Deseq2 analysis 1dpi

#LOAD PACKKAGES

library('BiocManager') #for installing bioconductor packages like DESeq2  
library('DESeq2') #for analyzing differential gene expression  
library("mixOmics") #for PCA and PLSDA  
library('ggplot2') #for plotting  
library("VennDiagram") #for venn diagrams  
library("ComplexHeatmap") #for heatmaps  
library("viridis") #for gene clustering heatmap colours  
library('RColorBrewer') #for sample-sample heatmap colours  
library('dendextend') #for editing dendrograms in gene clustering heatmap  
library("dplyr") #for data manipulation  
library('caret') #for removing zero variance data for PCA/PLSDA

#LOAD 1DPI DATA

#Think the way to do DESeq analysis/venn-diagrams is to split into two "experiments" and then compare the two to select primed metabolites  
#For PCA and PLS-DA do single experiment (i.e., mock + PM combined) as need to compare all\_1dpi treatments   
  
#MOCK  
  
#read in sample info file  
mock\_one\_dpi\_sample\_info <- read.table("raw\_count/1dpi/mock\_sample\_info\_1dpi.csv", header = TRUE, sep = ",")  
  
#add rownames to sample info = for labelling of plots (e.g., heatmaps)  
row.names(mock\_one\_dpi\_sample\_info) <- mock\_one\_dpi\_sample\_info$sample\_name  
  
#create factors for condition column so that water will be used as intercept/base-line/reference for the model  
mock\_one\_dpi\_sample\_info$condition <- factor(mock\_one\_dpi\_sample\_info$condition, levels = c("Mock\_Water", "Mock\_BABA", "Mock\_JA", "Mock\_SA"))  
  
#visualize the model matrix that will be used by Deseq2  
model.matrix(~condition, data = mock\_one\_dpi\_sample\_info)

## (Intercept) conditionMock\_BABA conditionMock\_JA conditionMock\_SA  
## 1 1 0 0 0  
## 2 1 0 0 0  
## 3 1 0 0 0  
## 4 1 0 0 0  
## 5 1 1 0 0  
## 6 1 1 0 0  
## 7 1 1 0 0  
## 8 1 1 0 0  
## 9 1 0 1 0  
## 10 1 0 1 0  
## 11 1 0 1 0  
## 12 1 0 1 0  
## 13 1 0 0 1  
## 14 1 0 0 1  
## 15 1 0 0 1  
## 16 1 0 0 1  
## attr(,"assign")  
## [1] 0 1 1 1  
## attr(,"contrasts")  
## attr(,"contrasts")$condition  
## [1] "contr.treatment"

#should display 4 columns - intercept (MW), MB, MJA, MSA  
  
#read in gene counts data  
mock\_one\_dpi\_raw\_gene\_data <- DESeqDataSetFromHTSeqCount(  
 sampleTable = mock\_one\_dpi\_sample\_info, directory = 'raw\_count/1dpi',  
 design = ~condition)  
  
#PM  
  
#read in sample info file  
PM\_one\_dpi\_sample\_info <- read.table("raw\_count/1dpi/PM\_sample\_info\_1dpi.csv", header = TRUE, sep = ",")  
  
#add rownames to sample info = for labelling of plots (e.g., heatmaps)  
row.names(PM\_one\_dpi\_sample\_info) <- PM\_one\_dpi\_sample\_info$sample\_name  
  
#create factors for condition column so that water will be used as intercept/base-line/reference for the model  
PM\_one\_dpi\_sample\_info$condition <- factor(PM\_one\_dpi\_sample\_info$condition, levels = c("PM\_Water", "PM\_BABA", "PM\_JA", "PM\_SA"))  
  
#visualize the model matrix that will be used by Deseq2  
model.matrix(~condition, data = PM\_one\_dpi\_sample\_info)

## (Intercept) conditionPM\_BABA conditionPM\_JA conditionPM\_SA  
## 1 1 0 0 0  
## 2 1 0 0 0  
## 3 1 0 0 0  
## 4 1 0 0 0  
## 5 1 1 0 0  
## 6 1 1 0 0  
## 7 1 1 0 0  
## 8 1 1 0 0  
## 9 1 0 1 0  
## 10 1 0 1 0  
## 11 1 0 1 0  
## 12 1 0 1 0  
## 13 1 0 0 1  
## 14 1 0 0 1  
## 15 1 0 0 1  
## attr(,"assign")  
## [1] 0 1 1 1  
## attr(,"contrasts")  
## attr(,"contrasts")$condition  
## [1] "contr.treatment"

#should display 4 columns - intercept (MW), MB, MJA, MSA  
  
#read in gene counts data  
PM\_one\_dpi\_raw\_gene\_data <- DESeqDataSetFromHTSeqCount(  
 sampleTable = PM\_one\_dpi\_sample\_info, directory = 'raw\_count/1dpi',  
 design = ~condition)  
  
#ALL DATA (MOCK + PM)   
  
#read in sample info file  
all\_one\_dpi\_sample\_info <- read.table("raw\_count/1dpi/sample\_info\_1dpi.csv", header = TRUE, sep = ",")  
  
#add rownames to sample info = for labelling of plots (e.g., heatmaps)  
row.names(all\_one\_dpi\_sample\_info) <- all\_one\_dpi\_sample\_info$Samplename  
  
#create factors for elicitor column so that water will be used as intercept/base-line  
all\_one\_dpi\_sample\_info$condition <- factor(all\_one\_dpi\_sample\_info$condition, levels = c("Mock\_Water", "Mock\_BABA", "Mock\_JA", "Mock\_SA", "PM\_Water", "PM\_BABA", "PM\_JA", "PM\_SA"))  
  
#visualize the model matrix that will be used by Deseq2  
model.matrix(~condition, data = all\_one\_dpi\_sample\_info)

## (Intercept) conditionMock\_BABA conditionMock\_JA conditionMock\_SA  
## 1MW\_1 1 0 0 0  
## 1MW\_2 1 0 0 0  
## 1MW\_3 1 0 0 0  
## 1MW\_4 1 0 0 0  
## 1MB\_1 1 1 0 0  
## 1MB\_2 1 1 0 0  
## 1MB\_3 1 1 0 0  
## 1MB\_4 1 1 0 0  
## 1MJ\_1 1 0 1 0  
## 1MJ\_2 1 0 1 0  
## 1MJ\_3 1 0 1 0  
## 1MJ\_4 1 0 1 0  
## 1MS\_1 1 0 0 1  
## 1MS\_2 1 0 0 1  
## 1MS\_3 1 0 0 1  
## 1MS\_4 1 0 0 1  
## 1PMW\_1 1 0 0 0  
## 1PMW\_2 1 0 0 0  
## 1PMW\_3 1 0 0 0  
## 1PMW\_4 1 0 0 0  
## 1PMB\_1 1 0 0 0  
## 1PMB\_2 1 0 0 0  
## 1PMB\_3 1 0 0 0  
## 1PMB\_4 1 0 0 0  
## 1PMJ\_1 1 0 0 0  
## 1PMJ\_2 1 0 0 0  
## 1PMJ\_3 1 0 0 0  
## 1PMJ\_4 1 0 0 0  
## 1PMS\_1 1 0 0 0  
## 1PMS\_2 1 0 0 0  
## 1PMS\_3 1 0 0 0  
## conditionPM\_Water conditionPM\_BABA conditionPM\_JA conditionPM\_SA  
## 1MW\_1 0 0 0 0  
## 1MW\_2 0 0 0 0  
## 1MW\_3 0 0 0 0  
## 1MW\_4 0 0 0 0  
## 1MB\_1 0 0 0 0  
## 1MB\_2 0 0 0 0  
## 1MB\_3 0 0 0 0  
## 1MB\_4 0 0 0 0  
## 1MJ\_1 0 0 0 0  
## 1MJ\_2 0 0 0 0  
## 1MJ\_3 0 0 0 0  
## 1MJ\_4 0 0 0 0  
## 1MS\_1 0 0 0 0  
## 1MS\_2 0 0 0 0  
## 1MS\_3 0 0 0 0  
## 1MS\_4 0 0 0 0  
## 1PMW\_1 1 0 0 0  
## 1PMW\_2 1 0 0 0  
## 1PMW\_3 1 0 0 0  
## 1PMW\_4 1 0 0 0  
## 1PMB\_1 0 1 0 0  
## 1PMB\_2 0 1 0 0  
## 1PMB\_3 0 1 0 0  
## 1PMB\_4 0 1 0 0  
## 1PMJ\_1 0 0 1 0  
## 1PMJ\_2 0 0 1 0  
## 1PMJ\_3 0 0 1 0  
## 1PMJ\_4 0 0 1 0  
## 1PMS\_1 0 0 0 1  
## 1PMS\_2 0 0 0 1  
## 1PMS\_3 0 0 0 1  
## attr(,"assign")  
## [1] 0 1 1 1 1 1 1 1  
## attr(,"contrasts")  
## attr(,"contrasts")$condition  
## [1] "contr.treatment"

#read in gene counts data  
all\_one\_dpi\_raw\_gene\_data <- DESeqDataSetFromHTSeqCount(  
 sampleTable = all\_one\_dpi\_sample\_info, directory ='raw\_count/1dpi',  
 design = ~condition)

