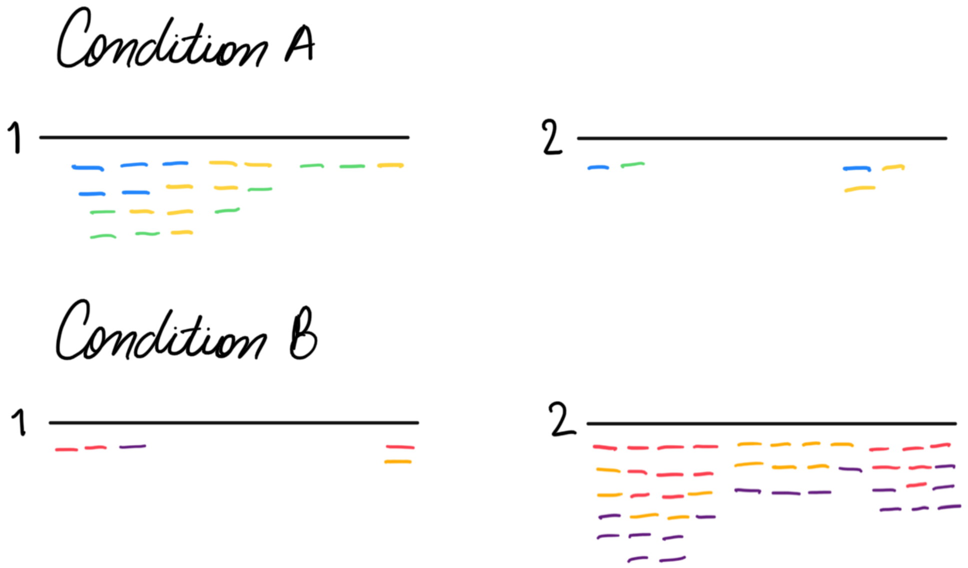
Started on Monday, May 17th, 2021 by Luigui Gallardo-Becerra (bfllg77@gmail.com)

Introduction to differential expression analysis

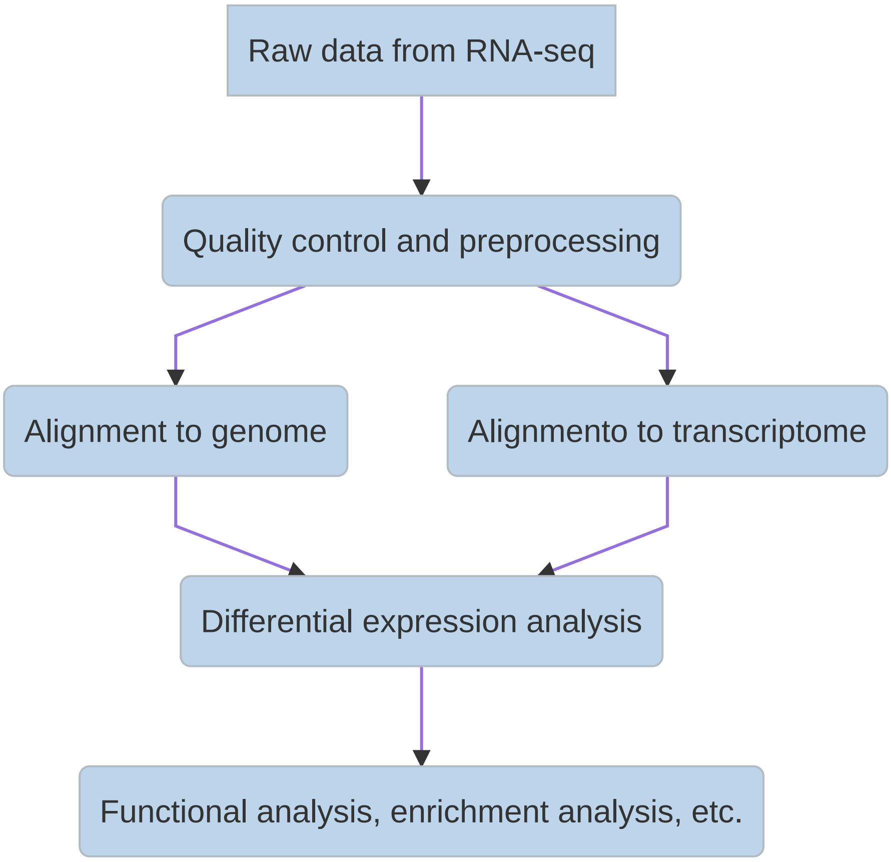
Differential expression (DE) analysis is used to answer the question: what are the genomic features (genes, transcripts or exons) that are expressed significantly different between groups of samples? To accomplish this, we need to quantify the differences between the RNA-seq data for each sample and group.

