RNAseq example 2 reps

Peter Thorpe

13/08/2021

0.1 The is a R markdown document

These are cool as you can make a nice .pdf when you are finished. The run the code, highlight the line of interest and press enter.

Load the library needed

```
library(edgeR)
library(knitr)
```

```
Warning: package 'knitr' was built under R version 3.6.3
```

```
# if not installed install.packages('BiocManager')
# BiocManager::install('edgeR')
```

Further information for edgeR can be found here.

Load the data

counts were already generated using salmon and counts.matrix generated using trinity.

```
setwd("C:/Users/pjt6/Desktop/RNAseq_lecture_workshop/DE_gene/two_bio_reps")


# check it
getwd()
```

```
[1] "C:/Users/pjt6/Desktop/RNAseq_lecture_workshop/DE_gene/two_bio_reps"
```

The counts data in contained in the counts.matrix, each gene has a digital count per condition/ rep

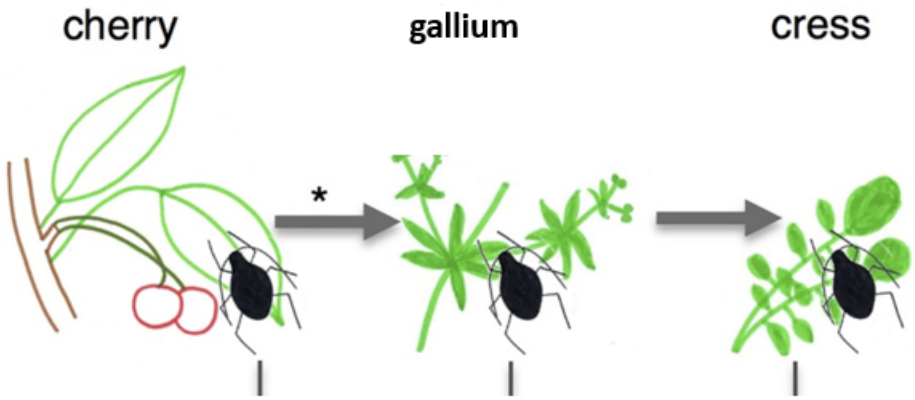
```
# see what is in the directory
dir()
```

```
 [1] "DE_2reps.Rmd"
 [2] "DE_2reps_no-PCA.Rmd"
 [3] "DE_2reps_no PCA.Rmd"
 [4] "DE_gene_R_commands.sh"
 [5] "example.matrix.TMM_info.txt"
 [6] "functions_not_used"
 [7] "M.cerasi_cherry_vs_gallium.edgeR.count_matrix"
 [8] "M.cerasi_cherry_vs_gallium.edgeR.DE_results.zip"
 [9] "M.cerasi_cherry_vs_gallium.GLM.edgeR.count_matrix"
[10] "M.cerasi_cherry_vs_gallium.GLM.edgeR.DE_results"
[11] "Replicas Image.png"
[12] "samples_described.txt"
[13] "TableOfCounts.txt"
```

```
# load in the data
data <- read.delim("TableOfCounts.txt", header = T, row.names = 1)

# group the replicas
group <- factor(c(1, 1, 2, 2))

# Include image replica
include_graphics("Replicas Image.png")
```




```
fig.cap = paste("Figure 1")
```

have a quick look at the data:

```
head(data)
```

```
         cherry1 cherry2 gallium1 gallium2
Mca00001    5945    8854    10377    13522
Mca00002     364     644      746      911
Mca00003      14      32       17       25
Mca00004    1504    2022     1658     1902
Mca00005       0       0        1        2
Mca00006      10       7       11       13
```

```
# store the data in a list-based object.
rnaseqMatrix <- DGEList(counts = data, group = group)
```

```
# have a little look at the data
head(rnaseqMatrix)
```

```
An object of class "DGEList"
$counts
         cherry1 cherry2 gallium1 gallium2
Mca00001    5945    8854    10377    13522
Mca00002     364     644      746      911
Mca00003      14      32       17       25
Mca00004    1504    2022     1658     1902
Mca00005       0       0        1        2
Mca00006      10       7       11       13

$samples
         group lib.size norm.factors
cherry1      1 22599843            1
cherry2      1 29376912            1
gallium1     2 22414071            1
gallium2     2 27296089            1
```

filter very low expression genes as these do not contribute and negatively affect the stats

```
keep <- filterByExpr(rnaseqMatrix)

rnaseqMatrix <- rnaseqMatrix[keep, , keep.lib.sizes = FALSE]

table(keep)
```

```
keep
FALSE  TRUE
15564 13124
```

