

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.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] "tf2bmbr"
```

```
sample2
```

```
## [1] "tf2cother"
```

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)
```

```
##           tf2bmbr2 tf2bmbr5 tf2bmbr6 tf2coth2 tf2coth5
## Solyc00g005040.2.1      0      0      1      3      8
## Solyc00g005050.2.1      0      3     16      4     10
## Solyc00g005060.1.1     19      0      0      1      1
## Solyc00g005070.1.1    230      7     12      4     11
## Solyc00g005080.1.1      6     12     33      9     21
## Solyc00g005150.1.1      0      1      1      1      6
##           tf2coth6 tf2coth7
## Solyc00g005040.2.1      4      1
## Solyc00g005050.2.1     16     12
## Solyc00g005060.1.1      2      1
## Solyc00g005070.1.1      5      5
## Solyc00g005080.1.1     14      3
## Solyc00g005150.1.1      1      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
## tf2bmbr2      tf2bmbr   189160         1
## tf2bmbr5      tf2bmbr   727355         1
## tf2bmbr6      tf2bmbr  1244342         1
## tf2cother2    tf2cother   723602         1
## tf2cother5    tf2cother  1216379         1
## tf2cother6    tf2cother   838942         1
## tf2cother7    tf2cother   676969         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 = 0.7634 , BCV = 0.8737
```

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

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

```
##           logFC logCPM    PValue
## Solyc00g005050.2.1  0.7305  3.533 4.209e-01
## Solyc00g005070.1.1 -7.0356  8.468 3.650e-10
## Solyc00g005080.1.1 -1.4513  4.559 1.286e-01
## Solyc00g005440.1.1  0.3762  4.379 6.344e-01
## Solyc00g005840.2.1  1.9078  5.241 4.566e-02
## Solyc00g006470.1.1  1.2717  8.024 1.931e-01
```

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

```
## Comparison of groups: tf2cother-tf2bmbr
##           logFC logCPM    PValue    FDR
## Solyc07g039270.2.1 -10.889  12.55 2.220e-17 3.236e-13
## Solyc08g022200.1.1 -10.615  12.31 6.384e-17 4.654e-13
## Solyc00g020010.1.1 -10.201  10.84 6.817e-16 3.313e-12
## Solyc10g036800.1.1  -9.834  11.91 1.755e-15 6.395e-12
## Solyc01g028970.1.1  -9.410  11.97 8.948e-15 2.609e-11
## Solyc10g050260.1.1  -9.229  12.44 1.715e-14 3.776e-11
```

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

```
## [1] 14579      4
```

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

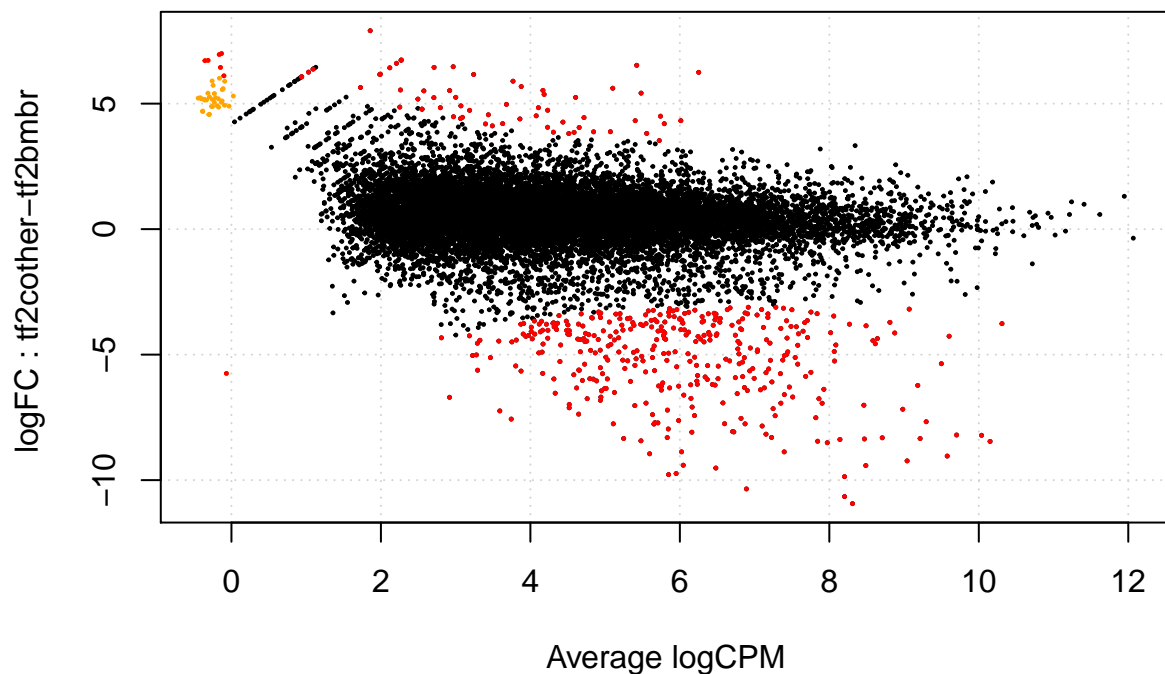
```
## [1] 441
```

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

```
##      [,1]  
## -1    372  
##  0   14138  
##  1      69
```

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

```
plotSmea(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

```

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=""))

```