

# DGE\_Chronic\_Experiment

Beatrix Silva

2023-10-26

## 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, uname

## 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

```

```

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

library(grid)

```

## Read files

```

setwd("C:/Users/USUARIO/Desktop/Ude_analysis/Gene_expression/DGE_Chronic_Experiments/DGE_Analysis")

data_counts <- read.table("countData_with_N20G4.txt", header = TRUE, sep = "\t", row.names = 1, check.names = FALSE)
sample_info <- read.table("colData_with_N20G4.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] 22696      30

## [1] 22696      30

```

## Boxplot with sample distribution

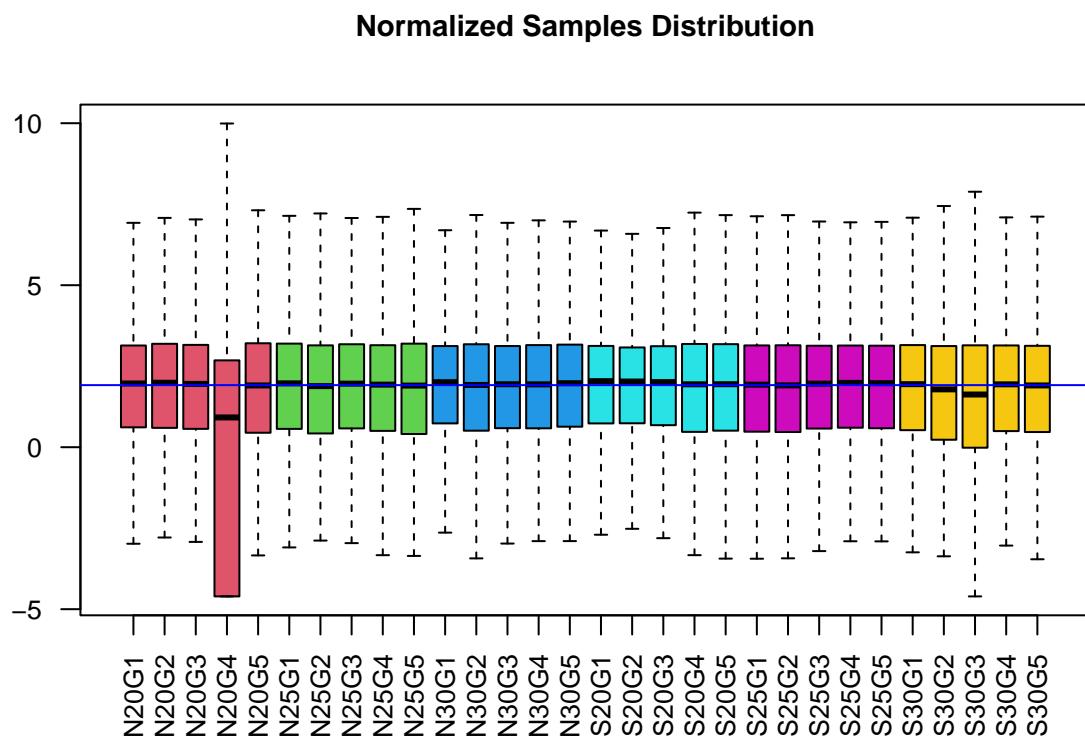
```

# Normalized Samples Distribution - Chronic

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", cex.main = 0.9)

```



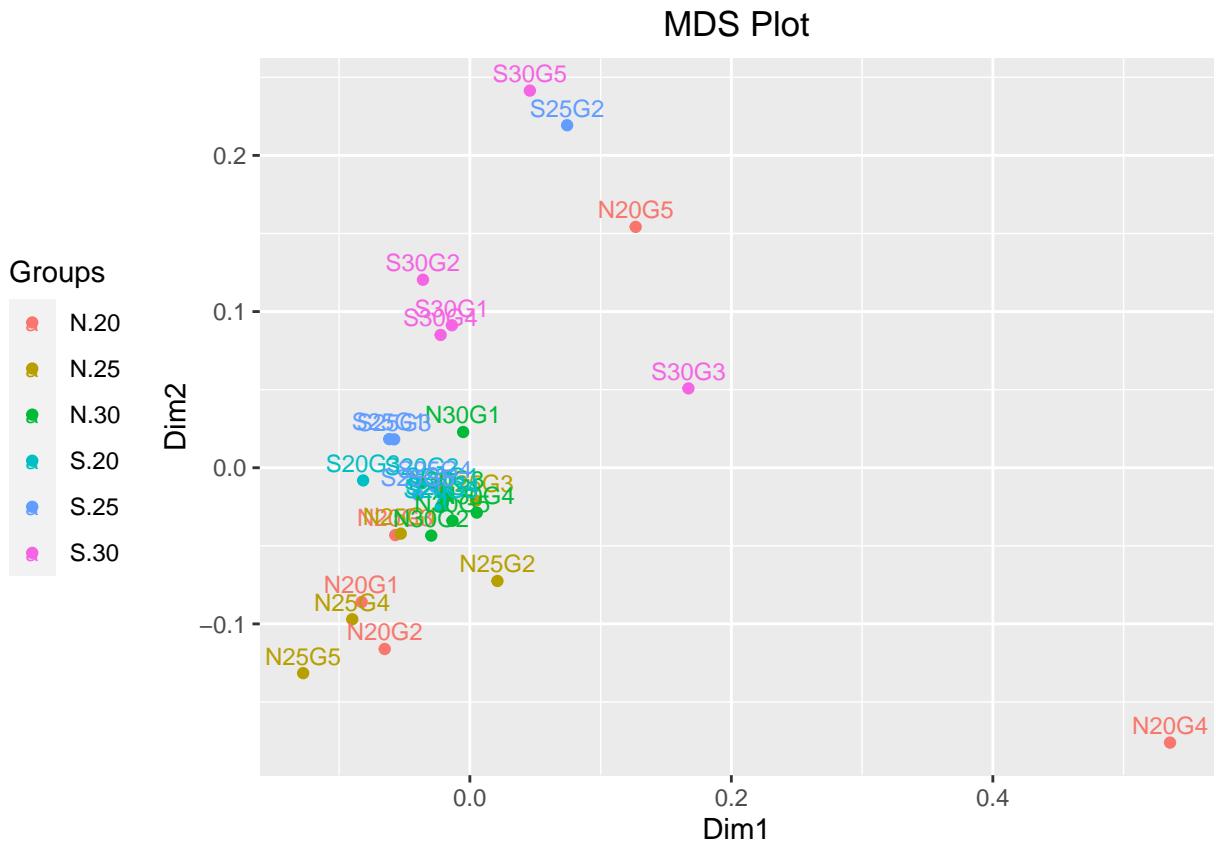
## MDS plot

```
# MDS plot

library(ggplot2)

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")
```



## Read files without N20G4

```
data_counts <- read.table("countData_wihout_N20G4.txt", header = TRUE, sep = "\t", row.names = 1, check.names = FALSE)
sample_info <- read.table("colData_without_N20G4.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)

# Create DGEList object
y <- DGEList(counts = data_counts, group = group)

# Calculate CPM
cpm_count <- cpm(y)

# Set thresholds
cpm_Val <- 1      # CPM value threshold
gThreshold <- 4     # At least number of samples threshold

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

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

y_filter <- y[thresholds,]

# Get CPM values for filtered data (y_filter)
cpm_count_filtered_final <- cpm(y_filter)

dim(cpm_count_filtered_final) # [1] 23202      29

## [1] 23202      29

head(cpm_count_filtered_final)

##          N20G1      N20G2      N20G3      N20G5      N25G1      N25G2
## UdeG00000000003 0.06382335 0.1301431 16.861059 0.02472601 9.0060176 0.0000000
## UdeG00000000004 0.14892115 0.6507156 10.316020 0.09890405 3.4719872 0.0000000
## UdeG00000000006 0.00000000 0.00000000 14.585466 0.00000000 10.0282270 0.0000000
## UdeG00000000008 0.04254890 0.1301431 5.851524 0.07417803 1.8505516 0.0000000
## UdeG00000000009 0.51058680 3.2796068 1.603751 0.00000000 0.8635907 0.3987797
## UdeG00000000010 4.95694684 5.2057251 3.142485 0.00000000 3.1371255 1.3735744
##          N25G3      N25G4      N25G5      N30G1      N30G2      N30G3
## UdeG00000000003 7.1868814 10.762820 0.02529561 5.513701 4.8623768 0.08325712
## UdeG00000000004 2.3320264 5.459027 0.00000000 2.330319 3.7770248 0.00000000
## UdeG00000000006 2.0988238 11.823579 0.00000000 3.391446 3.9072671 0.00000000
## UdeG00000000008 1.7172194 4.346523 0.27825175 3.557898 0.9985238 0.02081428
## UdeG00000000009 0.5724065 2.406111 0.27825175 2.912899 1.3458364 0.64524271
```

```

## UdeG00000000010 1.3568154 4.734606 0.30354736 4.161284 1.2373012 7.70128393
## N30G4 N30G5 S20G1 S20G2 S20G3 S20G4
## UdeG00000000003 0.1540222 6.457151 0.29432506 1.7452157 0.4724757 0.31107327
## UdeG00000000004 0.0000000 3.442314 0.00000000 0.3079792 0.3730071 0.27218911
## UdeG00000000006 0.0000000 4.724745 0.05518595 2.4296140 0.2238043 0.06480693
## UdeG00000000008 0.2640381 2.789849 0.07358127 0.3079792 0.4724757 0.81656734
## UdeG00000000009 0.7041016 2.429869 0.86457987 0.4790788 1.3925598 1.33502279
## UdeG00000000010 0.5720825 4.117278 0.93816113 0.4790788 5.6697079 2.11270596
## S20G5 S25G1 S25G2 S25G3 S25G4 S25G5
## UdeG00000000003 0.02225191 0.11367142 0.1621725 2.1748144 0.3205089 1.0004778
## UdeG00000000004 0.00000000 0.27281142 0.0000000 0.8883045 0.0000000 0.1111642
## UdeG00000000006 0.40053430 0.15913999 0.0000000 1.6540842 0.0000000 0.3334926
## UdeG00000000008 0.08900762 0.06820285 0.4401825 0.5819926 0.8470593 1.6896958
## UdeG00000000009 0.60080145 1.54593136 0.8571974 1.5315595 1.5796510 1.3117375
## UdeG00000000010 2.53671725 3.13733130 1.3668824 2.0829209 0.5265503 4.0019112
## S30G1 S30G2 S30G3 S30G4 S30G5
## UdeG00000000003 0.2661723 0.1041253 0.02584376 0.07790283 0.02176564
## UdeG00000000004 0.1330861 0.5206263 0.00000000 0.29213563 0.13059383
## UdeG00000000006 0.3105343 0.2863444 0.00000000 0.00000000 0.00000000
## UdeG00000000008 0.3696837 0.3123758 0.23259386 0.29213563 0.08706255
## UdeG00000000009 2.4103377 0.5206263 0.51687524 0.89588259 0.34825021
## UdeG00000000010 1.9962919 0.3644384 0.33596890 1.03221254 0.19589074

```

## Boxplot with sample distribution without N20G4