to account for sequencing depth, calcNormFactors finds a set of scaling facotrs for lib sizes. this minimised the log fold change between samples for most genes Note this is not FRPM or TPM normalisation, raw values need to be given to EdgeR, as these are needed to estimate the mean-variance relationship between the samples

```
rnaseqMatrix <- calcNormFactors(rnaseqMatrix)
```

write a table of the lib size and normalisation factors. Look at how these are different.

```
rnaseqMatrix$samples$eff.lib.size = rnaseqMatrix$samples$lib.size * rnaseqMatrix$samples$norm.factors
write.table(rnaseqMatrix$samples, file = "example.matrix.TMM_info.txt", quote = F,
    sep = "\t", row.names = F)
```

```
# have a look
rnaseqMatrix$samples
```

```
         group lib.size norm.factors eff.lib.size
cherry1      1 22591244    0.8379833     18931084
cherry2      1 29362976    0.9006571     26445974
gallium1     2 22398744    1.1617033     26020695
gallium2     2 27276133    1.1405385     31109479
```

```
# group are your samples
design <- model.matrix(~group)
```

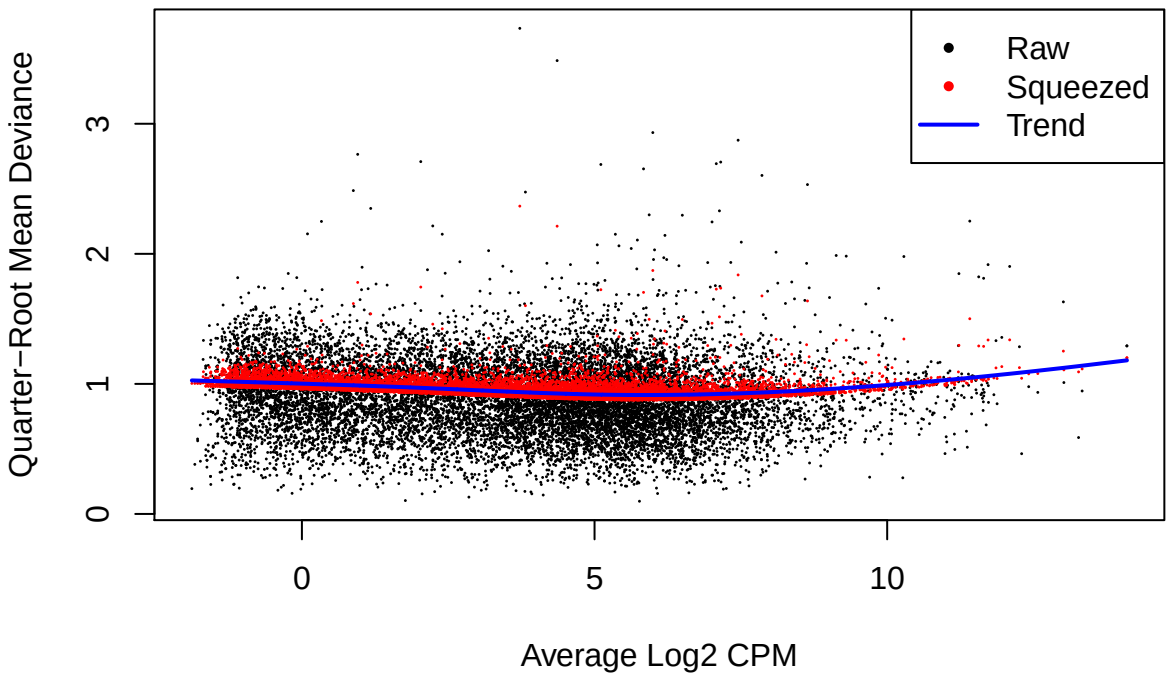
estimate the dispertion

```
rnaseqMatrix <- estimateDisp(rnaseqMatrix, design)
```

