R Notebook

Code

RNAseq analysis of *Arabidopsis thaliana* upon enhancement of dehydration resistance by interaction with flavonoid-attracted *Aeromonas* sp. from root microbiome

Authors:

Julio Ramírez Guerrero, Julián Román Camacho, Silvestre Ruano Rodríguez, Manuel Racero de la Rosa, Rafael Rubio Ramos.

Introduction

Background

In view of the current alarming climatic problems, the aim is to find plants that can withstand abiotic stress conditions. Flavonoids (a secondary metabolite synthesised by the plant when under stress) can be of great help in determining the plant-microbiome relationship, and to see how this improves resistance to water stress. This relationship is well studied in nodulating bacterial species, but not so much in non-nodulating bacteria such as those of the Aeromonadaceae family. Therefore, the motivation of the study is to check the behaviour of *Arabidopsis thaliana* plants in the presence/absence of *Aeromonas* sp.

Objective

To test differential expression in *Arabidopsis thaliana*, in the presence/absence of *Aeromonas* sp. and drought conditions.

Materials and Methods

Experimental Design

For our experimental design we have used as a reference the accession number GSE184872 of the GEO (Gene Expression Omnibus) website.

We have a series of points that describe the conditions of our study, according to the treatment we have applied to *Arabidopsis thaliana*, the plant that is the model organism of this study.

- Untreated (Controls):
    - GSM5599109 At_mock_1
    - GSM5599110 At_mock_2
    - GSM5599111 At_mock_3
- Drought treatment:
    - GSM5599112 At_Drought_1
    - GSM5599113 At_Drought_2
    - GSM5599114 At_Drought_3
- Aeromonas treatment:

    - GSM5599115 At_H1_1
    - GSM5599116 At_H1_2
    - GSM5599117 At_H1_3
- Drought and aeromonas treatments:

    - GSM5599118 At_H1_Drought_1
    - GSM5599119 At_H1_Drought_2
    - GSM5599120 At_H1_Drought_3

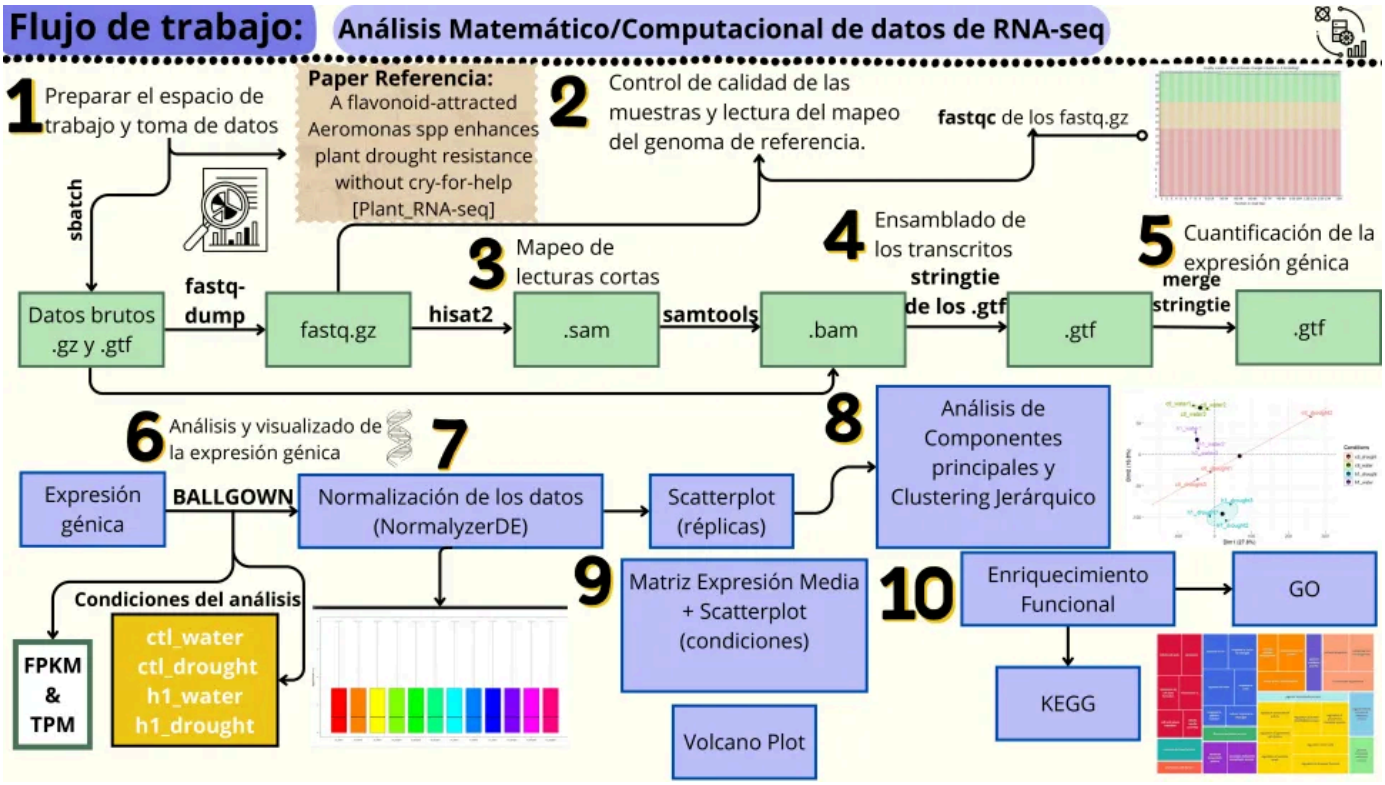
Workflow

The following is the series of steps followed to perform the mathematical/computational analysis of the RNA-seq data.

Firstly, the linux workspace is prepared, here the data is taken from the reference article and downloaded via sbatch. This will provide both .gz and .gtf files. Using a fastq-dump the SRA files are converted to FASTQ, then the fastqc quality analysis of the fastq.gz files is performed and checked to ensure that they are correctly analysed without contamination or other alteration.

Next, hisat2 will be executed on the fastq.gz and thus, the short sequence reads will be mapped against the reference genome, giving .sam files that are too heavy and inefficient when manipulating the data, so they are transferred to .bam using samtools, thus obtaining files of less weight. The indexes will also be generated. With the hisat2 alignments, the transcripts of our .bam are assembled with the stringtie tool. Once each of the samples has been assembled in the corresponding .gtf, it is necessary to compile the complete transcriptome, so the .gtf are joined with the stringtie merge tool. Once the complete transcriptome is obtained, gene expression is quantified.

The analysis is continued in the R program. With the ballgown library, the previously obtained expression data are read, the 4 conditions of the analysis are established, each one with its 3 replicates. The expression levels are extracted for both FPKM and TPM, then the data are normalised and the boxplots before and after normalisation are compared. The difference between replicates is checked with a scatterplot. With PCA and hierarchical clustering analysis the different distributions of the groups and their variability are verified. The expression matrix is used to cluster the data and check their distribution with scatterplots representing the different conditions and volcano plots to study the biological magnitude of the differences in gene expression between different experimental conditions. And finally, functional enrichment to highlight which biological processes are most represented.



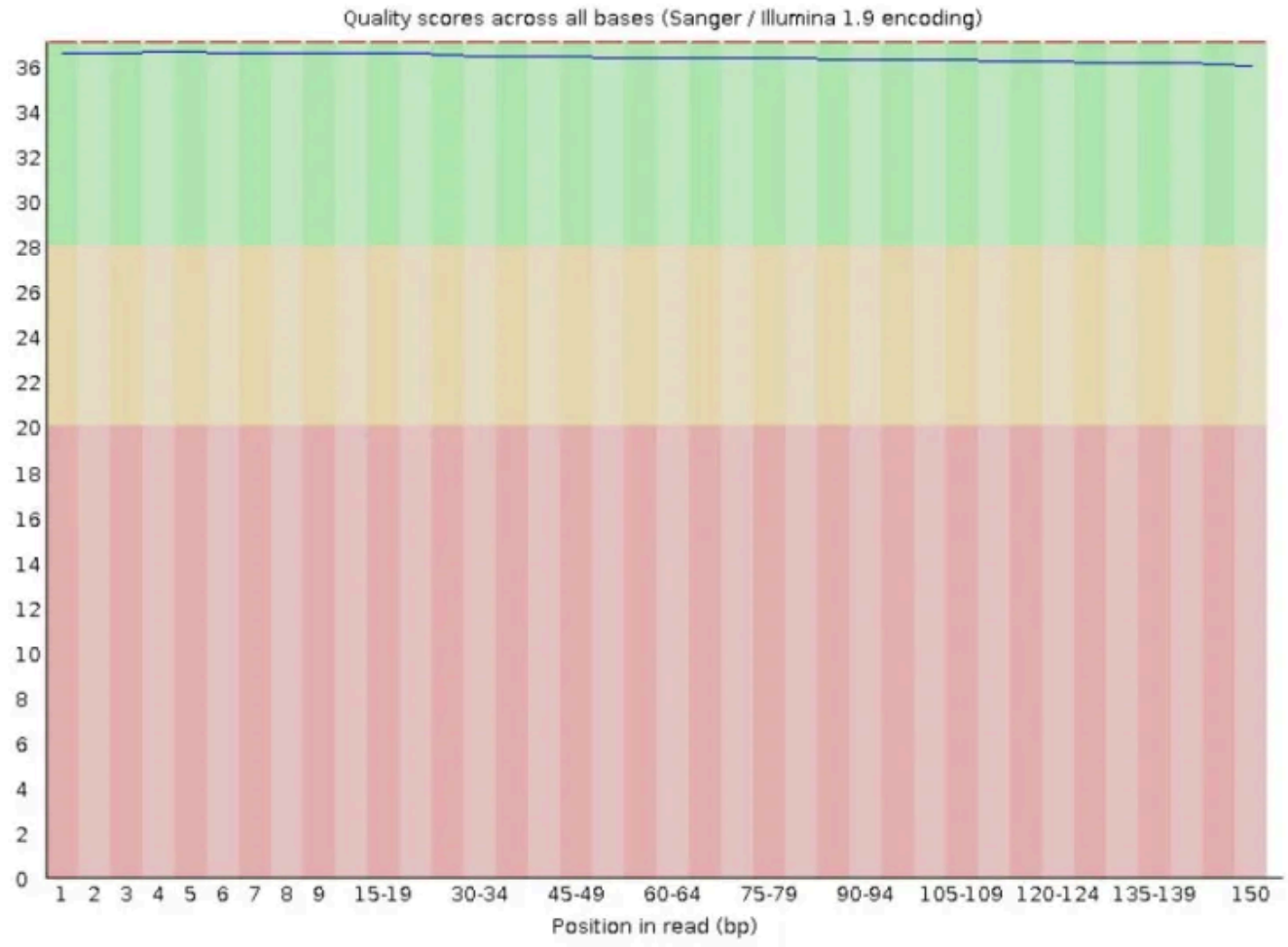
RNAseq Workflow.

Results

Quality analysis

In the graph corresponding to the quality analysis of sample 1, the results obtained are of high quality, as the boxplots of the graph remain at all times in the area coloured in green. In the rest of the samples, the results are similar, maintaining the high quality of the samples.

A pre-processing of the data is necessary to be able to compare the quality data of different samples. A normalisation algorithm was used to normalise the data for proper analysis.



Quality analysis.

Analysis of the global distribution of gene expression.

Hide

```
library(ballgown)
```

Hide

```
pheno.data <- read.csv("pheno_data.csv")
pheno.data
```

sample <chr>	aeromonas <chr>	drought <chr>
sample01	ctl	ctl
sample02	ctl	ctl
sample03	ctl	ctl
sample04	ctl	drought
sample05	ctl	drought
sample06	ctl	drought
sample07	h1	ctl
sample08	h1	ctl

sample <chr>	aeromonas <chr>	drought <chr>
sample09	h1	ctl
sample10	h1	drought

1-10 of 12 rows Previous **1** 2 Next

Hide

```
bg.data <- ballgown(dataDir = ".", samplePattern = "sample", pData=pheno.data)
bg.data
```

```
ballgown instance with 54013 transcripts and 12 samples
```

Hide

```
sampleNames(bg.data)
```

```
 [1] "sample01" "sample02" "sample03" "sample04" "sample05" "sample06" "sample07"
 [8] "sample08" "sample09" "sample10" "sample11" "sample12"
```

Hide

```
gene.expression <- gexpr(bg.data)
head(gene.expression)
```

```
          FPKM.sample01 FPKM.sample02 FPKM.sample03 FPKM.sample04 FPKM.sample05
AT1G01010      2.168046      2.241514      2.348751      2.605879      1.847383
AT1G01020      4.872953      5.080886      4.735107      4.310888      4.732878
AT1G01030      3.254484      3.117943      2.696376      2.371129      1.301995
AT1G01040      7.339939      7.840886      7.172608      7.074731      7.553657
AT1G01046      2.577555      2.459286      2.684554      1.737085      3.847410
AT1G01050     42.112597     41.229954     43.094138     35.266402     23.194304
          FPKM.sample06 FPKM.sample07 FPKM.sample08 FPKM.sample09 FPKM.sample10
AT1G01010      2.187437      2.680110      2.111505      2.974339      2.597551
AT1G01020      4.527134      5.144373      5.081418      5.510741      5.554990
AT1G01030      2.596881      3.168968      2.914683      3.625461      2.384258
AT1G01040      6.995763      7.963019      8.068400      8.367129      7.981829
AT1G01046      2.044677      3.689383      3.251185      4.423835      3.087394
AT1G01050     36.138577     42.320342     40.104723     37.942606     32.413188
          FPKM.sample11 FPKM.sample12
AT1G01010      2.686215      3.157594
AT1G01020      4.937633      5.254439
AT1G01030      1.866953      2.358055
AT1G01040      7.311965      7.588940
AT1G01046      1.946588      3.255827
AT1G01050     34.771758     34.457155
```

Hide

```
dim(gene.expression)
```

```
[1] 32833    12
```

Hide

```
gene.names <- rownames(gene.expression)
```

Hide

```
colnames(gene.expression) <- c("ctl_water1","ctl_water2","ctl_water3","ctl_drought1","ctl_dro
ught2","ctl_drought3", "h1_water1","h1_water2","h1_water3","h1_drought1","h1_drought2","h1_dr
ought3")
```

Hide

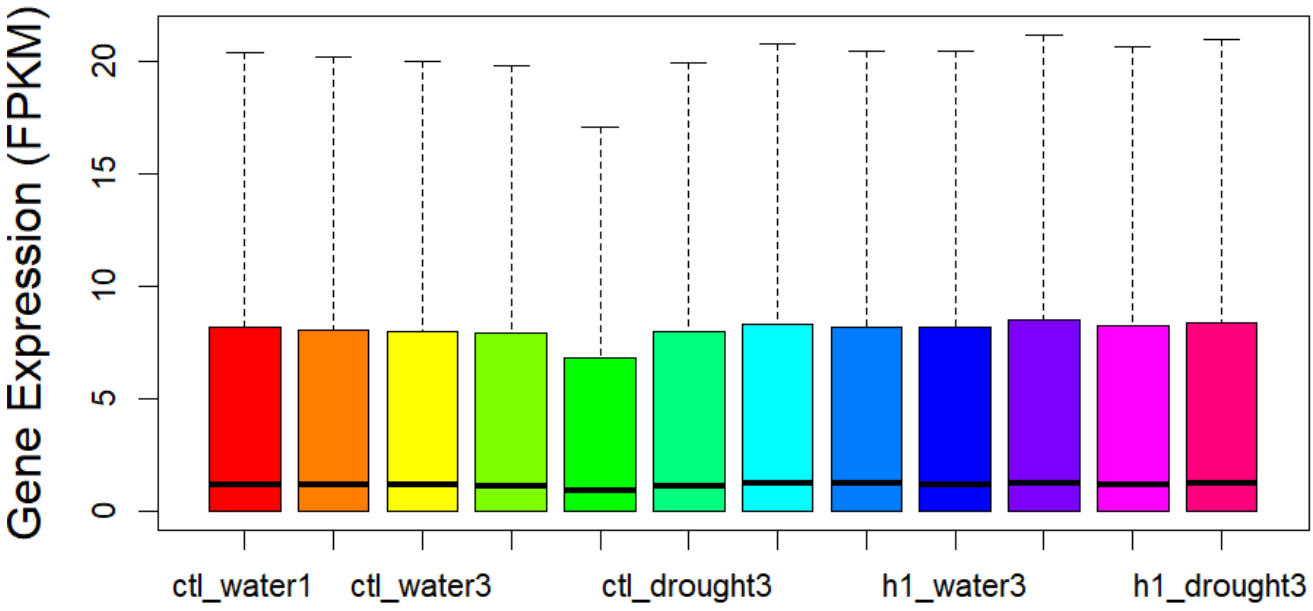
```
gene.expression.1 <- gene.expression + 1
```

Hide

```
write.table(x = gene.expression.1,file = "tarea1_gene_expression.tsv",
            quote = F,row.names = F,
            sep = "\t")
```

