Introduction to DGE - ARCHIVED

View on GitHub

Approximate time: 60 minutes

Learning Objectives

- Understanding the different steps in a differential expression analysis in the context of DESeq2
- Executing the differential expression analysis workflow with DESeq2
- Constructing design formulas appropriate for a given experimental design
- Exploring the importance of dispersion during differential expression analysis, and using the plots of the dispersion values to explore assumptions of the NB model

Differential expression analysis with DESeq2

The final step in the differential expression analysis workflow is fitting the raw counts to the NB model and performing the statistical test for differentially expressed genes. In this step we essentially want to determine whether the mean expression levels of different sample groups are significantly different.

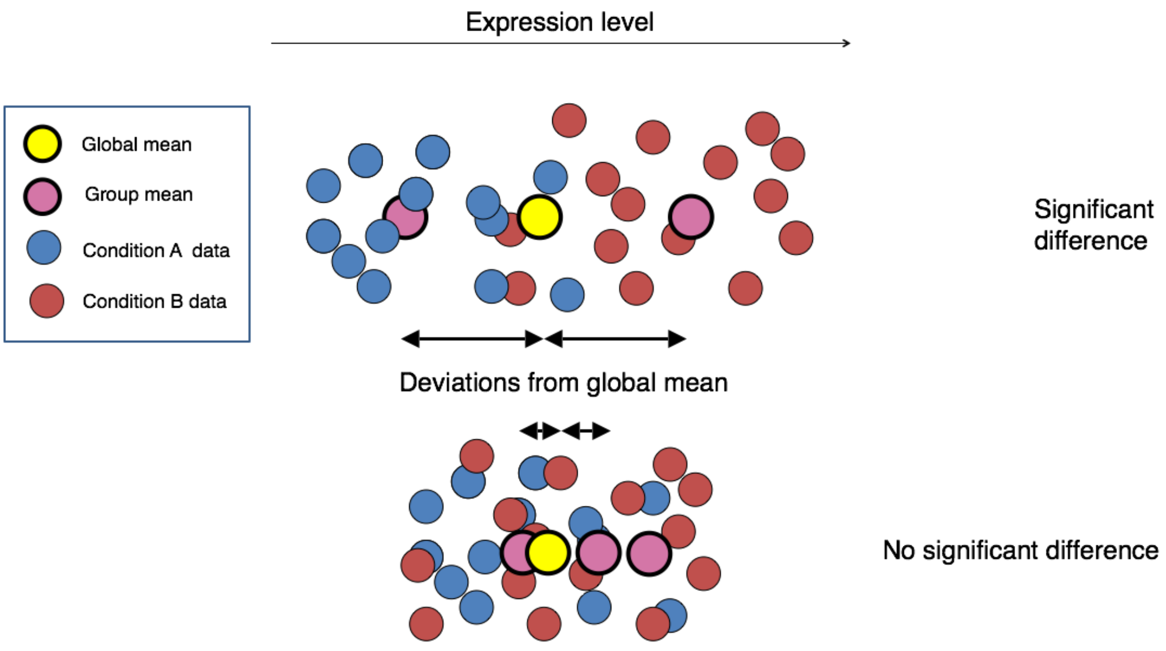
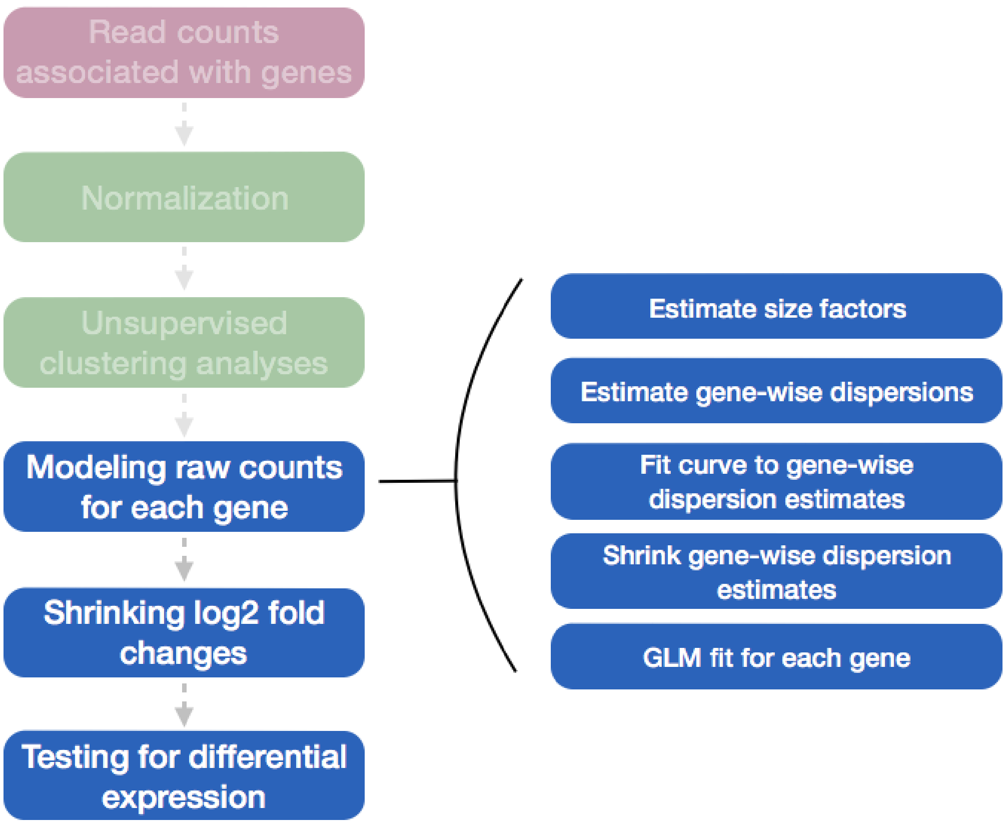


