Auxin dataset analysis from Deforges 2019

Marc Galland

2022-03-03

Contents

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.

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

Introduction

Q1 (0.5 point): In the publication of Deforges et al. 2019, the first paragraph of the “Material and Methods” section describes how plants were sown and treated (see paragraph below). What are the full names of each of the 4 hormones used? “ ” ” Arabidopsis thaliana seeds were germinated on agar-solidified half-strength MS medium for 10 d, after which the seedlings were flooded with a solution of half-strength MS containing 5 µM IAA, 10 µM ABA, 10 µM MeJA, 10 µM ACC, or no hormone for the untreated control. After 3 h of incubation, roots and shoots were split and harvested separately. For each of the 12 experimental conditions, 3 independent biological replicates were carried out at different times. “ ” ” Names are: - IAA: auxin,
- ABA: abscisic acid, - ACC: 1-amino-1-cyclopropane, - MeJA: methyl jasmonate.

Q2 (0.5 point): Can you name 3 different types of RNA that can be studied using RNA sequencing? - Long non-coding RNAs.
- Messenger RNAs.
- microRNAs.

Q3 (0.5 point): What is the sequencing platform used in this experiment? How many reads were obtain on average per sample?
Illumina HiSeq 2500, about 30 million reads per sample.
> “The libraries were sequenced on a HiSeq 2500 Illumina sequencer and about 30 million of paired-end reads per sample were obtained.”

Q4: (0.5 point): In the article, can you find a good complete one-sentence long definition of cis-NATs?
>“Cis-Natural Antisense Transcripts (cis-NATs), which overlap protein coding genes and are transcribed from the opposite DNA strand, constitute an important group of noncoding RNAs.”

Exercise 1: data import

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

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

Import gene counts

```r
raw_counts <- read.csv(
  file = "../gene_counts_and_samples2conditions/dataset01_IAA_arabidopsis_root_raw_counts.csv",
  header = TRUE,
  stringsAsFactors = FALSE) %>%
  # for DESeq subsequent data import
  column_to_rownames("gene")

# first five rows
knitr::kable(head(raw_counts, n = 5))
```

	root_control_1	root_control_2	root_control_3	root_IAA_1	root_IAA_2	root_IAA_3
AT1G01010	2029	1481	2694	2450	1767	2166
AT1G01020	1626	1608	1895	1816	2429	1716
AT1G01030	150	230	375	149	175	260
AT1G01040	3174	2599	4260	3753	2419	3838
AT1G01046	70	42	115	67	45	89

Q5: Can you determine how many genes are present in the table?

```
nrow(raw_counts)
## [1] 28642
```

There are **28642** genes in the "dataset_01_IAA_arabidopsis_root_raw_counts.csv" table.

Q6 (1 point): determine the minimum and maximum gene expression in the control condition and in the hormone-treated condition.
Hint: if you use the tidyverse package to do this, use "pivot_longer()" to get your data tidy and create a new column for the biological replicate number. Find the gene that has the maximum count in the control condition and auxin-treated conditions.

```
raw_counts %>%
  rownames_to_column("gene") %>%
  pivot_longer(- gene, values_to = "counts", names_to = "sample") %>%
  separate(sample, into = c("tissue","condition","rep"), sep = "_") %>%
  group_by(condition) %>%
  summarise(minimum = min(counts),
            maximum = max(counts)) %>%
  knitr::kable()
```

condition	minimum	maximum
control	0	519267
IAA	0	444483

Gene with maximum expression in control condition.

```
# 519267
raw_counts %>%
  rownames_to_column("gene") %>%
  pivot_longer(- gene, values_to = "counts", names_to = "sample") %>%
  separate(sample, into = c("tissue","condition","rep"), sep = "_") %>%
  filter(counts == "519267")
## # A tibble: 1 x 5
##   gene      tissue condition rep   counts
##   <chr>     <chr>  <chr>     <chr>  <int>
## 1 AT3G09260 root   control   3     519267
```

Gene with maximum expression in auxin-treated condition.

