Deforges 2023 rna-seq analysis

Marc Galland

2023-01-27

Contents

Assignment objectives **1**

Setup **1**
Add a table of contents 1
Disabling warnings and messages 1

Exercise 1: Dataset description **3**
Import gene counts 3
Descriptive metrics 3

Exercise 2: Volcano plot **5**
Get results for all genes 5
Volcano plot 5

Exercise 3: enrichment analysis **6**
Get the list of up-regulated genes (not part of assignment) 6
Import table of differentially expressed genes 8
Exercise 4: select 10 genes and plot them 8

This is the assignment for the Big Data course 2023 given to the students.

Assignment objectives

Goal 1: working in R with "big" datasets: reading, filtering, etc. **Goal 2:** working with the "simplified" output of a simple RNA-seq experiment (around 1000 genes). **Goal 3:** combining info from different tables (gene counts and diff. genes). **Goal 4:** literate programming = combining code + figures + explanations. **Goal 5:** selecting 10 genes from 1000 genes that you will further explore using dedicated databases (Michel) and related to your biological question/XP design.

Setup

Add a table of contents

Link to R Markdown guide (click me)

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.

```r
knitr::opts_chunk$set(echo = TRUE,
                      warning = FALSE,
                      message = FALSE,
                      collapse = TRUE)
```

Exercise 1: Dataset description

We first load the `tidyverse` package that contains most of the data transformation functions we will need.

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

Import gene counts

- Use the following code to import your data.

```
gene_counts <- read.csv("gene_counts.csv",
                        header = TRUE,
                        stringsAsFactors = FALSE) %>%
  # for DESeq subsequent data import
  column_to_rownames("gene") %>%
  as.matrix()

# first five rows
head(gene_counts, n = 5)
##           root_control_1 root_control_2 root_control_3 root_IAA_1 root_IAA_2
## AT1G01010           2029           1481           2694       2450       1767
## AT1G01020           1626           1608           1895       1816       2429
## AT1G01030            150            230            375        149        175
## AT1G01040           3174           2599           4260       3753       2419
## AT1G01046             70             42            115         67         45
##           root_IAA_3
## AT1G01010       2166
## AT1G01020       1716
## AT1G01030        260
## AT1G01040       3838
## AT1G01046         89
```

Descriptive metrics

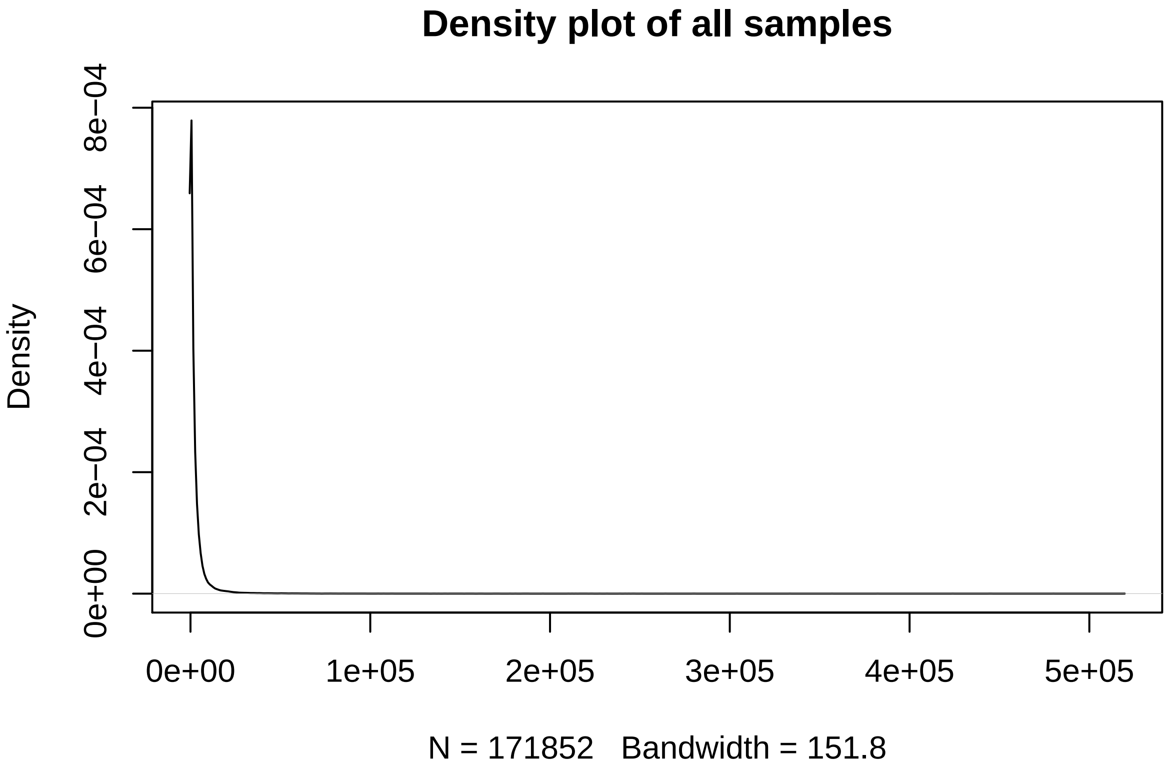
Q1: compute a series of basic descriptive metrics on a given RNA-seq dataset: - What is the maximum gene count value? - What is the minimum gene count value? - What is the median gene count value?

```
max(gene_counts)
## [1] 519267
min(gene_counts)
## [1] 0
median(gene_counts)
## [1] 699.5
```