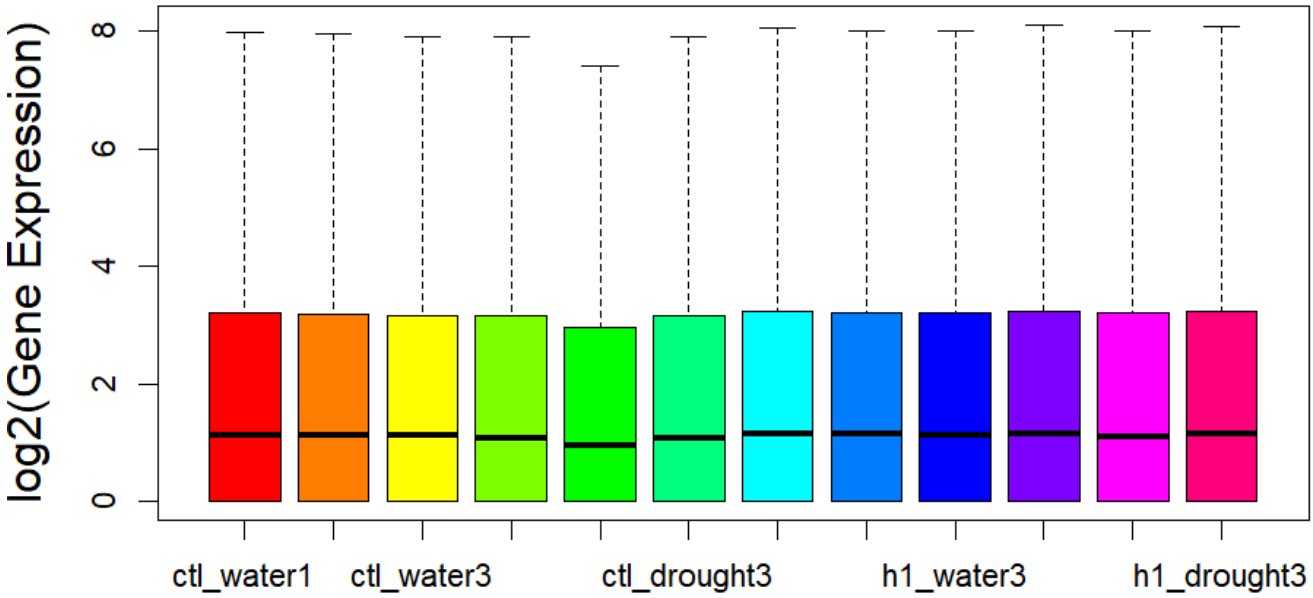
Hide

```
boxplot(gene.expression, outline=F,col=rainbow(12),ylab="Gene Expression (FPKM)",
        cex.lab=1.5)
```



Hide

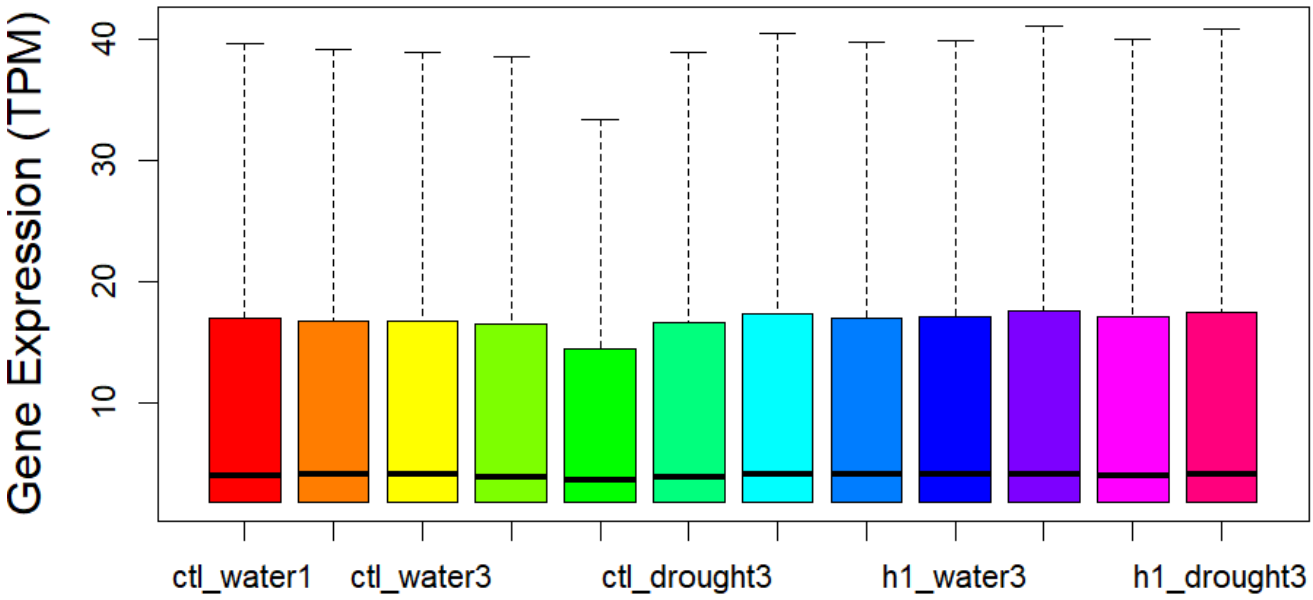
```
boxplot(log2(gene.expression.1), outline=F,col=rainbow(12),
        ylab="log2(Gene Expression)",
        cex.lab=1.5)
```



The analysis of the expression of the data by TPM is carried out to check that the results are the same as those obtained by FPKM:

Hide

```
tpm <- gene.expression.1 / colSums(gene.expression.1) * 1e6

## Save gene expression data in TPM format
write.table(x = tpm, file = "tarea1_gene_expression_tpm.TSV", quote = F, row.names = F, sep =
"\t")

## Representation of the global distribution of gene expression in TPM:
boxplot(tpm, outline = F, col = rainbow(12), ylab = "Gene Expression (TPM)", cex.lab = 1.5)
```



Data normalisation is necessary because errors can occur during sample preparation that alter the actual data that should be obtained. In this way, we correct the bias produced during sample preparation and facilitate the comparison of results between different samples. In addition, data normalisation facilitates the detection of significant differences in gene expression, and prevents misinterpretations because normalisation eliminates apparent differences that could be the result of technical rather than biological variations.

Hide

```
library(NormalyzerDE)
```

Hide

```
design <- data.frame(sample=colnames(gene.expression),
                     group=c(rep("ctl_water",3),rep("ctl_drought",3),rep("h1_water",3),rep("h
1_drought",3)))
```

Hide

```
write.table(x = design,file = "normalyzer_design.tsv",quote = F,row.names = F,
            sep = "\t")
```

Hide

```
normalyzer(jobName = "Tarea1",designPath = "normalyzer_design.tsv",
           dataPath = "tarea1_gene_expression.tsv",outputDir = ".")
```

Hide

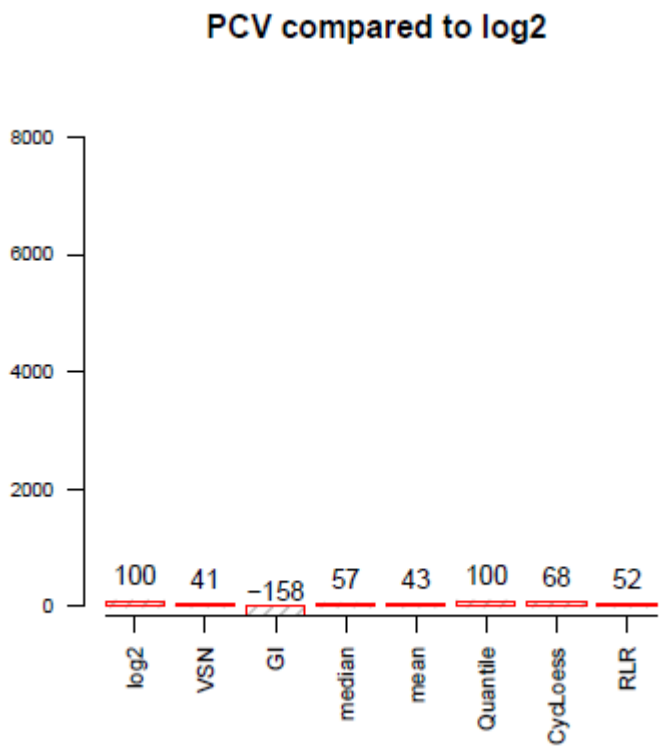
```
normalized.gene.expression <- read.table(file="Tarea1/Quantile-normalized.txt", header=T)
head(normalized.gene.expression)
```

	ctl_water1 <dbl>	ctl_water2 <dbl>	ctl_water3 <dbl>	ctl_drought1 <dbl>	ctl_drought2 <dbl>	ctl_drought3 <dbl>	h1_water1 <dbl>
1	1.645358	1.686830	1.749690	1.894330	1.697981	1.714079	1.838767
2	2.541439	2.592654	2.533577	2.437462	2.734918	2.487989	2.586446
3	2.076550	2.031644	1.892815	1.791262	1.369782	1.881226	2.010443
4	3.041992	3.141029	3.038819	3.037966	3.304541	3.011668	3.126146
5	1.821404	1.782762	1.886751	1.491930	2.492774	1.645044	2.191051
6	5.421459	5.407228	5.457798	5.191084	4.742492	5.200610	5.418182

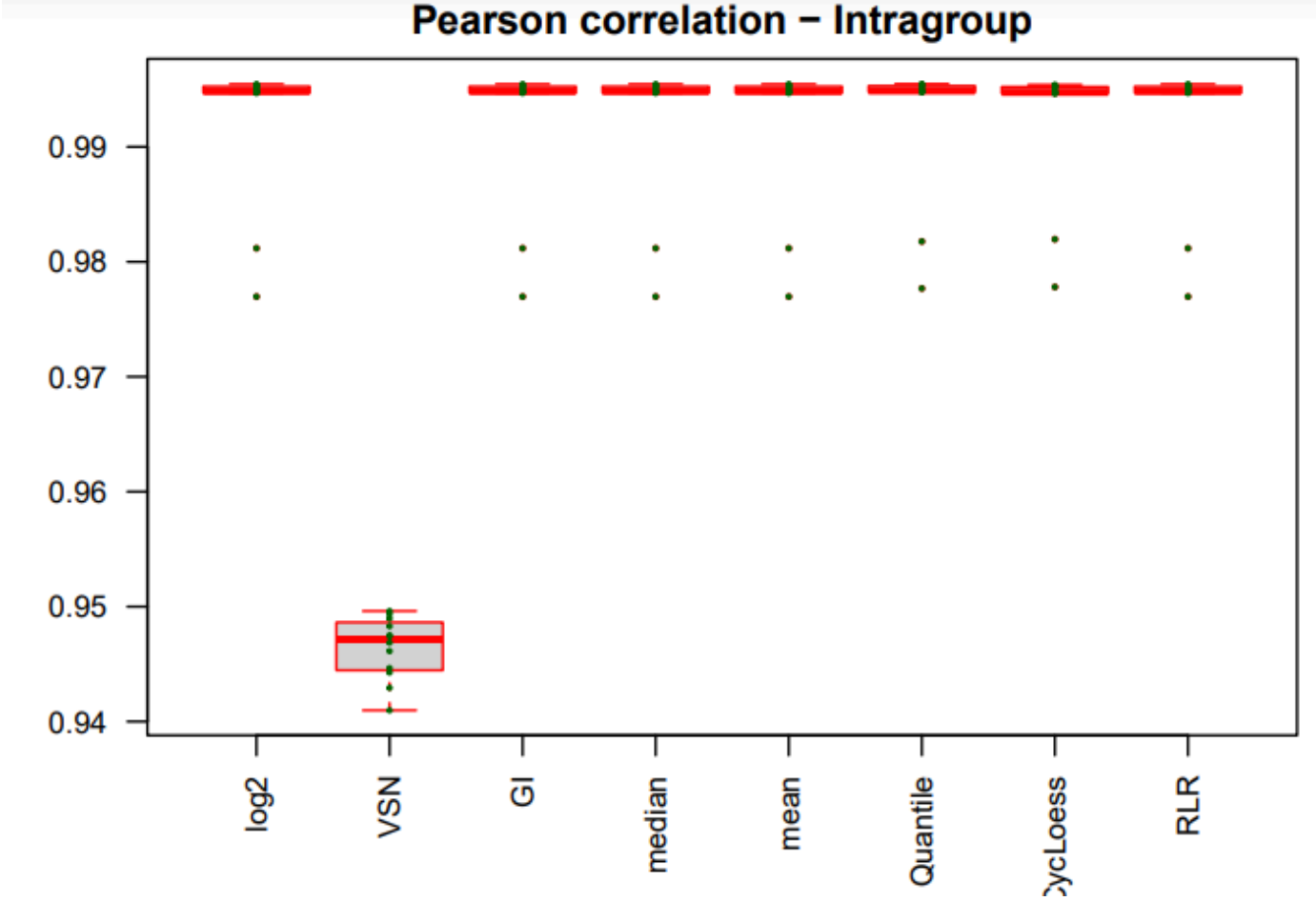
6 rows | 1-9 of 12 columns

Hide

```
rownames(normalized.gene.expression) <- gene.names
```



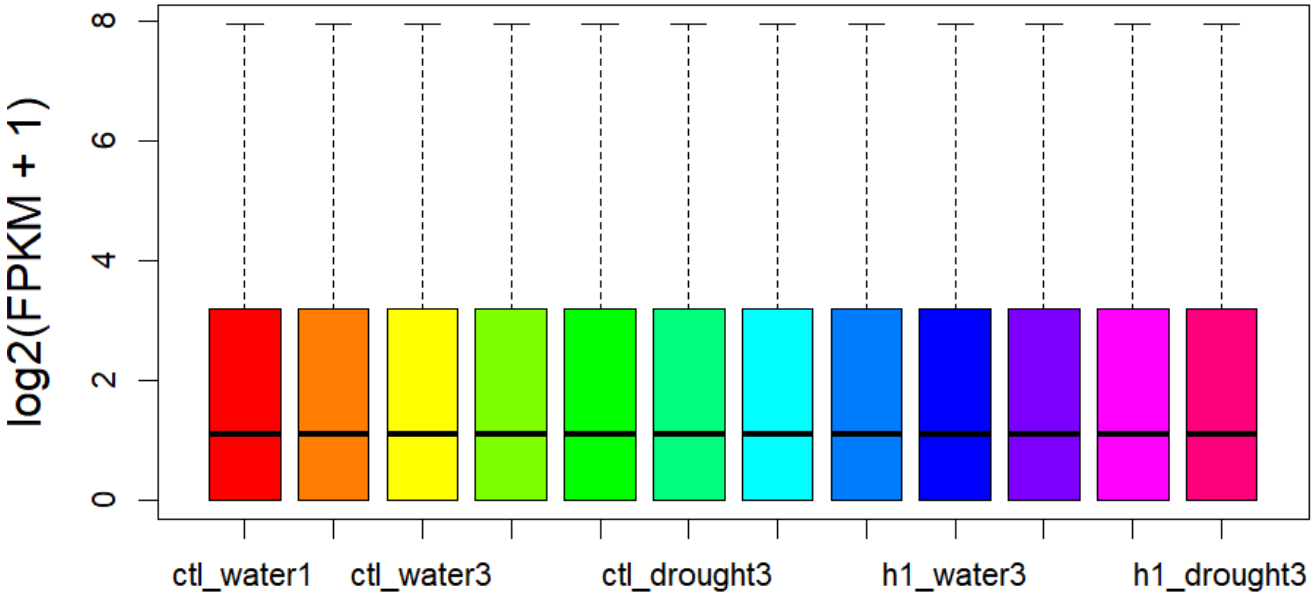
Normalisation analysis of the coefficient of variation within groups.
According to this assessment VSN is the best standardisation technique because it would reduce noise the most.



Pearson correlation.
However, according to Pearson's correlation coefficient VSN is not a suitable technique. Therefore, we took Quantil as a normalisation technique.

Hide

```
boxplot(normalized.gene.expression, outline=F,col=rainbow(12),
        ylab="log2(FPKM + 1)",cex.lab=1.5)
```



The diagrams are aligned, indicating that they are on the same scale and are comparable, thus confirming that data normalisation has been performed.