```

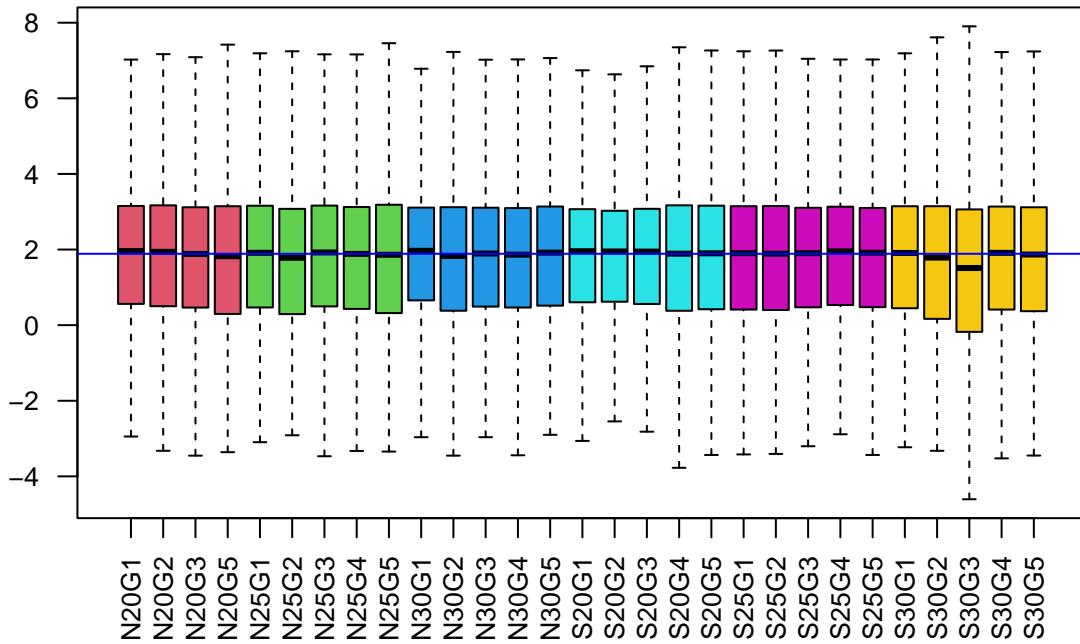
# Normalized Samples Distribution - Chronic without sample N20G4

statuscol <- as.numeric(factor(sample_info$group)) + 1
log_counts <- log(cpm_count_filtered_final + 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 without sample N20G4", cex.main = 0.9)

```

### Normalized Samples Distribution without sample N20G4

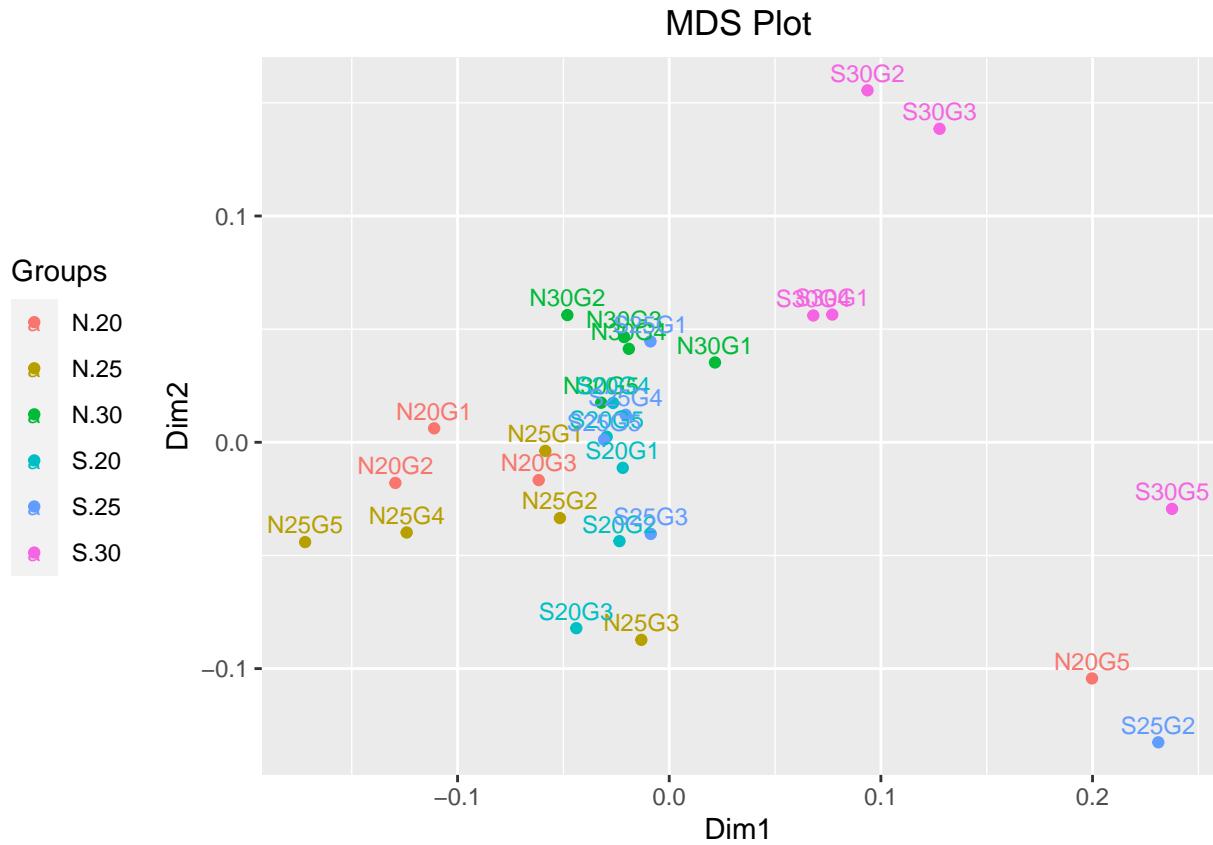


### MDS Plot without N20G4

```
# MDS plot without the sample N20G4

dist_matrix <- 1 - cor(cpm_count_filtered_final)
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")
```



## EdgeR

```
# Design matrix for glm approach

design <- model.matrix(~0 + group)

colnames(design) <- levels(group)

yf <- estimateDisp(y_filter, design)

fit <- glmFit(yf, design)

##### Contrasts all conditions #####
contrasts <- makeContrasts(
  S20_S25 = S.20-S.25,
  S20_S30 = S.20-S.30,
  S25_S30 = S.25-S.30,
  N20_N25 = N.20-N.25,
  N20_N30 = N.20-N.30,
  N25_N30 = N.25-N.30,
  N20_S20 = N.20-S.20,
  N25_S25 = N.25-S.25,
  N30_S30 = N.30-S.30,
```

```

        levels = colnames(coef(fit))
    )

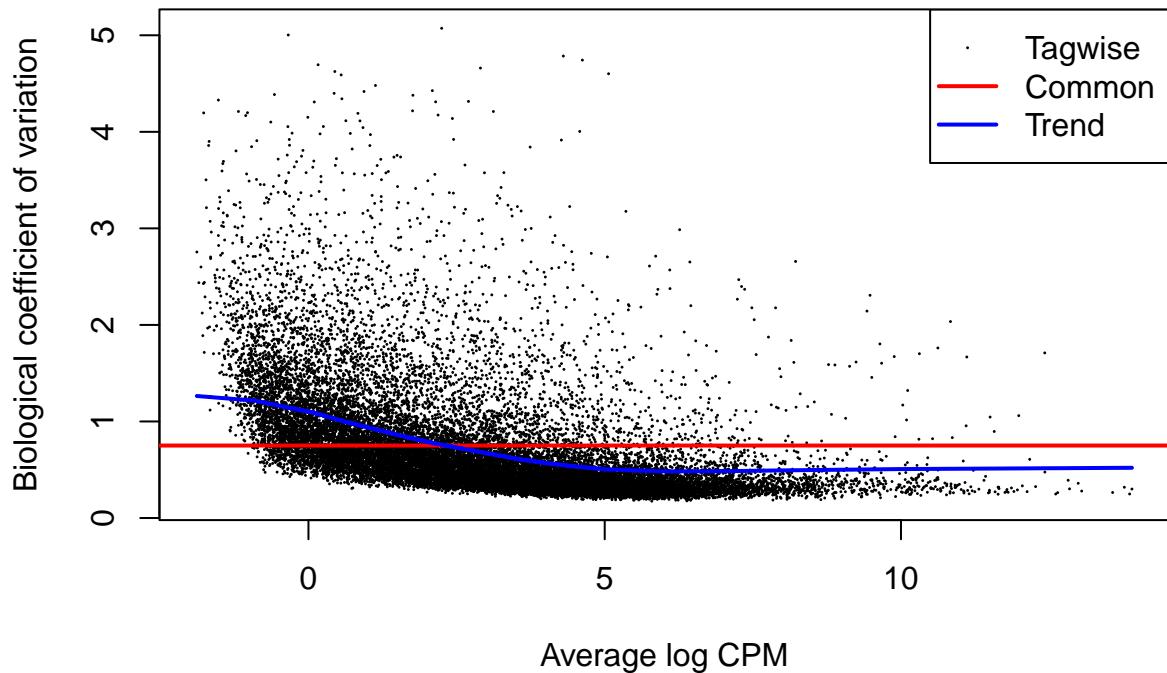
##### Contrasts - South Population #####
contrasts_south <- makeContrasts(
    S20_S25 = S.20 - S.25,
    S20_S30 = S.20 - S.30,
    levels = design
)

##### Contrasts - North Population #####
contrasts_north <- makeContrasts(
    N20_N25 = N.20 - N.25,
    N20_N30 = N.20 - N.30,
    levels = design
)

```

## Dispersion Plot - EdgeR

```
plotBCV(yf)
```



## EdgeR - North 20 °C vs South 20 °C

```

qlf_north_south_20 <- glmLRT(fit, contrast = contrasts[, "N20_S20"])

FDR_north_south_20 <- topTags(qlf_north_south_20, n = Inf, adjust.method = "BH", sort.by = "none")
FDR_north_south_df_20 <- data.frame(gene_id = rownames(FDR_north_south_20), FDR_north_south_20)
filtered_rows_north_south_20 <- FDR_north_south_df_20[FDR_north_south_df_20$FDR < 0.05 & abs(FDR_north_south_20$logFC) >= 1]
DGEs_EdgeR_north_south_20 <- filtered_rows_north_south_20[, c("gene_id", "logFC", "FDR")]

dim(DGEs_EdgeR_north_south_20) # [1] 583   6

## [1] 583   3

output_data_20 <- cpm_count_filtered_final[, grep("^N20|^S20", colnames(cpm_count_filtered_final))]

# Add the 'gene_id' column to the output_data matrix
data_gene_id_20 <- cbind(gene_id = rownames(output_data_20), output_data_20)

# Add the name of the first column: gene_id
names(data_gene_id_20) <- c("gene_id", names(data_gene_id_20)[-1])

# merge the data frames based on the "gene_id" column
merged_df_20 <- merge(data_gene_id_20, DGEs_EdgeR_north_south_20[, c("gene_id", "logFC", "FDR")], by = "gene_id")

# Selecting the desired columns
final_table_20 <- merged_df_20[, c("gene_id", "N20G1", "N20G2", "N20G3", "N20G5", "S20G1", "S20G2", "S20G3", "S20G4")]

# Convert character matrix to numeric matrix element-wise
numeric_matrix_20 <- final_table_20[, c("N20G1", "N20G2", "N20G3", "N20G5", "S20G1", "S20G2", "S20G3", "S20G4")]
numeric_matrix_20 <- apply(numeric_matrix_20, 2, as.numeric) # Convert each column to numeric