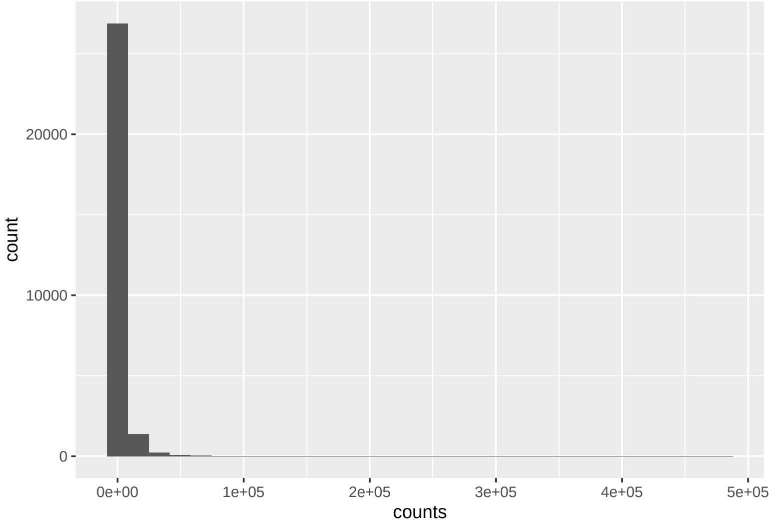
```
# 444483
raw_counts %>%
  rownames_to_column("gene") %>%
  pivot_longer(- gene, values_to = "counts", names_to = "sample") %>%
  separate(sample, into = c("tissue","condition","rep"), sep = "_") %>%
  filter(counts == "444483")
## # A tibble: 1 x 5
##   gene      tissue condition rep   counts
##   <chr>     <chr>  <chr>     <chr>  <int>
## 1 AT3G09260 root   IAA       1     444483
```

Conclusion: the same gene has the maximum expression in both conditions. From the TAIR website, it says: > "Encodes beta-glucosidase.The major constituent of ER bodies. One of the most abundant proteins in Arabidopsis seedlings."

Distribution of counts

Q7 (1 point):
A) plot the distribution of the gene counts of one sample using either the base R hist() function or the ggplot2 geom_histogram() function. What can you say about the distribution of the gene counts?
B) How can you display the distribution of an scale that better represents the distribution? Think about data transformation or the axis.

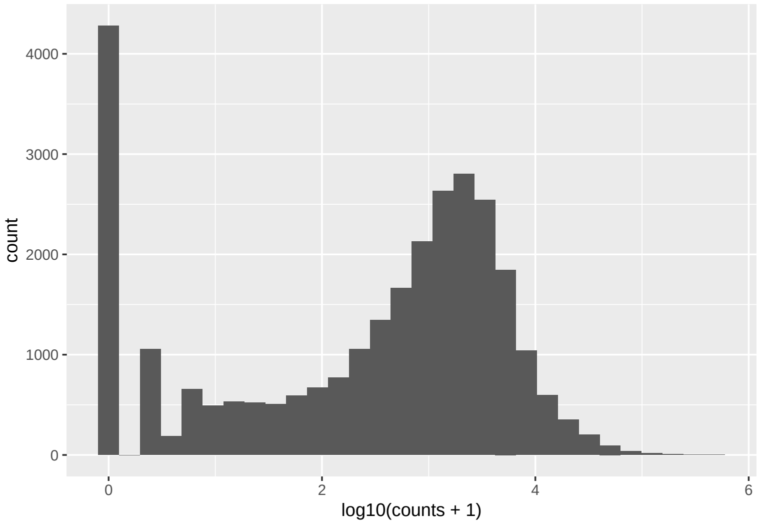
```
raw_counts %>%
  rownames_to_column("gene") %>%
  pivot_longer(- gene, values_to = "counts", names_to = "sample") %>%
  filter(sample == "root_control_1") %>%
  ggplot(., aes(x = counts)) +
  geom_histogram()
```



The count distribution is very skewed with a lot of data with counts < 50,000 counts. A few genes have a very high count number with the maximum being 479,700 counts.

