Assignment

[Write your own name here]

2023-03-03

Contents

Introduction

Instructions

This R Markdown file is a template that you will fill with: - R code, - normal text answers in English, - figures created using R to justify your answer, - and your conclusions and interpretation of the figures.

Regularly, you will "knit" (compile) your R Markdown into a PDF using the blue knit button. Once you've completed your assignment, you will have to generate a final PDF and upload it on Canvas. One PDF assignment per student is expected.

Below you will find some R code sections in grey that are sometimes pre-filled for your convenience (for instance to get the data easily). If you delete this template by error, the original R Markdown file is also available on Canvas.

Some R code sections have code commented with hastags (#). To run the code, delete the hastags and execute the code section using the green arrow.

Disabling warnings and messages

This will keep your final PDF report clean from execution alarms, unnecessary text, etc.
This code chunck sets global options for the execution of each code chunk. You can disable warnings and messages globally this way.

(execute code below)

Library loading

We first load the `tidyverse` package that contains most of the data transformation functions we will need. We will also load the `DESeq2` package to be able to perform the differential gene expression analysis. The `patchwork` library to place plots next to one another for example.

```
suppressPackageStartupMessages(library("tidyverse"))
suppressPackageStartupMessages(library("DESeq2"))
suppressPackageStartupMessages(library("patchwork"))
```

Dataset 01: import gene counts from control and ABA-treated shoot

We will use gene counts from shoot tissues from plants treated or not with ABA.

```
aba_counts <- read.csv(
  file = "./ABA/shoot_gene_counts_ABA.csv",
  header = TRUE,
  stringsAsFactors = FALSE) %>%
  # for DESeq subsequent data import
  column_to_rownames("gene")

head(aba_counts)
##           shoot_control_1 shoot_control_2 shoot_control_3 shoot_ABA_1
## AT1G01010             510             830             556         662
## AT1G01020            1094            1502            1082        1381
## AT1G01030             337             568             314         370
## AT1G01040            2489            3218            2157        3997
## AT1G01046              75              47              41         108
## AT1G01050            3402            5508            2468        5600
##           shoot_ABA_2 shoot_ABA_3
## AT1G01010         706         525
## AT1G01020        1133        1147
## AT1G01030         557         401
## AT1G01040        3517        3524
## AT1G01046          52          62
## AT1G01050        5663        4212
```

Exercise 1: Characteristics of gene counts coming from RNA-seq

Question 1

Q1: What fraction (percentage) of the genes are not expressed in the shoot? Hint: calculate the number of genes that have a sum of 0 (not expressed in any of the samples). You can use the rowSums() function to do so.

```
n_genes_0 = aba_counts %>% mutate(gene_sum_counts = rowSums(.)) %>% filter(gene_sum_counts == 0) %>% nr

# fraction of the genes not expressed in shoots
n_genes_0 / nrow(aba_counts) * 100
## [1] 10.87913
```