Scatterplots to compare replicas

Hide

```
par(mfrow=c(3,4))

plot(x = normalized.gene.expression[,"ctl_water1"],
     y = normalized.gene.expression[,"ctl_water2"],
     pch=19,col="black",xlab="ctl_water1",ylab="ctl_water2",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
      "cor = ",
      round(100*cor(normalized.gene.expression[,"ctl_water1"],
                    normalized.gene.expression[,"ctl_water2"]),
            digits = 2),
      "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"ctl_water1"],
     y = normalized.gene.expression[,"ctl_water3"],
     pch=19,col="black",xlab="ctl_water1",ylab="ctl_water3",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"ctl_water1"],
                     normalized.gene.expression[,"ctl_water3"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"ctl_water2"],
     y = normalized.gene.expression[,"ctl_water3"],
     pch=19,col="black",xlab="ctl_water2",ylab="ctl_water3",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"ctl_water2"],
                     normalized.gene.expression[,"ctl_water3"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"ctl_drought1"],
     y = normalized.gene.expression[,"ctl_drought2"],
     pch=19,col="black",xlab="ctl_drought1",ylab="ctl_drought2",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"ctl_drought1"],
                     normalized.gene.expression[,"ctl_drought2"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"ctl_drought1"],
     y = normalized.gene.expression[,"ctl_drought3"],
     pch=19,col="black",xlab="ctl_drought1",ylab="ctl_drought3",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"ctl_drought1"],
                     normalized.gene.expression[,"ctl_drought3"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"ctl_drought2"],
     y = normalized.gene.expression[,"ctl_drought3"],
     pch=19,col="black",xlab="ctl_drought2",ylab="ctl_drought3",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"ctl_drought2"],
                     normalized.gene.expression[,"ctl_drought3"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"h1_water1"],
     y = normalized.gene.expression[,"h1_water2"],
     pch=19,col="black",xlab="h1_water1",ylab="h1_water2",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"h1_water1"],
                     normalized.gene.expression[,"h1_water2"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"h1_water1"],
     y = normalized.gene.expression[,"h1_water3"],
     pch=19,col="black",xlab="h1_water1",ylab="h1_water3",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"h1_water1"],
                     normalized.gene.expression[,"h1_water3"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"h1_water2"],
     y = normalized.gene.expression[,"h1_water3"],
     pch=19,col="black",xlab="h1_water2",ylab="h1_water3",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"h1_water2"],
                     normalized.gene.expression[,"h1_water3"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

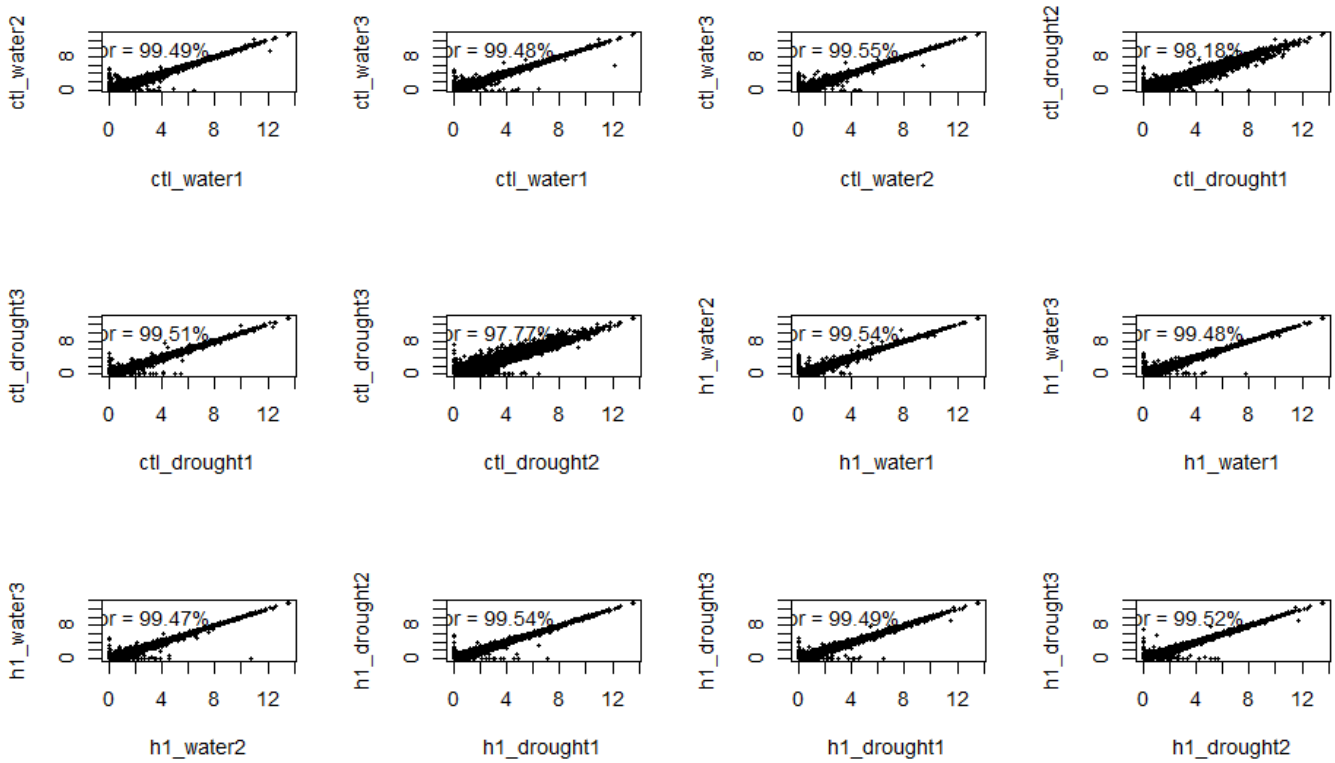
```
plot(x = normalized.gene.expression[,"h1_drought1"],
     y = normalized.gene.expression[,"h1_drought2"],
     pch=19,col="black",xlab="h1_drought1",ylab="h1_drought2",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"h1_drought1"],
                     normalized.gene.expression[,"h1_drought2"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"h1_drought1"],
     y = normalized.gene.expression[,"h1_drought3"],
     pch=19,col="black",xlab="h1_drought1",ylab="h1_drought3",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"h1_drought1"],
                     normalized.gene.expression[,"h1_drought3"]),
             digits = 2),
       "%"), collapse=""))
```

Hide

```
plot(x = normalized.gene.expression[,"h1_drought2"],
     y = normalized.gene.expression[,"h1_drought3"],
     pch=19,col="black",xlab="h1_drought2",ylab="h1_drought3",cex=0.5)
text(x=3,y=10,
     labels = paste(c(
       "cor = ",
       round(100*cor(normalized.gene.expression[,"h1_drought2"],
                     normalized.gene.expression[,"h1_drought3"]),
             digits = 2),
       "%"), collapse=""))
```



In the scatter plots of the similarity of replicates the results have indicated that the correlation between the replicates is very high, mostly around 99%, thus indicating a good relationship between the replicates.

Then, we can see in most of the graphs very low dispersion between the points thus forming in most cases a figure similar to a straight line.

Finally, it should also be noted that the replica ctl_drought2 has a greater dispersion of data and therefore greater variability, given that at the time of the study it was observed that ctl_drought2 was more distant from the rest of the data.

Principal Component Analysis and Hierarchical Clustering

Hide

```
library(FactoMineR)
library(factoextra)
```

Hide

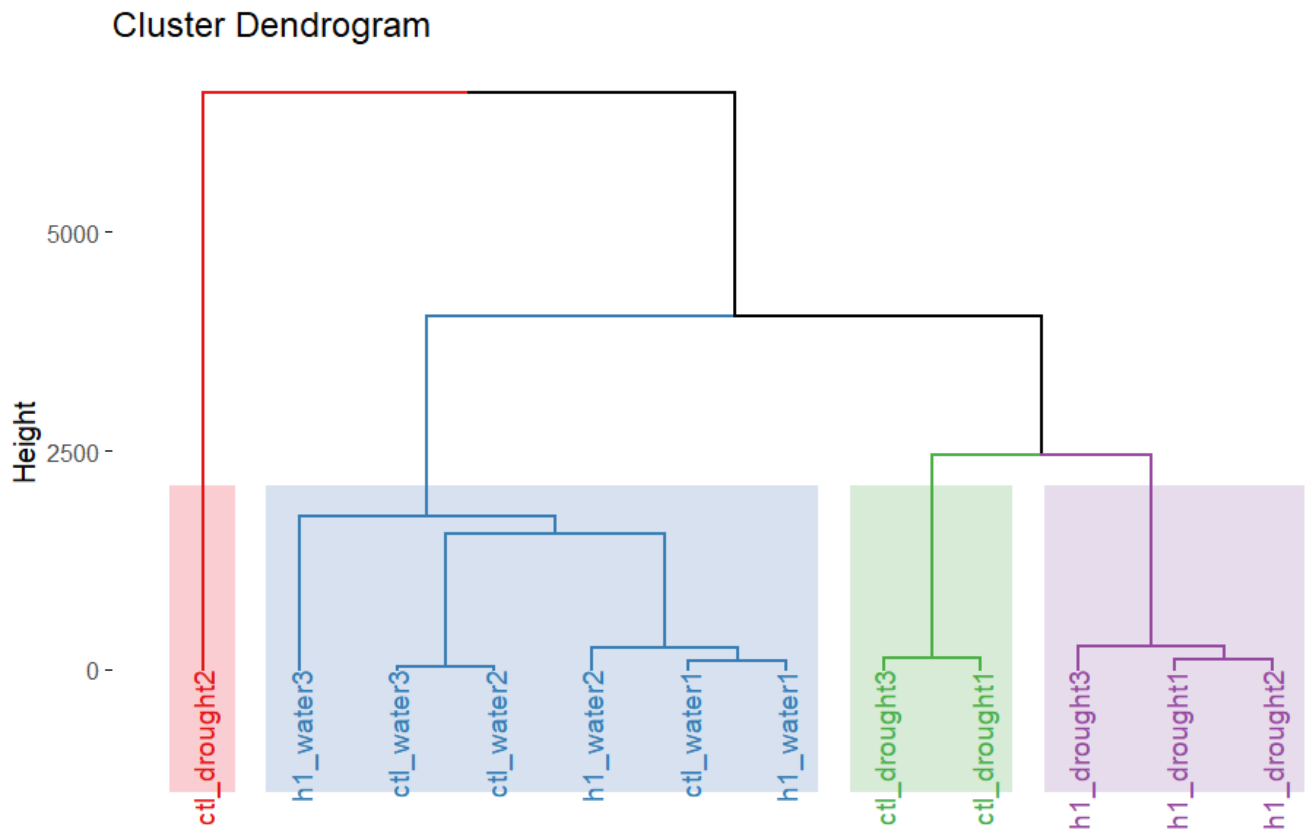
```
pca.gene.expression <- data.frame(colnames(normalized.gene.expression),
                                  t(normalized.gene.expression))
colnames(pca.gene.expression)[1] <- "Sample"
head(pca.gene.expression)
```

	Sample <chr>	AT1G01… <dbl>	AT1G01… <dbl>	AT1G01… <dbl>	AT1G01… <dbl>	AT1G01… <dbl>	AT1G01 <dl
ctl_water1	ctl_water1	1.645358	2.541439	2.076550	3.041992	1.821404	5.4214
ctl_water2	ctl_water2	1.686830	2.592654	2.031644	3.141029	1.782762	5.4072
ctl_water3	ctl_water3	1.749690	2.533577	1.892815	3.038819	1.886751	5.4577
ctl_drought1	ctl_drought1	1.894330	2.437462	1.791262	3.037966	1.491930	5.1910
ctl_drought2	ctl_drought2	1.697981	2.734918	1.369782	3.304541	2.492774	4.7424
ctl_drought3	ctl_drought3	1.714079	2.487989	1.881226	3.011668	1.645044	5.2006

6 rows | 1-9 of 32834 columns

Hide

```
res.pca <- PCA(pca.gene.expression, graph = FALSE,scale.unit = TRUE,quali.sup = 1 )
res.hcpc <- HCPC(res.pca, graph=FALSE,nb.clust = 4)
fviz_dend(res.hcpc,k=,
          cex = 0.75,
          palette = "Set1",
          rect = TRUE, rect_fill = TRUE,
          rect_border = "Set1",
          type="rectangle",
          labels_track_height = 1400
)
```

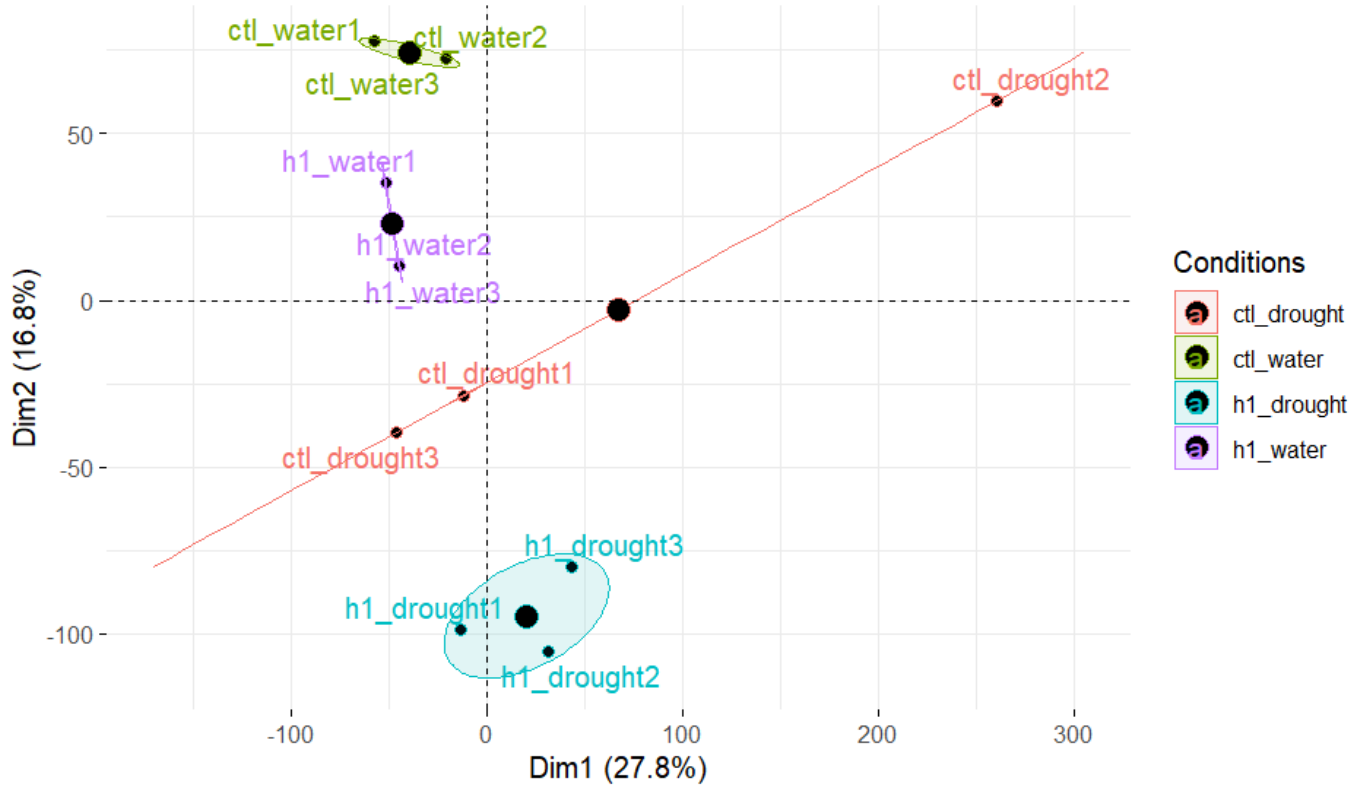


In the dendogram, we should initially have each condition grouped with its three replicas, however, we see that ctl_drought2 is isolated in a cluster because its data is moving away from its condition group. So we **could** remove that replica as it alters the stability of the data.

In addition, a disturbance is also visible in the cluster concerning “water”, where h1_water3 is deviated from its condition group h1_water. The same is true for ctl_water1 which is closer to the h1_water cluster. This is due to the variability of data obtained in these conditions.

Hide

