When you use TPM, the sum of all TPMs in each sample are the same. This makes it easier to compare the proportion of reads that mapped to a gene in each sample. In contrast, with RPKM and FPKM, the sum of the normalized reads in each sample may be different, and this makes it harder to compare samples directly.

Here’s an example. If the TPM for gene A in Sample 1 is 3.33 and the TPM in sample B is 3.33, then I know that the exact same proportion of total reads mapped to gene A in both samples. This is because the sum of the TPMs in both samples always add up to the same number (so the denominator required to calculate the proportions is the same, regardless of what sample you are looking at.)

With RPKM or FPKM, the sum of normalized reads in each sample can be different. Thus, if the RPKM for gene A in Sample 1 is 3.33 and the RPKM in Sample 2 is 3.33, I would not know if the same proportion of reads in Sample 1 mapped to gene A as in Sample 2. This is because the denominator required to calculate the proportion could be different for the two samples.

**The trimmed mean of M-values normalization method**

The total RNA production, *S**k*, cannot be estimated directly, since we do not know the expression levels and true lengths of every gene. However, the relative RNA production of two samples, *f**k**= S**k**/S**k'*, essentially a global fold change, can more easily be determined. We propose an empirical strategy that equates the overall expression levels of genes between samples under the assumption that the majority of them are not DE. One simple yet robust way to estimate the ratio of RNA production uses a weighted trimmed mean of the log expression ratios (trimmed mean of M values (TMM)). For sequencing data, we define the gene-wise log-fold-changes as:

[https://media.springernature.com/lw785/springer-static/image/art%3A10.1186%2Fgb-2010-11-3-r25/MediaObjects/13059_2009_Article_2318_Equb_HTML.gif](https://media.springernature.com/full/springer-static/image/art:10.1186/gb-2010-11-3-r25/MediaObjects/13059_2009_Article_2318_Equb_HTML.gif)

and absolute expression levels:

[https://media.springernature.com/lw785/springer-static/image/art%3A10.1186%2Fgb-2010-11-3-r25/MediaObjects/13059_2009_Article_2318_Equc_HTML.gif](https://media.springernature.com/full/springer-static/image/art:10.1186/gb-2010-11-3-r25/MediaObjects/13059_2009_Article_2318_Equc_HTML.gif)

To robustly summarize the observed M values, we trim both the M values and the A values before taking the weighted average. Precision (inverse of the variance) weights are used to account for the fact that log fold changes (effectively, a log relative risk) from genes with larger read counts have lower variance on the logarithm scale. See Materials and methods for further details.

For a two-sample comparison, only one relative scaling factor (*f**k*) is required. It can be used to adjust both library sizes (divide the reference by https://media.springernature.com/lw785/springer-static/image/art%3A10.1186%2Fgb-2010-11-3-r25/MediaObjects/13059_2009_Article_2318_IEq1_HTML.gifand multiply non-reference by https://media.springernature.com/lw785/springer-static/image/art%3A10.1186%2Fgb-2010-11-3-r25/MediaObjects/13059_2009_Article_2318_IEq1_HTML.gif) in the statistical analysis (for example, Fisher's exact test; see Materials and methods for more details).

Normalization factors across several samples can be calculated by selecting one sample as a reference and calculating the TMM factor for each non-reference sample. Similar to two-sample comparisons, the TMM normalization factors can be built into the statistical model used to test for DE. For example, a Poisson model would modify the observed library size to an effective library size, which adjusts the modeled mean (for example, using an additional offset in a generalized linear model; see Materials and methods for further details).

<https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-3-r25>

Quantile normalization : <https://en.wikipedia.org/wiki/Quantile_normalization>

**Library size** could mean one of 2 things: the total number of reads that were sequenced in the run or the total number of mapped reads. We will use the **total number of mapped reads** as the library size in our analyses.  Normalization of RNA-seq data proceeds by computing an "effective" library size, which is computed from the actual library size and the distribution of the counts.

library size as an offset for each sample.

**Normalization**

Because our analysis uses the actual counts, and the library sizes differ, we need to account for library size in our analysis.  The log-linear model does this by using the library size as an offset for each sample.

However, we can have other effects, such as a small number of highly expressing genes, that affects the read counts.  As in microarrays, one of the assumptions is that the distribution of expression should be about the same in every sample.  Normalization should achieve this.

In RNA-seq data, we typically normalize the data by creating a normalization factor.  The product of the normalization factor and the true library size is the effective library size.  If we needed to obtain adjusted counts for some type of visualization, the appropriate adjustment would be the normalization factor times the true count.  However, for statistical analysis we use the effective library size as an offset - we never adjust the counts because we would then obtain the wrong variance estimate.

The simplest normalization method is to compute some summary of the data, pick a central value of the summary, and then compute the ratio of all the summaries to the central value.  That ratio is the normalization factor.  Because of the high skewness of the counts, often we use a quantile of the distribution.  Using the 75th quantile (25% of the counts are higher, 75% lower) often works well.  However, a slightly better method is the TMM method [1] which is available in **edgeR**.  TMM  appears to work well when we expect that most features do not differentially express.  It attempts to minimize the number of genes that appear to differentially express between any two samples.   Even when I use **DESeq2**  or **voom** I still start by using **edgeR** to compute the normalization factors using TMM

Once you are at this point, you are ready to do the differential expression analysis.  This is covered in the R lab.

**DESEQ-2** => unnormalised counts

The values in the matrix should be un-normalized counts or estimated counts of sequencing reads (for single-end RNA-seq) or fragments (for paired-end RNA-seq)

only the count values allow assessing the measurement precision correctly. The DESeq2 model internally corrects for library size, so transformed or normalized values such as counts scaled by library size should not be used as input.

The object class used by the DESeq2 package to store the read counts and the intermediate estimated quantities during statistical analysis is the *DESeqDataSet*, which will usually be represented in the code here as an object **dds**.

Formula for DESEqDataObject = ~var + var2 + etc.

Note: In order to benefit from the default settings of the package, you should put the variable of interest at the end of the formula and make sure the control level is the first level.

Count Matrix Input : function DESeqDataSetFromMatrix  : Pastilla library into DESEq dataObject

### Collapsing technical replicates

DESeq2 provides a function collapseReplicates which can assist in combining the counts from technical replicates into single columns of the count matrix. The term technical replicate implies multiple sequencing runs of the same library. You should not collapse biological replicates using this function. See the manual page for an example of the use of collapseReplicates.

DesEq2 differential expression :

### Log fold change shrinkage for visualization and ranking

Shrinkage of effect size (LFC estimates) is useful for visualization and ranking of genes. To shrink the LFC, we pass the dds object to the function lfcShrink. Below we specify to use the apeglm method for effect size shrinkage (Zhu, Ibrahim, and Love 2018), which improves on the previous estimator.

**Independent hypothesis weighting**

A generalization of the idea of *p* value filtering is to *weight* hypotheses to optimize power. A Bioconductor package, [IHW](http://bioconductor.org/packages/IHW), is available that implements the method of *Independent Hypothesis Weighting* (Ignatiadis et al. 2016). Here we show the use of *IHW* for *p* value adjustment of DESeq2 results. For more details, please see the vignette of the [IHW](http://bioconductor.org/packages/IHW) package. The *IHW* result object is stored in the metadata.

## Exploring and exporting results (MA-plot)

In DESeq2, the function plotMA shows the log2 fold changes attributable to a given variable over the mean of normalized counts for all the samples in the DESeqDataSet. Points will be colored red if the adjusted p value is less than 0.1. Points which fall out of the window are plotted as open triangles pointing either up or down.