- Notes -

RNAseq data analysis

By Pawel Rosikiewicz
www.SimpleAI.ch

Data Pre-processing

The read counts provided for differential expression analysis need to be normalised, before giving them as input to DeSeq or edgeR(differential expression analysis tools in R). The normalisation techniques are :

- **Log Transform -**

  The log of raw read counts is taken which results in increasing the distance between small measurements and decreasing the distance between large measurements.

- **Quantile Normalisation -**

  Multi-sample normalization techniques such as quantile normalization have become a standard and essential part of analysis pipelines for high-throughput data.These techniques transform the original raw data to remove unwanted technical variation. Technical variation can cause perceived differences between samples processed on high-throughput technologies, irrespective of the biological variation. These differences are typically due to changes in experimental conditions that are hard or impossible to control and confusing them with biological variability can lead to false discoveries.

- **Filter -**

  The genes with relatively low or no expression value are filtered out from the final normalised read count dataset which is then given as input to DeSeq or edgeR.

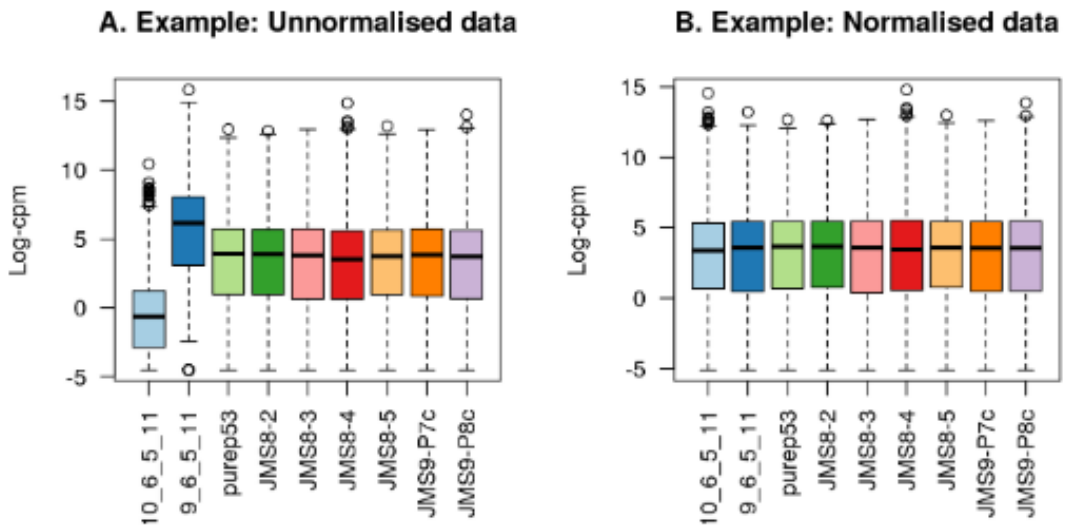

Fig 2 - Unnormalised and normalised distribution of different groups after data pre-processing

Technical Batch Effects

Technical batch effect occurs when non-biological factors in an experiment cause changes in the data produced by the experiment. Such effects can lead to inaccurate conclusions when their causes are correlated with one or more outcomes of interest in an experiment. Thus, we need to balance the variables of our interest in an experiment in order to remove the batch effects. Techniques involved are :

1. PCA(Principal Component Analysis)
2. MDS(Multidimensional Scaling)

These techniques reduce the representation of each sample from a vector of thousands of measurements to a vector of length of number of samples. Also, this vector captures the largest sources of variation in the dataset. The unwanted variations, which interfere with the variable of interest are then removed, thus eliminating the batch effect.