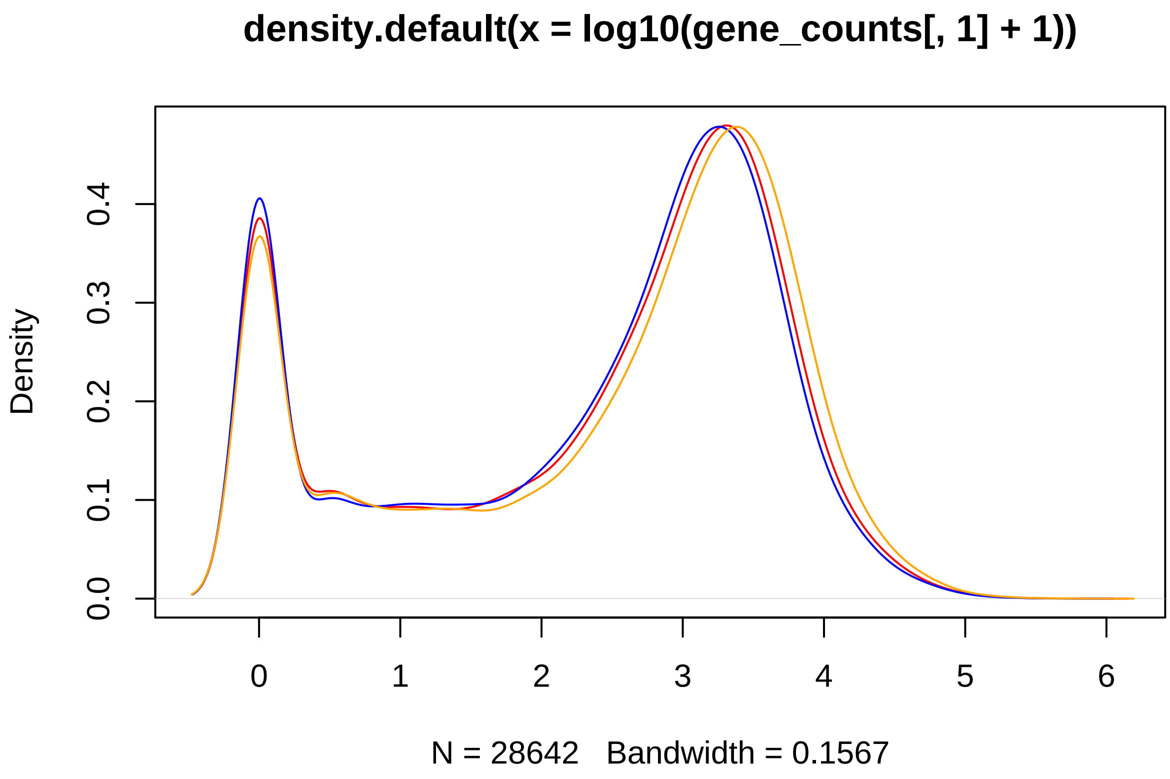
Q2: produce a distribution of the count values.