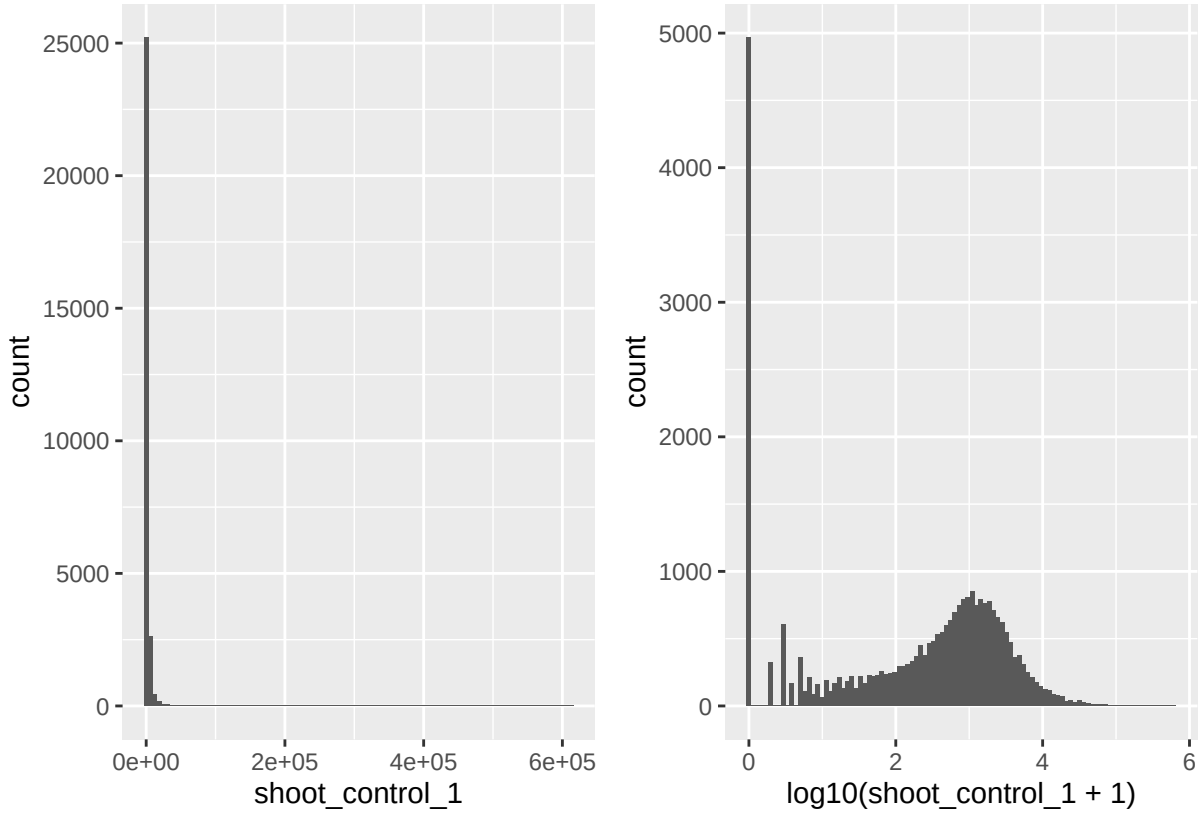
11% of the genes are not expressed at all.

Question 2

Q2: Select one sample and plot the distribution of gene counts and comment it. Can you think of a transformation to make this distribution more normal?

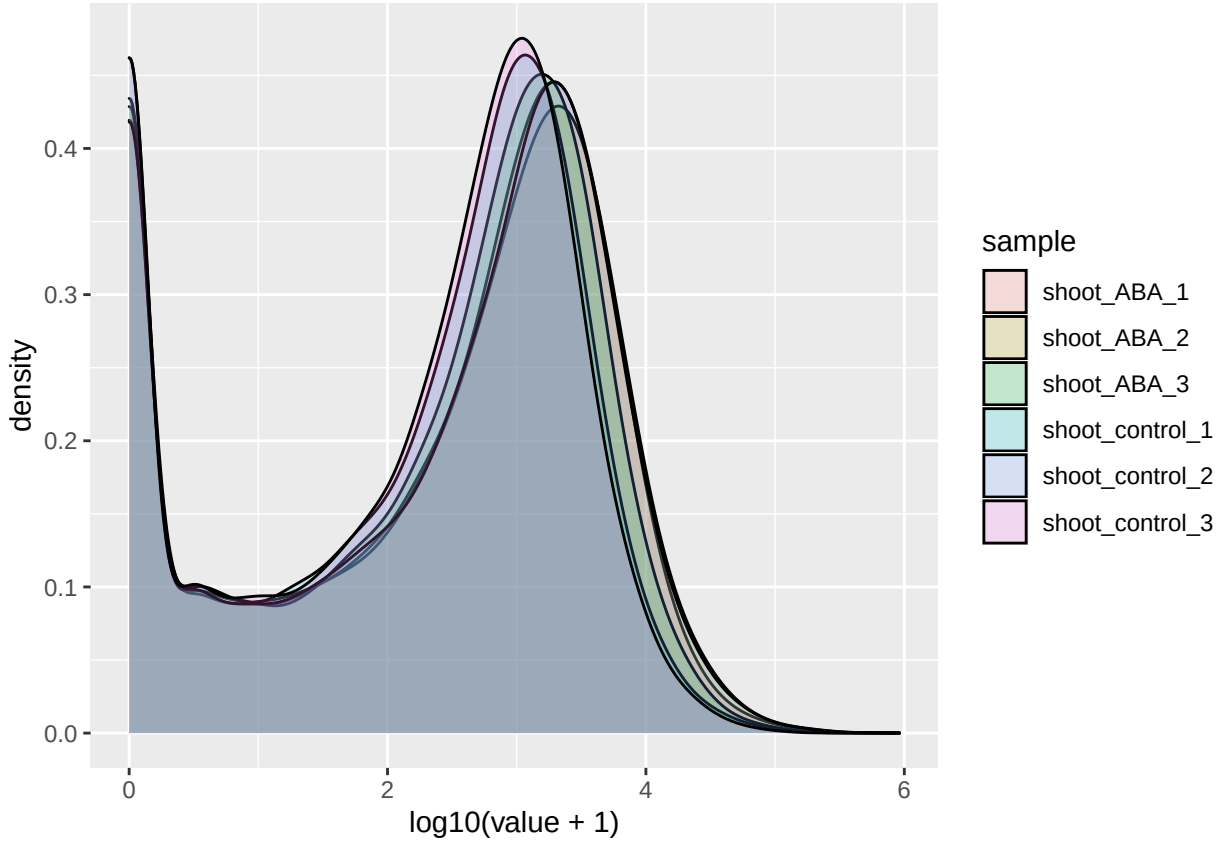
Answer Q2: the distribution is very skewed with very few highly expressed genes and a lot of little expressed genes.
A log10 transformation helps to make it more normal. Here, I use a log10 transformation with an offset of 1 (so that genes with 0 counts become log10(0+1)=0).

```
p1 <-
  ggplot(aba_counts, aes(x = shoot_control_1)) +
  geom_histogram(bins = 100)

p2 <-   ggplot(aba_counts, aes(x = log10(shoot_control_1 + 1))) +
  geom_histogram(bins = 100)

p1 + p2
```




```
aba_counts %>%
  rownames_to_column('gene') %>%
```

```
  pivot_longer(- gene, names_to = "sample") %>%
  ggplot(., aes(x = log10(value+1), fill=sample)) +
  geom_density(alpha=0.2)
```



