Analysis of microarray data for biennial bearing in apple

Baptiste Guitton

February 2015

Introduction

In the present study, our objective was to investigate whether the processes involved in the transition from juvenile to adult phase are also key regulators of induction of flowering in SAM of adult trees. To achieve this, we studied the effect of the presence of fruit on the expression of genes involved in flowering transition in the terminal meristem of spur bourse shoots in adult apple trees, using a microarray analysis to identify genes and biological processes leading to biennial bearing.

Shoot Apical Meristems (SAM) were collected from adult 'Gala' (clone 'Galaxy') apple trees with contrasted bearing behavior, i.e. 'ON' trees bearing heavy crop and 'OFF' trees bearing no fruits. 'OFF' trees were completely deflowered at full bloom and no chemical thinning was applied to 'ON' trees. For 'ON' trees, meristems were sampled in terminal position of spurs issued from bourses carrying at least one fruit and for 'OFF' trees, meristems were sampled in terminal position of spurs bearing no fruit. Three collections of terminal buds were performed from May to August 2010, at 28, 48 and 119 Days After Full Bloom (DAFB). This period was considered to cover floral induction, floral initiation and the beginning of floral differentiation on spurs.

All spot comparisons were made between 'ON' (control) and 'OFF' trees at 28, 48 and 119 DAFB. For each time point, total RNA of two trees per treatment and three meristems per tree and per date, corresponding to six separate RNA extractions mixed in equal quantity constituted a biological replicate. Two independent biological repeats were performed with the 'ON' and 'OFF' cDNA clones labelled with Cy3 and Cy5 fluorescent dyes, respectively. A dye-switch was performed to eliminate any bias resulting from the two fluorescent dyes.

The purpose of this present script is to analyze the normalized microarray data with the aim to identify statistically deferentially expressed transcripts between the two treatments.

Libraries loading

```
# Load the specified libraries
library(knitr)
library(grid)
library(VennDiagram)
library(cluster)
library(wesanderson)
```

1. Threshold values justification and set-up

In our microarray experimentation, the dye-switch method was applied, inverting biological replicates in the same time as the fluorochromes. In this case of experiment, a p-value threshold of 0.05 is usually accepted. However, a p-value of 0.01 can be used to consider only highly significant transcripts.
The microarray community recommend considering only genes with a ratio higher to 1 or lower to -1. A ratio value of 0.5 is usually not accepted for publication by reviewers. Based on experiments using both microarray and qRT-PCR, it has been estimated that a log2 ratio value of 1 in a microarray correspond to 1 PCR cycle of difference between the test and the control in a qRT-PCR experiment.
Since data were already normalized using the lowess method, there is no need to filter on expression intensity. Indeed, normalized intensity values were subtracted from the background to provide an estimation of the

transcript expression levels.
Due to the recent whole genome duplication that occurred in the apple genome (Velasco *et al.*, 2010), duplicated genes share high similarity sequence. In the present analysis, we considered only transcripts that were amplified by probes highly specific to the targeted transcript. Indeed, non-specific probes are not of interest since they can amplify several homeologous transcript copies and thus provide incorrect expression values for a given targeted transcript.

```
# Threshold for p-value
pval_threshold <- 0.01

# Threshold for ratio
ratio_threshold_up <- 1  # Negative ratios correspond to up-regulated genes in 'OFF' trees
ratio_threshold_down <- -1  # Positive ratios correspond to up-regulated genes in 'ON' trees

# Threshold for expression intensity
intensity_threshold <- -0.2

# Threshold for probe specificity
probe_specificity_threshold <- 2
```

2. Data importation

All parameters that will be used for filtering and processing the data are saved in R objects that will be used in the following procedure.

```
# Name of the source file
file <- "Microarray_biennial_bearing_apple.csv"

# Probe parameters
probe_id <- "genes_probe_id"
gene_id <- "genes_seq_id"
AT_id <- "AT_Id"
probe_specificity <- "Spec"
probe_sense <- "Sense"

# Date 1
Expression_T1_D1 <- "Meristeme_1_2_ON_date1"
Expression_T2_D1 <- "Meristeme_1_2_Off_date1"
Ratio_D1 <- "R1"
Pval_D1 <- "pval1"

# Date 2
Expression_T1_D2 <- "Meristeme_1_2_ON_date2"
Expression_T2_D2 <- "Meristeme_1_2_Off_date2"
Ratio_D2 <- "R2"
Pval_D2 <- "pval2"

# Date 3
Expression_T1_D3 <- "Meristeme_1_2_ON_date3"
Expression_T2_D3 <- "Meristeme_1_2_Off_date3"
Ratio_D3 <- "R3"
Pval_D3 <- "pval3"
```

The present analysis is run on data that were first normalized with the lowess method. The present csv file contains normalized expression values, log2 ratio with associated p-value and FDR, along with annotation information for each probe.

```
# Read data file
Microarray_Data <- read.csv(file, header=T, sep=";", dec=".", na.string="")
```

3. Graphic representation of probe sense before filtering

3.1. Selection of probe sense

The following analysis will only be conducted on sense probes.

```
# Creation of two datasets: one for sense and one for anti-sense expressed transcripts
forward_sense_probes <- Microarray_Data[Microarray_Data[,probe_sense] == "S", ]
reverse_sense_probes <-  Microarray_Data[Microarray_Data[,probe_sense] == "AS", ]

# The "_r" is deleted in the gene_id column
forward_sense_probes[,gene_id] <- gsub(pattern = "_r",
                            replacement = "", x = forward_sense_probes[,gene_id])
```

After probe sense filtering, 63011 forward sense probes are selected and 63011 reverse sense probes are put aside.

3.2. Graphic representation before filtering

Before applying any filters, the probe density is graphically represented in relation to different parameters: probe specificity, ratio P-value, expression intensity and log ratio value. These representations enable to describe how probes behave in relation to these parameters and enable to visualize the effect of the filtering procedures.

Fig.1. Distribution of probe specificity of forward sense probes

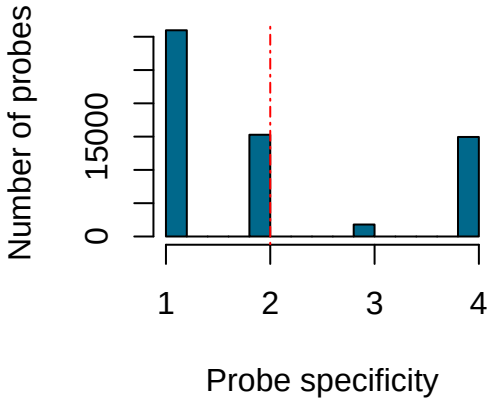


Fig.2. Distribution of expression intensity by treatment for each time point for forward sense probes

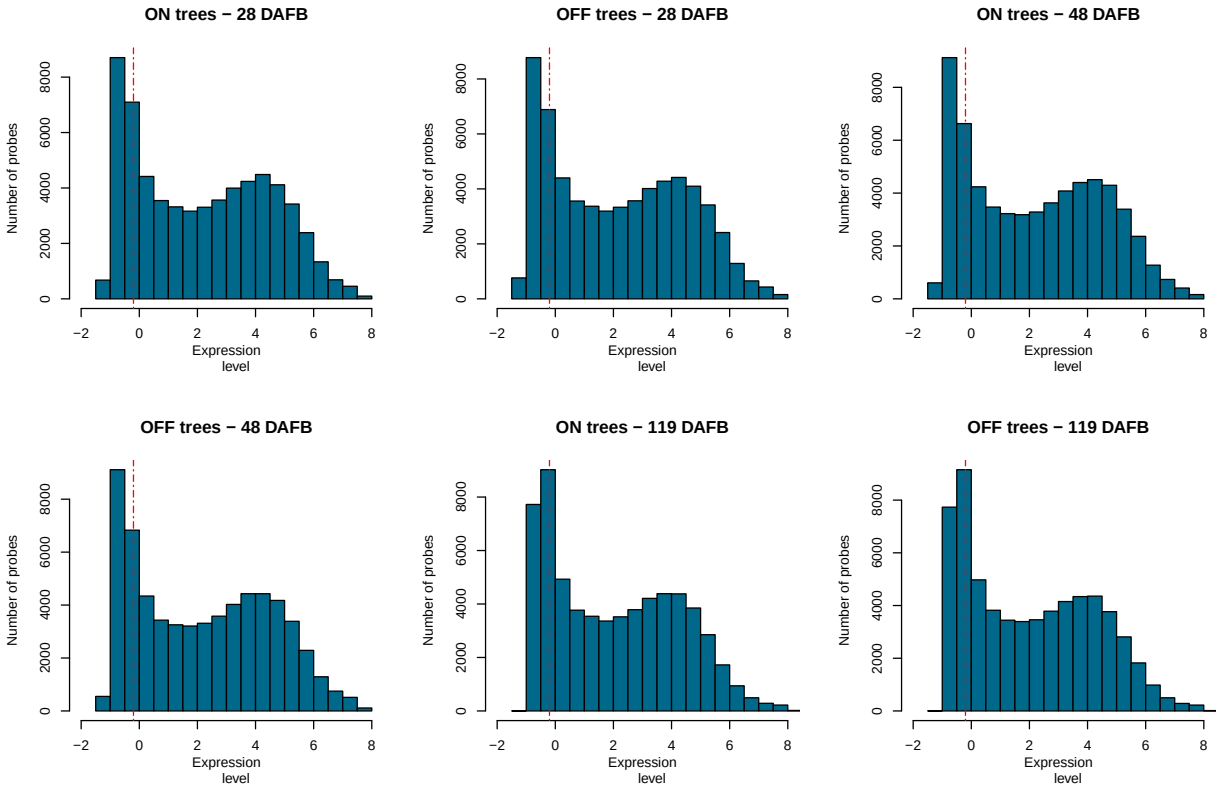


Fig.3. Distribution of p−value associated to the log2 ratio for each time point for forward sense probes