- How is the data distributed for all samples? Create a histogram of gene count values distribution. - What transformation could you do to normalize this data distribution (i.e. to make it more Gaussian)? - Overlay the distribution of the three control samples. Are they comparable?

```
plot(density(gene_counts), main = "Density plot of all samples")
```



```
# overlay three samples
plot(density(log10(gene_counts[,1]+1)), col="red")
lines(density(log10(gene_counts[,2]+1)), col="blue")
lines(density(log10(gene_counts[,3]+1)), col="orange")
```



Exercise 2: Volcano plot

A Volcano plot is a classic figure used to display the result of a differential expression analysis.

Get results for all genes

```
sample2condition <- read.csv("../gene_counts_and_samples2conditions/arabidopsis_root_hormones_sample2con
                             header = TRUE,
                             stringsAsFactors = FALSE) %>%
  filter(condition == "IAA" | condition == "control")
```

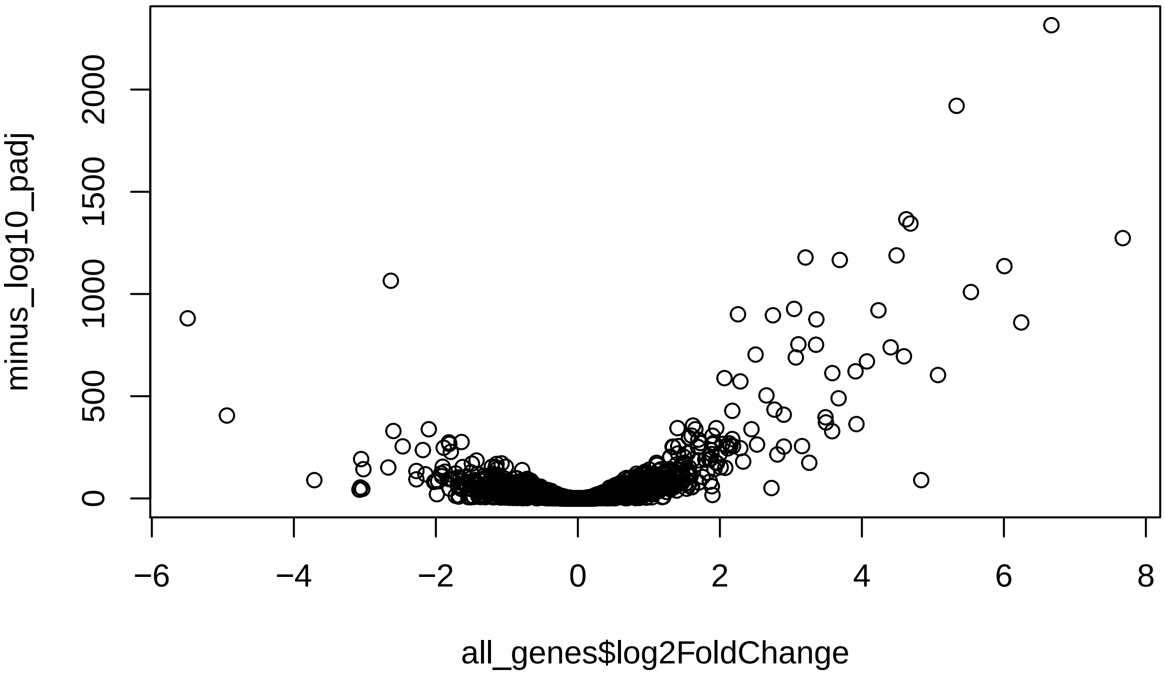
```
dds <- DESeqDataSetFromMatrix(countData = gene_counts,
                              colData = sample2condition,
                              design = ~ condition)
dds <- DESeq(dds)
```

```
all_genes <- results(dds) %>% as.data.frame() %>% rownames_to_column("gene") %>% filter(baseMean > medi
write.csv(x = all_genes, file = "results_all_genes.csv", row.names = FALSE)
```

Volcano plot

Q3: volcano plot - Why a Volcano plot is often used in differential gene expression analysis?
- What do you place on the x-axis and the y-axis?
- Make a Volcano plot based on the all_genes object. Select two thresholds that you could use to identify genes differentially regulated in response to auxin (one threshold for the x-axis and one for the y-axis).

```
minus_log10_padj <- -10*log10(all_genes$padj)
plot(x = all_genes$log2FoldChange, y = minus_log10_padj)
```