#MOCK QUALITY SCATTERPLOTS

#extract and save normalised counts  
mock\_one\_dpi\_raw\_gene\_data <- estimateSizeFactors(mock\_one\_dpi\_raw\_gene\_data)  
mock\_one\_dpi\_normalised\_raw\_gene\_data <- counts(mock\_one\_dpi\_raw\_gene\_data, normalized=TRUE)  
  
#set up pdf to save graphs  
quality\_scatterplots <- "Deseq\_analysis/one\_dpi\_out/mock\_one\_dpi\_quality\_scatterplots.pdf"  
pdf(file = quality\_scatterplots)  
  
#mock water  
par(mfrow = c(2,3)) #2 rows, 3 cols  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,1]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,2]), xlab = "1MW\_1", ylab = "1MW\_2")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "A")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,1]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,3]), xlab = "1MW\_1", ylab = "1MW\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "B")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,1]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,4]), xlab = "1MW\_1", ylab = "1MW\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "C")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,2]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,3]), xlab = "1MW\_2", ylab = "1MW\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "D")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,2]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,4]), xlab = "1MW\_2", ylab = "1MW\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "E")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,3]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,4]), xlab = "1MW\_3", ylab = "1MW\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "F")  
  
#mock BABA  
par(mfrow = c(2,3)) #2 rows, 3 cols  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,5]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,6]), xlab = "1MB\_1", ylab = "1MB\_2")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "A")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,5]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,7]), xlab = "1MB\_1", ylab = "1MB\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "B")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,5]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,8]), xlab = "1MB\_1", ylab = "1MB\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "C")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,6]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,7]), xlab = "1MB\_2", ylab = "1MB\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "D")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,6]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,8]), xlab = "1MB\_2", ylab = "1MB\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "E")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,7]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,8]), xlab = "1MB\_3", ylab = "1MB\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "F")  
  
#mock JA  
par(mfrow = c(2,3)) #2 rows, 3 cols  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,9]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,10]), xlab = "1MJ\_1", ylab = "1MJ\_2")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "A")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,9]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,11]), xlab = "1MJ\_1", ylab = "1MJ\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "B")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,9]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,12]), xlab = "1MJ\_1", ylab = "1MJ\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "C")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,10]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,11]), xlab = "1MJ\_2", ylab = "1MJ\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "D")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,10]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,12]), xlab = "1MJ\_2", ylab = "1MJ\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "E")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,11]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,12]), xlab = "1MJ\_3", ylab = "1MJ\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "F")  
  
#mock SA  
par(mfrow = c(2,3)) #2 rows, 3 cols  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,13]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,14]), xlab = "1MS\_1", ylab = "1MS\_2")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "A")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,13]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,15]), xlab = "1MS\_1", ylab = "1MS\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "B")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,13]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,16]), xlab = "1MS\_1", ylab = "1MS\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "C")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,14]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,15]), xlab = "1MS\_2", ylab = "1MS\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "D")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,14]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,16]), xlab = "1MS\_2", ylab = "1MS\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "E")  
  
plot(log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,15]), log(mock\_one\_dpi\_normalised\_raw\_gene\_data[,16]), xlab = "1MS\_3", ylab = "1MS\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "F")  
  
dev.off()

#PM QUALITY SCATTERPLOTS

#extract and save normalised counts  
PM\_one\_dpi\_raw\_gene\_data <- estimateSizeFactors(PM\_one\_dpi\_raw\_gene\_data)  
PM\_one\_dpi\_normalised\_raw\_gene\_data <- counts(PM\_one\_dpi\_raw\_gene\_data, normalized=TRUE)  
  
#set up pdf to save graphs  
quality\_scatterplots <-"Deseq\_analysis/one\_dpi\_out/PM\_one\_dpi\_quality\_scatterplots.pdf"  
pdf(file = quality\_scatterplots)  
  
#PM water  
par(mfrow = c(2,3)) #2 rows, 3 cols  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,1]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,2]), xlab = "1PMW\_1", ylab = "1PMW\_2")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "A")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,1]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,3]), xlab = "1PMW\_1", ylab = "1PMW\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "B")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,1]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,4]), xlab = "1PMW\_1", ylab = "1PMW\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "C")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,2]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,3]), xlab = "1PMW\_2", ylab = "1PMW\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "D")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,2]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,4]), xlab = "1PMW\_2", ylab = "1PMW\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "E")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,3]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,4]), xlab = "1PMW\_3", ylab = "1PMW\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "F")  
  
#PM BABA  
par(mfrow = c(2,3)) #2 rows, 3 cols  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,5]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,6]), xlab = "1PMB\_1", ylab = "1PMB\_2")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "A")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,5]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,7]), xlab = "1PMB\_1", ylab = "1PMB\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "B")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,5]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,8]), xlab = "1PMB\_1", ylab = "1PMB\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "C")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,6]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,7]), xlab = "1PMB\_2", ylab = "1PMB\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "D")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,6]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,8]), xlab = "1PMB\_2", ylab = "1PMB\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "E")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,7]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,8]), xlab = "1PMB\_3", ylab = "1PMB\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "F")  
  
#PM JA  
par(mfrow = c(2,3)) #2 rows, 3 cols  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,9]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,10]), xlab = "1PMJ\_1", ylab = "1PMJ\_2")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "A")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,9]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,11]), xlab = "1PMJ\_1", ylab = "1PMJ\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "B")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,9]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,12]), xlab = "1PMJ\_1", ylab = "1PMJ\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "C")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,10]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,11]), xlab = "1PMJ\_2", ylab = "1PMJ\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "D")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,10]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,12]), xlab = "1PMJ\_2", ylab = "1PMJ\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "E")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,11]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,12]), xlab = "1PMJ\_3", ylab = "1PMJ\_4")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "F")  
  
#PM SA  
par(mfrow = c(2,3)) #2 rows, 3 cols  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,13]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,14]), xlab = "1PMS\_1", ylab = "1PMS\_2")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "A")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,13]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,15]), xlab = "1PMS\_1", ylab = "1PMS\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "B")  
  
plot(log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,14]), log(PM\_one\_dpi\_normalised\_raw\_gene\_data[,15]), xlab = "1PMS\_2", ylab = "1PMS\_3")  
  
abline(a = 0, b = 1, col = "red")  
  
title(main = "C")  
  
dev.off()

#TRANSFORMATION

rlog\_all\_one\_dpi\_raw\_gene\_data <- rlog(all\_one\_dpi\_raw\_gene\_data)

## rlog() may take a few minutes with 30 or more samples,  
## vst() is a much faster transformation

#SAMPLE-SAMPLE DISTANCE HEATMAP

#Basically shows how similar the different samples are, uses normalised data  
  
#set up pdf to save graphs  
sample\_clustering\_dend <- "Deseq\_analysis/one\_dpi\_out/one\_dpi\_sample\_clustering\_dend.pdf"  
pdf(file = sample\_clustering\_dend, width = 10, height = 5)  
  
#first generate sample-sample distance matrix  
sampleDistances <- dist(t(assay(rlog\_all\_one\_dpi\_raw\_gene\_data)))  
sampleDistMatrix <- as.matrix(sampleDistances)  
colnames(sampleDistMatrix) <- rownames(all\_one\_dpi\_sample\_info)  
  
#perform hierachical clustering and plot dendrogram  
col = c("blue", "#AA4499", "#AA4499", "#117733", "#44AA99", "#44AA99", "#DDCC77", "#33bbee", "#33bbee", "#33bbee", "#33bbee", "#999933", "#999933", "#999933", "blue", "blue", "blue", "#117733", "#117733", "#AA4499", "#AA4499", "#DDCC77", "#DDCC77", "#DDCC77", "#44AA99", "#44AA99", "#117733", "#882255", "#882255", "#882255", "#882255") #set up sample colours  
one\_dpi\_sample\_dendrogram <- as.dendrogram(hclust(sampleDistances)) #cluster  
dend <- dendextend::set(one\_dpi\_sample\_dendrogram, "labels\_colors", col) #color code labels  
plot(dend) #plot   
dendextend::rect.dendrogram(dend, h = 95, border = "red", lty = "dashed") #add dashed red rectangles around clusters (visually a cut height of 95 ok)  
  
dev.off()

#set up pdf to save graphs  
sample\_clustering <- "Deseq\_analysis/one\_dpi\_out/one\_dpi\_sample\_clustering.pdf"  
pdf(file = sample\_clustering, width = 15, height = 15)  
  
#create colour scheme  
colours <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)  
  
#set up sample annotations for plot  
sample\_annotations <- all\_one\_dpi\_sample\_info[,3]  
  
treatments\_top <- HeatmapAnnotation(Condition = sample\_annotations,   
 col = list(Condition = c("Mock\_Water" = "blue", "Mock\_BABA" = "#117733", "Mock\_JA" = "#AA4499", "Mock\_SA" = "#DDCC77",  
 "PM\_Water" = "#33bbee", "PM\_BABA" = "#44AA99", "PM\_JA" = "#882255", "PM\_SA" = "#999933")),   
 show\_annotation\_name = F, gp = gpar(col = "black"))  
  
treatments\_left <- rowAnnotation(Condition = sample\_annotations,   
 col = list(Condition = c("Mock\_Water" = "blue", "Mock\_BABA" = "#117733", "Mock\_JA" = "#AA4499", "Mock\_SA" = "#DDCC77",   
 "PM\_Water" = "#33bbee", "PM\_BABA" = "#44AA99", "PM\_JA" = "#882255", "PM\_SA" = "#999933")),  
 show\_legend = F, show\_annotation\_name = F, gp = gpar(col = "black"))  
  
