limma_voom

jrm

11/28/2018

Data

Data was obtained from Julin Maloof's BIS180L course: http://jnmaloof.github.io/BIS180L_web/2018/05/22/RNAseq-edgeR/

A description of the experiment:

> "We will study gene expression levels in Brassica rapa internodes grown under two treatments, Dense Planting (DP) and Not Dense Planting (NDP). We will study the response to DP in two cultivars, IMB211 and R500. Click to download the internode count data. This data set has 12 samples with counts of 40991 genes."

```
library(edgeR)
```

```
## Loading required package: limma
```

```
library(limma)
library(Glimma)
```

Data preprocessing

```
rawCounts <- read.delim("gh_internode_counts2.tsv",
                        header = T, # Table has a header (column names)
                        row.names = 1) # Treat the first column as the row names
head(rawCounts)
```

```
##           IMB211_DP_1_INTERNODE.1_matched.merged.fq.bam
## *                                                805530
## Bra000001                                             1
## Bra000002                                            20
## Bra000003                                           228
## Bra000004                                             8
## Bra000005                                           328
##           IMB211_DP_2_INTERNODE.1_matched.merged.fq.bam
## *                                                245831
## Bra000001                                            NA
## Bra000002                                            25
## Bra000003                                            73
## Bra000004                                             2
## Bra000005                                            92
##           IMB211_DP_3_INTERNODE.1_matched.merged.fq.bam
## *                                                740348
## Bra000001                                            NA
## Bra000002                                            36
## Bra000003                                           184
## Bra000004                                             7
## Bra000005                                           284
##           IMB211_NDP_1_INTERNODE.1_matched.merged.fq.bam
## *                                                 574103
## Bra000001                                              2
## Bra000002                                             39
## Bra000003                                            149
## Bra000004                                              2
## Bra000005                                            300
##           IMB211_NDP_2_INTERNODE.1_matched.merged.fq.bam
## *                                                 444154
## Bra000001                                             NA
## Bra000002                                             60
## Bra000003                                            168
## Bra000004                                              3
## Bra000005                                            206
##           IMB211_NDP_3_INTERNODE.1_matched.merged.fq.bam
## *                                                 628348
## Bra000001                                             NA
## Bra000002                                             34
## Bra000003                                            168
## Bra000004                                              3
## Bra000005                                            305
##           R500_DP_1_INTERNODE.1_matched.merged.fq.bam
## *                                              757382
## Bra000001                                           1
## Bra000002                                          NA
## Bra000003                                         232
## Bra000004                                           3
## Bra000005                                         384
##           R500_DP_2_INTERNODE.1_matched.merged.fq.bam
## *                                              845280
## Bra000001                                          NA
## Bra000002                                          10
## Bra000003                                         221
## Bra000004                                           4
## Bra000005                                         655
##           R500_DP_3_INTERNODE.1_matched.merged.fq.bam
## *                                              774751
## Bra000001                                           1
## Bra000002                                           5
## Bra000003                                         182
## Bra000004                                           3
## Bra000005                                         404
##           R500_NDP_1_INTERNODE.1_matched.merged.fq.bam
## *                                               453083
## Bra000001                                           NA
## Bra000002                                            2
## Bra000003                                          160
## Bra000004                                            3
## Bra000005                                          384
##           R500_NDP_2_INTERNODE.1_matched.merged.fq.bam
## *                                               640260
## Bra000001                                           NA
## Bra000002                                            3
## Bra000003                                          154
## Bra000004                                            1
## Bra000005                                                 498
##            R500_NDP_3_INTERNODE.1_matched.merged.fq.bam
## *                                                     599118
## Bra000001                                                 NA
## Bra000002                                                  3
## Bra000003                                                145
## Bra000004                                                  2
## Bra000005                                                464
```

```r
## Remove unwanted entries
# Genes that didn't map to a gene are summed into a "*" feature
rawCounts <- rawCounts[-grep("\\*",rownames(rawCounts)),]

## Deal with missing values (NA)
any(is.na(rawCounts))
```

```
## [1] TRUE
```

```r
rawCounts[is.na(rawCounts)] <- 0 #Convert all missing values to 0
```

metadata

It is a good practice to create a *metadata* file. Essentially, a table with information (as much as you can) about the samples, for example:

- Genotype
- Treatment
- Age
- Time of the day at collection
- Replicate number
- Other
  - Even the sequencing lane or thestrips of PCR tubes in which samples are processed can have some influence (batch effects)

This data frame can be manually created using excel, or on the same script using R magic!

