# Install a few packages that are required.

source("http://bioconductor.org/biocLite.R")  
#biocLite(ggplot2)  
#biocLite(RColorBrewer)  
#biocLite(plot3D)  
#biocLite(cluster)  
#biocLite(WGCNA)  
#biocLite("metaMA")  
#biocLite("lattice")  
#biocLite("genefilter")  
#biocLite("dplyr"")  
#biocLite("pathview")  
library(lattice)  
library(genefilter)  
library(edgeR)  
library(ggplot2)  
library(RColorBrewer)  
library(plot3D)  
library(cluster)  
library(WGCNA)

## ==========================================================================  
## \*  
## \* Package WGCNA 1.51 loaded.  
## \*  
## \* Important note: It appears that your system supports multi-threading,  
## \* but it is not enabled within WGCNA in R.   
## \* To allow multi-threading within WGCNA with all available cores, use   
## \*  
## \* allowWGCNAThreads()  
## \*  
## \* within R. Use disableWGCNAThreads() to disable threading if necessary.  
## \* Alternatively, set the following environment variable on your system:  
## \*  
## \* ALLOW\_WGCNA\_THREADS=<number\_of\_processors>  
## \*  
## \* for example   
## \*  
## \* ALLOW\_WGCNA\_THREADS=4  
## \*  
## \* To set the environment variable in linux bash shell, type   
## \*  
## \* export ALLOW\_WGCNA\_THREADS=4  
## \*  
## \* before running R. Other operating systems or shells will  
## \* have a similar command to achieve the same aim.  
## \*  
## ==========================================================================

# Read in data

data <- read.table(file="all\_counts.txt", sep="\t", header=T, stringsAsFactors=F)  
pdata <- read.table(file="pdata.txt", sep="\t", header=T, stringsAsFactors=F)  
pdata$Cultivar <- factor(pdata$Cultivar, levels=c("C", "I5", "I8"))  
pdata$TimePoint <- factor(pdata$TimePoint, levels=c("6", "9"))  
head(pdata)

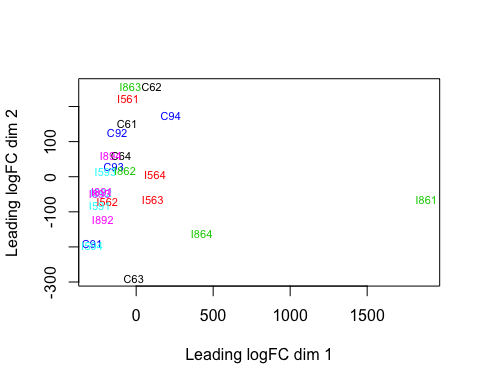
## Cultivar TimePoint  
## C61 C 6  
## C62 C 6  
## C63 C 6  
## C64 C 6  
## C91 C 9  
## C92 C 9

# generate normalized counts (counts-per-million) using cpm() function

keep <- rowSums(cpm(data) > 1) > 1  
counts <- data[keep,]  
norm.counts <- cpm(counts)

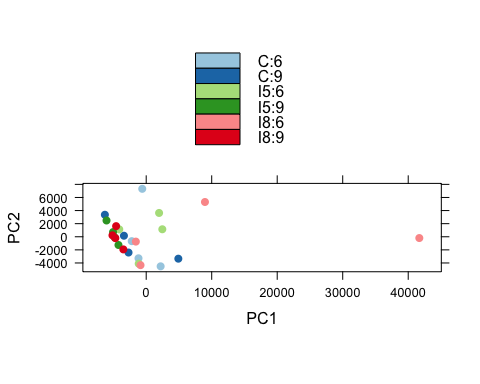
# generate MDS plot

group <- interaction(pdata$Cultivar, pdata$TimePoint)  
labs <- colnames(norm.counts)  
plotMDS(norm.counts, col=as.numeric(group), labels=labs, lwd=2, cex=0.7)

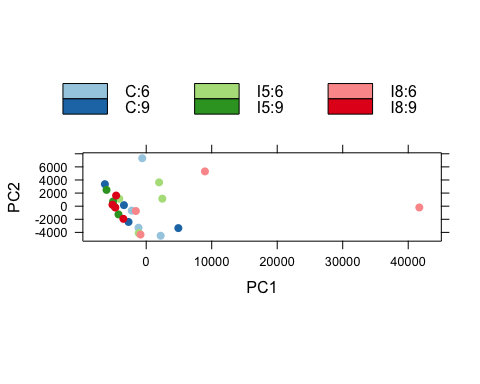


# PCA plot

rv <- rowVars(norm.counts)  
select <- order(rv, decreasing=TRUE)[seq\_len(100)]  
pca <- prcomp(t(norm.counts[select,]))  
fac <- factor(apply(pdata[,c("Cultivar", "TimePoint")], 1, paste, collapse=":"))  
colours <- brewer.pal(nlevels(fac), "Paired")  
pcafig <- xyplot(PC2 ~ PC1, groups=fac, data=as.data.frame(pca$x), pch=16, cex=1, aspect="iso",  
 col=colours, main=draw.key(key=list(rect=list(col=colours), text=list(levels(fac)),  
 rep=FALSE)))  
print(pcafig)