Image credit: Paul Pavlidis, UBC

The DESeq2 paper was published in 2014, but the package is continually updated and available for use in R through Bioconductor. It builds on good ideas for dispersion estimation and use of Generalized Linear Models from the DSS and edgeR methods.

Differential expression analysis with DESeq2 involves multiple steps as displayed in the flowchart below in blue. Briefly, DESeq2 will model the raw counts, using normalization factors (size factors) to account for differences in library depth. Then, it will estimate the gene-wise dispersions and shrink these estimates to generate more accurate estimates of dispersion to model the counts. Finally, DESeq2 will fit the negative binomial model and perform hypothesis testing using the Wald test or Likelihood Ratio Test.


Read counts associated with genes
Normalization
Unsupervised clustering analyses
Modeling raw counts for each gene
Shrinking log2 fold changes
Testing for differential expression
Estimate size factors
Estimate gene-wise dispersions
Fit curve to gene-wise dispersion estimates
Shrink gene-wise dispersion estimates
GLM fit for each gene

> **NOTE:** DESeq2 is actively maintained by the developers and continuously being updated. As such, it is important that you note the version you are working with. Recently, there have been some rather **big changes implemented** that impact the output. To find out more detail about the specific **modifications made to methods described in the original 2014 paper**, take a look at this section in the DESeq2 vignette.

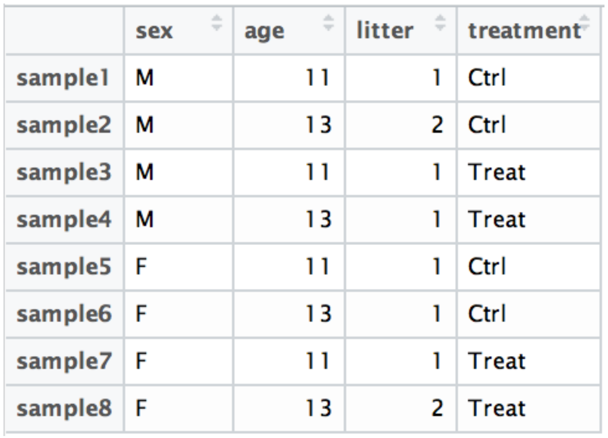
Running DESeq2

Prior to performing the differential expression analysis, it is a good idea to know what **sources of variation** are present in your data, either by exploration during the QC and/or prior knowledge. Once you know the major sources of variation, you can remove them prior to analysis or control for them in the statistical model by including them in your **design formula**.

Design formula

A design formula tells the statistical software the known sources of variation to control for, as well as, the factor of interest to test for during differential expression testing. For example, if you know that sex is a significant source of variation in your data, then `sex` should be included in your model. **The design formula should have all of the factors in your metadata that account for major sources of variation in your data. The last factor entered in the formula should be the condition of interest.**

For example, suppose you have the following metadata:



	sex	age	litter	treatment
sample1	M	11	1	Ctrl
sample2	M	13	2	Ctrl
sample3	M	11	1	Treat
sample4	M	13	1	Treat
sample5	F	11	1	Ctrl
sample6	F	13	1	Ctrl
sample7	F	11	1	Treat
sample8	F	13	2	Treat

If you want to examine the expression differences between treatments, and you know that major sources of variation include `sex` and `age`, then your design formula would be:

```
design <- ~ sex + age + treatment
```

The tilde (~) should always proceed your factors and tells DESeq2 to model the counts using the following formula. Note the **factors included in the design formula need to match the column names in the metadata**.

Exercises

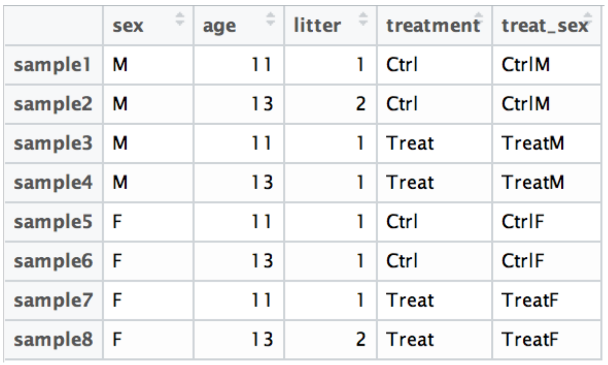
1. Suppose you wanted to study the expression differences between the two age groups in the metadata shown above, and major sources of variation were `sex` and `treatment`, how would the design formula be written?
2. Based on our **Mov10** `metadata` dataframe, which factors could we include in our design formula?
3. What would you do if you wanted to include a factor in your design formula that is not in your metadata?