Conclusion: this means that the samples have comparable count distributions.

Exercise 2: performing a differential expression analysis and exploring the results.

Run the 'import samples to conditions' R code section. Run the 'differential expression analysis' R code section to get the R object called *shoot_deseq_results* that contains the results of the differential gene expression analysis for the shoot samples. It contains the results for **all** the genes present in the gene counts.

Here is a brief explanation of what each column refers too in the *shoot_deseq_results* R object:
- baseMean: mean of normalized counts for all samples - log2FoldChange: log2 fold change - lfcSE: standard error - stat: Wald statistic - pvalue: Wald test p-value - padj: BH adjusted p-values

(Execute the code below).

```
aba_samples_to_conditions <- read.csv("./ABA/samples_to_conditions_ABA.csv", stringsAsFactors = F)
```

(Execute the code below).

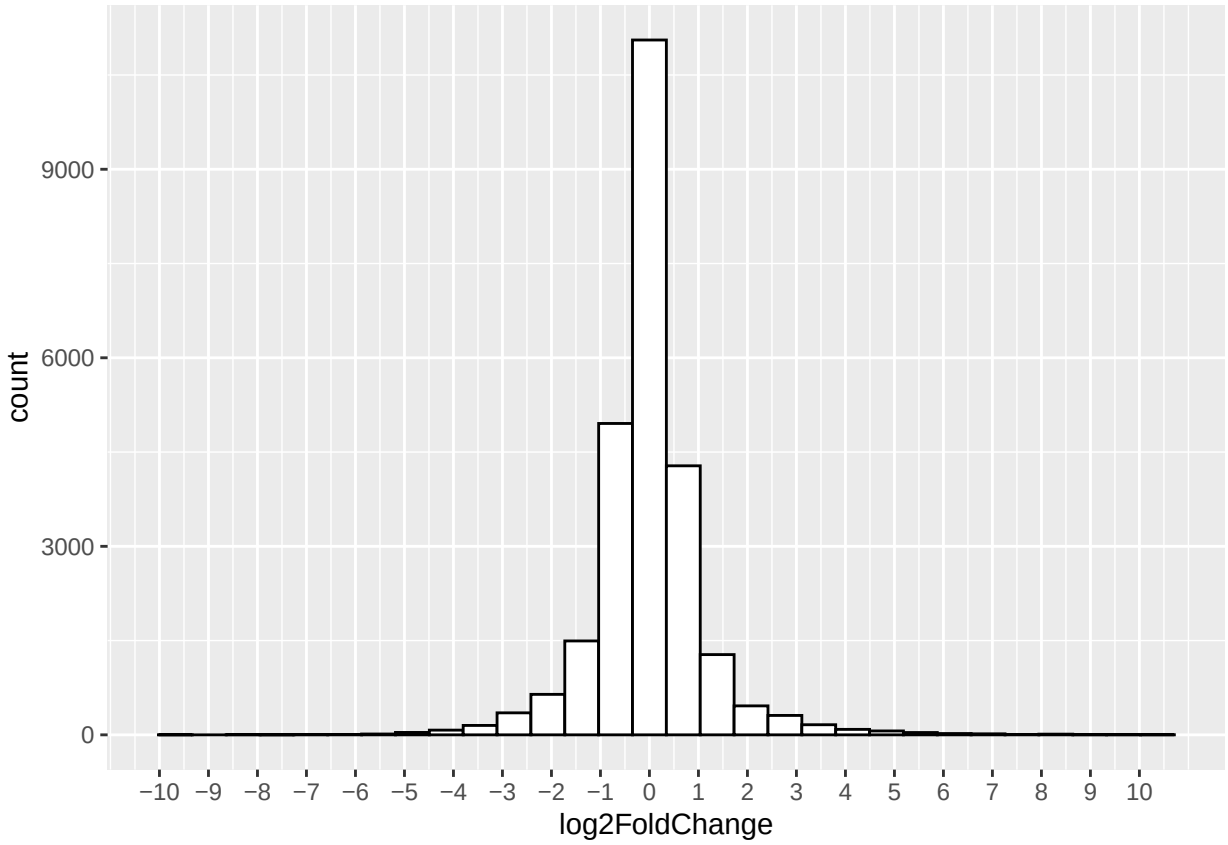
```
# import the required R objects into a new shoot_dds DESeqDataSet object
dds_aba <- DESeqDataSetFromMatrix(countData = aba_counts,
                                  colData = aba_samples_to_conditions,
                                  design = ~ treatment)
```

```r
# perform the diff. expr. analysis
dds_aba <- DESeq(dds_aba)

# extracts results for all genes
aba_deseq_results <- results(dds_aba, contrast = c("treatment", "ABA", "control")) %>%
  as.data.frame()
```

Question 3

Q3: Build a plot describing the distribution of the log2FoldChanges of all genes. Use a histogram. Describe the plot. What does a log2FoldChange of +1 means for a given gene? Explain in simple terms.

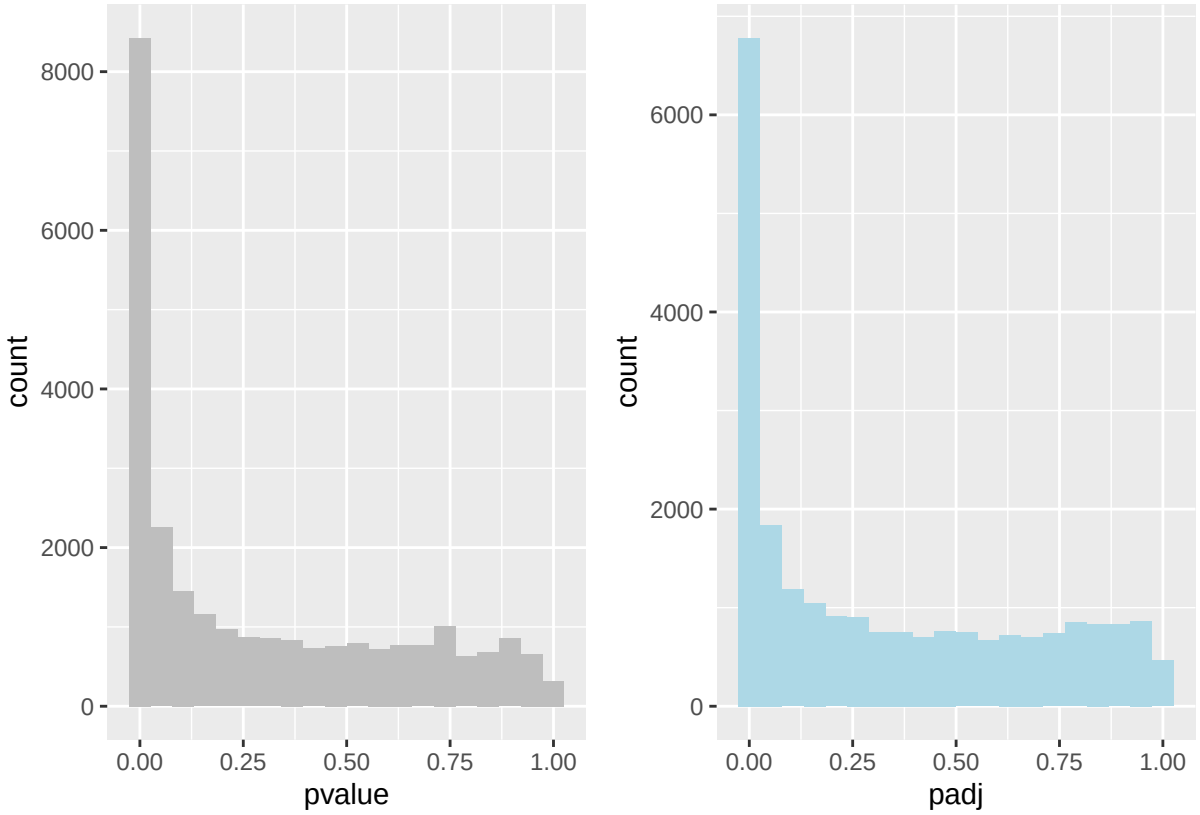
```r
ggplot(aba_deseq_results, aes(x = log2FoldChange)) +
  geom_histogram(bins = 30, color="black", fill="white") +
  scale_x_continuous(breaks = seq(-10,10,1))
```



