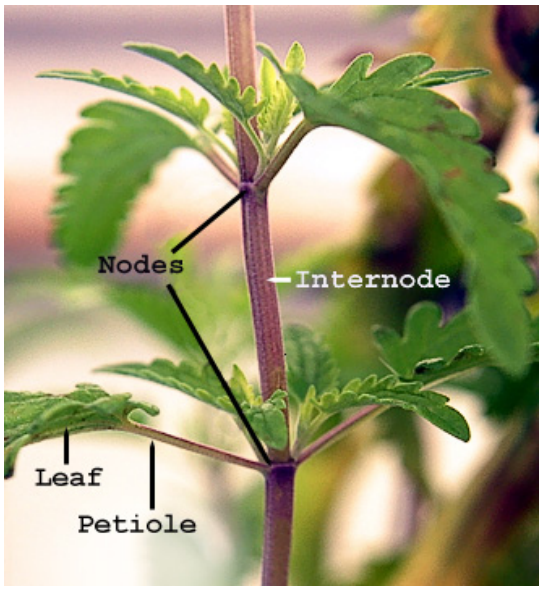
DESeq2 R practical

(Original author: Edouard Severing)

In this practical we will work with RNA-Seq read counts for maize measured in different internodes



Make sure DESeq2 is installed. In R run these two lines to do so:

```
source("http://bioconductor.org/biocLite.R")
biocLite("DESeq2")
```

To load the DESeq2 library and a set of custom functions, run: (choose **n** to update none)

```
library(DESeq2)
source("http://www.bioinformatics.nl/courses/BIF-30806/DEseq2Exercise.R")
```

Experiment

In our experiment we are interested in the differences between the first and fourth internodes of maize plants. We hope that comparing the gene expression between the internodes can provide useful information. To this end we have generated RNA-Seq data for three biological replicates of the first internode and three biological replicates of the fourth internode of maize plants.

The read count data is already available; you can load it by typing the following R command:

```
internode_data=read.table(
      "http://www.bioinformatics.nl/courses/BIF-30806/maize_e3.table",
      row.names=1, header=TRUE, sep ="\t")
```

Explanation: **`row.names = 1`** indicates that the first column in the file contains the name of the rows. **`header = TRUE`** indicates that file contains a header which in this case contains the names of the columns. **`sep = "\t"`** is there to indicate that the column fields are separated by tabs.

Let's look at a bit closer at the dimensions of the loaded RNA-Seq count data by running:
`dim(internode_data)`

two values will be displayed (after the **`[1]`**). These represent the number of rows and the number of columns in our dataset. What do you think the rows represent and what the columns? With the **`class`** function you can learn that the data type of **`internode_data`** is **`data.frame`**:
`class(internode_data)`

By running:
`colnames(internode_data)`

you will obtain a list with the column names. In this case all the columns except the last one correspond to RNA-Seq samples, the last column contains gene annotations.

This command will change the column names of the table to shorter ones:
`colnames(internode_data) = c("1-1","1-2","1-3","4-1","4-2","4-3","Annotation")`

i.e. **`1-3`** stands for the first internode, replicate 3.

By running:
`rownames(internode_data)`

you will obtain a list with all the row names. Row names correspond to the genes.

By running:
`summary(internode_data)`

you get a summary of the data in the different columns.

In order to get an indication of the total number of counts in each sample you can run:
`apply(internode_data[,1:6], 2, sum)`

With this command we sum all counts per column. The first parameter is set to `internode_data[,1:6]` and not just `internode_data` because we can only have the sum for column 1 to 6. Column number 7 contains annotations.

Cleaning

Now we would like to remove all genes that are not very informative. In this specific case we will remove genes that have in none of the samples more than 10 counts (arbitrary chosen threshold).

Run:
`mx = apply( internode_data[,1:6], 1, max )`

That command will create a list containing the maximum read count (over all our 6 samples) for each gene. Again we use `internode_data[,1:6]` because we must exclude column 7.

Next we will make a new table that only contains rows for which the maximum count is greater than 10:

```
internode_data = internode_data[ mx > 10, ]
```

Use `dim(internode_data)` to determine how many genes you have left in your set.

DESeq2

In order to continue we need make a data-object that DESeq2 can use for performing differential expression analysis.

Because we will work with the first six columns that contain the count data, we will first make a new **`data.frame`** with only that data:

```
count_data = internode_data[,1:6]
```

Run these commands to create a **`DESeqDataSet`** data object:

```
condition = factor(c("first","first","first","fourth","fourth","fourth" ))
col_data = DataFrame(condition)
dds = DESeqDataSetFromMatrix(count_data, col_data, ~condition)
```

The `col_data` parameter indicates that first three columns correspond to replicates from the first internode and the last three columns correspond to replicates from the fourth internode. The last parameter describes the design of the study.