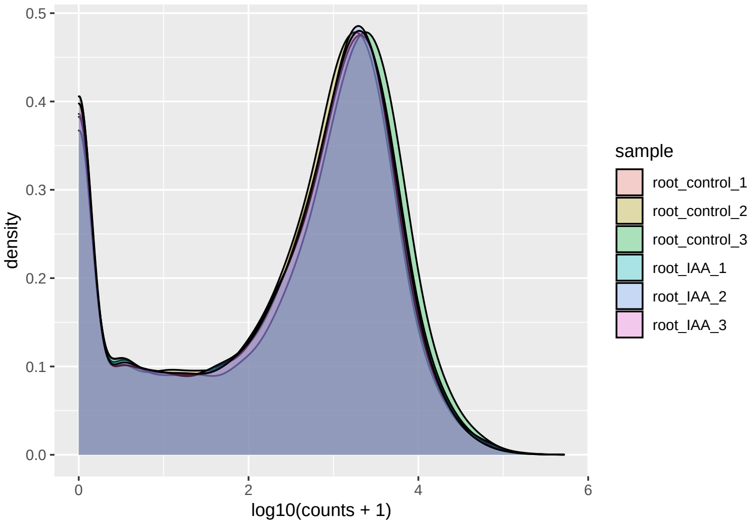
We can transform the data before plotting or use a transformed scale. Here, I use a log10 transformation with an offset of 1 (so that genes with 0 counts become log10(0+1)=0).

```
raw_counts %>%
  rownames_to_column("gene") %>%
  pivot_longer(- gene, values_to = "counts", names_to = "sample") %>%
  filter(sample == "root_control_1") %>%
  ggplot(., aes(x = log10(counts + 1))) + # log10 transform with offset
  geom_histogram()
```



Bonus: all samples overlaid.

```
raw_counts %>%
  rownames_to_column("gene") %>%
  pivot_longer(- gene, values_to = "counts", names_to = "sample") %>%
  ggplot(., aes(x = log10(counts + 1), fill = sample)) + # log10 transform with offset
  geom_density(alpha=0.3)
```



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

Exercise 2: differential expression

This `DESeqDataSet` object is used to store both data (gene counts) and metadata (sample to experimental condition correspondence) in one unique R object. Functions can be directly be applied to this object and corresponding results stored within the same object.

Import samples to condition

Q8 (0.5 point) Upload the "arabidopsis_root_hormones_samples_to_condition.csv" file to your R virtual machine. Then import this file into R and name the object samples_to_conditions. - Make a table that shows the number of biological replicates by condition. - Filter this table to keep only "control" samples and the samples related to your hormone of interest. You should have 6 samples in total (3 control and 3 hormone-treated).

```
samples_to_conditions <- read.csv(
  file = "../gene_counts_and_samples2conditions/arabidopsis_root_hormones_sample2conditions.csv",
  stringsAsFactors = F)

table(samples_to_conditions$condition)
##
##     ABA     ACC control     IAA    MeJA
##       3       3       3       3       3
```

```
samples_to_conditions_filtered <- filter(
  samples_to_conditions,
  condition == "IAA" | condition == "control")
samples_to_conditions_filtered
##           sample condition
## 1 root_control_1   control
## 2 root_control_2   control
## 3 root_control_3   control
## 4     root_IAA_1       IAA
## 5     root_IAA_2       IAA
## 6     root_IAA_3       IAA
```

Create the DESeqDataSetFromMatrix object

```
dds <- DESeqDataSetFromMatrix(countData = raw_counts,
                              colData = samples_to_conditions_filtered,
                              design = ~ condition)
```

