Differential Gene Expression Analysis (DGE)

Friederike Dündar

February 26, 2020

In addition to performing exploratory analyses based on normalized measures of expression levels, numerous efforts have been dedicated to optimize statistical tests to decide whether a (single!) given gene's expression varies between two (or more) conditions based on the information gleaned from as little as two or three replicates per condition. The two basic tasks of all DGE tools are:

1. Estimate the *magnitude* of differential expression between two or more conditions based on read counts from replicated samples, i.e., calculate the fold change of read counts, taking into account the differences in sequencing depth and variability.
2. Estimate the *significance* of the difference and correct for multiple testing.

The best performing tools tend to be `edgeR` (Robinson et al., 2010), `DESeq/DESeq2` (Anders and Huber, 2010; Love et al., 2014), and `limma-voom` (Ritchie et al., 2015) (see Rapaport et al. (2013); Soneson and Delorenzi (2013); Schurch et al. (2015) for reviews of DGE tools). `DESeq` and `limma-voom` tend to be more conservative than `edgeR` (better control of false positives), but `edgeR` is recommended for experiments with fewer than 12 replicates (Schurch et al., 2015). These tools are all based on the `R` language and make heavy use of numerous statistical methods that have been developed and implemented over the past two decades to improve the power to detect robust changes based on extremely small numbers of replicates and to help deal with the quirks of integer count data. These tools basically follow the same approach, i.e., they estimate the gene expression difference for a given gene using regression-based models (and taking the factors discussed during the session on normalization into account), followed by a statistical test based on the null hypothesis that the difference is close to zero, which would mean that there is no difference in the gene expression values that could be explained by the conditions. Table 1 has a summary of the key properties of the most popular DGE tools; the next two sections will explain some more details of the two key steps of the DGE analyses.

! All statistical methods developed for read counts rely on approximations of various kinds, so that assumptions must be made about the data properties. `edgeR` and `DESeq`, for example, assume that the majority of the transcriptome is *unchanged* between the two conditions. If this assumption is not met by the data, both log_2 fold change and the significance indicators are most likely incorrect!

Table 1: Comparison of programs for differential gene expression identification. Information shown here is based on the user guides of `DESeq2`, `edgeR`, `limmaVoom` and Rapaport et al. (2013), Seyednasrollah et al. (2015), and Schurch et al. (2015). LRT stands for log-likelihood ratio test.

Feature	DESeq2	edgeR	limmaVoom	Cuffdiff
Seq. depth normalization	Sample-wise size factor	Gene-wise trimmed median of means (TMM)	Gene-wise trimmed median of means (TMM)	FPKM-like or DESeq-like
Dispersion estimate	Cox-Reid approximate conditional inference with focus on maximum *individual* dispersion estimate	Cox-Reid approximate conditional inference moderated towards the *mean*	squeezes gene-wise residual variances towards the global variance	
Assumed distribution	Neg. binomial	Neg. binomial	*log*-normal	Neg. binomial
Test for DE	Wald test (2 factors); LRT for multiple factors	exact test for 2 factors; LRT for multiple factors	*t*-test	*t*-test
False positives	Low	Low	Low	High
Detection of differential isoforms	No	No	No	Yes
Support for multi-factored experiments	Yes	Yes	Yes	No
Runtime (3-5 replicates)	Seconds to minutes	Seconds to minutes	Seconds to minutes	Hours

1 Estimating the difference between read counts for a given gene

To determine whether the read count differences between different conditions for a given gene are greater than expected by chance, DGE tools must find a way to estimate that difference using the information from the replicates of each condition. `edgeR` (Robinson et al., 2010), `DESeq/DESeq2` (Anders and Huber, 2010; Love et al., 2014), and `limma-voom` (Ritchie et al., 2015) all use regression models that are applied to every single gene. Linear regression models usually take the following form: $Y = b_0 + b_1 * x + e$ and they are typically used to assess the **strength** of the relationship between Y and x, i.e., how much does Y really depend on x? The observed values are used to **estimate** the values of b_0 and b_1 to obtain the closest fit to the data at hand. Regression *coefficients* represent the mean change in the response variable, Y, for one unit of change in the predictor variable, x. Therefore, the closer b_1 is to zero, the weaker is the relationship between Y and x. Regression models are usually used to predict unknown values of Y, i.e., one often wants to find a function that returns Y at any given point along a certain trajectory captured by the model where x is typically sampled from a continuous distribution of values (Figure 1).