Exercise 3: enrichment analysis

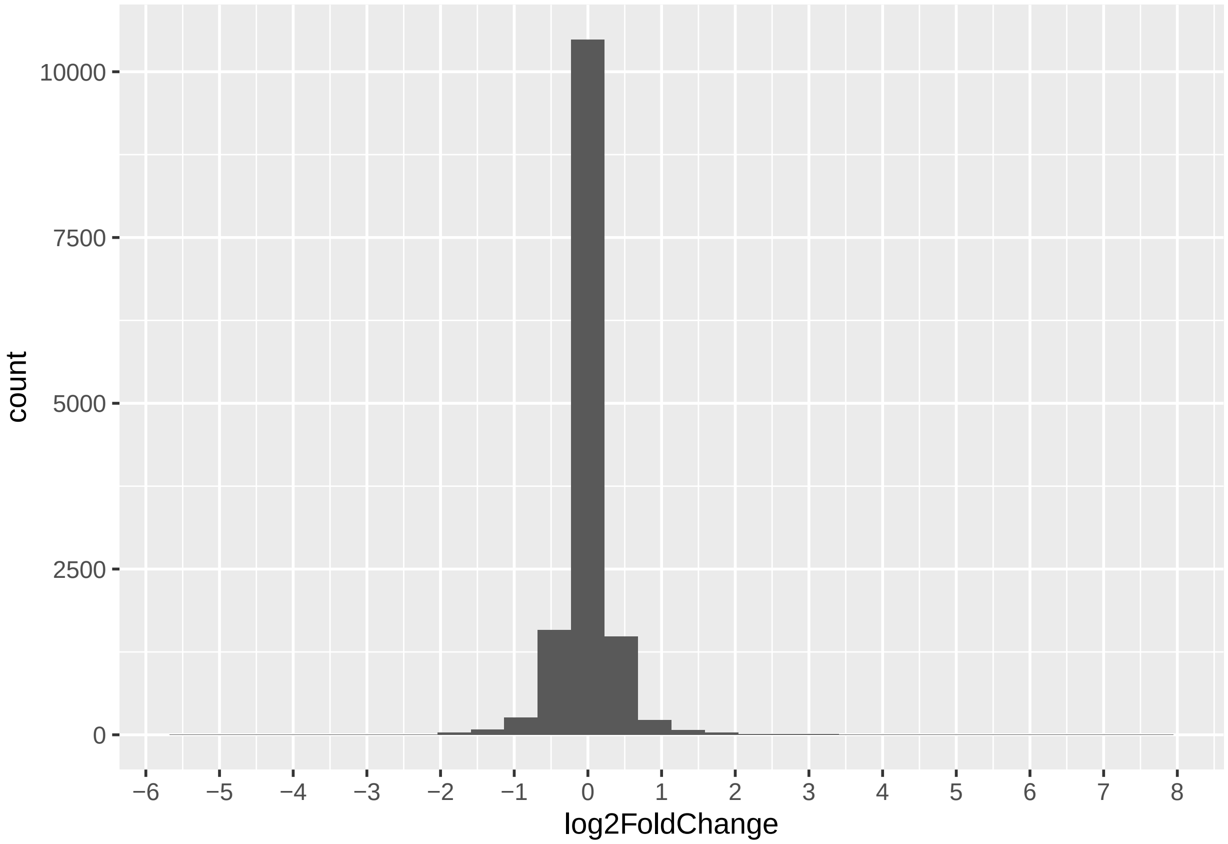
In this exercise, the list of genes up-regulated in response to auxin is searched for statistically enriched Gene Ontology (GO) categories. In order to interpret the biological pathways and functions that are affected by the auxin treatment.

Get the list of up-regulated genes (not part of assignment)