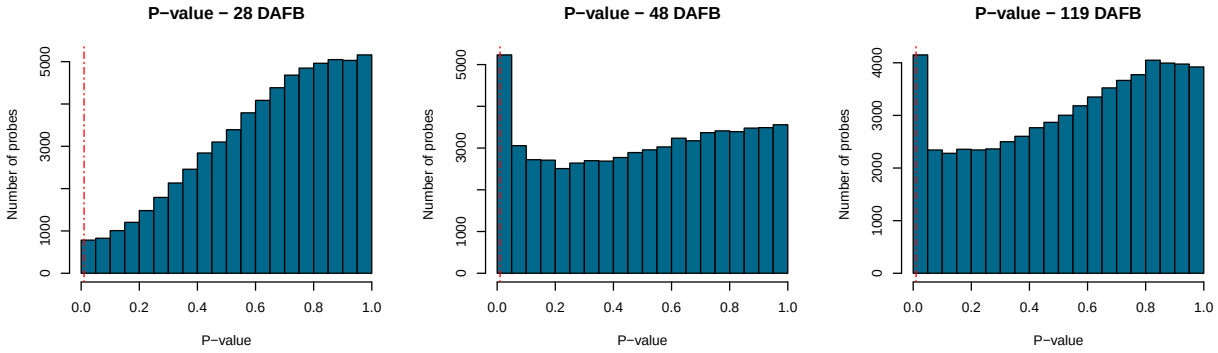
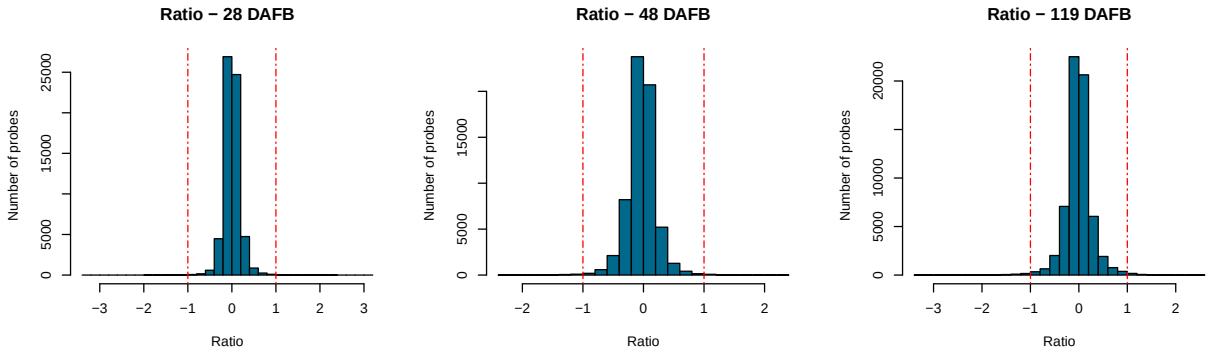


Fig.4. Distribution of log2 ratio for each time point for forward sense probes



4. Data filtering

4.1. Probe specificity filtering

Herein below, we are applying the probe specificity filter to consider only probes that show specificity lower than 2 within the sense probes.

```
specific_probes <- forward_sense_probes[forward_sense_probes[, probe_specificity]
                                        <= probe_specificity_threshold, ]
```

After filtering, 46261 sense probes with specific design will be considered in further analyses.

4.2. Expression intensity filtering

Herein below, we are applying the expression intensity filter to consider only probes that show expression higher than -0.2 within the sense probes.

```
# Application of the filter to the 3 dates
D1_expressed <- specific_probes[specific_probes[,Expression_T1_D1] >= intensity_threshold
                              & specific_probes[,Expression_T2_D1] >= intensity_threshold,]
D2_expressed <- specific_probes[specific_probes[,Expression_T1_D2] >= intensity_threshold
                              & specific_probes[,Expression_T2_D2] >= intensity_threshold,]
D3_expressed <- specific_probes[specific_probes[,Expression_T1_D3] >= intensity_threshold
                              & specific_probes[,Expression_T2_D3] >= intensity_threshold,]
```

After filtering, 35933 probes showed significant expression at least one of the three time points.

4.3. P-value filtering

Herein below, only probes that showed significant regulation between the two treatments in at least one of the three dates will be considered (p-value < 0.01).

```
# Application of the filter to the 3 dates
D1 <- D1_expressed[D1_expressed[, Pval_D1] <= pval_threshold,]
D2 <- D2_expressed[D2_expressed[, Pval_D2] <= pval_threshold,]
D3 <- D3_expressed[D3_expressed[, Pval_D3] <= pval_threshold,]
```

After p-value filtering, a total 2383 unique probes showed significant regulation for at least one of the three time points of the experiment, including 153, 1456 and 1148 probes for 28, 48 and 119 DAFB, respectively.

4.4. Log2 ratio filtering ~

The log ratio filtering enables to select only transcripts having a sufficient difference of expression between the two treatments which have biological meaning. In this analysis, only probes that show differential expression lower than -1 for up-regulated genes in 'ON' trees and higher than 1 for up-regulated genes in 'OFF' trees, will be considered within the previously selected probes.

```
# Application of the filter to the 3 dates and the two treatments
D1_ON_up <- D1[D1[, Ratio_D1] <= ratio_threshold_down,]
D1_OFF_up <- D1[D1[, Ratio_D1] >= ratio_threshold_up,]
D2_ON_up <- D2[D2[, Ratio_D2] <= ratio_threshold_down,]
D2_OFF_up <- D2[D2[, Ratio_D2] >= ratio_threshold_up,]
D3_ON_up <- D3[D3[, Ratio_D3] <= ratio_threshold_down,]
D3_OFF_up <- D3[D3[, Ratio_D3] >= ratio_threshold_up,]

# Bind results of ON and OFF trees to get the total per date
D1_ON_OFF <- unique(rbind(D1_ON_up,D1_OFF_up))
D2_ON_OFF <- unique(rbind(D2_ON_up,D2_OFF_up))
D3_ON_OFF <- unique(rbind(D3_ON_up,D3_OFF_up))

# Make a data frame containing transcripts that were differentially expressed at least one time
genes_modulated_at_least_one_time <- unique(rbind(D1_ON_OFF,D2_ON_OFF,D3_ON_OFF))
```

After log ratio filtering, a total of 648 unique probes will be considered for further analyses, including 67, 244 and 436 probes for 28, 48 and 119 DAFB, respectively.

4.5. Selected gene-set

A total of 648 genes were identified as statistically significant for at least one of the three time points, representing 1.0283919% of the sense probes present in the microarray. The number of up-regulated genes in both 'ON' (ratio < -1) and 'OFF' trees (ratio > 1) increased through time with a higher number of up-regulated genes in 'ON' trees than in 'OFF' trees (Tab.1 and Fig. 5). Among these genes, 32 were up-regulated in 'ON' trees at 28 DAFB whereas 35 in 'OFF' trees. At 48 DAFB, 152 and 92 genes were up-regulated in 'ON' and 'OFF' trees, respectively. At 119 DAFB, 254 genes were up-regulated in 'ON' trees while 182 in 'OFF' trees. Some of Arabidopsis gene accessions were found several times in the data-set, indicating that duplicated homeologous genes were present in the set of differentially expressed transcripts.

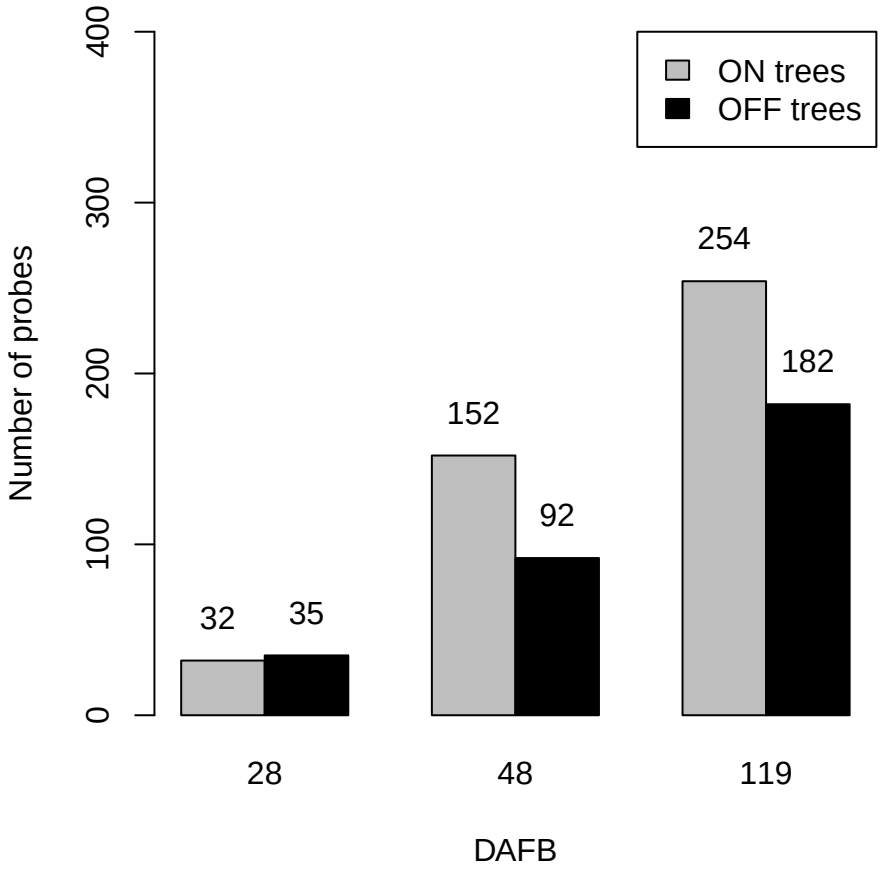
The following table and barplot give the number of probes to consider for further analysis per date and per treatment.