```
fviz_pca_ind(res.pca,
             col.ind = c("ctl_water", "ctl_water", "ctl_water", "ctl_drought", "ctl_drought",
"ctl_drought", "h1_water", "h1_water", "h1_water", "h1_drought", "h1_drought", "h1_drought"),
             pointsize = 2,
             pointshape = 21,
             fill = "black",
             repel = TRUE,
             addEllipses = TRUE,
             ellipse.type = "confidence",
             legend.title = "Conditions",
             title = "",
             show.legend = TRUE,
             show.guide = TRUE
)
```



In the confidence ellipses plot, the ellipses indicate the clustering of the data, in this case all replicates are clustered in their respective conditions. Both ctl_water, h1_drought and h1_water are correctly clustered. The replicates in these groups are not very far from the midpoint indicating little variability in the data.

Again, we see that the ctl_drought2 condition is quite dispersed with respect to the replicates of the same condition, forming an ellipse that looks more like a straight line.

Unlike the dendogram, we see in this plot that the replicates are well clustered.

Differential Gene Expression Analysis

Comparison of the generated transcriptome with the reference transcriptome

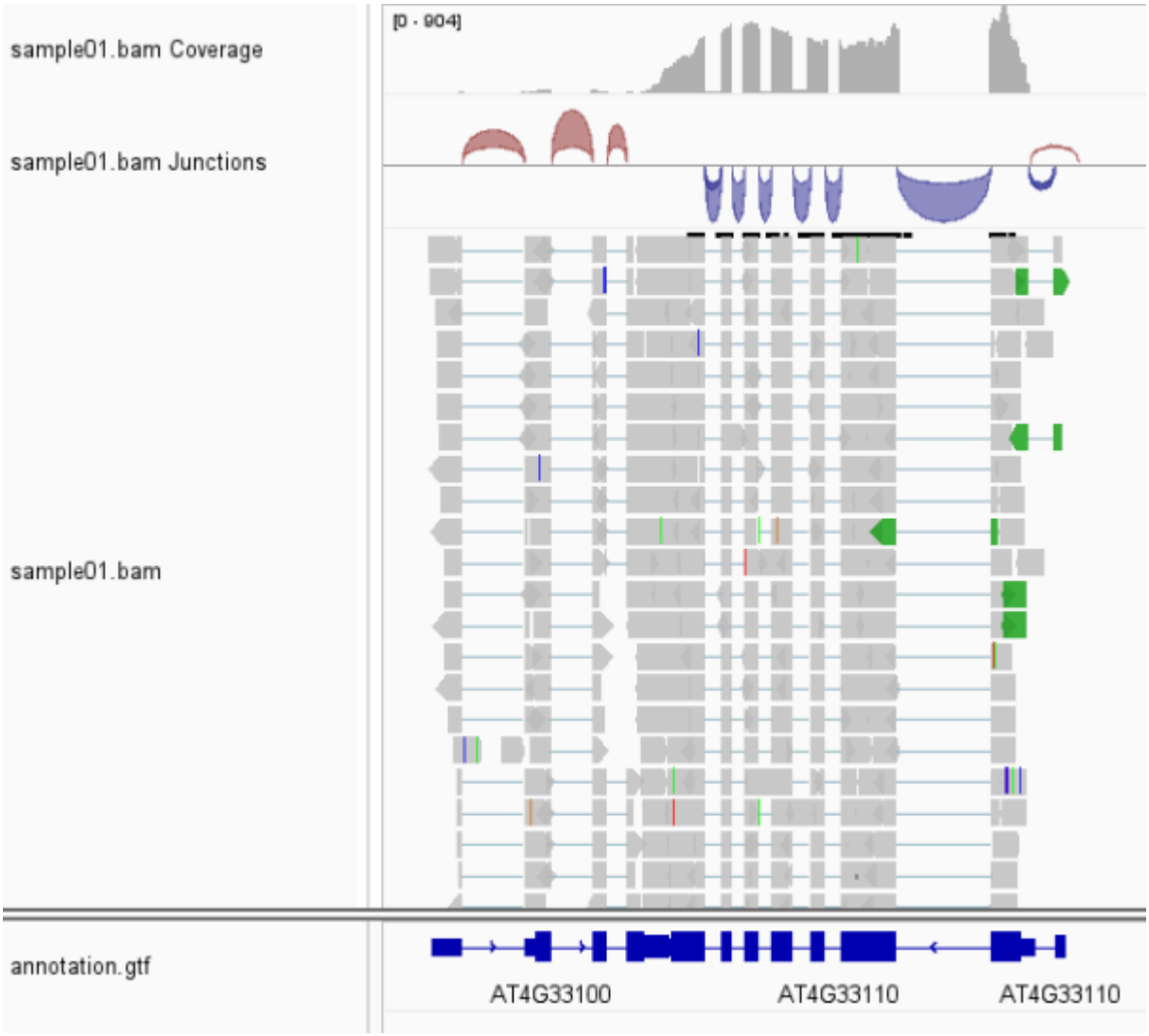
Before proceeding further with the data analysis, it is useful to confirm that the annotation used (i.e. downloaded from the database) is suitable for the study. To do this, we can compare the transcriptome in the database with the transcripts that have been detected in total in the study (in all samples).

```
# gffcompare v0.10.1 | Command line was:
#gffcompare -r ../annotation/annotation.gtf -G -o merged stringtie_merged.gtf
#

#= Summary for dataset: stringtie_merged.gtf
#     Query mRNAs :   53296 in   32048 loci  (42647 multi-exon transcripts)
#           (10567 multi-transcript loci, ~1.7 transcripts per locus)
# Reference mRNAs :   53518 in   32046 loci  (42816 multi-exon)
# Super-loci w/ reference transcripts:    31942
#-----------------| Sensitivity | Precision  |
        Base level:    100.0     |    99.9    |
        Exon level:     97.9     |    99.1    |
      Intron level:    100.0     |   100.0    |
Intron chain level:     99.6     |   100.0    |
  Transcript level:     99.2     |    99.6    |
       Locus level:     99.8     |    99.8    |

     Matching intron chains:   42644
       Matching transcripts:   53096
              Matching loci:   31992

          Missed exons:       0/192405  (  0.0%)
           Novel exons:       0/185113  (  0.0%)
        Missed introns:       1/132525  (  0.0%)
         Novel introns:       0/132525  (  0.0%)
           Missed loci:       0/32046   (  0.0%)
            Novel loci:       0/32048   (  0.0%)

 Total union super-loci across all input datasets: 32046
53296 out of 53296 consensus transcripts written in merged.annotated.gtf (0 discarded as redundant)
```

Transcriptome comparison.
As we can see, the transcriptome used contains 100% of the transcripts observed.

Mapping visualisation

We can see the result of the mapping in IGV. The following image shows some of the reads that have mapped to the AT4G33100 gene.



IGV screenshot.
Each line represents a read, so that the grey rectangles are genome sequences found in the read and the lines between the rectangles correspond to fragments that are present in the genome, but not in the reads. As can be seen in the annotation, these areas correspond to the introns of the gene (blue lines).

Differential Gene Expression Analysis

First, we will use scatterplots to visualise the effect of the different treatments performed in the study.

Hide

```
# We calculate the average expression of each gene in each condition
ctl_water <- (normalized.gene.expression[,"ctl_water1"] + normalized.gene.expression[,"ctl_wa
ter2"]+ normalized.gene.expression[,"ctl_water3"])/3

ctl_drought <- (normalized.gene.expression[,"ctl_drought1"] + normalized.gene.expression[,"ct
l_drought2"]+ normalized.gene.expression[,"ctl_drought3"])/3

h1_water <- (normalized.gene.expression[,"h1_water1"] + normalized.gene.expression[,"h1_water
2"]+ normalized.gene.expression[,"h1_water3"])/3

h1_drought <- (normalized.gene.expression[,"h1_drought1"] + normalized.gene.expression[,"h1_d
rought2"]+ normalized.gene.expression[,"h1_drought3"])/3

mean.expression <- matrix(c(ctl_water,ctl_drought, h1_water,h1_drought),ncol=4)
colnames(mean.expression) <- c("ctl_water","ctl_drought","h1_water","h1_drought")
rownames(mean.expression) <- rownames(normalized.gene.expression)
head(mean.expression)
```

```
          ctl_water ctl_drought h1_water h1_drought
AT1G01010  1.693959    1.768797 1.799234   1.874969
AT1G01020  2.555890    2.553456 2.613927   2.590527
AT1G01030  2.000336    1.680757 2.046434   1.635062
AT1G01040  3.073947    3.118058 3.164996   3.057699
AT1G01046  1.830305    1.876582 2.226698   1.848300
AT1G01050  5.428828    5.044729 5.355597   5.090008
```

Hide

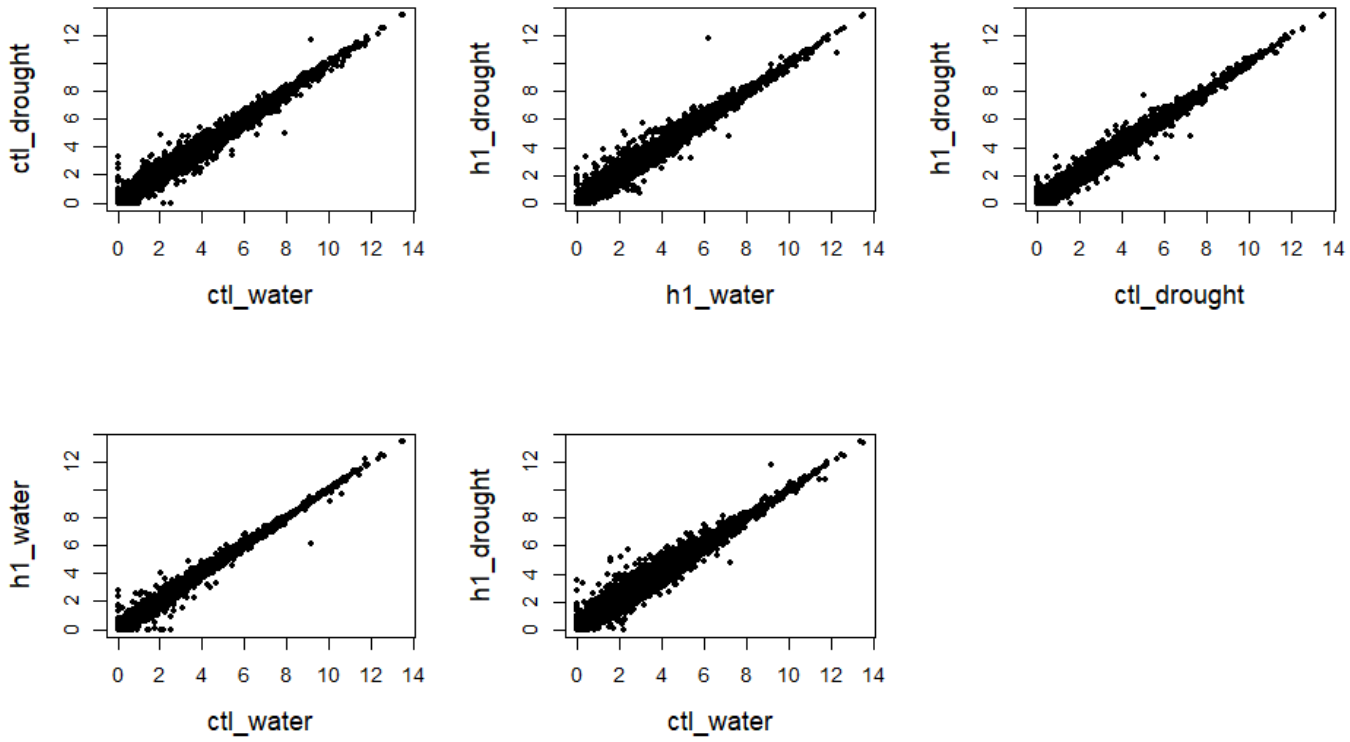
```
par(mfrow=c(2,3))

plot(ctl_water,ctl_drought,pch=19,cex=0.7,xlab="ctl_water",
     ylab="ctl_drought",cex.lab=1.25,col="black")

plot(h1_water,h1_drought,pch=19,cex=0.7,xlab="h1_water",
     ylab="h1_drought",cex.lab=1.25,col="black")
```

Hide

```


plot(ctl_drought,h1_drought,pch=19,cex=0.7,xlab="ctl_drought",
     ylab="h1_drought",cex.lab=1.25,col="black")

plot(ctl_water,h1_water,pch=19,cex=0.7,xlab="ctl_water",
     ylab="h1_water",cex.lab=1.25,col="black")
```

Hide

```


plot(ctl_water,h1_drought,pch=19,cex=0.7,xlab="ctl_water",
     ylab="h1_drought",cex.lab=1.25,col="black")
```



We can see that the transcriptome is not strongly altered by the drought and inoculation treatments with *Aeromonas* sp., so we will have to be very permissive when determining which genes are differentially expressed. In this case, we consider as differentially expressed genes (DEGs) those genes where the observed change in gene expression is substantial and significant. Thresholds of log2(1.5) and q-value<0.01 will be used.

Hide

```
library(limma)
# We define the experimental design and determine the comparisons to be made.
experimental.design <- model.matrix(~ -1+factor(c(1,1,1,2,2,2,3,3,3,4,4,4)))
colnames(experimental.design) <- c("ctl_water","ctl_drought","h1_water","h1_drought")
linear.fit <- lmFit(normalized.gene.expression, experimental.design)
contrast.matrix <- makeContrasts(ctl_drought-ctl_water, h1_drought-h1_water, h1_drought-ctl_d
rought,h1_water-ctl_water, h1_drought-ctl_water,levels=c("ctl_water","ctl_drought","h1_wate
r","h1_drought"))
contrast.linear.fit <- contrasts.fit(linear.fit, contrast.matrix)
contrast.results <- eBayes(contrast.linear.fit)
```

Comparison 1. CTL_Drought vs CTL_Water

The aim of this first comparison is to identify the genes involved in drought response.

Hide

```
ctl_drought.ctl_water <- topTable(contrast.results, number=32833,coef=1,sort.by="logFC")
log.fold.change <- ctl_drought.ctl_water$logFC
q.value <- ctl_drought.ctl_water$adj.P.Val
genes.ids <- rownames(ctl_drought.ctl_water)
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c1 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
repressed.c1 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
length(activated.c1)
```

```
[1] 112
```

Hide

```
length(repressed.c1)
```

```
[1] 36
```

Hide

```
log.q.val <- -log10(q.value)
plot(log.fold.change,log.q.val,pch=19,col="black",cex=0.8,
     xlim=c(-6,6),ylim = c(0,4),
     xlab="log2(Fold-chage)",ylab="-log10(q-value)",cex.lab=1.5,
     main="CTL_drought vs CTL_water")

points(x = log.fold.change[activated.c1],
       y = log.q.val[activated.c1],col="red",cex=0.8,pch=19)
```

Hide

```
points(x = log.fold.change[repressed.c1],
       y = log.q.val[repressed.c1],col="blue",cex=0.8,pch=19)
arrows(log.fold.change["AT5G02020"],log.q.val["AT5G02020"],2.5,1.2,length=0)
```

Hide

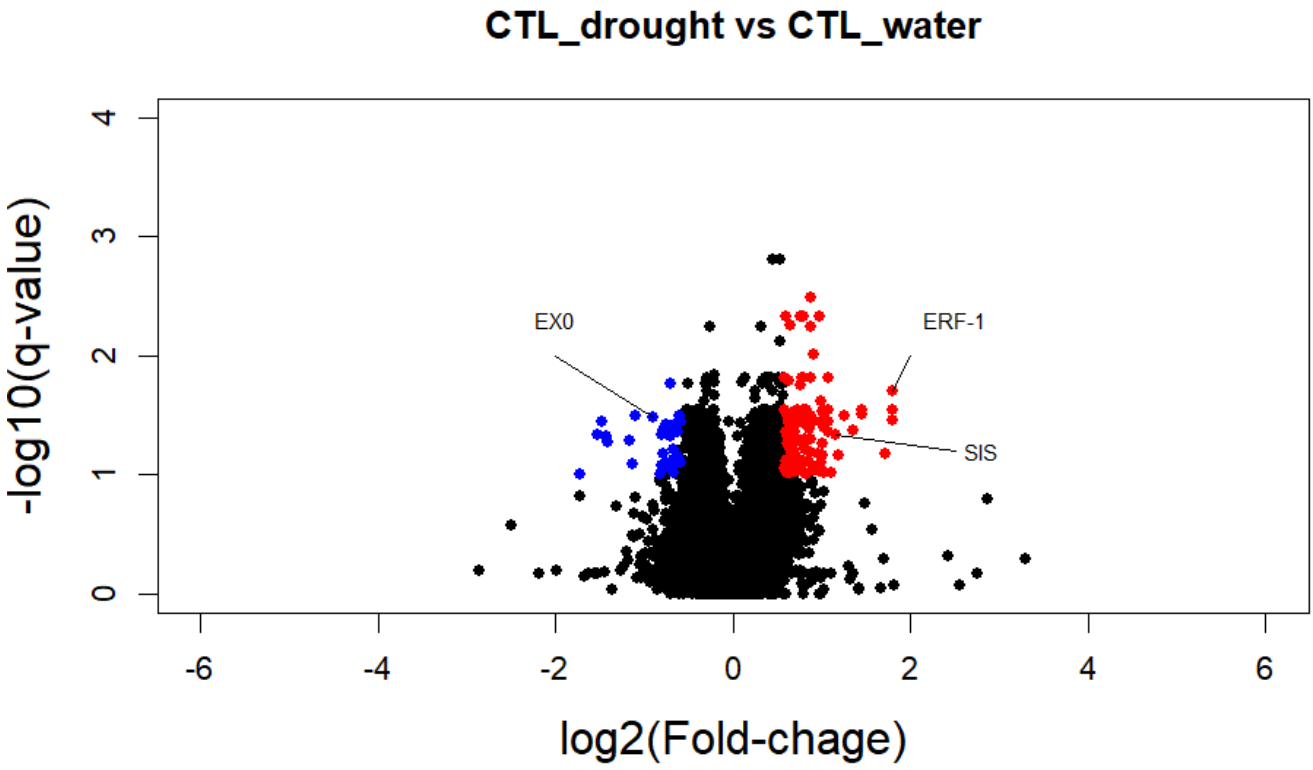
```
text(2.8,1.2,"SIS",cex=0.7)
arrows(log.fold.change["AT4G17500"],log.q.val["AT4G17500"],2,2,length=0)
```

Hide

```
text(2.5,2.3,"ERF-1",cex=0.7)
arrows(log.fold.change["AT4G08950"],log.q.val["AT4G08950"],-2,2,length=0)
```

Hide