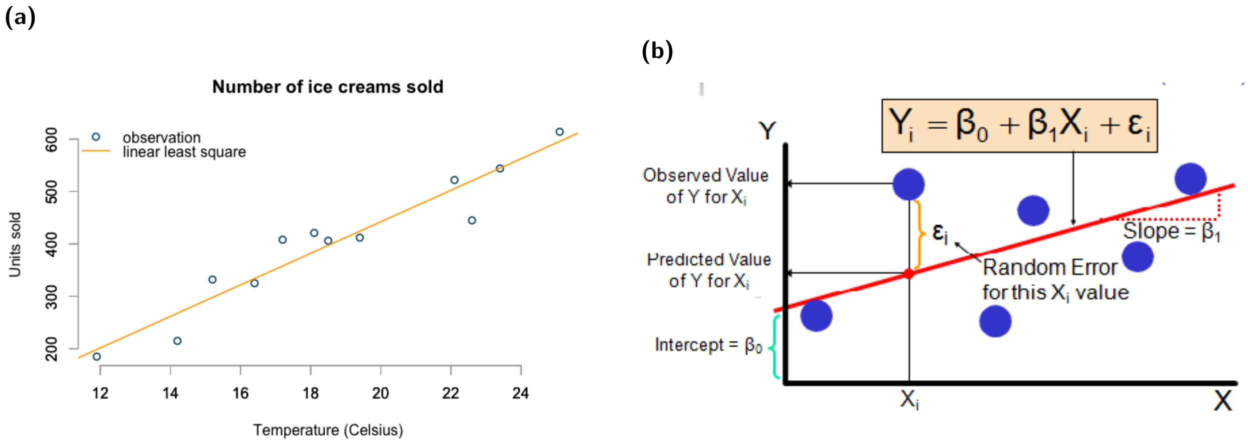

Figure 1: (a) Typical example of a regression model application. Here, Y represent the numbers of ice creams sold and the question of interest is the dependence of Y on the outside temperature (x). (b) Explanations for the relationship of the different terms of the linear model. Figures from `https://bit.ly/2PoYJ6d` and `https://bit.ly/3cbogJJ`.

In the case of RNA-seq, Y represents the observed expression values and x represents the different *conditions* from which the expression values of Y stem, i.e. instead of x assuming continuous values, we are assigning ordinal values to x. Since the regression *coefficients* represent the mean change in Y for one unit of change in x, we can use b_1 to determine whether the expression values for one specific gene change depending on which group of x they came from. For normally distributed and abundantly replicated data, the same goal could be achieved with a t-test. Remember, however, that RNA-seq data does not meet either criterion, which is why more sophisticated models are used to estimate the regression coefficients.

More specifically:

- Y will entail *all* read counts (from all conditions) for a given gene;
- x encodes the condition (for RNA-seq, this is very often a discrete factor, e.g., “WT” or “mutant”, or, in mathematical terms, 0 or 1);
- the value of the *intercept*, b_0, represents the expression values of the baseline condition;
- the *regression coefficient*, b_1, happens to capture the difference between Y from samples of different conditions;
- e captures the error or uncertainty, i.e. the difference of the regression estimates from the observed expression values.

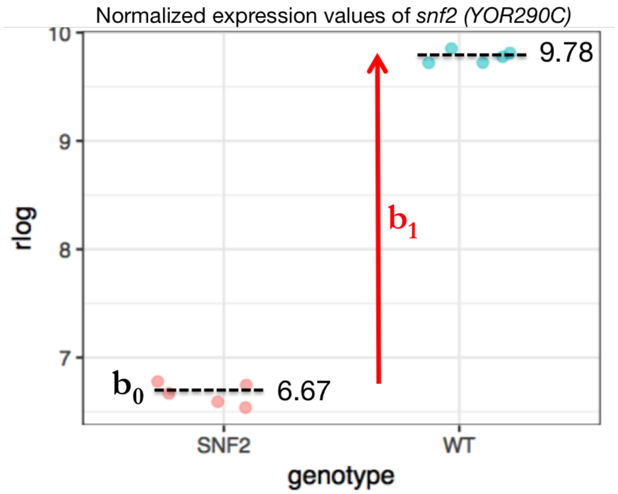

Figure 2: For the most basic comparison of two conditions, imagine a set of normalized expression values, Y, which differ depending on which group of x they belong to: "SNF2" or "WT". If we want to understand how Y changes depending on which instance of x is chosen, we can use a regression model. x is therefore interpreted as a discrete parameter, which is set to 0 for the baseline condition (here: SNF2) and set to 1 for the non-reference group (here: WT) The intercept, b_0, should then be close to the average values of Y values of the baseline group. As shown in the figure, it then follows that the regression coefficient, b_1, represents the *difference* between baseline and non-baseline group: $Y = b_0 + b_1 * x$.

The very simple model illustrated in Figure 2 could be fitted in R using the function `lm(rlog.norm[, 'gene_Z' ~ genotype)` [*], which will return estimates for both b_0 and b_1, so that the average expression values of the baseline genotype (e.g., SNF2 = 0) would correspond to $Y = b_0 + b_1 * 0 + e$. This is equivalent to $Y = b_0$ (assuming that e is very small), thereby demonstrating why the intercept (b_0) can be interpreted as the average of our baseline group. b_1, on the other hand, will be the coefficient whose closeness to zero will be evaluated during the statistical testing step since it represents the magnitude of the difference for Y that is explained by the two different groups of x.