By looking at the column names we can get an idea of the samples in the data. We can observe two things:

1. Naming convention is consistent across samples (This will make our life easier!)
2. There is some unecessary information that we can discard right away (ie, `.fq.bam` extension of the files)

```r
colnames(rawCounts) # Check the names of the samples
```

```
##  [1] "IMB211_DP_1_INTERNODE.1_matched.merged.fq.bam"
##  [2] "IMB211_DP_2_INTERNODE.1_matched.merged.fq.bam"
##  [3] "IMB211_DP_3_INTERNODE.1_matched.merged.fq.bam"
##  [4] "IMB211_NDP_1_INTERNODE.1_matched.merged.fq.bam"
##  [5] "IMB211_NDP_2_INTERNODE.1_matched.merged.fq.bam"
##  [6] "IMB211_NDP_3_INTERNODE.1_matched.merged.fq.bam"
##  [7] "R500_DP_1_INTERNODE.1_matched.merged.fq.bam"
##  [8] "R500_DP_2_INTERNODE.1_matched.merged.fq.bam"
##  [9] "R500_DP_3_INTERNODE.1_matched.merged.fq.bam"
## [10] "R500_NDP_1_INTERNODE.1_matched.merged.fq.bam"
## [11] "R500_NDP_2_INTERNODE.1_matched.merged.fq.bam"
## [12] "R500_NDP_3_INTERNODE.1_matched.merged.fq.bam"
```

```
samples <- colnames(rawCounts) #Store the sample names into a new variable

samples <- gsub("_matched.merged.fq.bam","",samples) #Remove unecessary information

## Now we want to separate each bit of information into columns:
# I will nest two functions:
# the inner (strsplit), will separate into unique fields the sample names each time it sees an undersco
# the middle (do.call), will take these fields and arrange them into a matrix (by rows, with rbind)
# the outer just converts the matrix into a data frame and adds the samples as rownames
# Since the naming convention is the same for all samples, we don't need to worry about having a table

meta <- data.frame( do.call("rbind",
                        strsplit(samples,"_")
                    ),row.names = samples)

## Manually add the column names
colnames(meta) <- c("genotype","treatment","replicate","tissue")

## We can create a new group that combines genotype and treatment to make comparisons easier:
meta$group <- paste(meta$genotype,meta$treatment,sep = "_")


meta
```

```
##                          genotype treatment replicate      tissue
## IMB211_DP_1_INTERNODE.1    IMB211        DP         1 INTERNODE.1
## IMB211_DP_2_INTERNODE.1    IMB211        DP         2 INTERNODE.1
## IMB211_DP_3_INTERNODE.1    IMB211        DP         3 INTERNODE.1
## IMB211_NDP_1_INTERNODE.1   IMB211       NDP         1 INTERNODE.1
## IMB211_NDP_2_INTERNODE.1   IMB211       NDP         2 INTERNODE.1
## IMB211_NDP_3_INTERNODE.1   IMB211       NDP         3 INTERNODE.1
## R500_DP_1_INTERNODE.1        R500        DP         1 INTERNODE.1
## R500_DP_2_INTERNODE.1        R500        DP         2 INTERNODE.1
## R500_DP_3_INTERNODE.1        R500        DP         3 INTERNODE.1
## R500_NDP_1_INTERNODE.1       R500       NDP         1 INTERNODE.1
## R500_NDP_2_INTERNODE.1       R500       NDP         2 INTERNODE.1
## R500_NDP_3_INTERNODE.1       R500       NDP         3 INTERNODE.1
##                               group
## IMB211_DP_1_INTERNODE.1   IMB211_DP
## IMB211_DP_2_INTERNODE.1   IMB211_DP
## IMB211_DP_3_INTERNODE.1   IMB211_DP
## IMB211_NDP_1_INTERNODE.1 IMB211_NDP
## IMB211_NDP_2_INTERNODE.1 IMB211_NDP
## IMB211_NDP_3_INTERNODE.1 IMB211_NDP
## R500_DP_1_INTERNODE.1       R500_DP
## R500_DP_2_INTERNODE.1       R500_DP
## R500_DP_3_INTERNODE.1       R500_DP
## R500_NDP_1_INTERNODE.1     R500_NDP
## R500_NDP_2_INTERNODE.1     R500_NDP
## R500_NDP_3_INTERNODE.1     R500_NDP
```

An important thing to consider is that the names of the columns in the counts table match the rownames in the metadata table

```
colnames(rawCounts) == rownames(meta) #Since we manually got rid of the extra info, we need to make sur
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE
```

```
# We simply apply the same line on the column names of the counts table
colnames(rawCounts) <- gsub("_matched.merged.fq.bam","",colnames(rawCounts))

# done!
all(colnames(rawCounts) == rownames(meta))
```

```
## [1] TRUE
```

Differential Gene Expression

Create the DGE object

edgeR (and limma) use a specific type of object to do all of their inner workings, so we need to call a function to create it, using the count data

```
dge <- DGEList(counts=rawCounts,
               remove.zeros = T) # Removes genes that were consistently not expressed (0 counts) across
```

```
## Removing 5467 rows with all zero counts
```

```
nrow(rawCounts)
```

```
## [1] 40990
```

```
nrow(dge)
```

```
## [1] 35523
```

Remove lowly expressed genes

To remove noisy genes, we can filter the data. This ensures we're working with biologically meaningful genes.

