DGE_Acute_Experiment

Beatriz Silva

2023-11-06

Loading libraries

```
library(edgeR)
```

```
## Loading required package: limma
```

```
library(limma)
library(DESeq2)
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following object is masked from 'package:limma':
##
##     plotMA

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

##
## Attaching package: 'S4Vectors'
```

```
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:Biobase':
##
##     combine

## The following object is masked from 'package:matrixStats':
##
##     count

## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect

## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union

## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(pheatmap)
library(ggplot2)
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space

## The following object is masked from 'package:S4Vectors':
##
##     space

## The following object is masked from 'package:stats':
##
##     lowess
```

Read files

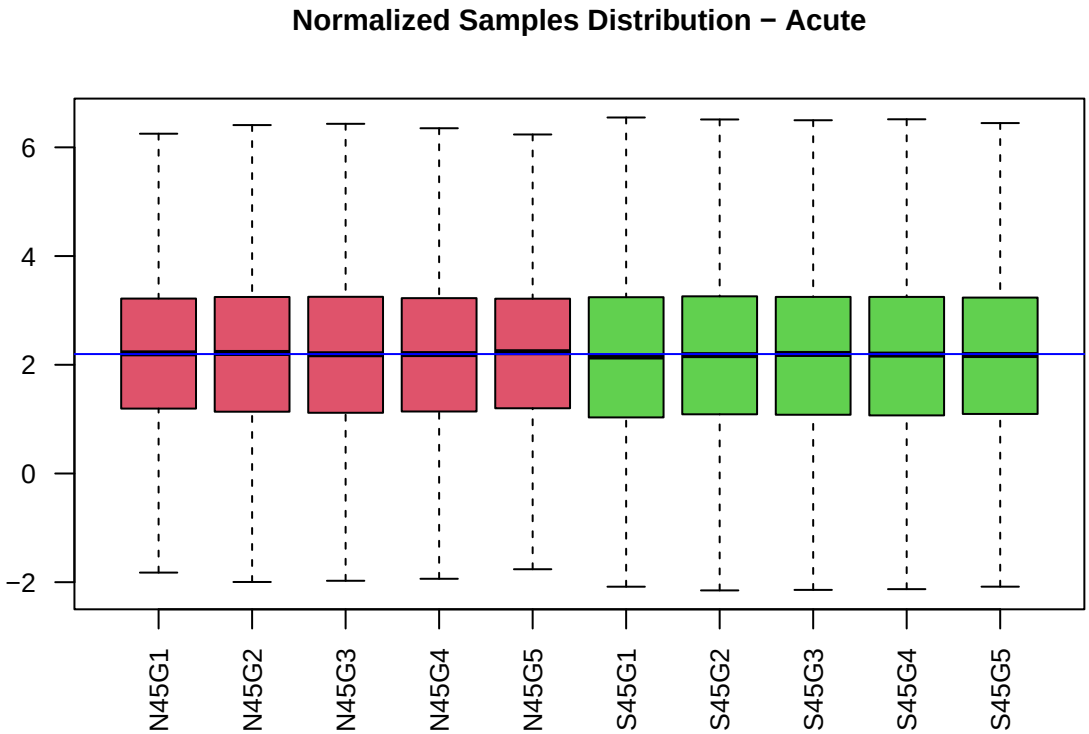
```r
setwd(paste0("C:/Users/USUARIO/Desktop/Ude_analysis/Gene_expression/",
             "DGE_Chronic_Experiments/DGE_Analysis"))

data_counts <- read.table("countData_acute.txt", header = TRUE, sep = "\t",
                          row.names = 1, check.names = FALSE)
sample_info <- read.table("colData_acute.txt", header = TRUE, sep = "\t",
                          row.names = 1, check.names = FALSE)

group <- factor(paste(sample_info$pop, sample_info$temperature, sep="."))
sample_info <- cbind(sample_info, group = group)

# DGEList object

y <- DGEList(counts = data_counts, group = group)

cpm_count <- cpm(y)

# thresholds

cpm_Val <- 1   # CPM value threshold
gThreshold <- 5   # At least number of samples threshold

thresholds <- rowSums(cpm_count > cpm_Val) >= gThreshold

y <- calcNormFactors(y, lib.size = T, method = "TMM")

# Apply filtering

y_filter <- y[thresholds,]
```

```
# Get CPM values for filtered data (y_filter)

cpm_count_filtered <- cpm(y_filter)

dim(cpm_count_filtered)
```

```
## [1] 18509    10
```