You can have a quick peek at the number of genes, number of samples, etc. by calling the `dds` object.

```
dds
## class: DESeqDataSet
## dim: 28642 6
## metadata(1): version
## assays(1): counts
## rownames(28642): AT1G01010 AT1G01020 ... ATMG01400 ATMG01410
## rowData names(0):
## colnames(6): root_control_1 root_control_2 ... root_IAA_2 root_IAA_3
## colData names(2): sample condition
```

Call differential genes

Q9 (0.5 point): using the raw_counts and the samples_to_conditions, create a DESeqDataSet object called dds that will be used for DESeq2 differential analysis. Call the differential genes using the DESeq() function and call this object diff_genes. Filter this object to keep only the significantly differentially expressed genes (adjusted p-value < 0.01). Hint 1: the gene identifiers have to be assigned to the row names of raw_counts. Hint 2: convert the diff_genes object to a dataframe with "dds = as.data.frame(dds)".

```
dds <- DESeq(dds)
```

```
diff_genes <- results(dds, contrast = c("condition", "IAA", "control")) %>%
  as.data.frame() %>%
  filter(padj < 0.01)
head(diff_genes)
##             baseMean log2FoldChange     lfcSE       stat       pvalue
## AT1G01140  381.6898      0.7037351 0.2046857  3.438125 5.857581e-04
## AT1G01180  339.7232     -0.7970381 0.1465982 -5.436890 5.421857e-08
## AT1G01190  354.1737     -1.6711002 0.3863219 -4.325668 1.520701e-05
## AT1G01225  675.4774      0.5279669 0.1379490  3.827262 1.295764e-04
## AT1G01430 1491.2021     -0.4800477 0.1242080 -3.864869 1.111489e-04
## AT1G01750 2164.8321     -0.8342648 0.2329959 -3.580599 3.428076e-04
##                   padj
## AT1G01140 8.074445e-03
## AT1G01180 2.053462e-06
## AT1G01190 3.394469e-04
## AT1G01225 2.206326e-03
## AT1G01430 1.954241e-03
## AT1G01750 5.156175e-03
```

Number of diff genes and max log2FC

Q10 (0.5 point): how many genes are differentially expressed (adjusted p-value < 0.01)? How many genes are positively regulated in response to the hormone treatment?
How many genes are negatively regulated in response to the hormone treatment?

```
n_diff <- diff_genes %>% nrow()
pos <- diff_genes %>% filter(log2FoldChange > 0) %>% nrow()
neg <- diff_genes %>% filter(log2FoldChange < 0) %>% nrow()
```

- Total number of genes diff. regulated (padj < 0.01) is 1740 genes.

- Total number of genes *positively* & diff. regulated (padj < 0.01) is 849 genes.

- Total number of genes *negatively* & diff. regulated (padj < 0.01) is 891 genes.

Q11 (0.5 point) : display a table of the top 20 positively differentially expressed genes based on their log2 fold change.
What is the maximum positive log2fold change? Convert this log2 fold change to the untransformed fold change value = revert the log2 transformation.

```
top20 <-
  diff_genes %>%
  arrange(desc(log2FoldChange)) %>%
  head(n = 20)
knitr::kable(top20)
```

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
AT3G23635	34.247907	8.574702	1.2534645	6.840802	0.0000000	0.0000000
AT3G58190	3921.938845	7.674375	0.3143347	24.414665	0.0000000	0.0000000
AT2G17680	66.682582	7.669079	1.0726570	7.149610	0.0000000	0.0000000
AT2G23170	66400.261615	7.625918	0.1876534	40.638307	0.0000000	0.0000000
AT3G23637	9.870389	6.779288	1.4214319	4.769337	0.0000018	0.0000519
AT3G17760	161.227042	6.744007	0.5976111	11.284942	0.0000000	0.0000000
AT4G37390	38155.491125	6.668904	0.2031833	32.822100	0.0000000	0.0000000
AT2G43860	8.822320	6.618902	1.7387161	3.806776	0.0001408	0.0023742
AT3G49700	330.503040	6.600502	0.3770867	17.503937	0.0000000	0.0000000
AT1G06920	145.806542	6.476745	0.5110485	12.673445	0.0000000	0.0000000
AT2G03740	211.401405	6.341633	0.5323651	11.912187	0.0000000	0.0000000
AT2G10608	7.083708	6.301980	1.5296653	4.119843	0.0000379	0.0007550
AT1G15600	6.933524	6.270421	1.6851867	3.720906	0.0001985	0.0032116