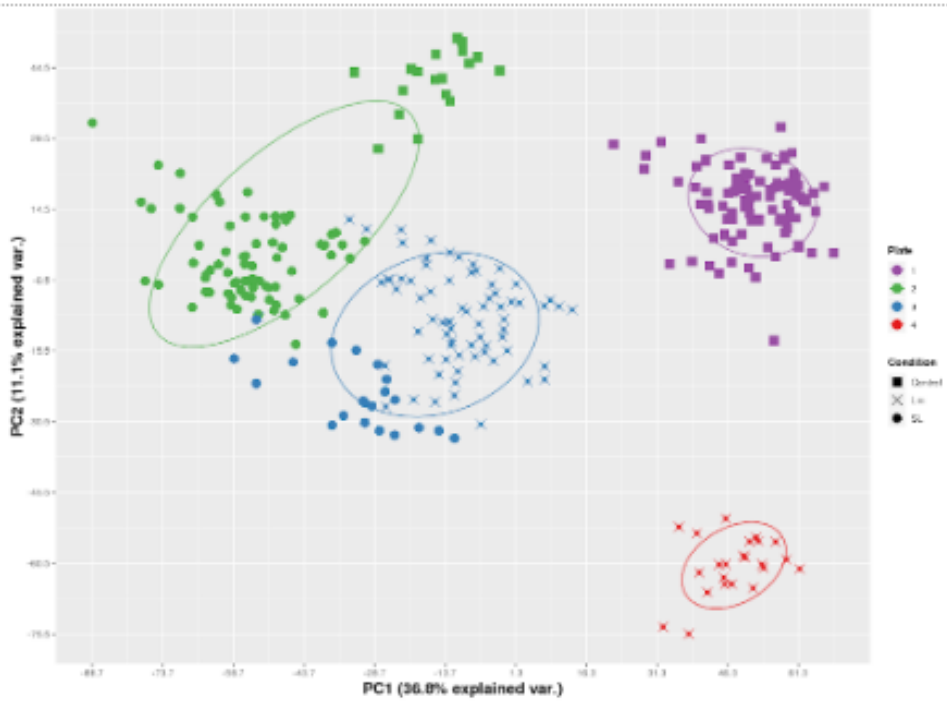

Fig 3 - Batch effect observed among different groups of same experiment after PCA

Methods for Differential Expression Analysis

There are different pipelines for differential expression analysis in R such as **edgeR** and **DESeq** based on **negative binomial (NB) distributions**. It is important to consider the experimental design when choosing an analysis method. While some of the differential expression tools can only perform **pairwise comparison,** others such as **edgeR, limma-voom, DESeq can perform multiple comparisons**.

These tools basically follow the same approach, i.e. ,

- ⇒ they estimate the gene expression difference for a given gene,
- ⇒ 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.
- ⇒ 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. Estimating the difference between read counts :

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:

Why Negative Binomial ?

- individual reads can be interpreted as binary data (Bernoulli trials): they either originate from a single gene A 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 A is quite small.

Why Poisson distribution ?

- 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.

So why today we are using Negative Binomial ?

- 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.

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.
- **The p-values** are assigned by these tools using 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?).
- Once a list of p-values for all the genes of our data set is obtained, it is important to realize that the same type of test has been performed for thousands and thousands of genes. That means, that even if genes with a p-value smaller than 0.05 are considered, and there are 1000 genes, there may be $0.05 * 1000 = 50$ false positive hits. Consequently, all the tools offer some sort of **correction for this multiple testing hypotheses like Benjamini-Hochberg formula**.

RAseq data analysis - general Methodology

1. Raw reads (**FASTQ files**) undergo **quality assessment** and filtering.
2. The quality-filtered reads are **aligned to the reference genome** via aligners like **HiSat or TopHat2**
3. The mapped reads are **summarised and aggregated over genes via HTSeq**.
4. For baseline expression, the **FPKMsn (Fragments Per Kilobase Million)** are calculated from the raw counts by **iRAP**.
5. These are **averaged for each set of technical replicates**,
6. and then **quantile normalised within each set of biological replicates** using limma.
7. Finally, they are **averaged for all biological replicates** (if any).
8. For differential expression, **genes expressed differentially between the test and the reference** groups of each **pairwise** contrast are identified using **DESeq2**.

Biological interpretation of gene expression data

1. A common method of visualising gene expression data is to display it as a heatmap (Figure 12). The heatmap may also be combined with clustering methods which group genes and/or samples together based on the similarity of their gene expression pattern. This can be useful for identifying genes that are commonly regulated, upregulated or downregulated(based on log2 fold change values), or biological signatures associated with a particular condition (e.g a disease or an environmental condition).