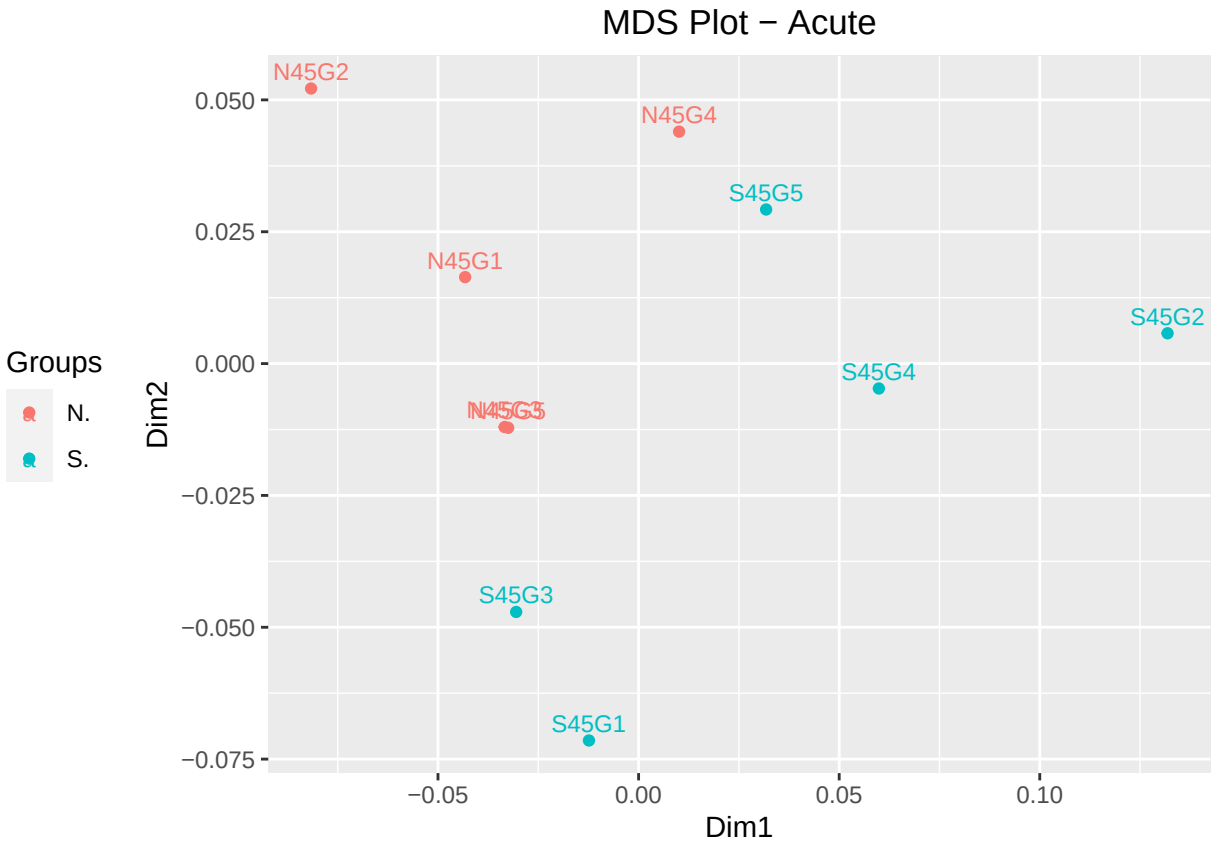
Boxplot with sample distribution

```
# Normalized Samples Distribution - Acute

statuscol <- as.numeric(factor(sample_info$group)) + 1
log_counts <- log(cpm_count_filtered + 1e-02)

boxplot(log_counts,
        col = statuscol,
        xlab = "", las = 2,
        cex.axis = 0.8,
        outline = FALSE)
abline(h = median(as.matrix(log_counts)), col = "blue")
title("Normalized Samples Distribution - Acute", cex.main = 0.9)
```