# Set row names
rownames(numeric_matrix_20) <- final_table_20$gene_id

# Set column names
colnames(numeric_matrix_20) <- c("N20G1", "N20G2", "N20G3", "N20G5", "S20G1", "S20G2", "S20G3", "S20G4")

head(numeric_matrix_20)

##          N20G1      N20G2      N20G3      N20G5      S20G1
## UdeG000000000150  1.0211736  0.3644008  1.777130  0.9148624  2.35460048
## UdeG000000000279  0.0000000  0.0000000  0.000000  0.0000000  0.000000000
## UdeG000000000328 226.0835799 381.5275934 109.965311 313.6494579 24.30021281
## UdeG000000000429  0.9360758  0.8589446  1.191977  0.7665064  0.03679063
## UdeG000000000475  3.2549908  1.9521469  1.560406  2.6704093  0.73581265
## UdeG000000000505  5.1909658  2.1083187  5.288044  1.8049989 21.26498562
##          S20G2      S20G3      S20G4      S20G5
## UdeG000000000150  8.89717820 7.31093916 10.2006110  5.2959536
## UdeG000000000279  2.73759329 2.03910548  3.4477288  1.5353815
## UdeG000000000328 19.30003271 28.39827388 40.9839034 27.4365997
## UdeG000000000429  0.06843983  0.04973428  0.2721891  0.0000000

```

```

## UdeG00000000475 0.00000000 0.54707708 1.4516753 0.7343129
## UdeG00000000505 24.56989979 13.92559840 5.0549407 2.9817554

dim(numeric_matrix_20)

## [1] 583    9

```

## EdgeR - North 25 °C vs South 25 °C

```

qlf_north_south_25 <- glmLRT(fit, contrast = contrasts[, "N25_S25"])
FDR_north_south_25 <- topTags(qlf_north_south_25, n = Inf, adjust.method = "BH", sort.by = "none")
FDR_north_south_df_25 <- data.frame(gene_id = rownames(FDR_north_south_25), FDR_north_south_25)
filtered_rows_north_south_25 <- FDR_north_south_df_25[FDR_north_south_df_25$FDR < 0.05 & abs(FDR_north_
DGEs_EdgeR_north_south_25 <- filtered_rows_north_south_25[, c("gene_id", "logFC", "FDR")]

dim(DGEs_EdgeR_north_south_25) # [1] 1527    6

```

```
## [1] 1527    3
```

## EdgeR - North 30 °C vs South 30 °C

```

qlf_north_south_30 <- glmLRT(fit, contrast = contrasts[, "N30_S30"])
FDR_north_south_30 <- topTags(qlf_north_south_30, n = Inf, adjust.method = "BH", sort.by = "none")
FDR_north_south_df_30 <- data.frame(gene_id = rownames(FDR_north_south_30), FDR_north_south_30)
filtered_rows_north_south_30 <- FDR_north_south_df_30[FDR_north_south_df_30$FDR < 0.05 & abs(FDR_north_
DGEs_EdgeR_north_south_30 <- filtered_rows_north_south_30[, c("gene_id", "logFC", "FDR")]

dim(DGEs_EdgeR_north_south_30) # [1] 2853    6

```

```
## [1] 2853    3
```

## EdgeR - North 20 °C vs 25 °C vs 30 °C

```

qlf_north <- glmLRT(fit, contrast = contrasts_north)
FDR_north <- topTags(qlf_north, n = Inf, adjust.method = "BH", sort.by = "none")
FDR_north_df <- data.frame(gene_id = rownames(FDR_north), FDR_north)
filtered_rows_north <- FDR_north_df[FDR_north_df$FDR < 0.05 & abs(FDR_north_df$logFC.N20_N25) > 1 & abs

filtered_rows_north <- filtered_rows_north[, c("gene_id", "logFC.N20_N25", "logFC.N20_N30")]

filtered_rows_north <- filtered_rows_north[, -1]

# Create a data frame with the "gene_id" column containing the row names and the "South_20" column filled with 0
DGEs_EdgeR_north_20_25_30 <- data.frame(gene_id = rownames(filtered_rows_north), North_20 = 0, filtered

dim(DGEs_EdgeR_north_20_25_30) # [1] 75    4

```

```
## [1] 75 4
```

## EdgeR - South 20 °C vs 25 °C vs 30 °C

```
qlf_south <- glmLRT(fit, contrast = contrasts_south)
FDR_south <- topTags(qlf_south, n = Inf, adjust.method = "BH", sort.by = "none")
FDR_south_df <- data.frame(gene_id = rownames(FDR_south), FDR_south)
filtered_rows_south <- FDR_south_df[FDR_south_df$FDR < 0.05 & abs(FDR_south_df$logFC.S20_S25) > 1 & abs(FDR_south_df$logFC.S20_S30) > 1]
filtered_rows_south <- filtered_rows_south[, c("gene_id", "logFC.S20_S25", "logFC.S20_S30")]
filtered_rows_south <- filtered_rows_south[, -1]

# Create a data frame with the "gene_id" column containing the row names and the "South_20" column filled with 0
DGEs_EdgeR_south_20_25_30 <- data.frame(gene_id = rownames(filtered_rows_south), South_20 = 0, filtered_rows_south)

dim(DGEs_EdgeR_south_20_25_30) # [1] 534 4
```

```
## [1] 534 4
```

## EdgeR - North vs South

```
# Design matrix for glm approach

design <- model.matrix(~sample_info$pop + sample_info$temperature)
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] 2891 3
```

```
## [1] 2891 3
```

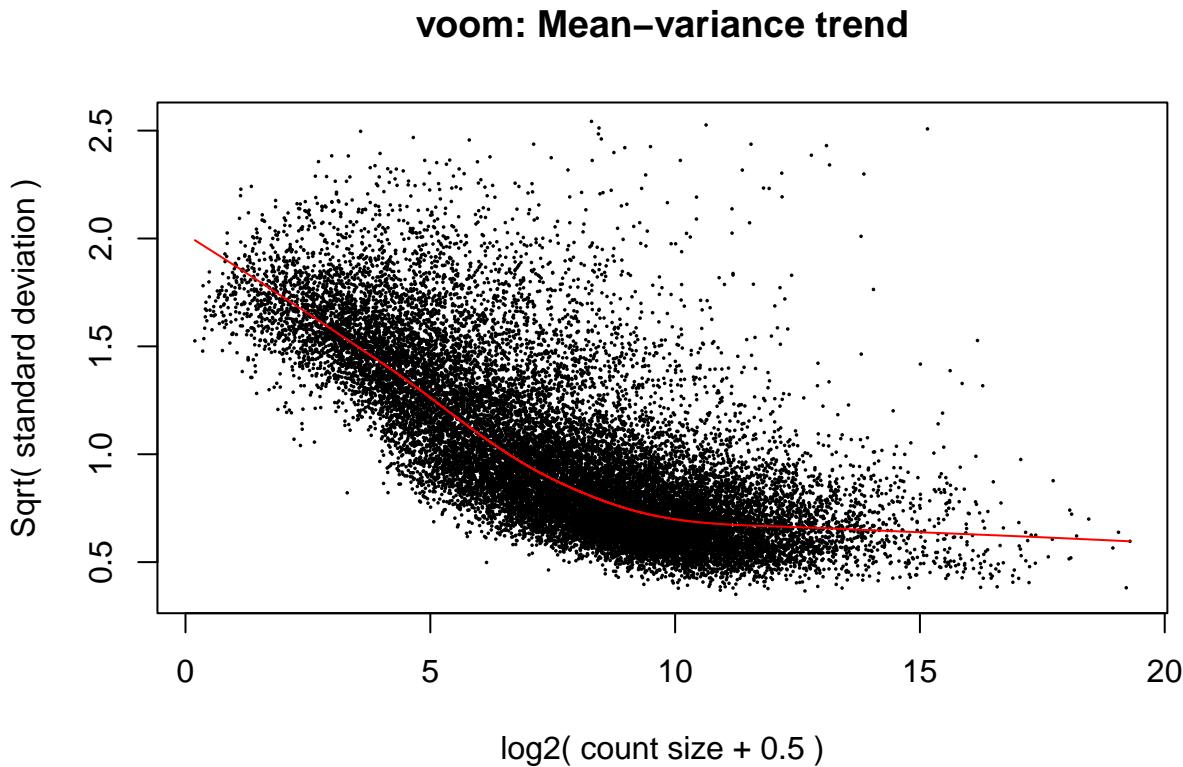
## limma

```
# design matrix - voom transformation

design <- model.matrix(~0+group)
```

```
colnames(design) <- levels(group)

yf <- voom(y_filter, design, normalize="quantile", plot=T)
```



```

##### Contrasts South Population #####
contrasts_south <- makeContrasts(
  S20_S25 = S.25-S.20,
  S20_S30 = S.30-S.20,
  levels=design
)

#####
##### Contrasts North Population #####
contrasts_north <- makeContrasts(
  N20_N25 = N.25-N.20,
  N20_N30 = N.30-N.20,
  levels=design
)

tmp <- contrasts.fit(fit, contrasts)
tmp_f <- eBayes(tmp)

## Contrasts - South all

tmp_south <- contrasts.fit(fit, contrasts_south)
tmp_south_f <- eBayes(tmp_south)

## Contrasts - North all

tmp_north <- contrasts.fit(fit, contrasts_north)
tmp_north_f <- eBayes(tmp_north)

```

## limma - North 20 °C vs South 20 °C

```

tmp_North_South_20 <- contrasts.fit(fit, contrasts[, "N20_S20"])
tmp_North_South_20_f <- eBayes(tmp_North_South_20)

top.table_North_South_20 <- topTable(tmp_North_South_20_f, number=Inf, adjust="BH", sort.by="none")
FDR_North_South_df_20 <- data.frame(gene_id = rownames(top.table_North_South_20), top.table_North_South_20)
filtered_North_South_20 <- FDR_North_South_df_20[FDR_North_South_df_20$adj.P.Val < 0.05 & abs(FDR_North_South_df_20$logFC) > 1]
DGEs_Limma_north_south_20 <- filtered_North_South_20[, c("gene_id", "logFC", "adj.P.Val")]

dim(DGEs_Limma_north_south_20) # [1] 169      7

## [1] 169      3

```

## limma - North 25 °C vs South 25 °C

```

# thresholds North 25 vs South 25

tmp_North_South_25 <- contrasts.fit(fit, contrasts[, "N25_S25"])
tmp_North_South_25_f <- eBayes(tmp_North_South_25)

top.table_North_South_25 <- topTable(tmp_North_South_25_f, number=Inf, adjust="BH", sort.by="none")
FDR_North_South_df_25 <- data.frame(gene_id = rownames(top.table_North_South_25), top.table_North_South_25)
filtered_North_South_25 <- FDR_North_South_df_25[FDR_North_South_df_25$adj.P.Val < 0.05 & abs(FDR_North_South_df_25$logFC) > 1]
DGEs_Limma_north_south_25 <- filtered_North_South_25[, c("gene_id", "logFC", "adj.P.Val")]

dim(DGEs_Limma_north_south_25) # [1] 718    7

```

## [1] 718 3

## limma - North 30 °C vs South 30 °C