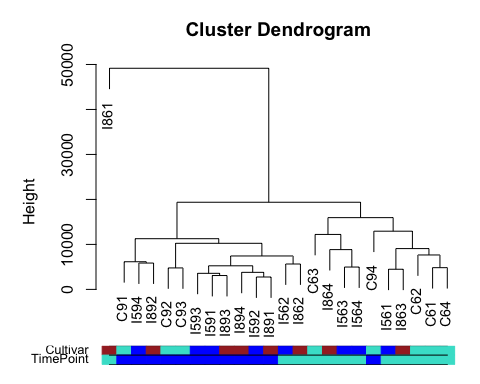


## Draw colour keys non-default way  
pcafig <- xyplot(PC2 ~ PC1, groups=fac, data=as.data.frame(pca$x), pch=16, cex=1, aspect="iso",  
 col=colours, main=draw.key(key=list(rect=list(col=colours), text=list(levels(fac)),  
 rep=FALSE, columns=3)))  
print(pcafig)



# Clustering of samples

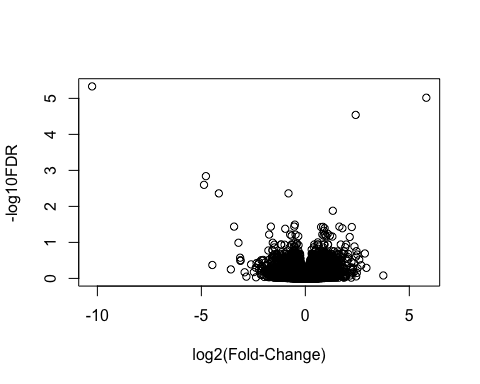
dist.matrix <- dist(t(norm.counts))  
sampleTree <- hclust(dist.matrix)  
colours <- data.frame(Cultivar=labels2colors(pdata$Cultivar), TimePoint=labels2colors(pdata$TimePoint))  
plotDendroAndColors(sampleTree, colors=colours, groupLabels=c("Cultivar", "TimePoint"), colorHeight=0.1, autoColorHeight=FALSE)



# Visualize differential expression results by volcano plot

library(dplyr)

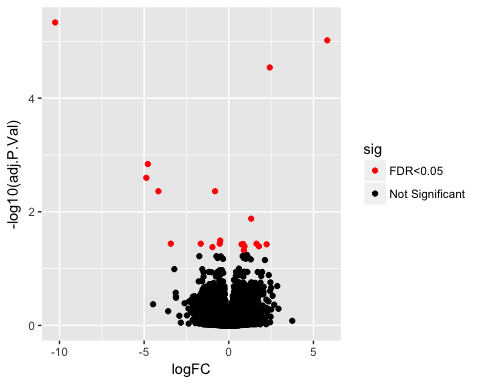
data <- read.table(file="I5\_v\_C\_time6.txt", sep="\t", header=T, stringsAsFactors=F)  
logFDR <- -log10(data$adj.P.Val)  
## plot -log10(adj.P.Val) ~ logFC  
plot(data$logFC, logFDR, xlab="log2(Fold-Change)", ylab="-log10FDR")



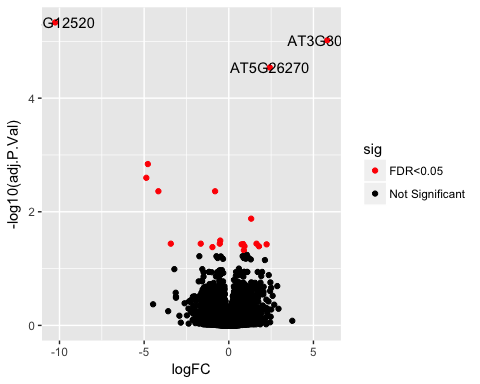
## label significant genes  
data <- mutate(data, sig=ifelse(data$adj.P.Val<0.05, "FDR<0.05", "Not Significant"))

