Introduction to Bulk RNAseq data analysis

Differential Expression of RNA-seq data

Last modified: 27 Sep 2024

Contents

3. Creating the design model formula

Exercise 1

> This time create and investigate the model matrix for the variable "Status". 1. Create a model formula to investigate the effect of "Status" on gene expression.

```
simple.model <- as.formula(~ Status)
```

What does this look like as a model matrix?

```
model.matrix(simple.model, data = sampleinfo)
```

```
##    (Intercept) StatusUninfected
## 1            1                0
## 2            1                0
## 3            1                0
## 4            1                0
## 5            1                1
## 6            1                0
## 7            1                1
## 8            1                1
## 9            1                1
## 10           1                1
## 11           1                0
## 12           1                1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$Status
## [1] "contr.treatment"
```

2. Look at the model matrix and identify which is the reference group in your model.

The β_1 coeffient is encoded in the second column of the model matrix. The column header tell us that this is `StatusUninfected`, therefore, logically, the reference must be `StatusUninfected`.

4. Build a DESeq2DataSet

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

```
## 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
```

```
## 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    -0.00802952  0.102877  -0.078050    0.937788
## ENSMUSG00000000028   58.60055     0.30498077  0.254312   1.199239    0.230435
## ENSMUSG00000000037   49.23586    -0.05272685  0.416862  -0.126485    0.899348
## ENSMUSG00000000049    7.98789     0.38165132  0.644869   0.591827    0.553966
## ENSMUSG00000000056 1981.00402    -0.16921845  0.128542  -1.316449    0.188024
##                          padj
##                     <numeric>
## ENSMUSG00000000001   0.975584
## ENSMUSG00000000028   0.480598
## ENSMUSG00000000037   0.961314
## ENSMUSG00000000049   0.772123
## ENSMUSG00000000056   0.426492
##  [ reached getOption("max.print") -- omitted 6 rows ]
```

Exercise 2

Now we have made our results table using our simple model, let have a look at which genes are changing and how many pass our 0.05 threshold.

a) how many genes are significantly (with an FDR < 0.05) up-regulated?

```
sum(results.simple$padj < 0.05 & results.simple$log2FoldChange > 0, na.rm = TRUE)
```

```
## [1] 1879
```

b) how many genes are significantly (with an FDR < 0.05) down-regulated?

```
sum(results.simple$padj < 0.05 & results.simple$log2FoldChange < 0, na.rm = TRUE)
```

```
## [1] 1005
```

c) Here is the results table for two of the genes:

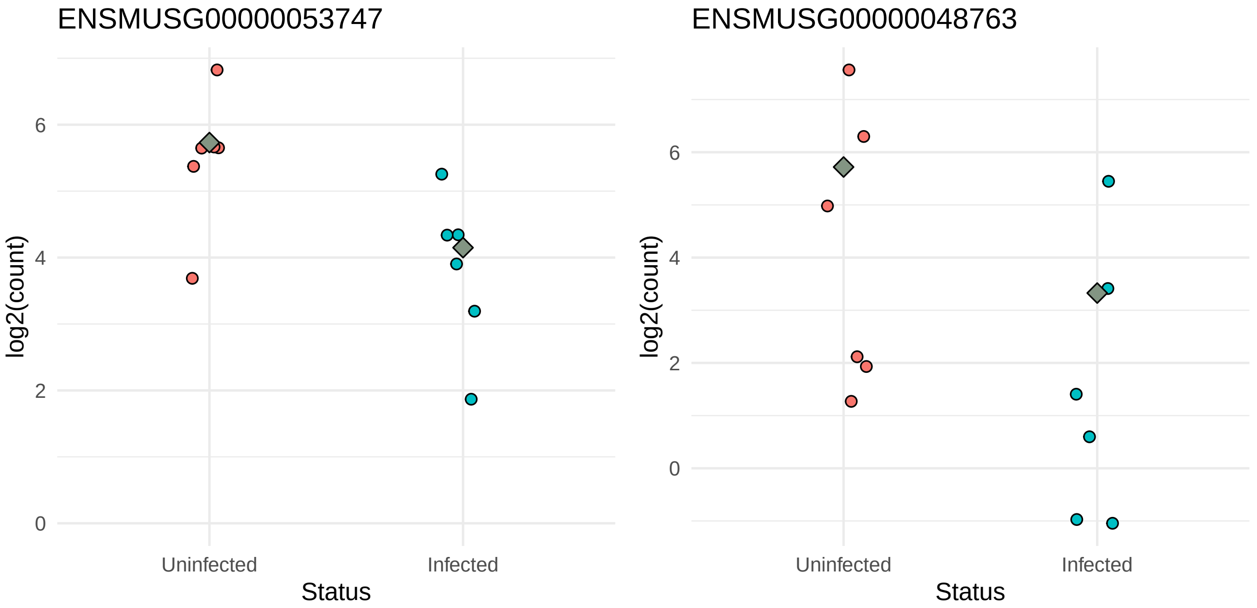
	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
ENSMUSG00000053747	34.930	-1.61	0.531	-3.04	0.00238	0.0199
ENSMUSG00000048763	30.853	-2.46	1.450	-1.70	0.08970	0.2700