```
text(-2,2.3,"EX0",cex=0.7)
```



Hide

```
write.table(activated.c1,file="activated_ctl_drought_ctl_water.txt",row.names = FALSE,quote=F
ALSE,col.names = FALSE)
write.table(repressed.c1,file="repressed_ctl_drought_ctl_water.txt",row.names = FALSE,quote=F
ALSE,col.names = FALSE)
```

Activated genes comparison 1 (activated_ctl_drought_ctl_water.txt) Repressed genes comparison 1 (repressed_ctl_drought_ctl_water.txt)

In this comparison, 112 activated genes and 36 repressed genes have been detected. Among the activated genes, we found SIS (Salt Induced Serine Rich), a protein known to be involved in salinity tolerance (a condition closely related to low water availability).

Comparison 2. H1_drought vs H1_water

The objective in this case is to determine which genes are involved in the drought response when plants have been inoculated with *Aeromonas*.

Hide

```
h1_drought.h1_water <- topTable(contrast.results, number=32833,coef=2,sort.by="logFC")
log.fold.change <- h1_drought.h1_water$logFC
q.value <- h1_drought.h1_water$adj.P.Val
genes.ids <- rownames(h1_drought.h1_water)
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c2 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
repressed.c2 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
length(activated.c2)
```

```
[1] 282
```

Hide

```
length(repressed.c2)
```

```
[1] 45
```

Hide

```
log.q.val <- -log10(q.value)
plot(log.fold.change,log.q.val,pch=19,col="black",cex=0.8,
     xlim=c(-6,6),ylim = c(0,4),
     xlab="log2(Fold-chage)",ylab="-log10(q-value)",cex.lab=1.5,
     main="H1_drought vs H1_water")

points(x = log.fold.change[activated.c2],
       y = log.q.val[activated.c2],col="red",cex=0.8,pch=19)
```

Hide

```
points(x = log.fold.change[repressed.c2],
       y = log.q.val[repressed.c2],col="blue",cex=0.8,pch=19)
arrows(log.fold.change["AT3G23550"],log.q.val["AT3G23550"],2.5,2.5,length=0)
```

Hide

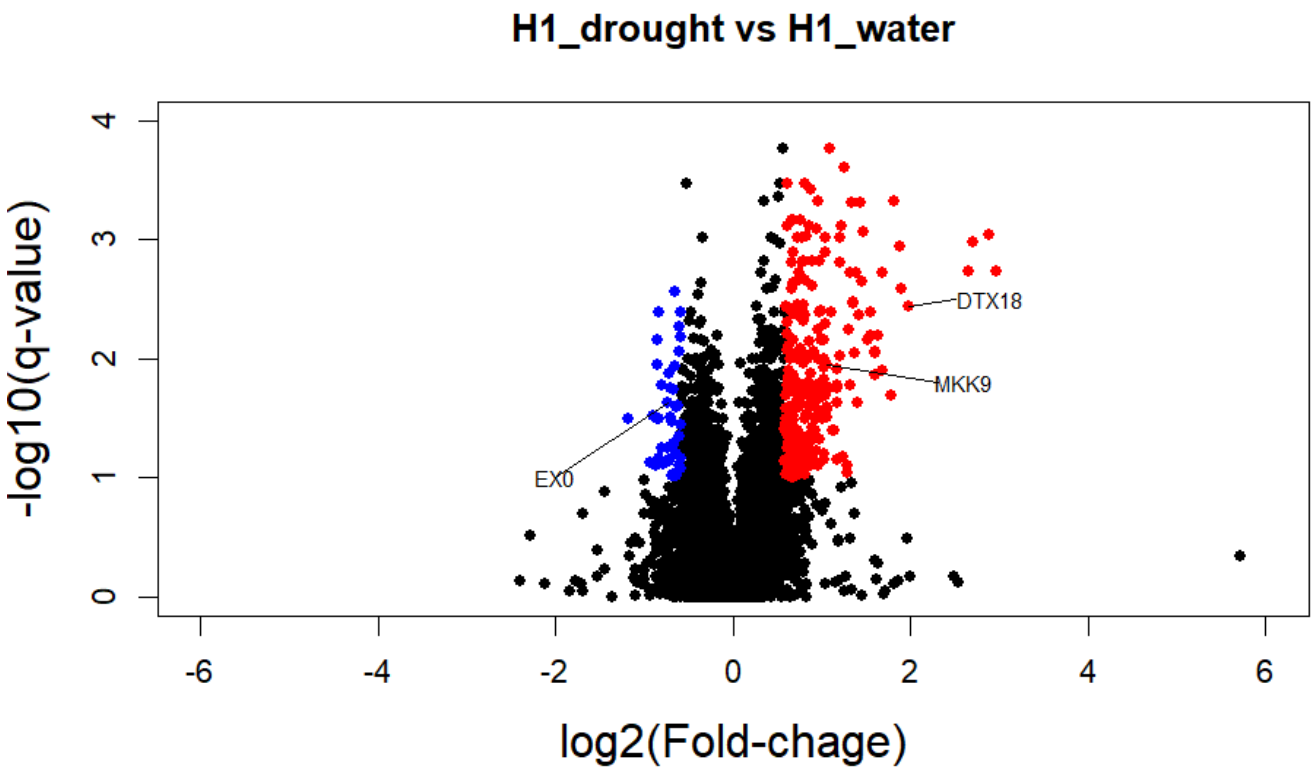
```
text(2.9,2.5,"DTX18",cex=0.7)
arrows(log.fold.change["AT1G73500"],log.q.val["AT1G73500"],2.3,1.8,length=0)
```

Hide

```
text(2.6,1.8,"MKK9",cex=0.7)
arrows(log.fold.change["AT5G15600"],log.q.val["AT5G15600"],-2,1,length=0)
```

Hide

```
text(-2,1,"EX0",cex=0.7)
```



Hide

```
write.table(activated.c2,file="activated_h1_drought_h1_water.txt",row.names = FALSE,quote=FAL
SE,col.names = FALSE)
write.table(repressed.c2,file="repressed_h1_drought_h1_water.txt",row.names = FALSE,quote=FAL
SE,col.names = FALSE)
```

Activated genes comparison 2 (activated_h1_drought_h1_water.txt) Repressed genes comparison 2 (repressed_h1_drought_h1_water.txt)

Using the selected criteria, we found 282 activated genes and 45 repressed genes. In this case, one of the activated genes is the mitogen-activated protein kinase MKK9, which has intracellular signalling function.

Comparison 3. H1_drought vs CTL_drought.

With this comparison, we analysed the effect of inoculating *Aeromonas* into *Arabidopsis* roots under drought conditions. Thus, we looked for genes whose expression is changed by *Aeromonas*.

Hide

```
h1_drought.ctl_drought <- topTable(contrast.results, number=32833,coef=3,sort.by="logFC")
log.fold.change <- h1_drought.ctl_drought$logFC
q.value <- h1_drought.ctl_drought$adj.P.Val
genes.ids <- rownames(h1_drought.ctl_drought)
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c3 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
repressed.c3 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
length(activated.c3)
```

```
[1] 84
```

Hide

```
length(repressed.c3)
```

```
[1] 28
```

Hide

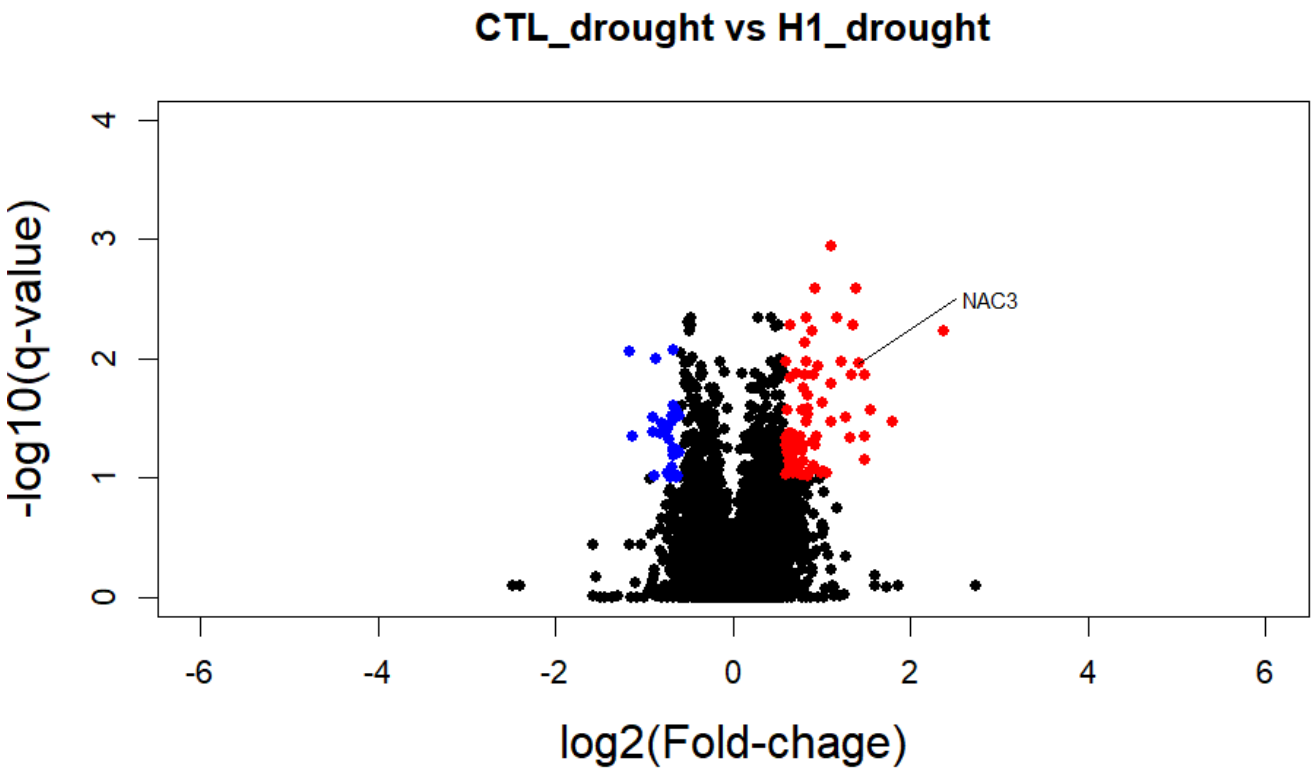
```
log.q.val <- -log10(q.value)
plot(log.fold.change,log.q.val,pch=19,col="black",cex=0.8,
     xlim=c(-6,6),ylim = c(0,4),
     xlab="log2(Fold-chage)",ylab="-log10(q-value)",cex.lab=1.5,
     main="CTL_drought vs H1_drought")

points(x = log.fold.change[activated.c3],
       y = log.q.val[activated.c3],col="red",cex=0.8,pch=19)
```

Hide

```
points(x = log.fold.change[repressed.c3],
       y = log.q.val[repressed.c3],col="blue",cex=0.8,pch=19)
arrows(log.fold.change["AT3G15500"],log.q.val["AT3G15500"],2.5,2.5,length=0)
```

Hide

```
text(2.9,2.5,"NAC3",cex=0.7)
```



Hide

```
write.table(activated.c3,file="activated_h1_drought_ctl_drought.txt",row.names = FALSE,quote=
FALSE,col.names = FALSE)
write.table(repressed.c3,file="repressed_h1_drought_ctl_drought.txt",row.names = FALSE,quote=
FALSE,col.names = FALSE)
```

Activated genes comparison 3 (activated_h1_drought_ctl_drought.txt) Repressed genes comparison 3 (repressed_h1_drought_ctl_drought.txt)

In this comparison, 84 activated genes and 28 repressed genes were detected. The activated genes include NAC3, a transcription factor involved in developmental processes.

Comparison 4. H1_water vs CTL_water

With this fourth comparison, we studied the effect of inoculation with *Aeromonas* when *Arabidopsis* is under basal conditions.

Hide

```
h1_water.ctl_water <- topTable(contrast.results, number=32833,coef=4,sort.by="logFC")
log.fold.change <- h1_water.ctl_water$logFC
q.value <- h1_water.ctl_water$adj.P.Val
genes.ids <- rownames(h1_water.ctl_water)
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c4 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
repressed.c4 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
length(activated.c4)
```

```
[1] 0
```

Hide

```
length(repressed.c4)
```

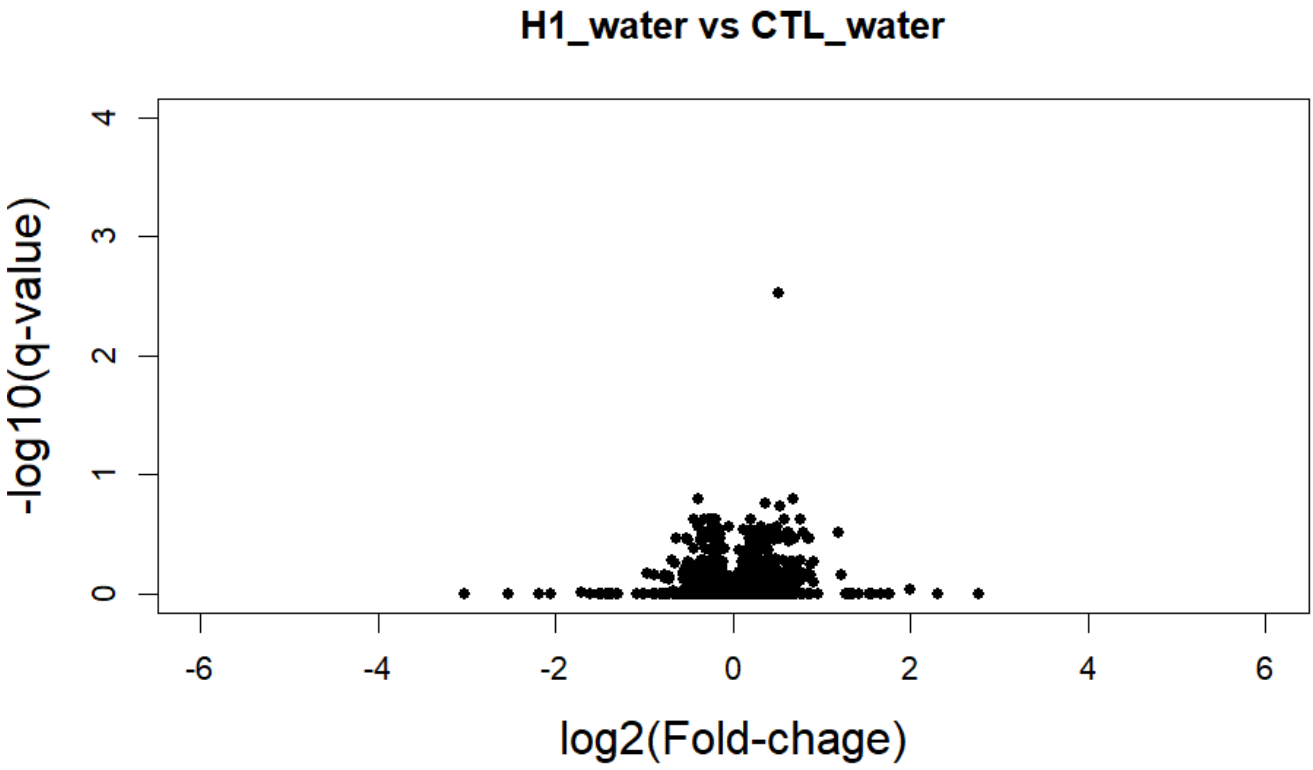
```
[1] 0
```

Hide

```
log.q.val <- -log10(q.value)
plot(log.fold.change,log.q.val,pch=19,col="black",cex=0.8,
     xlim=c(-6,6),ylim = c(0,4),
     xlab="log2(Fold-chage)",ylab="-log10(q-value)",cex.lab=1.5,
     main="H1_water vs CTL_water")

points(x = log.fold.change[activated.c4],
       y = log.q.val[activated.c4],col="red",cex=0.8,pch=19)
```

Hide

```
points(x = log.fold.change[repressed.c4],
       y = log.q.val[repressed.c4],col="blue",cex=0.8,pch=19)
```



We can see that no genes are differentially expressed in this comparison. When plants are not subjected to drought stress, treatment with *Aeromonas* has no effect on the transcriptome of the aerial part.

Comparison 5. H1_drought vs CTL_water

Finally, we compared gene expression in control plants (irrigated and uninoculated) with plants inoculated with *Aeromonas* and under drought conditions. In this way, we can observe a synergistic effect between the drought treatment and the effect produced by the bacteria.

Hide

```
h1_drought.ctl_water <- topTable(contrast.results, number=32833,coef=5,sort.by="logFC")
log.fold.change <- h1_drought.ctl_water$logFC
q.value <- h1_drought.ctl_water$adj.P.Val
genes.ids <- rownames(h1_drought.ctl_water)
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c5 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
repressed.c5 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
length(activated.c5)
```