## Warning: package 'bindrcpp' was built under R version 3.2.5

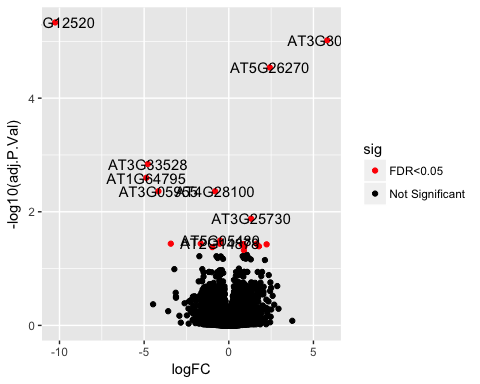
## plot volcano with significant genes in red  
p <- ggplot(data, aes(logFC, -log10(adj.P.Val))) + geom\_point(aes(col=sig)) + scale\_color\_manual(values=c("red", "black"))  
p



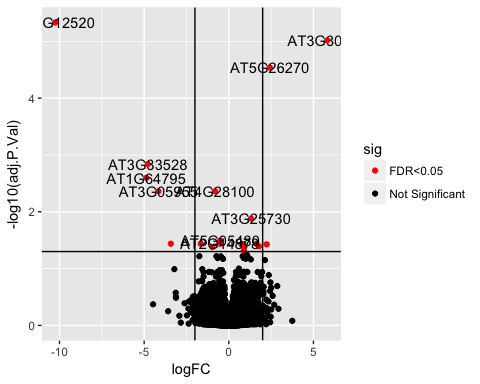
## add gene names to selected genes  
p + geom\_text(data=filter(data, adj.P.Val<0.001), aes(label=Gene))



p + geom\_text(data=data[order(data$adj.P.Val, decreasing=FALSE)[1:10],], aes(label=Gene))

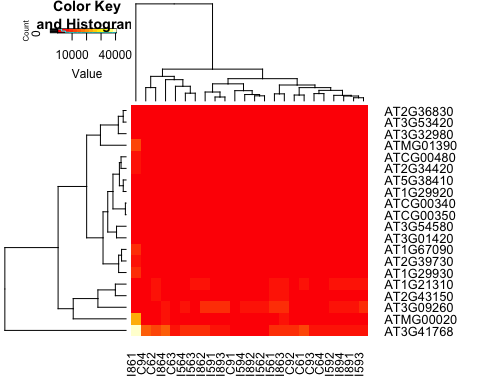


p + geom\_text(data=data[order(data$adj.P.Val, decreasing=FALSE)[1:10],], aes(label=Gene)) + geom\_vline(xintercept=c(-2,2), colour="black") + geom\_hline(yintercept=1.3, colour="black")

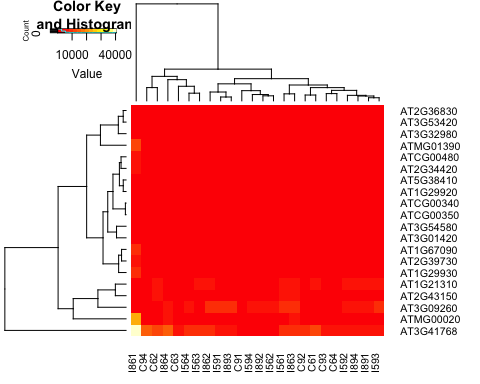


# Heatmaps

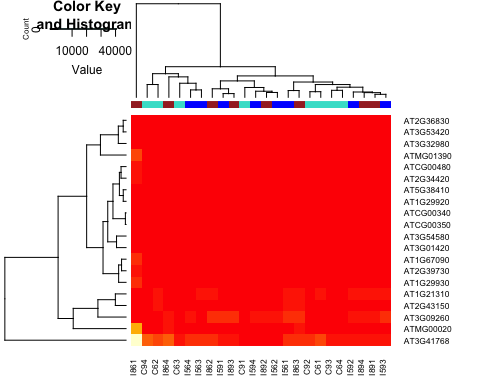
library(devtools)  
library(gplots)  
slt <- order(rv, decreasing=TRUE)[seq\_len(20)]  
heatmap.2(norm.counts[slt,], col=heat.colors, trace="none", margin=c(3,7))



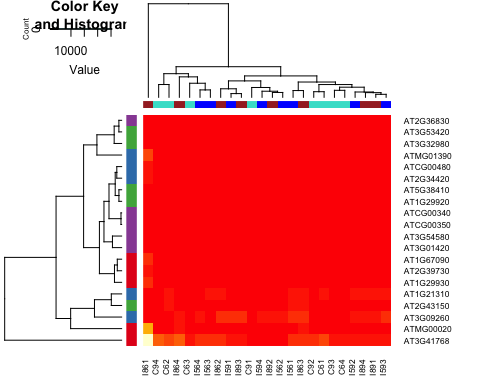
heatmap.2(norm.counts[slt,], col=heat.colors, trace="none", margin=c(3,6), cexRow=0.8, cexCol=0.8)



heatmap.2(norm.counts[slt,], col=heat.colors, trace="none", margin=c(3,7), cexRow=0.8, cexCol=0.8, ColSideColors=labels2colors(pdata$Cultivar))



rowcols <- rep(brewer.pal(4, 'Set1'), each=5)  
names(rowcols) <- rownames(norm.counts[slt,])  
heatmap.2(norm.counts[slt,], col=heat.colors, trace="none", margin=c(3,7), cexRow=0.8, cexCol=0.8, ColSideColors=labels2colors(pdata$Cultivar), RowSideColors=rowcols)



