Analysis and Exploration of Proteomics Data

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Proteomics Data Analysis

Below are shown the steps typically used to perform Differential Gene Expression analysis. The analysis steps have been applied over the *EGF-driven protein synthesis* transcriptomics data from D.A. Rothenberg et al. A Proteomics Approach to Profiling the Temporal Translational Response to Stress and Growth. iScience. 2018; 9:367-381.

Loading of R-Packages

We start by loading the R-packages we need to use for our analysis:

```
library(edgeR)
library(lattice)
library(biomaRt)
library(gplots)
library(gridExtra)
library(openxlsx)
library(tools)
library(ggplot2)
library(dplyr)
library(Glimma)
library(openxlsx)
```

Loading of the data

We load the generated **subread_counts** object, define the experimental conditions for each available sample and prepare the data in the *DGEObj* format which represents a flexible container to manage and annotate Differential Gene Expression (DGE) analysis results.

```
load("course.Rdata")

groups <- factor(c("PBS","EGF","PBS","EGF"))

label <- c("PBS2", "EGF1", "PBS1", "EGF2")

print(colnames(subread_counts$counts))
```

```
## [1] "SRR7451179.Rep.2.MP60.2.mRNA.Seq.PBS.bam"
## [2] "SRR7451182.Rep.1.ME60.1.mRNA.Seq.EGF.bam"
## [3] "SRR7451187.Rep.1.MP60.1.mRNA.Seq.PBS.bam"
## [4] "SRR7451201.Rep.2.ME60.2.mRNA.Seq.EGF.bam"
```

```
print(colnames(subread_counts$counts))
```

```
## [1] "SRR7451179.Rep.2.MP60.2.mRNA.Seq.PBS.bam"
## [2] "SRR7451182.Rep.1.ME60.1.mRNA.Seq.EGF.bam"
## [3] "SRR7451187.Rep.1.MP60.1.mRNA.Seq.PBS.bam"
## [4] "SRR7451201.Rep.2.ME60.2.mRNA.Seq.EGF.bam"
```

```
DGEObj <- DGEList(group=groups, counts=subread_counts$counts, genes=subread_counts$annotation[,c("GeneI

print(DGEObj)
```

```
## An object of class "DGEList"
## $counts
##                 SRR7451179.Rep.2.MP60.2.mRNA.Seq.PBS.bam
## ENSG00000223972                                        0
## ENSG00000227232                                       14
## ENSG00000278267                                        0
## ENSG00000243485                                        0
## ENSG00000284332                                        0
##                 SRR7451182.Rep.1.ME60.1.mRNA.Seq.EGF.bam
## ENSG00000223972                                        0
## ENSG00000227232                                       24
## ENSG00000278267                                        0
## ENSG00000243485                                        0
## ENSG00000284332                                        0
##                 SRR7451187.Rep.1.MP60.1.mRNA.Seq.PBS.bam
## ENSG00000223972                                        0
## ENSG00000227232                                       10
## ENSG00000278267                                        0
## ENSG00000243485                                        0
## ENSG00000284332                                        0
##                 SRR7451201.Rep.2.ME60.2.mRNA.Seq.EGF.bam
## ENSG00000223972                                        0
## ENSG00000227232                                       22
## ENSG00000278267                                        0
## ENSG00000243485                                        0
## ENSG00000284332                                        0
## 58879 more rows ...
##
## $samples
##                                         group lib.size norm.factors
## SRR7451179.Rep.2.MP60.2.mRNA.Seq.PBS.bam   PBS 26653631            1
## SRR7451182.Rep.1.ME60.1.mRNA.Seq.EGF.bam   EGF 30763088            1
## SRR7451187.Rep.1.MP60.1.mRNA.Seq.PBS.bam   PBS 28377976            1
## SRR7451201.Rep.2.ME60.2.mRNA.Seq.EGF.bam   EGF 31739139            1
##
## $genes
##                            GeneID Length
## ENSG00000223972 ENSG00000223972   1735
## ENSG00000227232 ENSG00000227232   1351
## ENSG00000278267 ENSG00000278267     68
## ENSG00000243485 ENSG00000243485   1021
## ENSG00000284332 ENSG00000284332    138
## 58879 more rows ...
```