One of these is strongly downregulated with log2(fold-change) of -2.46. On a linear scale this is $2^{2.46} = 5.5$ times more highly expressed in the uninfected group relative to the infected group, but its adjusted p-value is 0.27. By contrast, the other gene has a lower LFC ~ -1.61 (on a linear scale $2^{1.61} = 3.05$), but it's adjusted p-value is 0.0023.

How can you explain this apparent contradiction?

The statistical significance embodied by the p-value is factor of both the mean effect size - here this is the log2FoldChange - and the error or variance around this mean - in this case the standard error of the log2FoldChange. Although the fold change of the first gene is considerably higher the standard error is much larger relative to the fold change and therefore we can less certain that this mean is a true measure of the difference between our populations.

We can look at this with by plotting the read counts for these two genes:



The grey diamonds show the mean values for each group, the points are counts for individual samples. The points for individual samples have been "jittered" around the x-axis so that overlapping points are visible. You can see that for both groups the values for individual samples are much closer to the mean for ENSMUSG00000053747 than for ENSMUSG00000048763. The code for this plot has been omitted, but we will see how to plot something similar in the visualisation section of the course.

7. The additive model

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

Exercise 3

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)
```

Questions: a) How many coefficients are there in the additive model?

To view the coefficients, generate the model matrix:

```
model.matrix(additive.model, data = sampleinfo)
```

```
##    (Intercept) TimePointd33 StatusInfected
## 1            1            0              1
## 2            1            0              1
## 3            1            0              1
## 4            1            1              1
## 5            1            1              0
## 6            1            1              1
## 7            1            0              0
## 8            1            0              0
## 9            1            0              0
## 10           1            1              0
## 11           1            1              1
## 12           1            1              0
## attr(,"assign")
## [1] 0 1 2
## attr(,"contrasts")
## attr(,"contrasts")$TimePoint
## [1] "contr.treatment"
##
## attr(,"contrasts")$Status
## [1] "contr.treatment"
```

So we have three coefficients:

- `Intercept`
- `TimePointd33`
- `StatusInfected`

```
    b) What is the reference group in the additive model?
```

Based on the above the reference (`Intercept`) must be day 11 and uninfected.

```
    c) What contrasts could we perform using this model?
```

We can perform two contrasts:

- $TimePoint_{d33}$ vs $TimePoint_{d11}$
- $Status_{Infected}$ vs $Status_{Uninfected}$

```
    d) What contrast does the `results.additive` object represent?
```

If we look a the top of the results

```
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 ]
```

We can see that the contrast that has been selected is “Status Infected v Uninfected”.

By default the `results` function has returned the contrast from the last coefficient in the model.

```
    e) 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 4

```
resultsNames(ddsObj)
```

```
## [1] "Intercept"                     "TimePoint_d33_vs_d11"
## [3] "Status_Infected_vs_Uninfected"
```

How do the named coefficients above relate to β_i coefficients in the design formula:

$$expression = \beta_0 + \beta_1 \cdot TimePoint_{d33} + \beta_2 \cdot Status_{Infected}$$

- The β_0 coefficient is the `Interecept`
- The β_1 coefficient gives us the change related to the contrast "TimePoint_d33_vs_d11"
- The β_2 coefficient gives us the change related to the contrast "Status_Infected_vs_Uninfected"

Exercise 5

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_d33_v_d11 <- results(ddsObj, name = "TimePoint_d33_vs_d11")
```

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

```
sum(results_d33_v_d11$padj < 0.05, na.rm = TRUE)
```

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

8. The interaction model

Exercise 6

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. Extract a table of results using the default `results` command. What is the contrast that these results are for?