```

# thresholds North 30 vs South 30

tmp_North_South_30 <- contrasts.fit(fit, contrasts[, "N30_S30"])
tmp_North_South_30_f <- eBayes(tmp_North_South_30)

top.table_North_South_30 <- topTable(tmp_North_South_30_f, number=Inf, adjust="BH", sort.by="none")
FDR_North_South_df_30 <- data.frame(gene_id = rownames(top.table_North_South_30), top.table_North_South_30)
filtered_North_South_30 <- FDR_North_South_df_30[FDR_North_South_df_30$adj.P.Val < 0.05 & abs(FDR_North_South_df_30$logFC) > 1]
DGEs_Limma_north_south_30 <- filtered_North_South_30[, c("gene_id", "logFC", "adj.P.Val")]

dim(DGEs_Limma_north_south_30) # [1] 2090    7

```

## [1] 2090 3

## limma - North 20 °C vs 25 °C vs 30 °C

```

top.table_north <- topTable(tmp_north_f, number=Inf, adjust="BH", sort.by="none")

FDR_north_df <- data.frame(rowname = rownames(top.table_north), top.table_north)

# Filter the selected rows
selected_rows <- top.table_north[top.table_north$adj.P.Val < 0.05 & abs(top.table_north$N20_N25) > 1 & abs(top.table_north$N30_N30) > 1]

# Create a data frame with the desired columns, including "gene_id" and "North_20" with zeros
DGEs_Limma_north_20_25_30 <- data.frame(gene_id = rownames(selected_rows), North_20 = 0, North_25 = selected_rows$N20_N25, North_30 = selected_rows$N30_N30)

# Print the first few rows of filtered_rows_north
dim(DGEs_Limma_north_20_25_30) # 38 5

```

## [1] 38 5

## limma - South 20 °C vs 25 °C vs 30 °C

```
# thresholds - south 20 vs 25 vs 30

top.table_south <- topTable(tmp_south_f, number=Inf, adjust="BH", sort.by="none")
FDR_south_df <- data.frame(gene_id = rownames(top.table_south), top.table_south)
# Filter the selected rows
selected_rows <- top.table_south[top.table_south$adj.P.Val < 0.05 & abs(top.table_south$S20_S25) > 1 & ...

# Create a data frame with the desired columns, including "gene_id" and "North_20" with zeros
DGEs_Limma_south_20_25_30 <- data.frame(gene_id = rownames(selected_rows), South_20 = 0, South_25 = sel ...

# Print the first few rows of filtered_rows_north
dim(DGEs_Limma_south_20_25_30)

## [1] 266    5
```

## limma - North vs South

```
# design matrix

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

# 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] 3030    7

## [1] 3030    7
```

## DESeq2

```
# DESeq2 analysis

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

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_names <- resultsNames(dds)
results_names

## [1] "Intercept"           "group_N.25_vs_N.20" "group_N.30_vs_N.20"
## [4] "group_S.20_vs_N.20" "group_S.25_vs_N.20" "group_S.30_vs_N.20"

##### Contrasts - South vs North Population #####
results_North_South_20 <- results(dds, contrast=c("group", "S.20", "N.20"))
results_North_South_25 <- results(dds, contrast=c("group", "S.25", "N.25"))
results_North_South_30 <- results(dds, contrast=c("group", "S.30", "N.30"))

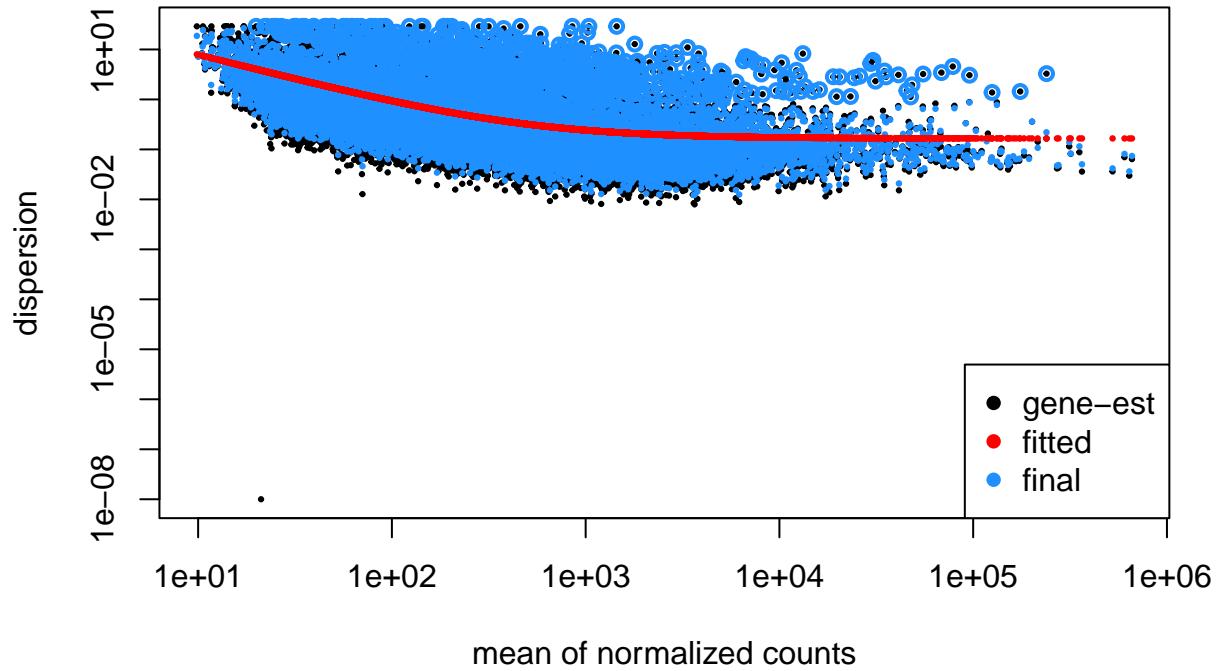
##### Contrasts - North #####
results_25_20_north <- results(dds, contrast=c("group", "N.25", "N.20"))
results_30_20_north <- results(dds, contrast=c("group", "N.30", "N.20"))

##### Contrasts - South #####
results_25_20_south <- results(dds, contrast=c("group", "S.25", "S.20"))
results_30_20_south <- results(dds, contrast=c("group", "S.30", "S.20"))

```

## Dispersion Plot - DESeq2

```
plotDispEsts(dds)
```



## DESeq2 - North 20 °C vs South 20 °C

```

results_North_South_20 <- na.omit(results_North_South_20)
filter_results_N_S_20 <- results_North_South_20[results_North_South_20$padj < 0.05 & abs(results_North_South_20$log2FoldChange) >= 1]
filter_results_N_S_20 <- cbind(gene_id = rownames(filter_results_N_S_20), filter_results_N_S_20)
DGEs_DESeq2_north_south_20 <- filter_results_N_S_20[, c("gene_id", "log2FoldChange", "padj")]

dim(DGEs_DESeq2_north_south_20)    # [1] 1115    6

## [1] 1115    3

```

## DESeq2 - North 25 °C vs South 25 °C

```

results_North_South_25 <- na.omit(results_North_South_25)
filter_results_N_S_25 <- results_North_South_25[results_North_South_25$padj < 0.05 & abs(results_North_South_25$log2FoldChange) >= 1]
filter_results_N_S_25 <- cbind(gene_id = rownames(filter_results_N_S_25), filter_results_N_S_25)
DGEs_DESeq2_north_south_25 <- filter_results_N_S_25[, c("gene_id", "log2FoldChange", "padj")]

dim(DGEs_DESeq2_north_south_25)    # [1] 1912    3

```

```
## [1] 1912     3
```

## DESeq2 - North 30 °C vs South 30 °C

```
results_North_South_30 <- na.omit(results_North_South_30)
filter_results_N_S_30 <- results_North_South_30[results_North_South_30$padj < 0.05 & abs(results_North_South_30$log2FoldChange) >= 1]
filter_results_N_S_30 <- cbind(gene_id = rownames(filter_results_N_S_30), filter_results_N_S_30)
DGEs_DESeq2_north_south_30 <- filter_results_N_S_30[, c("gene_id", "log2FoldChange", "padj")]

dim(DGEs_DESeq2_north_south_30)    # [1] 3259     3
```

```
## [1] 3259     3
```

## DESeq2 - South 20 °C 25 °C

```
results_25_20_south <- na.omit(results_25_20_south)
filter_results_S_25_20 <- results_25_20_south[results_25_20_south$padj < 0.05 & abs(results_25_20_south$log2FoldChange) >= 1]
filter_results_S_25_20 <- cbind(gene_id = rownames(filter_results_S_25_20), filter_results_S_25_20)
DGEs_DESeq2_south_25_20 <- filter_results_S_25_20[, c("gene_id", "log2FoldChange", "padj")]

dim(DGEs_DESeq2_south_25_20)
```

```
## [1] 139     3
```

## DESeq2 - South 20 °C 30 °C

```
results_30_20_south <- na.omit(results_30_20_south)
filter_results_S_20_30 <- results_30_20_south[results_30_20_south$padj < 0.05 & abs(results_30_20_south$log2FoldChange) >= 1]
filter_results_S_20_30 <- cbind(gene_id = rownames(filter_results_S_20_30), filter_results_S_20_30)
DGEs_DESeq2_south_20_30 <- filter_results_S_20_30[, c("gene_id", "log2FoldChange", "padj")]

dim(DGEs_DESeq2_south_20_30)
```

```
## [1] 3313     3
```

## DESeq2 - South 20 °C vs 25 °C vs 30 °C

```

gene_id_30_20 <- DGEs_DESeq2_south_20_30[, "gene_id", drop = FALSE]
gene_id_25_20 <- DGEs_DESeq2_south_25_20[, "gene_id", drop = FALSE]

colnames(gene_id_25_20) <- "gene_id"
colnames(gene_id_30_20) <- "gene_id"

resulting_dataframe <- rbind(gene_id_25_20, gene_id_30_20)

resulting_dataframe <- unique(resulting_dataframe)

resulting_dataframe$log2FC_25 <- DGEs_DESeq2_south_25_20$log2FoldChange[match(resulting_dataframe$gene_id, colnames(DGEs_DESeq2_south_25_20))]

resulting_dataframe$log2FC_30 <- DGEs_DESeq2_south_20_30$log2FoldChange[match(resulting_dataframe$gene_id, colnames(DGEs_DESeq2_south_20_30))]

DGEs_DESeq2_south_20_25_30 <- na.omit(resulting_dataframe)
DGEs_DESeq2_south_20_25_30

```

```

## DataFrame with 77 rows and 3 columns
##           gene_id log2FC_25 log2FC_30
##      <character> <numeric> <numeric>
## 1   UdeG00000000426 -1.36901 -1.39926
## 2   UdeG00000003641  22.84076 21.24443
## 3   UdeG00000003772 -2.70751 -2.11722
## 4   UdeG00000004352  4.12266  4.05823
## 5   UdeG00000004517  3.43933  4.43814
## ...       ...
## 73  UdeG00000038795 -1.66671 -2.84935
## 74  UdeG00000039369 -1.95290 -3.02002
## 75  UdeG00000039542 -2.82139 -2.96147
## 76  UdeG00000039886 24.16327 23.25911
## 77  UdeG00000039887 23.54424 22.36500

```

## DESeq2 - North 20 °C 25 °C

