Read in YAML guide

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

sample1
```

```
## [1] "wtambr"
```

```
sample2
```

```
## [1] "wtaother"
```

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 expirement

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] "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)
```

```
##                    wtambr2 wtambr4 wtambr5 wtaother1 wtaother5 wtaother6
## Solyc00g005040.2.1       0       2       8         1         1         1
## Solyc00g005050.2.1       0       6       6        17        16         9
## Solyc00g005060.1.1       0       0       1         0         0         0
## Solyc00g005070.1.1      24       3       9         8         6         5
## Solyc00g005080.1.1       9      15      19        18        37         6
## Solyc00g005150.1.1       0       1       2         2         5         0
##                    wtaother7 wtaother8
## Solyc00g005040.2.1         0         2
## Solyc00g005050.2.1         2         3
## Solyc00g005060.1.1         0         2
## Solyc00g005070.1.1         5         6
## Solyc00g005080.1.1        10         7
## Solyc00g005150.1.1         0         2
```

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
## wtambr2     wtambr   395165            1
## wtambr4     wtambr   792542            1
## wtambr5     wtambr   632686            1
## wtaother1 wtaother   929017            1
## wtaother5 wtaother  1555921            1
## wtaother6 wtaother   498294            1
## wtaother7 wtaother   479003            1
## wtaother8 wtaother   510148            1
```

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

```
## Disp = 0.3091 , BCV = 0.556
```

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

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

```
##                        logFC logCPM    PValue
## Solyc00g005050.2.1  0.5863  3.636 3.961e-01
## Solyc00g005070.1.1 -2.4393  4.357 2.113e-04
## Solyc00g005080.1.1 -0.8616  4.558 1.405e-01
## Solyc00g005440.1.1  0.1874  4.544 7.346e-01
## Solyc00g005840.2.1 -0.6429  4.978 2.500e-01
## Solyc00g006470.1.1 -2.9538 11.691 4.068e-08
```

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

```
## Comparison of groups:  wtaother-wtambr
##                       logFC logCPM    PValue       FDR
## Solyc03g062850.1.1 -6.838  6.956 4.894e-23 7.442e-19
## Solyc08g079850.1.1 -5.678  9.228 3.985e-21 2.325e-17
## Solyc06g024350.1.1 -5.724  8.162 4.588e-21 2.325e-17
## Solyc09g091110.2.1 -5.444  7.953 1.039e-19 3.948e-16
## Solyc01g058490.1.1 -6.000  6.360 1.378e-19 4.191e-16
## Solyc07g025190.1.1 -5.725  6.911 1.970e-19 4.992e-16
```

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

```
## [1] 15204     4
```

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

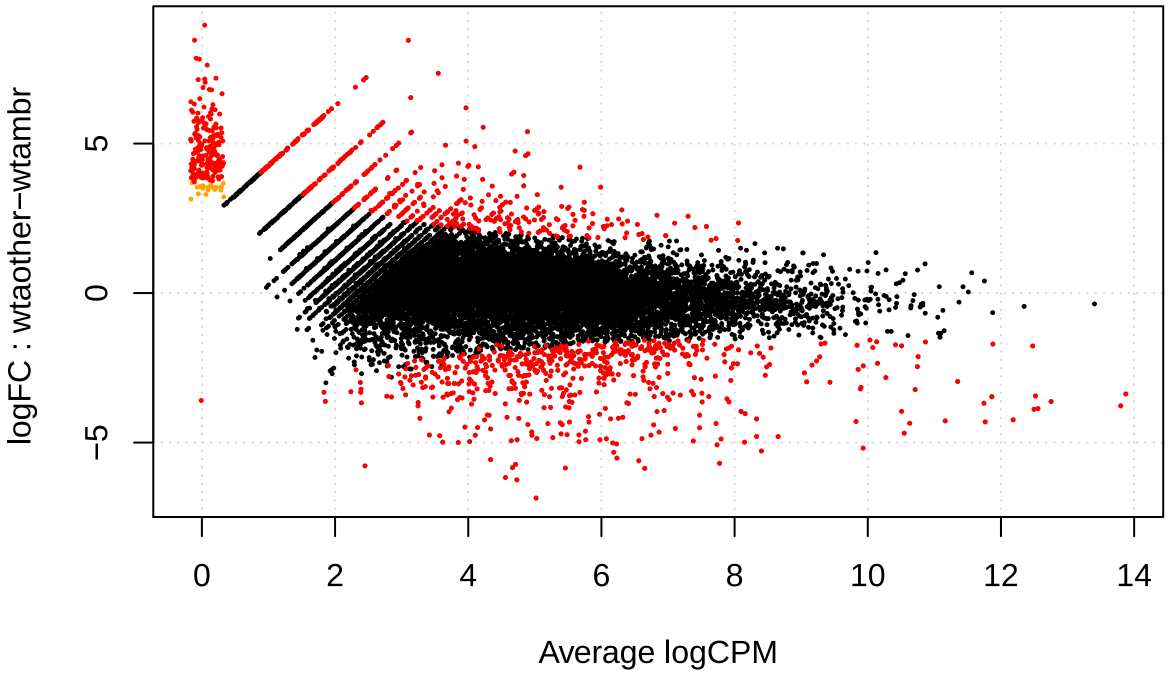
```
## [1] 1251
```

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

```
##    [,1]
## -1   602
## 0  13953
## 1    649
```

```
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”)