Treatment	Date	Nb_probes
ON	D1	32
OFF	D1	35
Total	D1	67
ON	D2	152
OFF	D2	92

Treatment	Date	Nb_probes
Total	D2	244
ON	D3	254
OFF	D3	182
Total	D3	436

Tab.1. Number of genes significantly up-regulated in each treatment and each date.

Fig.5. Number of genes significantly up−regulated in each treatment and each date



5. Graphic representation after filtering

5.1. Probe distribution

Fig.6. Distribution of probe density in relation to expression intensity of selected probes

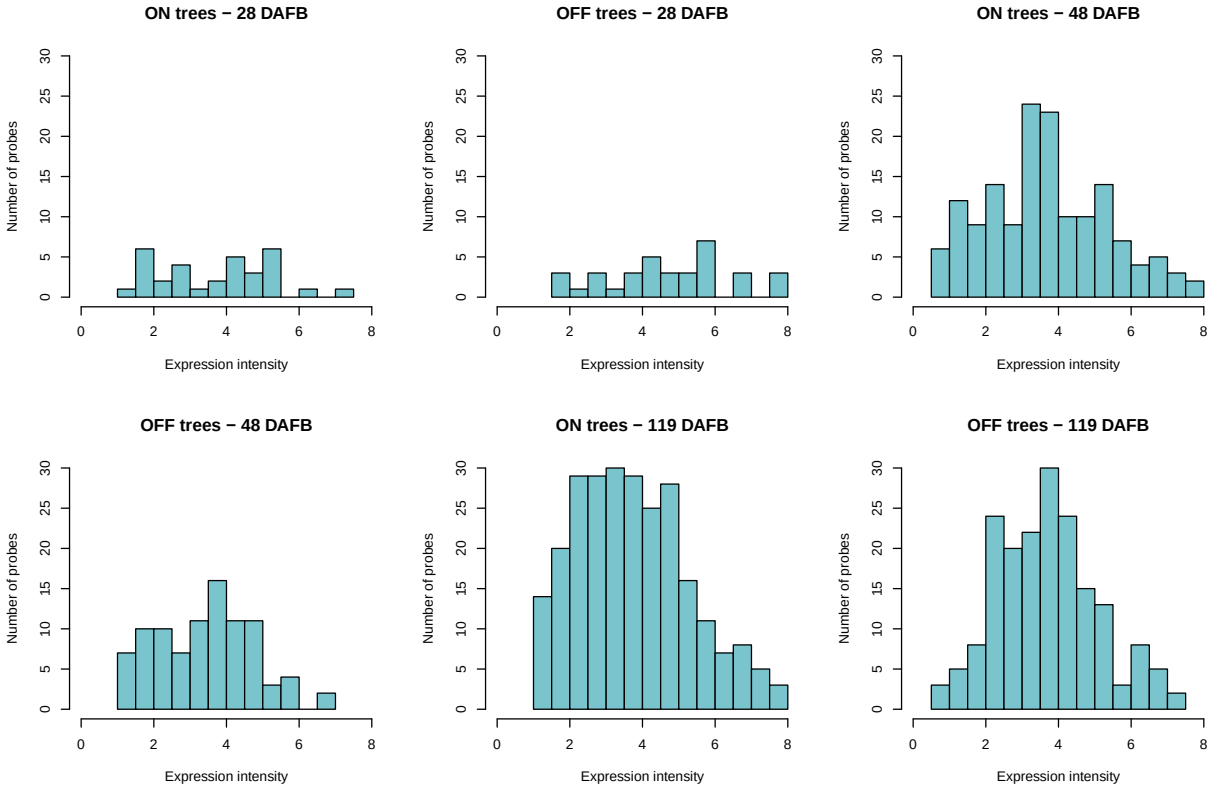
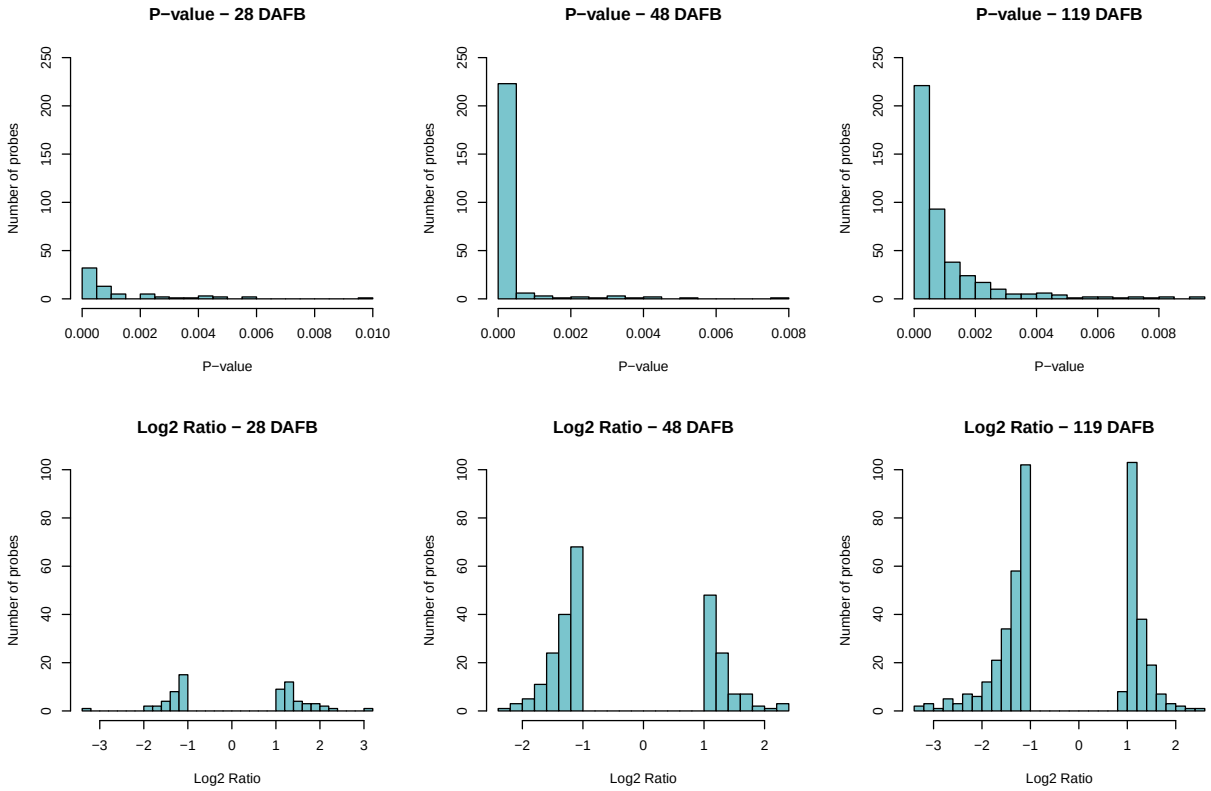


Fig.7. Distribution of probe density in relation to p−value and log2 ratio of selected probes



5.2. Venn Diagram

All comparisons were made within 'OFF' trees and 'ON' trees at each time point. Using venn diagram we are representing transcripts significantly regulated at each time point (28, 48 and 119 DAFB) and jointly over time points. For this, the number of transcript common over time points first is determined. Then, venn diagrams are represented using these groups and the function "draw.triple.venn" (package "VennDiagram").

```
# Merge matrixes of probes up-regulated in 'OFF' trees to found common genes over dates
D12_OFF_up <- merge(x= D1_OFF_up, y= D2_OFF_up, by= "genes_seq_id")
D13_OFF_up <- merge(x= D1_OFF_up, y= D3_OFF_up, by= "genes_seq_id")
D23_OFF_up <- merge(x= D2_OFF_up, y= D3_OFF_up, by= "genes_seq_id")
D123_OFF_up <- merge(x= D12_OFF_up, y= D3_OFF_up, by= "genes_seq_id")

# Merge matrixes of probes up-regulated in 'ON' trees to found common genes over dates
D12_ON_up <- merge(x= D1_ON_up, y= D2_ON_up, by= "genes_seq_id")
D13_ON_up <- merge(x= D1_ON_up, y= D3_ON_up, by= "genes_seq_id")
D23_ON_up <- merge(x= D2_ON_up, y= D3_ON_up, by= "genes_seq_id")
D123_ON_up <- merge(x= D12_ON_up, y= D3_ON_up, by= "genes_seq_id")
```

Fig.8. Generalized Venn diagram with three sets of 28 (grey), 48 (red), and 119 (blue) DAFB and their intersections for up-regulated genes in 'ON' trees ~

```
par(mfrow=c(1,1), pty='m',oma = c(0, 0, 3, 0))
# Venndiagram for up-regulated genes in ON trees
venn.plot_down <- draw.triple.venn(
  area1 = length(unique(D1_ON_up$genes_seq_id)),
  area2 = length(unique(D2_ON_up$genes_seq_id)),
  area3 = length(unique(D3_ON_up$genes_seq_id)),
  n12 = length(unique(D12_ON_up$genes_seq_id)),
  n23 = length(unique(D23_ON_up$genes_seq_id)),
  n13 = length(unique(D13_ON_up$genes_seq_id)),
  n123 = length(unique(D123_ON_up$genes_seq_id)),
  category = c("28 DAFB", "48 DAFB", "119 DAFB"),
  fill = c("#899DA4", "#C93312", "cornflowerblue"),
  col = c("#899DA4", "#C93312", "cornflowerblue"),
  lty = "dashed", lwd = 1,
  alpha = 0.8,
  cex = 2,
  cat.cex = 1.2,
  cat.col = c("#899DA4", "#C93312", "cornflowerblue"));
```

