Skeleton Key for RNAseq analysis

Written By: Ciera Martinez

About

This is the script used to perform differential gene expression analysis using `edgeR`. The tissue came from p4 leaves of *Solanum lycopersicum* using Laser Capture Microdissection.

Key to Samples

genotype: either wildtype of *tf2*

region: A. tip B. early emmerging leaflet C. base

type: MBR = Marginal Blastozone Region, other = the rachis or midvein region

Run this first chunk before rendering knittedn document

See README.md for more detailed instructions of how to use script

```
library(edgeR)
library(yaml)


### Set Working Directory
rstudioapi::getActiveDocumentContext
```

```
## function ()
## {
##     context <- callFun("getActiveDocumentContext")
##     context$selection <- as.document_selection(context$selection)
##     structure(context, class = "document_context")
## }
## <environment: namespace:rstudioapi>
```

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

### Read in YAML guide
## This reads in the information in the `de.yml` file which has the two names of the samples you are in

yamls <- yaml.load_file("de.yml")

## This part assigns your YMAL to a object in R.  This will be used throughout the script to specify whi

sample1 <- yamls$sample1
sample2 <- yamls$sample2

sample1
```

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

```
sample2
```

```
## [1] "wtcother"
```

To make report

Run the `render()` function below and everything will be run with report at end.

```
library(rmarkdown)
render("skeletonDE_sept2017.Rmd", "pdf_document", output_file = paste(sample1,"_",sample2,"_","DE.pdf",s
```

Samples:

```
print(sample1)
```

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

```
print(sample2)
```

```
## [1] "wtcother"
```

Analysis (Acutal start)

Read in Data

Read in raw count data per gene.

```
counts <- read.delim("../../data/Ciera_coveragebed_counts.txt", row.names = 1)

colnames(counts)
#need to convert NA to 0 counts
counts[is.na(counts)] <- 0

## Get rid of low count libraries "wtbother1.4", "wtbmbr8", "tf2ambr3"
counts <- counts[,-c(36,35,2)]
```

Subset DE expirement

Start by subsetting the particular treatments which are being compared. This might need to be modified depending on the naming of your samples. In my case each sample is named by sample and rep number, so the script is identifying any sample with the sample name given in the `de.yml` file.

```
colnames(counts)
```

```
##  [1] "tf2ambr1"     "tf2ambr4"     "tf2ambr6"     "tf2aother1"
##  [5] "tf2aother2"   "tf2aother4"   "tf2aother7"   "tf2bmbr2"
##  [9] "tf2bmbr5"     "tf2bmbr6"     "tf2bother1"   "tf2bother3"
## [13] "tf2bother4"   "tf2bother6"   "tf2cmbr1.4"   "tf2cmbr3"
## [17] "tf2cmbr6"     "tf2cmbr7"     "tf2cother2"   "tf2cother5"
## [21] "tf2cother6"   "tf2cother7"   "wtambr2"      "wtambr4"
## [25] "wtambr5"      "wtaother1"    "wtaother5"    "wtaother6"
## [29] "wtaother7"    "wtaother8"    "wtbmbr2"      "wtbmbr3"
## [33] "wtbmbr6"      "wtbother3"    "wtbother5"    "wtbother8"
## [37] "wtcmbr1.4.6"  "wtcmbr10"     "wtcmbr2"      "wtcmbr3"
## [41] "wtcmbr7"      "wtcmbr9"      "wtcother1.3.4" "wtcother2"
## [45] "wtcother6"
```

```r
counts1 <- counts[,grep(sample1, colnames(counts), value = TRUE)]
count1Len <- length(colnames(counts1)) #used 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)
```

```
##                 wtcmbr1.4.6 wtcmbr10 wtcmbr2 wtcmbr3 wtcmbr7 wtcmbr9
## Solyc00g069887           7        1      10       4       7       4
## Solyc00g009145          13        1       9      21       1       1
## Solyc00g021530           3        1       2       3       0       2
## Solyc00g023020           1        0       1       1       0       1
## Solyc00g024690           3        1       0       1       0       2
## Solyc00g042147           0        0       1       0       0       0
##                 wtcother1.3.4 wtcother2 wtcother6
## Solyc00g069887              7        26        10
## Solyc00g009145           1488        37        23
## Solyc00g021530              5         2         0
## Solyc00g023020              8         0         4
## Solyc00g024690             11         0         1
## Solyc00g042147              8         1         0
```

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.

```r
group <- c(rep(sample1, count1Len), rep(sample2, count2Len))
d <- DGEList(counts = counts, group = group)
```

Check to see if the group column matches your sample name and they are appropriate.

```r
d$samples
```

```
##                  group lib.size norm.factors
## wtcmbr1.4.6     wtcmbr  6733959            1
## wtcmbr10        wtcmbr  2342026            1
## wtcmbr2         wtcmbr  6640162            1
## wtcmbr3         wtcmbr  9131769            1
## wtcmbr7         wtcmbr  2168431            1
## wtcmbr9         wtcmbr  2269177            1
## wtcother1.3.4 wtcother  2321532            1
## wtcother2     wtcother  2766600            1
## wtcother6     wtcother  7721503            1
```

Differential expression using edgeR

Make sure there is full understanding of each edgeR command being used. The manual is amazing so read it *before* running the DE analysis below edgeR manual. There are many options and they must be set to be appropriate for your analysis.