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
AT3G10870	3042.295905	6.244112	0.3105858	20.104307	0.0000000	0.0000000
AT5G57520	804.027573	6.007091	0.2604378	23.065357	0.0000000	0.0000000
AT5G06080	530.656086	5.692930	0.2839566	20.048594	0.0000000	0.0000000
AT2G45420	2203.610321	5.535856	0.2544301	21.757864	0.0000000	0.0000000
AT4G32280	848.883488	5.334033	0.1782872	29.918206	0.0000000	0.0000000
AT1G21990	56.645231	5.270486	0.6023755	8.749502	0.0000000	0.0000000
AT5G19700	112.687486	5.147691	1.1440036	4.499716	0.0000068	0.0001660

Max log2 fold change is equal to 8.6. This corresponds to a fold change of 2^8.6 = 388. Therefore the AT3G23635 gene is 388 more expressed in auxin-treated seedlings than in control conditions.

Exercise 3: volcano plot

Shrinkage

Q12 (0.5 point): Shrink the log2 fold changes in order to shrink high log2 fold changes from lowly expressed genes. Use the related DESeq2 function that we have seen in the tutorial: https://scienceparkstudygroup.github.io/rna-seq-lesson/06-differential-analysis/index.html#3-volcano-plot
- First, extract the results completely with “results(dds, . . .)” - Then shrink the log2 fold changes with the “lfcShrink()” function and the “normal” shrinkage estimator. Call this new object res_shrink.

```r
all_gene_results <- results(dds, contrast = c("condition", "IAA", "control"))

resLFC <- lfcShrink(dds = dds,
                    res = all_gene_results,
                    type = "apeglm",
                    coef = 2)
```