Fig.9. Generalized Venn diagram with three sets of 28 (grey), 48 (red), and 119 (blue) DAFB and their intersections for up-regulated genes in 'OFF' trees. ~

```
# Venndiagram for up-regulated genes in OFF trees
grid.newpage();
venn.plot_up <- draw.triple.venn(
  area1 = length(unique(D1_OFF_up$genes_seq_id)),
  area2 = length(unique(D2_OFF_up$genes_seq_id)),
  area3 = length(unique(D3_OFF_up$genes_seq_id)),
  n12 = length(unique(D12_OFF_up$genes_seq_id)),
  n23 = length(unique(D23_OFF_up$genes_seq_id)),
  n13 = length(unique(D13_OFF_up$genes_seq_id)),
  n123 = length(unique(D123_OFF_up$genes_seq_id)),
  category = c("28 DAFB", "48 DAFB", "119 DAFB"),
  fill = c("#899DA4", "#C93312", "cornflowerblue"),
  scaled = TRUE, euler.d = TRUE,
  lwd = 1, col = c("#899DA4", "#C93312", "cornflowerblue"),
  lty = "dashed",
  alpha = 0.8,
  rotation.degree = 0,
  cex = 2,
  cat.cex = 1.2,
  cat.col = c("#899DA4", "#C93312", "cornflowerblue"));
```

6. Cluster analysis by k-means

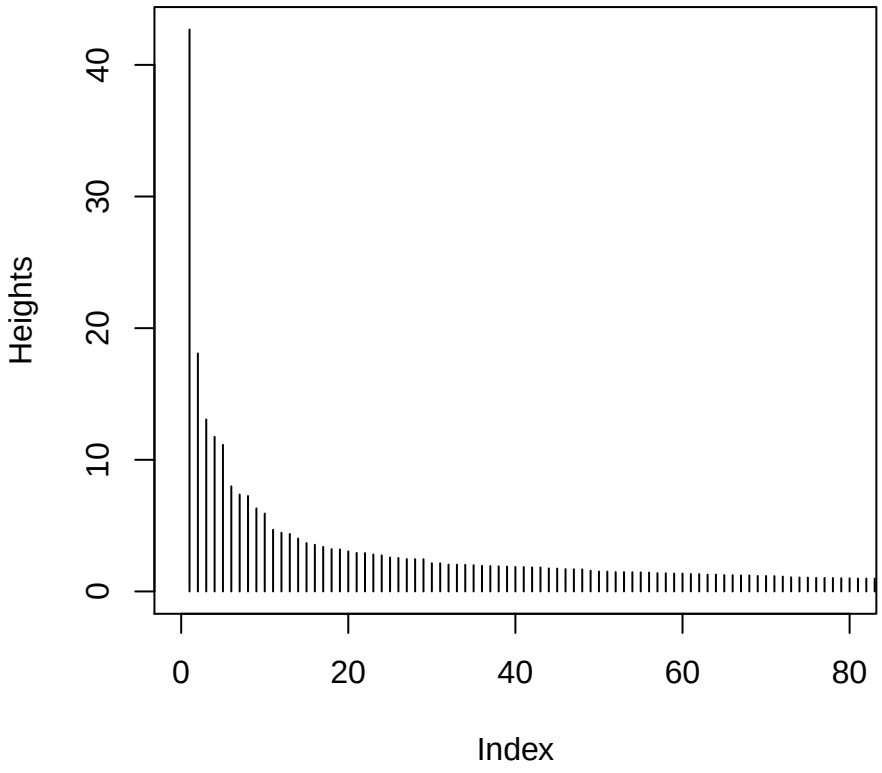
Transcripts that significantly shared similar profiles were visualized using a hierarchical ascendant classification analysis and were grouped into clusters according to their log2 ratio using k-means.

After computing agglomerative hierarchical clustering of the dataset using the agnes function (package "cluster"), heights of the clustering are used to choose judiciously the number of clusters.

6.1. Hierarchical ascendant classification

```
# Convert the content of the "AT_id" column as character
genes_modulated_at_least_one_time[,AT_id] <-
  as.character(genes_modulated_at_least_one_time[,AT_id])

# Make a data frame with Malus gene ID, Arabidopsis gene ID and the ratio for the three dates
cluster_data<-cbind.data.frame(
         "gene_id" = genes_modulated_at_least_one_time[,gene_id],
         "AT_id" = genes_modulated_at_least_one_time[,AT_id],
         "R1" = genes_modulated_at_least_one_time[,Ratio_D1],
         "R2" = genes_modulated_at_least_one_time[,Ratio_D2],
         "R3" = genes_modulated_at_least_one_time[,Ratio_D3])

# Computes agglomerative hierarchical clustering using the agnes function (library "cluster")
classif <- agnes(cluster_data[,3:5], method = "ward", metric = "pearson")
# Converts objects from other hierarchical clustering functions to class "hclust"
classif1 <- as.hclust(classif)
# Plot heights
plot(rev(classif1$height), type = "h", ylab = "Heights", xlim = c(0,80),
     main = "Fig.10. Heights of the hierarchical ascendant classification", cex.main = 0.8)
```

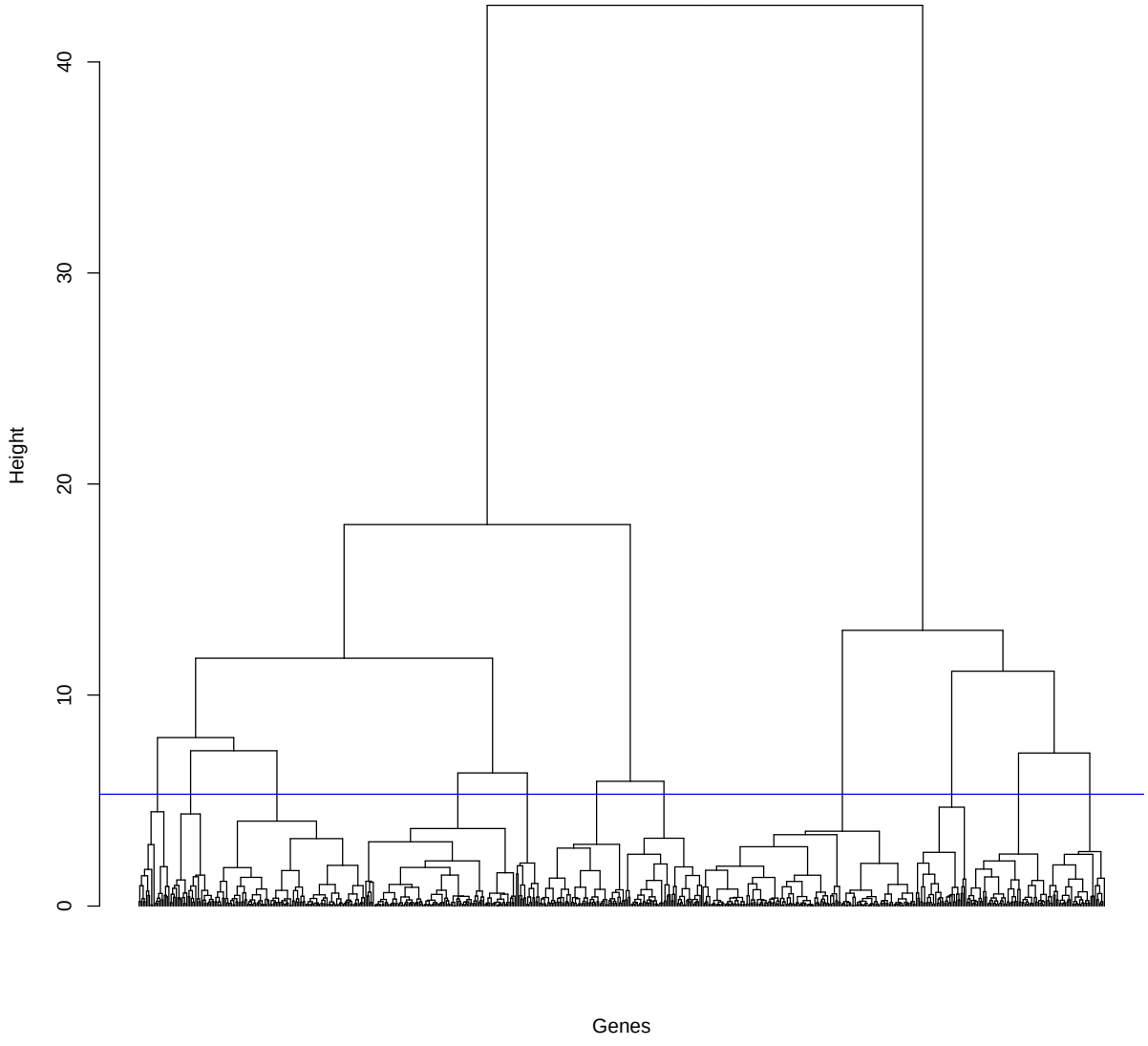
Fig.10. Heights of the hierarchical ascendant classification



Breaks between successive bars indicate the number of group to form. Since we are interested in groups with sufficient size and genes with similar profile, we are choosing to form 11 groups because of the break between the 10th and the 11th bar.

```
# Definition of the number of clusters
nb_groups <- 11

# Plot the tree
classif2 <- as.dendrogram(classif1)
plot(classif2, leaflab = "none", xlab = "Genes", ylab = "Height", main = "Fig.11. Cluster dendogram of
abline(h = 5.3, col = "blue")
```

Fig.11. Cluster dendogram of transcripts with statistically significant differences in expression. Blue line represents where the tree has been cut to get 11 clusters



```
# Cut the tree in the number of pre-defined groups
classes <- cutree(classif, k = nb_groups)
```

6.2. K-means clustering

After determining the number of clusters using hierarchical ascendant classification, transcripts are grouped into clusters using k-means clustering.

```
# Perform k-means clustering
classe.kmeans <- kmeans(cluster_data[,3:5], centers = nb_groups)

# Paste the cluster number to the data frame
cluster_data.comp <- cbind.data.frame(cluster_data, as.factor(classe.kmeans$cluster))
colnames(cluster_data.comp)[6] <- "Group"
```

6.3. Cluster characterisation

After grouping genes in clusters, clusters are described by their center value.