Experimental Design

We make an experimental design matrix where we indicate the condition and replicate ID's for each sample. We then define the testing contrasts of our experimental design (*EGF_vs_PBS*). Our targeted *EGF_vs_PBS* contrast tests whether the average expression across all *EGF* groups is equal to the average expression across all *PBS* groups using *makeContrasts*.

```
design <- model.matrix(~0+groups)
colnames(design) <- levels(groups)
print(design)
```

```
##   EGF PBS
## 1   0   1
## 2   1   0
## 3   0   1
## 4   1   0
## attr(,"assign")
## [1] 1 1
## attr(,"contrasts")
## attr(,"contrasts")$groups
## [1] "contr.treatment"
```

```
contrasts <- makeContrasts(
  EGF_vs_PBS = EGF - PBS,
  levels=design
)
counts <- DGEObj$counts
```

DGEObj Processing

We simplify a bit the names of the samples.

```
head(DGEObj$samples)
```

```
##                                         group lib.size norm.factors
## SRR7451179.Rep.2.MP60.2.mRNA.Seq.PBS.bam   PBS 26653631            1
## SRR7451182.Rep.1.ME60.1.mRNA.Seq.EGF.bam   EGF 30763088            1
## SRR7451187.Rep.1.MP60.1.mRNA.Seq.PBS.bam   PBS 28377976            1
## SRR7451201.Rep.2.ME60.2.mRNA.Seq.EGF.bam   EGF 31739139            1
```

```
rownames(DGEObj$samples) <- gsub("\\.", "_", rownames(DGEObj$samples))
rownames(DGEObj$samples) <- gsub("KD_", "", rownames(DGEObj$samples))
rownames(DGEObj$samples) <- gsub("_circSLC8A1", "", rownames(DGEObj$samples))
rownames(DGEObj$samples) <- gsub("_bam", "", rownames(DGEObj$samples))
rownames(DGEObj$counts) <- gsub("_1_mRNA_Seq_PBS", "", rownames(DGEObj$counts))
rownames(DGEObj$counts) <- gsub("_1_mRNA_Seq_EGF", "", rownames(DGEObj$counts))
rownames(DGEObj$counts) <- gsub("_2_mRNA_Seq_PBS", "", rownames(DGEObj$counts))
rownames(DGEObj$counts) <- gsub("_2_mRNA_Seq_EGF", "", rownames(DGEObj$counts))

colnames(DGEObj$counts) <- gsub("\\.", "_", colnames(DGEObj$counts))
colnames(DGEObj$counts) <- gsub("KD_", "", colnames(DGEObj$counts))
colnames(DGEObj$counts) <- gsub("_1_mRNA_Seq_PBS", "", colnames(DGEObj$counts))
colnames(DGEObj$counts) <- gsub("_1_mRNA_Seq_EGF", "", colnames(DGEObj$counts))
colnames(DGEObj$counts) <- gsub("_2_mRNA_Seq_PBS", "", colnames(DGEObj$counts))
colnames(DGEObj$counts) <- gsub("_2_mRNA_Seq_EGF", "", colnames(DGEObj$counts))
colnames(DGEObj$counts) <- gsub("_bam", "", colnames(DGEObj$counts))
```

```
DGEObj$samples
```

```
##                                     group lib.size norm.factors
## SRR7451179_Rep_2_MP60_2_mRNA_Seq_PBS   PBS 26653631            1
## SRR7451182_Rep_1_ME60_1_mRNA_Seq_EGF   EGF 30763088            1
## SRR7451187_Rep_1_MP60_1_mRNA_Seq_PBS   PBS 28377976            1
## SRR7451201_Rep_2_ME60_2_mRNA_Seq_EGF   EGF 31739139            1
```

```
head(DGEObj$counts)
```

```
##                 SRR7451179_Rep_2_MP60 SRR7451182_Rep_1_ME60
## ENSG00000223972                     0                     0
## ENSG00000227232                    14                    24
## ENSG00000278267                     0                     0
## ENSG00000243485                     0                     0
## ENSG00000284332                     0                     0
## ENSG00000237613                     0                     0
##                 SRR7451187_Rep_1_MP60 SRR7451201_Rep_2_ME60
## ENSG00000223972                     0                     0
## ENSG00000227232                    10                    22
## ENSG00000278267                     0                     0
## ENSG00000243485                     0                     0
## ENSG00000284332                     0                     0
## ENSG00000237613                     0                     0
```

```
head(DGEObj$genes)
```

```
##                         GeneID Length
## ENSG00000223972 ENSG00000223972   1735
## ENSG00000227232 ENSG00000227232   1351
## ENSG00000278267 ENSG00000278267     68
## ENSG00000243485 ENSG00000243485   1021
## ENSG00000284332 ENSG00000284332    138
## ENSG00000237613 ENSG00000237613   1219
```

```
DGEObj
```

```
## An object of class "DGEList"
## $counts
##                 SRR7451179_Rep_2_MP60 SRR7451182_Rep_1_ME60
## ENSG00000223972                     0                     0
## ENSG00000227232                    14                    24
## ENSG00000278267                     0                     0
## ENSG00000243485                     0                     0
## ENSG00000284332                     0                     0
##                 SRR7451187_Rep_1_MP60 SRR7451201_Rep_2_ME60
## ENSG00000223972                     0                     0
## ENSG00000227232                    10                    22
## ENSG00000278267                     0                     0
## ENSG00000243485                     0                     0
## ENSG00000284332                     0                     0
## 58879 more rows ...
##
## $samples
##                                     group lib.size norm.factors
```

```
## SRR7451179_Rep_2_MP60_2_mRNA_Seq_PBS   PBS 26653631            1
## SRR7451182_Rep_1_ME60_1_mRNA_Seq_EGF   EGF 30763088            1
## SRR7451187_Rep_1_MP60_1_mRNA_Seq_PBS   PBS 28377976            1
## SRR7451201_Rep_2_ME60_2_mRNA_Seq_EGF   EGF 31739139            1
##
## $genes
##                          GeneID Length
## ENSG00000223972 ENSG00000223972   1735
## ENSG00000227232 ENSG00000227232   1351
## ENSG00000278267 ENSG00000278267     68
## ENSG00000243485 ENSG00000243485   1021
## ENSG00000284332 ENSG00000284332    138
## 58879 more rows ...
```

