RNA-seq analysis in R

Differential Expression of RNA-seq data

Exercise 1

So far we have fitted a simple model considering just “Status”, but in reality we want to model the effects of both “Status” and “Time Point”.

Let’s start with the model with only main effects - an additive model with no interaction. The main assumption here is that the effects of Status and the effects of Time Point are indepedent.

Recapitulate the above steps to generate a new DESeq2 object with the additive model. Then we will extract the results table as above.

Load the raw data, remembering to set the factor on the Status so that “Uninfected” will be set as the intercept:

```r
txi <- readRDS("RObjects/txi.rds")
sampleinfo <- read_tsv("data/samplesheet_corrected.tsv", col_types="cccc") %>%
                   mutate(Status = fct_relevel(Status, "Uninfected"))
```

```r
additive.model <- as.formula(~ TimePoint + Status)
```

```r
ddsObj.raw <- DESeqDataSetFromTximport(txi = txi,
                                       colData = sampleinfo,
                                       design = additive.model)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
## using counts and average transcript lengths from tximport
```

```
keep <- rowSums(counts(ddsObj.raw)) > 5
ddsObj.filt <- ddsObj.raw[keep,]
```

You are now ready to run the differential gene expression analysis Run the DESeq2 analysis

1. Run the size factor estimation, dispersion estimation and modelling steps using the `DESeq` command as above.

```
ddsObj <- DESeq(ddsObj.filt)
```

```
## estimating size factors
```

```
## using 'avgTxLength' from assays(dds), correcting for library size
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

2. Extract the default contrast using the `results` command into a new object called `results.additive`

```
results.additive <- results(ddsObj, alpha=0.05)
results.additive
```

```
## log2 fold change (MLE): Status Infected vs Uninfected
## Wald test p-value: Status Infected vs Uninfected
## DataFrame with 20091 rows and 6 columns
##                      baseMean log2FoldChange     lfcSE       stat      pvalue
##                     <numeric>      <numeric> <numeric>  <numeric>   <numeric>
## ENSMUSG00000000001 1102.56094     -0.0110965  0.106195  -0.104492    0.916779
## ENSMUSG00000000028   58.60055      0.3007930  0.265626   1.132391    0.257470
## ENSMUSG00000000037   49.23586     -0.0481414  0.429685  -0.112039    0.910793
## ENSMUSG00000000049    7.98789      0.4110498  0.656171   0.626437    0.531028
## ENSMUSG00000000056 1981.00402     -0.1907691  0.119694  -1.593809    0.110979
##                           padj
##                      <numeric>
## ENSMUSG00000000001    0.967428
## ENSMUSG00000000028    0.514578
## ENSMUSG00000000037    0.965220
## ENSMUSG00000000049    0.757304
## ENSMUSG00000000056    0.314608
##  [ reached getOption("max.print") -- omitted 6 rows ]
```

a) What contrast are these results for? If you have constructed the model correctly, then it should be the same as previous `results.simple` Again this results table is for the contrast Infected v Uninfected.
b) How many genes have an adjusted p-value of less than 0.05

```
sum(results.additive$padj < 0.05, na.rm = TRUE)
```

```
## [1] 2766
```

Exercise 2

If we want a different contrast we can just pass the `results` function the **name** of the contrast, as given by `resultsNames(ddsObj)`. Look at the help page for the `results` command to see how to do this.

1. Retrieve the results for the contrast of d33 versus d11.

```
results.d33vd11 <- results(ddsObj, name= "TimePoint_d33_vs_d11", alpha=0.05)
```

2. How many differentially expressed genes are there at FDR < 0.05?

```
sum(results.d33vd11$padj < 0.05, na.rm = TRUE)
```

```
## [1] 109
```

Exercise 3

When we looked at the PCA it did seem that an interaction model might be warranted. Let's test that.

1. Create a new DESeq2 object using a model with an interaction between TimePoint and Status. The model formula should be

   `~TimePoint + Status + TimePoint:Status`

   where `TimePoint:Status` is the parameter for the interaction beteween TimePoint and Status.

Note that `*` can be used as shortcut to add the interaction term, e.g. `~TimePoint * Status`, however, writing out in long form is clearer here.

Remember to filter to remove uninformative genes.

```
interaction.model <- as.formula(~ TimePoint * Status)
ddsObj.raw <- DESeqDataSetFromTximport(txi = txi,
                                       colData = sampleinfo,
                                       design = interaction.model)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
## using counts and average transcript lengths from tximport
```

```
keep <- rowSums(counts(ddsObj.raw)) > 5
ddsObj.filt <- ddsObj.raw[keep,]
```

2. Run the statistical analysis using the `DESeq` command and create a new analysis object called `ddsObj.interaction`.

```
ddsObj.interaction <- DESeq(ddsObj.filt)
```

```
## estimating size factors
```

```
## using 'avgTxLength' from assays(dds), correcting for library size
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
## final dispersion estimates
```

```
## fitting model and testing
```

3. Use the LRT to compare this to the simpler additive model (`~TimePoint + Status`)

```
ddsObj.LRT <- DESeq(ddsObj.interaction, test="LRT", reduced=additive.model)
```

```
## using pre-existing normalization factors
```

```
## estimating dispersions
```

```
## found already estimated dispersions, replacing these
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
results.Interaction_v_Additive <- results(ddsObj.LRT)
```

4. Extract a table of results using `results`. For how many genes is interaction model a better fit?

```
table(results.Interaction_v_Additive$padj < 0.05)
```

```
##
## FALSE  TRUE
## 16474   455
```

Exercise 4

Let’s investigate the uninfected mice

1. Extract the results for d33 v d11 for uninfected mice
   The the intercept is Uninfected mice at 11 days post infection, so the main effect `TimePoint_d33_vs_d11` is the result that we want.

```
results.d33_v_d11_uninfected <- results(ddsObj.interaction,
                                        name="TimePoint_d33_vs_d11")
```

How many genes have an adjusted p-value less than 0.05?

```
table(results.d33_v_d11_uninfected$padj < 0.05)
```

```
##
## FALSE  TRUE
## 20043     1
```

Is this remarkable?
Maybe not. Do we really expect vast gene expression differences between the brains of mice that are slightly older than one another? It is possible that there could have been confounding factors, such as changes in enviromental conditions such as temperature or feeding regime, that may have effected gene expression. In which case it was important to set the experiment up with control for both time points. Does this suggest another approach to analysing this data set?
Could we possibly treat the six uninfected samples as a single group with six replicates? This is really a biological question and not a statistical one.