For example, in the next picture we want to compare two conditions (A and B), for the genes 1 and 2; each condition has 3 replicates marked in different colors. We could see that there are a DE between conditions: Condition A has more reads for the gene 1, while Condition B has more reads for the gene 2. The DE does not imply that there is no expression of one or more genes at all in one condition or another, but rather that it exists in a differential proportion considering similar depth sequence.



Pipeline overview

To know what are the DE genomic features, we must to follow a pipeline like the one in the figure below. As we saw before, the fundamental step is the quantification of the RNA-seq data, but first we should include previous steps to control the quality of our data. Subsequently, we must align our reads against the genome (if available) or the transcriptome (could be 'de novo' assembled with the same reads). Finally, we have the DE analysis and after we could do functional analysis, enrichment analysis, etc.



Raw data from RNA-seq
Quality control and preprocessing
Alignment to genome
Alignmento to transcriptome
Differential expression analysis
Functional analysis, enrichment analysis, etc.

Differential expression with DESeq2

There a many algorithms used to make DE analysis, one of the most used is DESeq2. We could use the R package, to install it we run this inside R:

```
#### Instalation of BiocManager, DESeq2, apeglm
#### Run only once in your computer
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("DESeq2")
BiocManager::install("apeglm")

#### Optional libraries to generate graphs: instalation of ggplot2, pheatmap and
RColorBrewer
#### Run only once in your computer
install.packages("ggplot2")
install.packages("pheatmap")
install.packages("RColorBrewer")

#### Load libraries
library(DESeq2)
library(apeglm)

#### Load optional libraries
library(ggplot2)
library(pheatmap)
library(RColorBrewer)
```

After, we import the counting matrix (generated with RSEM) and the metadata file with the sample-group correspondence:

```
rsem_count <- as.matrix(read.delim(file = "RSEM.gene.counts.matrix",
    sep="\t",
    row.names = 1))

column_data <- read.csv(file = "metadata.txt",
    sep = "\t")

deseq_data_set <- DESeqDataSetFromMatrix(countData = rsem_count,
    colData = column_data,
    design = ~group)

deseq_data_set$group <- relevel(deseq_data_set$group,
    ref = "dextrosa")
```

As a recommendation, we should remove the low abundance transcripts:

```
keep <- rowSums(counts(deseq_data_set)) >= 10

deseq_data_set <- deseq_data_set[keep,]
```

With the filtered matrix, we could start with the DE analysis:

```
dif_expression_deseq_data_set <- DESeq(deseq_data_set)

res_dif_expression_deseq_data_set <- results(dif_expression_deseq_data_set)

summary_dif_expression_deseq_data_set <-
summary(res_dif_expression_deseq_data_set)
```

We could filter the results, like the p-value and log2 fold change:

```
res05_dif_expression_deseq_data_set <- results(dif_expression_deseq_data_set,
    alpha=0.05)

summary_dif_expression_deseq_data_set <-
summary(res05_dif_expression_deseq_data_set)
```

Finally, to export the final table we run:

```
write.table(as.data.frame(res05_dif_expression_deseq_data_set),sep = '\t' ,
    file="results.tsv")
```

