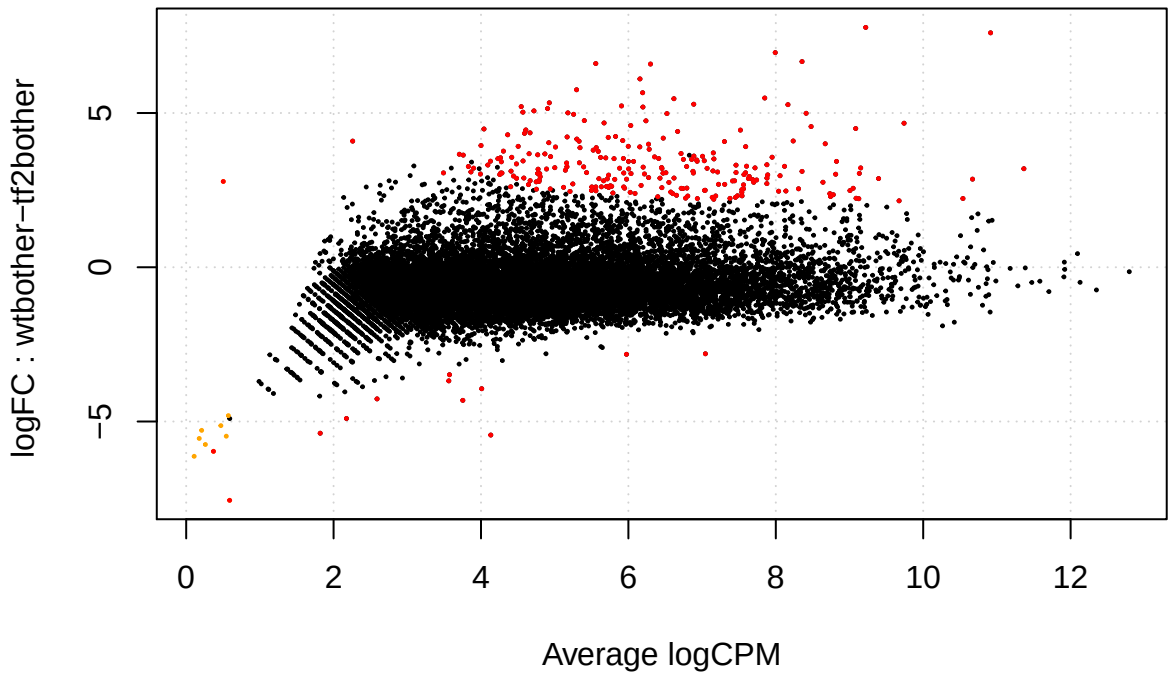
```
## [1] 252
```

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

```
##    [,1]
## -1    12
## 0  14866
## 1    240
```

```
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.

```
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
```