Question 4

Q4: search online type I error. Justify why this is needed in the case of a differential gene expression analysis. Plot the distribution of the raw p-values and the adjusted p-values using a histogram with 20 bins. Do the same with the distribution of the adjusted p-values. Do you notice differences between the two plots?

```r
p1 <- ggplot(aba_deseq_results, aes(x=pvalue)) +geom_histogram(bins = 20, fill="grey")
p2 <- ggplot(aba_deseq_results, aes(x=padj)) +geom_histogram(bins = 20, fill="lightblue")
p1 + p2
```



Q6: filter the *aba_deseq_results* to keep only genes with a log2FoldChange higher than +2 and (positively regulated by ABA) and considered differential (padj < 0.01). How many genes do you find? What does a log2FoldChange of +2 means for gene X in this experiment? Explain in simple terms.

```
aba_diff_genes <-
  aba_deseq_results %>%
  filter(log2FoldChange > +2) %>%
  filter(padj < 0.01) %>%
  rownames_to_column("gene")
nrow(aba_diff_genes)
## [1] 595
```

Answer: 595 genes.

Answer: A log2FoldChange of +1 means that the gene is two times more abundant in the ABA condition than in the control condition.

```
gene_annotations <- read.csv("TAIR10_functional_descriptions.csv", stringsAsFactors = F) %>%
  select(- gene_model) %>%
  distinct()

# This will add a human-readable description to your list of diff. genes
aba_diff_genes_with_description <- inner_join(x = aba_diff_genes,
                                              y = gene_annotations,
                                              by = "gene")
```

Save this file to a .csv file.

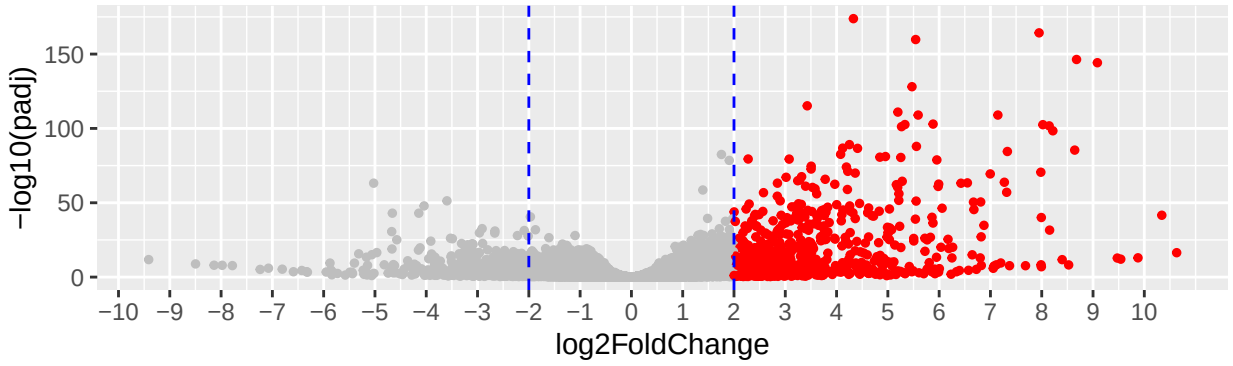
```
write.csv(x = aba_diff_genes_with_description, file = "ABA/aba_diff_genes.csv", row.names = FALSE)
```

Exercise 3: Data vizualisation of a differential expression analysis

Q6: A classic way to represent the results of a differential gene expression analysis is to create a so-called volcano plot. Find online what is a volcano plot related to gene expression. Give a short definition of a volcano plot and describe what is mapped to the x-axis and y-axis?