```

results_25_20_north <- na.omit(results_25_20_north)
filter_results_N_25_20 <- results_25_20_north[results_25_20_north$padj < 0.05 & abs(results_25_20_north$log2FC) >= 1,]

filter_results_N_25_20 <- cbind(gene_id = rownames(filter_results_N_25_20), filter_results_N_25_20)
DGEs_DESeq2_north_25_20 <- filter_results_N_25_20[, c("gene_id", "log2FoldChange", "padj")]

dim(DGEs_DESeq2_north_25_20)

## [1] 107    3

```

## DESeq2 - North 20 °C 30 °C

```

results_30_20_north <- na.omit(results_30_20_north)
filter_results_N_20_30 <- results_30_20_north[results_30_20_north$padj < 0.05 & abs(results_30_20_north$padj) >= 1]
filter_results_N_20_30 <- cbind(gene_id = rownames(filter_results_N_20_30), filter_results_N_20_30)
DGEs_DESeq2_north_20_30 <- filter_results_N_20_30[, c("gene_id", "log2FoldChange", "padj")]

dim(DGEs_DESeq2_north_20_30)

## [1] 143   3

```

## DESeq2 - North 20 °C vs 25 °C vs 30 °C

```

gene_id_30_20 <- DGEs_DESeq2_north_20_30[, "gene_id", drop = FALSE]
gene_id_25_20 <- DGEs_DESeq2_north_25_20[, "gene_id", drop = FALSE]

colnames(gene_id_25_20) <- "gene_id"
colnames(gene_id_30_20) <- "gene_id"

resulting_dataframe <- rbind(gene_id_25_20, gene_id_30_20)

resulting_dataframe <- unique(resulting_dataframe)

resulting_dataframe$log2FC_25 <- DGEs_DESeq2_north_25_20$log2FoldChange[match(resulting_dataframe$gene_id, colnames(DGEs_DESeq2_north_25_20))]
resulting_dataframe$log2FC_30 <- DGEs_DESeq2_north_20_30$log2FoldChange[match(resulting_dataframe$gene_id, colnames(DGEs_DESeq2_north_20_30))]

DGEs_DESeq2_north_20_25_30 <- na.omit(resulting_dataframe)
DGEs_DESeq2_north_20_25_30

## DataFrame with 37 rows and 3 columns
##          gene_id log2FC_25 log2FC_30
## 1    UdeG00000000167  21.06698  15.71457
## 2    UdeG00000000426   1.40409 -1.44072
## 3    UdeG00000001883   9.65962   9.53020
## 4    UdeG00000003881 -19.09240 -19.41514
## 5    UdeG00000010548  19.63359  19.30829
## ...
## 33   UdeG00000038795   1.20712 -2.50329
## 34   UdeG00000040141  16.63691  18.34863
## 35   UdeG00000040228  23.71543  22.85586
## 36   UdeG00000042301  22.98205  21.29825
## 37   UdeG00000043499   7.28668   7.25470

```

## DESeq2 - North vs South

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

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

## the design formula contains one or more numeric variables with integer values,
## specifying a model with increasing fold change for higher values.
## did you mean for this to be a factor? if so, first convert
## this variable to a factor using the factor() function

## the design formula contains one or more numeric variables that have mean or
## standard deviation larger than 5 (an arbitrary threshold to trigger this message).
## Including numeric variables with large mean can induce collinearity with the intercept.
## Users should center and scale numeric variables in the design to improve GLM convergence.

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

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## 4 rows did not converge in beta, labelled in mcols(object)$betaConv. Use larger maxit argument with mcols if needed

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] 3095 6

## [1] 3095 3
```

## Overlap 3 methods - North vs South 20 °C

```

DESeq2 <- DGEs_DESeq2_north_south_20$gene_id
EdgeR <- DGEs_EdgeR_north_south_20$gene_id
Limma <- DGEs_Limma_north_south_20$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_20[, c("logFC", "adj.P.Val")]

limma_data$gene_id <- rownames(limma_data)

North_South_overlap_20 <- merge(overlap_3_methods, limma_data, by = "gene_id", all.x = TRUE)

head(North_South_overlap_20)

##           gene_id      logFC   adj.P.Val
## 1 UdeG00000000328 -3.022098 0.002532289
## 2 UdeG00000000429 -3.907947 0.042769473
## 3 UdeG00000000763  2.045311 0.034605617
## 4 UdeG00000001424  1.874900 0.016098611
## 5 UdeG00000001541  1.066869 0.015854230
## 6 UdeG0000001893 -1.944597 0.003381123

dim(North_South_overlap_20)

## [1] 130    3

write.table(North_South_overlap_20, file = "North_South_overlap_20.txt", sep = "\t", quote = FALSE, row.names = FALSE)

output_data_20 <- cpm_count_filtered_final[, grep("N20|S20", colnames(cpm_count_filtered_final))]

# Add the 'gene_id' column to the output_data matrix
data_gene_id_20 <- cbind(gene_id = rownames(output_data_20), output_data_20)

# Add the name of the first column: gene_id
names(data_gene_id_20) <- c("gene_id", names(data_gene_id_20)[-1])

# merge the data frames based on the "gene_id" column
merged_df_20 <- merge(data_gene_id_20, North_South_overlap_20[, c("gene_id", "logFC", "adj.P.Val")], by = "gene_id")

# Selecting the desired columns
final_table_20 <- merged_df_20[, c("gene_id", "N20G1", "N20G2", "N20G3", "N20G5", "S20G1", "S20G2", "S20G3")]

# Convert character matrix to numeric matrix element-wise
numeric_matrix_20 <- final_table_20[, c("N20G1", "N20G2", "N20G3", "N20G5", "S20G1", "S20G2", "S20G3")]
numeric_matrix_20 <- apply(numeric_matrix_20, 2, as.numeric) # Convert each column to numeric

# Set row names

```

```

rownames(numeric_matrix_20) <- final_table_20$gene_id

# Set column names
colnames(numeric_matrix_20) <- c("N20G1", "N20G2", "N20G3", "N20G5", "S20G1", "S20G2", "S20G3", "S20G4")

head(numeric_matrix_20)

##          N20G1      N20G2      N20G3      N20G5      S20G1
## UdeG00000000328 226.0835799 381.5275934 109.965311 313.6494579 24.30021281
## UdeG00000000429   0.9360758   0.8589446   1.191977   0.7665064   0.03679063
## UdeG00000000763   6.9780196   3.0974064   4.139412   5.0441064 17.80666616
## UdeG00000001424   1.9572494   1.8220038   3.662621   2.4478752  8.97691435
## UdeG00000001541  11.5307519  11.3484807  9.427456  9.2228023 22.47907650
## UdeG00000001893  27.4653149  48.6995584 31.186457  80.3595379 10.46693496
##          S20G2      S20G3      S20G4      S20G5
## UdeG00000000328 19.30003271 28.39827388 40.9839034 27.43660
## UdeG00000000429   0.06843983  0.04973428  0.2721891  0.00000
## UdeG00000000763  21.79808658 22.20635602 20.2845695 12.72809
## UdeG00000001424   9.20515744  7.55961056  8.7359744 10.39164
## UdeG00000001541  26.14401593 23.30051018 19.1958131 14.88652
## UdeG00000001893 15.33052243  5.99298074 15.5666249 10.30263

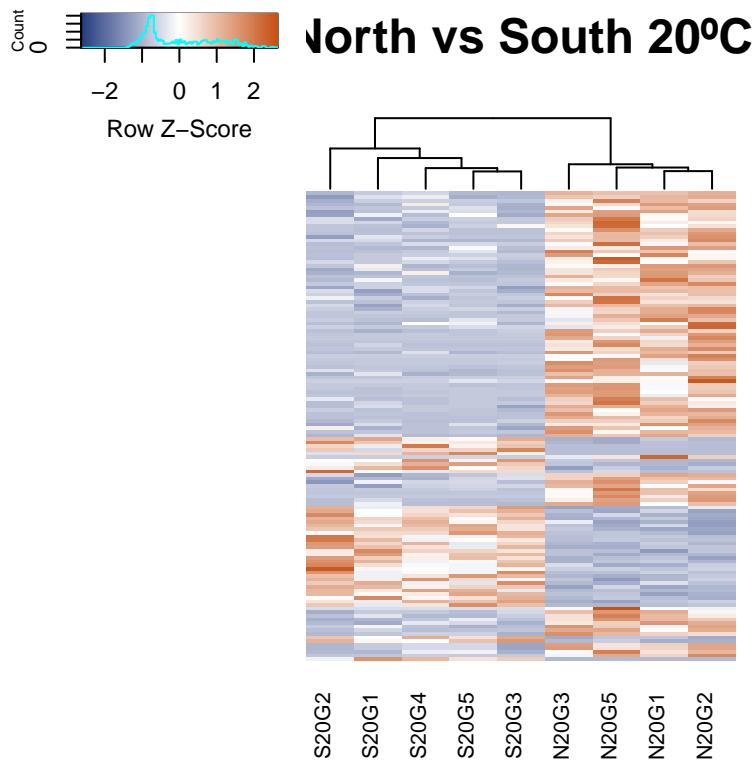
dim(numeric_matrix_20)

## [1] 130    9

gene_labels <- rownames(numeric_matrix_20)
condition_labels <- colnames(numeric_matrix_20)

heatmap.2(numeric_matrix_20,
           scale = "row",
           trace = "none",
           col = colorRampPalette(colors = c("#273D82", "white", "#C75218"))(100),
           main = "North vs South 20°C",
           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
)

```



## Data for volcano plots

```

colnames(North_South_overlap_20) <- c("gene_id", "logFC", "adj.P.Val")

# Add a column to the data frame to specify if they are UP- or DOWN- regulated
North_South_overlap_20$diffexpressed <- "NO"

# if log2Foldchange > 0.6 and pvalue < 0.05, set as "Up regulated"
North_South_overlap_20$diffexpressed[North_South_overlap_20$logFC > 0.6 & North_South_overlap_20$adj.P.V]

# if log2Foldchange < -0.6 and pvalue < 0.05, set as "Down regulated"
North_South_overlap_20$diffexpressed[North_South_overlap_20$logFC < -0.6 & North_South_overlap_20$adj.P.V]

# number of DGes - up and down regulated
up_regulated_count <- sum(North_South_overlap_20$diffexpressed == "Up regulated")
down_regulated_count <- sum(North_South_overlap_20$diffexpressed == "Down regulated")
cat("Number of up-regulated genes:", up_regulated_count, "\n")

## Number of up-regulated genes: 40

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

## Number of down-regulated genes: 90