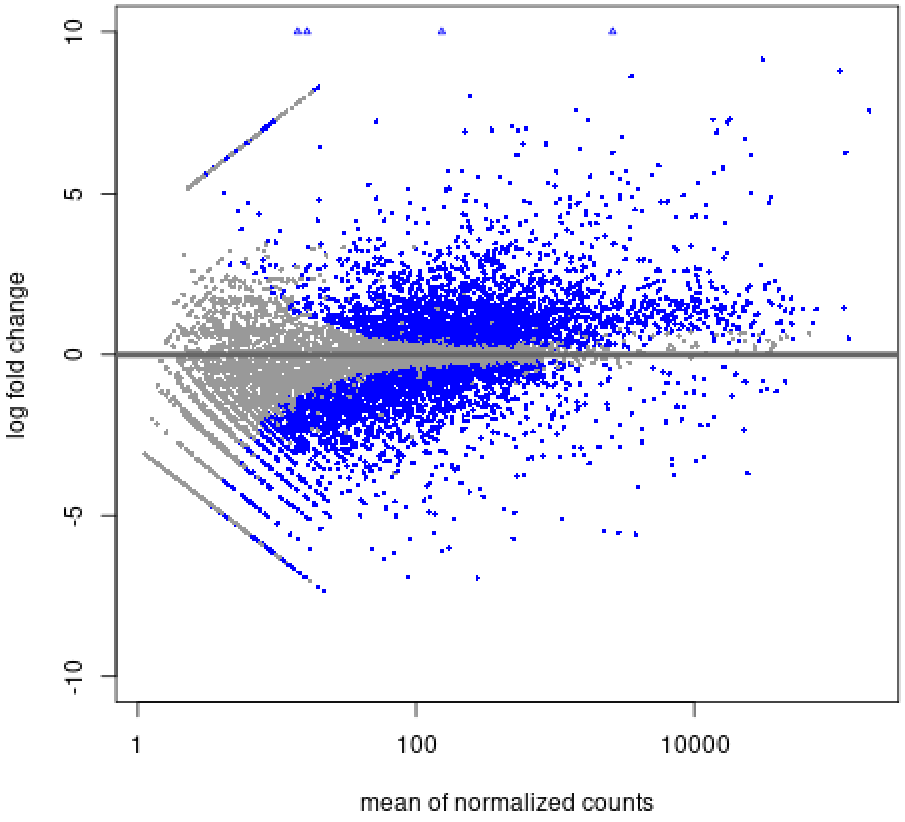
Graphs

After we obtain the final matrix from DESeq2, we can generate several graphs. For example, to generate a Volcano plot we could run this:

```
png("volcano_plot.png")

plotMA(res05_dif_expression_deseq_data_set,
    ylim=c(-10,10))

dev.off()
```

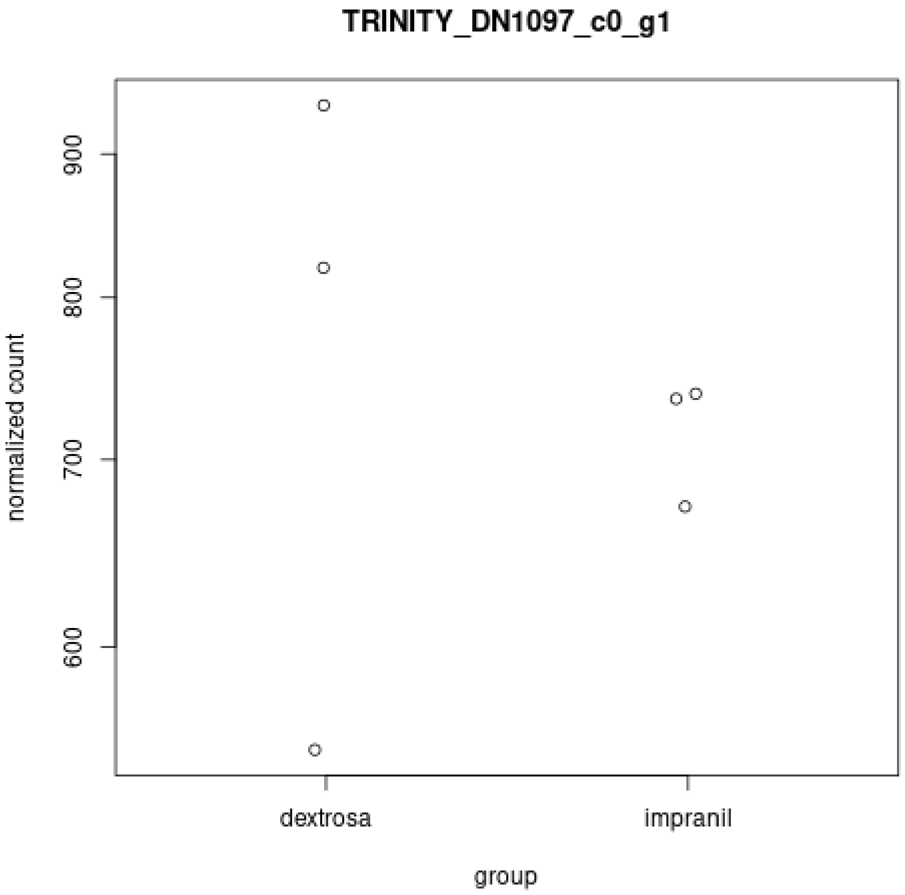


To generate an individual plot (only one feature) we run:

```
png("TRINITY_DN1097_c0_g1.png")

plotCounts(dif_expression_deseq_data_set,
    gene= "TRINITY_DN1097_c0_g1",
    intgroup="group")

dev.off()
```



