Introduction to Bulk RNAseq data analysis

Initial exploration of RNA-seq data - solutions

Last modified: 21 Apr 2021

Contents

Data

```
library(tximport)
library(DESeq2)
library(tidyverse)
library(ggfortify)
```

Exercise 1

> We have loaded in the raw counts here. These are what we need for the differential expression analysis. For other investigations we might want counts normalised to library size. `tximport` allows us to import "transcript per million" (TPM) scaled counts instead.
>
> 1. Create a new object called `tpm` that contains length scaled TPM counts. You will need to add an extra argument to the command. Use the help page to determine how you need to change the code: `?tximport`.

```
tpm <- tximport(files, type = "salmon", tx2gene = tx2gene,
                countsFromAbundance = "lengthScaledTPM")
```

```
## reading in files with read_tsv
```

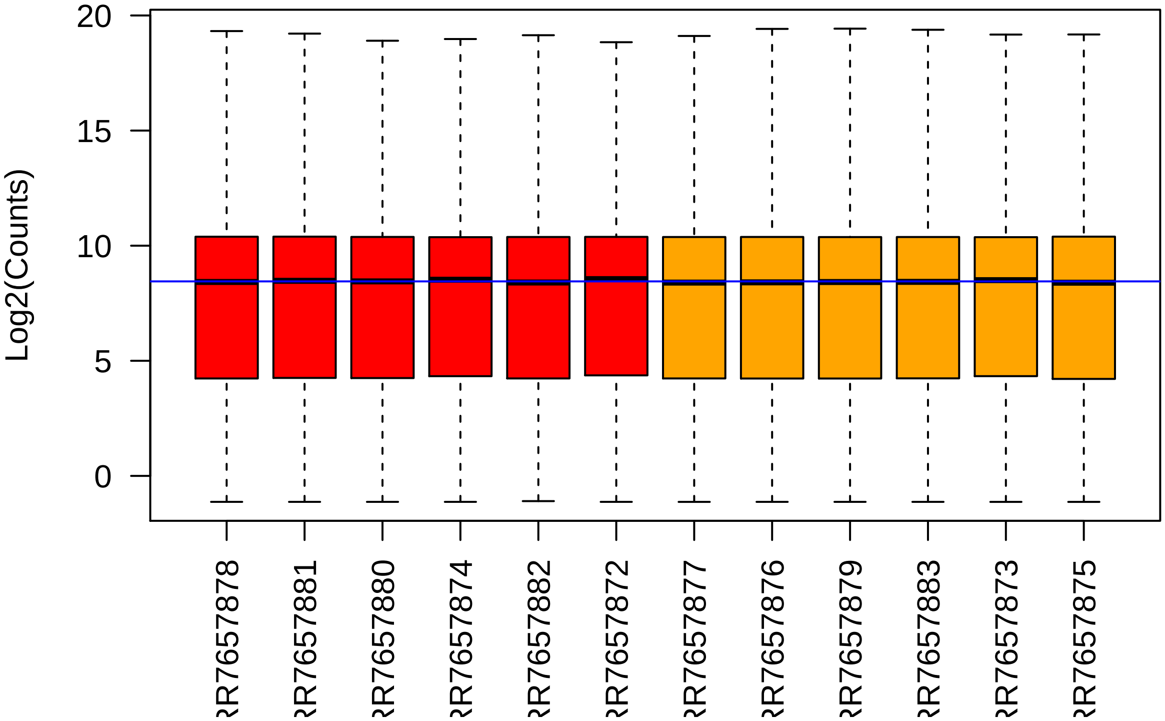
```
## 1 2 3 4 5 6 7 8 9 10 11 12
## summarizing abundance
## summarizing counts
## summarizing length
```

Exercise 2

> 1. Use the `DESeq2` function `rlog` to transform the count data. This function also normalises for library size.
> 2. Plot the count distribution boxplots with this data. How has this effected the count distributions?

```
rlogcounts <- rlog(filtCounts)

# Check distributions of samples using boxplots
boxplot(rlogcounts,
        xlab="",
        ylab="Log2(Counts)",
        las=2,
        col=statusCols)
# Let's add a blue horizontal line that corresponds to the median logCPM
abline(h=median(as.matrix(rlogcounts)), col="blue")
```