We ask for genes for which, in at least 3 (which is the number of replicates per group) of *ANY* samples, the CPM is higher than 1.

```
isexpr <- rowSums(cpm(dge) > 1) >= 3
dge <- dge[isexpr,,
           keep.lib.size = FALSE] #Recalculates the library size after removing lowly expressed genes
```

For a deeper discussion on this, check the article `RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR` (https://f1000research.com/articles/5-1408/)

Normalization

The normalization step is used to ensure that libraries are comparable among one another, and aims to reduce variation that is not biologically interesting.

```
dge <- calcNormFactors(dge)
```

Design and contrast matrix

This is an important step since it defines what will be compared and how.

Using the metadata object we created earlier, we'll tell edgeR to create the design matrix using a linear model. The linear model explanation and its use in differential expression analysis is outside of the scope of this primer. In short, using the model we're telling edgeR that expression of the gene is influenced by different factors (ie, genotype and/or treatment). Later, by use of a fit and regression, we can ask if either (or an interaction) are significantly influencing expression of the gene.

```
### Create design matrix

# model.matrix(~0+genotype + treatment, data=meta)
design <- model.matrix(~0+group, data=meta) #To make comparisons, the grouping factor is a better optio
colnames(design) <- gsub("genotype|treatment|group|:|-|/","",colnames(design)) #
head(design)
```

```
##                            IMB211_DP IMB211_NDP R500_DP R500_NDP
## IMB211_DP_1_INTERNODE.1          1          0       0        0
## IMB211_DP_2_INTERNODE.1          1          0       0        0
## IMB211_DP_3_INTERNODE.1          1          0       0        0
## IMB211_NDP_1_INTERNODE.1         0          1       0        0
## IMB211_NDP_2_INTERNODE.1         0          1       0        0
## IMB211_NDP_3_INTERNODE.1         0          1       0        0
```

```
colSums(design) # Should be the same as number of bioreps per group
```

```
##  IMB211_DP IMB211_NDP    R500_DP   R500_NDP
##          3          3          3          3
```