Volcano plot

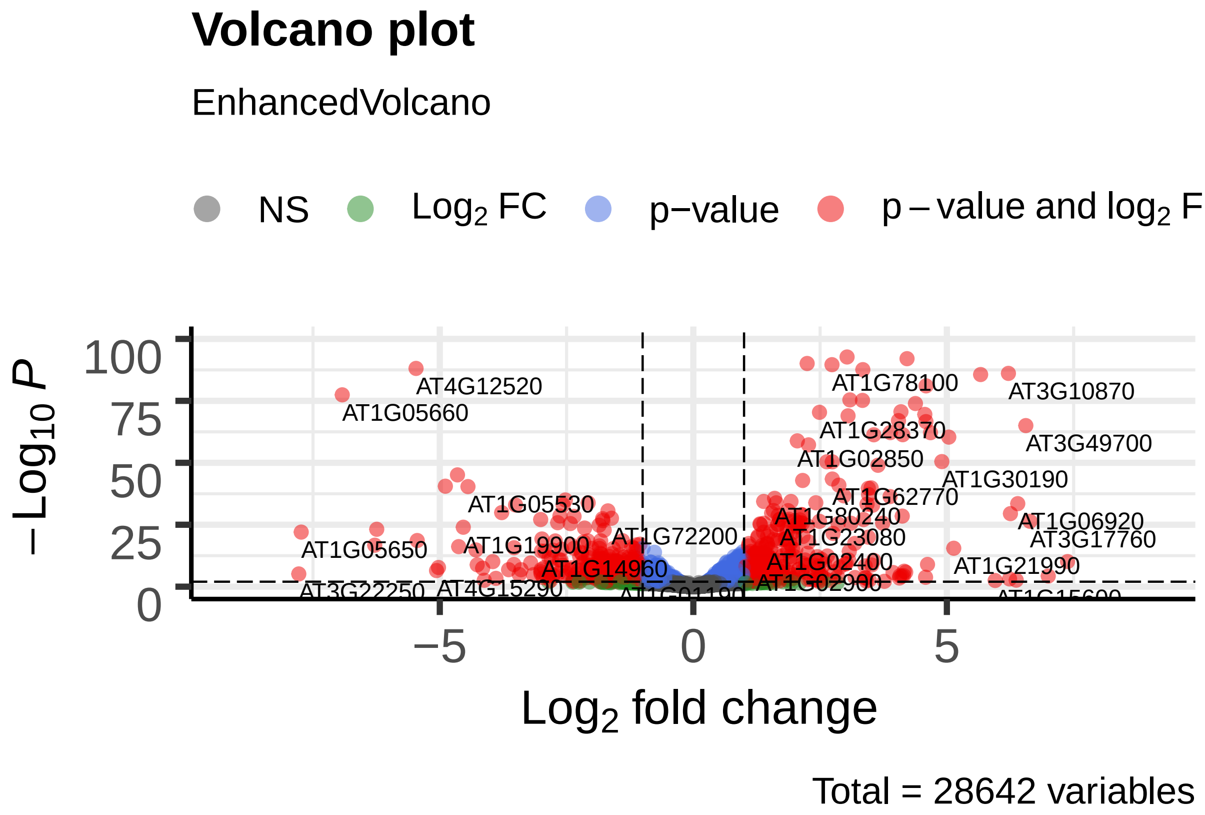
Q13 (1 point): make a volcano plot, a type of scatterplot that shows, for each gene, the magnitude of the shrinked log2 fold change (x-axis) versus statistical significance (adjusted p-value) as seen in the tutorial. Use the “res_shrink” object that you have built in the previous task. Hint: make sure you change the default “FCcutoff” and “pCutoff” values so that they better reflect your min/max log2 fold changes and adjusted p-values. Do not keep the default values for these two parameters.

What are the values to be set for the limits of x and y?

```r
min(resLFC$log2FoldChange, na.rm = T)
## [1] -7.780899
```

```r
max(resLFC$log2FoldChange, na.rm = T)
## [1] 9.336077
```

```r
library("EnhancedVolcano")
EnhancedVolcano(toptable = resLFC,
                x = "log2FoldChange",
                y = "padj",
                lab = rownames(resLFC),
                xlim = c(-9, +9),
                ylim = c(0,100),
                pCutoff = 0.01,
                transcriptPointSize = 2.0,
                FCcutoff = 1,
                title = "Volcano plot",
                legend=c(
                  'Not significant',
                  'Log2 fold-change (but do not pass p-value cutoff)',
                  'Pass p-value cutoff',
                  'Pass both p-value & Log2 fold change')) +
  guides(legend = NULL)
```



Q14: - Most up-regulated genes: top right of the volcano plot.
- Most down-regulated genes: top left of the volcano plot.
- Most statistically significant genes are the highest on the y-axis.

Exercise 4: find annotation of the 5 selected candidate genes

Q15 (1 point): for each of your 5 selected genes, make a plot showing their expression in control and hormone-treated seedlings.