Tab.2. Centre value of each cluster identified by k-means. ~

```
# The following table gives the average ratio value per group
kable(classe.kmeans$centers, row.names=TRUE)
```

	R1	R2	R3
1	-0.6160526	-0.9271930	-1.1433333
2	-0.2121739	-1.2671739	-0.1148913
3	2.0933333	1.5250000	0.5083333
4	0.2638462	1.2280769	1.1040385
5	-0.0748571	0.0225714	1.1787143
6	0.0066667	0.5311111	1.7966667
7	0.2052542	0.4559322	1.1055932
8	-0.3062500	-0.7387500	-2.4303125
9	-0.0226277	-0.0037956	-1.2878832
10	0.0794872	1.3025641	0.0835897
11	1.3500000	0.2710345	0.1189655

Log2 ratio values for each transcript in each cluster are then graphically represented to appreciate the uniformity of the formed groups.

```
# Make a data frame including ratio, group and DAFB for interaction plot
D_1<-cbind.data.frame("genes_seq_id"=cluster_data.comp[,1],
                      "Group"=cluster_data.comp[,6],
                     "Ratio"=cluster_data.comp[,3], "DAFB"=28)
```

```
D_2<-cbind.data.frame("genes_seq_id"=cluster_data.comp[,1],
                      "Group"=cluster_data.comp[,6],
                     "Ratio"=cluster_data.comp[,4], "DAFB"=48)
D_3<-cbind.data.frame("genes_seq_id"=cluster_data.comp[,1],
                      "Group"=cluster_data.comp[,6],
                     "Ratio"=cluster_data.comp[,5], "DAFB"=119)
# Bind the data frames
cluster_data_interaction_plot <- rbind(D_1,D_2,D_3)
cluster_data_interaction_plot <- cluster_data_interaction_plot[order(
  cluster_data_interaction_plot$Group, decreasing = FALSE), ]

# Plot log2 ratio profil for each transcript within its respective cluster
par(mfrow=c(4,3))
uniq <- unique(unlist(cluster_data_interaction_plot$Group))
for (i in 1:length(uniq)){
    Group_i <- subset(cluster_data_interaction_plot, Group == uniq[i])
    with(Group_i, interaction.plot(DAFB, genes_seq_id, Ratio, legend=F,
          main=paste("Group",uniq[i] ," - N=",nrow(subset(cluster_data.comp,
          Group == uniq[i]))), lwd=1, ylim=c(-4,4), ylab="Log2 ratio"))
    abline(h=0,col=1,lty=4)
    }
```

Fig.12. Cluster analysis of transcripts with statistically significant differences in expression. Clusters contain the representations of log2 ratio value for each transcript. Transcripts were clustered in a user-defined number of 11 clusters using the k-means algorithm. Clusters are numbered 1 to 11, and the total number of transcripts per set is recorded (N). For each diagram, the x-axis shows the three time points (28, 48, and 119 DAFB), while the y-axis corresponds to fold change in the set of transcripts.

7. Exportation of transcript lists

7.1. List of Arabidopsis accessions homologous to differentially expressed transcripts

Different lists of Arabidopsis accessions homologous to differentially expressed transcripts are exported in text files. These lists can be then used as input for single enrichment analyses (AgriGO, http://bioinfo.cau.edu.cn/agriGO/) or for gene network investigation (STRING 9.1., http://string-db.org/). The entire set of differentially expressed transcripts at least one of the three time points (648 transcripts) is divided in different sub-groups (by time point, by treatment, by cluster) which will enable to perform annotation studies at different levels.

7.1.1. List of Arabidopsis accessions per time point (28, 48 and 119 DAFB) for transcripts significantly regulated ~

```
write.table(as.data.frame(sort(D1_ON_OFF[, AT_id])),file = "AT_id_date_1.txt", sep="\t",
            quote = F, row.names = FALSE, col.names=FALSE, na = "NA")
write.table(as.data.frame(sort(D2_ON_OFF[, AT_id])),file = "AT_id_date_2.txt", sep="\t",
            quote = F, row.names = FALSE, col.names=FALSE, na = "NA")
write.table(as.data.frame(sort(D3_ON_OFF[, AT_id])),file = "AT_id_date_3.txt", sep="\t",
            quote = F, row.names = FALSE, col.names=FALSE, na = "NA")
```

7.1.2. List of Arabidopsis accessions for each time point for transcripts up-regulated in 'OFF' trees ~

```
write.table(as.data.frame(sort(D12_OFF_up$AT_Id.x)),file = "AT_id_date_12_OFF.txt",
            sep = "\t", quote = F, row.names = FALSE, col.names = FALSE, na = "NA")
write.table(as.data.frame(sort(D13_OFF_up$AT_Id.x)),file = "AT_id_date_13_OFF.txt",
            sep = "\t", quote = F, row.names = FALSE, col.names = FALSE, na = "")
write.table(as.data.frame(sort(D23_OFF_up$AT_Id.x)),file = "AT_id_date_23_OFF.txt",
            sep = "\t", quote = F, row.names = FALSE, col.names = FALSE, na = "")
write.table(as.data.frame(sort(D123_OFF_up$AT_Id.x)),file = "AT_id_date_123_OFF.txt",
            sep = "\t", quote = F, row.names = FALSE, col.names = FALSE, na = "")
```

7.1.3. List of Arabidopsis accessions for each time point for transcripts up-regulated in 'ON' trees ~

```
write.table(as.data.frame(sort(D12_ON_up$AT_Id.x)),file = "AT_id_date_12_ON.txt",
            sep = "\t", quote = F, row.names = FALSE, col.names = FALSE)
write.table(as.data.frame(sort(D13_ON_up$AT_Id.x)),file = "AT_id_date_13_ON.txt",
            sep = "\t", quote = F, row.names = FALSE, col.names = FALSE)
write.table(as.data.frame(sort(D23_ON_up$AT_Id.x)),file = "AT_id_date_23_ON.txt",
            sep = "\t", quote = F, row.names = FALSE, col.names = FALSE)
write.table(as.data.frame(sort(D123_ON_up$AT_Id.x)),file = "AT_id_date_123_ON.txt",
            sep = "\t", quote = F, row.names = FALSE, col.names = FALSE)
```

7.1.4. List of Arabidopsis accessions for each group of the clustering analysis ~

```
uniq <- unique(unlist(cluster_data.comp$Group))
for (i in 1:length(uniq)){
    Group_i <- subset(cluster_data.comp, Group == uniq[i])
  Group_i$AT_id <- as.character(Group_i$AT_id)
  out_i <- as.data.frame(sort(Group_i$AT_id))
write.table(out_i,file = paste("AT_id_cluster_", uniq[i], ".txt"), sep = "\t", quote = F,
            row.names = FALSE, col.names = FALSE)
}
```

7.2. Matrices of significantly differentially expressed transcripts

After the filtering procedure, the matrix containing only significantly differentially expressed transcripts is exported.

```
# Export the matrix of the differentially expressed transcripts
write.table(genes_modulated_at_least_one_time, file = "genes_modulated_at_least_one_time.txt",
            sep = "\t", quote = F, row.names = FALSE)
# Export the matrices of the differentially expressed transcripts per time point
write.table(as.data.frame(D1_ON_OFF),file = "Date_1_matrix.txt", sep = "\t", quote = F,
            row.names = FALSE, col.names = TRUE, na = "NA")
write.table(as.data.frame(D2_ON_OFF),file = "Date_2_matrix.txt", sep = "\t", quote = F,
            row.names = FALSE, col.names = TRUE, na = "NA")
write.table(as.data.frame(D3_ON_OFF),file = "Date_3_matrix.txt", sep = "\t", quote = F,
            row.names = FALSE, col.names = TRUE, na = "NA")
```

8. Duplicated genes in the apple genome

Of the 648 transcripts differentially expressed in our experiment, 535 were significantly homologous to an Arabidopsis gene, while the 113 remaining transcripts did not share significant homology with any Arabidopsis gene. The 535 annotated apple transcripts corresponded to 426 unique Arabidopsis genes, due to the recent whole genome duplication that occurred in the apple genome (Velasco *et al.*, 2010). These transcripts are of particular interest since their expression is altered by the treatment which is confirmed on two or several transcripts. These transcripts can be studied in detail.

8.1. Duplicated genes at 28 DAFB

```
# Set variables as character
D1_ON_OFF$AT_Id <- as.character(D1_ON_OFF$AT_Id)
D1_ON_OFF$Molecular_Function <- as.character(D1_ON_OFF$Molecular_Function)
# Select the 30 first characters of the AT_function
D1_ON_OFF$At_Function <- as.character(substr(D1_ON_OFF$At_Function, 1, 30))

# Select only probes with an Arabidopsis accession
D1_ON_OFF_1 <- subset(x = D1_ON_OFF, AT_Id != "NA")

# Count the number of MDP per Arabidopsis accession
Freq_D1 <- as.data.frame(table(D1_ON_OFF_1$AT_Id))
colnames(Freq_D1) <- c("AT_Id", "Freq")
# Paste the number of MDP per Arabidopsis accession to the data frame
D1_ON_OFF_comp <- merge(x = D1_ON_OFF_1, y = Freq_D1, by = "AT_Id")

# Make a table
D1_AT_Id <- as.data.frame(cbind(D1_ON_OFF_comp$AT_Id, D1_ON_OFF_comp$Freq,
            D1_ON_OFF_comp$At_Function, D1_ON_OFF_comp$Molecular_Function))
colnames(D1_AT_Id) <- c("AT_Id", "Nb MDP", "At_Function","Molecular_Function")
D1_AT_Id <- D1_AT_Id[order(D1_AT_Id[,"Nb MDP"],decreasing =T),]
# Remove duplicated rows
D1_AT_Id <- unique.data.frame(D1_AT_Id)
# Edit the table
kable(D1_AT_Id[1:7,], row.names=FALSE)
```

