HLphyDis3 DE gene analysis: reannotated Ewan and Paolo 01 Oct 2021, A_vs_B_GLM_with_animal_as_batch drop KI rep 3

Peter Thorpe

01 Oct 2021

#0.1 QC!

see Word document. There is a collection of graphs from multiple sources, how the counts were created

##load the libs needed

```
library(tidyverse)
```

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

-- Attaching packages --------------------------------------- tidyverse 1.3.1 --

v ggplot2 3.3.3     v purrr   0.3.4
v tibble  3.1.0     v dplyr   1.0.6
v tidyr   1.1.3     v stringr 1.4.0
v readr   1.4.0     v forcats 0.5.1

Warning: package 'ggplot2' was built under R version 3.6.3

Warning: package 'tibble' was built under R version 3.6.3

Warning: package 'tidyr' was built under R version 3.6.3

Warning: package 'readr' was built under R version 3.6.3

Warning: package 'purrr' was built under R version 3.6.3

Warning: package 'dplyr' was built under R version 3.6.3

Warning: package 'forcats' was built under R version 3.6.3

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```r
library(DESeq2) # not used, yet(?)
```

```
Loading required package: S4Vectors

Warning: package 'S4Vectors' was built under R version 3.6.3

Loading required package: stats4

Loading required package: BiocGenerics

Loading required package: parallel


Attaching package: 'BiocGenerics'

The following objects are masked from 'package:parallel':

    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply, parCapply, parLapply,
    parLapplyLB, parRapply, parSapply, parSapplyLB

The following objects are masked from 'package:dplyr':

    combine, intersect, setdiff, union

The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

    anyDuplicated, append, as.data.frame, basename, cbind, colnames,
    dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
    grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
    union, unique, unsplit, which, which.max, which.min


Attaching package: 'S4Vectors'

The following objects are masked from 'package:dplyr':

    first, rename

The following object is masked from 'package:tidyr':

    expand
```

```
The following object is masked from 'package:base':

    expand.grid

Loading required package: IRanges


Attaching package: 'IRanges'

The following objects are masked from 'package:dplyr':

    collapse, desc, slice

The following object is masked from 'package:purrr':

    reduce

The following object is masked from 'package:grDevices':

    windows

Loading required package: GenomicRanges

Loading required package: GenomeInfoDb

Warning: package 'GenomeInfoDb' was built under R version 3.6.3

Loading required package: SummarizedExperiment

Loading required package: Biobase

Welcome to Bioconductor

    Vignettes contain introductory material; view with
    'browseVignettes()'. To cite Bioconductor, see
    'citation("Biobase")', and for packages 'citation("pkgname")'.

Loading required package: DelayedArray

Warning: package 'DelayedArray' was built under R version 3.6.3

Loading required package: matrixStats

Warning: package 'matrixStats' was built under R version 3.6.3


Attaching package: 'matrixStats'

The following objects are masked from 'package:Biobase':

    anyMissing, rowMedians
```

```
The following object is masked from 'package:dplyr':

    count

Loading required package: BiocParallel


Attaching package: 'DelayedArray'

The following objects are masked from 'package:matrixStats':

    colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

The following object is masked from 'package:purrr':

    simplify

The following objects are masked from 'package:base':

    aperm, apply, rowsum
```

```
library(plotly)
```

```
Attaching package: 'plotly'

The following object is masked from 'package:IRanges':

    slice

The following object is masked from 'package:S4Vectors':

    rename

The following object is masked from 'package:ggplot2':

    last_plot

The following object is masked from 'package:stats':

    filter

The following object is masked from 'package:graphics':

    layout
```

```
library(ggplot2)
library(tidyverse)
library(ggrepel)
```

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

```
library(htmlwidgets)
```

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

```
library(edgeR)
```

```
Loading required package: limma
```

```

Attaching package: 'limma'
```

```
The following object is masked from 'package:DESeq2':

    plotMA
```

```
The following object is masked from 'package:BiocGenerics':

    plotMA
```

```
library(tibble)
library(data.table)
```

```
Warning: package 'data.table' was built under R version 3.6.3
```

```

Attaching package: 'data.table'
```

```
The following object is masked from 'package:SummarizedExperiment':

    shift
```

```
The following object is masked from 'package:GenomicRanges':

    shift
```

```
The following object is masked from 'package:IRanges':

    shift
```

```
The following objects are masked from 'package:S4Vectors':

    first, second
```

```
The following objects are masked from 'package:dplyr':

    between, first, last
```

```
The following object is masked from 'package:purrr':

    transpose
```

#0.2 load the data

NOTE: replicate 2 removed for GFP_Striatum and KI_Striatum - previously shown not to worked.

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