```
[1] 556
```

Hide

```
length(repressed.c5)
```

```
[1] 150
```

Hide

```
log.q.val <- -log10(q.value)
plot(log.fold.change,log.q.val,pch=19,col="black",cex=0.8,
     xlim=c(-6,6),ylim = c(0,4),
     xlab="log2(Fold-chage)",ylab="-log10(q-value)",cex.lab=1.5,
     main="H1_drought vs CTL_water")

points(x = log.fold.change[activated.c5],
       y = log.q.val[activated.c5],col="red",cex=0.8,pch=19)
```

Hide

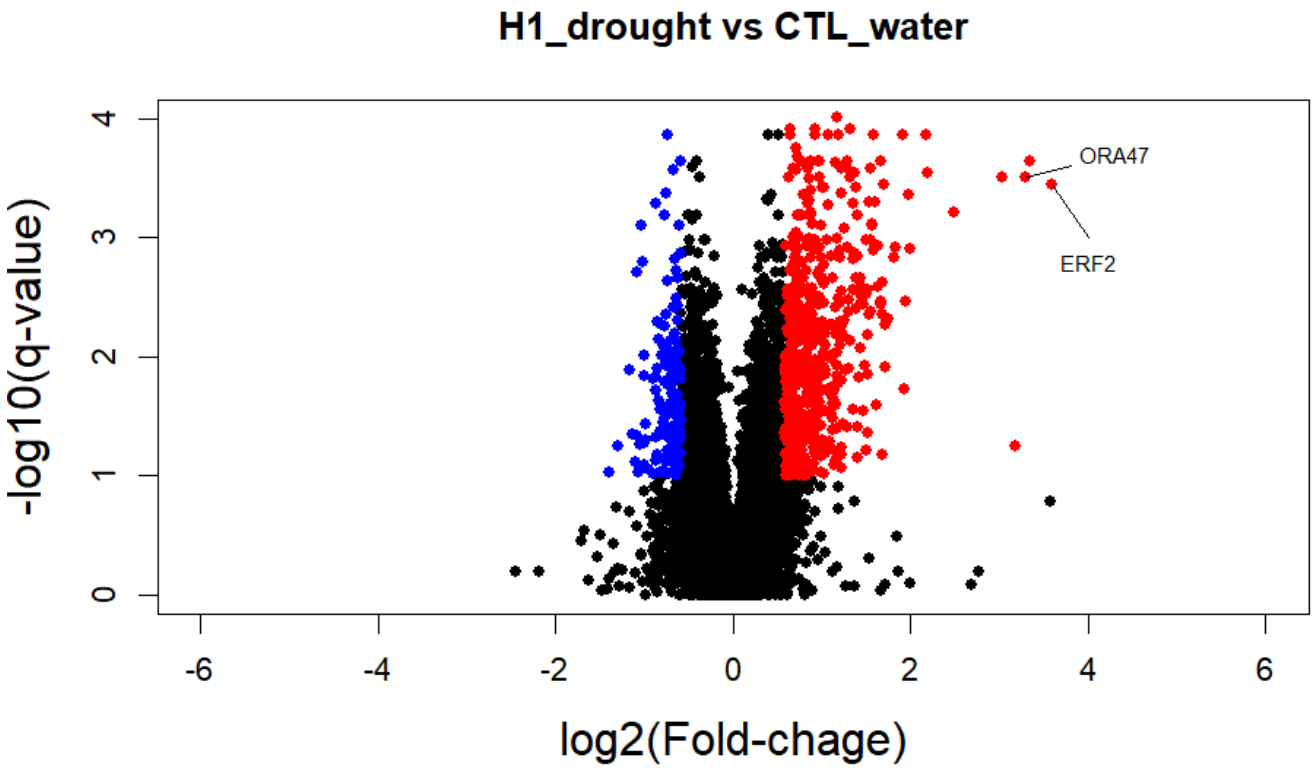
```
points(x = log.fold.change[repressed.c5],
       y = log.q.val[repressed.c5],col="blue",cex=0.8,pch=19)
arrows(log.fold.change["AT1G74930"],log.q.val["AT1G74930"],3.8,3.6,length=0)
```

Hide

```
text(4.3,3.7,"ORA47",cex=0.7)
arrows(log.fold.change["AT5G47220"],log.q.val["AT5G47220"],4,3,length=0)
```

Hide

```
text(4,2.8,"ERF2",cex=0.7)
```



Hide

```
write.table(activated.c5,file="activated_h1_drought_ctl_water.txt",row.names = FALSE,quote=FA
LSE,col.names = FALSE)
write.table(repressed.c5,file="repressed_h1_drought_ctl_water.txt",row.names = FALSE,quote=FA
LSE,col.names = FALSE)
```

Activated genes comparison 5 (activated_h1_drought_ctl_water.txt) Repressed genes comparison 5 (repressed_h1_drought_ctl_water.txt)

In the latter comparison, 556 activated genes and 150 repressed genes have been identified. The gene ORA47, a transcription factor that also belongs to the family known as ERF/AP2, stands out.

Analysis using DESeq2

Hide

```
library(DESeq2)
gene.count.matrix <- read.table(file = "gene_count_matrix.csv",header = T,sep = ",")
gene.ids <- sapply(X = strsplit(x = gene.count.matrix$gene_id,split = "\\|"),FUN = function
(x){return(x[1])})

gene.count.matrix <- gene.count.matrix[,-1]
rownames(gene.count.matrix) <- gene.ids

dds <- DESeqDataSetFromMatrix(countData=gene.count.matrix, colData=pheno.data, design = ~ aer
omonas + drought)
dds$aeromonas <- relevel(dds$aeromonas, ref = "ctl")
dds$drought <- relevel(dds$drought, ref = "ctl")

dds <- DESeq(dds)
mod_mat <- model.matrix(design(dds), colData(dds))
ctl_water <- colMeans(mod_mat[dds$aeromonas == "ctl" & dds$drought == "ctl", ])
ctl_drought <- colMeans(mod_mat[dds$aeromonas == "ctl" & dds$drought == "drought", ])
h1_water <- colMeans(mod_mat[dds$aeromonas == "h1" & dds$drought == "ctl", ])
h1_drought <- colMeans(mod_mat[dds$aeromonas == "h1" & dds$drought == "drought", ])

# Comparison 1
res1 <- results(dds, contrast = ctl_drought - ctl_water)
genes.ids <- rownames(res1)
log.fold.change <- res1$log2FoldChange
q.value <- res1$padj
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c1.deseq2 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
activated.c1.deseq2 <- activated.c1.deseq2[!is.na(activated.c1.deseq2)]
repressed.c1.deseq2 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
repressed.c1.deseq2 <- repressed.c1.deseq2[!is.na(repressed.c1.deseq2)]
length(activated.c1.deseq2)
```

```
[1] 324
```

Hide

```
length(repressed.c1.deseq2)
```

```
[1] 183
```

Hide

```
# Comparison 2
res2 <- results(dds, contrast = h1_drought - h1_water)
genes.ids <- rownames(res2)
log.fold.change <- res2$log2FoldChange
q.value <- res2$padj
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c2.deseq2 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
activated.c2.deseq2 <- activated.c2.deseq2[!is.na(activated.c2.deseq2)]
repressed.c2.deseq2 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
repressed.c2.deseq2 <- repressed.c2.deseq2[!is.na(repressed.c2.deseq2)]
length(activated.c2.deseq2)
```

```
[1] 324
```

Hide

```
length(repressed.c2.deseq2)
```

```
[1] 183
```

Hide

```
# Comparison 3
res3 <- results(dds, contrast = h1_drought - ctl_drought)
genes.ids <- rownames(res3)
log.fold.change <- res3$log2FoldChange
q.value <- res3$padj
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c3.deseq2 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
activated.c3.deseq2 <- activated.c3.deseq2[!is.na(activated.c3.deseq2)]
repressed.c3.deseq2 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
repressed.c3.deseq2 <- repressed.c3.deseq2[!is.na(repressed.c3.deseq2)]
length(activated.c3.deseq2)
```

```
[1] 101
```

Hide

```
length(repressed.c3.deseq2)
```

```
[1] 30
```

Hide

```
# Comparison 4
res4 <- results(dds, contrast = h1_water - ctl_water)
genes.ids <- rownames(res4)
log.fold.change <- res4$log2FoldChange
q.value <- res4$padj
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c4.deseq2 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
activated.c4.deseq2 <- activated.c4.deseq2[!is.na(activated.c4.deseq2)]
repressed.c4.deseq2 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
repressed.c4.deseq2 <- repressed.c4.deseq2[!is.na(repressed.c4.deseq2)]
length(activated.c4.deseq2)
```

```
[1] 101
```

Hide

```
length(repressed.c4.deseq2)
```

```
[1] 30
```

Hide

```
# Comparison 5
res5 <- results(dds, contrast = h1_drought - ctl_water)
genes.ids <- rownames(res5)
log.fold.change <- res5$log2FoldChange
q.value <- res5$padj
names(log.fold.change) <- genes.ids
names(q.value) <- genes.ids
activated.c5.deseq2 <- genes.ids[log.fold.change > log2(1.5) & q.value < 0.1]
activated.c5.deseq2 <- activated.c5.deseq2[!is.na(activated.c5.deseq2)]
repressed.c5.deseq2 <- genes.ids[log.fold.change < - log2(1.5) & q.value < 0.1]
repressed.c5.deseq2 <- repressed.c5.deseq2[!is.na(repressed.c5.deseq2)]
length(activated.c5.deseq2)
```

```
[1] 613
```

Hide

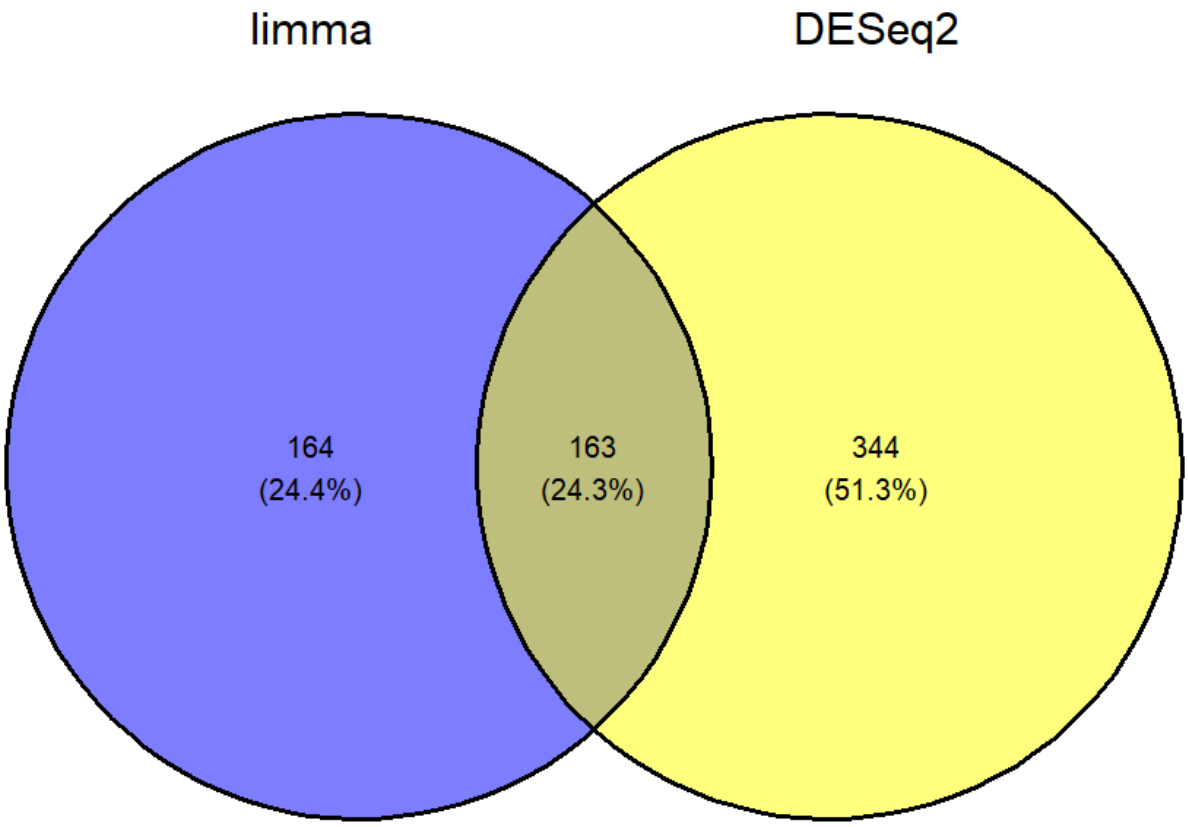
```
length(repressed.c5.deseq2)
```

```
[1] 282
```

Results comparison limma vs DESeq2

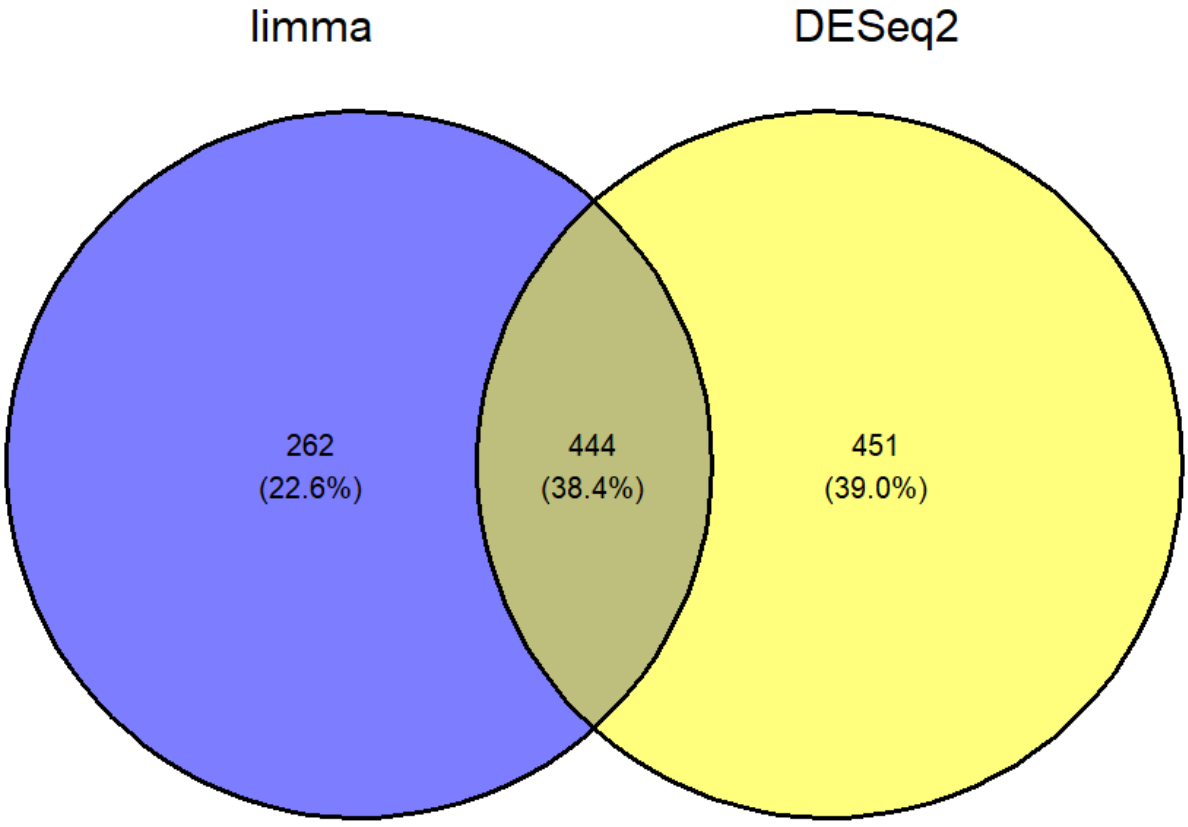
Hide

```
library(ggvenn)
ggvenn(list(limma=c(activated.c2,repressed.c2),DESeq2=c(activated.c2.deseq2,repressed.c2.dese
q2)))
```



Hide