MDS plot

```
dist_matrix <- 1 - cor(cpm_count_filtered)
mds_result <- cmdscale(dist_matrix, k = 2)
mds_df <- data.frame(Sample = rownames(mds_result), Dim1 = mds_result[, 1],
                     Dim2 = mds_result[, 2])

ggplot(mds_df, aes(x = Dim1, y = Dim2, color = group, label = Sample)) +
  geom_point() +
  geom_text(vjust = -0.5, hjust = 0.5, size = 3) +
  scale_color_discrete(guide = guide_legend(title = "Groups")) +
  labs(title = "MDS Plot") +
  theme(legend.position = "left",
        legend.justification = "center",
        plot.title = element_text(hjust = 0.5),
        legend.text = element_text(angle = 0)) +
  ggtitle("MDS Plot - Acute")
```



EdgeR

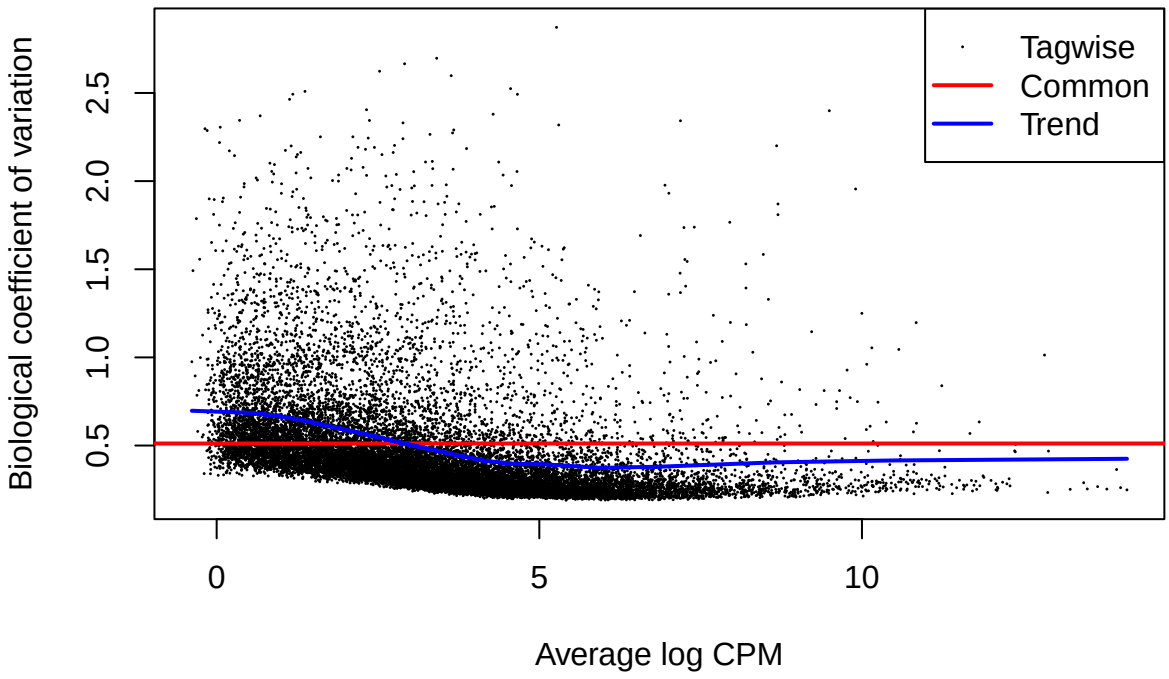
```
# Design matrix for glm approach
```

```
design <- model.matrix(~sample_info$pop)
rownames(design) <- colnames(y_filter)
yf <- estimateDisp(y_filter, design)
fit <- glmFit(yf, design)

glm <- glmLRT(fit, coef=2)

FDR_chronic_all <- topTags(glm, n = Inf, adjust.method = "BH", sort.by = "none")
FDR_chronic_all <- data.frame(gene_id = rownames(FDR_chronic_all),
                              FDR_chronic_all)
filtered_rows_chronic_all <- FDR_chronic_all[FDR_chronic_all$FDR < 0.05
                             & abs(FDR_chronic_all$logFC) > 1, ]
DGEs_EdgeR_North_South <- filtered_rows_chronic_all[, c("gene_id", "logFC",
                                                        "FDR")]

dim(DGEs_EdgeR_North_South)
```

```
## [1] 1427    3
```