```
setwd("C:/Users/pjt6/Documents/bat_RNAseq/updated_annotation_01oct2021/A_vs_B_GLM_with_animal_as_batch/c

cts <- read.delim("HLphyDis3.GFP_Striatum_vs_KI_Striatum.count_matrix",
                  header=T, row.names=1)

coldata = read.table("metadata_2.txt", header=T, com='',
                     sep="\t", check.names=F, row.names=1)
```

have a quick look at the data:

```
head(cts)
```

```
           GFP_Striatum_rep1 GFP_Striatum_rep3 GFP_Striatum_rep4
EPB41L3              2217.782          2215.000          2184.561
LOC04577               11.000            18.000            17.000
GEMIN2                 55.000            39.000            53.000
GLUL                15770.805         22588.040         23652.590
AC114490.3              0.000             0.000             0.000
COCH                 1248.000          1062.148           788.000
           KI_Striatum_rep1 KI_Striatum_rep4
EPB41L3             2291.00         2445.004
LOC04577              15.00           16.000
GEMIN2                69.00           59.000
GLUL                20208.57        26049.849
AC114490.3              0.00            0.000
COCH                  815.00          605.000
```

```
head(coldata)
```

```
                    condition animal  side Age_of_bat              reps
GFP_Striatum_rep1 GFP_Striatum      1 right          8 GFP_Striatum_rep1
GFP_Striatum_rep3 GFP_Striatum      3 right          3 GFP_Striatum_rep3
GFP_Striatum_rep4 GFP_Striatum      4 right          3 GFP_Striatum_rep4
KI_Striatum_rep1   KI_Striatum      1  left          8  KI_Striatum_rep1
KI_Striatum_rep4   KI_Striatum      4  left          3  KI_Striatum_rep4
```

convert to factors

```
coldata <- coldata[,c("condition", "animal", "side","Age_of_bat", "reps")]
coldata$condition <- factor(coldata$condition)
coldata$animal <- factor(coldata$animal)
coldata$side <- factor(coldata$side)
coldata$Age_of_bat <- factor(coldata$Age_of_bat)
coldata$reps <- factor(coldata$reps)
animal <- factor(coldata$animal)
condition <- factor(coldata$condition)
```

check that the order of the table and the colomns in the counts.matrix march, if not, then fix:

```
all(rownames(coldata) %in% colnames(cts))
```

```
[1] TRUE
```

```
# TRUE

rownames(coldata) <-  rownames(coldata)

all(rownames(coldata) == colnames(cts))
```

```
[1] TRUE
```

```
# FALSE

cts <- cts[, rownames(coldata)]

all(rownames(coldata) == colnames(cts))
```

```
[1] TRUE
```

```
# TRUE
cts <- cts[, rownames(coldata)]
all(rownames(coldata) == colnames(cts))
```

```
[1] TRUE
```

explicitly set up the replicates order. This matches the order in the counts.matrix and in the sample.table

```
# this is not actually used
replicates = c(1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 1)
```

explicitly set up the group (yes, this is contained in the table!)

```
group=c('GFP_Striatum', 'GFP_Striatum', 'GFP_Striatum',
        'KI_Striatum', 'KI_Striatum')
```

#0.3 R Packages now to start using edgeR to do some analysis: https://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeR.pdf

```
y <- DGEList(counts=cts, group=group)

condition<-coldata$condition # just so I dont have to keep using dollars :)
```

filter low expressing genes. Stats perfomred in a larger number, when lots are zero to low expressing negatively impacts on the data. 5,640 genes dropped from the analysis.