```
ggvenn(list(limma=c(activated.c5,repressed.c5),DESeq2=c(activated.c5.deseq2,repressed.c5.dese
q2)))
```



In these Venn diagrams, we compare the DEGs obtained using limma with those identified by DESeq2 in the h1_drought vs h1_water and h1_drought vs ctl_drought comparisons.

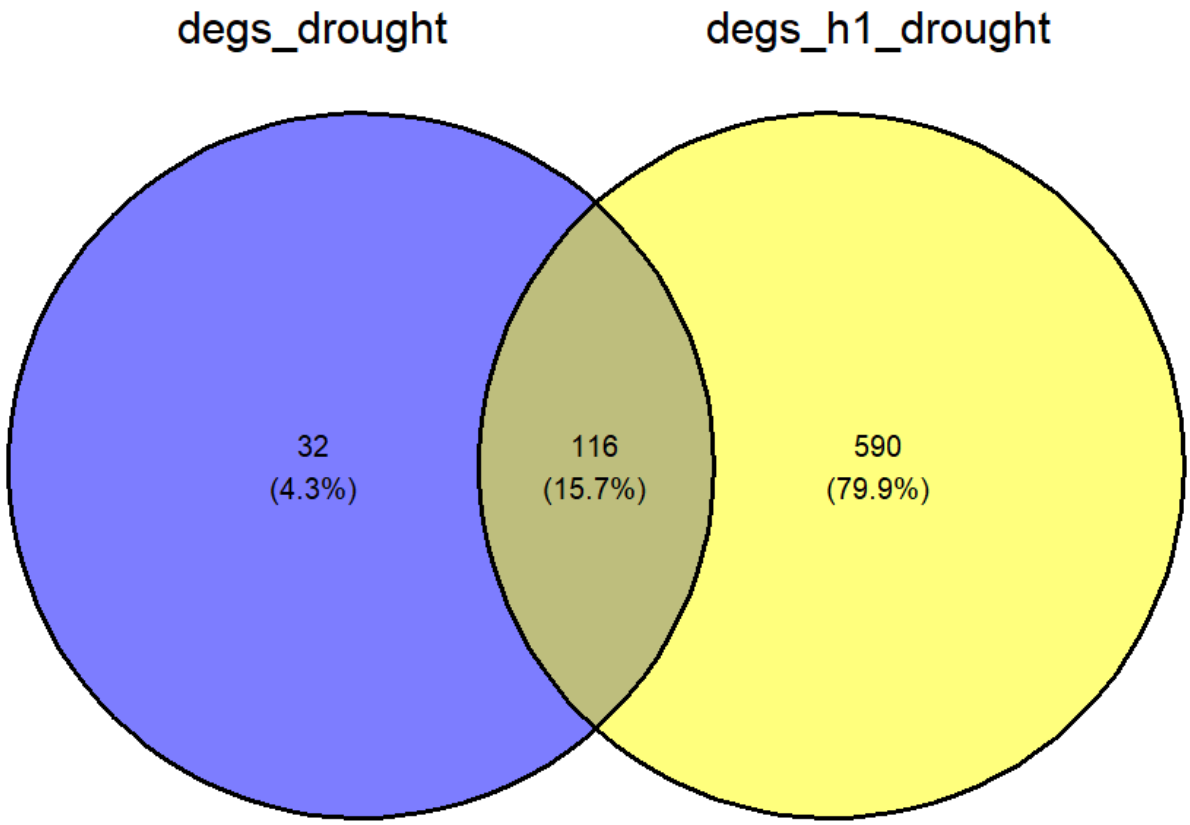
As we can see, the gene sets obtained by analysing the data with limma and DESeq2 are quite different. We do not expect a total overlap, as there are differences between the analyses with limma (which uses fpkm to measure gene expression) and DESeq2 (counts). As is typical, DESeq2 identifies a greater number of DEGs, as it performs more sensitive analyses. However, DESeq2 also tends to result in a higher number of false positives.

Furthermore, in this particular study, we have observed that the effect on the transcriptome is relatively subtle. In case of a more pronounced effect, where we could set more restrictive fold-change and q-value thresholds, we might observe better agreement between analyses.

Comparisons between gene sets

Hide

```
ggvenn(list(degs_drought=c(activated.c1,repressed.c1),degs_h1_drought=c(activated.c5,repressed.c5)))
```



In this Venn diagram, we compare two sets of genes:

- DEGs Ctl_water vs Ctl_drought
- DEGs Ctl_water vs H1_drought

We can see that there are 116 genes that change in both cases, i.e. genes that are activated by drought conditions but in which *Aeromonas* is not involved. On the other hand, there are 590 genes that change when drought is applied to inoculated plants but not to uninoculated plants. These must be genes whose expression is changed by *Aeromonas*, and give the plant improved drought tolerance.

Gene Ontology Term Enrichment Analysis

Gene ontology allows the systematic and unambiguous incorporation of information into genes by annotation. It consists of a structured (from more generic to more specific) and controlled vocabulary of terms that describe gene products in terms of biological processes, cellular components and molecular functions.

An enrichment of gene ontology terms aims to improve the representation of genetic information, making it more accurate and complete, so that it is possible to effectively interpret the results.

Hide

```
library(clusterProfiler)
library(org.At.tair.db)
library(enrichplot)
```

Differentially activated genes H1_drought vs CTL_drought

Here we perform GO term enrichment of the differentially activated gene set in *Arabidopsis thaliana* plants inoculated with *Aeromonas* sp. under drought conditions compared to non-inoculated plants under drought conditions.

Hide

```
activated_h1_d_ctl_d <- read.table(file="activated_h1_drought_ctl_drought.txt")[[1]]

activated_h1_d_ctl_d.atha.enrich.go <- enrichGO(gene          = activated_h1_d_ctl_d,
                                                OrgDb         = org.At.tair.db,
                                                ont           = "BP",
                                                pAdjustMethod = "BH",
                                                pvalueCutoff  = 0.05,
                                                readable      = FALSE,
                                                keyType = "TAIR")

df.1 <- as.data.frame(activated_h1_d_ctl_d.atha.enrich.go)

write.table(as.data.frame(df.1$pvalue,df.1$ID), file = "activated_h1_d_ctl_d_revigo", quote =
FALSE)
```

ReVIGO screenshot.

GO Term and q-value	Description	Representative Genes
GO:0002376 (GO:0002376) 7.369807e-05	immune system process	ERF2/PEN3/CYP81F2
GO:0008219 (GO:0008219) 0.0005994776	cell death	PEN3/PLA2A/NUDT7
GO:0009867 (GO:0009867) 1.141716e-12	jasmonic acid mediated signaling pathway	JAZ7/ERF2/NAC3
GO:0042430 (GO:0042430) 3.081945e-05	indole-containing compound metabolic process	HSPRO2/PEN3/CYP81F2
GO:0042908 (GO:0042908) 0.004019064	xenobiotic transport	DTX18
GO:0090693 (GO:0090693) 0.001098587	plant organ senescence	BGLU11/MKK9/NAC6

The table shows the most represented biological process from each group of interrelated processes that are significantly altered. Among the most represented, the jasmonic acid (JA)-mediated signalling pathway is significantly enriched compared to the others by having the lowest qvalue. These observations can be corroborated by looking at the treemap image, which shows that the largest block of all is the one corresponding to this pathway. Moreover, the size of the same group (blocks of the same colour) clearly occupies almost the entire image indicating that these are very important processes for the resistance to water stress provided by *Aeromonas* sp., which is why they are mostly activated pathways.

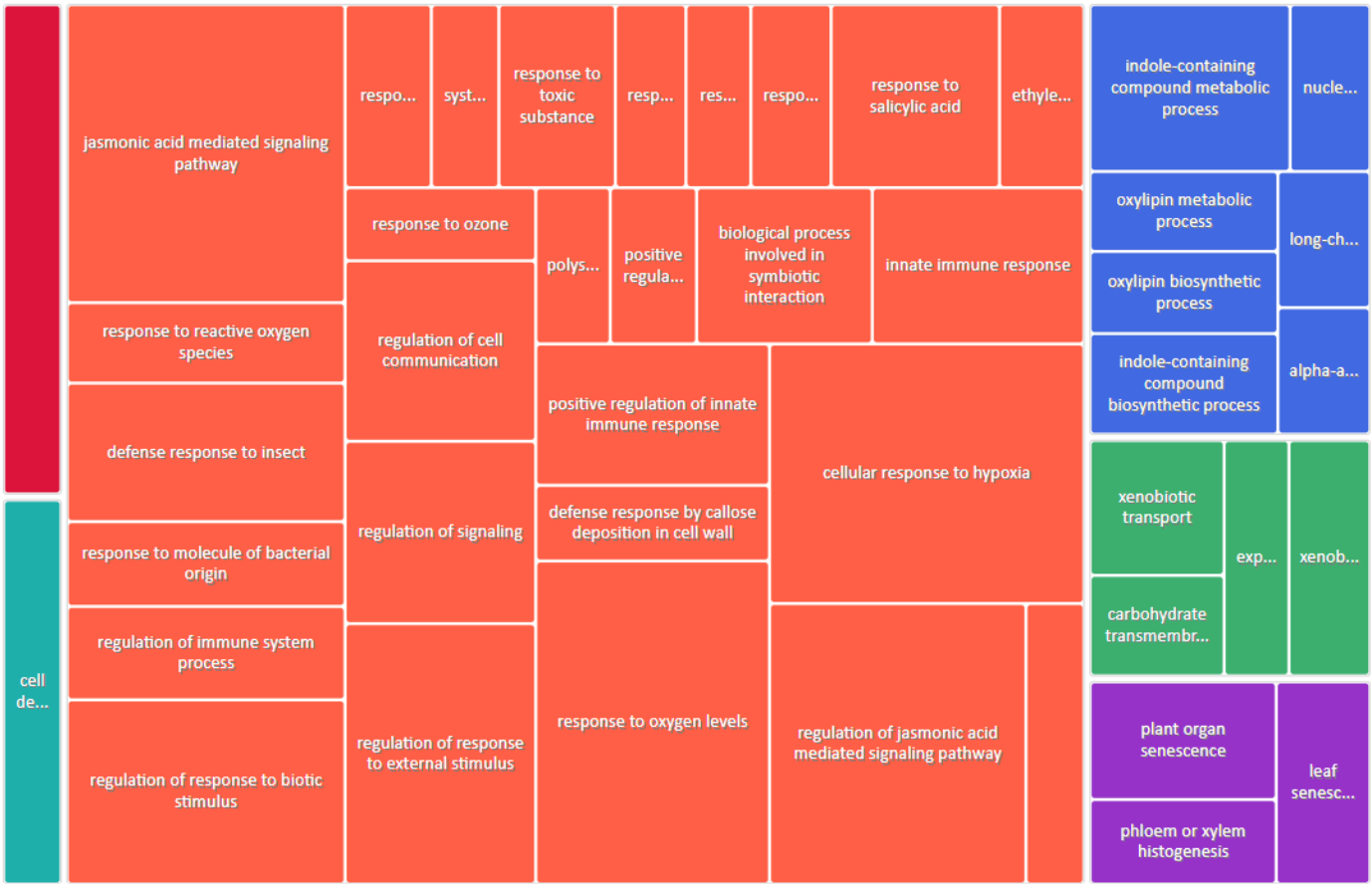
Furthermore, the GO analysis highlights JA-related DEGs, such as JAZ7, ERF2 and NAC3, which are known to promote plant resistance to dehydration stress. NAC3 has been obtained previously in analyses of differentially activated genes for these conditions, and ERF2 appears to be involved in other enriched processes as well. We can also highlight PEN3 as regulating several important processes.

Differentially repressed genes H1_drought vs CTL_drought

Here we perform GO term enrichment of the set of differentially repressed genes in *Arabidopsis thaliana* plants inoculated with *Aeromonas* sp. under drought conditions compared to non-inoculated plants under drought conditions.

Hide

```
repressed_h1_d_ctl_d <- read.table(file="repressed_h1_drought_ctl_drought.txt")[[1]]

repressed_h1_d_ctl_d.atha.enrich.go <- enrichGO(gene          = repressed_h1_d_ctl_d,
                                                OrgDb         = org.At.tair.db,
                                                ont           = "BP",
                                                pAdjustMethod = "BH",
                                                pvalueCutoff  = 0.05,
                                                readable      = FALSE,
                                                keyType = "TAIR")

df.2 <- as.data.frame(repressed_h1_d_ctl_d.atha.enrich.go)

write.table(as.data.frame(df.2$pvalue,df.2$ID), file="repressed_h1_d_ctl_d_revigo", quote=FAL
SE)
```



ReVIGO screenshot.

GO Term and q-value	Description	Representative Genes
GO:0009639 (GO:0009639) 0.0002585459	response to red or far red light	ERF34/ERD9/BBX32
GO:0009813 (GO:0009813) 0.009531447	flavonoid biosynthetic process	TT4/FLS1
GO:0031537 (GO:0031537) 0.0118085	regulation of anthocyanin metabolic process	TT4

In this case it is possible to see both in the treemap and in the table that there is an enrichment of repressed genes in response to red or far-red light in the inoculated plants, followed by the process of flavonoid biosynthesis.

Differentially activated genes H1_drought vs CTL_water

Here we perform GO term enrichment of the differentially activated gene set in *Arabidopsis thaliana* plants inoculated with *Aeromonas* sp. under drought conditions compared to non-inoculated and irrigated plants.

Hide

```
activated_h1_d_ctl_w <- read.table(file="activated_h1_drought_ctl_water.txt")[[1]]

activated_h1_d_ctl_w.atha.enrich.go <- enrichGO(gene          = activated_h1_d_ctl_w,
                                                OrgDb         = org.At.tair.db,
                                                ont           = "BP",
                                                pAdjustMethod = "BH",
                                                pvalueCutoff  = 0.05,
                                                readable      = FALSE,
                                                keyType = "TAIR")

df.3 <- as.data.frame(activated_h1_d_ctl_w.atha.enrich.go)

write.table(as.data.frame(df.3$pvalue,df.3$ID), file="activated_h1_d_ctl_w_revigo", quote=FAL
SE)
```

ReVIGO screenshot.

GO Term and q-value	Description	Representative Genes
GO:0002376 (GO:0002376) 4.074928e-18	immune system process	ERF2/PLA2A/PEN3
GO:0008219 (GO:0008219) 1.272887e-08	cell death	PLA2A/PEN3/NAC6
GO:0009751 (GO:0009751) 2.481618e-49	response to salicylic acid	BT2/ATS40.4/SYP122
GO:0009966 (GO:0009966) 1.864563e-14	regulation of signal transduction	JAZ7/JAZ1/JAZ6
GO:0010150 (GO:0010150) 3.679929e-21	leaf senescence	RD26/MKK9/SAG21

GO Term and q-value	Description	Representative Genes
GO:0012501 (GO:0012501) 8.360128e-09	programmed cell death	PLA2A/PEN3/NAC6
GO:0034219 (GO:0034219) 0.003716663	carbohydrate transmembrane transport	ERD6/STP1/ESL1
GO:0042430 (GO:0042430) 2.39423e-23	indole-containing compound metabolic process	IGMT1/UGT74E2/MKK9
GO:0045229 (GO:0045229) 0.01758479	external encapsulating structure organization	PEN3/TCH4/CYP81F2

The response to salicylic acid is noteworthy for its high enrichment compared to the rest of the processes, as is the whole group compared to the rest of the groups.

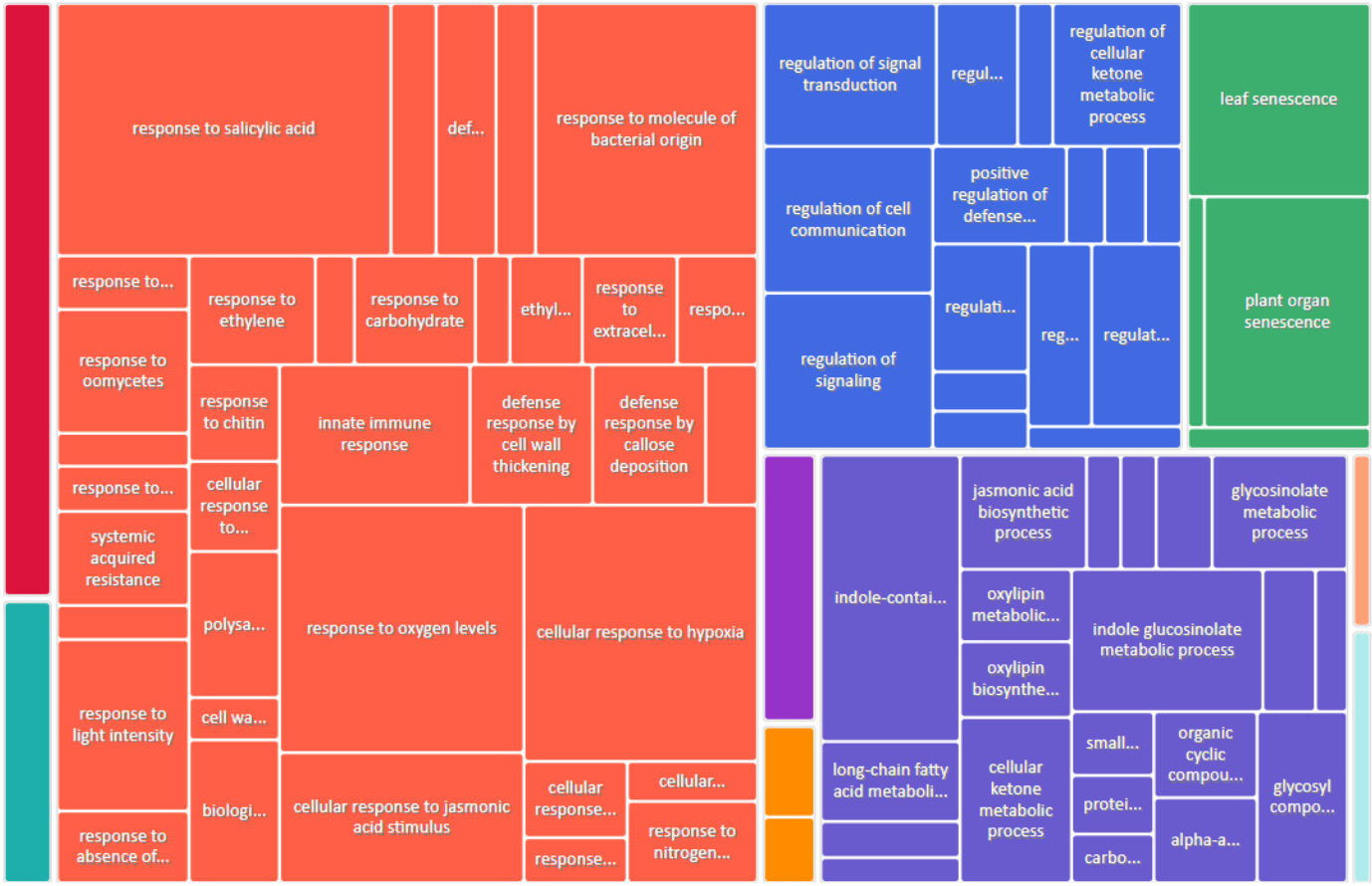
In relation to the most representative genes, we can highlight ERF2 corresponding to the activation of the immune system, one of the most enriched processes. On the other hand, we again see the JAZ7 gene, as in the comparison of H1_drought vs CTL_drought activated genes, which suggests that it is fundamental for the response to water stress.

Differentially repressed genes H1_drought vs CTL_water

Here we perform GO term enrichment of the set of differentially repressed genes in *Arabidopsis thaliana* plants inoculated with *Aeromonas* sp. under drought conditions compared to non-inoculated and irrigated plants.

Hide