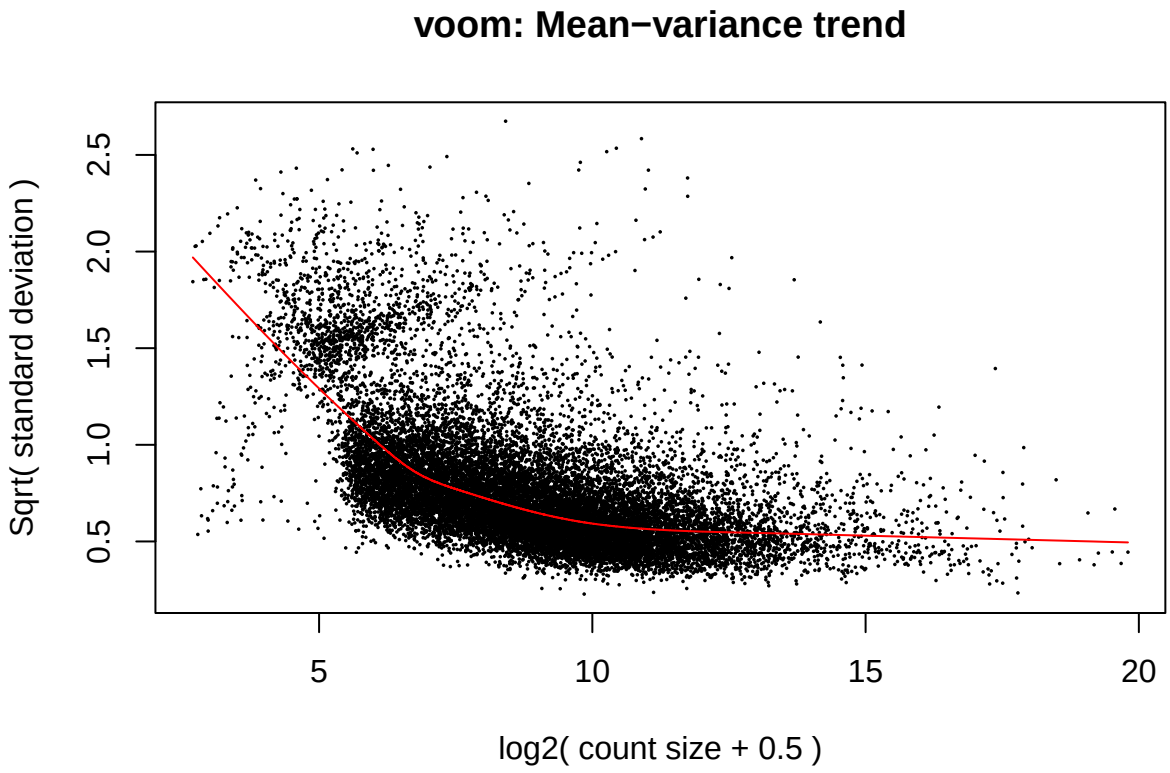
Estimated dispersion plot - EdgeR

```
plotBCV(yf)
```



limma

```
# design matrix

design <- model.matrix(~sample_info$pop)
rownames(design) <- colnames(y_filter)
y <- voom(y_filter, design, normalize="quantile", plot=T)
```




```
# Fitting linear models in limma

fit <- lmFit(y, design)

tmp_all <- contrasts.fit(fit, coef=2)
tmp_all_f <- eBayes(tmp_all)

top.table_all <- topTable(tmp_all_f, number=Inf, adjust="BH", sort.by="none")
FDR_all_df <- data.frame(gene_id = rownames(top.table_all), top.table_all)

DGEs_Limma_North_South <- FDR_all_df[top.table_all$adj.P.Val < 0.05
                                     & abs(top.table_all$logFC ) > 1, ]

dim(DGEs_Limma_North_South)
```

```
## [1] 798   7
```

DESeq2

```
dds <- DESeqDataSetFromMatrix(y_filter$counts, colData = sample_info,
                              design = formula(~pop))
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
dds <- DESeq(dds, test="Wald")
```