AT_Id	Nb MDP	At_Function	Molecular_Function
AT1G23740	5	oxidoreductase, zinc-binding d	metabolic process
AT2G44130	2	.	.
AT3G23240	2	ATERF1/ERF1 (ETHYLENE RESPONSE	regulation of transcription, DNA-dependent
AT3G47340	2	ASN1 (DARK INDUCIBLE 6)	asparagine biosynthetic process
AT3G52740	2	unknown protein	biological process unknown
AT4G35770	2	SEN1 (DARK INDUCIBLE 1)	aging
AT5G48850	2	male sterility MS5 family prot	biological process unknown

8.2. Duplicated genes at 48 DAFB

```
# Set variables as character
D2_ON_OFF$AT_Id <- as.character(D2_ON_OFF$AT_Id)
D2_ON_OFF$Molecular_Function <- as.character(D2_ON_OFF$Molecular_Function)
# Select the 30 first characters of the AT_function
D2_ON_OFF$At_Function <- as.character(substr(D2_ON_OFF$At_Function, 1, 30))

# Select only probes with an Arabidopsis accession
D2_ON_OFF_1 <- subset(x = D2_ON_OFF, AT_Id != "NA")

# Count the number of MDP per Arabidopsis accession
Freq_D2 <- as.data.frame(table(D2_ON_OFF_1$AT_Id))
colnames(Freq_D2) <- c("AT_Id", "Freq")
# Paste the number of MDP per Arabidopsis accession to the data frame
D2_ON_OFF_comp <- merge(x = D2_ON_OFF_1, y = Freq_D2, by = "AT_Id")

# Make a table
D2_AT_Id <- as.data.frame(cbind(D2_ON_OFF_comp$AT_Id, D2_ON_OFF_comp$Freq,
             D2_ON_OFF_comp$At_Function, D2_ON_OFF_comp$Molecular_Function))
colnames(D2_AT_Id) <- c("AT_Id", "Nb MDP", "At_Function","Molecular_Function")
D2_AT_Id <- D2_AT_Id[order(D2_AT_Id[,"Nb MDP"],decreasing =T),]
# Remove duplicated rows
D2_AT_Id <- unique.data.frame(D2_AT_Id)
kable(D2_AT_Id[1:35,], row.names=FALSE)
```

AT_Id	Nb MDP	At_Function	Molecular_Function
AT4G37870	4	phosphoenolpyruvate carboxykin	gluconeogenesis
AT1G24020	3	Bet v I allergen family protei	defense response
AT2G29500	3	17.6 kDa class I small heat sh	response to oxidative stress
AT2G36640	3	ATECP63 (EMBRYONIC CELL PROTEI	embryonic development ending in seed dormancy
AT3G15850	3	FAD5 (FATTY ACID DESATURASE 5)	lipid metabolic process
AT3G51895	3	SULTR3;1 (SULFATE TRANSPORTER	transport
AT4G16730	3	lyase/ magnesium ion binding	metabolic process
AT4G32480	3	unknown protein	biological process unknown
AT5G15630	3	COBL4/IRX6 (COBRA-LIKE4); hydr	cellulose and pectin-containing secondary cell wall b
AT5G59720	3	HSP18.2 (HEAT SHOCK PROTEIN 18	response to heat
AT1G53430	2	leucine-rich repeat family pro	protein amino acid phosphorylation
AT1G73040	2	jacalin lectin family protein	biological process unknown
AT1G77380	2	AAP3 (amino acid permease 3);	riboflavin biosynthetic process
AT2G16060	2	AHB1 (ARABIDOPSIS HEMOGLOBIN 1	oxygen transport
AT2G24430	2	no apical meristem (NAM) famil	multicellular organismal development
AT2G28315	2	transporter-related	biological process unknown
AT2G45510	2	CYP704A2 (cytochrome P450, fam	electron transport
AT3G12720	2	AtMYB67/AtY53 (myb domain prot	regulation of transcription, DNA-dependent

AT_Id	Nb MDP	At_Function	Molecular_Function
AT3G19270	2	CYP707A4 (cytochrome P450, fam	electron transport
AT3G47340	2	ASN1 (DARK INDUCIBLE 6)	asparagine biosynthetic process
AT3G48280	2	CYP71A25 (cytochrome P450, fam	electron transport
AT3G49260	2	IQD21 (IQ-domain 21); calmodul	biological process unknown
AT3G54040	2	photoassimilate-responsive pro	biological process unknown
AT3G63110	2	ATIPT3 (Arabidopsis thaliana i	tRNA processing
AT4G21200	2	ATGA2OX8 (GIBBERELLIN 2-OXIDAS	biological process unknown
AT4G27670	2	HSP21 (HEAT SHOCK PROTEIN 21)	response to heat
AT4G35160	2	O-methyltransferase family 2 p	RNA splicing
AT4G35770	2	SEN1 (DARK INDUCIBLE 1)	aging
AT5G05340	2	peroxidase, putative	electron transport
AT5G12020	2	17.6 kDa class II heat shock p	response to heat
AT5G41040	2	transferase family protein	biological process unknown
AT5G42650	2	AOS (ALLENE OXIDE SYNTHASE); h	electron transport
AT5G44030	2	CESA4 (CELLULOSE SYNTHASE 4);	cellulose and pectin-containing cell wall biogenesis
AT5G53390	2	unknown protein	biological process unknown
AT5G60490	2	FLA12 (fasciclin-like arabinog	protein targeting to vacuole

8.3. Duplicated genes at 119 DAFB

```
# Set variables as character
D3_ON_OFF$AT_Id <- as.character(D3_ON_OFF$AT_Id)
D3_ON_OFF$Molecular_Function <- as.character(D3_ON_OFF$Molecular_Function)
# Select the 30 first characters of the AT_function
D3_ON_OFF$At_Function <- as.character(substr(D3_ON_OFF$At_Function, 1, 30))

# Select only probes with an Arabidopsis accession
D3_ON_OFF_1 <- subset(x = D3_ON_OFF, AT_Id != "NA")

# Count the number of MDP per Arabidopsis accession
Freq_D3 <- as.data.frame(table(D3_ON_OFF_1$AT_Id))
colnames(Freq_D3) <- c("AT_Id", "Freq")
# Paste the number of MDP per Arabidopsis accession to the data frame
D3_ON_OFF_comp <- merge(x = D3_ON_OFF_1, y = Freq_D3, by = "AT_Id")

# Make a table
D3_AT_Id <- as.data.frame(cbind(D3_ON_OFF_comp$AT_Id, D3_ON_OFF_comp$Freq,
                D3_ON_OFF_comp$At_Function, D3_ON_OFF_comp$Molecular_Function))
colnames(D3_AT_Id) <- c("AT_Id", "Nb MDP", "At_Function","Molecular_Function")
D3_AT_Id <- D3_AT_Id[order(D3_AT_Id[,"Nb MDP"],decreasing =T),]
# Remove duplicated rows
D3_AT_Id <- unique.data.frame(D3_AT_Id)
kable(D3_AT_Id[1:56,], row.names=FALSE)
```

AT_Id	Nb MDP	At_Function	Molecular_Function
AT1G52800	3	oxidoreductase, 2OG-Fe(II) oxy	carbohydrate biosynthetic process
AT2G47180	3	ATGOLS1 (ARABIDOPSIS THALIANA	response to heat
AT3G16150	3	L-asparaginase, putative / L-a	glycoprotein catabolic process
AT3G25400	3	unknown protein	biological process unknown
AT3G51895	3	SULTR3;1 (SULFATE TRANSPORTER	transport
AT5G08020	3	replication protein, putative	DNA replication
AT5G15630	3	COBL4/IRX6 (COBRA-LIKE4); hydr	cellulose and pectin-containing secondary cell wall
AT5G59845	3	gibberellin-regulated family p	response to gibberellin stimulus
AT1G02610	2	zinc finger (C3HC4-type RING f	biological process unknown
AT1G04560	2	AWPM-19-like membrane family p	biological process unknown
AT1G10670	2	ACLA-1 (ATP-citrate lyase A-1)	biological process unknown
AT1G16070	2	AtTLP8 (TUBBY LIKE PROTEIN 8)	regulation of transcription
AT1G21230	2	WAK5 (WALL ASSOCIATED KINASE 5	protein amino acid phosphorylation
AT1G22170	2	phosphoglycerate/bisphosphogly	glycolysis
AT1G22410	2	2-dehydro-3-deoxyphosphohepton	aromatic amino acid family biosynthetic process
AT1G54070	2	dormancy/auxin associated prot	biological process unknown
AT1G60470	2	ATGOLS4 (ARABIDOPSIS THALIANA	carbohydrate biosynthetic process
AT1G66920	2	serine/threonine protein kinas	protein amino acid phosphorylation
AT1G73040	2	jacalin lectin family protein	biological process unknown
AT1G78440	2	ATGA2OX1 (GIBBERELLIN 2-OXIDAS	gibberellin catabolic process
AT2G15890	2	MEE14 (maternal effect embryo	embryonic development ending in seed dormancy
AT2G23340	2	AP2 domain-containing transcri	regulation of transcription, DNA-dependent
AT2G23970	2	defense-related protein, putat	glutamine metabolic process
AT2G28315	2	transporter-related	biological process unknown
AT2G37330	2	ALS3 (ALUMINUM SENSITIVE 3), R	biological process unknown
AT2G39420	2	esterase/lipase/thioesterase f	biological process unknown
AT2G42200	2	squamosa promoter-binding prot	regulation of transcription
AT2G44130	2	.	.
AT3G05620	2	pectinesterase family protein	cell wall modification
AT3G14740	2	PHD finger family protein	ubiquitin-dependent protein catabolic process
AT3G22740	2	HMT3 (Homocysteine S-methyltra	biological process unknown
AT3G23240	2	ATERF1/ERF1 (ETHYLENE RESPONSE	regulation of transcription, DNA-dependent
AT3G27060	2	TSO2 (TSO2); ribonucleoside-di	DNA replication
AT3G28857	2	transcription regulator	regulation of transcription
AT3G47340	2	ASN1 (DARK INDUCIBLE 6)	asparagine biosynthetic process