Normalization

Before any comparison can be made between samples the counts have to be normalized. The reason for this is that the counts for a gene not only depend on its expression level but also on the depth of sequencing (total number of reads per experiment).

Run this to calculate the (linear) correction factors for each sample:

```
dds = estimateSizeFactors(dds)
```

The correction (size) factors can be retrieved using:

```
sizeFactors(dds)
```

In order to show you the importance of normalization: Type the following in the console:

```
norm_versus_non_norm( dds, 1, 2, left = 2, right = 8 )
```

This command calls a custom function from `DEseq2Exercise.R` script. It takes the first and second column (=sample) of the count dataset and generates two scatter plots, each dot represents the count of a gene in the first and second replicate for the first internode. The first scatter plot contains the non-normalized and the second plot the normalized counts. Do you see why normalization is important?

Cluster analysis of the samples

It is very important to check whether your samples cluster as you expect them to. You generally expect the gene expression values to be more similar between replicates than between samples from different conditions. In this section you will perform a simple and quick clustering analysis.

Before we can cluster, we first have to transform the counts to get a more 'normal' distribution; this avoids highly expressed genes dominating the results. Type the following command in the console:

```
rld = rlog(dds)
```

this creates a table in which the normalized counts are transformed to log counts. Compare the distributions of the values using these commands:

```
plot(density(assay(dds)[,1]), main="counts")
plot(density(assay(rld)[,1]), main="log counts")
```

To compare the six samples, we calculate the (euclidean) distances between the samples using their gene expression values. With this command you create a distance matrix:

```
dists = dist(t(assay(rld)))
```

With this command you plot a tree of the distances between the samples:

```
plot(hclust(dists))
```

This tree represents a hierarchical clustering of the samples. Do they cluster as expected?

Gene-specific dispersions

In order to detect differential expression DESeq2 has to estimate the expression variance for each gene. DESeq2 assumes that gene counts within conditions follow the negative binomial distribution. According to this model the variance in expression of a gene depends on its mean expression-level as follows:

$\sigma^2 = s\mu + \alpha s^2 \mu^2$

The left term is the variance, which depends on the mean μ. In the formula s is a scaling factor that is constant for all genes in a sample/condition and α is called the dispersion. DESeq2 tries to determine the dispersion value for each gene from the normalized count data. It later will use the dispersions to determine the gene-expression variance for each gene so it can test for differential expression. To estimate the dispersions, run:

```
dds = estimateDispersions(dds)
```

With this command the gene-specific dispersion values are estimated over all samples.

Now type the following command in the console:

```
plotDispEsts(dds)
```

You should see a plot of the dispersion versus the expression level for all genes. The black dots are the dispersion values that were calculated per gene from the normalized count data. As you can see DESeq2 fitted a (red) line through the data. This assumes that the dispersion value is a function of the mean expression value. The dispersion value for each gene is subsequently adjusted towards the red line, giving the blue dots. The black dots with blue circles are dispersion outliers these are not adjusted.

Differential expression.

We have now arrived at the step where we can perform a differential expression analysis.

Type the following command in the console:

```
dds = nbinomWaldTest(dds)
```

This command will perform the differential expression tests between our two samples.

To get a table with differential expression values for the genes, type:

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

To see the top rows from the differential expression table, type the following command in the console:

```
head(res)
```

The **padj** column contains p-values that are adjusted for multiple testing. **BaseMean** lists the mean count values for the six samples and the **log2FoldChange** is the log2 of the fold change (obviously).

For a small number of genes no **padj** could be calculated, these have no value which will cause problems later, so set them to 1 with this command:

```
res$padj = ifelse(is.na(res$padj), 1, res$padj)
```

Next we are going to add our annotation back to the results table, type in the console:

```
res$annotation = internode_data[,7]
```

With this command we add a new column to results table **res**, which contains the annotation column (number 7) from the internode_data table. Confirm this using the **head** command.

Now we write an output a table that you can open in Excel later, type in the console:

```
write.table(res, col.names=NA, row.names=T, file ="internodes.tsv", sep ="\t")
```

With this command we write table `res` to disk with the `row.names` and `col.names`. We use tabs for separating fields (`sep = "\t"`). The file that is created is called: `internodes.tsv`

MA plot

To end this exercise we will make an MA plot. You can make the plot by typing:

```
plotMA(res, main="MA plot",ylim=c(-8,8),alpha=0.01)
```

Each point in an MA plot represents a gene. The x-coordinate corresponds to the mean expression of the gene, and the y-axis corresponds to the log2 fold change between the two conditions/tissues. All red points correspond to genes with that are differentially expressed according to the adjusted p-value threshold (alpha) of 0.01.

What can you say about the overall gene expression profiles of the first and fourth internodes, are they very similar?

Can you get a list of the 10 genes with the highest significant fold change?

How many genes are differentially expressed if you take a cut-off for the adjusted p-value of 0.01?