```

```

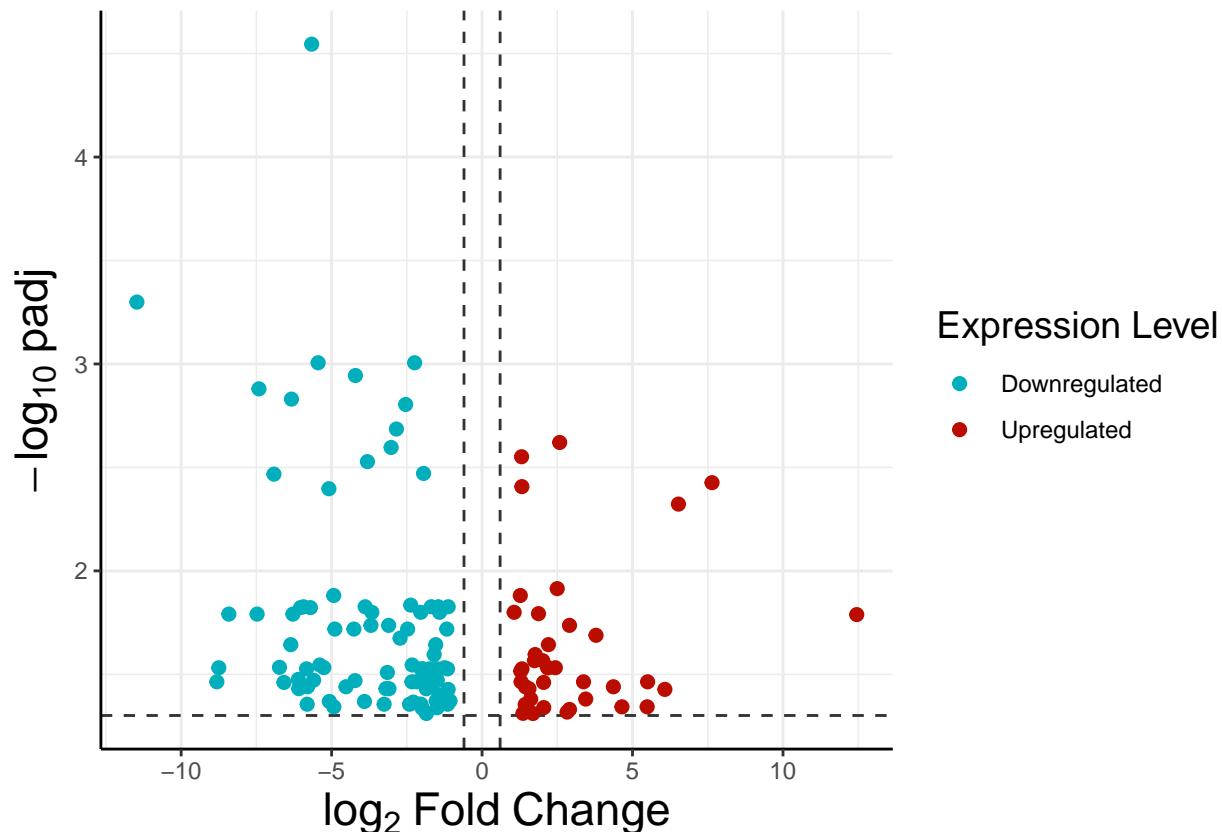
ggplot(data = North_South_overlap_20, aes(x = logFC, y = -log10(adj.P.Val), col = diffexpressed)) +
  geom_vline(xintercept = c(-0.6, 0.6), col = "gray20", linetype = 'dashed') +
  geom_hline(yintercept = -log10(0.05), col = "gray20", linetype = 'dashed') +
  geom_point(size = 2) +
  scale_color_manual(values = 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.

```



## Overlap 3 methods - North vs South 25 °C

```
DESeq2 <- DGEs_DESeq2_north_south_25$gene_id
EdgeR <- DGEs_EdgeR_north_south_25$gene_id
Limma <- DGEs_Limma_north_south_25$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)

dim(overlap_3_methods)

## [1] 594    1

limma_data <- DGEs_Limma_north_south_25[, c("logFC", "adj.P.Val")]

limma_data$gene_id <- rownames(limma_data)

North_South_overlap_25 <- merge(overlap_3_methods, limma_data, by = "gene_id", all.x = TRUE)

head(North_South_overlap_25)

##           gene_id      logFC   adj.P.Val
## 1 UdeG00000000188  1.536609 0.020874547
## 2 UdeG00000000221 -1.187146 0.015712601
## 3 UdeG00000000278  1.047899 0.039779724
## 4 UdeG00000000328 -2.189747 0.009089818
## 5 UdeG00000000426 -1.795901 0.006635745
## 6 UdeG00000000430 -8.223373 0.010480924

dim(North_South_overlap_25)

## [1] 594    3

write.table(North_South_overlap_25, file = "North_South_overlap_25.txt", sep = "\t", quote = FALSE, row.names = FALSE)

output_data_25 <- cpm_count_filtered_final[, grep("N25|S25", colnames(cpm_count_filtered_final))]

# Add the 'gene_id' column to the output_data matrix
data_gene_id_25 <- cbind(gene_id = rownames(output_data_25), output_data_25)

# 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_25[, c("gene_id", "logFC", "adj.P.Val")], by = "gene_id")

# Selecting the desired columns
```

```

final_table_25 <- merged_df_25[, c("gene_id", "N25G1", "N25G2", "N25G3", "N25G4", "N25G5", "S25G1", "S25G2", "S25G3")]

# Convert character matrix to numeric matrix element-wise
numeric_matrix_25 <- final_table_25[, c( "N25G1", "N25G2", "N25G3", "N25G4", "N25G5", "S25G1", "S25G2", "S25G3")]
numeric_matrix_25 <- apply(numeric_matrix_25, 2, as.numeric) # Convert each column to numeric

# Set row names
rownames(numeric_matrix_25) <- final_table_25$gene_id

# Set column names
colnames(numeric_matrix_25) <- c("N25G1", "N25G2", "N25G3", "N25G4", "N25G5", "S25G1", "S25G2", "S25G3")

head(numeric_matrix_25)

##          N25G1      N25G2      N25G3      N25G4      N25G5
## UdeG00000000188  3.489612  1.728045  4.558052  2.380239  3.566682
## UdeG00000000221 19.175239 29.022298 21.666645 34.047767 31.670108
## UdeG00000000278  3.930219  4.918283  2.713631  6.054086  6.450381
## UdeG00000000328 154.547491 157.872442 104.962388 518.271180 373.843872
## UdeG00000000426  38.491472  29.819858  79.013295  33.840790  45.355035
## UdeG00000000430  2.731767  5.361371  3.773643  6.468041  4.452028
##          S25G1      S25G2      S25G3      S25G4      S25G5
## UdeG00000000188  8.366217 10.726551  9.189357 10.66836794  9.382258
## UdeG00000000221 13.663305  8.340299  9.281250 12.66010187 13.650964
## UdeG00000000278  9.252854 10.402207 14.457921  7.18855694  8.448479
## UdeG00000000328 43.263344 27.059637 73.177911 32.98952405 56.204619
## UdeG00000000426 15.777594 16.750101 12.742575  9.84420218  8.759739
## UdeG00000000430  0.000000  0.000000  0.000000  0.06868048  0.000000

dim(numeric_matrix_25)

## [1] 594 10

```

## Heatmap

```

gene_labels <- rownames(numeric_matrix_25)
condition_labels <- colnames(numeric_matrix_25)

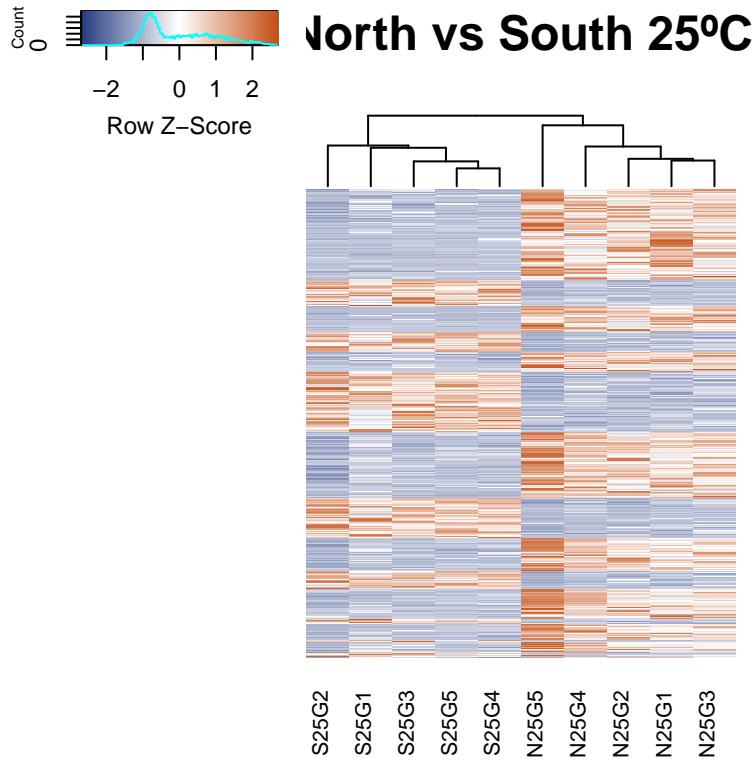
heatmap.2(numeric_matrix_25,
          scale = "row",
          trace = "none",
          col = colorRampPalette(colors = c("#273D82", "white", "#C75218"))(100),
          main = "North vs South 25°C",
          cex.main = 1,
          Colv = TRUE,
          hclustfun = function(c) hclust(c, method = "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
)

```



## Data for volcano plot

```

colnames(North_South_overlap_25) <- c("gene_id", "logFC", "adj.P.Val")

North_South_overlap_25$diffexpressed <- "NO"

North_South_overlap_25$diffexpressed[North_South_overlap_25$logFC > 0.6 & North_South_overlap_25$adj.P.Valor <= 0.05] <- "Up regulated"
North_South_overlap_25$diffexpressed[North_South_overlap_25$logFC < -0.6 & North_South_overlap_25$adj.P.Valor <= 0.05] <- "Down regulated"

# number of DEGs - up and down regulated
up_regulated_count <- sum(North_South_overlap_25$diffexpressed == "Up regulated")
down_regulated_count <- sum(North_South_overlap_25$diffexpressed == "Down regulated")
cat("Number of up-regulated genes:", up_regulated_count, "\n")

```

```

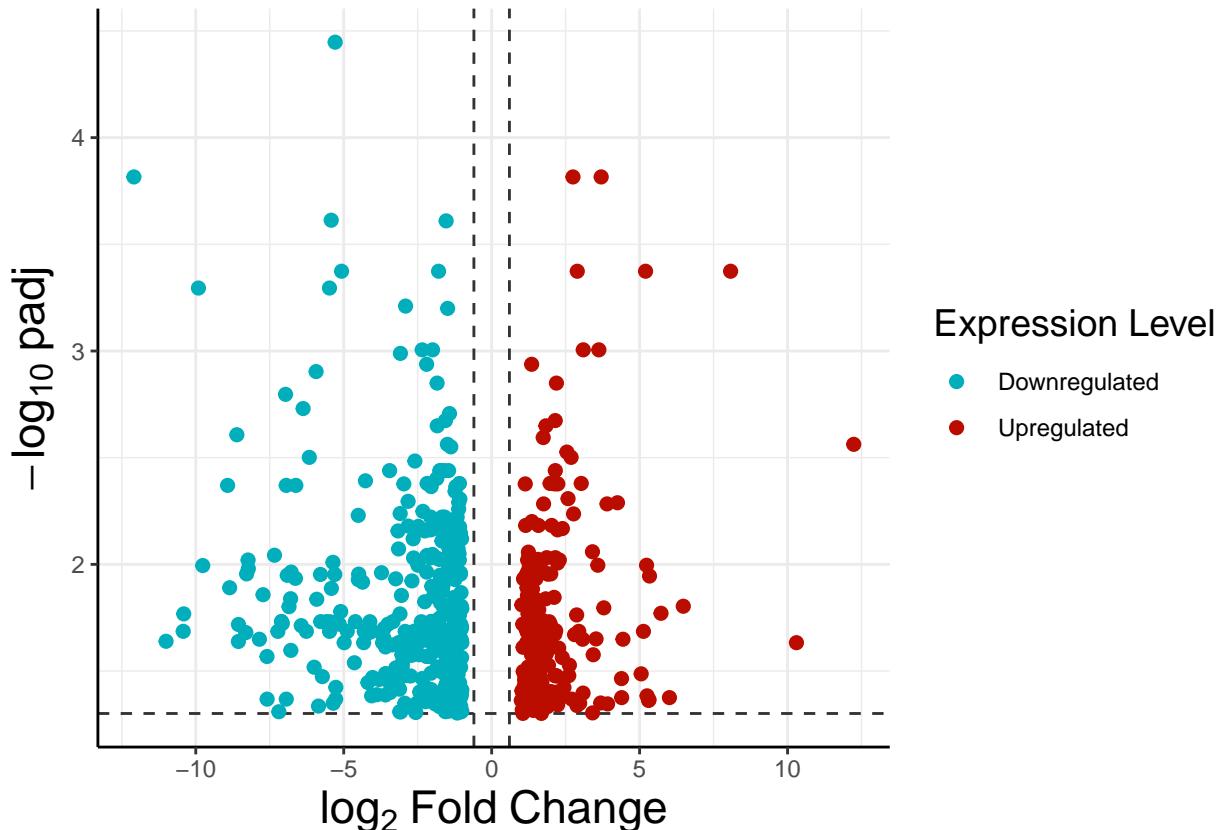
## Number of up-regulated genes: 222

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

## Number of down-regulated genes: 372

ggplot(data = North_South_overlap_25, 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
  )