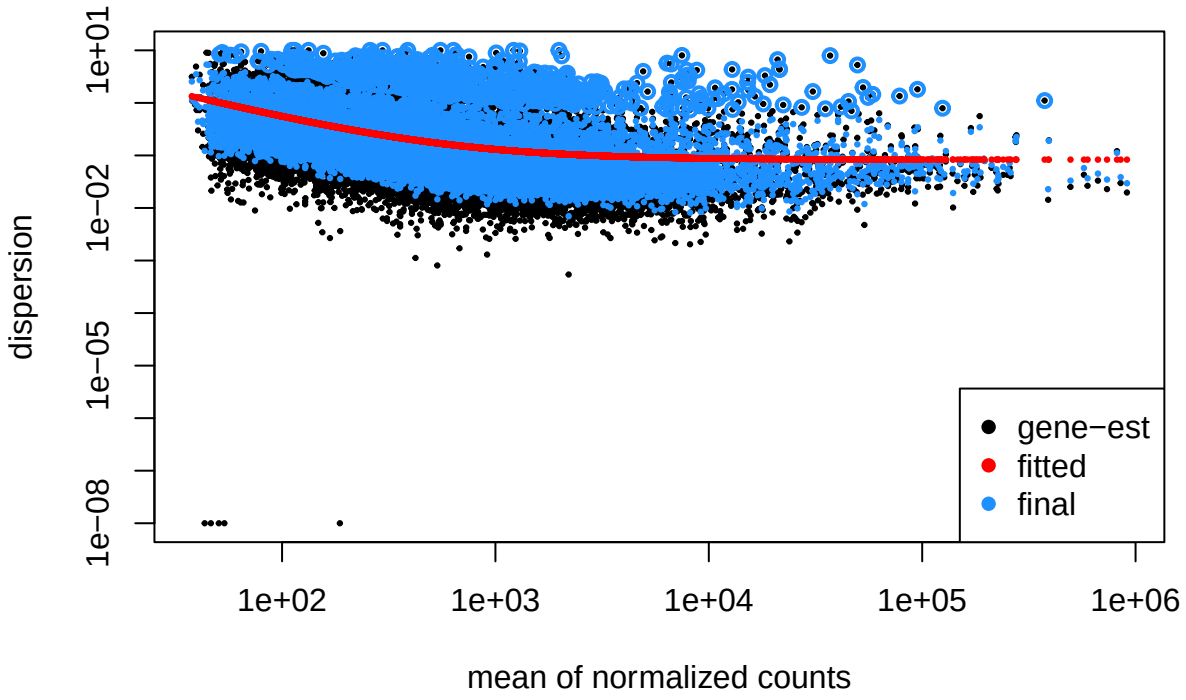
```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
results_all <- results(dds, contrast=c("pop", "S", "N"))

results_all <- na.omit(results_all)
filter_results_all <- results_all[results_all$padj < 0.05
                                  & abs(results_all$log2FoldChange) > 1 ,]

filter_results_all <- cbind(gene_id = rownames(filter_results_all),
                            filter_results_all)
DGEs_DESeq2_North_South <- filter_results_all[, c("gene_id",
                                                  "log2FoldChange", "padj")]


dim(DGEs_DESeq2_North_South)
```

```
## [1] 1583    3
```

Estimated dispersion plot - DESeq2

```
plotDispEsts(dds)
```



Overlap 3 methods - North vs South

```
DESeq2 <- DGEs_DESeq2_North_South$gene_id
EdgeR <- DGEs_EdgeR_North_South$gene_id
Limma <- DGEs_Limma_North_South$gene_id

common_rows <- DESeq2[DESeq2 %in% EdgeR & DESeq2 %in% Limma]

common_rows_list <- unlist(common_rows)

overlap_3_methods <- data.frame(gene_id = common_rows_list)

limma_data <- DGEs_Limma_North_South[, c("logFC", "adj.P.Val")]

limma_data$gene_id <- rownames(limma_data)


North_South_overlap <- merge(overlap_3_methods, limma_data,
                             by = "gene_id", all.x = TRUE)


dim(North_South_overlap)
```

```
## [1] 735   3
```

```
write.table(North_South_overlap, file = "North_South_acute.txt",
            sep = "\t", quote = FALSE, row.names = FALSE)
```

Data for volcano plot

```
colnames(North_South_overlap) <- c("gene_id", "logFC", "adj.P.Val")

North_South_overlap$diffexpressed <- "NO"

North_South_overlap$diffexpressed[North_South_overlap$logFC > 1] <- "Up regulated"

North_South_overlap$diffexpressed[North_South_overlap$logFC < 1] <- "Down regulated"


# number of DGEs - up and down regulated

up_regulated_count <- sum(North_South_overlap$diffexpressed == "Up regulated")
down_regulated_count <- sum(North_South_overlap$diffexpressed == "Down regulated")
cat("Number of up-regulated genes:", up_regulated_count, "\n")
```

```
## Number of up-regulated genes: 373
```