ComplexHeatmap::Heatmap(sampleDistMatrix, clustering\_distance\_rows = sampleDistances, clustering\_distance\_columns = sampleDistances,  
 col = colours, top\_annotation = treatments\_top, left\_annotation = treatments\_left,  
 heatmap\_legend\_param = list(title = NULL, border = "black"),  
 show\_row\_names = F, show\_column\_names = F, column\_title = "1dpi Sample Clustering Heatmap")  
  
dev.off()

#SET UP PCA AND PLSDA

#extract rlog expression data (unsure if use rlog expression or normal expression)  
rlog\_expression\_all\_one\_dpi <- t(assay(rlog\_all\_one\_dpi\_raw\_gene\_data))  
dim(rlog\_expression\_all\_one\_dpi) #check dimensions of resulting matrix

## [1] 31 25808

#need to remove colums with all 0s/zero variance as otherwise PLS/PCA will throw error: columns with zero variance in 'X'  
all\_cols <- colnames(rlog\_expression\_all\_one\_dpi) #identify names of all cols   
remove\_cols <- caret::nearZeroVar(rlog\_expression\_all\_one\_dpi, names = TRUE) #identify cols with zero var to remove   
rlog\_expression\_all\_one\_dpi <- rlog\_expression\_all\_one\_dpi[ , setdiff(all\_cols, remove\_cols)] #remove cols with zero var   
dim(rlog\_expression\_all\_one\_dpi) #check dimensions of resulting matrix, should be smaller

## [1] 31 22862

#extract classes (conditions)  
rlog\_classes\_all\_one\_dpi <- colData(rlog\_all\_one\_dpi\_raw\_gene\_data)  
rlog\_classes\_all\_one\_dpi <- rlog\_classes\_all\_one\_dpi$condition  
  
#check that all conditions extracted as expected  
summary(rlog\_classes\_all\_one\_dpi)

## Mock\_Water Mock\_BABA Mock\_JA Mock\_SA PM\_Water PM\_BABA PM\_JA   
## 4 4 4 4 4 4 4   
## PM\_SA   
## 3

#PCA SCORES

#create PCA model  
pca\_one\_dpi <- pca(rlog\_expression\_all\_one\_dpi, ncomp = 14, center = TRUE, scale = TRUE)  
  
#set up pdf to save graphs  
PCA <- "Deseq\_analysis/one\_dpi\_out/one\_dpi\_PCA\_scores\_plots.pdf"  
pdf(file = PCA)  
  
#plot pca for all samples comps 1 and 2  
plotIndiv(  
 pca\_one\_dpi, comp = c(1,2), group = rlog\_classes\_all\_one\_dpi, ind.names = FALSE,   
 ellipse = TRUE, ellipse.level = 0.95, legend = TRUE, title = '1dpi PCA: PC1 vs PC2',  
 col.per.group = c("blue", "#117733", "#AA4499", "#DDCC77", "#33bbee", "#44AA99", "#882255", "#999933"),  
 pch = c(19,19,19,19,17,17,17,17), cex = 3, pch.levels = c("mock", "mock", "mock", "mock", "PM", "PM", "PM", "PM"),  
 size.xlabel = rel(2), size.ylabel = rel(2), size.axis = rel(1.5))  
  
plotIndiv(  
 pca\_one\_dpi, comp = c(1,3), group = rlog\_classes\_all\_one\_dpi, ind.names = FALSE,   
 ellipse = TRUE, ellipse.level = 0.95, legend = TRUE, title = '1dpi PCA: PC1 vs PC3',   
 col.per.group = c("blue", "#117733", "#AA4499", "#DDCC77", "#33bbee", "#44AA99", "#882255", "#999933"),   
 pch = c(19,19,19,19,17,17,17,17), cex = 3, pch.levels = c("mock", "mock", "mock", "mock", "PM", "PM", "PM", "PM"),  
 size.xlabel = rel(2), size.ylabel = rel(2), size.axis = rel(1.5))   
  
plotIndiv(  
 pca\_one\_dpi, comp = c(2,3), group = rlog\_classes\_all\_one\_dpi, ind.names = FALSE,   
 ellipse = TRUE, ellipse.level = 0.95, legend = TRUE, title = '1dpi PCA: PC2 vs PC3',   
 col.per.group = c("blue", "#117733", "#AA4499", "#DDCC77", "#33bbee", "#44AA99", "#882255", "#999933"),   
 pch = c(19,19,19,19,17,17,17,17), cex = 3, pch.levels = c("mock", "mock", "mock", "mock", "PM", "PM", "PM", "PM"),  
 size.xlabel = rel(2), size.ylabel = rel(2), size.axis = rel(1.5))  
  
dev.off()

#PCA LOADINGS

#set up pdf to save graphs  
pca\_loadings <- "Deseq\_analysis/one\_dpi\_out/one\_dpi\_PCA\_loadings.pdf"  
pdf(file = pca\_loadings)  
  
#extract loadings data  
one\_dpi\_pca\_loadings <- as.data.frame(pca\_one\_dpi$loadings)  
  
#PC1 vs PC2  
ggplot(data = one\_dpi\_pca\_loadings)+  
 geom\_point(mapping = aes(x = X.PC1, y = X.PC2))+  
 scale\_y\_continuous(limits = c(-0.02,0.02), breaks = seq(-0.02,0.02,0.01))+  
 scale\_x\_continuous(limits = c(-0.02,0.02), breaks = seq(-0.02,0.02,0.01))+  
 xlab("Loadings 1")+  
 ylab("Loadings 2")+  
 ggtitle("1dpi loadings plot")  
  
#PC1 vs PC3  
ggplot(data = one\_dpi\_pca\_loadings)+  
 geom\_point(mapping = aes(x = X.PC1, y = X.PC3))+  
 scale\_y\_continuous(limits = c(-0.02,0.03), breaks = seq(-0.02,0.03,0.01))+  
 scale\_x\_continuous(limits = c(-0.02,0.02), breaks = seq(-0.02,0.02,0.01))+  
 xlab("Loadings 1")+  
 ylab("Loadings 3")+  
 ggtitle("1dpi loadings plot")  
  
#PC2 vs PC3  
ggplot(data = one\_dpi\_pca\_loadings)+  
 geom\_point(mapping = aes(x = X.PC2, y = X.PC3))+  
 scale\_y\_continuous(limits = c(-0.02,0.03), breaks = seq(-0.02,0.03,0.01))+  
 scale\_x\_continuous(limits = c(-0.02,0.02), breaks = seq(-0.02,0.02,0.01))+  
 xlab("Loadings 2")+  
 ylab("Loadings 3")+  
 ggtitle("1dpi loadings plot")  
  
dev.off()

#PCA SCREE PLOT

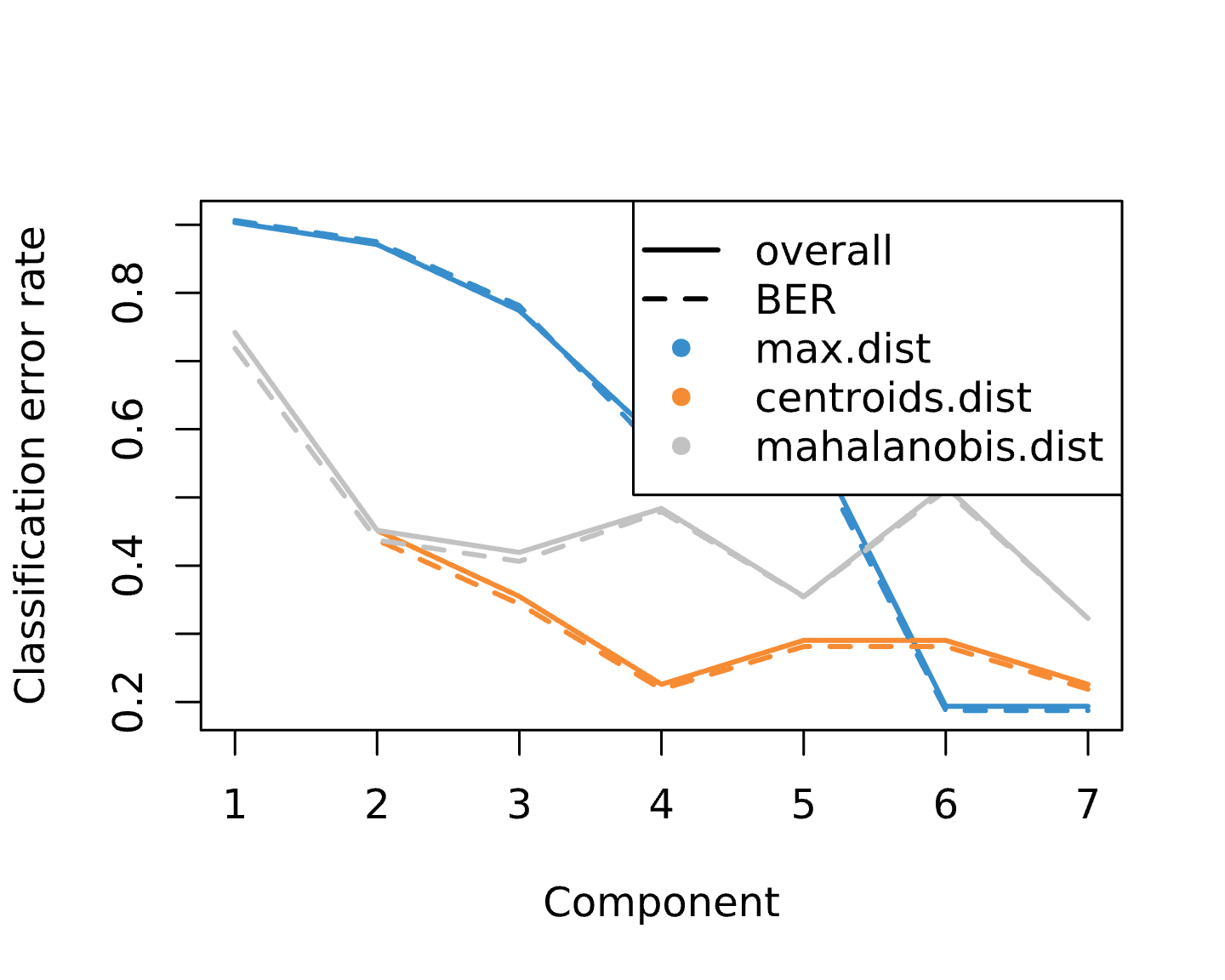
#set up and plot scree plot  
one\_dpi\_expl\_var <- as.data.frame(pca\_one\_dpi$prop\_expl\_var) #extract explained variance  
names(one\_dpi\_expl\_var)[names(one\_dpi\_expl\_var) == 'X'] <- 'percent\_var' #rename percent col  
one\_dpi\_expl\_var <- cbind(PC\_index = 1:14, one\_dpi\_expl\_var) #add PC\_index column  
one\_dpi\_expl\_var$percent\_var <- one\_dpi\_expl\_var$percent\_var \* 100 #convert decimal to percent  
one\_dpi\_expl\_var$cum\_var <- cumsum(one\_dpi\_expl\_var$percent\_var) #calculate cumulative variance explained  
one\_dpi\_expl\_var$percent\_var <- round(one\_dpi\_expl\_var$percent\_var, digits = 1) #round to 1dp  
one\_dpi\_expl\_var$cum\_var <- round(one\_dpi\_expl\_var$cum\_var, digits = 1)  
one\_dpi\_expl\_var