Complex designs

DESeq2 also allows for the analysis of complex designs. You can explore interactions or 'the difference of differences' by specifying for it in the design formula. For example, if you wanted to explore the effect of sex on the treatment effect, you could specify for it in the design formula as follows:

```
design <- ~ sex + age + treatment + sex:treatment
```

Since the interaction term `sex:treatment` is last in the formula, the results output from DESeq2 will output results for this term. Alternatively, as recommended in the DESeq2 vignette, we could create a new factor variable in our metadata based on the two interaction factors as shown in the table below:



	sex	age	litter	treatment	treat_sex
sample1	M	11	1	Ctrl	CtrlM
sample2	M	13	2	Ctrl	CtrlM
sample3	M	11	1	Treat	TreatM
sample4	M	13	1	Treat	TreatM
sample5	F	11	1	Ctrl	CtrlF
sample6	F	13	1	Ctrl	CtrlF
sample7	F	11	1	Treat	TreatF
sample8	F	13	2	Treat	TreatF

The design formula would be:

```
design <- ~ age + treat_sex
```

> **NOTE:** The design `~ sex + age + treatment + treat_sex` won't work (because the model can't be fit) because `treatment` and `treat_sex` are confounded (same goes for `sex`). Therefore, we drop the terms that went into `treat_sex` from the design formula.
>
> The testing method for using the combined `treat_sex` factor column will be slightly different, using the LRT method, which we will discuss later.

MOV10 DE analysis

Now that we know how to specify the model to DESeq2, we can run the differential expression pipeline on the **raw counts**.

To get our differential expression results from our raw count data, we only need to run 2 lines of code!

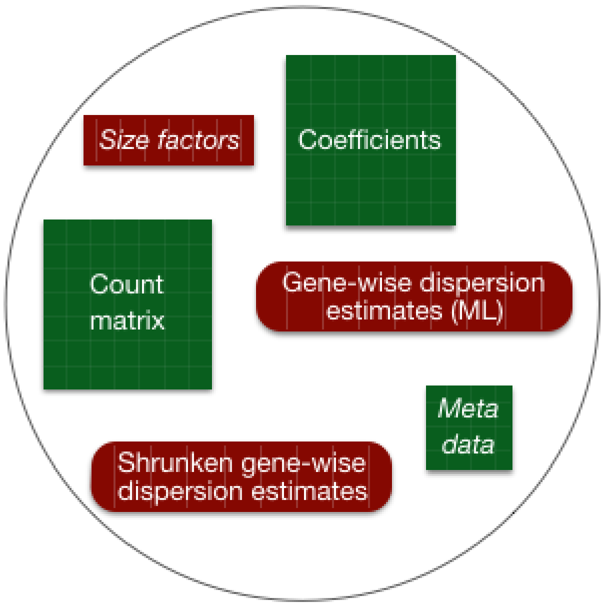
First we create a DESeqDataSet as we did in the 'Count normalization' lesson and specify the location of our raw counts and metadata, and input our design formula:

```
## Create DESeq object
dds <- DESeqDataSetFromMatrix(countData = data, colData = meta, design = ~ samplet
```

Then, to run the actual differential expression analysis, we use a single call to the function `DESeq()`.

```
## Run analysis
dds <- DESeq(dds)
```

By re-assigning the results of the function back to the same variable name (`dds`), we can fill in the `slots` of our `DESeqDataSet` object.


Size factors
Coefficients
Count matrix
Gene-wise dispersion estimates (ML)
Meta data
Shrunken gene-wise dispersion estimates

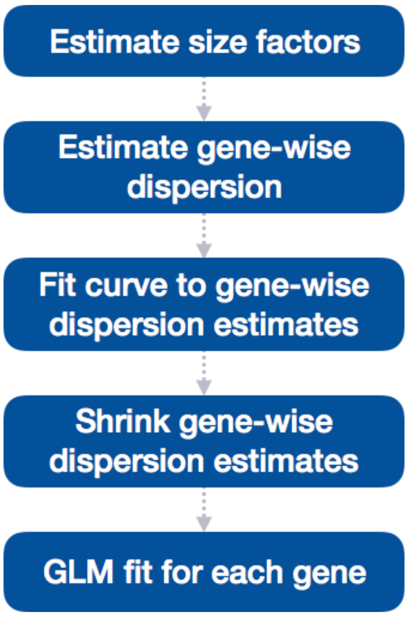
Everything from normalization to linear modeling was carried out by the use of a single function! This function will print out a message for the various steps it performs:

```
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
```

> **NOTE:** There are individual functions available in DESeq2 that would allow us to carry out each step in the workflow in a step-wise manner, rather than a single call. We demonstrated one example when generating size factors to create a normalized matrix. By calling `DESeq()` , the individual functions for each step are run for you.