```
results.int <- results(ddsObj.interaction)
results.int
```

```
## log2 fold change (MLE): TimePointd33.StatusInfected
## Wald test p-value: TimePointd33.StatusInfected
## DataFrame with 20091 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE      stat      pvalue
##                    <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSMUSG00000000001 1102.56094       0.305525  0.199432   1.531973 1.25529e-01
## ENSMUSG00000000028   58.60055      -0.256926  0.556994  -0.461272 6.44603e-01
## ENSMUSG00000000037   49.23586       0.378460  0.885650   0.427324 6.69143e-01
## ENSMUSG00000000049    7.98789      -0.843936  1.373878  -0.614273 5.39035e-01
## ENSMUSG00000000056 1981.00402      -0.638294  0.141439  -4.512864 6.39580e-06
##                          padj
##                    <numeric>
## ENSMUSG00000000001 0.608828033
## ENSMUSG00000000028 0.933364969
## ENSMUSG00000000037 0.940835275
## ENSMUSG00000000049 0.902491625
## ENSMUSG00000000056 0.000779972
##  [ reached getOption("max.print") -- omitted 6 rows ]
```

If we look at the results names we can relate this to the model equation:

$$expression = \beta_0 + \beta_1 \cdot TimePoint_{d33} + \beta_2 \cdot Status_{Infected} + \beta_3 \cdot TimePoint_{d33} \cdot Status_{Infected}$$

```
resultsNames(ddsObj.interaction)
```

```
## [1] "Intercept"                       "TimePoint_d33_vs_d11"
## [3] "Status_Infected_vs_Uninfected" "TimePointd33.StatusInfected"
```

- β_0 = `Intercept`
- β_1 = `TimePoint_d33_vs_d11`
- β_2 = `Status_Infected_vs_Uninfected`
- β_3 = `TimePointd33.StatusInfected`

So, the results we've obtained are for the interaction term β_3. The adjusted p_values show whether or not the β_3 coefficient is reliably different from 0, i.e. how likely it is that there is an interaction between Status and TimePoint for each gene. The log2FoldChange shows the degree to which the main effects (β_1 and β_2) should be adjusted in order to compensate for this interaction, i.e. the bigger the fold change the greater the interaction between the two factors.

9. Extracting specific contrasts from an interaction model

```r
sum(results.interaction.11$padj < 0.05, na.rm = TRUE)
```

```
## [1] 1072
```

Number of genes with padj $<$ 0.05 for Test v Control at day 33:

```r
sum(results.interaction.33$padj < 0.05, na.rm = TRUE)
```

```
## [1] 2782
```

We can see that there is a strong difference in the effects of infection on gene expression between days 11 and 33.

Exercise 7

Let's investigate the uninfected mice

1. Extract the results for d33 v d11 for Infected mice.

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

```r
sum(results.int.Inf$padj < 0.05, na.rm = TRUE)
```

```
## [1] 1134
```

2. Extract the results for d33 v d11 for Uninfected mice.

```r
results.int.Uninf <- results(ddsObj.interaction,
                             name = "TimePoint_d33_vs_d11",
                             alpha = 0.05)
```

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

```r
sum(results.int.Uninf$padj < 0.05, na.rm = TRUE)
```

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

Is this remarkable?

There's only 1 significant gene, however, perhaps this is not surprising as these samples are all from normal adult mouse brain tissue, just that in one group the mice were a few weeks older.

Do these results suggest another approach to analysing this data set?

Whilst we see a great difference in the infected mice between the two time points, there is apparently no difference in the gene expression in the brains of the two groups of uninfected mice. Perhaps we don't need to treat the two control time points as separate groups and could just consider 3 experimental conditions:

- Control (6 samples)
- Acute Infected (3 samples)
- Chronic Infected (3 samples)

This would reduce the number of coeffients we are estimating by 25% and might increase our statistical power, however, this decision would need to be based on both our biological knowledge and our analysis of the data.