## PC\_index percent\_var cum\_var  
## PC1 1 15.5 15.5  
## PC2 2 10.7 26.2  
## PC3 3 9.3 35.5  
## PC4 4 6.1 41.6  
## PC5 5 5.1 46.7  
## PC6 6 4.7 51.4  
## PC7 7 4.4 55.8  
## PC8 8 3.7 59.5  
## PC9 9 3.3 62.7  
## PC10 10 3.1 65.8  
## PC11 11 3.0 68.8  
## PC12 12 2.7 71.4  
## PC13 13 2.5 74.0  
## PC14 14 2.4 76.3

scree\_plot <- ggplot(data = one\_dpi\_expl\_var)+  
 geom\_point(mapping = aes(x = PC\_index, y = percent\_var))+  
 geom\_line(mapping = aes(x = PC\_index, y = percent\_var), col = "blue")+  
 geom\_text(aes(x=PC\_index, y=percent\_var, label=percent\_var), nudge\_y = 4)+  
 geom\_point(mapping = aes(x = PC\_index, y = cum\_var))+  
 geom\_line(mapping = aes(x = PC\_index, y = cum\_var), col = "#117733")+  
 geom\_text(aes(x=PC\_index, y=cum\_var, label=cum\_var), nudge\_y = 4)+  
 scale\_x\_continuous(breaks = seq(0,15, by = 1))+  
 xlab("PC index")+  
 ylab("Percentage of variance explained (%)")+  
 theme(panel.grid.major.y = element\_blank(), panel.grid.minor.y = element\_blank(), panel.grid.minor.x = element\_blank(), panel.grid.major.x = element\_line(linetype = "dashed", colour = "grey"))+  
 ggtitle("1dpi Scree Plot")  
  
ggsave(filename = "one\_dpi\_PCA\_scree\_plot.pdf", plot = scree\_plot, width = 15, height = 9, path = "Deseq\_analysis/one\_dpi\_out", device = "pdf")

#PLSDA TUNING

initial\_plsda\_one\_dpi <- mixOmics::plsda(rlog\_expression\_all\_one\_dpi, rlog\_classes\_all\_one\_dpi, ncomp = 7)  
  
perf\_initial\_plsda\_one\_dpi <- perf(initial\_plsda\_one\_dpi,   
 validation = "loo", #use leave one out cross-validation  
 auc = TRUE, #calculate area under the curve performance of model  
 dist = "all")   
  
plot(perf\_initial\_plsda\_one\_dpi)



print(perf\_initial\_plsda\_one\_dpi$choice.ncomp)

## max.dist centroids.dist mahalanobis.dist  
## overall 6 4 7  
## BER 7 7 7

#PLSDA SCORES

final\_plsda\_one\_dpi <- mixOmics::plsda(rlog\_expression\_all\_one\_dpi, rlog\_classes\_all\_one\_dpi, ncomp = 7)   
  
#set up pdf to save graphs  
plsda <- "Deseq\_analysis/one\_dpi\_out/one\_dpi\_PLSDA\_scores\_plots.pdf"  
pdf(file = plsda)  
  
#plot final pls model for comps 1 and 2  
plotIndiv(  
 final\_plsda\_one\_dpi, comp = c(1,2), group = rlog\_classes\_all\_one\_dpi, ind.names = FALSE,   
 ellipse = TRUE, ellipse.level = 0.95, legend = TRUE, title = '1dpi PLSDA: Component 1 vs 2',  
 col.per.group = c("blue", "#117733", "#AA4499", "#DDCC77", "#33bbee", "#44AA99", "#882255", "#999933"),   
 pch = c(19,19,19,19,17,17,17,17), cex = 3, pch.levels = c("mock", "mock", "mock", "mock", "PM", "PM", "PM", "PM"),  
 size.xlabel = rel(2), size.ylabel = rel(2), size.axis = rel(1.5))  
  
#plot final pls model for comps 1 and 3  
plotIndiv(  
 final\_plsda\_one\_dpi, comp = c(1,3), group = rlog\_classes\_all\_one\_dpi, ind.names = FALSE,   
 ellipse = TRUE, ellipse.level = 0.95, legend = TRUE, title = '1dpi PLSDA: Component 1 vs 3',   
 col.per.group = c("blue", "#117733", "#AA4499", "#DDCC77", "#33bbee", "#44AA99", "#882255", "#999933"),   
 pch = c(19,19,19,19,17,17,17,17), cex = 3, pch.levels = c("mock", "mock", "mock", "mock", "PM", "PM", "PM", "PM"),  
 size.xlabel = rel(2), size.ylabel = rel(2), size.axis = rel(1.5))  
  
#plot final pls model for comps 2 and 3  
plotIndiv(  
 final\_plsda\_one\_dpi, comp = c(2,3), group = rlog\_classes\_all\_one\_dpi, ind.names = FALSE,   
 ellipse = TRUE, ellipse.level = 0.95, legend = TRUE, title = '1dpi PLSDA: Component 2 vs 3',  
 col.per.group = c("blue", "#117733", "#AA4499", "#DDCC77", "#33bbee", "#44AA99", "#882255", "#999933"),   
 pch = c(19,19,19,19,17,17,17,17), cex = 3, pch.levels = c("mock", "mock", "mock", "mock", "PM", "PM", "PM", "PM"),  
 size.xlabel = rel(2), size.ylabel = rel(2), size.axis = rel(1.5))  
  
dev.off()

#PLSDA LOADINGS

#set up pdf to save graphs  
plsda\_loadings <- "Deseq\_analysis/one\_dpi\_out/one\_dpi\_PLSDA\_loadings.pdf"  
pdf(file = plsda\_loadings)  
  
#extract loadings data  
one\_dpi\_plsda\_loadings <- as.data.frame(final\_plsda\_one\_dpi$loadings[[1]])  
  
#PC1 vs PC2  
ggplot(data = one\_dpi\_plsda\_loadings)+  
 geom\_point(mapping = aes(x = comp1, y = comp2))+  
 scale\_y\_continuous(limits = c(-0.02,0.02), breaks = seq(-0.02,0.02,0.01))+  
 scale\_x\_continuous(limits = c(-0.02,0.02), breaks = seq(-0.02,0.02,0.01))+  
 xlab("Loadings 1")+  
 ylab("Loadings 2")+  
 ggtitle("1dpi loadings plot")  
  
#PC1 vs PC3  
ggplot(data = one\_dpi\_plsda\_loadings)+  
 geom\_point(mapping = aes(x = comp1, y = comp3))+  
 scale\_y\_continuous(limits = c(-0.03,0.03), breaks = seq(-0.03,0.03,0.01))+  
 scale\_x\_continuous(limits = c(-0.02,0.02), breaks = seq(-0.02,0.02,0.01))+  
 xlab("Loadings 1")+  
 ylab("Loadings 3")+  
 ggtitle("1dpi loadings plot")  
  
#PC2 vs PC3  
ggplot(data = one\_dpi\_plsda\_loadings)+  
 geom\_point(mapping = aes(x = comp2, y = comp3))+  
 scale\_y\_continuous(limits = c(-0.03,0.03), breaks = seq(-0.03,0.03,0.01))+  
 scale\_x\_continuous(limits = c(-0.02,0.02), breaks = seq(-0.02,0.02,0.01))+  
 xlab("Loadings 2")+  
 ylab("Loadings 3")+  
 ggtitle("1dpi loadings plot")  
  
dev.off()