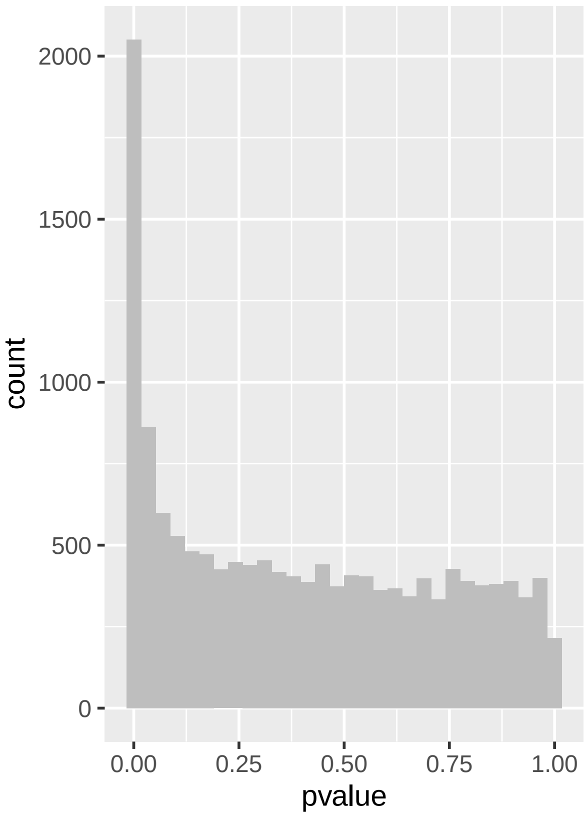
Q4: Import output of diff. gene analysis called "results_all_genes.csv"
- How many genes are present in the differential genes? - Make a histogram of the log2 fold change values. - Make a histogram of the raw p-values (pvalue column) and adjusted p-values (padj column).

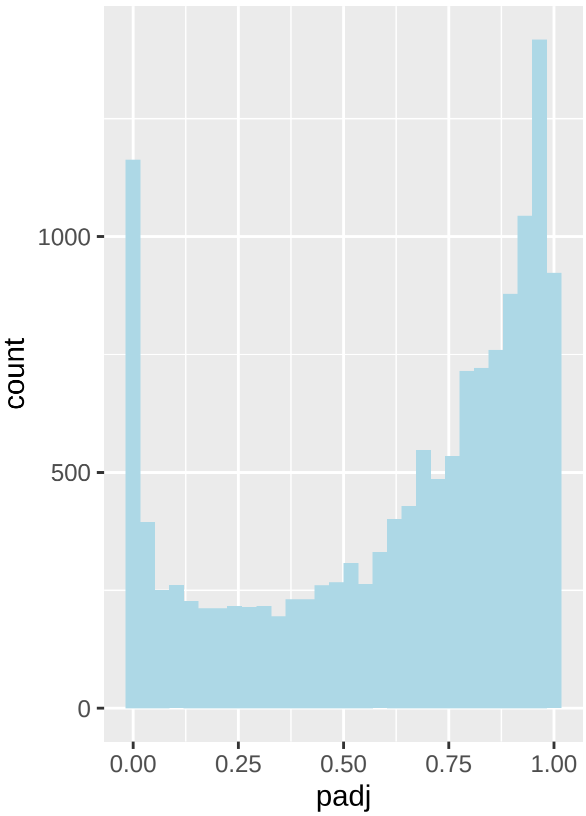
```
all_genes <- read.csv("results_all_genes.csv", stringsAsFactors = F)
nrow(all_genes)
## [1] 14321
```

```
ggplot(all_genes, aes(x = log2FoldChange)) +
  geom_histogram() +
  scale_x_continuous(breaks = seq(-8, +8, 1))
```



```
library("patchwork")
p1 <- ggplot(all_genes, aes(x = pvalue)) +
  geom_histogram(fill="grey")
p2 <- ggplot(all_genes, aes(x = padj)) +
  geom_histogram(fill="lightblue")
p1 + p2
```





Import table of differentially expressed genes

Based on the histogram of log2 fold changes and the volcano plot, a threshold of -1/+1 for the log2FC seems good to select genes. A FDR of 0.01 is often used.

```
diff_genes <- results(dds) %>%
  as.data.frame() %>%
  rownames_to_column("gene") %>%
  filter(padj < 0.05) %>%
  filter(log2FoldChange > +1) %>%
  filter(baseMean > 1000)
write.csv(diff_genes, file = "diff_genes.csv", row.names = FALSE, quote = FALSE)

diff_genes <- read.csv("diff_genes.csv", stringsAsFactors = F)
```