```
cpm.d <- cpm(d) #counts per million
d <- d[rowSums(cpm.d > 5) >= 3,] #This might be a line to adjust. It is removing genes with low counts.
d <- estimateCommonDisp(d,verbose = T)
```

```
## Disp = 0.18909 , BCV = 0.4348
```

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

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

```
##                        logFC   logCPM       PValue
## Solyc00g107055  0.01518736 5.499731 9.502433e-01
## Solyc00g143770  0.28248583 2.891572 4.821981e-01
## Solyc00g183555  6.34035010 7.720783 5.454238e-44
## Solyc00g212265  5.49840038 7.776976 7.906618e-36
## Solyc00g005050 -0.07287431 3.632599 8.828720e-01
## Solyc00g005092  1.52603034 5.712538 1.934824e-04
```

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

dim(results$table)
```

```
## [1] 23427     4
```

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

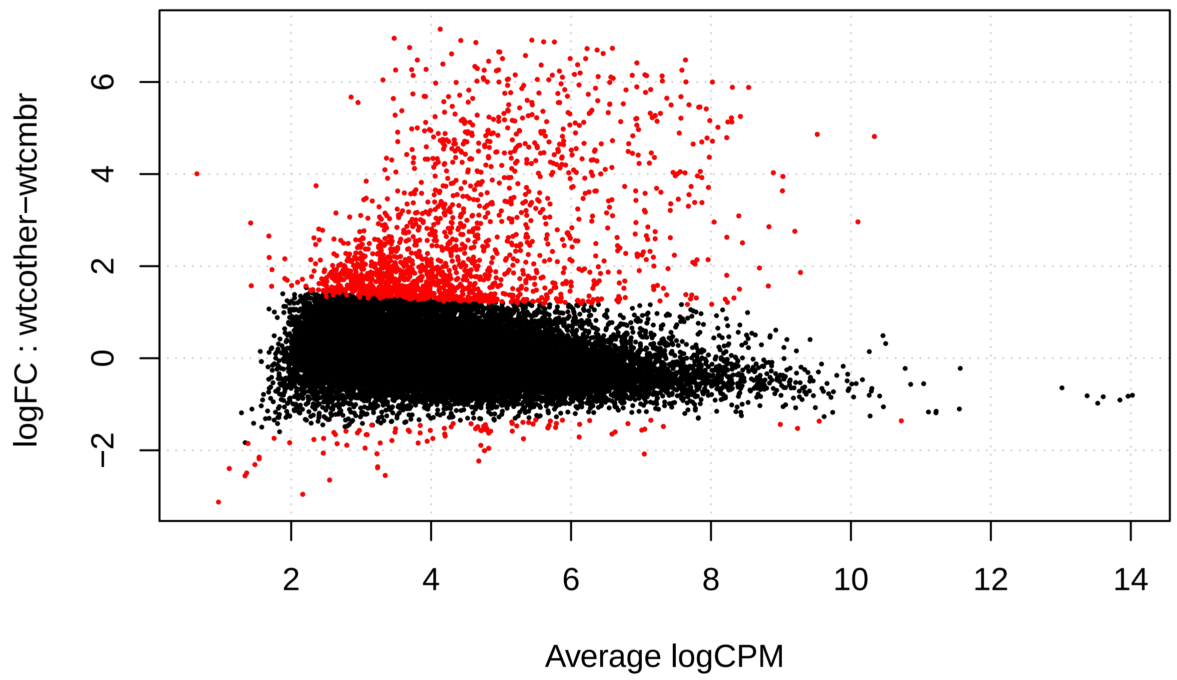
```
## [1] 1816
```

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

```
##    [,1]
## -1    94
## 0  21611
## 1   1722
```

```
sig.genes <- rownames(results$table[results$table$FDR < 0.05,]) # outputs just significant gene names

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



Subset all the genes with a significant FDR score less than .05.

```
results.sig <- subset(results$table, results$table$FDR < 0.05)
dim(results.sig)
```

```
## [1] 1816    4
```

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("../../data/ITAG2.3_all_Arabidopsis_ITAG_annotations.tsv", header = FALSE)  #
colnames(annotation1) <- c("ITAG", "SGN_annotation")
annotation2 <- read.delim("../../data/ITAG2.3_all_Arabidopsis_annotated.tsv")
annotation <- merge(annotation1,annotation2, by = "ITAG")

## Remove the trailing
annotation$ITAG <- gsub("^(.*)[.].*", "\\1", annotation$ITAG)
annotation$ITAG <- gsub("^(.*)[.].*", "\\1", annotation$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", all.x = TRUE) #This is merging only

#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.names = F)
write.table(results.sig.annotated, file = paste(sample1, "_", sample2, "_", "DE_sig.txt", sep = ""),
            sep = "\t", row.names = F)
```

Now run the script below for a full **knitr** report of what was run and leave this report in the folder that the analysis was done with output files.