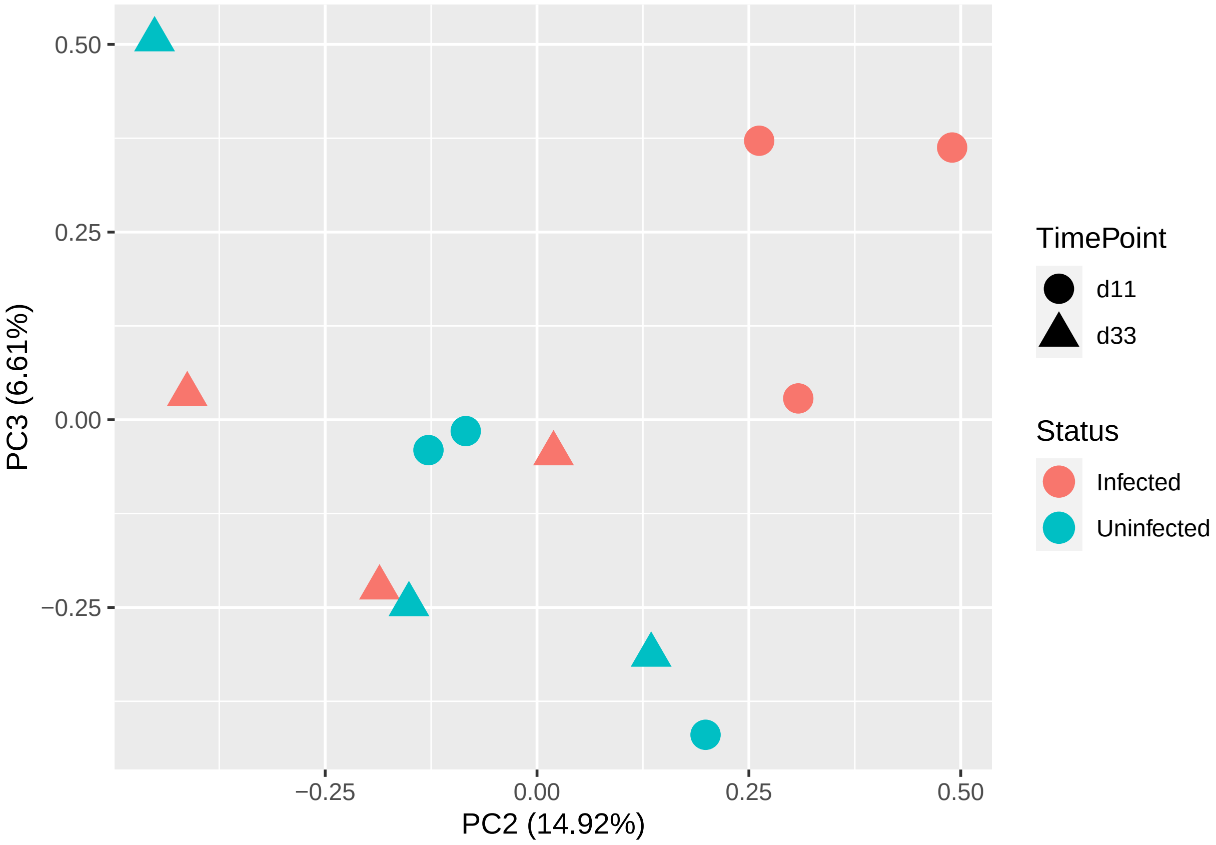


Exercise 3

The plot we have generated shows us the first two principle components. This shows us the relationship between the samples according to the two greatest sources of variation. Sometime, particularly with more complex experiments with more than two experimental factors, or where there might be confounding factors, it is helpful to look at more principle components.

1. Redraw the plot, but this time plot the 2nd principle component on the x-axis and the 3rd prinicple component on the y axis. To find out how to do the consult the help page for the `prcomp` data method for the `autoplot` function: `?autoplot.prcomp`.

```
autoplot(pcDat,
         data = sampleinfo,
         colour="Status",
         shape="TimePoint",
         x=2,
         y=3,
         size=5)
```


0.50
0.25
0.00
−0.25
PC3 (6.61%)
−0.25
0.00
0.25
0.50
PC2 (14.92%)
TimePoint
d11
d33
Status
Infected
Uninfected