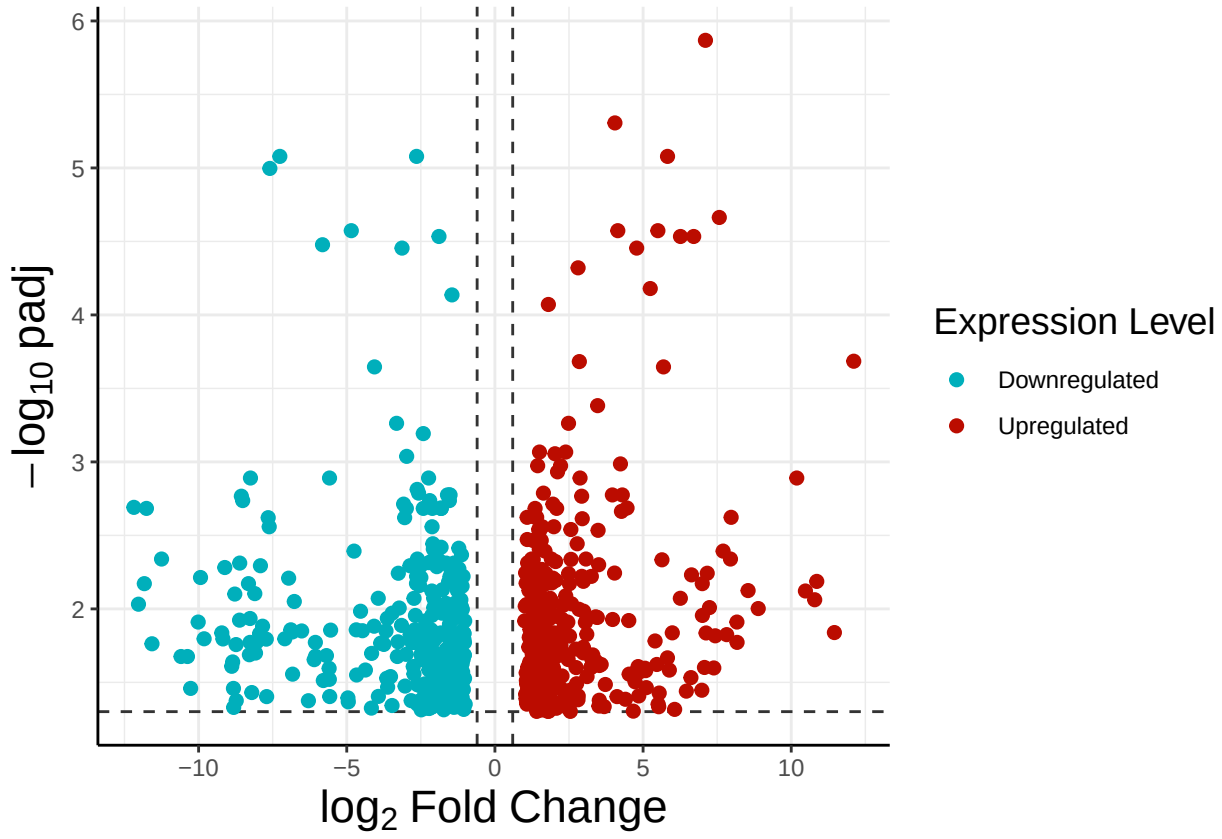
```
cat("Number of down-regulated genes:", down_regulated_count, "\n")
```

```
## Number of down-regulated genes: 362
```

