Read in YAML guide

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

sample1
```

```
## [1] "wtbmbr"
```

```
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 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)
```

```
##                     wtbmbr2 wtbmbr3 wtbmbr6 wtbmbr8 wtcother1.3.4 wtcother2
## Solyc00g005040.2.1       2       4       3       0             0         0
## Solyc00g005050.2.1      20       5      18       0             2         6
## Solyc00g005060.1.1       1       2       1       1            13         0
## Solyc00g005070.1.1      14       6      12      14           169         6
## Solyc00g005080.1.1      25      15      27       0            11        26
## Solyc00g005150.1.1       0       0       3       0             2         1
##                     wtcother6
## Solyc00g005040.2.1         12
## Solyc00g005050.2.1         37
## Solyc00g005060.1.1          0
## Solyc00g005070.1.1         24
## Solyc00g005080.1.1         35
## Solyc00g005150.1.1          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
## wtbmbr2         wtbmbr  1355352            1
## wtbmbr3         wtbmbr  1213142            1
## wtbmbr6         wtbmbr  1598917            1
## wtbmbr8         wtbmbr    48352            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 = 0.4215 , BCV = 0.6492
```

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

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

```
##                       logFC logCPM   PValue
## Solyc00g005050.2.1 1.04037  4.199 0.174588
## Solyc00g005070.1.1 2.92061  7.636 0.001903
## Solyc00g005080.1.1 1.95787  5.256 0.006475
## Solyc00g005160.1.1 1.82953  3.361 0.030356
## Solyc00g005440.1.1 0.01193  5.152 1.000000
## Solyc00g005840.2.1 0.18517  4.694 0.719440
```

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

```
## Comparison of groups:  wtcother-wtbmbr
##                     logFC logCPM    PValue       FDR
## Solyc01g104030.2.1 6.994  7.899 2.602e-13 3.829e-09
## Solyc04g057980.2.1 5.548  7.044 3.316e-10 2.440e-06
## Solyc10g050210.1.1 5.598  6.547 7.915e-10 3.883e-06
## Solyc12g009110.1.1 5.241  7.064 1.086e-09 3.995e-06
## Solyc09g005140.1.1 4.650  7.543 6.779e-09 1.995e-05
## Solyc10g084320.1.1 5.291  6.017 1.042e-08 2.557e-05
```

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

```
## [1] 14718     4
```

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

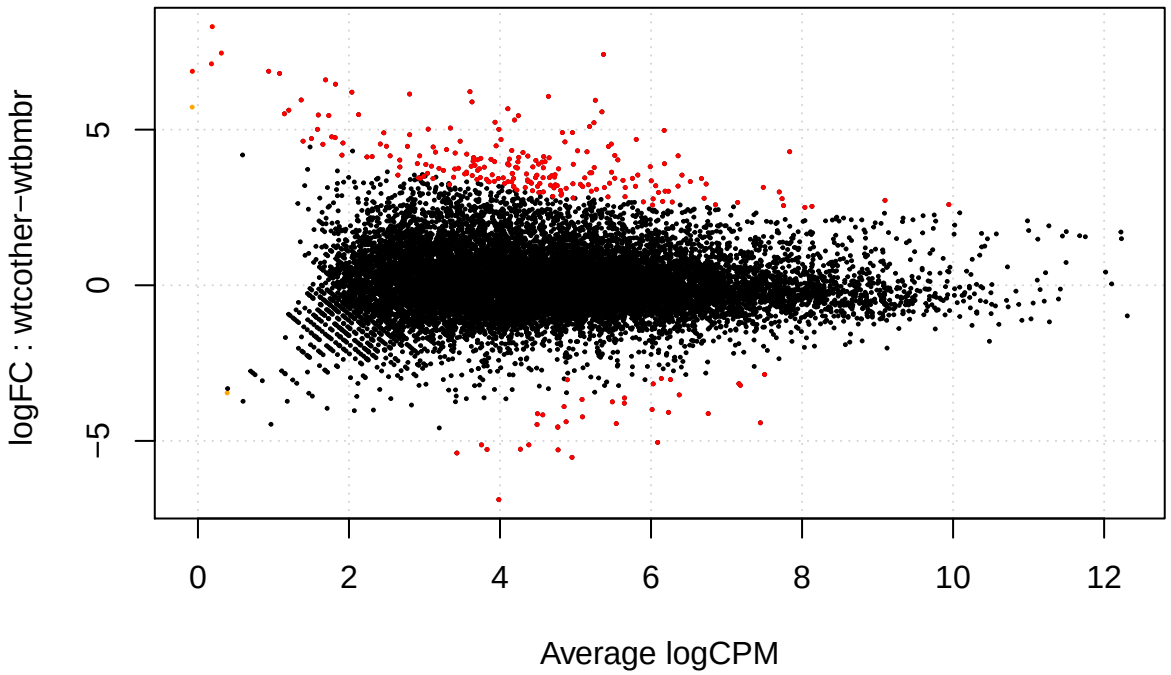
```
## [1] 252
```

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

```
##    [,1]
## -1    34
## 0  14466
## 1    218
```

```
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")