A brief introduction to differential expression analysis with DESeq2

Peter Carbonetto

September 26, 2023

This is only a brief tutorial on DESeq2. For a much more in-depth introduction to DESeq2, I strongly recommend the vignette included in the DESeq2 R package.

Disclaimer: This tutorial may contain bugs or mistakes in the text. Please report any mistakes or bug fixes by posting a GitHub Issue or, better yet, by submitting a pull request.

Initial setup

Run this line of code to check that you have a heartbeat (*i.e.*, a working virtual machine):

```
sample(1000000,1)
```

This step is optional, but I run this every time in Jupyter notebook or Google Colab to make the code outputs look like they do in RStudio:

```
options(jupyter.rich_display = FALSE)
```

Install the packages from CRAN and Bioconductor that we will use to run the differential expression analysis. Note this may take a few minutes to complete as DESeq2 depends on many other packages that will also need to be installed.

```
install.packages("ashr")
install.packages("BiocManager")
BiocManager::install("DESeq2")
```

```
library(DESeq2)
library(ashr)
```

Now download and import the RNA-seq data into R. The data are from a recent paper studying transcriptional responses to sustained hypoxia.

```
download.file("https://raw.githubusercontent.com/pcarbo/rnaseq_tutorial/main/hypoxia.RData",
"hypoxia.RData")
```

```
load("hypoxia.RData")
```

A quick look at the data

If you were successful, you should have two new objects in your environment containing the RNA-seq data.

```
ls()
```

The data consist of RNA-seq read counts measured in two conditioins (normoxia and hypoxia) for 3 ENH5-knockout and 3 wild-type clones.

```
samples
```

The RNA-seq data are counts stored in a matrix with rows corresponding to genes and columns corresponding to samples.

```
class(counts)
nrow(counts)
ncol(counts)
head(counts,n = 10)
```

For the moment, let's focus on the data from the wild-type mice:

```
i <- which(samples$Line == "WT")
counts <- counts[,i]
samples <- samples[i,]
dim(counts)
samples
```

What is a DE analysis?

Fundamentally, in order to study the effect of hypoxia on expression, we would like to compare the read counts in the "hypoxia" condition to the read counts in the "normoxia" condition. It is helpful to take a few examples to see what these comparisons look like. Take the gene *EGNL3:

```
pdat <- data.frame(Treatment = samples$Treatment,
                   EGLN3 = counts["EGLN3",])
pdat
```

```
options(repr.plot.width = 4,repr.plot.height = 4.5,repr.plot.res = 175)
boxplot(EGLN3 ~ Treatment,data = pdat,boxwex = 0.25)
```

```
e1 <- mean(subset(pdat,Treatment == "Hypoxia")$EGLN3)
e0 <- mean(subset(pdat,Treatment == "Normoxia")$EGLN3)
e1
e0
log2(e1/e0)
```

This gene was actually highlighted in the paper because of its very large response to hypoxia, and indeed from a back-of-the-envelope calculation we got a very large LFC. However, most genes chosen at random will see changes to expression, both up and down, but the changes are more ambiguous. Let's take another example:

```
pdat <- data.frame(Treatment = samples$Treatment,
                   EGLN3 = counts["TNFRSF12A",])
boxplot(EGLN3 ~ Treatment,data = pdat,boxwex = 0.25)
```

```
e1 <- mean(subset(pdat,Treatment == "Hypoxia")$EGLN3)
e0 <- mean(subset(pdat,Treatment == "Normoxia")$EGLN3)
e1
e0
log2(e1/e0)
```

A central question is *which genes* have changes that are *significantly different* between the two conditions. We will use DESeq2 to answer this question more systematically. Again, we will focus on changes in hypoxia vs. normoxia in the WT mice.