Exercise 4: select 10 genes and plot them

Arranged by decreasing fold change

```
twenty_genes <- diff_genes %>%
  arrange(desc(log2FoldChange), desc(baseMean)) %>%
  head(n=20)
print.data.frame(twenty_genes)
##         gene  baseMean log2FoldChange     lfcSE      stat         pvalue
## 1  AT3G58190  3921.939       7.674375 0.3143347 24.414665 1.194906e-131
## 2  AT2G23170 66400.262       7.625918 0.1876534 40.638307  0.000000e+00
## 3  AT4G37390 38155.491       6.668904 0.2031833 32.822100 2.849459e-236
## 4  AT3G10870  3042.296       6.244112 0.3105858 20.104307  6.765936e-90
## 5  AT2G45420  2203.610       5.535856 0.2544301 21.757864 5.819721e-105
## 6  AT4G37295  1887.586       5.071886 0.3005574 16.874936  6.879904e-64
## 7  AT2G41850  1299.186       4.835829 0.7169774  6.744743  1.532978e-11
## 8  AT4G27260 26004.160       4.683959 0.1867231 25.085059 7.239757e-139
## 9  AT3G15540  4662.038       4.625266 0.1829427 25.282591 4.964609e-141
## 10 AT4G37290  1255.412       4.592326 0.2538247 18.092510  3.650738e-73
## 11 AT2G39370  4868.703       4.488461 0.1902009 23.598531 3.990354e-123
## 12 AT2G14960  4075.768       4.404746 0.2361934 18.648897  1.289091e-77
## 13 AT1G60010  4957.196       3.688673 0.1578485 23.368446 8.950015e-121
## 14 AT5G52900  1343.282       3.583623 0.2107514 17.004032  7.666036e-65
## 15 AT5G65320  1345.882       3.581318 0.2852544 12.554820  3.740292e-36
## 16 AT3G62100  1051.064       3.487866 0.2534489 13.761614  4.337352e-43
## 17 AT3G14362  1733.389       3.358833 0.1656358 20.278421  1.994292e-91
## 18 AT4G13195  2848.528       3.354159 0.1783616 18.805387  6.822228e-79
## 19 AT1G74110  1012.797       3.259106 0.3522701  9.251726  2.208965e-20
## 20 AT4G28640  3358.725       3.205611 0.1364438 23.494001 4.697387e-122
##             padj
## 1  4.600986e-128
## 2   0.000000e+00
## 3  3.291553e-232
## 4   8.227022e-87
## 5  1.120442e-101
## 6   4.295849e-61
## 7   1.073224e-09
## 8  3.345202e-135
## 9  2.867434e-137
## 10  3.012250e-70
## 11 1.316988e-119
## 12  1.191275e-74
## 13 2.297469e-117
## 14  4.919679e-62
## 15  1.309272e-33
## 16  1.964820e-40
## 17  2.559673e-88
## 18  6.567247e-76
## 19  3.568792e-18
## 20 1.356547e-118
```

This is the list for students.r

```
twenty_genes %>%  select(gene) %>% print.data.frame()
##         gene
## 1  AT3G58190
## 2  AT2G23170
## 3  AT4G37390
## 4  AT3G10870
## 5  AT2G45420
## 6  AT4G37295
## 7  AT2G41850
## 8  AT4G27260
## 9  AT3G15540
## 10 AT4G37290
## 11 AT2G39370
## 12 AT2G14960
## 13 AT1G60010
## 14 AT5G52900
## 15 AT5G65320
## 16 AT3G62100
## 17 AT3G14362
## 18 AT4G13195
## 19 AT1G74110
## 20 AT4G28640
```