```



## Overlap 3 methods - North vs South 30 °C

```
DESeq2 <- DGEs_DESeq2_north_south_30$gene_id
EdgeR <- DGEs_EdgeR_north_south_30$gene_id
Limma <- DGEs_Limma_north_south_30$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)

dim(overlap_3_methods)

## [1] 1609     1

limma_data <- DGEs_Limma_north_south_30[, c("logFC", "adj.P.Val")]

limma_data$gene_id <- rownames(limma_data)

North_South_overlap_30 <- merge(overlap_3_methods, limma_data, by = "gene_id", all.x = TRUE)

head(North_South_overlap_30)

##           gene_id      logFC    adj.P.Val
## 1 UdeG00000000120  2.068766 4.746887e-02
## 2 UdeG00000000140  1.249509 4.942802e-02
## 3 UdeG00000000227 -1.194311 5.622638e-03
## 4 UdeG00000000247  2.286263 7.010305e-03
## 5 UdeG00000000291 -1.284528 5.309249e-04
## 6 UdeG0000000328 -3.802700 2.796557e-05

dim(North_South_overlap_30)

## [1] 1609     3

write.table(North_South_overlap_30, file = "North_South_overlap_30.txt", sep = "\t", quote = FALSE, row.names = FALSE)

output_data_30 <- cpm_count_filtered_final[, grep("N30|S30", colnames(cpm_count_filtered_final))]

# Add the 'gene_id' column to the output_data matrix
data_gene_id_30 <- cbind(gene_id = rownames(output_data_30), output_data_30)

# Add the name of the first column: gene_id
names(data_gene_id_30) <- c("gene_id", names(data_gene_id_30)[-1])

# merge the data frames based on the "gene_id" column
merged_df_30 <- merge(data_gene_id_30, North_South_overlap_30[, c("gene_id", "logFC", "adj.P.Val")], by = "gene_id")

# Selecting the desired columns
```

```

final_table_30 <- merged_df_30[, c("gene_id", "N30G1", "N30G2", "N30G3", "N30G4", "N30G5", "S30G1", "S30G2", "S30G3", "S30G4", "S30G5")]

# Convert character matrix to numeric matrix element-wise
numeric_matrix_30 <- final_table_30[, c( "N30G1", "N30G2", "N30G3", "N30G4", "N30G5", "S30G1", "S30G2", "S30G3", "S30G4", "S30G5")]
numeric_matrix_30 <- apply(numeric_matrix_30, 2, as.numeric) # Convert each column to numeric

# Set row names
rownames(numeric_matrix_30) <- final_table_30$gene_id

# Set column names
colnames(numeric_matrix_30) <- c("N30G1", "N30G2", "N30G3", "N30G4", "N30G5", "S30G1", "S30G2", "S30G3", "S30G4", "S30G5")

head(numeric_matrix_30)

##          N30G1      N30G2      N30G3      N30G4      N30G5
## UdeG00000000120  0.644999  4.7104275  0.6452427  0.9461365  1.529917
## UdeG00000000140  6.096281  4.1243374  3.2470278 10.5395202  3.554808
## UdeG00000000227 28.234312 31.6705702 41.0457619 53.3136897 33.815674
## UdeG00000000247  1.955803  0.8031605  0.8533855  0.6380920  1.259932
## UdeG00000000291 23.906577 21.7504533 24.6232943 20.3089293 19.123967
## UdeG00000000328 98.310335 177.1511470 285.4262339 180.5360398 674.941052
##          S30G1      S30G2      S30G3      S30G4      S30G5
## UdeG00000000120  1.951930  8.408114 12.560068  2.239706  4.614315
## UdeG00000000140  8.635811 15.670850 27.549450  6.816498 10.316912
## UdeG00000000227 22.417619  9.475398 20.054759 17.391808 14.017071
## UdeG00000000247  3.977797  4.529448  1.085438  4.265180  9.533349
## UdeG00000000291  9.848374  6.221484  4.806940  8.121370 15.627728
## UdeG00000000328 37.692949 16.894322  8.244160 17.333380 14.387087

dim(numeric_matrix_30)

## [1] 1609   10

```

## Heatmap

```

library(gplots)

gene_labels <- rownames(numeric_matrix_30)
condition_labels <- colnames(numeric_matrix_30)

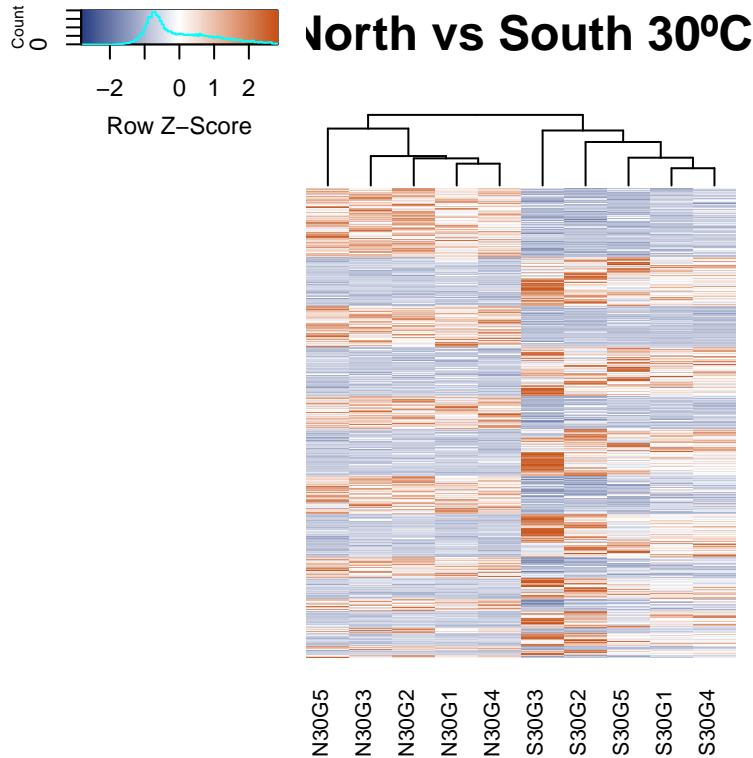
heatmap.2(numeric_matrix_30,
          scale = "row",
          trace = "none",
          col = colorRampPalette(colors = c("#273D82", "white", "#C75218"))(100),
          main = "North vs South 30°C",
          cex.main = 1,
          Colv = TRUE,
          hclustfun = function(c) hclust(c, method = "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
)

```



## Data for volcano plot

```

colnames(North_South_overlap_30) <- c("gene_id", "logFC", "adj.P.Val")

North_South_overlap_30$diffexpressed <- "NO"

North_South_overlap_30$diffexpressed[North_South_overlap_30$logFC > 0.6 & North_South_overlap_30$adj.P.1]

North_South_overlap_30$diffexpressed[North_South_overlap_30$logFC < -0.6 & North_South_overlap_30$adj.P.1]

# number of DGEs - up and down regulated
upregulated_count <- sum(North_South_overlap_30$diffexpressed == "Up regulated")

```

```

downregulated_count <- sum(North_South_overlap_30$diffexpressed == "Down regulated")
cat("Number of up-regulated genes:", upregulated_count, "\n")

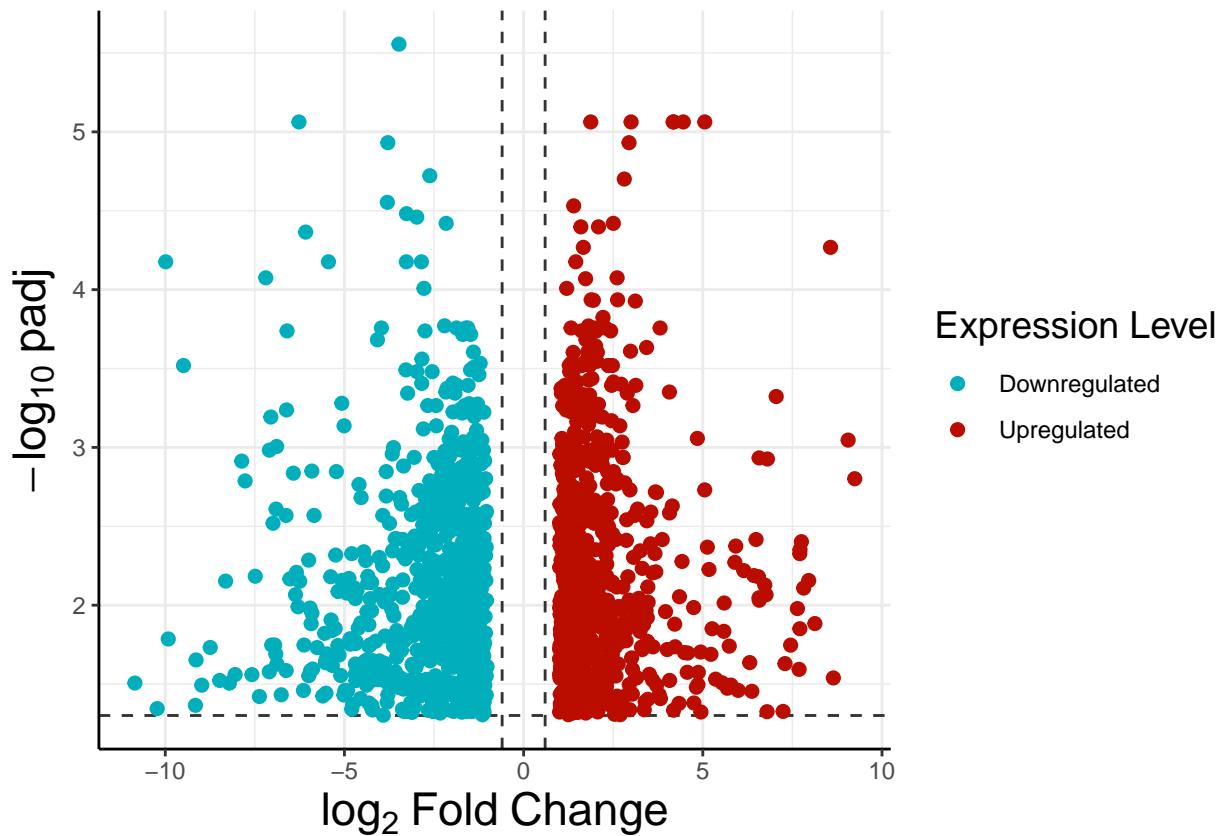
## Number of up-regulated genes: 838

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

## Number of down-regulated genes: 771

ggplot(data = North_South_overlap_30, 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
  )