Q7: Build a volcano plot using the *shoot_deseq_results* R object. Describe the plot briefly and explain in what portion of the volcano plot you expect to find the most interesting ABA-responsive genes? Justify your answer.

```
ggplot(aba_deseq_results, aes(x = log2FoldChange, y = -log10(padj))) +
  geom_point(size=1, color=ifelse(aba_deseq_results$log2FoldChange>2, "red", "grey")) +
  geom_vline(xintercept = -2, color="blue", linetype="dashed") +
  geom_vline(xintercept = +2, color="blue", linetype="dashed") +
  scale_x_continuous(breaks = seq(-10, 10, 1))
```



Exercise 4: over-representation analysis of the differentially expressed genes.

To go beyond a list of genes and be able to provide a more meaningful interpretation

Go to ShinyGO (http://bioinformatics.sdstate.edu/go/) in your web browser and copy-paste the list of genes identifiers from your list of differential genes.

Exercise 5: select 3 promising genes

Filter the list of differential genes to select the most promising genes. You can filter on: 1) The log2FoldChange: the higher the more up-regulated your gene is. 2) The adjusted p-value: the smaller the better. 3) The baseMean (mean of expression in all samples): the higher the more expressed is your gene.

And of course you can combine these 3 criteria.

Part II: jasmonate analysis and comparison with ABA

Import data (execute code below)

```
meja_counts <- read.csv("./MeJA/shoot_gene_counts_MeJA.csv", stringsAsFactors = F) %>%
  column_to_rownames("gene")
head(meja_counts)
##           shoot_control_1 shoot_control_2 shoot_control_3 shoot_MeJA_1
## AT1G01010             510             830             556         598
## AT1G01020            1094            1502            1082        1595
## AT1G01030             337             568             314         866
## AT1G01040            2489            3218            2157        4197
## AT1G01046              75              47              41         115
```

```
## AT1G01050            3402            5508            2468         4215
##           shoot_MeJA_2 shoot_MeJA_3
## AT1G01010          687          618
## AT1G01020         1368         1639
## AT1G01030          913          599
## AT1G01040         2954         3300
## AT1G01046           31           65
## AT1G01050         3857         3259
```

Import samples to conditions correspondence for MeJA (execute code below)

```
meja_samples_to_conditions <- read.csv("MeJA/samples_to_conditions_MeJA.csv")
meja_samples_to_conditions
##            sample treatment tissue replicate
## 1 shoot_control_1   control  shoot      rep1
## 2 shoot_control_2   control  shoot      rep2
## 3 shoot_control_3   control  shoot      rep3
## 4    shoot_MeJA_1      MeJA  shoot      rep1
## 5    shoot_MeJA_2      MeJA  shoot      rep2
## 6    shoot_MeJA_3      MeJA  shoot      rep3
```

```
dds_meja <- DESeqDataSetFromMatrix(countData = meja_counts,
                                   colData = meja_samples_to_conditions,
                                   design = ~ treatment)

# perform the diff. expr. analysis
dds_meja <- DESeq(dds_meja)

# extracts results for all genes
meja_deseq_results <- results(dds_meja, contrast = c("treatment", "MeJA", "control")) %>%
  as.data.frame()
```

```
meja_diff_genes <-
  meja_deseq_results %>%
  filter(log2FoldChange > +2) %>%
  filter(padj < 0.01) %>%
  rownames_to_column("gene")
nrow(meja_diff_genes)
## [1] 191
```

```
# This will add a human-readable description to your list of diff. genes
meja_diff_genes_with_description <- inner_join(x = meja_diff_genes,
                                               y = gene_annotations,
                                               by = "gene")
write.csv(meja_diff_genes_with_description, file = "MeJA/diff_genes_meja.csv", row.names = F)
```