#PLSDA AUROC

#set up pdf to save graphs  
plsda\_auroc <- "Deseq\_analysis/one\_dpi\_out/one\_dpi\_PLSDA\_AUROC.pdf"  
pdf(file = plsda\_auroc)  
  
#check performance of final model using aucroc curve  
auc.plsda.one.dpi\_1 <- auroc(final\_plsda\_one\_dpi, roc.comp = 1, print = TRUE)

## $Comp1  
## AUC p-value  
## Mock\_Water vs Other(s) 0.9630 0.003216  
## Mock\_BABA vs Other(s) 0.7222 0.157300  
## Mock\_JA vs Other(s) 0.7685 0.087480  
## Mock\_SA vs Other(s) 0.6574 0.316500  
## PM\_Water vs Other(s) 0.8148 0.045130  
## PM\_BABA vs Other(s) 0.6481 0.345800  
## PM\_JA vs Other(s) 0.7407 0.125500  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp2  
## AUC p-value  
## Mock\_Water vs Other(s) 0.9630 0.003216  
## Mock\_BABA vs Other(s) 0.7037 0.194800  
## Mock\_JA vs Other(s) 0.8426 0.029240  
## Mock\_SA vs Other(s) 0.9630 0.003216  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 0.9259 0.006717  
## PM\_JA vs Other(s) 0.7130 0.175300  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp3  
## AUC p-value  
## Mock\_Water vs Other(s) 0.9722 0.002654  
## Mock\_BABA vs Other(s) 0.7130 0.175300  
## Mock\_JA vs Other(s) 0.8796 0.015690  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 0.8611 0.021560  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp4  
## AUC p-value  
## Mock\_Water vs Other(s) 1.0000 0.001463  
## Mock\_BABA vs Other(s) 0.7407 0.125500  
## Mock\_JA vs Other(s) 0.9167 0.008010  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 1.0000 0.001463  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp5  
## AUC p-value  
## Mock\_Water vs Other(s) 1.0000 0.001463  
## Mock\_BABA vs Other(s) 0.7407 0.125500  
## Mock\_JA vs Other(s) 0.9352 0.005614  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 1.0000 0.001463  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp6  
## AUC p-value  
## Mock\_Water vs Other(s) 1.0000 0.001463  
## Mock\_BABA vs Other(s) 0.7778 0.077100  
## Mock\_JA vs Other(s) 1.0000 0.001463  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 1.0000 0.001463  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp7  
## AUC p-value  
## Mock\_Water vs Other(s) 1 0.001463  
## Mock\_BABA vs Other(s) 1 0.001463  
## Mock\_JA vs Other(s) 1 0.001463  
## Mock\_SA vs Other(s) 1 0.001463  
## PM\_Water vs Other(s) 1 0.001463  
## PM\_BABA vs Other(s) 1 0.001463  
## PM\_JA vs Other(s) 1 0.001463  
## PM\_SA vs Other(s) 1 0.005012

auc.plsda.one.dpi\_1to2 <- auroc(final\_plsda\_one\_dpi, roc.comp = 2, print = TRUE)

## $Comp1  
## AUC p-value  
## Mock\_Water vs Other(s) 0.9630 0.003216  
## Mock\_BABA vs Other(s) 0.7222 0.157300  
## Mock\_JA vs Other(s) 0.7685 0.087480  
## Mock\_SA vs Other(s) 0.6574 0.316500  
## PM\_Water vs Other(s) 0.8148 0.045130  
## PM\_BABA vs Other(s) 0.6481 0.345800  
## PM\_JA vs Other(s) 0.7407 0.125500  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp2  
## AUC p-value  
## Mock\_Water vs Other(s) 0.9630 0.003216  
## Mock\_BABA vs Other(s) 0.7037 0.194800  
## Mock\_JA vs Other(s) 0.8426 0.029240  
## Mock\_SA vs Other(s) 0.9630 0.003216  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 0.9259 0.006717  
## PM\_JA vs Other(s) 0.7130 0.175300  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp3  
## AUC p-value  
## Mock\_Water vs Other(s) 0.9722 0.002654  
## Mock\_BABA vs Other(s) 0.7130 0.175300  
## Mock\_JA vs Other(s) 0.8796 0.015690  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 0.8611 0.021560  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp4  
## AUC p-value  
## Mock\_Water vs Other(s) 1.0000 0.001463  
## Mock\_BABA vs Other(s) 0.7407 0.125500  
## Mock\_JA vs Other(s) 0.9167 0.008010  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 1.0000 0.001463  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp5  
## AUC p-value  
## Mock\_Water vs Other(s) 1.0000 0.001463  
## Mock\_BABA vs Other(s) 0.7407 0.125500  
## Mock\_JA vs Other(s) 0.9352 0.005614  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 1.0000 0.001463  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp6  
## AUC p-value  
## Mock\_Water vs Other(s) 1.0000 0.001463  
## Mock\_BABA vs Other(s) 0.7778 0.077100  
## Mock\_JA vs Other(s) 1.0000 0.001463  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 1.0000 0.001463  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp7  
## AUC p-value  
## Mock\_Water vs Other(s) 1 0.001463  
## Mock\_BABA vs Other(s) 1 0.001463  
## Mock\_JA vs Other(s) 1 0.001463  
## Mock\_SA vs Other(s) 1 0.001463  
## PM\_Water vs Other(s) 1 0.001463  
## PM\_BABA vs Other(s) 1 0.001463  
## PM\_JA vs Other(s) 1 0.001463  
## PM\_SA vs Other(s) 1 0.005012

auc.plsda.one.dpi\_1to3 <- auroc(final\_plsda\_one\_dpi, roc.comp = 3, print = TRUE) # AUROC for all 3 components

## $Comp1  
## AUC p-value  
## Mock\_Water vs Other(s) 0.9630 0.003216  
## Mock\_BABA vs Other(s) 0.7222 0.157300  
## Mock\_JA vs Other(s) 0.7685 0.087480  
## Mock\_SA vs Other(s) 0.6574 0.316500  
## PM\_Water vs Other(s) 0.8148 0.045130  
## PM\_BABA vs Other(s) 0.6481 0.345800  
## PM\_JA vs Other(s) 0.7407 0.125500  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp2  
## AUC p-value  
## Mock\_Water vs Other(s) 0.9630 0.003216  
## Mock\_BABA vs Other(s) 0.7037 0.194800  
## Mock\_JA vs Other(s) 0.8426 0.029240  
## Mock\_SA vs Other(s) 0.9630 0.003216  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 0.9259 0.006717  
## PM\_JA vs Other(s) 0.7130 0.175300  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp3  
## AUC p-value  
## Mock\_Water vs Other(s) 0.9722 0.002654  
## Mock\_BABA vs Other(s) 0.7130 0.175300  
## Mock\_JA vs Other(s) 0.8796 0.015690  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 0.8611 0.021560  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp4  
## AUC p-value  
## Mock\_Water vs Other(s) 1.0000 0.001463  
## Mock\_BABA vs Other(s) 0.7407 0.125500  
## Mock\_JA vs Other(s) 0.9167 0.008010  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 1.0000 0.001463  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp5  
## AUC p-value  
## Mock\_Water vs Other(s) 1.0000 0.001463  
## Mock\_BABA vs Other(s) 0.7407 0.125500  
## Mock\_JA vs Other(s) 0.9352 0.005614  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 1.0000 0.001463  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp6  
## AUC p-value  
## Mock\_Water vs Other(s) 1.0000 0.001463  
## Mock\_BABA vs Other(s) 0.7778 0.077100  
## Mock\_JA vs Other(s) 1.0000 0.001463  
## Mock\_SA vs Other(s) 1.0000 0.001463  
## PM\_Water vs Other(s) 1.0000 0.001463  
## PM\_BABA vs Other(s) 1.0000 0.001463  
## PM\_JA vs Other(s) 1.0000 0.001463  
## PM\_SA vs Other(s) 1.0000 0.005012  
##   
## $Comp7  
## AUC p-value  
## Mock\_Water vs Other(s) 1 0.001463  
## Mock\_BABA vs Other(s) 1 0.001463  
## Mock\_JA vs Other(s) 1 0.001463  
## Mock\_SA vs Other(s) 1 0.001463  
## PM\_Water vs Other(s) 1 0.001463  
## PM\_BABA vs Other(s) 1 0.001463  
## PM\_JA vs Other(s) 1 0.001463  
## PM\_SA vs Other(s) 1 0.005012

dev.off()

#DESEQ ANALYSIS

#mock  
mock\_one\_dpi\_DESeq\_out <- DESeq(mock\_one\_dpi\_raw\_gene\_data)  
  
#PM  
PM\_one\_dpi\_DESeq\_out <- DESeq(PM\_one\_dpi\_raw\_gene\_data)

#EXTRACT DEGs

#mock  
  
factor='condition'  
reference='Mock\_Water'  
treatment='Mock\_BABA'  
DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water <- results(mock\_one\_dpi\_DESeq\_out, contrast = c(factor, treatment, reference))   
  
factor='condition'  
reference='Mock\_Water'  
treatment='Mock\_JA'  
DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water <- results(mock\_one\_dpi\_DESeq\_out, contrast = c(factor, treatment, reference))  
  
factor='condition'  
reference='Mock\_Water'  
treatment='Mock\_SA'  
DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water <- results(mock\_one\_dpi\_DESeq\_out, contrast = c(factor, treatment, reference))   
  