run the DE analysis:

```
# To perform quasi-likelihood F-tests: (better for low numbers of reps)
fit <- glmQLFit(rnaseqMatrix, design)

plotQLDisp(fit)
```




```
# value 0.01 is good for DE analysis.
rnaseqMatrix$common.dispersion
```

```
[1] 0.01303526
```

```
# have a little look
fit
```

```
An object of class "DGEGLM"
$coefficients
         (Intercept)      group2
Mca00001   -8.033403  0.25036907
Mca00002  -10.730566  0.28181235
Mca00003  -13.819376 -0.30229678
Mca00004   -9.459581 -0.22195319
Mca00006  -14.763717  0.09269888
13119 more rows ...

$fitted.values
             cherry1    cherry2    gallium1    gallium2
Mca00001 6141.952180 8580.06367 10843.89211 12964.59711
Mca00002  413.859946  578.14594   754.06693   901.53736
Mca00003   18.766578   26.21617    19.03154    22.75348
Mca00004 1475.382107 2061.05031  1624.23221  1941.87806
Mca00006    7.241005   10.11540    10.93361    13.07186
```

```
13119 more rows ...

$deviance
 Mca00001  Mca00002  Mca00003  Mca00004  Mca00006
0.5446221 2.2824817 1.2739681 0.1939206 1.2835792
13119 more elements ...

$method
[1] "oneway"

$counts
         cherry1 cherry2 gallium1 gallium2
Mca00001    5945    8854    10377    13522
Mca00002     364     644      746      911
Mca00003      14      32       17       25
Mca00004    1504    2022     1658     1902
Mca00006      10       7       11       13
13119 more rows ...

$unshrunk.coefficients
         (Intercept)      group2
Mca00001   -8.033418  0.25037240
Mca00002  -10.730788  0.28186656
Mca00003  -13.824238 -0.30406685
Mca00004   -9.459644 -0.22196874
Mca00006  -14.776556  0.09399462
13119 more rows ...

$df.residual
[1] 2 2 2 2 2
13119 more elements ...

$design
  (Intercept) group2
1           1      0
2           1      0
3           1      1
4           1      1
attr(,"assign")
[1] 0 1
attr(,"contrasts")
attr(,"contrasts")$group
[1] "contr.treatment"


$offset
         [,1]     [,2]    [,3]     [,4]
[1,] 16.75632 17.09061 17.0744 17.25302
attr(,"class")
[1] "CompressedMatrix"
attr(,"Dims")
[1] 5 4
attr(,"repeat.row")
[1] TRUE
```

```
attr(,"repeat.col")
[1] FALSE
13119 more rows ...

$dispersion
[1] 0.010462028 0.010174595 0.061957458 0.007621599 0.066620557
13119 more elements ...

$prior.count
[1] 0.125

$AveLogCPM
[1]  8.53415005  4.67441852 -0.09639445  6.13380718 -1.06223789
13119 more elements ...

$df.residual.zeros
[1] 2 2 2 2 2
13119 more elements ...

$df.prior
[1] 10.64767

$var.post
 Mca00001  Mca00002  Mca00003  Mca00004  Mca00006
0.7233471 0.7884051 0.9532540 0.6049958 0.9988231
13119 more elements ...

$var.prior
 Mca00001  Mca00002  Mca00003  Mca00004  Mca00006
0.8080673 0.7221304 1.0126604 0.7004224 1.0658864
13119 more elements ...

$samples
         group lib.size norm.factors eff.lib.size
cherry1      1 22591244    0.8379833     18931084
cherry2      1 29362976    0.9006571     26445974
gallium1     2 22398744    1.1617033     26020695
gallium2     2 27276133    1.1405385     31109479
```

```
qlf <- glmQLFTest(fit, coef = 2)

topTags(qlf)
```

```
Coefficient:  group2
             logFC   logCPM        F       PValue          FDR
Mca25862 -9.582837 7.090278 2171.891 1.566119e-15 2.055374e-11
Mca13168 -6.045580 5.529698 1676.339 7.974973e-15 5.100619e-11
Mca03967 -9.213173 5.019073 1577.936 1.165945e-14 5.100619e-11
Mca22824 -4.705682 6.250036 1414.574 2.314630e-14 5.347249e-11
Mca28507 11.325421 5.127184 1409.723 2.365038e-14 5.347249e-11
Mca18026 -6.605461 5.066950 1381.380 2.686279e-14 5.347249e-11
Mca24560  7.561758 5.416823 1368.247 2.852083e-14 5.347249e-11
Mca13431 -4.383117 6.175406 1290.353 4.118395e-14 6.756227e-11
Mca13432 -4.496860 6.626225 1220.102 5.848519e-14 8.528440e-11
```

```
Mca17238  7.395542 5.024287 1130.127 9.448081e-14 1.165511e-10
```

```
tTags = topTags(qlf, n = NULL)

result_table = tTags$table
```

write the results to files

```
result_table = data.frame(sampleA = "cherry", sampleB = "gallium", result_table)

result_table <- result_table[order(result_table$logFC), ]

write.table(result_table, file = "M.cerasi_cherry_vs_gallium.GLM.edgeR.DE_results",
    sep = "\t", quote = F, row.names = T)

write.table(rnaseqMatrix, file = "M.cerasi_cherry_vs_gallium.GLM.edgeR.count_matrix",
    sep = "\t", quote = F, row.names = T)
```

plot a PCA

‘‘‘