Volcano Plot

```
ggplot(data = North_South_overlap, aes(x = logFC,
                                       y = -log10(adj.P.Val),
                                       col = diffexpressed)) +
  geom_vline(xintercept = c(-0.6, 0.6), col = "gray20", linetype = 'dashed') +
  geom_hline(yintercept = -log10(0.05), col = "gray20", linetype = 'dashed') +
  geom_point(size = 2) +
  scale_color_manual(values = c("#00AFBB", "#bb0c00"),
                     labels = c("Downregulated", "Upregulated")) +
  xlab(expression(log[2]~Fold~Change)) +
  ylab(expression(-log[10]~padj)) +
  theme_bw() +
  theme(
    axis.line = element_line(color = "black", size = 0.5),  # Customize axis lines
    panel.border = element_blank(),  # Remove plot border
    axis.title = element_text(size = 17),  # Set axis title font size
    legend.title = element_text(size = 14),  # Set legend title font size
    legend.position = "right"  # Move the legend to the right
  ) +
  guides(
    color = guide_legend(title = "Expression Level")  # Customize legend title
  )
```

```
## Warning: The ‘size‘ argument of ‘element_line()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
output_data_North_South <- cpm_count_filtered[, grepl("^N45|^S45",
                                                      colnames(cpm_count_filtered))]

# Add the 'gene_id' column to the output_data matrix
data_gene_id_25 <- cbind(gene_id = rownames(output_data_North_South),
                         output_data_North_South)

# Add the name of the first column: gene_id
names(data_gene_id_25) <- c("gene_id", names(data_gene_id_25)[-1])

# merge the data frames based on the "gene_id" column
merged_df_25 <- merge(data_gene_id_25, North_South_overlap[,
                                                           c("gene_id", "logFC",
                                                             "adj.P.Val")],
                      by = "gene_id")

# Selecting the desired columns
final_table_25 <- merged_df_25[, c("gene_id", "N45G1", "N45G2", "N45G3",
                                   "N45G4", "N45G5", "S45G1", "S45G2",
```

```
                                    "S45G3", "S45G4", "S45G5")]

# Convert character matrix to numeric matrix element-wise
numeric_matrix_North_South <- final_table_25[, c( "N45G1", "N45G2",
                                                  "N45G3", "N45G4", "N45G5",
                                                  "S45G1", "S45G2", "S45G3",
                                                  "S45G4", "S45G5")]
numeric_matrix_North_South <- apply(numeric_matrix_North_South, 2, as.numeric)  # Convert each column t

# Set row names
rownames(numeric_matrix_North_South) <- final_table_25$gene_id

# Set column names
colnames(numeric_matrix_North_South) <- c("N45G1", "N45G2", "N45G3", "N45G4",
                                          "N45G5", "S45G1", "S45G2", "S45G3",
                                          "S45G4", "S45G5")

head(numeric_matrix_North_South)
```

```
##                      N45G1     N45G2     N45G3     N45G4     N45G5     S45G1
## UdeG00000000189 15.278722 12.802260 13.712445 19.799505 15.828621  3.437342
## UdeG00000000298  3.614089  3.106122  1.369633  3.688307  5.104583  6.095553
## UdeG00000000307 42.481773 37.252477 39.719361 28.215552 39.630396 21.151109
## UdeG00000000473 16.360785 14.859016 14.389204 18.475067 16.917207  4.949772
## UdeG00000000543 23.134496 41.365989 47.808253 37.939272 42.440123 78.554716
## UdeG00000000544  4.782716  8.185051 11.392125  5.113335  6.001931 16.659650
##                      S45G2     S45G3     S45G4     S45G5
## UdeG00000000189  4.486623  9.223455  6.566057  8.505603
## UdeG00000000298  7.128943  9.671195  5.148137  8.001228
## UdeG00000000307 10.214605  7.539950 11.583309 14.122511
## UdeG00000000473  6.206791  1.271583  8.311188  1.260939
## UdeG00000000543 77.584889 69.686333 79.687058 86.431604
## UdeG00000000544 20.766150 12.948656 19.829055 16.598536
```

```
dim(numeric_matrix_North_South)
```

```
## [1] 735  10
```

Heatmap

```
gene_labels <- rownames(numeric_matrix_North_South)
condition_labels <- colnames(numeric_matrix_North_South)

heatmap.2(numeric_matrix_North_South,
          scale = "row",
          trace = "none",
          col = colorRampPalette(colors = c("#273D82", "white", "#C75218"))(100),
          main = "North vs South",
          cex.main = 1,
          Colv = TRUE,
```

```
            hclustfun = function(c) hclust(c, method = "average"),
            dendrogram = "column",
            key = TRUE,
            key.title = " ",
            key.xlab = "Row Z-Score",
            margins = c(5, 15),
            cexRow = 0.8,
            cexCol = 0.8,
            Rowv = TRUE,
            labRow = FALSE
)
```