AT_Id	Nb MDP	At_Function	Molecular_Function
AT3G48280	2	CYP71A25 (cytochrome P450, fam	electron transport
AT3G54040	2	photoassimilate-responsive pro	biological process unknown
AT3G58070	2	GIS (GLABROUS INFLORESCENCE ST	response to gibberellin stimulus
AT3G63110	2	ATIPT3 (Arabidopsis thaliana i	tRNA processing
AT4G16730	2	lyase/ magnesium ion binding	metabolic process
AT4G27450	2	unknown protein	biological process unknown
AT4G32480	2	unknown protein	biological process unknown
AT4G35770	2	SEN1 (DARK INDUCIBLE 1)	aging
AT5G04530	2	beta-ketoacyl-CoA synthase fam	metabolic process
AT5G06800	2	myb family transcription facto	regulation of transcription
AT5G15310	2	AtMIXTA/AtMYB16 (myb domain pr	cell morphogenesis
AT5G16250	2	unknown protein	biological process unknown
AT5G16490	2	RIC4 (ROP-INTERACTIVE CRIB MOT	metabolic process
AT5G17420	2	IRX3 (IRREGULAR XYLEM 3, MURUS	cellulose and pectin-containing secondary cell wall
AT5G27670	2	histone H2A, putative	nucleosome assembly
AT5G41761	2	unknown protein	biological process unknown
AT5G44030	2	CESA4 (CELLULOSE SYNTHASE 4);	cellulose and pectin-containing cell wall biogenesis
AT5G54840	2	GTP-binding family protein	small GTPase mediated signal transduction
AT5G55050	2	GDSL-motif lipase/hydrolase fa	lipid metabolic process
AT5G59590	2	UDP-glucoronosyl/UDP-glucosyl	metabolic process
AT5G60490	2	FLA12 (fasciclin-like arabinog	protein targeting to vacuole

9. Genomic position of genes differentially expressed

The best hit reports of blastp of Malus x domestica genome v1.0 proteins file (Malus_x_domestica.v1.0_gene_pep_function_101210.formated.xls) made available by the Genome Database for Rosaceae (GDR, November 2014 , http://www.rosaceae.org/) includes genome position of predicted genes. This file is used to retrieve genomic position of genes that showed significant differential expression for at least one of the three dates.

```
# Before to be read into R, the original file
  # (Malus_x_domestica.v1.0_gene_pep_function_101210.formated.xls)
  # has been formatted and saved in a text file.

# Read data file
gene_positions <- read.table("Malus_x_domestica_v1_0_gene_pep_function_101210.txt",
                  header=T)

# Add the position information (chromosome, transcript start and stop position) to the
  # matrix of genes that showed significant differential expression for at least one of
  # the three dates.
position_genes_modulated_at_least_one_time <- merge(x = gene_positions[,1:4],
                    y = genes_modulated_at_least_one_time, by = "genes_seq_id")
```

```
  # Declare variables as character.
      position_genes_modulated_at_least_one_time$genes_seq_id <- as.character(
        position_genes_modulated_at_least_one_time$genes_seq_id)
      position_genes_modulated_at_least_one_time$Scaffold_ID <- as.character(
        position_genes_modulated_at_least_one_time$Scaffold_ID)

# Save the data frame in a text file.
    write.table(position_genes_modulated_at_least_one_time,
                file = "position_genes_modulated_at_least_one_time.txt",
                sep = "\t", quote=FALSE, row.names=FALSE)
```

Position of SSR markers of the apple segregating population, 'Starkrimson' x 'Granny Smith' (STK x GS), genetic map (Guitton *et al.*, 2011) were retrieved by either blasting the marker sequence (primer or amplicon) available on Hidras database (Hidras, November 2014 , http://www.hidras.unimi.it/index.php), or by using the Genome Database for Rosaceae (GDR, November 2014, http://www.rosaceae.org/, file Malus_x_domestica.v1.0.markers.xls). For SNP markers, the position of the gene prediction (MDP) used to design the marker (Guitton *et al.*, 2011) was considered as the genomic position of the marker.

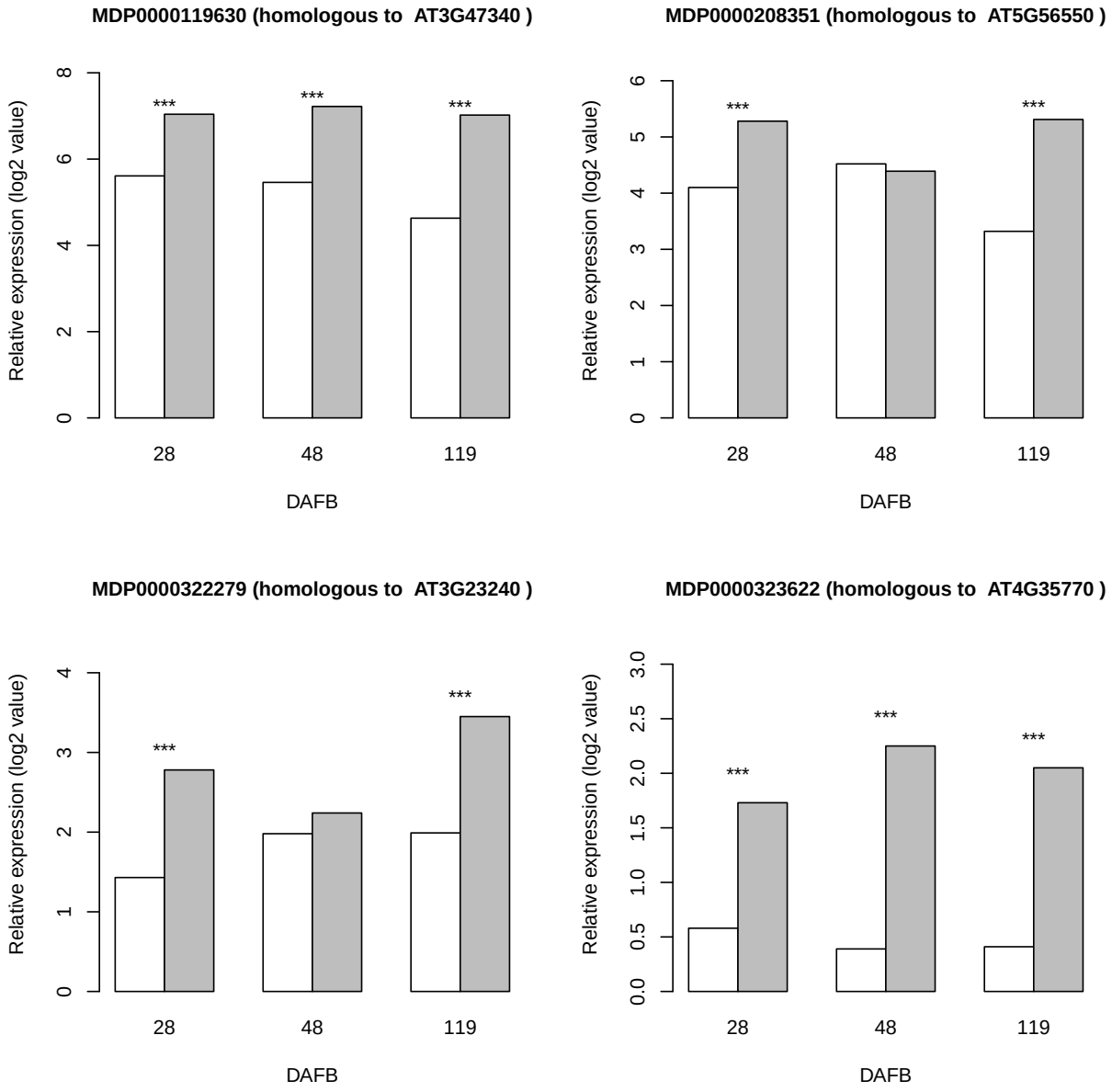
A file containing genomic position of the genetic markers of the STK x GS map has been thus generated and is now used to compare position of genes with QTL.