DESeq2 differential gene expression analysis workflow

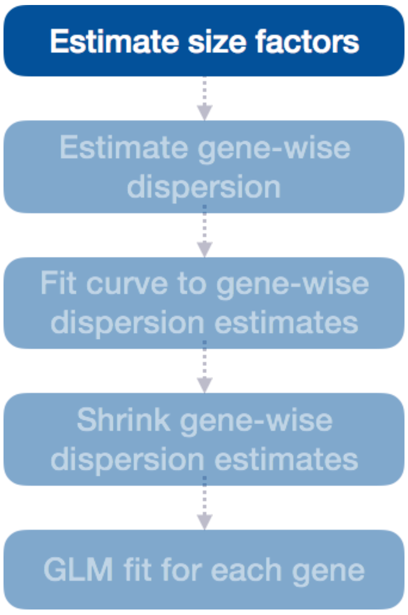
With the 2 lines of code above, we just completed the workflow for the differential gene expression analysis with DESeq2. The steps in the analysis are output below:


Estimate size factors
Estimate gene-wise dispersion
Fit curve to gene-wise dispersion estimates
Shrink gene-wise dispersion estimates
GLM fit for each gene

We will be taking a detailed look at each of these steps to better understand how DESeq2 is performing the statistical analysis and what metrics we should examine to explore the quality of our analysis.

Step 1: Estimate size factors

The first step in the differential expression analysis is to estimate the size factors, which is exactly what we already did to normalize the raw counts.


Estimate size factors
Estimate gene-wise dispersion
Fit curve to gene-wise dispersion estimates
Shrink gene-wise dispersion estimates
GLM fit for each gene

DESeq2 will automatically estimate the size factors when performing the differential expression analysis. However, if you have already generated the size factors using `estimateSizeFactors()` , as we did earlier, then DESeq2 will use these values.

To normalize the count data, DESeq2 calculates size factors for each sample using the *median of ratios method* discussed previously in the 'Count normalization' lesson.

MOV10 DE analysis: examining the size factors

Let’s take a quick look at size factor values we have for each sample:

```
## Check the size factors
sizeFactors(dds)

Mov10_kd_2 Mov10_kd_3 Mov10_oe_1 Mov10_oe_2 Mov10_oe_3 Irrel_kd_1 Irrel_kd_2 Irre
 1.5646728  0.9351760  1.2016082  1.1205912  0.6534987  1.1224020  0.9625632  0.7
```

These numbers should be identical to those we generated initially when we had run the function `estimateSizeFactors(dds)`. Take a look at the total number of reads for each sample:

```
## Total number of raw counts per sample
colSums(counts(dds))
```

How do the numbers correlate with the size factor?

Now take a look at the total depth after normalization using:

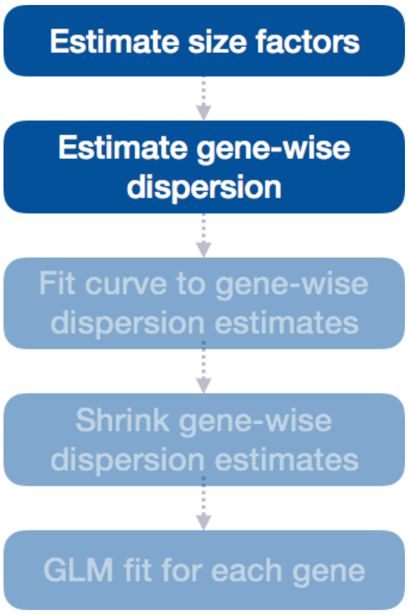
```
## Total number of normalized counts per sample
colSums(counts(dds, normalized=T))
```

How do the values across samples compare with the total counts taken for each sample?

> **NOTE:** It can be advantageous to calculate gene-specific normalization factors (size factors) to account for further sources of technical biases such as differing dependence on GC content, gene length or the like, and these can be supplied to DESeq2 instead of using the median of ratios method.

Step 2: Estimate gene-wise dispersion

The next step in the differential expression analysis is the estimation of gene-wise dispersions. Before we get into the details, we should have a good idea about what dispersion is referring to in DESeq2.


Estimate size factors
Estimate gene-wise dispersion
Fit curve to gene-wise dispersion estimates
Shrink gene-wise dispersion estimates
GLM fit for each gene

