sanalyzing RNAseq for differential expression of LCM data

script modified from a script given to me by Aashish Ranjan called `edgeR_DE.R`

Ciera Martinez

Install

```
source("http://bioconductor.org/biocLite.R") biocLite("edgeR")
```

```
library(edgeR)
```

Read in Data

Read in raw count data per gene. Add checknames to FALSE because it was making the columns unique.

```
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.

WT

Marginal Blastozone vs Other

```
wtcregion <- counts[,40:48]
head(wtcregion)
```

```
##                      wtcmbr10 wtcmbr1.4.6 wtcmbr2 wtcmbr3 wtcmbr7 wtcmbr9
## 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
```

```r
#convert data to a form that edgeR wants
group <- c(rep("wtcmbr", 6), rep("wtcother",3))
d <- DGEList(counts=wtcregion,group=group)
d$samples
```

```
##                  group lib.size norm.factors
## 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
## wtcother1.3.4 wtcother   197345            1
## wtcother2     wtcother   319043            1
## wtcother6     wtcother  1525172            1
```

Computes counts per million (CPM) then, Filter to exclude genes that have <2 counts in (N Rep)-1

```r
cpm.d<- cpm(d)
d <- d[rowSums(cpm.d>2)>=3,]
```

Estimate Common Negative Binomial Dispersion by Conditional Maximum Likelihood. Maximizes the negative binomial conditional common likelihood to give the estimate of the common dispersion across all tags.

```r
d <- estimateCommonDisp(d,verbose=T)
```

```
## Disp = 0.3408 , BCV = 0.5838
```

Normalize library

```r
d <- calcNormFactors(d)
```

Estimate overdispersion Important so that the correct model is fit

```r
d <- estimateCommonDisp(d)
```

Calculate DE genes

```r
DEtest <- exactTest(d,pair=c("wtcmbr","wtcother"))
head(DEtest$table)
```

```
##                     logFC logCPM    PValue
## Solyc00g005040.2.1 0.8537  2.712 5.520e-01
## Solyc00g005050.2.1 0.2292  4.349 7.426e-01
## Solyc00g005060.1.1 3.9245  3.103 3.197e-05
## Solyc00g005070.1.1 5.3610  6.757 6.618e-16
## Solyc00g005080.1.1 1.9088  4.870 2.075e-03
## Solyc00g005150.1.1 1.1149  2.647 2.580e-01
```

Create a table of the results, with multiple testing correction.

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

```
## Comparison of groups:  wtcother-wtcmbr
##                     logFC logCPM    PValue       FDR
## Solyc10g052420.1.1 7.993  9.492 4.739e-30 8.754e-26
## Solyc08g023400.1.1 8.040  8.701 2.454e-29 2.267e-25
## Solyc10g050260.1.1 7.640 10.118 5.369e-29 2.999e-25
## Solyc07g039270.2.1 7.695  9.454 6.494e-29 2.999e-25
## Solyc01g028970.1.1 7.603  9.505 2.268e-28 8.379e-25
## Solyc11g020560.1.1 7.357 11.341 3.482e-28 9.974e-25
```

These are the topTags, but I want to continue with all the DE genes. How many genes are DE?

How many genes in each direction?

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

```
## [1] 18470     4
```

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

```
## [1] 714
```

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

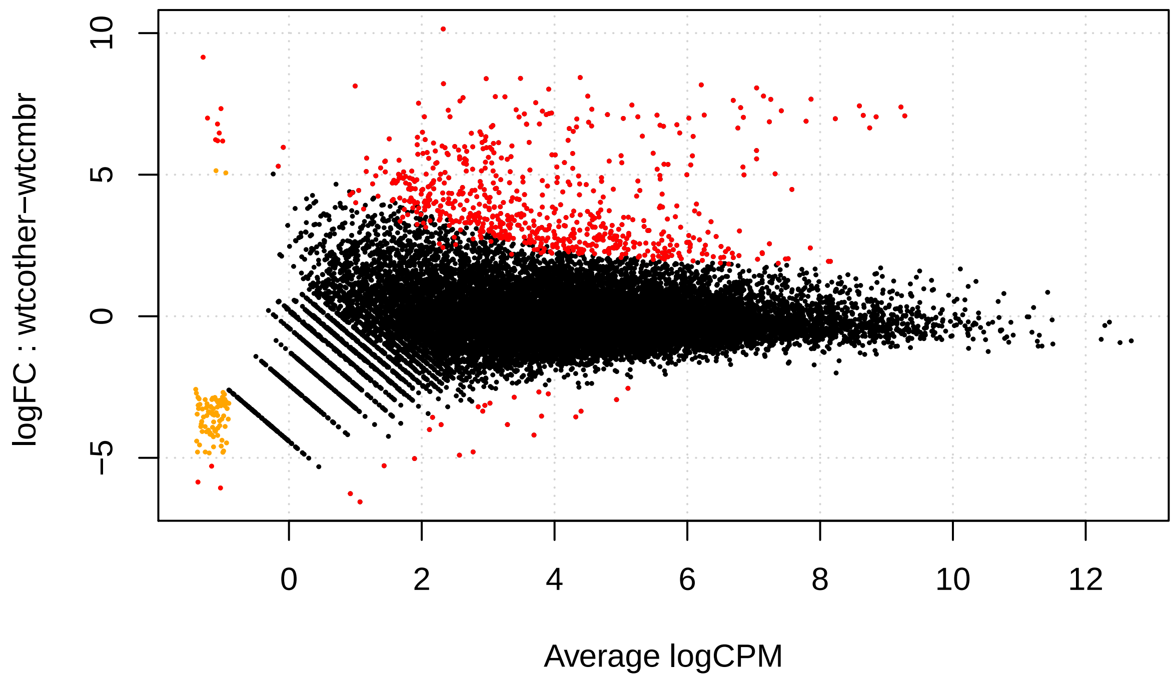
```
##    [,1]
## -1    26
## 0  17756
## 1    688
```

Plot the results First create a table of DE to highlight those with p < 0.05

