

Skeleton Key for RNAseq analysis

Written By: Ciera Martinez

libraries

```
library(edgeR)
```

Read in YAML guide

```
library(yaml)
yaml1 <- yaml.load_file("./de.yml")
```

```
sample1 <- yaml1$sample1
sample2 <- yaml1$sample2
```

```
sample1
```

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

```
sample2
```

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

Read in Data

Read in raw count data per gene.

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

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

Subset per DE experiment

I am going to start by subsetting the particular treatments I am looking at.

```
colnames(counts)
```

```
## [1] "tf2ambr1"      "tf2ambr3"      "tf2ambr4"      "tf2ambr6"
## [5] "tf2aother1"    "tf2aother2"    "tf2aother4"    "tf2aother7"
## [9] "tf2bmr2"       "tf2bmr5"       "tf2bmr6"       "tf2bmr1"
## [13] "tf2bmr3"       "tf2bmr4"       "tf2bmr6"       "tf2cmbr1.4"
## [17] "tf2cmbr3"      "tf2cmbr6"      "tf2cmbr7"      "tf2cother2"
## [21] "tf2cother5"    "tf2cother6"    "tf2cother7"    "wtambr2"
## [25] "wtambr4"       "wtambr5"       "wtaother1"     "wtaother5"
## [29] "wtaother6"     "wtaother7"     "wtaother8"     "wtbmr2"
## [33] "wtbmr3"        "wtbmr6"        "wtbmr8"        "wtbmr1.4"
## [37] "wtbmr3"        "wtbmr5"        "wtbmr8"        "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)
```

```
##                wtcmb10 wtcmb1.4.6 wtcmb2 wtcmb3 wtcmb7 wtcmb9
## Solyc00g005040.2.1      0          9      3      1      0      0
## Solyc00g005050.2.1      5         38     21     11      4      7
## Solyc00g005060.1.1      1          3      0      0      1      0
## Solyc00g005070.1.1      5         12      7      4      6      1
## Solyc00g005080.1.1      0          7     19     45      4      7
## Solyc00g005150.1.1      0          1      3      3      2      1
##                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)
```

```
d$samples
```

```
##                group lib.size norm.factors
## wtcmb10          wtcmb  459717           1
## wtcmb1.4.6       wtcmb 1158809           1
## wtcmb2           wtcmb 1130695           1
```

```
## wtcnbr3          wtcnbr 1560130      1
## wtcnbr7          wtcnbr  374882      1
## wtcnbr9          wtcnbr  386974      1
## wtcother1.3.4    wtcother 197345      1
## wtcother2        wtcother  319043      1
## wtcother6        wtcother 1525172      1
```

```
cpm.d <- cpm(d)
d <- d[rowSums(cpm.d>5)>=3,] #change to 5
d <- estimateCommonDisp(d,verbose=T)
```

```
## Disp = 0.3151 , BCV = 0.5614
```

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

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

```
##              logFC logCPM   PValue
## Solyc00g005050.2.1  0.24832  4.288 7.080e-01
## Solyc00g005070.1.1  5.47739  6.832 9.953e-18
## Solyc00g005080.1.1  1.92902  4.908 1.338e-03
## Solyc00g005160.1.1  1.73665  3.041 2.264e-02
## Solyc00g005440.1.1 -0.01889  5.024 9.531e-01
## Solyc00g005840.2.1 -0.21612  4.981 7.299e-01
```

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

```
## Comparison of groups: wtcother-wtcnbr
##              logFC logCPM   PValue   FDR
## Solyc10g052420.1.1  8.110  9.592 1.187e-33 1.881e-29
## Solyc08g023400.1.1  8.170  8.802 1.488e-32 1.179e-28
## Solyc10g050260.1.1  7.762 10.220 2.352e-32 1.243e-28
## Solyc07g039270.2.1  7.822  9.553 3.323e-32 1.317e-28
## Solyc01g028970.1.1  7.723  9.608 8.692e-32 2.755e-28
## Solyc11g020560.1.1  7.485 11.445 1.156e-31 3.008e-28
```

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

```
## [1] 15850      4
```

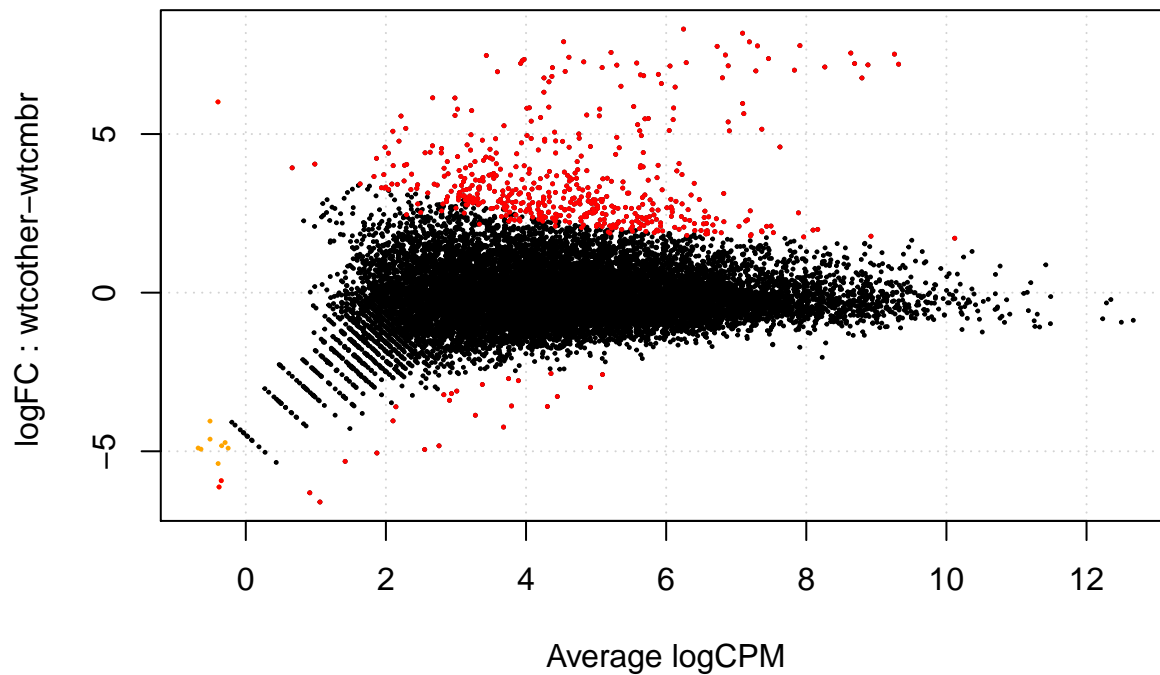
```
sum(results$table$FDR<.05) # How many are DE genes?
```

```
## [1] 538
```

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

```
##      [,1]
## -1    25
##  0 15312
##  1   513
```

```
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 ones with a significant 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("../ITAG2.3_all_Arabidopsis_ITAG_annotations.tsv", header=FALSE) #Changed to
colnames(annotation1) <- c("ITAG", "SGN_annotation")
annotation2<- read.delim("../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=""))
```