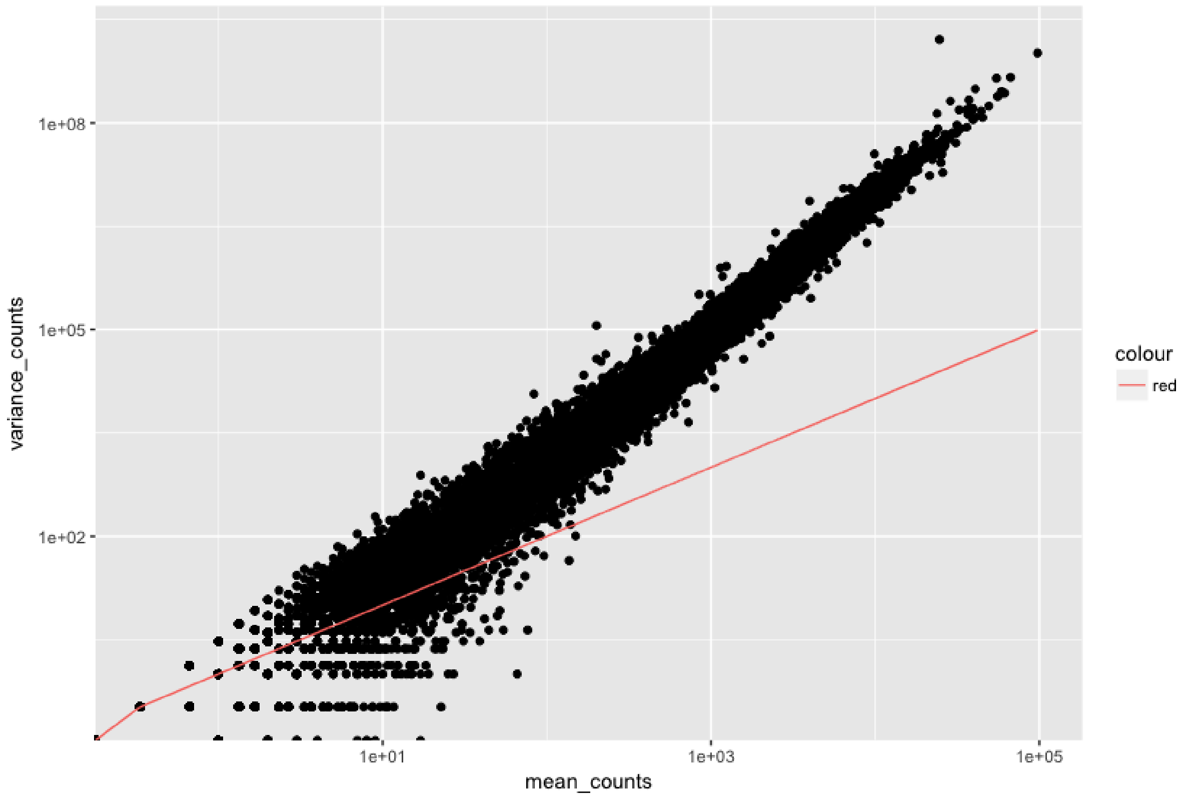
What is dispersion?

Dispersion is a measure of spread or variability in the data. Variance, standard deviation, IQR, among other measures, can all be used to measure dispersion. However, DESeq2 uses a specific measure of dispersion (α) related to the mean (μ) and variance of the data: `Var = μ + α*μ^2` . For genes with moderate to high count values, the square root of dispersion will be equal to the coefficient of variation (`Var / μ`). So 0.01 dispersion means 10% variation around the mean expected across biological replicates.

What does the DESeq2 dispersion represent?

The DESeq2 dispersion estimates are **inversely related to the mean** and **directly related to variance**. **Based on this relationship, the dispersion is higher for small mean counts and lower for large mean counts.** The dispersion estimates for genes with the same mean will differ only based on their variance. **Therefore, the dispersion estimates reflect the variance in gene expression for a given mean value.**

The plot of mean versus variance in count data below shows the variance in gene expression increases with the mean expression (each black dot is a gene). Notice that the relationship between mean and variance is linear on the log scale, and for higher means, we could predict the variance relatively accurately given the mean. However, **for low mean counts, the variance estimates have a much larger spread; therefore, the dispersion estimates will differ much more between genes with small means**.



How does the dispersion relate to our model?

To accurately model sequencing counts, we need to generate accurate estimates of within-group variation (variation between replicates of the same sample group) for each gene. With only a few (3-6) replicates per group, the **estimates of variation for each gene are often unreliable** (due to the large differences in dispersion for genes with similar means).

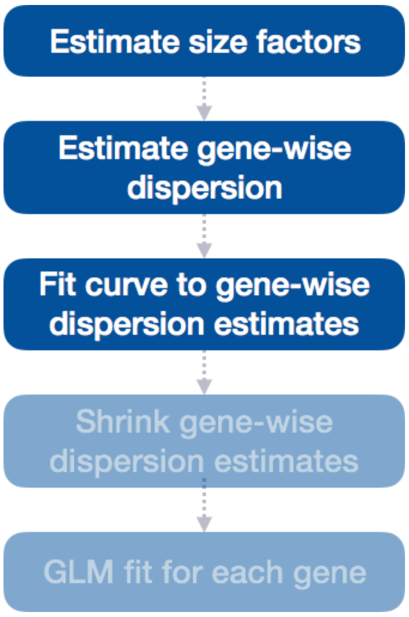
To address this problem, DESeq2 **shares information across genes** to generate more accurate estimates of variation based on the mean expression level of the gene using a method called 'shrinkage'. **DESeq2 assumes that genes with similar expression levels have similar dispersion.**

Estimating the dispersion for each gene separately:

