

# 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] "wtbmbr"
```

```
sample2
```

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

## 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)
```

```
##          wtbmr2 wtbmr3 wtbmr6 wtbmr8 wtcmr10 wtcmr1.4.6
## 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
##          wtcmr2 wtcmr3 wtcmr7 wtcmr9
## 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)
```

```
d$samples
```

```
##          group lib.size norm.factors
## wtbmr2      wtbmr 1355352          1
## wtbmr3      wtbmr 1213142          1
## wtbmr6      wtbmr 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
```

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

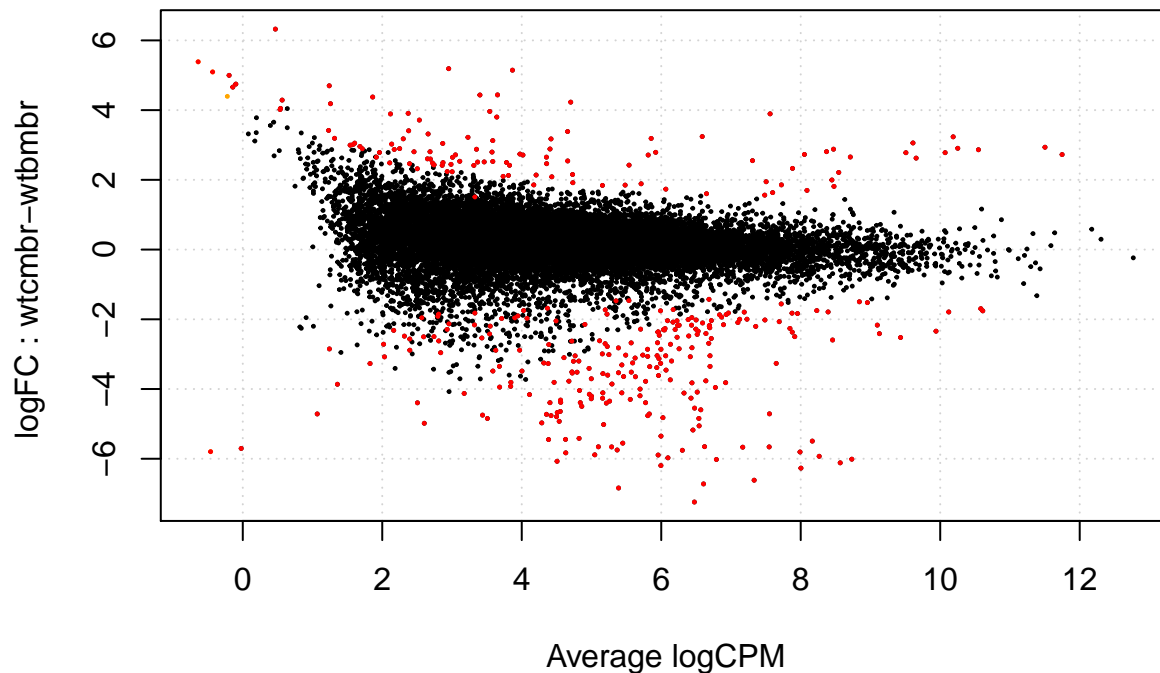
```
sum(results$table$FDR<.05) # How many are DE genes?
```

```
## [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
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
```

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