Interaction of O.Sativa with its pathogen analyzed through Differential Gene Expression

Clemente Calabrese

2023-06-05

Pre-Analytical Steps

Loading libraries

Loading Experimental Data

```
colData <- read.table("input_Data/experiment_Design.tsv", sep = "\t", header = TRUE, row.names = 1) %>%
  filter(`Analysed` == "Yes", #I exclude samples that were discarded
         `Sample.Characteristic.developmental.stage.` == "seedling")  %>% #I exclude flowering stage sa
  transmute(`resistance` = as.factor(`Factor.Value.phenotype.`),
            `infection` = as.factor(`Factor.Value.infect.`)) #then I leave only the factors of interest

# recoding the levels for quicker access
levels(colData$resistance) <- c("R", "S")
levels(colData$infection) <- c("TRT", "CTRL")


countData <- read.delim("input_Data/rawCounts.tsv", sep = "\t", header = TRUE, row.names = 1) %>%
  select(all_of(row.names(colData))) #select all the samples remaining from colData
```

How many genes are we taking into account?

```
dim(countData)
```

```
## [1] 38866    18
```

Generating multiple datasets

Since we're going to make pairwise analyses, we will need different datasets to account for different contrasts we're going to make:

One dds grouping all inoculated samples (Resistant vs Susceptible phenotypes)

```
colData_infected <- colData[colData$infection == "TRT",]

countData_infected <- read.delim("input_Data/rawCounts.tsv",
```

```
                                  sep = "\t",
                                  header = TRUE,
                                  row.names = 1) %>%
  select(all_of(row.names(colData_infected)))

dim(colData_infected)[1] == dim(countData_infected)[2]
```

```
## [1] TRUE
```

One dds grouping all resistant samples (control vs inoculated R)

```
colData_R <- colData[colData$resistance == "R",]

countData_R <- read.delim("input_Data/rawCounts.tsv",
                          sep = "\t",
                          header = TRUE,
                          row.names = 1) %>%
  select(all_of(row.names(colData_R)))

dim(colData_R)[1] == dim(countData_R)[2]
```

```
## [1] TRUE
```

One dds grouping all susceptible samples (control vs inoculated R)

```
colData_S <- colData[colData$resistance == "S",]

countData_S <- read.delim("input_Data/rawCounts.tsv",
                          sep = "\t",
                          header = TRUE,
                          row.names = 1) %>%
  select(all_of(row.names(colData_S)))

dim(colData_S)[1] == dim(countData_S)[2]
```

```
## [1] TRUE
```

Checking if the two vectors contain the same elements and in the same order:

```
# this checks if they're the same vector
all(rownames(colData) == colnames(countData))
```

```
## [1] TRUE
```

```
all(rownames(colData_infected) == colnames(countData_infected))
```

```
## [1] TRUE
```

```
all(rownames(colData_R) == colnames(countData_R))
```

```
## [1] TRUE
```

```
all(rownames(colData_S) == colnames(countData_S))
```

```
## [1] TRUE
```

Choosing a suitable design formula

```
design <- ~ resistance + infection
```

Contrast n.1 - Resistant vs Susceptible

In this dataset there are infected samples belonging both to

Creating the DESeq Dataset

```
dds_infected <- DESeqDataSetFromMatrix(countData = countData_infected,
                                colData = colData_infected,
                                design = ~ resistance)
dim(dds_infected)
```

```
## [1] 38866    12
```

Excluding the low-expression genes from our analysis:

```
# only use the genes actually expressed
dds_infected <- dds_infected[rowSums(dds_infected@assays@data@listData[["counts"]]) > 1,]
dim(dds_infected)
```

```
## [1] 32250    12
```

How many genes did we discard?

```
dim(countData_infected)[1] - dim(dds_infected)[1]
```

```
## [1] 6616
```

Run DESeq and retrieve the results:

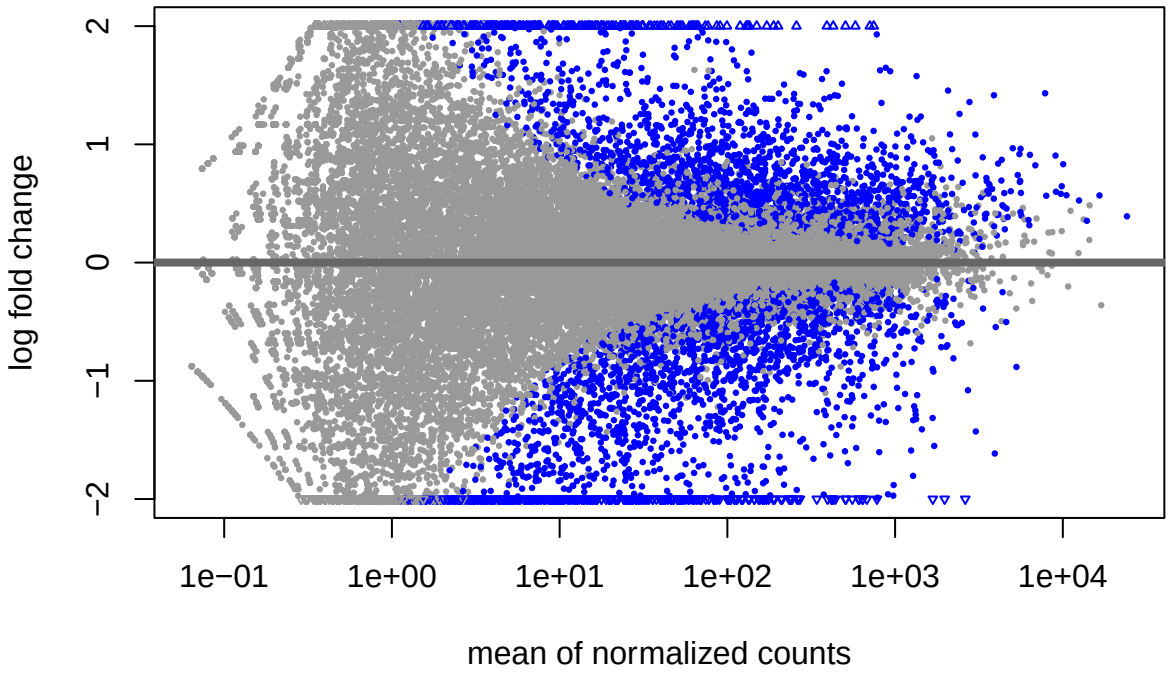
```
dds_infected <- DESeq(dds_infected)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
results_RvS <- results(dds_infected, contrast=c("resistance", "R", "S"))
```

Inspect quality of the results with an MA plot:

```
plotMA(results_RvS, ylim=c(-2,2))
```