Arrange by smallest adjusted p-value

```
twenty_genes2 <- diff_genes %>%
  arrange(desc(padj)) %>%
  head(n=20)
print.data.frame(twenty_genes2)
##         gene  baseMean log2FoldChange     lfcSE     stat       pvalue
## 1  AT2G19970  2525.369       1.895696 0.6095564 3.109960 1.871129e-03
## 2  AT1G49310  3159.251       1.043441 0.3105746 3.359710 7.802419e-04
## 3  AT2G18980 12088.034       1.001391 0.2734268 3.662374 2.498883e-04
## 4  AT1G10380  2029.234       1.039728 0.2812623 3.696650 2.184633e-04
## 5  AT2G24430  1387.523       1.247068 0.2988708 4.172599 3.011442e-05
## 6  AT4G37370  1344.435       1.143437 0.2584222 4.424688 9.658194e-06
## 7  AT4G36120  1080.255       1.111435 0.2462304 4.513800 6.367637e-06
## 8  AT2G35770  1385.269       1.057420 0.2288331 4.620924 3.820353e-06
## 9  AT2G41380  7628.933       1.014615 0.2029023 5.000510 5.717872e-07
## 10 AT5G55050  1279.563       1.531007 0.3056298 5.009351 5.461383e-07
## 11 AT1G75640  2501.308       1.168479 0.2312116 5.053720 4.332858e-07
## 12 AT4G30140 13345.048       2.727002 0.5280265 5.164517 2.410604e-07
## 13 AT5G60580  2484.636       1.023107 0.1963787 5.209868 1.889746e-07
## 14 AT1G71380  1212.097       1.192436 0.2272253 5.247811 1.539171e-07
## 15 AT3G14620  1182.501       1.211792 0.2274148 5.328552 9.899873e-08
## 16 AT4G37900  2558.439       1.599809 0.2988282 5.353607 8.621812e-08
## 17 AT1G64400  1379.695       1.359394 0.2516354 5.402237 6.581486e-08
## 18 AT4G30080  4767.754       1.129443 0.2080080 5.429802 5.641664e-08
## 19 AT5G14130  1528.771       1.260650 0.2269664 5.554345 2.786551e-08
## 20 AT3G61160  1762.541       1.177887 0.2114416 5.570742 2.536571e-08
##            padj
## 1  2.115942e-02
## 2  1.031232e-02
## 3  3.921990e-03
## 4  3.492843e-03
## 5  6.173323e-04
## 6  2.256150e-04
## 7  1.571704e-04
## 8  9.956296e-05
## 9  1.802183e-05
## 10 1.726051e-05
## 11 1.403956e-05
## 12 8.178000e-06
## 13 6.545548e-06
## 14 5.379648e-06
## 15 3.573699e-06
## 16 3.136846e-06
## 17 2.444567e-06
## 18 2.126254e-06
## 19 1.115731e-06
## 20 1.026312e-06
```

Arrange by highest baseMean

```
twenty_genes3 <- diff_genes %>%
  arrange(desc(baseMean)) %>%
  head(n = 20)
twenty_genes3
##         gene  baseMean log2FoldChange     lfcSE      stat        pvalue
## 1  AT4G34710 72178.288       1.000517 0.1398746  7.152955  8.492920e-13
## 2  AT2G23170 66400.262       7.625918 0.1876534 40.638307  0.000000e+00
## 3  AT5G65670 58995.380       1.341632 0.1219285 11.003430  3.678673e-28
## 4  AT4G37390 38155.491       6.668904 0.2031833 32.822100 2.849459e-236
## 5  AT5G06865 27179.374       1.693266 0.1537091 11.016041  3.198181e-28
## 6  AT5G06860 27176.992       1.693353 0.1537484 11.013789  3.279152e-28
## 7  AT4G27260 26004.160       4.683959 0.1867231 25.085059 7.239757e-139
## 8  AT4G30140 13345.048       2.727002 0.5280265  5.164517  2.410604e-07
## 9  AT1G69530 13060.250       1.090465 0.1522962  7.160156  8.058519e-13
## 10 AT2G18980 12088.034       1.001391 0.2734268  3.662374  2.498883e-04
## 11 AT5G54510 11630.733       3.105393 0.1648941 18.832651  4.078254e-79
## 12 AT3G23030 11415.626       2.254970 0.1096378 20.567452  5.371512e-94
## 13 AT1G80240 10544.680       1.621038 0.1240755 13.064935  5.222936e-39
## 14 AT1G78100 10156.014       2.746628 0.1339130 20.510540  1.733585e-93
## 15 AT1G23080  9813.848       1.729189 0.1513374 11.426054  3.098734e-30
## 16 AT3G07390  9711.700       2.523475 0.2237856 11.276307  1.718082e-29
## 17 AT5G47060  9029.432       1.107702 0.1198306  9.243895  2.376842e-20
## 18 AT1G02850  8772.239       2.064658 0.1238880 16.665511  2.334859e-62
## 19 AT2G33310  8061.449       2.288431 0.1391899 16.441073  9.718373e-61
## 20 AT1G02900  7656.383       1.296305 0.1952841  6.638048  3.178641e-11
##             padj
## 1   7.240293e-11
## 2   0.000000e+00
## 3   8.171960e-26
## 4  3.291553e-232
## 5   7.243880e-26
## 6   7.355169e-26
## 7  3.345202e-135
## 8   8.178000e-06
## 9   6.895406e-11
## 10  3.921990e-03
## 11  4.096517e-76
## 12  8.273203e-91
## 13  2.193918e-36
## 14  2.503188e-90
## 15  8.422360e-28
## 16  4.361852e-27
## 17  3.813346e-18
## 18  1.419533e-59
## 19  5.757014e-58
## 20  2.092863e-09
```