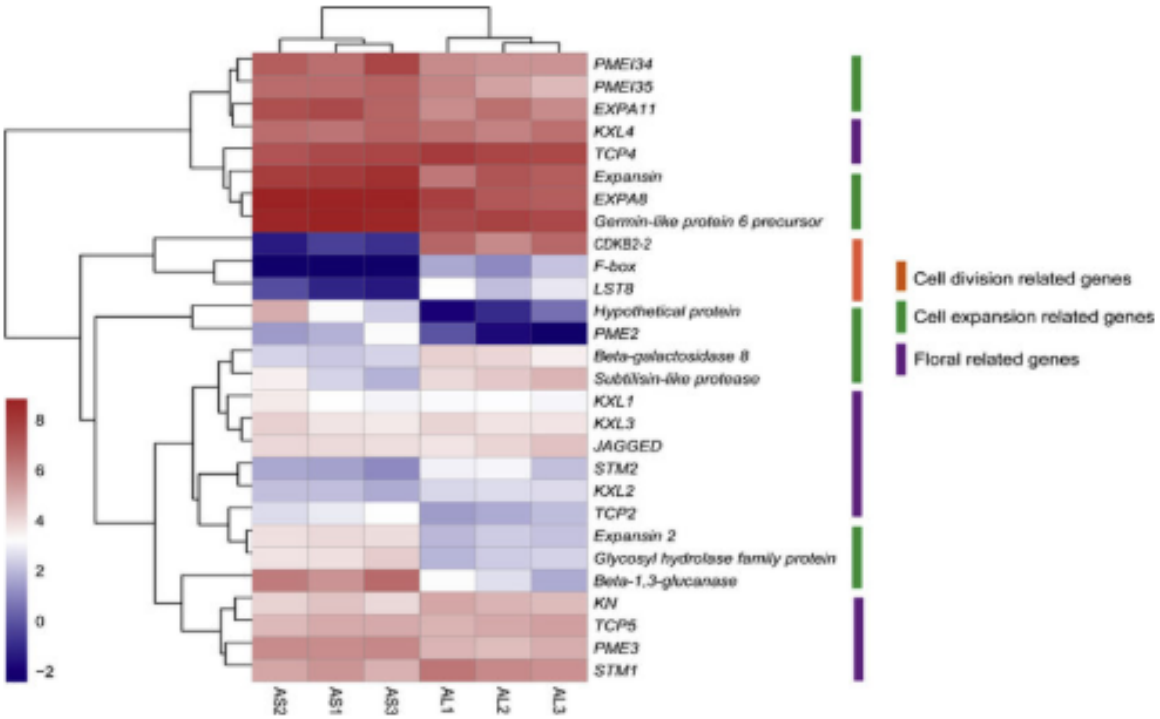

Fig 4 - Example of a heatmap to visualise differentially expressed genes

1. A common approach to interpreting gene expression data is **gene set enrichment analysis based on the functional annotation of the differentially expressed genes**. This is useful for finding out if the differentially expressed genes are associated with a certain biological process or molecular function.
2. **The Gene Ontology**, containing standardised annotation of gene products, is commonly used for this purpose. It works by comparing the frequency of individual annotations in the gene list (e.g differentially expressed genes) with a reference list (usually all genes on the microarray or in the genome). Enrichment of biological pathways supplied by **KEGG, Ingenuity, Reactome or WikiPathways** can be performed in a similar way.

Reference:

- https://www.ebi.ac.uk/training/online/course/functional-genomics-ii-common-technologies-and-data-analysis-methods/differential-gene
- https://www.datacamp.com/courses/differential-expression-analysis-with-limma-in-r
- https://chagall.med.cornell.edu/RNASEQcourse/Intro2RNAseq.pdf

- https://en.wikipedia.org/wiki/Batch_effect

RPKM, FPKM and TPM

From : https://www.rna-seqblog.com/rpkm-fpkm-and-tpm-clearly-explained/

It used to be when you did RNA-seq, you reported your results in **RPKM (Reads Per Kilobase Million)** or **FPKM (Fragments Per Kilobase Million).** However, **TPM (Transcripts Per Kilobase Million**) is now becoming quite popular. Since there seems to be a lot of confusion about these terms, I thought I'd use a StatQuest to clear everything up.

These three metrics attempt to normalize for sequencing depth and gene length.

RPKM (Reads Per Kilobase Million)

Here's how you do it for RPKM: - RPKM was made for single-end RNA-seq

1. **Count up the total reads in a sample** and divide that number by 1,000,000 – this is our "per million" scaling factor.
2. **Divide the read counts by the "per million" scaling factor**. This normalizes for sequencing depth, giving you reads per million (RPM)
3. **Divide the RPM values by the length of the gene, in kilobases**. This gives you RPKM.
4.

FPKM (Fragments Per Kilobase Million).

FPKM is very similar to RPKM. RPKM was made for single-end RNA-seq, where every read corresponded to a single fragment that was sequenced. FPKM was **made for paired-end RNA-seq**. With paired-end RNA-seq, two reads can correspond to a single fragment, or, if one read in the pair did not map, one read can correspond to a single fragment. The only difference between RPKM and FPKM is that FPKM **takes into account that two reads can map to one fragment** (and so it doesn't count this fragment twice).

TPM (Transcripts Per Kilobase Million)

TPM is very similar to RPKM and FPKM. The only difference is the order of operations. Here's how you calculate TPM:

1. **Divide the read counts by the length of each gene in kilobases**. This gives you reads per kilobase (RPK).
2. Count up all the RPK values in a sample and divide this number by 1,000,000. This is your "per million" scaling factor.
3. Divide the RPK values by the "per million" scaling factor. This gives you TPM.

So you see, when calculating TPM, the only difference is that you normalize for gene length first, and then normalize for sequencing depth second. However, the effects of this difference are quite profound.

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

EXAMPLES :

⇒ 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.

LINKS

⇒ **RNA seq data: Differential expression analysis**
R · Fibrosis SMOC2 Raw Counts
https://www.kaggle.com/code/vsevolodcherepanov/rna-seq-data-differential-expression-analysis/notebook

⇒ **Differential Gene Expression Analysis**
https://www.kaggle.com/code/garimabansal/differential-gene-expression-analysis/notebook

⇒ **EDA scanpy (bioinformatics standard analysis)**
https://www.kaggle.com/code/yyoshiaki/eda-scanpy-bioinformatics-standard-analysis