#PM  
  
factor='condition'  
reference='PM\_Water'  
treatment='PM\_BABA'  
DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water <- results(PM\_one\_dpi\_DESeq\_out, contrast = c(factor, treatment, reference))   
  
factor='condition'  
reference='PM\_Water'  
treatment='PM\_JA'  
DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water <- results(PM\_one\_dpi\_DESeq\_out, contrast = c(factor, treatment, reference))   
  
factor='condition'  
reference='PM\_Water'  
treatment='PM\_SA'  
DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water <- results(PM\_one\_dpi\_DESeq\_out, contrast = c(factor, treatment, reference))

#SELECT PRIMED DEGs

#note sig threshold of padj <= 0.05  
  
#select significant DEGs mock  
sig\_DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water <- subset(DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water, DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water$padj <= 0.05)  
sig\_DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water <- subset(DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water, DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water$padj <= 0.05)  
sig\_DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water <- subset(DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water, DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water$padj <= 0.05)  
  
print(paste(nrow(sig\_DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water), "significant mock BABA DEGs at 1dpi"))

## [1] "366 significant mock BABA DEGs at 1dpi"

print(paste(nrow(sig\_DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water), "significant mock JA DEGs at 1dpi"))

## [1] "1114 significant mock JA DEGs at 1dpi"

print(paste(nrow(sig\_DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water), "significant mock SA DEGs at 1dpi"))

## [1] "3431 significant mock SA DEGs at 1dpi"

#select significant DEGs PM  
sig\_DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water <- subset(DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water, DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water$padj <= 0.05)  
sig\_DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water <- subset(DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water, DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water$padj <= 0.05)  
sig\_DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water <- subset(DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water, DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water$padj <= 0.05)  
  
print(paste(nrow(sig\_DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water), "significant PM BABA DEGs at 1dpi"))

## [1] "6458 significant PM BABA DEGs at 1dpi"

print(paste(nrow(sig\_DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water), "significant PM JA DEGs at 1dpi"))

## [1] "3955 significant PM JA DEGs at 1dpi"

print(paste(nrow(sig\_DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water), "significant PM SA DEGs at 1dpi"))

## [1] "5095 significant PM SA DEGs at 1dpi"

#select row names from mock treatment  
BABA\_rows\_to\_remove <- rownames(sig\_DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water)  
JA\_rows\_to\_remove <- rownames(sig\_DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water)  
SA\_rows\_to\_remove <- rownames(sig\_DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water)  
  
#select DEGs only present during infection (i.e., PM DEGs not shared with mock) = primed DEGs  
BABA\_primed\_all\_1dpi <- sig\_DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water[!rownames(sig\_DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water) %in% BABA\_rows\_to\_remove,]  
JA\_primed\_all\_1dpi <- sig\_DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water[!rownames(sig\_DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water) %in% JA\_rows\_to\_remove,]  
SA\_primed\_all\_1dpi <- sig\_DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water[!rownames(sig\_DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water) %in% SA\_rows\_to\_remove,]  
  
print(paste(nrow(BABA\_primed\_all\_1dpi), "total primed BABA DEGs at 1dpi"))

## [1] "6307 total primed BABA DEGs at 1dpi"

print(paste(nrow(JA\_primed\_all\_1dpi), "total primed JA DEGs at 1dpi"))

## [1] "3671 total primed JA DEGs at 1dpi"

print(paste(nrow(SA\_primed\_all\_1dpi), "total primed SA DEGs at 1dpi"))

## [1] "4006 total primed SA DEGs at 1dpi"

#select row names for primed DEGs  
BABA\_primed\_rows\_to\_remove <- rownames(BABA\_primed\_all\_1dpi)  
JA\_primed\_rows\_to\_remove <- rownames(JA\_primed\_all\_1dpi)  
SA\_primed\_rows\_to\_remove <- rownames(SA\_primed\_all\_1dpi)  
  
#select primed DEGs exclusive to each treatment  
BABA\_primed\_exclusive\_1dpi <- BABA\_primed\_all\_1dpi[!rownames(BABA\_primed\_all\_1dpi) %in% JA\_primed\_rows\_to\_remove,]  
BABA\_primed\_exclusive\_1dpi <- BABA\_primed\_exclusive\_1dpi[!rownames(BABA\_primed\_exclusive\_1dpi) %in% SA\_primed\_rows\_to\_remove,]  
JA\_primed\_exclusive\_1dpi <- JA\_primed\_all\_1dpi[!rownames(JA\_primed\_all\_1dpi) %in% BABA\_primed\_rows\_to\_remove,]  
JA\_primed\_exclusive\_1dpi <- JA\_primed\_exclusive\_1dpi[!rownames(JA\_primed\_exclusive\_1dpi) %in% SA\_primed\_rows\_to\_remove,]  
SA\_primed\_exclusive\_1dpi <- SA\_primed\_all\_1dpi[!rownames(SA\_primed\_all\_1dpi) %in% BABA\_primed\_rows\_to\_remove,]  
SA\_primed\_exclusive\_1dpi <- SA\_primed\_exclusive\_1dpi[!rownames(SA\_primed\_exclusive\_1dpi) %in% JA\_primed\_rows\_to\_remove,]  
  
print(paste(nrow(BABA\_primed\_exclusive\_1dpi), "BABA exclusive primed DEGs at 1dpi"))

## [1] "2914 BABA exclusive primed DEGs at 1dpi"

print(paste(nrow(JA\_primed\_exclusive\_1dpi), "JA exclusive primed DEGs at 1dpi"))

## [1] "952 JA exclusive primed DEGs at 1dpi"

print(paste(nrow(SA\_primed\_exclusive\_1dpi), "SA exclusive primed DEGs at 1dpi"))

## [1] "1638 SA exclusive primed DEGs at 1dpi"

#EXTRACT UP/DOWN PRIMED DEGs fc1

#MOCK UP  
UP\_BABA\_mock\_1dpi <- subset(DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water, DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water$log2FoldChange > 1 & DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water$padj <= 0.05)  
  
UP\_JA\_mock\_1dpi <- subset(DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water, DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water$log2FoldChange > 1 & DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water$padj <= 0.05)  
  
UP\_SA\_mock\_1dpi <- subset(DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water, DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water$log2FoldChange > 1 & DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water$padj <= 0.05)  
  
#MOCK DOWN  
DOWN\_BABA\_mock\_1dpi <- subset(DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water, DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water$log2FoldChange < -1 & DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water$padj <= 0.05)  
  
DOWN\_JA\_mock\_1dpi <- subset(DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water, DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water$log2FoldChange < -1 & DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water$padj <= 0.05)  
  
DOWN\_SA\_mock\_1dpi <- subset(DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water, DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water$log2FoldChange < -1 & DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water$padj <= 0.05)  
  
#PM UP  
UP\_BABA\_PM\_1dpi <- subset(DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water, DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water$log2FoldChange > 1 & DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water$padj <= 0.05)  
  
UP\_JA\_PM\_1dpi <- subset(DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water, DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water$log2FoldChange > 1 & DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water$padj <= 0.05)  
  
UP\_SA\_PM\_1dpi <- subset(DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water, DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water$log2FoldChange > 1 & DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water$padj <= 0.05)  
  
#PM DOWN  
DOWN\_BABA\_PM\_1dpi <- subset(DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water, DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water$log2FoldChange < -1 & DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water$padj <= 0.05)  
  
DOWN\_JA\_PM\_1dpi <- subset(DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water, DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water$log2FoldChange < -1 & DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water$padj <= 0.05)  
  
DOWN\_SA\_PM\_1dpi <- subset(DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water, DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water$log2FoldChange < -1 & DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water$padj <= 0.05)  
  
#PRIMED UP ALL - FOR VENN  
UP\_BABA\_primed\_all\_1dpi <- subset(BABA\_primed\_all\_1dpi, BABA\_primed\_all\_1dpi$log2FoldChange > 1 & BABA\_primed\_all\_1dpi$padj <= 0.05)   
  
UP\_JA\_primed\_all\_1dpi <- subset(JA\_primed\_all\_1dpi, JA\_primed\_all\_1dpi$log2FoldChange > 1 & JA\_primed\_all\_1dpi$padj <= 0.05)   
  
UP\_SA\_primed\_all\_1dpi <- subset(SA\_primed\_all\_1dpi, SA\_primed\_all\_1dpi$log2FoldChange > 1 & SA\_primed\_all\_1dpi$padj <= 0.05)   
  
#PRIMED DOWN ALL - FOR VENN  
DOWN\_BABA\_primed\_all\_1dpi <- subset(BABA\_primed\_all\_1dpi, BABA\_primed\_all\_1dpi$log2FoldChange < -1 & BABA\_primed\_all\_1dpi$padj <= 0.05)   
  
DOWN\_JA\_primed\_all\_1dpi <- subset(JA\_primed\_all\_1dpi, JA\_primed\_all\_1dpi$log2FoldChange < -1 & JA\_primed\_all\_1dpi$padj <= 0.05)   
  
DOWN\_SA\_primed\_all\_1dpi <- subset(SA\_primed\_all\_1dpi, SA\_primed\_all\_1dpi$log2FoldChange < -1 & SA\_primed\_all\_1dpi$padj <= 0.05)   
  