Data Filtering

We filter the data by keeping only candidates with > 2 counts per million mapped reads AND in 3 libs.

```
keep <- rowSums(cpm(DGEObj) > 1) >= 3

# select sub set from above
DGEObj <- DGEObj[keep, keep.lib.sizes = FALSE]

dim(DGEObj)
```

```
## [1] 14691     4
```

edgeR Analysis

We calculate the normalization factor, estimate the dispersion, fit the linear model, and then summarize the results in an edgeR table format.

```
# The filtered raw counts are then normalized with calcNormFactors according to
# the weighted trimmed mean of M-values (TMM) to eliminate composition biases
# between libraries.
DGEObj <- calcNormFactors(DGEObj) # recalc norm factors - TMM

# With the normalized gene counts and design matrix we can now generate the
# negative binomial (NB) dispersion estimates using the estimateDisp function.
# The NB dispersion estimates reflect the overall biological variability under
# the Quasi-Likelihood framework in edgeR.
DGEObj <- estimateDisp(DGEObj, design) # estimate dispersion
DGEObj$common.dispersion
```

```
## [1] 0.001686058
```

```
# A generalized linear model (GLM) of the Negative Binomial family is used to
# fit the data.
fit <- glmFit(DGEObj, design) # fit generalized linear model
lrt <- glmLRT(fit, contrast = contrasts[, "EGF_vs_PBS"])

# main data table of edgeR results
edgerTable <- topTags(lrt, n = nrow(DGEObj))$table

y <- cpm(DGEObj, log = TRUE, prior.count = 1)
lcpm <- cpm(DGEObj, log = TRUE)
```