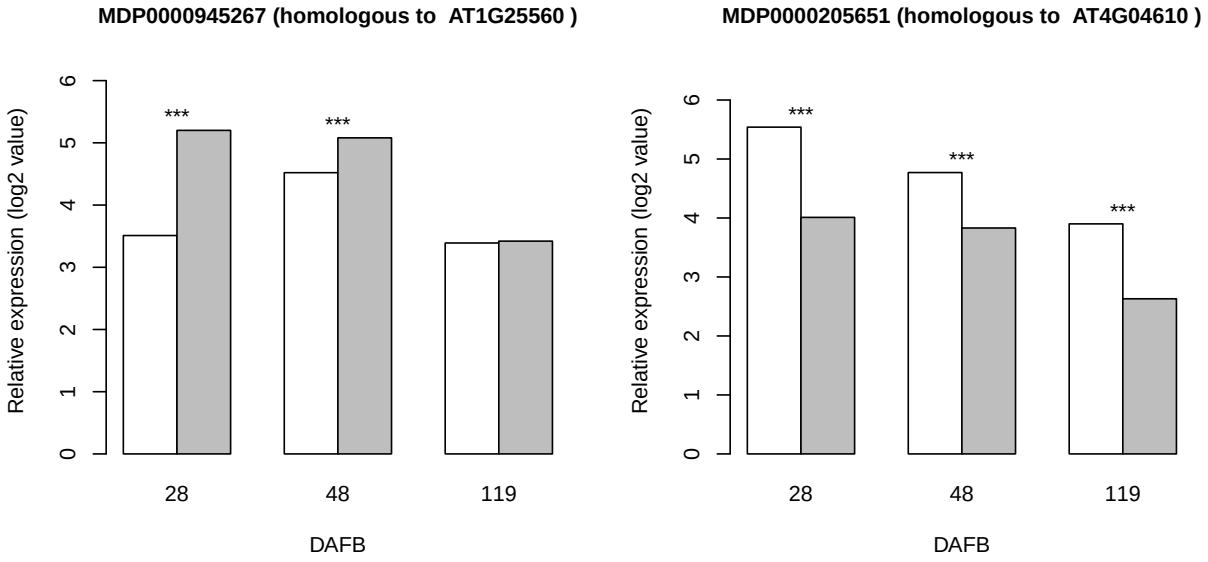
```
# Read datafile
STKxGS_map <- read.table("STKxGS_map.txt", header=T)

# Make a data frame of position information of genes that showed significant differential
  # expression for at least one of the three dates.
position_genes_modulated <- position_genes_modulated_at_least_one_time[,1:3]
colnames(position_genes_modulated) <- c("Marker_name", "chr", "Position_bp")

# Merge the two files
position_STKxGS_map <- rbind(STKxGS_map, position_genes_modulated)

# Make a data frame for input in a drawing software
  position_STKxGS_map$Marker_name <- as.character(position_STKxGS_map$Marker_name)
  position_STKxGS_map$chr <- as.character(position_STKxGS_map$chr)
  position_STKxGS_map$Position_bp <- as.numeric(position_STKxGS_map$Position_bp)
  # Order the data frame by chromosome and position
  position_STKxGS_map <- position_STKxGS_map[order(position_STKxGS_map$chr,
                          position_STKxGS_map$Position_bp, decreasing = FALSE),]

# Save the data frame in a text file.
    write.table(position_STKxGS_map, file = "position_STKxGS_map_output.txt",
                sep = "\t", quote = FALSE, row.names = FALSE)
```

10. Graphic representation of expression level

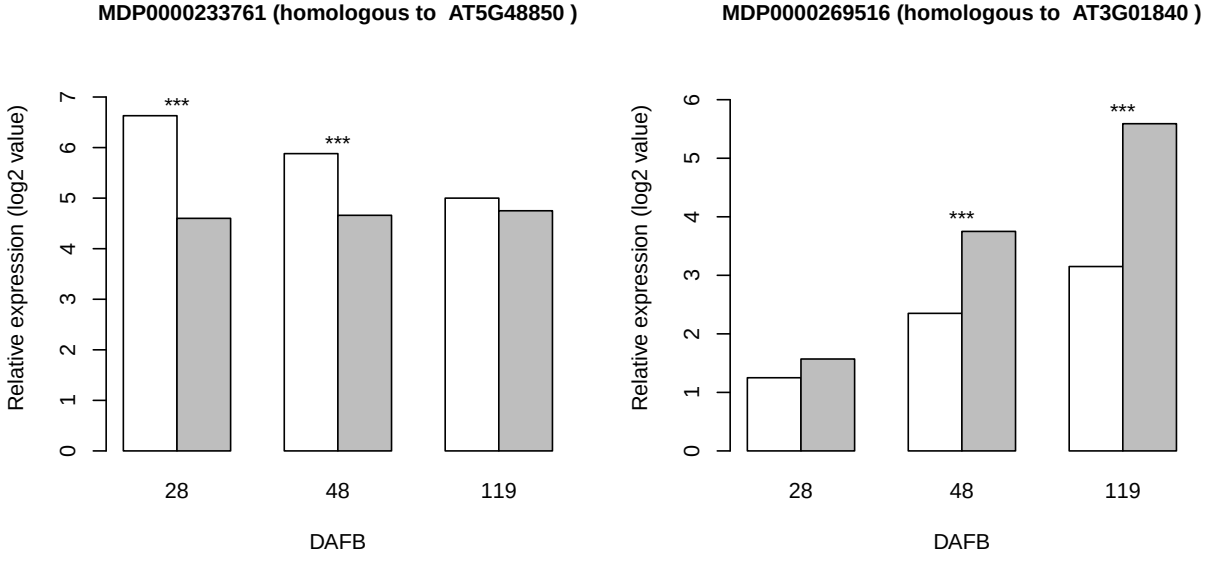
Following the entire analysis, a short list of transcripts appears to be of particular interest for the molecular control of biennial bearing. These transcripts are discussed in the submitted article and to illustrate this discussion, their relative expression values are graphically represented.

Fig.13. Kinetics of the relative expression values (log2 ratio) of transcripts differentially expressed in apple meristems between trees initiating flowering ('OFF') and trees inhibiting flowering ('ON') at three developmental stages (28, 48 and 119 DAFB). The array data were normalized with the lowess method. Normalized intensities (i.e. expression levels) were then subtracted from the background. Stars (***) indicate significant differences of expression between the two treatments.

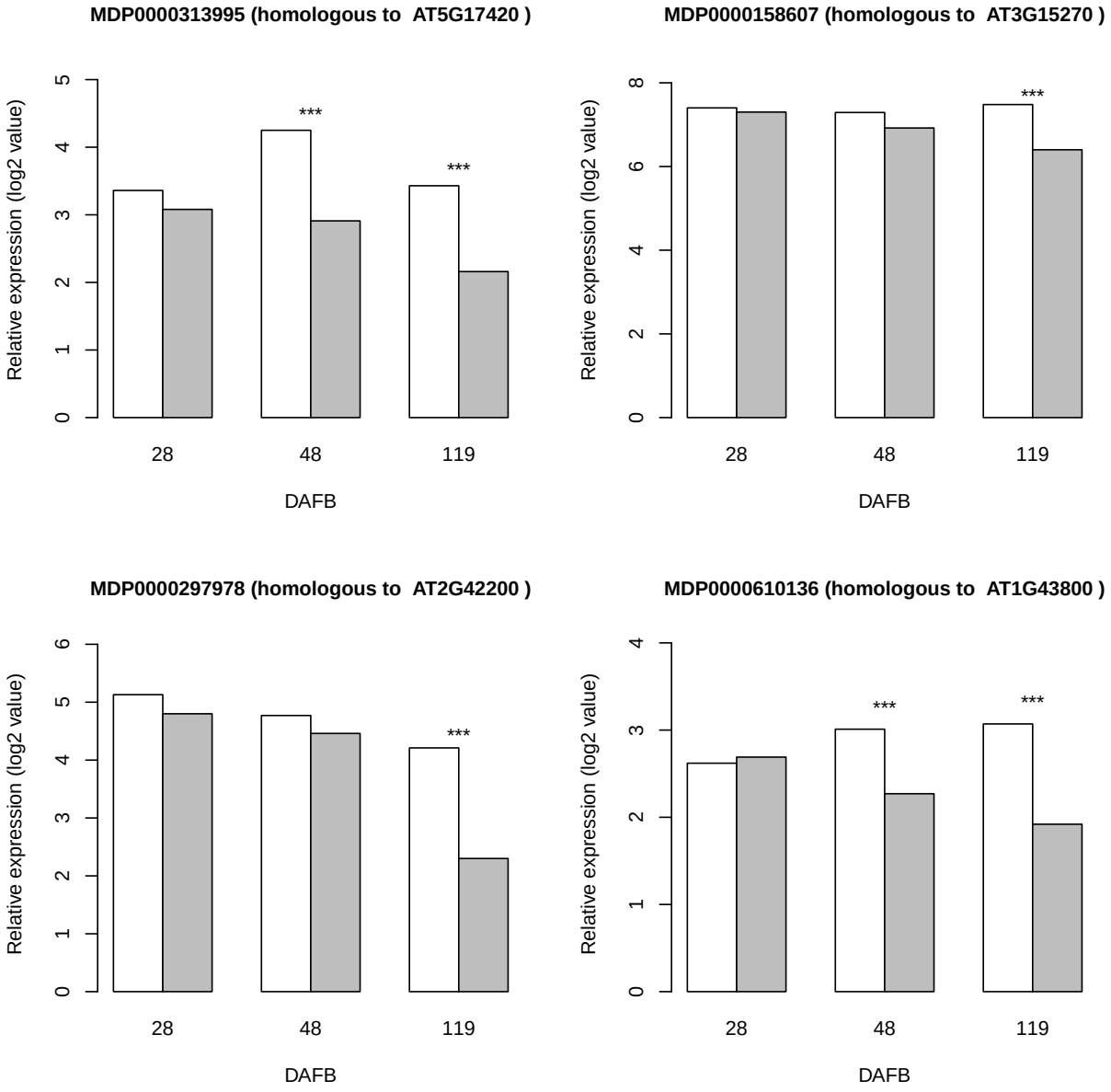



MDP0000945267 (homologous to AT1G25560)
Relative expression (log2 value)
0 1 2 3 4 5 6
***
***
28
48
119
DAFB

MDP0000205651 (homologous to AT4G04610)
Relative expression (log2 value)
0 1 2 3 4 5 6
***
***
***
28
48
119
DAFB


MDP0000233761 (homologous to AT5G48850)
Relative expression (log2 value)
0 1 2 3 4 5 6 7
***
***
28
48
119
DAFB

MDP0000269516 (homologous to AT3G01840)
Relative expression (log2 value)
0 1 2 3 4 5 6
***
***
28
48
119
DAFB


MDP0000313995 (homologous to AT5G17420)
Relative expression (log2 value)
0
1
2
3
4
5
***
***
28
48
119
DAFB
MDP0000158607 (homologous to AT3G15270)
Relative expression (log2 value)
0
2
4
6
8
***
28
48
119
DAFB
MDP0000297978 (homologous to AT2G42200)
Relative expression (log2 value)
0
1
2
3
4
5
6
***
28
48
119
DAFB
MDP0000610136 (homologous to AT1G43800)
Relative expression (log2 value)
0
1
2
3
4
***
***
28
48
119
DAFB