#PRIMED UP EXCLUSIVE - FOR SUMMARY TABLE  
UP\_BABA\_primed\_exclusive\_1dpi <- subset(BABA\_primed\_exclusive\_1dpi, BABA\_primed\_exclusive\_1dpi$log2FoldChange > 1 & BABA\_primed\_exclusive\_1dpi$padj <= 0.05)  
  
UP\_JA\_primed\_exclusive\_1dpi <- subset(JA\_primed\_exclusive\_1dpi, JA\_primed\_exclusive\_1dpi$log2FoldChange > 1 & JA\_primed\_exclusive\_1dpi$padj <= 0.05)  
  
UP\_SA\_primed\_exclusive\_1dpi <- subset(SA\_primed\_exclusive\_1dpi, SA\_primed\_exclusive\_1dpi$log2FoldChange > 1 & SA\_primed\_exclusive\_1dpi$padj <= 0.05)  
  
#PRIMED DOWN EXCLUSIVE - FOR SUMMARY TABLE  
DOWN\_BABA\_primed\_exclusive\_1dpi <- subset(BABA\_primed\_exclusive\_1dpi, BABA\_primed\_exclusive\_1dpi$log2FoldChange < -1 & BABA\_primed\_exclusive\_1dpi$padj <= 0.05)  
  
DOWN\_JA\_primed\_exclusive\_1dpi <- subset(JA\_primed\_exclusive\_1dpi, JA\_primed\_exclusive\_1dpi$log2FoldChange < -1 & JA\_primed\_exclusive\_1dpi$padj <= 0.05)  
  
DOWN\_SA\_primed\_exclusive\_1dpi <- subset(SA\_primed\_exclusive\_1dpi, SA\_primed\_exclusive\_1dpi$log2FoldChange < -1 & SA\_primed\_exclusive\_1dpi$padj <= 0.05)

#DATA SUMMARY TABLE

one\_dpi\_summary\_data <-  
 data.frame('Timepoint' =   
 c("1 dpi", "1 dpi", "1 dpi", "1 dpi", "1 dpi", "1 dpi", "1dpi", "1dpi", "1dpi"),   
 'Comparison' =   
 c("Mock BABA vs Mock Water", "Mock SA vs Mock Water", "Mock JA vs Mock Water",   
 "PM BABA vs PM Water", "PM SA vs PM Water", "PM JA vs PM Water",  
 "BABA primed", "SA primed", "JA primed"),  
 'All\_significant\_DEGs' =   
 c(nrow(sig\_DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water), nrow(sig\_DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water), nrow(sig\_DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water),  
 nrow(sig\_DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water), nrow(sig\_DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water), nrow(sig\_DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water),  
 nrow(BABA\_primed\_exclusive\_1dpi), nrow(SA\_primed\_exclusive\_1dpi), nrow(JA\_primed\_exclusive\_1dpi)),  
 'Upregulated\_DEGs' =  
 c(nrow(UP\_BABA\_mock\_1dpi), nrow(UP\_SA\_mock\_1dpi), nrow(UP\_JA\_mock\_1dpi),  
 nrow(UP\_BABA\_PM\_1dpi), nrow(UP\_SA\_PM\_1dpi), nrow(UP\_JA\_PM\_1dpi),  
 nrow(UP\_BABA\_primed\_exclusive\_1dpi), nrow(UP\_SA\_primed\_exclusive\_1dpi), nrow(UP\_JA\_primed\_exclusive\_1dpi)),  
 'Downregulated\_DEGs' =  
 c(nrow(DOWN\_BABA\_mock\_1dpi), nrow(DOWN\_SA\_mock\_1dpi), nrow(DOWN\_JA\_mock\_1dpi),  
 nrow(DOWN\_BABA\_PM\_1dpi), nrow(DOWN\_SA\_PM\_1dpi), nrow(DOWN\_JA\_PM\_1dpi),  
 nrow(DOWN\_BABA\_primed\_exclusive\_1dpi), nrow(DOWN\_SA\_primed\_exclusive\_1dpi), nrow(DOWN\_JA\_primed\_exclusive\_1dpi))  
 )  
  
  
#save one dpi summary  
write.csv(one\_dpi\_summary\_data, "Deseq\_analysis/one\_dpi\_out/summary\_data\_1dpi\_fc1.csv", row.names=FALSE)

#VENN DIAGRAMS

#NOTE NEED ALL PRIMED DEGS NOT ONES EXCLUSIVE TO TREATMENT FOR VENN DIAGRAMS  
#sa = yellow, ja = magenta baba = dark-green  
  
#set up pdf to save graphs  
venns <- "Deseq\_analysis/one\_dpi\_out/one\_dpi\_Venn\_diagrams\_fc1.pdf"  
pdf(file = venns)  
  
#all primed DEGs  
Venn\_BABA\_primed\_all\_1dpi <- row.names(BABA\_primed\_all\_1dpi)  
Venn\_SA\_primed\_all\_1dpi <- row.names(SA\_primed\_all\_1dpi)  
Venn\_JA\_primed\_all\_1dpi <- row.names(JA\_primed\_all\_1dpi)  
  
vennPlot <- venn.diagram(list(Venn\_SA\_primed\_all\_1dpi, Venn\_JA\_primed\_all\_1dpi, Venn\_BABA\_primed\_all\_1dpi), filename = NULL, fill=c("#DDCC77", "#AA4499", "#117733"), print.mode = c("raw", "percent"), alpha=c(0.5,0.5,0.5), cex=2.5, category.names = c(" "," ", " "), main = "All primed DEGs 1dpi", main.cex = 2, ext.text = TRUE, ext.percent = 0.1, ext.pos = 3, ext.length = 0.7, ext.dist = -0.02)  
grid.newpage()  
grid.draw(vennPlot)  
  
#upregulated genes  
Venn\_UP\_BABA\_primed <- row.names(UP\_BABA\_primed\_all\_1dpi)  
Venn\_UP\_SA\_primed <- row.names(UP\_SA\_primed\_all\_1dpi)  
Venn\_UP\_JA\_primed <- row.names(UP\_JA\_primed\_all\_1dpi)  
  
vennPlot <- venn.diagram(list(Venn\_UP\_SA\_primed, Venn\_UP\_JA\_primed, Venn\_UP\_BABA\_primed), filename = NULL, fill=c("#DDCC77", "#AA4499", "#117733"), print.mode = c("raw", "percent"), alpha=c(0.5,0.5,0.5), cex=2.5, category.names = c(" "," ", " "), main = "Upregulated primed genes 1dpi", main.cex = 2, ext.text = TRUE, ext.percent = 0.1, ext.pos = 3, ext.length = 0.7, ext.dist = -0.02)  
grid.newpage()  
grid.draw(vennPlot)  
  
#downregulated genes  
Venn\_DOWN\_BABA\_primed <- row.names(DOWN\_BABA\_primed\_all\_1dpi)  
Venn\_DOWN\_SA\_primed <- row.names(DOWN\_SA\_primed\_all\_1dpi)  
Venn\_DOWN\_JA\_primed <- row.names(DOWN\_JA\_primed\_all\_1dpi)  
  
vennPlot <- venn.diagram(list(Venn\_DOWN\_SA\_primed, Venn\_DOWN\_JA\_primed, Venn\_DOWN\_BABA\_primed), filename = NULL, fill=c("#DDCC77", "#AA4499", "#117733"), print.mode = c("raw", "percent"), alpha=c(0.5,0.5,0.5), cex=2.5, category.names = c(" "," ", " "), main = "Downregulated primed genes 1dpi", main.cex = 2, ext.text = TRUE, ext.percent = 0.1, ext.pos = 3, ext.length = 0.7, ext.dist = -0.02)  
grid.newpage()  
grid.draw(vennPlot)  
  
dev.off()

#GENE CLUSTERING HEATMAP

#set up pdf to save graphs  
heatmaps <- "Deseq\_analysis/one\_dpi\_out/one\_dpi\_heatmaps.pdf"  
pdf(file = heatmaps)  
  
#select significant DEGs for BABA mock  
significant\_BABA\_mock\_DEGs\_heatmap <- DEGs\_1dpi\_mock\_BABA\_vs\_mock\_Water$padj <= 0.05   
  
sum(significant\_BABA\_mock\_DEGs\_heatmap, na.rm = TRUE) #check number = same as before

## [1] 366

#select significant DEGs for SA mock  
significant\_SA\_mock\_DEGs\_heatmap <- DEGs\_1dpi\_mock\_SA\_vs\_mock\_Water$padj <= 0.05   
  
sum(significant\_SA\_mock\_DEGs\_heatmap, na.rm = TRUE) #check number = same as before

## [1] 3431

#select significant DEGs for JA mock  
significant\_JA\_mock\_DEGs\_heatmap <- DEGs\_1dpi\_mock\_JA\_vs\_mock\_Water$padj <= 0.05   
  
sum(significant\_JA\_mock\_DEGs\_heatmap, na.rm = TRUE) #check number = same as before

## [1] 1114

#select significant DEGs for BABA PM  
significant\_BABA\_PM\_DEGs\_heatmap <- DEGs\_1dpi\_PM\_BABA\_vs\_PM\_Water$padj <= 0.05   
  
sum(significant\_BABA\_PM\_DEGs\_heatmap, na.rm = TRUE) #check number = same as before

## [1] 6458

#select significant DEGs for SA PM  
significant\_SA\_PM\_DEGs\_heatmap <- DEGs\_1dpi\_PM\_SA\_vs\_PM\_Water$padj <= 0.05   
  
sum(significant\_SA\_PM\_DEGs\_heatmap, na.rm = TRUE) #check number = same as before