```



## 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)

#head(North_South_overlap)

dim(North_South_overlap)

```

```

## [1] 2150    3

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

output_data_chronic_all <- cpm_count_filtered_final[, grep1("N|S", colnames(cpm_count_filtered_final))]

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

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

# merge the data frames based on the "gene_id" column
merged_df_chronic_all <- merge(data_gene_id_chronic_all, North_South_overlap[, c("gene_id", "logFC", "adjPValue")])

# Selecting the desired columns
final_table_chronic_all <- merged_df_chronic_all[, c("gene_id", "N20G1", "N20G2", "N20G3", "N20G5", "S20G1", "S20G2", "S20G3", "S20G5")]

# Convert character matrix to numeric matrix element-wise
North_South_all_overlap <- final_table_chronic_all[, c( "N20G1", "N20G2", "N20G3", "N20G5", "S20G1", "S20G2", "S20G3", "S20G5")]
North_South_all_overlap <- apply(North_South_all_overlap, 2, as.numeric)

# Set row names
rownames(North_South_all_overlap) <- final_table_chronic_all$gene_id

# Set column names
colnames(North_South_all_overlap) <- c("N20G1", "N20G2", "N20G3", "N20G5", "S20G1", "S20G2", "S20G3", "S20G5")

dim(North_South_all_overlap)

## [1] 2150    29

library(gplots)

gene_labels <- rownames(North_South_all_overlap)
condition_labels <- colnames(North_South_all_overlap)

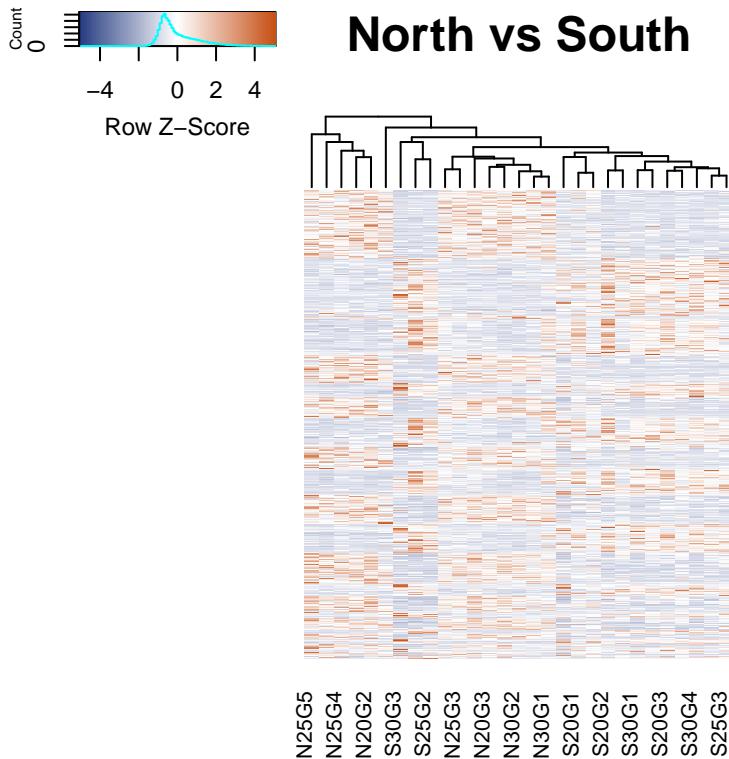
heatmap.2(North_South_all_overlap,
          scale = "row",
          trace = "none",
          col = colorRampPalette(colors = c("#273D82", "white", "#C75218"))(100),
          main = "North vs South",
          cex.main = 1,
          Colv = TRUE,
          hclustfun = function(c) hclust(c, method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

```



## 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 > 0.6 & North_South_overlap$adj.P.Val < 0.05] <- "Up regulated"
North_South_overlap$diffexpressed[North_South_overlap$logFC < -0.6 & North_South_overlap$adj.P.Val < 0.05] <- "Down regulated"

# number of DGes - up and down regulated
upregulated_count <- sum(North_South_overlap$diffexpressed == "Up regulated")
downregulated_count <- sum(North_South_overlap$diffexpressed == "Down regulated")
cat("Number of up-regulated genes:", upregulated_count, "\n")

```

```

## Number of up-regulated genes: 1020

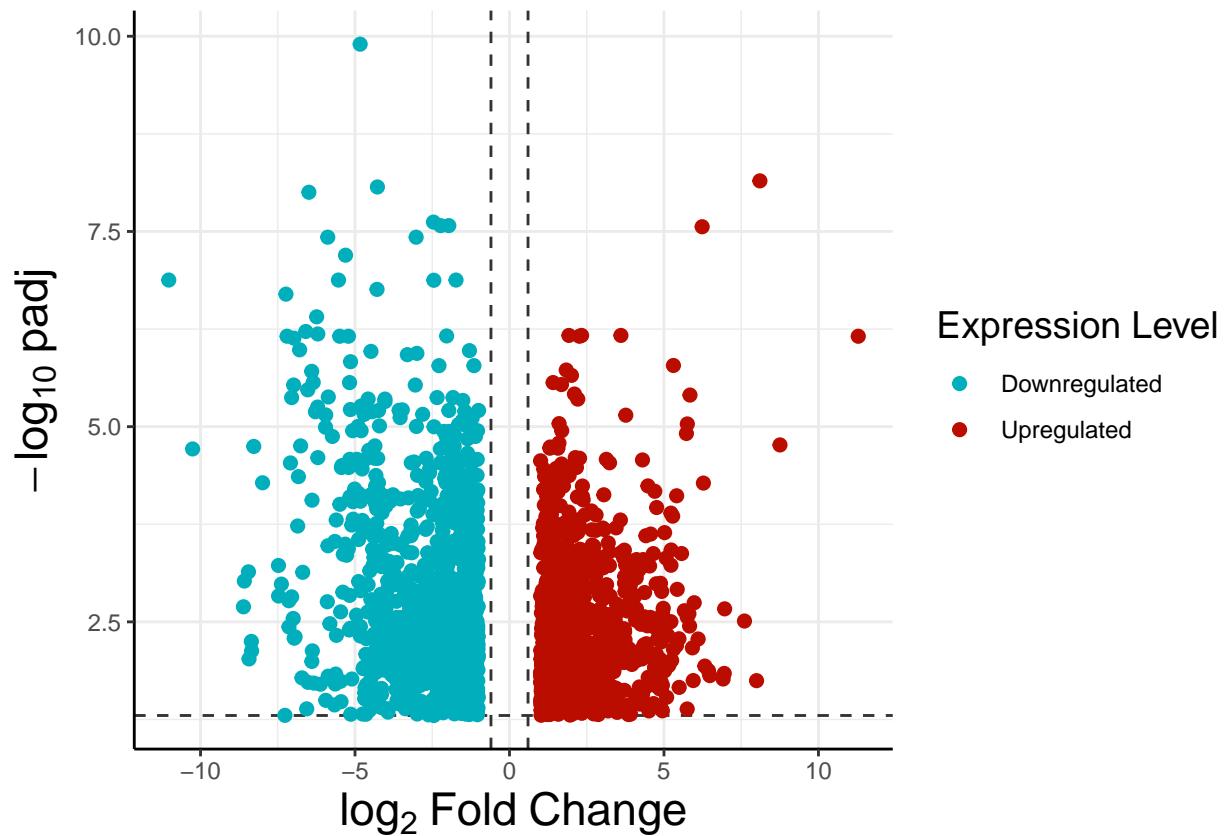
```

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

```
## Number of down-regulated genes: 1130
```

## 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
  )
```



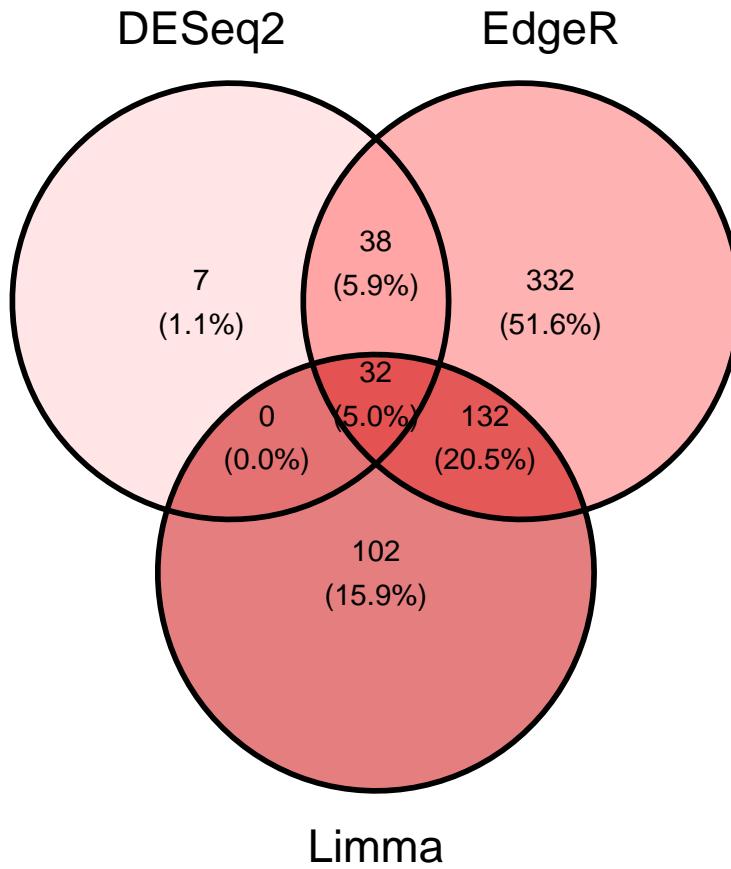
### Overlap 3 methods - South 20 °C vs 25 °C vs 30 °C

```
library(ggvenn)

DESeq2 <- DGEs_DESeq2_south_20_25_30$gene_id
EdgeR <- DGEs_EdgeR_south_20_25_30$gene_id
Limma <- DGEs_Limma_south_20_25_30$gene_id

# Create a list with each vector as an element
myList <- list(DESeq2 = DESeq2, EdgeR = EdgeR, Limma = Limma)

ggvenn(myList, fill_color = c("#FFCCCC", "#FF6666", "#CC0000"))
```



### Overlap 3 methods - North 20 °C vs 25 °C vs 30 °C

```

library(ggvenn)

DESeq2 <- DGEs_DESeq2_north_20_25_30$gene_id
EdgeR <- DGEs_EdgeR_north_20_25_30$gene_id
Limma <- DGEs_Limma_north_20_25_30$gene_id

# Create a list with each vector as an element
myList <- list(DESeq2 = DESeq2, EdgeR = EdgeR, Limma = Limma)

# Create a Venn diagram
ggvenn(myList, fill_color=c("#AED6F1", "#5499C7", "#1f618d"))

```

