How to extract results for different comparisons in DESeq2 - a cheatsheet

Author: Adam Reid
Date: 29th November 2023

Background

Here we will go through the most common experimental designs and how to extract all the most common comparisons from them. There are multiple ways to extract your results, which can be confusing, but for the more common experimental designs, there are standard approaches. Note that the terminology can also be confusing and is not always used consistently. For a more detailed explanation of why some of these things are the way they are, read the [Linear Models walkthrough](#) from the course and the *?results* documentation in R.

Extracting particular comparisons for particular designs

1. Single factor (Status), two levels (Uninfected, Infected)

```
design = as.formula(~Status)

results = results(dds)
```

There is only one way to make a comparison, so the default option will return log2 fold changes for the second level over the first (Infected over Uninfected). This is equivalent to:

```
results = results(dds, contrast = c("Status", "Infected", "Uninfected"))
```

2. Single factor (Treatment), three levels (None, Drug1, Drug2)

```
design = as.formula(~Treatment)
```

The default comparison returns log2 fold changes for *Drug2* over *None*.

```
results = results(dds)
```

This is equivalent to:

```
results = results(dds, contrast = c("Treatment", "Drug2", "None"))
```

To get the change in gene expression due to *Drug1*:

```
results = results(dds, contrast = c("Treatment", "Drug1", "None"))
```

The get the difference in effect between drugs:

```
results = results(dds, contrast = c("Treatment", "Drug2",
"Drug1"))
```

3. Two factors, each with two levels (Status: Uninfected, Infected; TimePoint: d11, d33), no interaction term.

Here we assume that differences between timepoints are not affected by status and differences between status are not affected by time point. This is also known as *blocking*, and is frequently used to make comparisons where we want to control for a *nuisance factor*. We know a factor has an effect, but we want to exclude it. For instance, we might know that male and female mice respond differently to infection, and we used a mixture of these in the experiment, but we want to look at the effect beyond that driven by sex. In this case, we include sex in the design but don't include it when we extract the results. Because the effect of either factor on the other is not influenced by the particular levels we are comparing (or we want ot exclude this), it does not matter that the intercept represents a particular level of each factor. This only becomes important when using an interaction term.

```
design = as.formula(~Status + TimePoint)
```

At this point check to see what the coefficients are called:

```
resultsNames(dds)
```

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

To get log2 fold changes for *d33* compared to *d11* (assuming Status has no effect):

```
results = results(dds, name = "TimePoint_d33_vs_d11")
```

To get log2 fold changes for *Infected* versus *Uninfected* (assuming TimePoint has no effect):

```
results = results(dds, name = "Status_Infected_vs_Uninfected")
```

4. Two factors, each with two levels (Status: Uninfected, Infected; TimePoint: d11, d33), and an interaction term.

Here you need to be clear whether you are comparing against the intercept (and therefore one set of levels for each factor) or whether you use the interaction term to switch the levels. This is because we are now allowing there to be differences between TimePoints which are driven by Status and differences between levels of Status which are driven by TimePoint.

```
design = as.formula(~Status + TimePoint + Status:TimePoint)
```

At this point check to see what the coefficients are called:

```
resultsNames(dds)

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

Here the intercept represents samples which are *TimePointd11* and *StatusUninfected* - the opposite of the levels in the interaction term coefficient (the last one in the list).

Therefore, to get log2 fold changes for Infected over Uninfected in day 11 samples can just supply one coefficient, which will be compared against the intercept:

```
results(dds, name="Status_Infected_vs_Uninfected")
```

To get log2 fold changes for day 33 over day 11 in Uninfected samples we can choose the coefficient representing the difference between TimePoint levels and (by default) compare that against the intercept, which represents Uninfected samples at day 11:

```
results(dds, name="TimePoint_d33_vs_d11")
```

To get log2 fold changes for Infected over Uninfected in day 33 samples, we now use the contrast argument, because we need to refer to multiple coefficients. We include the interaction term which tells us the effect of TimePoint33 versus TimePoint11 amongst Infected samples.

```
results(dds, contrast =
list(c("Status_Infected_vs_Uninfected","TimePointd33.StatusInfecte
d")))
```

To get log2 fold changes for d33 over d11 in Infected samples, we again use the contrast argument, because we need to refer to multiple coefficients.

```
results(dds, contrast =
list(c("TimePoint_d33_vs_d11","TimePointd33.StatusInfected")))
```

You can also specify just the interaction term and this will compare the interaction term against the intercept, giving significant log2 fold change for genes which have an interaction effect, i.e. where the Status affects the level of expression differently at different TimePoints.

```
results(dds, name="genotypeII.conditionB")
```

5. Using a grouping variable

Sometimes it is most convenient to compare different levels of different factors as if they were simply all different levels of one factor. We might compare Uninfected samples at day 11 versus Infected samples at day 33. This can be achieved by making a new variable in the colData called, for example, TimePointStatus, with values such as Uninfectedd11, Infectedd11, Uninfectedd33 and Infectedd33. This allows us to use a single factor design and then compare each pair of conditions, e.g.

```
colData(dds)$TimePointStatus = paste(colData(dds)$TimePoint,
colData(dds)$Status, sep="")

design = as.formula(~TimePointStatus)
```

Get log2 fold changes for Infected day11 samples over Uninfected day 33 samples:

```
results = results(dds, contrast = c("TimePointStatus",
"Infectedd11", "Uninfectedd33"))
```

Summary

The aim of this "cheatsheet" is to point you towards a suitable way of extracting the results you need, with hopefully a bit of clarification about why this works. There are of course other more complicated designs. The DESeq2 manual discusses some of these specifically, but if it is not clear what to do for your particular experimental design, consult a friendly bioinformatician, or more likely a card-carrying statistician.