Mapping

We do mappping of the Ensembl gene ID's.

```
ensembl <- useMart("ensembl", dataset = "hsapiens_gene_ensembl", host="ensembl.org")
```

```
## Warning: Ensembl will soon enforce the use of https.
## Ensure the 'host' argument includes "https://"
```

```
genemap <- getBM(attributes = c("external_gene_name", "ensembl_gene_id", "description"),
                 values = edgerTable$GeneID, mart = ensembl)
genemap2 <- getBM(attributes = c("ensembl_gene_id", "go_id"), filter = "ensembl_gene_id",
                  values = edgerTable$GeneID, mart = ensembl)

edgerTable.idx <- match(edgerTable$GeneID, genemap$ensembl_gene_id)
edgerTable.idx2 <- match(edgerTable$GeneID, genemap2$ensembl_gene_id)

# add biomart data to results tablesTRUE
edgerTable$external_gene_name <- genemap$external_gene_name[edgerTable.idx]
# edgerTable$entrezgene <- genemap$entrezgene[edgerTable.idx]
edgerTable$description <- genemap$description[edgerTable.idx]
edgerTable$GO <- genemap2$go_id[edgerTable.idx2]
edgerTable$ensembl_gene_id <- genemap$ensembl_gene_id[edgerTable.idx]
lrt$table$gene <- genemap$external_gene_name[edgerTable.idx]
gene_names <- data.frame(lrt$table$gene)

edgerTable <- edgerTable[c("external_gene_name", "GeneID" , "Length" , "logFC",
                           "logCPM", "PValue", "FDR", "GO", "description")]

ttop_dge <- edgerTable
head(ttop_dge)
```

```
##                 external_gene_name          GeneID Length    logFC   logCPM
## ENSG00000125740               FOSB ENSG00000125740   5553 4.962212 6.902998
## ENSG00000138166              DUSP5 ENSG00000138166   2535 3.983957 6.291272
## ENSG00000119508              NR4A3 ENSG00000119508   6314 3.302585 8.107344
## ENSG00000175592              FOSL1 ENSG00000175592   1887 3.307155 5.967628
## ENSG00000171223               JUNB ENSG00000171223   1830 3.283124 7.112127
## ENSG00000162772               ATF3 ENSG00000162772   4040 3.119576 7.369775
##                       PValue           FDR         GO
## ENSG00000125740  0.000000e+00  0.000000e+00 GO:0003677
## ENSG00000138166  0.000000e+00  0.000000e+00 GO:0016791
## ENSG00000119508  0.000000e+00  0.000000e+00 GO:0003677
## ENSG00000175592 5.605306e-296 2.058689e-292 GO:0003677
## ENSG00000171223 1.089103e-281 3.200002e-278 GO:0003677
## ENSG00000162772 9.076298e-280 2.222331e-276 GO:0003677
##                                                                                                                                  descri
## ENSG00000125740 FosB proto-oncogene, AP-1 transcription factor subunit [Source:HGNC Symbol;Acc:HGNC:3
## ENSG00000138166                           dual specificity phosphatase 5 [Source:HGNC Symbol;Acc:HGNC:3
## ENSG00000119508         nuclear receptor subfamily 4 group A member 3 [Source:HGNC Symbol;Acc:HGNC:7
## ENSG00000175592        FOS like 1, AP-1 transcription factor subunit [Source:HGNC Symbol;Acc:HGNC:13
## ENSG00000171223 JunB proto-oncogene, AP-1 transcription factor subunit [Source:HGNC Symbol;Acc:HGNC:6
## ENSG00000162772                       activating transcription factor 3 [Source:HGNC Symbol;Acc:HGNC
```