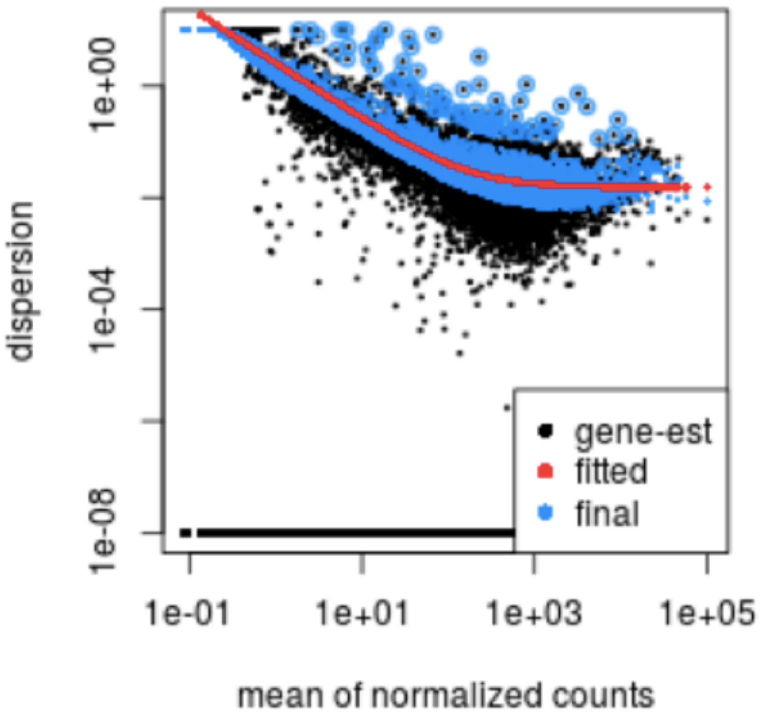
To model the dispersion based on expression level (mean counts of replicates), the dispersion for each gene is estimated using maximum likelihood estimation. In other words, **given the count values of the replicates, the most likely estimate of dispersion is calculated**.

Step 3: Fit curve to gene-wise dispersion estimates

The next step in the workflow is to fit a curve to the dispersion estimates for each gene. The idea behind fitting a curve to the data is that different genes will have different scales of biological variability, but, over all genes, there will be a distribution of reasonable estimates of dispersion.

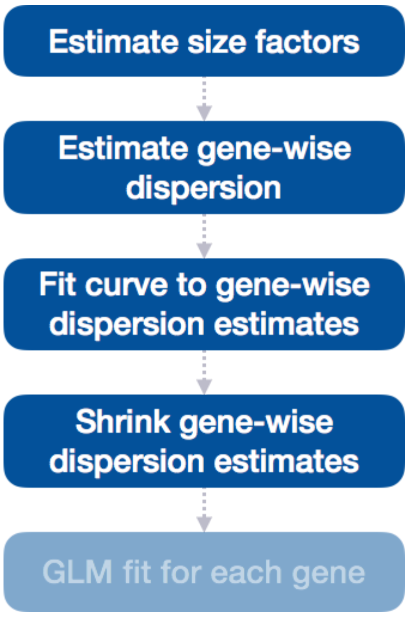

Estimate size factors
Estimate gene-wise dispersion
Fit curve to gene-wise dispersion estimates
Shrink gene-wise dispersion estimates
GLM fit for each gene

This curve is displayed as a red line in the figure below, which plots the estimate for the **expected dispersion value for genes of a given expression strength**. Each black dot is a gene with an associated mean expression level and maximum likelihood estimation (MLE) of the dispersion (Step 1).



Step 4: Shrink gene-wise dispersion estimates toward the values predicted by the curve

The next step in the workflow is to shrink the gene-wise dispersion estimates toward the expected dispersion values.

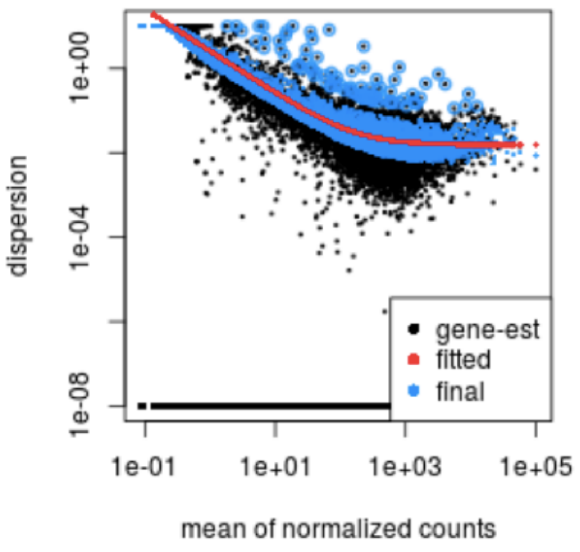

Estimate size factors
Estimate gene-wise dispersion
Fit curve to gene-wise dispersion estimates
Shrink gene-wise dispersion estimates
GLM fit for each gene