A DE analysis using DESeq2

The first step in analyzing data using DESeq2 is to convert the data to an encoding that is convenient for DESeq2:

```
dat <- DESeqDataSetFromMatrix(counts,samples,~Treatment)
```

```
print(dat)
```

Once we have set up our data for DESeq2, analyzing the data using DESeq2 is quite straightforward:

```
deseq <- DESeq(dat)
```

Having run this analysis, we now have more rigorous LFC estimates:

```
lfc <- coef(deseq)[,"Treatment_Hypoxia_vs_Normoxia"]
hist(lfc,n = 64)
lfc[which(lfc > 8)]
```

We can extract other statistics such as standard errors and p-values using the `results()` function:

```
res <- results(deseq,alpha = 0.05,contrast = list("Treatment_Hypoxia_vs_Normoxia"))
```

```
head(res)
```

```
res["EGLN3",]
```

```
res["TNFRSF12A",]
```

With these statistics we can easily generate a "volcano plot" showing the LFC estimate on the X axis and some measure of support or significance on the Y axis (e.g., a p-value).

```
pdat <- data.frame(lfc = res$log2FoldChange,
                   pval = -log10(res$padj))
pdat <- transform(pdat,pval = pmin(50,pval))
plot(pdat$lfc,pdat$pval,pch = 20,
     xlab = "LFC",ylab = "-log10 p-value",
     main = "hypoxia vs. normoxia")
```

```
subset(res,log2FoldChange > 6 & padj < 1e-5)
```

A brief behind-the-scenes look

DESeq2 has two features that are important to most DE analyses:

One, it can account for different *sequencing depths* among the RNA-seq samples:

```
sizeFactors(deseq)
```

Two, it accounts for *overdispersion*—that is, additional variation that cannot be captured by the Poisson distribution:

```
summary(dispersions(deseq))
```

Improving the LFC estimates using shrinkage

A nice feature of DESeq2 is that it provides an additional function `lfcShrink` for "shrinking" the LFC estimates—essentially, borrowing information across all genes to reduce the "noise" in the estimates. Let's see how doing this improves our analysis.

```
res_shrunk <- lfcShrink(deseq,coef = "Treatment_Hypoxia_vs_Normoxia",
                        type = "ashr",svalue = TRUE)
```

```
head(res_shrunk)
```

Let's now compare the LFC estimates before and after shrinkage:

```
pdat <- data.frame(est = coef(deseq)[,"Treatment_Hypoxia_vs_Normoxia"],
                   shrunk = res_shrunk$log2FoldChange)
plot(pdat$est,pdat$shrunk,pch = 20,
     xlab = "original estimate",
     ylab = "shrunken estimate")
abline(a = 0,b = 1,lty = "dashed",col = "magenta")
```

Here is what the volcano plot looks like with the shrunken estimates. (It is recommended to use the s-values provided by `ashr` instead of the p-values.)

```
pdat <- data.frame(lfc = res_shrunk$log2FoldChange,
                   sval = -log10(res_shrunk$svalue))
pdat <- transform(pdat,sval = pmin(40,sval))
plot(pdat$lfc,pdat$sval,pch = 20,
     xlab = "shrunken LFC",ylab = "-log10 s-value")
```

"Blunting" of hypoxia response due to deletion of the *ENH5* enhancer

To practice our newly acquired skills, let's redo the steps of the DESeq2 analysis using the RNA-seq data from the knockout (KO) mice.

```
load("hypoxia.RData")
i <- which(samples$Line == "KO")
counts <- counts[,i]
samples <- samples[i,]
dat_ko <- DESeqDataSetFromMatrix(counts,samples,~Treatment)
deseq_ko <- DESeq(dat_ko)
res_ko <- results(deseq_ko,alpha = 0.05,contrast = list("Treatment_Hypoxia_vs_Normoxia"))
```

Reassuringly, many of the top upregulated genes in the knockout mice are the same as what we saw before in the wild-type mice:

```
subset(res_ko,log2FoldChange > 8)
```

To reproduce one of the main results of the paper, compare the LFC estimates from our DESeq2 analysis of the wild-type and knockout samples:

```
pdat <- data.frame(wt = res$log2FoldChange,
                   ko = res_ko$log2FoldChange)
fit <- lm(ko ~ wt,pdat)
coef(fit)
```

```
plot(pdat$wt,pdat$ko,pch = 20,xlab = "wild-type",ylab = "knockout")
abline(a = coef(fit)[1],b = coef(fit)[2],col = "magenta",lty = "dashed")
```

Compare this plot to Fig. 3 of the paper.