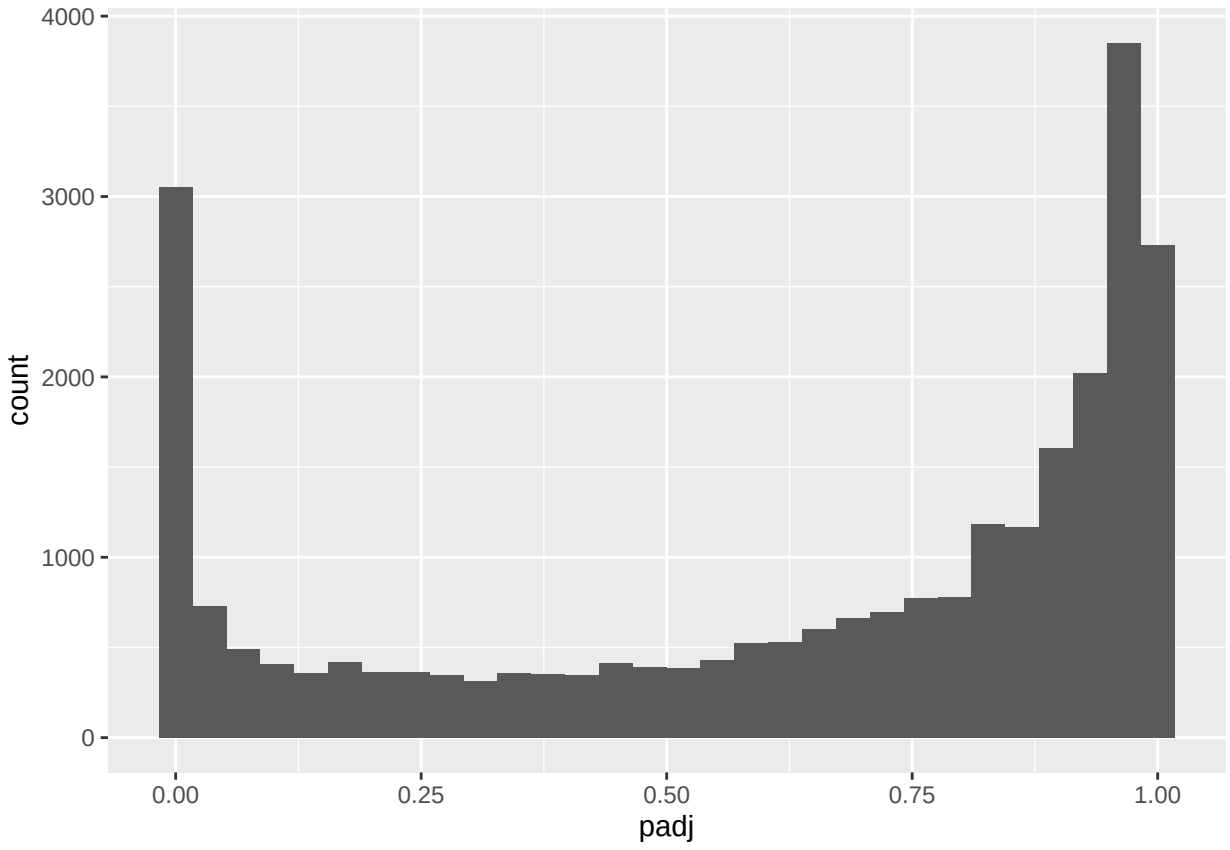


Inspect pvalue distribution:

```
ggplot(as.data.frame(results_RvS), aes(x = padj)) +
  geom_histogram()
```

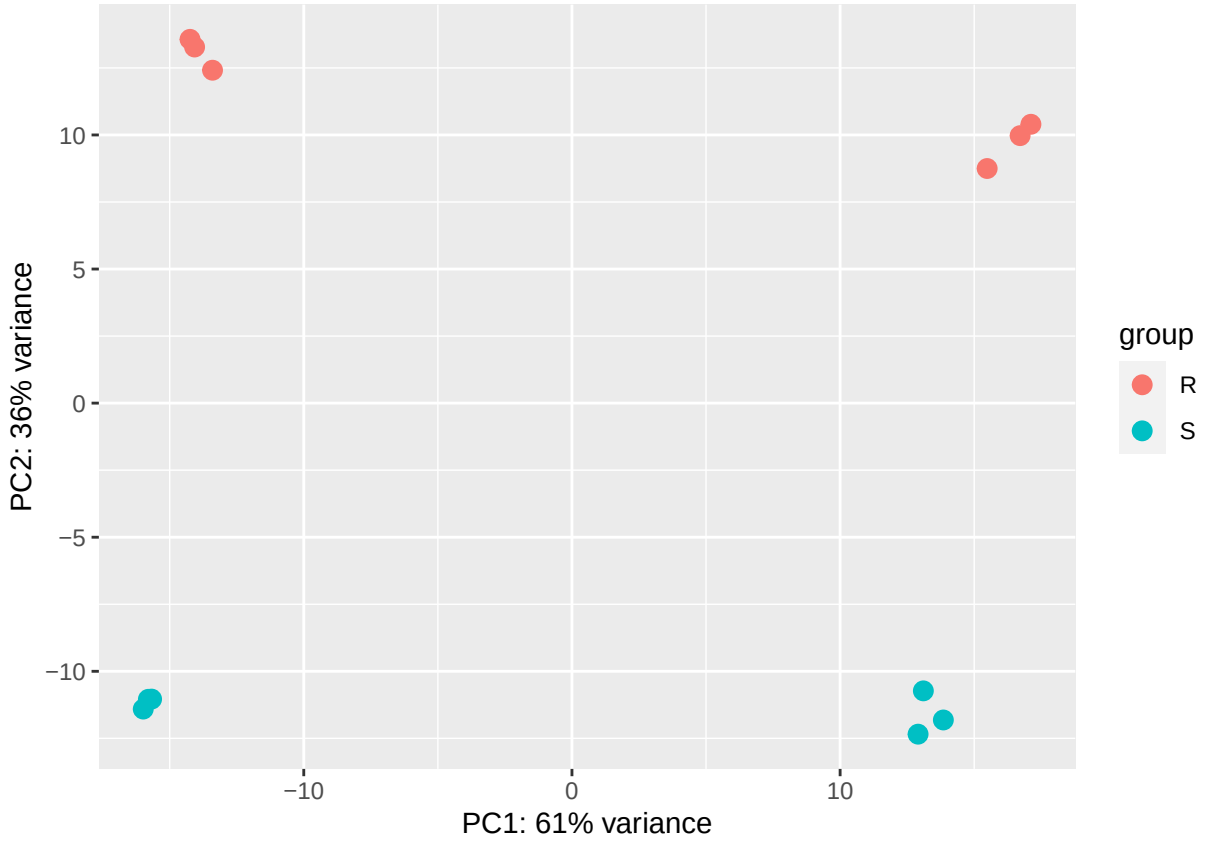
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 5630 rows containing non-finite values (`stat_bin()`).
```



Inspect PCA plot:

```
#first i need the normalized counts
dds_infected_n <- rlog(dds_infected)
DESeq2::plotPCA(object = dds_infected_n, intgroup = "resistance")
```



Inspect magnitudes of DEGs with a Volcano Plot:

```
#da aggiungere
```

```
summary(results_RvS)
```

```
##
## out of 32250 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 2218, 6.9%
## LFC < 0 (down)     : 2213, 6.9%
## outliers [1]       : 2, 0.0062%
## low counts [2]     : 5628, 17%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Filter diff.expressed genes:

```
  # in the paper it is stated that
  # FDR = 0.05 and FoldChange = 2 were the cutoffs

RvS <- as.data.frame(results_RvS) %>%
  filter(!is.na(.$padj)) %>%
```

```
  filter(.$padj < .05) %>%
  filter(.$log2FoldChange > 1 |.$log2FoldChange < 1 )


# How many genes are we left with?
dim(RvS)[1]
```

```
## [1] 3755
```

GO enrichment analysis

```
# translating transcript location to HGNC symbols with GeneKitR
RvS$entrez <- transId(rownames(RvS),
                      transTo = "ENTREZID",
                      org="osativa",
                      unique=TRUE,
                      keepNA=TRUE)[,2]
```

```
## Some ID occurs one-to-many match, like "Os01g0835900, Os02g0684500, Os03g0119900"...
```

```
## 62.74% genes are mapped to entrezid
```

```
# filtering the genes that mapped to multiple (or none) symbols
RvS <- filter(RvS, !is.na(RvS$entrez))

#How many genes are we left with?
dim(RvS)
```

```
## [1] 2356    7
```

Retrieve OrgDb data to map IDs to GO terms

I found the correct OrgDb by querying AnnotationHub

```
# query(ah,'org.Oryza_sativa_Japonica_Group.eg.sqlite')
```

It gave me the OrgDb name (AH107685) that I can use to access the annotation.

```
ah <- AnnotationHub()
os.db <- ah[["AH107685"]]
```

```
## loading from cache
```

```
## Caricamento del pacchetto richiesto: AnnotationDbi
```

```
##
## Caricamento pacchetto: 'AnnotationDbi'
```

```
## Il seguente oggetto è mascherato da 'package:clusterProfiler':
##
##     select

## Il seguente oggetto è mascherato da 'package:dplyr':
##
##     select
```

Run the GO analysis

```
GO_BP <- enrichGO(RvS$entrez, OrgDb = os.db,
                  keyType = "ENTREZID", ont = "BP")
GO_MF <- enrichGO(RvS$entrez, OrgDb = os.db,
                  keyType = "ENTREZID", ont = "MF")
GO_CC <- enrichGO(RvS$entrez, OrgDb = os.db,
                  keyType = "ENTREZID", ont = "CC")
```