## [1] 5095

#select significant DEGs for JA PM  
significant\_JA\_PM\_DEGs\_heatmap <- DEGs\_1dpi\_PM\_JA\_vs\_PM\_Water$padj <= 0.05   
  
sum(significant\_JA\_PM\_DEGs\_heatmap, na.rm = TRUE) #check number = same as before

## [1] 3955

#combine significant DEGs into single vector = merges duplicates  
significant\_all\_DEGs\_heatmap <- base::Reduce('|', list(significant\_BABA\_mock\_DEGs\_heatmap, significant\_SA\_mock\_DEGs\_heatmap, significant\_JA\_mock\_DEGs\_heatmap, significant\_BABA\_PM\_DEGs\_heatmap, significant\_SA\_PM\_DEGs\_heatmap, significant\_JA\_PM\_DEGs\_heatmap))  
  
sum(significant\_all\_DEGs\_heatmap, na.rm = TRUE) #check number

## [1] 11598

#create df  
heatmap\_df <- data.frame(assay(rlog\_all\_one\_dpi\_raw\_gene\_data)[which(significant\_all\_DEGs\_heatmap),])  
  
#remove rows containing all zeros  
heatmap\_df[heatmap\_df==0] <- NA #convert 0 to NAs  
heatmap\_df <- na.omit(heatmap\_df) #remove NAs  
  
#calculate mean  
m = apply(heatmap\_df, 1, mean, na.rm = T)  
  
#calculate sd  
sd = apply(heatmap\_df, 1, sd, na.rm = T)  
  
#scale  
scaled\_heatmap\_df <- (heatmap\_df - m)/sd  
  
# #note scaling is based on the scale\_rows function within pheatmap which performs the following   
# scale\_rows = function(x){  
# m = apply(x, 1, mean, na.rm = T)  
# s = apply(x, 1, sd, na.rm = T)  
# return((x - m) / s)}  
  
#set up gene clustering dendrogram  
gene\_dend <- as.dendrogram(hclust(dist(scaled\_heatmap\_df)))   
#dist calculates distances using euclidean distances as default   
#hclust performs hierarchical clustering using the complete linkage method by default  
  
gene\_dend #height = 10.80835

## 'dendrogram' with 2 branches and 11598 members total, at height 10.80835

#try different heights to cut tree to find optimal number of clusters to colour code  
gene\_dend1 <- color\_branches(gene\_dend, h=10.8, col = brewer.pal(8,"Dark2"))  
plot(gene\_dend1, main = "cut height 10.8") #3 clusters = too few  
  
#based on first plot a cut around 10.5 should be about right  
gene\_dend2 <- color\_branches(gene\_dend, h=10.5, col = brewer.pal(8,"Dark2"))  
plot(gene\_dend2, main = "cut height 10.5") #7 clusters = probably about right  
  
gene\_dend3 <- color\_branches(gene\_dend, h=10.4, col = brewer.pal(8,"Dark2"))  
plot(gene\_dend3, main = "cut height 10.4") #10 clusters = probably too many  
  
gene\_dend4 <- color\_branches(gene\_dend, h=10.6, col = brewer.pal(8,"Dark2"))  
plot(gene\_dend4, main = "cut height 10.6") #5 clusters = probaly too few  
  
#save optimal number of clusters as k, based on trying both 10.6 and 10.5 use 10.5  
cut <- cutree(gene\_dend, h=10.5)  
k <- as.numeric(max(levels(as.factor(cut))))  
  
#set up sample annotations for plots  
sample\_annotations <- all\_one\_dpi\_sample\_info[,3]  
conditions <- HeatmapAnnotation(Condition = sample\_annotations, col = list(Condition = c("Mock\_Water" = "blue", "Mock\_BABA" = "#117733", "Mock\_JA" = "#AA4499", "Mock\_SA" = "#DDCC77", "PM\_Water" = "#33bbee", "PM\_BABA" = "#44AA99", "PM\_JA" = "#882255", "PM\_SA" = "#999933")), gp = gpar(col = "black"))  
  
#set up column split to separate mock and infected in plots  
column\_split = rep("Mock", 31)  
column\_split[17:31] = "PM"  
  
#plot heatmap  
htmp1 <- ComplexHeatmap::Heatmap(scaled\_heatmap\_df, cluster\_columns = FALSE,   
 cluster\_rows = gene\_dend2, #use optimal cluster dendrogram for colour coding  
 split = k, #split up dendrogram to number clusters  
 col = viridis(7), show\_row\_names = FALSE,   
 heatmap\_legend\_param = list(title = NULL, direction = "horizontal", at = c(-5,-2.5,0,2.5,5)),   
 top\_annotation = conditions, show\_column\_names = FALSE, column\_split = column\_split)

## `use\_raster` is automatically set to TRUE for a matrix with more than  
## 2000 rows. You can control `use\_raster` argument by explicitly setting  
## TRUE/FALSE to it.  
##   
## Set `ht\_opt$message = FALSE` to turn off this message.

draw(htmp1, heatmap\_legend\_side = "bottom")  
  
dev.off()

#SAVE COUNTS + BABA PRIMING DATA FOR INTEGRATION

#EXTRACT COUNTS FOR 1DPI  
  
#select counts data  
counts\_1dpi <- as.data.frame(counts(all\_one\_dpi\_raw\_gene\_data))   
  
#add column of gene names  
counts\_1dpi <- cbind(gene = row.names(counts\_1dpi), counts\_1dpi)   
  
#save counts data  
write.csv(counts\_1dpi,   
 file = "Deseq\_analysis/one\_dpi\_out/gene\_counts\_1dpi.csv",  
 row.names = FALSE)  
  
#SELECT GENES EXCLUSIVE TO BABA PRIMING  
BABA\_primed\_1dpi <- data.frame(gene = rownames(BABA\_primed\_exclusive\_1dpi))   
  
#LOAD COUNTS DATA FOR 0 AND 2DPI  
  
#load 0dpi counts  
counts\_0dpi <- read.csv(file = "Deseq\_analysis/zero\_dpi\_out/gene\_counts\_0dpi.csv")  
  
#load 2dpi counts  
counts\_2dpi <- read.csv(file = "Deseq\_analysis/two\_dpi\_out/gene\_counts\_2dpi.csv")  
  
#FILTER NON-PRIMED GENES FROM COUNTS DATA  
#here we select only genes in gene\_counts data that are exclusive to BABA priming at 1dpi  
  
#0dpi  
BABA\_primed\_counts\_0dpi <- subset(counts\_0dpi, gene %in% BABA\_primed\_1dpi$gene)   
  
#1dpi  
BABA\_primed\_counts\_1dpi <- subset(counts\_1dpi, gene %in% BABA\_primed\_1dpi$gene)   
  
#2dpi  
BABA\_primed\_counts\_2dpi <- subset(counts\_2dpi, gene %in% BABA\_primed\_1dpi$gene)   
  
#MERGE, PREP AND SAVE FINAL TABLE  
#note done starting with 2dpi so columns in time order in final df  
  
#append 2dpi gene counts to BABA primed 1dpi gene names  
BABA\_primed\_gene\_names\_and\_counts\_1dpi <- as.data.frame(append(BABA\_primed\_counts\_2dpi, BABA\_primed\_1dpi))  
  
#append 1dpi gene counts to BABA primed 1dpi gene names  
BABA\_primed\_gene\_names\_and\_counts\_1dpi <- as.data.frame(append(BABA\_primed\_counts\_1dpi, BABA\_primed\_gene\_names\_and\_counts\_1dpi))  
  
#append 0dpi gene counts to BABA primed 1dpi gene names  
BABA\_primed\_gene\_names\_and\_counts\_1dpi <- as.data.frame(append(BABA\_primed\_counts\_0dpi, BABA\_primed\_gene\_names\_and\_counts\_1dpi))   
  
#remove gene name columns added from gene\_counts  
BABA\_primed\_gene\_names\_and\_counts\_1dpi <- subset(BABA\_primed\_gene\_names\_and\_counts\_1dpi, select = c(-17, -49, -81))  
  
#save final df as csv  
write.csv(BABA\_primed\_gene\_names\_and\_counts\_1dpi, file = "Deseq\_analysis/one\_dpi\_out/BABA\_primed\_genes\_1dpi\_counts.csv", row.names = FALSE)

#SAVE DATA FOR GO TERM ANALYSIS

load(file = "GO\_analysis/R\_topGO/Initial\_GO\_input\_data\_fc1.RData") #load data save for 0dpi  
  
#save 0dpi + 1dpi primed data  
save(zero\_dpi\_raw\_gene\_data,  
 DEGs\_0dpi\_BABA,   
 DEGs\_0dpi\_SA,   
 DEGs\_0dpi\_JA,  
 BABA\_primed\_all\_1dpi,  
 SA\_primed\_all\_1dpi,  
 JA\_primed\_all\_1dpi,  
 DOWN\_BABA\_primed\_all\_1dpi,  
 DOWN\_SA\_primed\_all\_1dpi,  
 DOWN\_JA\_primed\_all\_1dpi,  
 UP\_BABA\_primed\_all\_1dpi,  
 UP\_SA\_primed\_all\_1dpi,  
 UP\_JA\_primed\_all\_1dpi,  
 file = "GO\_analysis/R\_topGO/Initial\_GO\_input\_data\_fc1.RData")