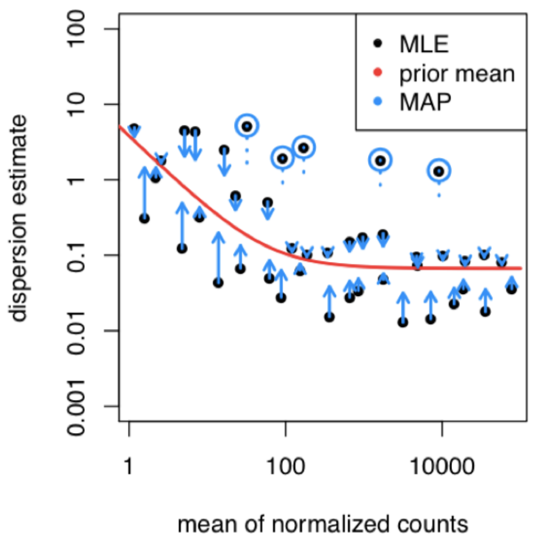
The curve allows for more accurate identification of differentially expressed genes when sample sizes are small, and the strength of the shrinkage for each gene depends on :

- how close gene dispersions are from the curve
- sample size (more samples = less shrinkage)

This shrinkage method is particularly important to reduce false positives in the differential expression analysis. Genes with low dispersion estimates are shrunken towards the curve, and the more accurate, higher shrunken values are output for fitting of the model and differential expression testing.

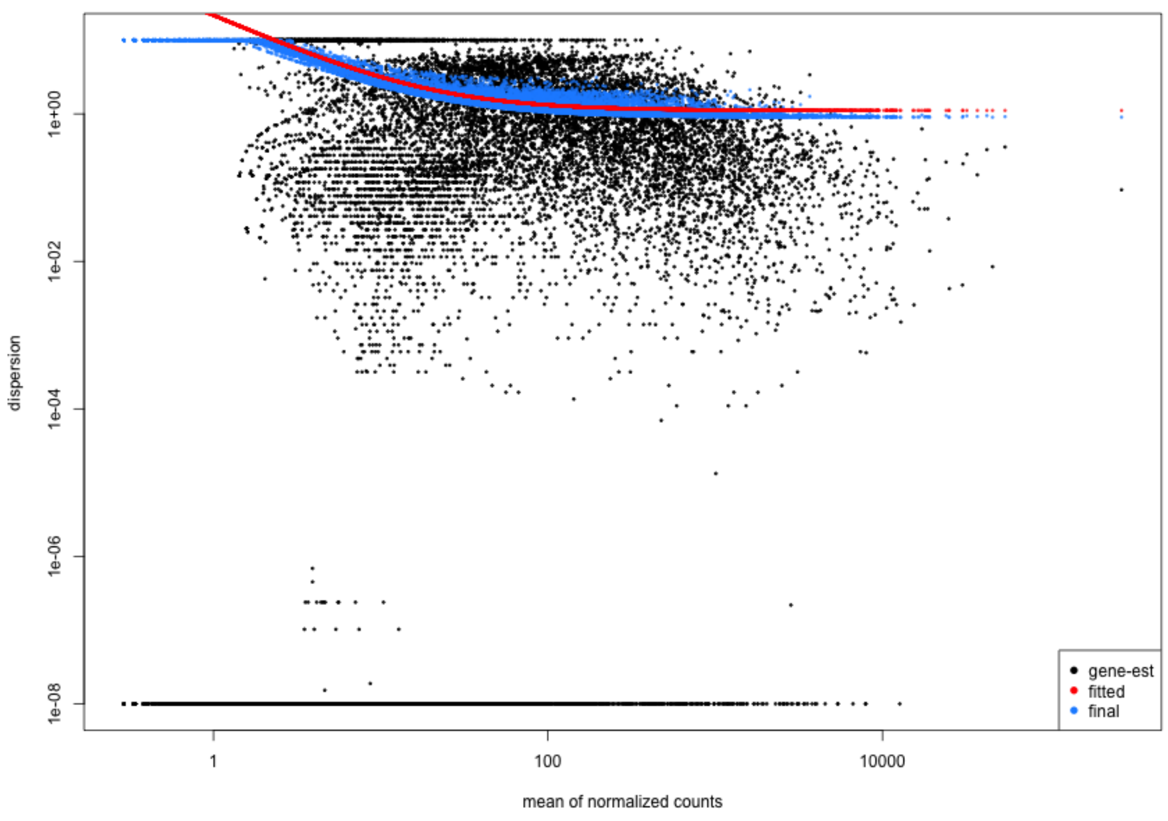
Dispersion estimates that are slightly above the curve are also shrunk toward the curve for better dispersion estimation; however, genes with extremely high dispersion values are not. This is due to the likelihood that the gene does not follow the modeling assumptions and has higher variability than others for biological or technical reasons [1]. Shrinking the values toward the curve could result in false positives, so these values are not shrunken. These genes are shown surrounded by blue circles below.