## Heatmaps with multiple side bars using heatmap.3()

source\_url("https://raw.githubusercontent.com/obigriffith/biostar-tutorials/master/Heatmaps/heatmap.3.R")

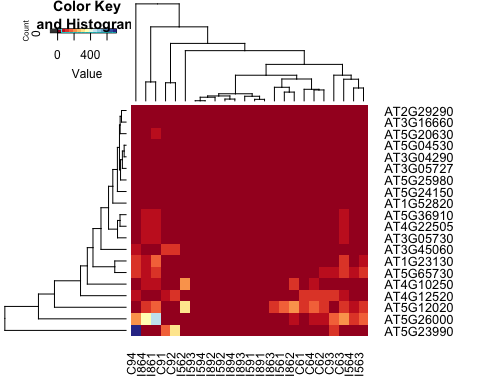
## SHA-1 hash of file is 015fc0457e61e3e93a903e69a24d96d2dac7b9fb

rlab <- t(rowcols)  
rownames(rlab) <- "GeneType"  
clab <- cbind(labels2colors(pdata$Cultivar), labels2colors(pdata$TimePoint))  
colnames(clab) <- c("Cultivar", "TimePoint")  
# The plot will be saved to a pdf file because of the size of the figure  
pdf("test\_heatmap3.pdf")  
heatmap.3(norm.counts[slt,], col=heat.colors, trace="none", cexRow=0.8, cexCol=0.8, ColSideColors=clab, RowSideColors=rlab, ColSideColorsSize=2, RowSideColorsSize=2, margin=c(5,5))  
dev.off()

## quartz\_off\_screen   
## 2

## Using log transformed data to select feature for generating heatmap.

log.counts <- cpm(counts, log=TRUE)  
rv <- rowVars(log.counts)  
slt <- order(rv, decreasing=TRUE)[seq\_len(20)]  
# use non-default color scheme  
mypalette <- brewer.pal(11, "RdYlBu")  
morecols <- colorRampPalette(mypalette)  
heatmap.2(norm.counts[slt,], col=morecols, trace="none", margin=c(3,7))



# Visulize pathway enrichment results using bioconductor package "pathview"

library(pathview)

DE.paths <- read.table(file="I5\_v\_C\_time6\_KEGG.txt", sep="\t", header=T, stringsAsFactors=F)  
head(DE.paths, 1)

## pathway.code pathway.name p.value  
## 1 ath03010 Ribosome - Arabidopsis thaliana (thale cress) 5.378589e-35  
## Annotated  
## 1 301

pid <- DE.paths$pathway.code[3]  
DE.expr <- read.table(file="I5\_v\_C\_time6.txt", sep="\t", header=T, stringsAsFactors=F)  
head(DE.expr, 2)

## Gene logFC AveExpr P.Value adj.P.Val  
## 1 AT4G12520 -10.254556 0.3581132 2.206726e-10 4.651998e-06  
## 2 AT3G30720 5.817438 3.3950689 9.108689e-10 9.601014e-06

rownames(DE.expr) <- DE.expr$Gene  
colnames(DE.expr)

## [1] "Gene" "logFC" "AveExpr" "P.Value" "adj.P.Val"

gene.data <- subset(DE.expr, select="logFC")  
head(gene.data)

## logFC  
## AT4G12520 -10.254556  
## AT3G30720 5.817438  
## AT5G26270 2.421030  
## AT3G33528 -4.780814  
## AT1G64795 -4.872595  
## AT3G05955 -4.158939

pv.out <- pathview(gene.data=gene.data, pathway.id=pid, species="ath", gene.idtype="KEGG", kegg.native=T)

## Info: Working in directory /Users/jli/Jessie/Research/BioInfo/presentation/Figure

## Info: Writing image file ath04141.pathview.png

By default, running pathview() will create an image file named by the pathway id (for example, in this case there should be a file named "ath04712.pathview.png" in the current directory).

# Visulize GO enrichment results use revigo.irb.hr web application

In a web browser (Safari, explorer, chrome, firefox), open revigo.irb.hr.