Select based on:
1. Highest log2 fold change, then
2. Highest baseMean

```
diff_genes <- results(dds, contrast = c("condition", "IAA", "control"))
diff_genes %>%
  as.data.frame() %>%
  filter(log2FoldChange > 0) %>%
  arrange(desc(log2FoldChange)) %>%
  head(n = 20) %>%
  arrange(desc(baseMean))
##               baseMean log2FoldChange     lfcSE      stat        pvalue
## AT2G23170 66400.261616       7.625918 0.1876534 40.638307  0.000000e+00
## AT4G37390 38155.491125       6.668904 0.2031833 32.822100 2.849459e-236
## AT3G58190  3921.938846       7.674375 0.3143347 24.414665 1.194906e-131
## AT3G10870  3042.295905       6.244112 0.3105858 20.104307  6.765936e-90
## AT2G45420  2203.610321       5.535856 0.2544301 21.757864 5.819721e-105
## AT4G32280   848.883488       5.334033 0.1782872 29.918206 1.140843e-196
## AT5G57520   804.027573       6.007091 0.2604378 23.065357 1.031594e-117
## AT5G06080   530.656086       5.692930 0.2839566 20.048594  2.076288e-89
## AT3G49700   330.503040       6.600502 0.3770867 17.503937  1.336999e-68
## AT2G03740   211.401405       6.341633 0.5323651 11.912187  1.022599e-32
## AT3G17760   161.227042       6.744007 0.5976111 11.284942  1.557434e-29
## AT1G06920   145.806542       6.476745 0.5110485 12.673445  8.298845e-37
## AT2G17680    66.682582       7.669079 1.0726570  7.149610  8.702493e-13
## AT3G23635    34.247907       8.574702 1.2534645  6.840802  7.875105e-12
## AT1G25210    19.047448       6.755164 3.4246083  1.972536            NA
## AT3G23637     9.870389       6.779288 1.4214319  4.769337  1.848330e-06
## AT2G43860     8.822320       6.618903 1.7387161  3.806776  1.407903e-04
## AT2G10608     7.083708       6.301980 1.5296653  4.119843  3.791312e-05
## AT1G15600     6.933524       6.270421 1.6851867  3.720906  1.985092e-04
## AT1G51330     3.773501       5.390898 1.7517928  3.077361  2.088424e-03
##                    padj
## AT2G23170  0.000000e+00
## AT4G37390 3.291553e-232
## AT3G58190 4.600986e-128
## AT3G10870  8.227022e-87
## AT2G45420 1.120442e-101
## AT4G32280 8.785632e-193
## AT5G57520 2.383291e-114
## AT5G06080  2.398424e-86
## AT3G49700  9.652719e-66
## AT2G03740  3.281266e-30
## AT3G17760  3.997934e-27
## AT1G06920  3.092390e-34
## AT2G17680  7.391680e-11
## AT3G23635  5.757549e-10
## AT1G25210            NA
## AT3G23637  5.188573e-05
## AT2G43860  2.374217e-03
## AT2G10608  7.549555e-04
## AT1G15600  3.211596e-03
## AT1G51330  2.308557e-02
```

AT2G23170 = GH3.3
AT4G37390 = AUR3 Auxin upregulated 3 GH3-2
AT3G58190 = SYMMETRIC LEAVES 2-LIKE 16
AT3G10870 = ARABIDOPSIS THALIANA METHYL ESTERASE 17
AT2G45420 = LOB DOMAIN-CONTAINING PROTEIN 18