```
sig.genes <- rownames(results$table[results$table$FDR<0.05,])
```

Visualize with smear plot

```
plotSmear(d,de.tags=sig.genes)
```



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

```
## [1] 18470     4
```

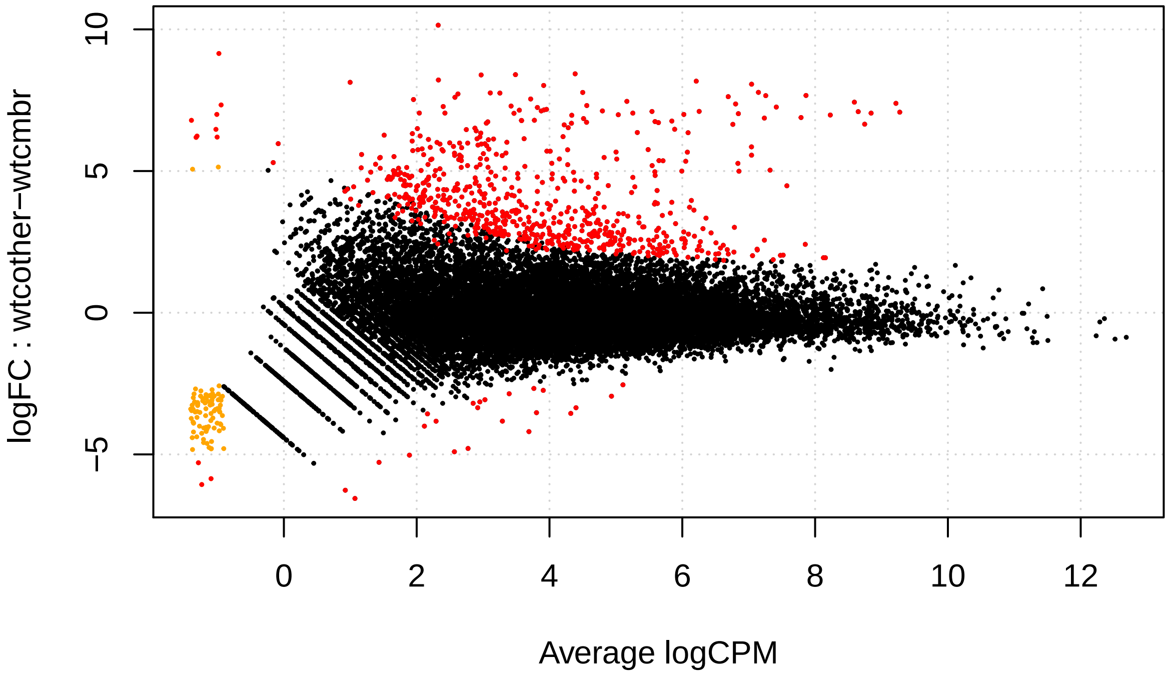
```
sum(results$table$FDR<.05) # How many are DE genes?
```

```
## [1] 714
```

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

```
##     [,1]
## -1     26
## 0   17756
## 1     688
```

```
sig.genes <- rownames(results$table[results$table$FDR<0.05,])
plotSmear(d,de.tags=sig.genes)
```



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") #This s
```

Write table with results

```
write.table(results.all.annotated,"wtcmbr_wtcother_DE_all.txt",sep="\t",row.names=F)
write.table(results.sig.annotated,"wtcmbr_wtcother_DE.txt",sep="\t",row.names=F)
```