Read in YAML guide

```r
library(yaml)
yamls <- yaml.load_file("de.yml")
sample1 <- yamls$sample1
sample2 <- yamls$sample2

sample1
```

```
## [1] "tf2cmbr"
```

```r
sample2
```

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

Read in Data

Read in raw count data per gene.

```r
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 expirement

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

```r
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)
```

```
##                     tf2cmbr1.4 tf2cmbr3 tf2cmbr6 tf2cmbr7 wtcmbr10
## Solyc00g005040.2.1          0        6        8        4        0
## Solyc00g005050.2.1          1       34       17       12        5
## Solyc00g005060.1.1          0        1        0        0        1
## Solyc00g005070.1.1         23       11        8        9        5
## Solyc00g005080.1.1         22        7        8       12        0
## Solyc00g005150.1.1          1        3        0        0        0
##                     wtcmbr1.4.6 wtcmbr2 wtcmbr3 wtcmbr7 wtcmbr9
## Solyc00g005040.2.1            9       3       1       0       0
## Solyc00g005050.2.1           38      21      11       4       7
## Solyc00g005060.1.1            3       0       0       1       0
## Solyc00g005070.1.1           12       7       4       6       1
## Solyc00g005080.1.1            7      19      45       4       7
## Solyc00g005150.1.1            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
## tf2cmbr1.4  tf2cmbr   443572            1
## tf2cmbr3    tf2cmbr  1337575            1
## tf2cmbr6    tf2cmbr   790129            1
## tf2cmbr7    tf2cmbr   832907            1
## wtcmbr10     wtcmbr   459717            1
## wtcmbr1.4.6  wtcmbr  1158809            1
## wtcmbr2      wtcmbr  1130695            1
## wtcmbr3      wtcmbr  1560130            1
## wtcmbr7      wtcmbr   374882            1
## wtcmbr9      wtcmbr   386974            1
```

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

```
## Disp = 0.3524 , BCV = 0.5936
```

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

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

```
##                         logFC logCPM  PValue
## Solyc00g005050.2.1 -0.01411  4.243 1.00000
## Solyc00g005070.1.1 -1.53465  4.024 0.01710
## Solyc00g005080.1.1 -0.59400  4.296 0.28619
## Solyc00g005440.1.1  0.38877  4.832 0.53700
## Solyc00g005840.2.1  0.36635  4.835 0.51212
## Solyc00g005880.1.1 -1.50325  3.183 0.03331
```

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

```
## Comparison of groups:  wtcmbr-tf2cmbr
##                      logFC logCPM    PValue       FDR
## Solyc02g023990.2.1 -5.920  6.429 1.417e-18 2.223e-14
## Solyc11g013430.1.1 -7.053  5.063 1.393e-16 1.093e-12
## Solyc01g056770.1.1 -6.343  5.055 1.634e-15 8.418e-12
## Solyc06g069460.1.1 -5.717  5.087 2.146e-15 8.418e-12
## Solyc07g044980.2.1 -4.743  7.744 7.227e-15 2.267e-11
## Solyc01g098190.2.1 -4.638  5.937 8.269e-14 2.162e-10
```

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

```
## [1] 15687     4
```

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

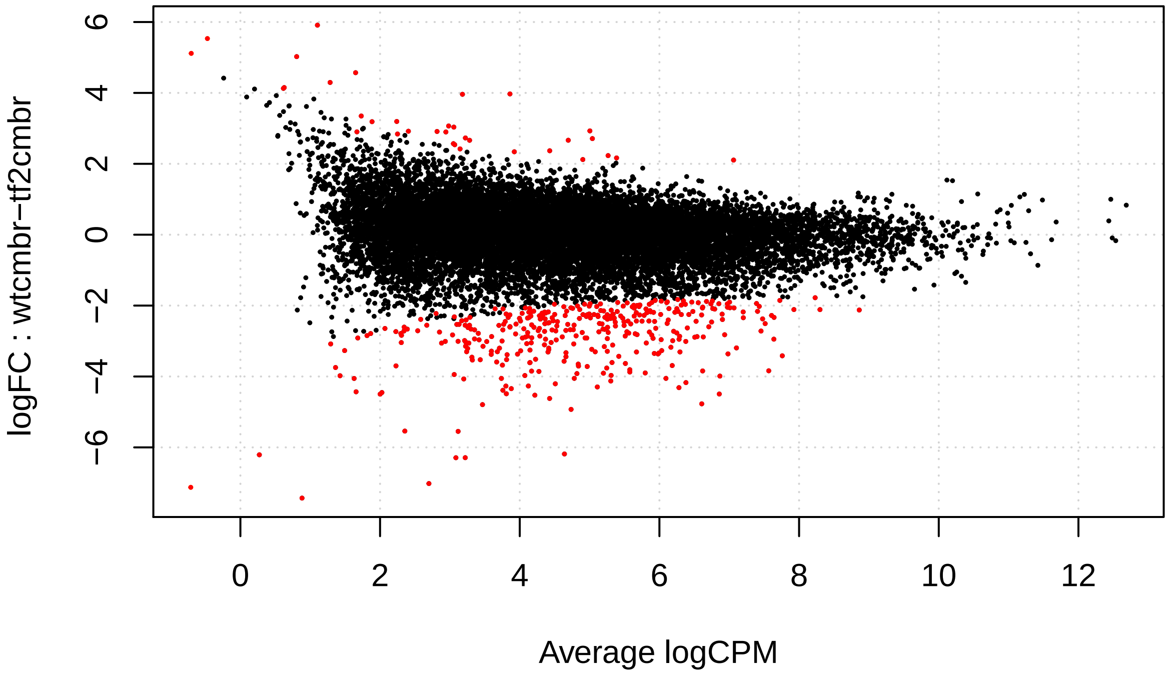
```
## [1] 401
```

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

```
##    [,1]
## -1   367
## 0  15286
## 1     34
```

```
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 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,"DE_all.txt",sep="\t",row.names=F)
write.table(results.sig.annotated,"DE_sig.txt",sep="\t",row.names=F)
```

library(rmarkdown) render(“skeletonDE.Rmd”, “pdf_document”)