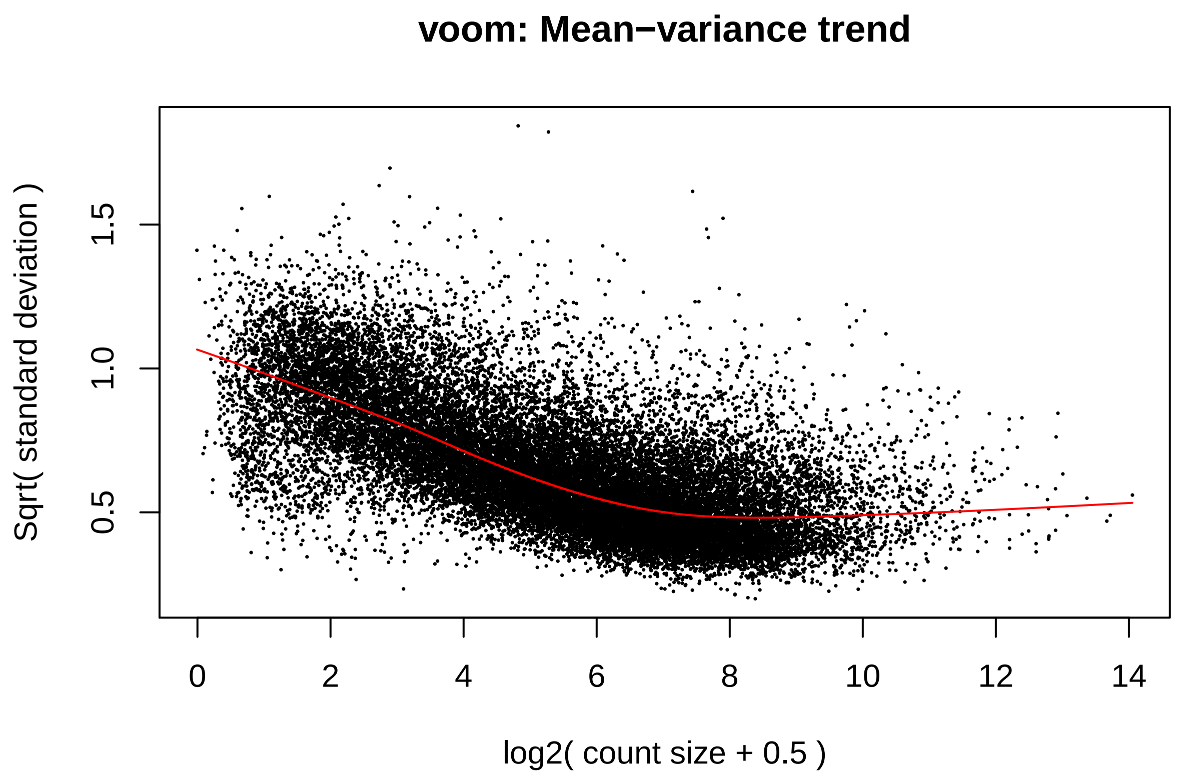
voom() function in limma

voom() transforms the values to logCPM (assumed to be normally distributed). It also makes use of the design matrix defined previously. Further normalization can be obtained using the `normalize.method` parameter.

> "When the library sizes are quite variable between samples, then the voom approach is theoretically more powerful than limma-trend. In this approach, the voom transformation is applied to the normalized and filtered DGEList object:"

(https://www.bioconductor.org/packages/release/bioc/vignettes/limma/inst/doc/usersguide.pdf)

```
v <- voom(dge, design, plot = TRUE,normalize.method = "quantile")
```



Distribution of the data

To look at how the samples behave once the counts have been normalized we can generate an interactive MDS plot

```
glMDSPlot(v, labels=rownames(meta), groups=meta,
          folder=paste0("glimma_voom"),
          launch=T)
```

```
### Contrasts & DGE
v2 <- lmFit(v, design)
contrastMatrix <- makeContrasts(

  "DensePlanting_vs_NonDP_IMB211"=(IMB211_DP-IMB211_NDP),
  "DensePlanting_vs_NonDP_R500"=(R500_DP-R500_NDP),
  levels = design)
```

```
fit2 <- contrasts.fit(v2, contrastMatrix)
fit2 <- eBayes(fit2)
results <- decideTests(fit2)
t(summary(results))
```

```
##                                Down NotSig   Up
## DensePlanting_vs_NonDP_IMB211   462  25907  711
## DensePlanting_vs_NonDP_R500       1  27079    0
```

Automatization: Get DEGs per contrast

```r
# Define an adjusted P-value cut to call significant genes
pValCut <- 0.05

## Define the contrasts
uniqContrasts <- colnames(contrastMatrix)

### Prepare lists to save results
DEList <- list()
DESignificant <- list()

for (contrast in uniqContrasts){
  cat(" - - - \n")
  ##

  cat("Contrast:", paste0(contrast),"\n")
  #
  tmp <- topTable(fit2, coef=contrast,number = Inf,sort.by = "none")


  ### Change names of columns
  colnames(tmp) <- paste(contrast,colnames(tmp),sep = ".")

  ## Save to list
  DEList[[contrast]] <- tmp

  ## Filter
  tmpIDX <- grep("adj.P.Val",colnames(tmp))
  tmpSign <- tmp[tmp[,tmpIDX] < pValCut,]
  nrow(tmpSign)
  DESignificant[[contrast]] <- tmpSign

  cat ("Number of DEGs on",contrast,":",nrow(tmpSign),"\n")
  #
}
```

```
##  - - -
## Contrast: DensePlanting_vs_NonDP_IMB211
## Number of DEGs on DensePlanting_vs_NonDP_IMB211 : 1173
##  - - -
## Contrast: DensePlanting_vs_NonDP_R500
## Number of DEGs on DensePlanting_vs_NonDP_R500 : 1
```

```r
names(DEList)
```

```
## [1] "DensePlanting_vs_NonDP_IMB211" "DensePlanting_vs_NonDP_R500"
```

Filter significant genes

```r
sapply(DEList,function(x){
  nrow(x[x[,grep("adj.P.Val",colnames(x))] < 0.05,])
})
```

```
## DensePlanting_vs_NonDP_IMB211   DensePlanting_vs_NonDP_R500
##                          1173                             1
```

```r
significantDE <- lapply(DEList,function(x){ x[x[,grep("adj.P.Val",colnames(x))] < 0.05,] })
sapply(significantDE,nrow)
```

```
## DensePlanting_vs_NonDP_IMB211   DensePlanting_vs_NonDP_R500
##                          1173                             1
```