While understanding the *linear* model approach is useful in order to understand why regression is used in the first place for DE analyses, `DESeq2` and `edgeR` rely on a *negative binomial* model to fit the observed read counts to arrive at the estimate for the difference.

Originally, read counts had been modeled using the *Poisson* distribution because:

- individual reads can be interpreted as binary data (Bernoulli trials): they either originate from gene i or not.
- we are trying to model the discrete probability distribution of the number of successes (success = read is present in the sequenced library).
- the pool of possible reads that could be present is large, while the proportion of reads belonging to gene i is quite small.

The convenient feature of a Poisson distribution is that *variance = mean*. Thus, if the RNA-seq experiment gives us a precise estimate of the mean read counts per condition, we implicitly know what kind of variance to expect for read counts that are not truly changing between two conditions. This, in turn, then allows us to identify those genes that show greater differences between the two conditions than expected by chance.

Unfortunately, only read counts of the same library preparation (= technical replicates) can be well approximated by the Poisson distribution; biological replicates have been shown to display greater variance (noise). This *overdispersion* can be captured with the *negative binomial* distribution, which is a more general form of the Poisson distribution where the variance is allowed to exceed the mean. This means that we now need to estimate two parameters from the read counts: the mean as well as the dispersion. The precision of these estimates strongly depends on the number (and variation) of replicates – the more replicates, the better the grasp on the underlying mean expression values of unchanged genes and the variance that is due to biological variation rather than the experimental treatment. For most RNA-seq experiments, only two to three replicates are available, which is obviously not sufficient for robust mean and variance estimates. Some tools therefore compensate for the lack of replication by borrowing information across genes with similar expression values to artificially shrink a given gene's variance towards the regressed values. These fitted values of the mean and dispersion are then used instead of the raw estimates to test for differential gene expression.

[*] In plain English: rlog-normalized expression values for gene Z are modeled based on the genotype (Figure 2).

2 Testing the null hypothesis

The null hypothesis is that there is no systematic difference between the average read count values of the different conditions for a given gene. In terms of the regression models this means that we are testing whether the regression coefficient, b_1, helps explain the differences among the observed expression values. Which test is used to assign a p-value again depends on the tool (Table 1), but generally you can think of them as some variation of the well-known $t-$test (How dissimilar are the means of two populations?) or ANOVAs (How well does a reduced model capture the data when compared to the full model with all coefficients?). `DESeq2` uses the Wald statistic, which is defined as $W = \frac{\hat{\beta}}{\hat{se}(\hat{\beta})}$ where the hat symbol denotes the estimates of the regression coefficient. If the resulting Wald statistic is close to zero (e.g. because the standard error, se, is large), the null hypothesis cannot be rejected, which will be reflected by a p-value close to 1.

Once you've obtained a list of p-values for all the genes of your data set, it is important to realize that you just performed the same type of test for thousands and thousands of genes. That means, that even if you decide to focus on genes with a p-value smaller than 0.05, if you've looked at 10,000 genes your final list may contain $0.05 * 10,000 = 500$ false positive hits. To guard yourself against this, all the tools will offer some sort of correction for the multiple hypotheses you tested, e.g. in the form of the Benjamini-Hochberg formula. Generally, the severity of the "punishment" for the p-values will correspond to the number of tests, i.e. the more genes you test, the smaller the raw p-values will have to be in order to pass the final adjusted p-value threshold. You should definitely rely on the adjusted p-values rather than the original p-values to identify possible candidate genes for downstream analyses and follow-up studies, but do look into the independent filtering approach that `DESeq2` employs (`https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#indfilt`).

References

Anders S and Huber W. DESeq: Differential expression analysis for sequence count data. *Genome Biology*, **11**:R106, 2010. doi:10.1186/gb-2010-11-10-r106.

Love MI, Huber W, and Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, **15**(12):550, 2014. doi:10.1186/s13059-014-0550-8.

Rapaport F, Khanin R, Liang Y, Pirun M, Krek A, Zumbo P, Mason CE, Socci ND, and Betel D. Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome Biology*, **14**(9):R95, 2013. doi:10.1186/gb-2013-14-9-r95.

Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, and Smyth GK. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, **43**(7):e47–, 2015. doi: 10.1093/nar/gkv007.

Robinson MD, McCarthy DJ, and Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**(1), 2010. doi:10.1093/bioinformatics/btp616.

Schurch NJ, Schofield P, Gierliński M, Cole C, Simpson GG, Hughes TO, Blaxter M, and Barton GJ. Evaluation of tools for differential gene expression analysis by RNA-seq on a 48 biological replicate experiment. *ArXiv e-prints*, 2015. URL `http://arxiv.org/abs/1505.02017`.

Seyednasrollah F, Laiho A, and Elo LL. Comparison of software packages for detecting differential expression in RNA-seq studies. *Briefings in Bioinformatics*, **16**(1):59–70, 2015. doi:10.1093/bib/bbt086.

Soneson C and Delorenzi M. A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics*, **14**(1):91, 2013. doi:10.1186/1471-2105-14-91.