```
repressed_h1_d_ctl_w <- read.table(file="repressed_h1_drought_ctl_water.txt")[[1]]

repressed_h1_d_ctl_w.atha.enrich.go <- enrichGO(gene          = repressed_h1_d_ctl_w,
                                                OrgDb         = org.At.tair.db,
                                                ont           = "BP",
                                                pAdjustMethod = "BH",
                                                pvalueCutoff  = 0.05,
                                                readable      = FALSE,
                                                keyType = "TAIR")

df.4 <- as.data.frame(repressed_h1_d_ctl_w.atha.enrich.go)

write.table(as.data.frame(df.4$pvalue,df.4$ID), file="repressed_h1_d_ctl_w_revigo", quote=FAL
SE)
```



ReVIGO screenshot.

GO Term and q-value	Description	Representative Genes
GO:0000278 (GO:0000278) 0.001719728	mitotic cell cycle	CYCA1;1/TPX2/CYCB1;1
GO:0007017 (GO:0007017) 0.01155884	microtubule-based process	TPX2/SP1L4/PAKRP2
GO:0008356 (GO:0008356) 0.01895755	asymmetric cell division	TMM/BASL
GO:0009411 (GO:0009411) 0.0008523922	response to UV	ELIP1/MYB4/F3H
GO:0009813 (GO:0009813) 0.0001719778	flavonoid biosynthetic process	FLS1/F3H/RHM1
GO:0010374 (GO:0010374) 0.001380033	stomatal complex development	TMM/CKX6/EPF2
GO:0042546 (GO:0042546) 0.004344226	cell wall biogenesis	ERF34/AGP17/RGP1
GO:0051338 (GO:0051338) 0.003546343	regulation of transferase activity	CYCA1;1/TPX2/CDC20.1

In this case, no major differences are observed in the enrichment of processes or groups of processes, the q-values being quite similar and the treemap quite homogeneous in terms of size. Although the most enriched processes coincide with those of the H1_drought vs CTL_drought differentially repressed gene condition.

Enrichment of KEGG route terms

Hide

```
# Differentially activated genes H1_drought vs CTL_drought
activated_h1_d_ctl_d.atha.enrich.kegg <- enrichKEGG(gene  = activated_h1_d_ctl_d,
                                                    organism = "ath",
                                                    pAdjustMethod = "BH",
                                                    pvalueCutoff  = 0.05)

df.activated_h1_d_ctl_d.atha.enrich.kegg <- as.data.frame(activated_h1_d_ctl_d.atha.enrich.ke
gg)

# Differentially activated genes H1_drought vs CTL_water
activated_h1_d_ctl_w.atha.enrich.kegg <- enrichKEGG(gene  = activated_h1_d_ctl_w,
                                                    organism = "ath",
                                                    pAdjustMethod = "BH",
                                                    pvalueCutoff  = 0.05)

df.activated_h1_d_ctl_w.atha.enrich.kegg <- as.data.frame(activated_h1_d_ctl_w.atha.enrich.ke
gg)
```

Visualisation of the alpha-linolenic acid metabolism pathway.

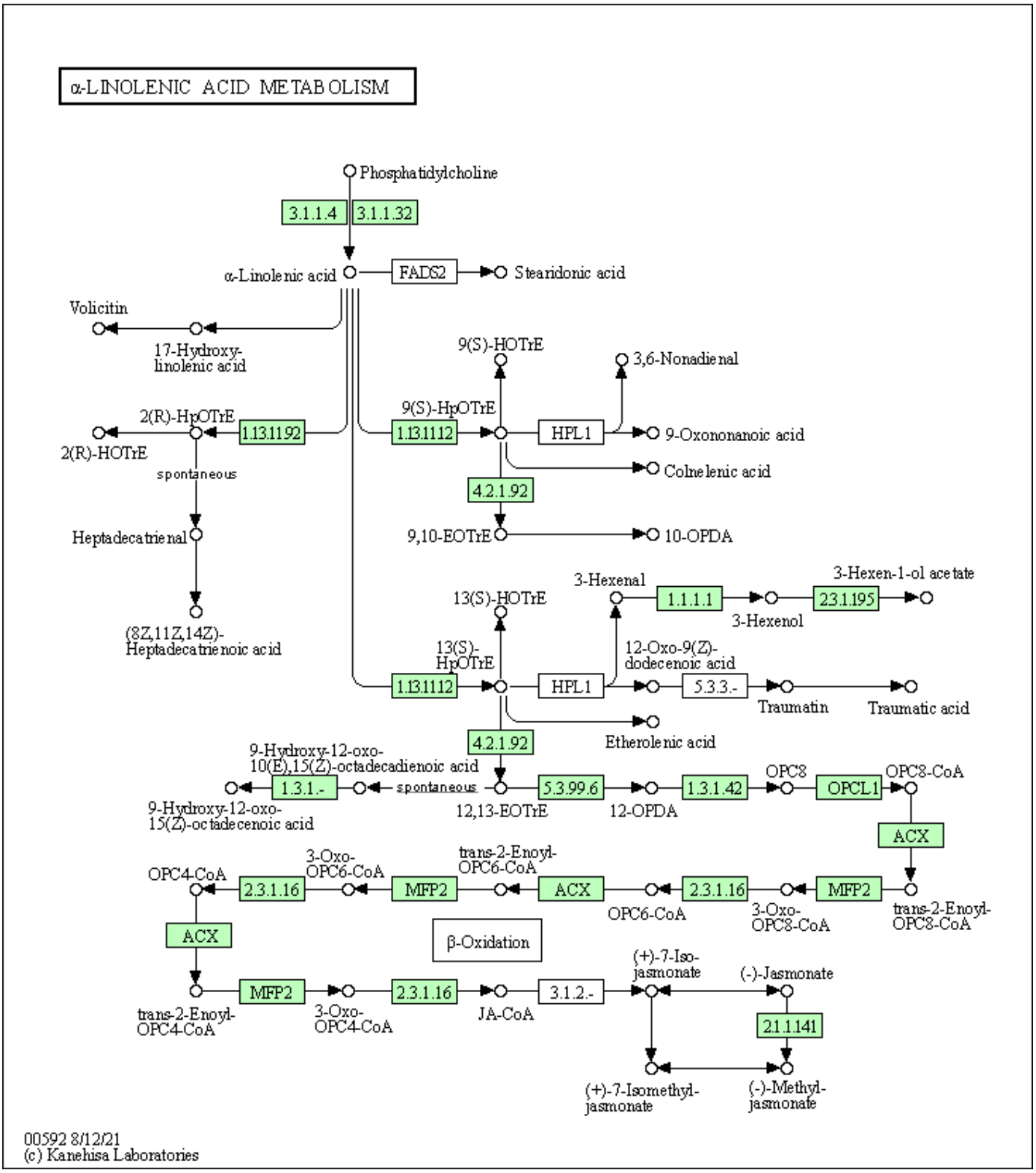
Hide

```
library(pathview)
```

Hide

```
pathview(gene.data = sort(log.fold.change,decreasing = TRUE),
         pathway.id = "ath00592",
         species = "ath",
         limit = list(gene=max(abs(log.fold.change)), cpd=1), gene.idtype = "TAIR")
```

```
[1] "Note: 6119 of 32833 unique input IDs unmapped."
```


α-LINOLENIC ACID METABOLISM
Phosphatidylcholine
3.1.1.4
3.1.1.32
α-Linolenic acid
FADS2
Stearidonic acid
Volicitin
17-Hydroxy-linolenic acid
9(S)-HOTrE
3,6-Nonadienal
9(S)-HpOTrE
2(R)-HpOTrE
1.13.11.92
1.13.11.12
HPL1
9-Oxononanoic acid
2(R)-HOTrE
spontaneous
Colnelenic acid
4.2.1.92
Heptadecatrienal
9,10-EOTrE
10-OPDA
3-Hexenal
1.1.1.1
23.1.195
3-Hexen-1-ol acetate
13(S)-HOTrE
3-Hexenol
(8Z,11Z,14Z)-Heptadecatrienoic acid
13(S)-HpOTrE
12-Oxo-9(Z)-dodecenoic acid
5.3.3.-
Traumatin
Traumatic acid
Etherolenic acid
9-Hydroxy-12-oxo-10(E),15(Z)-octadecadienoic acid
1.3.1.-
5.3.99.6
1.3.1.42
OPC8
OPCL1
OPC8-CoA
9-Hydroxy-12-oxo-15(Z)-octadecenoic acid
12,13-EOTrE
12-OPDA
ACX
OPC4-CoA
3-Oxo-OPC6-CoA
trans-2-Enoyl-OPC6-CoA
2.3.1.16
MFP2
ACX
OPC6-CoA
2.3.1.16
MFP2
3-Oxo-OPC8-CoA
trans-2-Enoyl-OPC8-CoA
β-Oxidation
ACX
(+)-7-Iso-jasmonate
(-)-Jasmonate
MFP2
2.3.1.16
3.1.2.-
21.1.141
trans-2-Enoyl-OPC4-CoA
3-Oxo-OPC4-CoA
JA-CoA
(+)-7-Isomethyl-jasmonate
(-)-Methyl-jasmonate
00592 8/12/21
(c) Kanehisa Laboratories

Hide

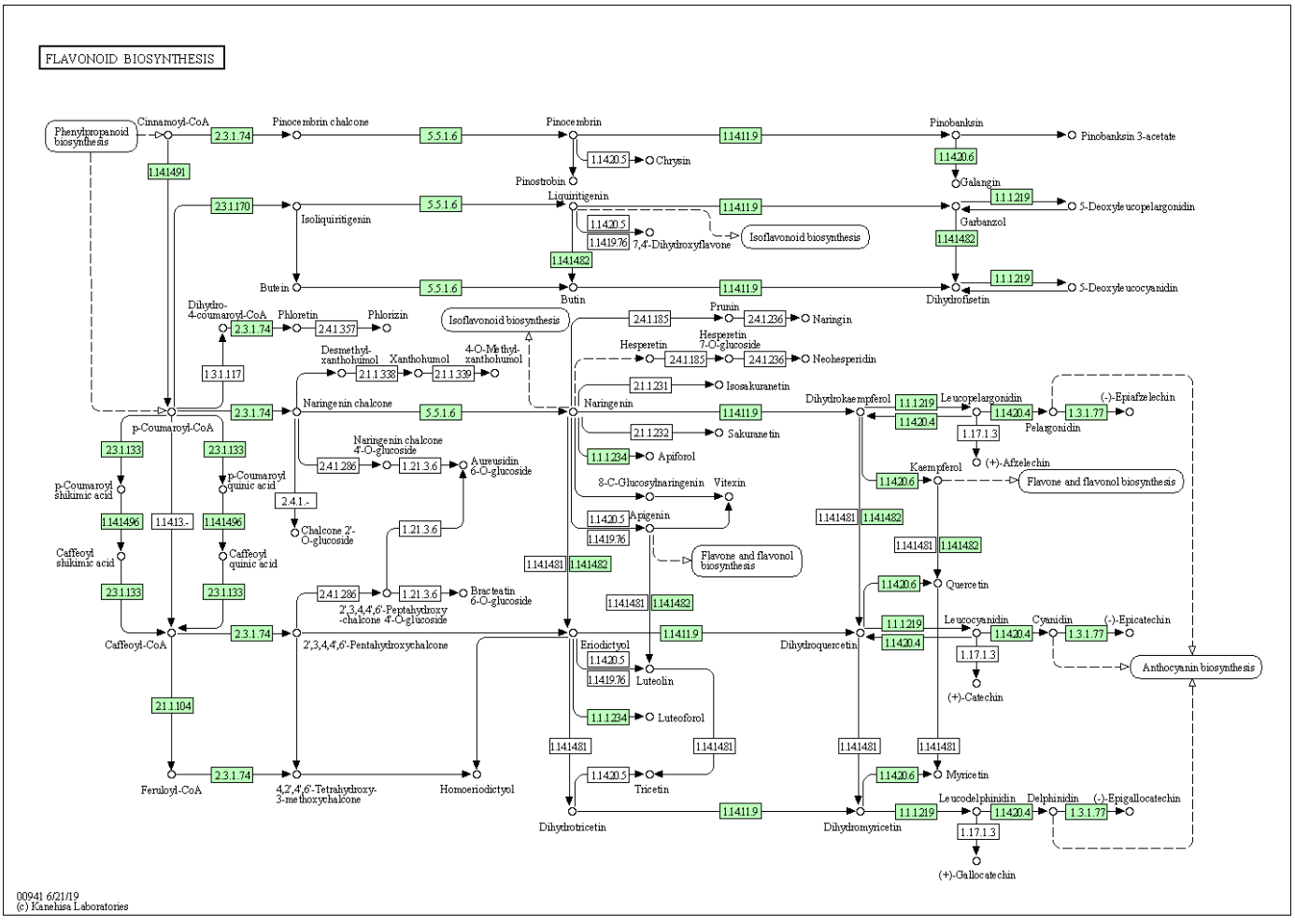
```
# Differentially repressed genes H1_drought vs CTL_drought
repressed_h1_d_ctl_w.atha.enrich.kegg <- enrichKEGG(gene  = repressed_h1_d_ctl_w,
                                                    organism = "ath",
                                                    pAdjustMethod = "BH",
                                                    pvalueCutoff  = 0.05)

df.repressed_h1_d_ctl_w.atha.enrich.kegg <- as.data.frame(repressed_h1_d_ctl_w.atha.enrich.ke
gg)

# Differentially repressed genes H1_drought vs CTL_water
repressed_h1_d_ctl_d.atha.enrich.kegg <- enrichKEGG(gene  = repressed_h1_d_ctl_d,
                                                    organism = "ath",
                                                    pAdjustMethod = "BH",
                                                    pvalueCutoff  = 0.05)

df.repressed_h1_d_ctl_d.atha.enrich.kegg <- as.data.frame(repressed_h1_d_ctl_d.atha.enrich.ke
gg)
```

Visualisation of the flavonoid biosynthesis pathway.

Hide

```
pathview(gene.data = sort(log.fold.change,decreasing = TRUE),
         pathway.id = "ath00941",
         species = "ath",
         limit = list(gene=max(abs(log.fold.change)), cpd=1), gene.idtype = "TAIR")
```

```
[1] "Note: 6119 of 32833 unique input IDs unmapped."
```



Conclusions

With the analyses carried out, we have obtained the differentially expressed genes for drought stress conditions and when inoculated with *Aeromonas* sp. We have observed how *Arabidopsis thaliana* under drought conditions is able to produce a stronger response to this abiotic stress (differentially expresses more genes) when treated with *Aeromonas* sp.

By studying the enrichment of GO terms, we have been able to verify that, to our surprise, the genes that are activated when plants are inoculated with *Aeromonas* under drought conditions are involved in the response to water stress. Our enrichment analyses have also revealed that jasmonic acid plays an important role in this process. On the other hand, we have seen that when comparing conditions in which the plant is watered, treatment with *Aeromonas* sp. does not have a detrimental effect on the plant.

As in our analysis, the Venn diagrams in the article show a synergistic behaviour between *Aeromonas* sp. and the response to drought stress in *Arabidopsis thaliana*. In addition, the article also points to the importance of the signalling pathway mediated by jasmonic acid and certain genes of particular importance for drought resistance such as JAZ7, ERF2 and NAC3 appear in the article. On the other hand, the study shows how flavonoids play an important role in the plant-microorganism interaction. This is consistent with the fact that flavonoid synthesis pathway genes are repressed when the plant is inoculated with *Aeromonas* sp. (as we have seen in the KEGG pathway term enrichment analysis).

A simple but effective experiment that would help us to explain the effects that *Aeromonas* sp would have on *Arabidopsis thaliana*, would be to compare two plants under dehydration conditions, one of them being treated with the bacterium and the other not. To do this, we would transplant the two plants, one week old and previously grown under the same conditions, to exactly the same medium. After this step, one of the plants is inoculated with a certain controlled amount of *Aeromonas* sp and from that moment on, the dehydration conditions begin. After about 2 weeks of cultivation, the plants are watered again. In this way, it can be seen that the plant inoculated with the bacteria shows a recovery process that the plant without *Aeromonas* is unable to achieve.

Reference

He, D., Kaushal, R., Peng, L., Singh, SK., et al (2022). Flavonoid-attracted Aeromonas sp. from the Arabidopsis root microbiome enhances plants dehydration resistance. The ISME Journal (cursiva), Volume 16, Issue 11, pp.2622-2632