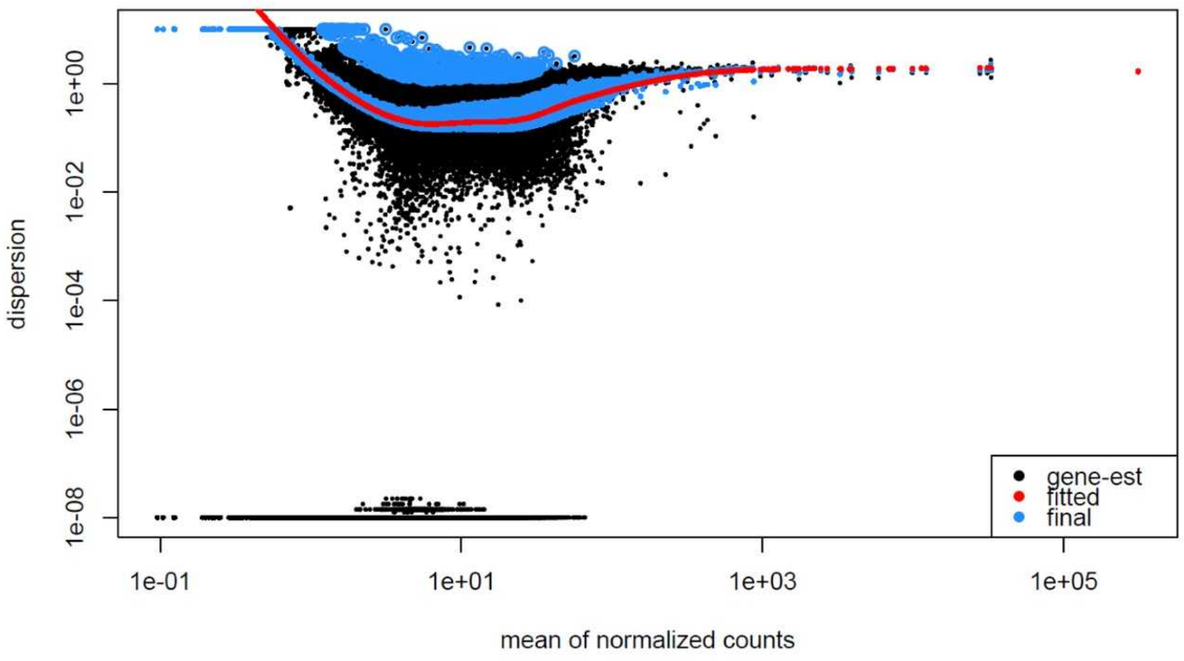




This is a good plot to examine to ensure your data is a good fit for the DESeq2 model. You expect your data to generally scatter around the curve, with the dispersion decreasing with increasing mean expression levels. If you see a cloud or different shapes, then you might want to explore your data more to see if you have contamination (mitochondrial, etc.) or outlier samples. Note how much shrinkage you get across the whole range of means in the `plotDispEsts()` plot for any experiment with low degrees of freedom.

Examples of **worrisome dispersion plots** are shown below:

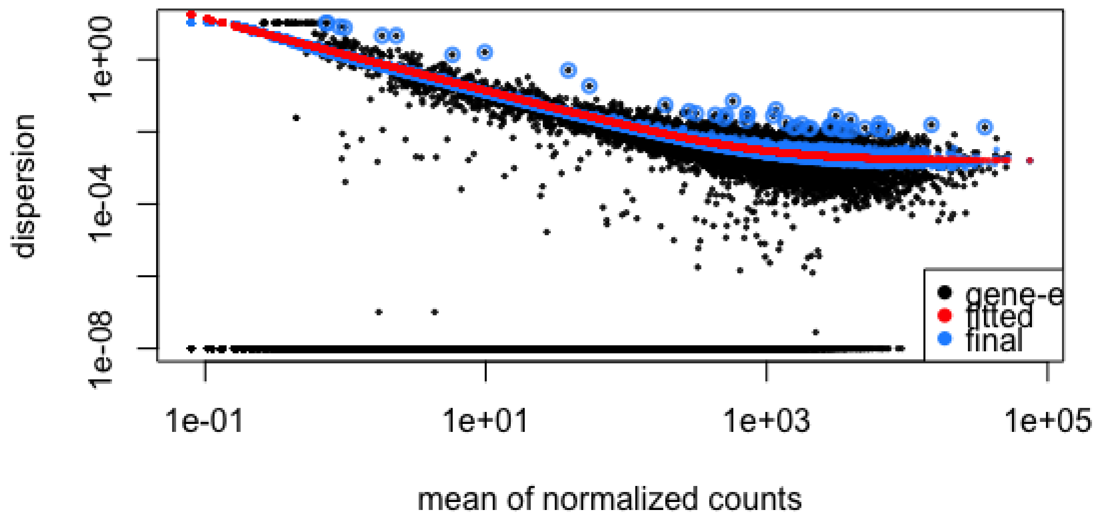




MOV10 DE analysis: exploring the dispersion estimates and assessing model fit

Let's take a look at the dispersion estimates for our MOV10 data:

```
## Plot dispersion estimates
plotDispEsts(dds)
```



Since we have a small sample size, for many genes we see quite a bit of shrinkage. Do you think our data are a good fit for the model?