```
keep <- filterByExpr(y)

table(keep)
```

```
keep
FALSE  TRUE
 5998 14883
```

calculate nomrlaisation factors and normalise the data for lib depth differences

```
y <- y[keep, , keep.lib.sizes=FALSE]
#The TMM normalization is applied to account for the compositional biases:
y <- calcNormFactors(y)

y$samples
```

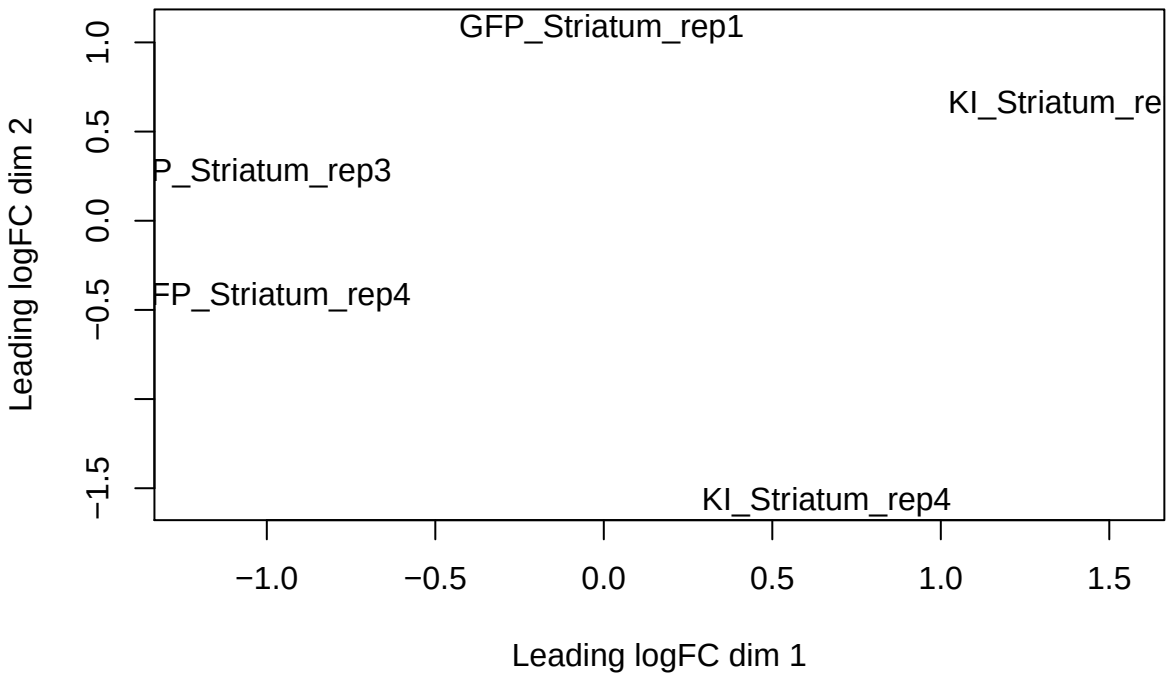
```
                        group lib.size norm.factors
GFP_Striatum_rep1 GFP_Striatum  7906193    0.9912429
GFP_Striatum_rep3 GFP_Striatum  8136418    0.9897226
GFP_Striatum_rep4 GFP_Striatum  8251404    1.0000408
KI_Striatum_rep1   KI_Striatum  7812055    1.0025940
KI_Striatum_rep4   KI_Striatum  8731462    1.0166316
```

following this https://support.bioconductor.org/p/56637/ with a problem with y, see the fix at the webpage. Reasign to d:

```
d <- DGEList(counts=y,group=group)

keep <- filterByExpr(d)

table(keep)
```

```
keep
 TRUE
14883
```

```
d <- d[keep, , keep.lib.sizes=FALSE]

d <- calcNormFactors(d)

plotMDS(d)
```



Before we fit GLMs, we need to define our design matrix based on the experimental design. We want to test for differential expressions between our conditions within batches, i.e. adjusting for differences between batches. In statistical terms, this is an additive linear model. So the design matrix is created as:

```
design <- model.matrix(~0 + condition + animal)
```

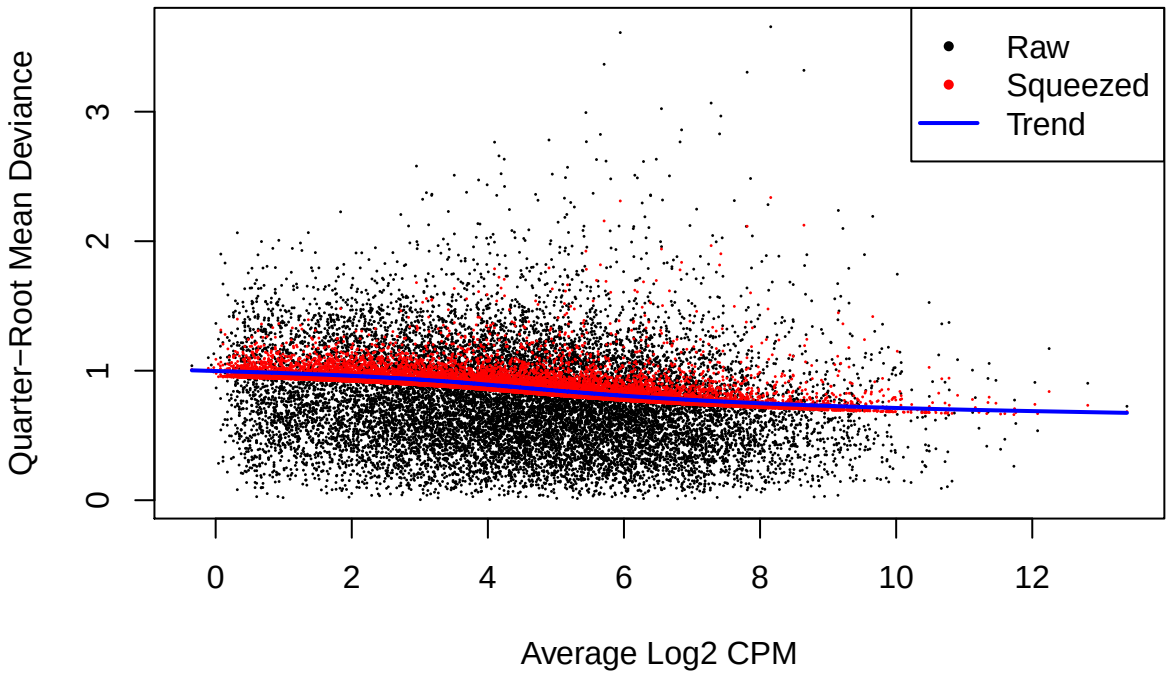
```
rownames(design) <- colnames(d)
```

```
design
```

```
                  conditionGFP_Striatum conditionKI_Striatum animal3 animal4
GFP_Striatum_rep1                     1                     0       0       0
GFP_Striatum_rep3                     1                     0       1       0
GFP_Striatum_rep4                     1                     0       0       1
KI_Striatum_rep1                      0                     1       0       0
KI_Striatum_rep4                      0                     1       0       1
attr(,"assign")
[1] 1 1 2 2
attr(,"contrasts")
attr(,"contrasts")$condition
[1] "contr.treatment"

attr(,"contrasts")$animal
[1] "contr.treatment"
```

run the DE analysis:

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

# this GLM is better for low numbers of reps.
 fit <- glmQLFit(d, design)


 plotQLDisp(fit)
```



#0.4 specific comparisons To do the specific comparisons: GFP_Cortex_vs_GFP_Striatum. Coefficient: (Intercept) groupGFP_Cortex groupGFP_Striatum groupKI_auditory_Cortex groupKI_Cortex groupKI_Striatum

```
my.contrasts <- makeContrasts(GFP_striatum_vs_KI_striatum = conditionGFP_Striatum - conditionKI_Striatu
                              levels=design)
```

get the pair wise comparisons.

GFP_striatum_vs_KI_striatum

```
# GFP_striatum_vs_KI_striatum

qlf <- glmQLFTest(fit, contrast=my.contrasts[,"GFP_striatum_vs_KI_striatum"])

tTags = topTags(qlf,n=NULL)

result_table = tTags$table
```

```
result_table = data.frame(sampleA="GFP_striatum", sampleB="KI_striatum",
                          result_table)
result_table = merge(result_table, cts,by="row.names",all.x=TRUE)
result_table <- result_table[order(result_table$logFC),]


write.table(result_table, file='GFP_striatum_vs_KI_striatum.GLM.edgeR.DE_results',
            sep='\t', quote=F, row.names=T, col.names = NA)
```