

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

libraries

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

Read in YAML guide

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

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] "wtbmbr"
```

```
sample2
```

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

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 experiment

Start by subsetting the particular treatments which are being compared.

```
colnames(counts)
```

```
## [1] "tf2ambr1"      "tf2ambr3"      "tf2ambr4"      "tf2ambr6"
## [5] "tf2aoth1"      "tf2aoth2"      "tf2aoth4"      "tf2aoth7"
## [9] "tf2bmbr2"      "tf2bmbr5"      "tf2bmbr6"      "tf2bth1"
## [13] "tf2bth3"      "tf2bth4"      "tf2bth6"      "tf2cmbr1.4"
## [17] "tf2cmbr3"      "tf2cmbr6"      "tf2cmbr7"      "tf2coth2"
## [21] "tf2coth5"      "tf2coth6"      "tf2coth7"      "wtambr2"
## [25] "wtambr4"      "wtambr5"      "wtaoth1"      "wtaoth5"
## [29] "wtaoth6"      "wtaoth7"      "wtaoth8"      "wtbmbr2"
## [33] "wtbmbr3"      "wtbmbr6"      "wtbmbr8"      "wtbth1.4"
## [37] "wtbth3"      "wtbth5"      "wtbth8"      "wtcmbr10"
## [41] "wtcmbr1.4.6"  "wtcmbr2"      "wtcmbr3"      "wtcmbr7"
## [45] "wtcmbr9"      "wtcoth1.3.4"  "wtcoth2"      "wtcoth6"
```

```
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)
```

```
##           wtbmbr2 wtbmbr3 wtbmbr6 wtbmbr8 wtcmb10 wtcmb1.4.6
## Solyc00g005040.2.1      2      4      3      0      0      9
## Solyc00g005050.2.1     20      5     18      0      5     38
## Solyc00g005060.1.1      1      2      1      1      1      3
## Solyc00g005070.1.1     14      6     12     14      5     12
## Solyc00g005080.1.1     25     15     27      0      0      7
## Solyc00g005150.1.1      0      0      3      0      0      1
##           wtcmb2 wtcmb3 wtcmb7 wtcmb9
## Solyc00g005040.2.1      3      1      0      0
## Solyc00g005050.2.1     21     11      4      7
## Solyc00g005060.1.1      0      0      1      0
## Solyc00g005070.1.1      7      4      6      1
## Solyc00g005080.1.1     19     45      4      7
## Solyc00g005150.1.1      3      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)
```

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

```
d$samples
```

```
##           group lib.size norm.factors
## wtbmbr2      wtbmbr 1355352          1
## wtbmbr3      wtbmbr 1213142          1
## wtbmbr6      wtbmbr 1598917          1
## wtbmbr8      wtbmbr  48352          1
## wtcnbr10     wtcnbr  459717          1
## wtcnbr1.4.6  wtcnbr 1158809          1
## wtcnbr2      wtcnbr 1130695          1
## wtcnbr3      wtcnbr 1560130          1
## wtcnbr7      wtcnbr  374882          1
## wtcnbr9      wtcnbr  386974          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 = 0.2077 , BCV = 0.4558
```

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

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

```
##           logFC logCPM   PValue
## Solyc00g005050.2.1  0.75767  4.024 9.450e-02
## Solyc00g005070.1.1 -2.04638  4.004 5.419e-05
## Solyc00g005080.1.1 -0.05269  4.209 8.636e-01
## Solyc00g005440.1.1  0.01732  5.026 1.000e+00
## Solyc00g005840.2.1  0.34245  4.939 2.785e-01
## Solyc00g005880.1.1 -0.12539  2.492 3.808e-01
```

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

```
## Comparison of groups: wtcnbr-wtbmbr
##           logFC logCPM   PValue   FDR
## Solyc08g022200.1.1 -6.906  8.478 1.040e-39 1.602e-35
## Solyc10g050260.1.1 -6.515  9.125 2.307e-37 1.777e-33
## Solyc07g039270.2.1 -6.508  8.325 4.765e-36 2.228e-32
## Solyc08g060910.1.1 -6.182  9.704 5.785e-36 2.228e-32
## Solyc11g020560.1.1 -6.034 10.234 1.900e-35 5.854e-32
## Solyc11g017260.1.1 -5.948 10.359 1.009e-34 2.592e-31
```

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

```
## [1] 15406      4
```

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

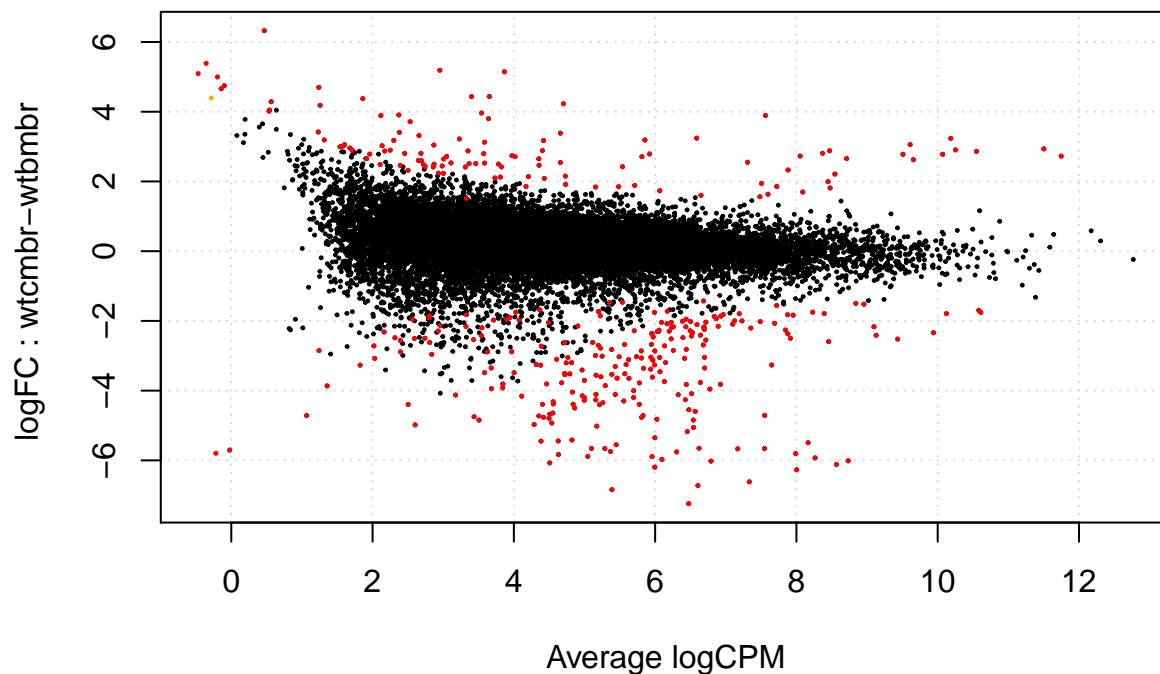
```
## [1] 358
```

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

```
##      [,1]  
## -1    242  
##  0   15048  
##  1     116
```

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

```
plotSmeard(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)
```

```
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 ***- DOEM91_9ROSI
## 3
## 4
## 5 Oxygen-evolving enhancer protein 1, chloroplastic (AHRD V1 ***- PSBO_SOLTU); contains Interpro dom
## 6 Serine/threonine-protein phosphatase 6 regulatory subunit 3 (AHRD V1 ***- SAPS3_HUMAN); contain
##      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 to only

#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.