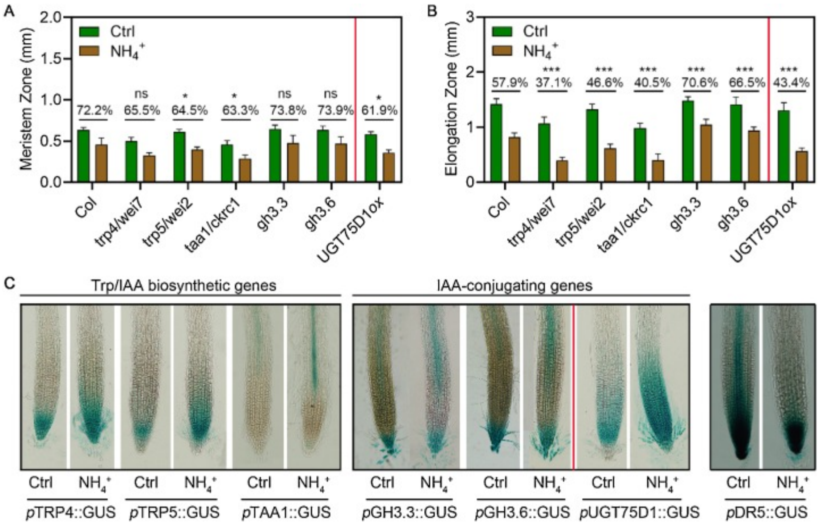
AT2G23170

Names: Auxin-responsive GRETCHEN HAGEN3.3 (GH3.3) protein
Description: conjugates amino acids to auxin and regulates its homeostasis.

> "The production of amide-linked IAA-amino acid conjugates is catalysed by Group II GRETCHEN HAGEN3 (GH3) acyl amido synthetases"

Papers: - Paper 1: Di DW, Li G, Sun L, Wu J, Wang M, Kronzucker HJ, Fang S, Chu J, Shi W. High ammonium inhibits root growth in Arabidopsis thaliana by promoting auxin conjugation rather than inhibiting auxin biosynthesis. J Plant Physiol. 2021 Apr 18;261:153415. doi: 10.1016/j.jplph.2021.153415. Epub ahead of print. PMID: 33894579. - Paper 2: Staswick PE, Serban B, Rowe M, Tiryaki I, Maldonado MT, Maldonado MC, Suza W. Characterization of an Arabidopsis enzyme family that conjugates amino acids to indole-3-acetic acid. Plant Cell. 2005 Feb;17(2):616-27. doi: 10.1105/tpc.104.026690. Epub 2005 Jan 19. PMID: 15659623; PMCID: PMC548830.

Images



This figure shows that ammonium (NH4+) induces the expression of GH3.3 in the elongation zone. This in turn reduces the amount of biologically active auxin.

AT3G58190

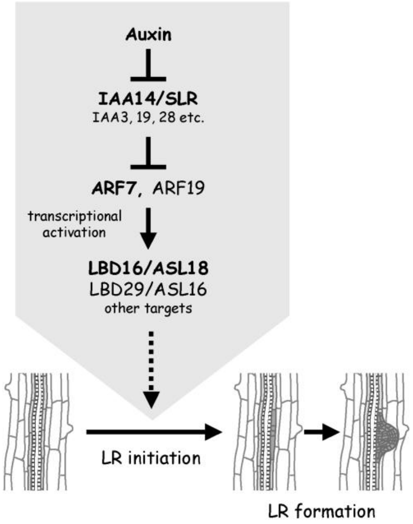
Names: * ASYMMETRIC LEAVES 2-LIKE 16 * LATERAL ORGAN BOUNDARIES DOMAIN 29 (LBD29)

Papers: - Paper 1: Zhang F, Tao W, Sun R, Wang J, Li C, et al. (2020) PRH1 mediates ARF7-LBD dependent auxin signaling to regulate lateral root development in Arabidopsis thaliana. PLOS Genetics 16(2): e1008044. https://doi.org/10.1371/journal.pgen.1008044. - Paper 2: Okushima Y, Fukaki H, Onoda M, Theologis A, Tasaka M. ARF7 and ARF19 regulate lateral root formation via direct activation of LBD/ASL genes in Arabidopsis. Plant Cell. 2007;19(1):118-130. doi:10.1105/tpc.106.047761. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1820965/

Quotes:

> "The development of lateral roots in Arabidopsis thaliana is strongly dependent on signaling directed by the AUXIN RESPONSE FACTOR7 (ARF7), which in turn activates LATERAL ORGAN BOUNDARIES DOMAIN (LBD) transcription factors (LBD16, LBD18 and LBD29)"

Images


Auxin
IAA14/SLR
IAA3, 19, 28 etc.
ARF7, ARF19
transcriptional activation
LBD16/ASL18
LBD29/ASL16
other targets
LR initiation
LR formation