We could generate a heatmap with a selection of the transcripts. In the example below we obtain the first 20 transcripts and plot them in a heatmap:

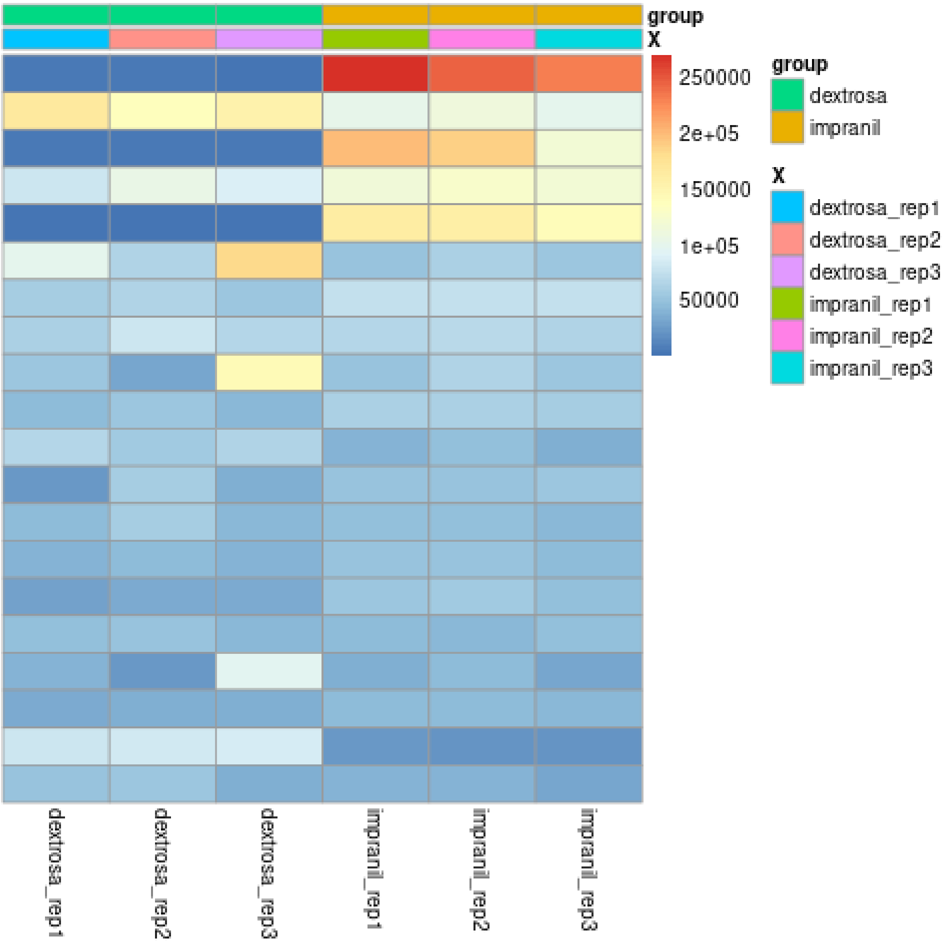
```
select <- order(rowMeans(counts(dif_expression_deseq_data_set,normalized=TRUE)),
    decreasing=TRUE)[1:20]

df <- as.data.frame(colData(dif_expression_deseq_data_set)[,c("X","group")])

png("heatmap.png")

pheatmap(assay(dif_expression_deseq_data_set)[select,],
    cluster_rows=FALSE,
    show_rownames=FALSE,
    cluster_cols=FALSE,
    annotation_col=df)

dev.off()
```



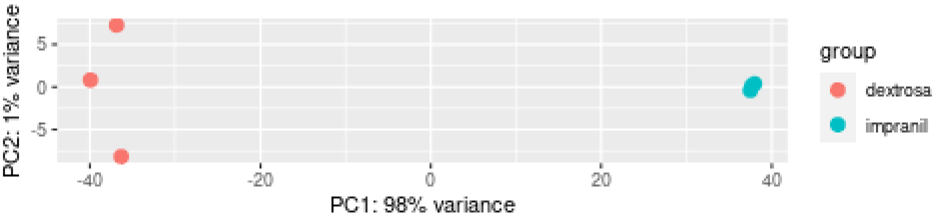
To generate a PCA, we run:

```
vsd <- vst(dif_expression_deseq_data_set, blind=FALSE)

png("PCA.png")

plotPCA(vsd, intgroup=c("group"))

dev.off()
```


PC2: 1% variance
5
0
-5
-40
-20
0
20
40
PC1: 98% variance
group
dextrosa
impranil