Differential Gene Expression - Baggerley's Test

Payel Sardar

This script performs Differential Gene Expression Analysis using Unpaired sample Baggerley's Test and creates corresponding visualisations.

Reading and Pre-processing Datasets

Setting the home working directory.

```
setwd("D:/KCL2024/Courses/7BBG1002_Cloud_computing/Project")
```

Loading necessary packages

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.4.2
```

```
library(tidyr)
library(data.table)
```

```
Warning: package 'data.table' was built under R version 4.4.2
```

```
Attaching package: 'data.table'
```

```
The following objects are masked from 'package:dplyr':

    between, first, last
```

```
library(ggplot2)
```

Reading the file on sample information

```
sample_info <- read.csv("../metadata/sample_types.csv", sep = ",", header = TRUE)
head(sample_info)
```

```
         Run    BioSample  Bases   Bytes carbon_source Experiment GEO_Accession
1 SRR1166442 SAMN02639514 2.03 G 1.29 Gb       Glucose  SRX468698    GSM1324496
2 SRR1166443 SAMN02639516 2.25 G 1.42 Gb       Glucose  SRX468699    GSM1324497
3 SRR1166444 SAMN02639513 1.74 G 1.05 Gb       Glucose  SRX468700    GSM1324498
4 SRR1166445 SAMN02639515 2.28 G 1.44 Gb    Cellobiose  SRX468701    GSM1324499
5 SRR1166446 SAMN02639512 1.91 G 1.20 Gb    Cellobiose  SRX468702    GSM1324500
6 SRR1166447 SAMN02639517 1.73 G 1.04 Gb    Cellobiose  SRX468703    GSM1324501
          create_date Sample.Name            source_name
1 2014-02-1010:25:00Z  GSM1324496    Glucose-grown cells
2 2014-02-1010:25:00Z  GSM1324497    Glucose-grown cells
3 2014-02-1010:24:00Z  GSM1324498    Glucose-grown cells
4 2014-02-1010:25:00Z  GSM1324499 Cellobiose-grown cells
5 2014-02-1010:24:00Z  GSM1324500 Cellobiose-grown cells
6 2014-02-1010:24:00Z  GSM1324501 Cellobiose-grown cells
```

Reading the Normalized RPKM data for Glucose and Cellobiose samples

```
glucose_data <- read.csv("../data/processed/Normalized_data/glucose_merged.csv", header = TRU
cellobiose_data <- read.csv("../data/processed/Normalized_data/cellobiose_merged.csv", header

head(glucose_data)
```

```
     Geneid SRR1166442 SRR1166443 SRR1166444
1   YAL068C  8.4379100 11.1595834  10.898455
2 YAL067W-A  0.3630822  0.3290228   0.564276
3   YAL067C 55.5291618 53.1827885  40.971760
4   YAL065C  3.4225421  3.5537864   2.825754
5 YAL064W-B  3.3315888  4.0691744   3.292351
6 YAL064C-A 10.1396180 10.1729361   8.273086
```

```
head(cellobiose_data)
```

```
     Geneid SRR1166445 SRR1166446 SRR1166447
1   YAL068C 11.6956377  9.3393236 10.1604814
2 YAL067W-A  0.5115573  0.4956395  0.1336905
3   YAL067C 15.5382521 17.9623442 15.2407221
4   YAL065C  1.8685722  2.3360375  2.5991929
5 YAL064W-B  3.0000621  2.7435795  2.8801365
6 YAL064C-A 10.1022500  7.8599846 10.0004739
```

Setting Geneid as row names and dropping the Geneid column to obtain a numeric dataset for downstream analysis

```
gene_ids <- glucose_data$Geneid

row.names(glucose_data) <- glucose_data$Geneid
glucose_data <- glucose_data[, -1]

row.names(cellobiose_data) <- cellobiose_data$Geneid
cellobiose_data <- cellobiose_data[, -1]
```

Differential Gene Expression Analysis

Performing an unpaired sample analysis using the Baggerley's test

Function for performing Baggerley's test

```
baggerley_test <- function(prop1, prop2, pseudo_count = 0.001) {

  prop1 <- as.numeric(prop1)
  prop2 <- as.numeric(prop2)
```

```
  # Adding pseudo-count to avoid zero mean
  prop1 <- prop1 + pseudo_count
  prop2 <- prop2 + pseudo_count

  # Computing the sample size of each group
  n1 <- length(prop1)
  n2 <- length(prop2)

  # Computing the mean for each group
  mean1 <- mean(prop1)
  mean2 <- mean(prop2)

  # Compute the sample variance for each group
  var1 <- var(prop1)  # Variance for group 1
  var2 <- var(prop2)  # Variance for group 2

  # Calculate pooled variance
  #Adding 1e-6 prevent "Divide by zero" error in Z-score computation
  pooled_var <- (((n1 - 1) * var1) + ((n2 - 1) * var2)) / (n1 + n2 - 2) + 1e-6

  # Z-score using pooled standard deviation
  z <- (mean1 - mean2) / sqrt(pooled_var * (1/n1 + 1/n2))

  # Two-tailed test
  p_value <- 2 * (1 - pnorm(abs(z)))

  return(p_value)
}
```

Iterating over rows(genes) to compute p-values for each gene using the Baggerley's test.

```
baggerley_results <- sapply(1:nrow(glucose_data), function(i) {
  prop1 <- cellobiose_data[i, ]
  prop2 <- glucose_data[i, ]
  baggerley_test(prop1, prop2, 0.001)
})
```

Creating a dataframe to store the results from the Baggerley's test

```
# Storing results (p-values) in a data frame
result_df2 <- data.frame(
```

```
  # Mean Normalised RPKM for glucose
  glucose_norm_rpkm_mean = rowMeans(glucose_data),
  # Mean Normalised RPKM for cellobiose samples
  cellobiose_norm_rpkm_mean = rowMeans(cellobiose_data),
  # P-values from Baggerley's test
  p_value = baggerley_results
)
```

FDR correction for P-value

```
result_df2$adjusted_p_value <- p.adjust(result_df2$p_value, method = "fdr")
```

Computing the fold change and log2 transformed fold change

```
# Pseudo-count of 0.001 is added to prevent 0/0 error or log(0)
result_df2$fold_change <- (result_df2$cellobiose_norm_rpkm_mean + 0.001) /
  (result_df2$glucose_norm_rpkm_mean + 0.001)
result_df2$log2_fold_change <-  log2(abs(result_df2$fold_change))
```

Extracting the Differentially Expressed genes based on the significance threshold used in the original study

```
significant_genes <- result_df2[result_df2$adjusted_p_value <= 0.001 &
                                  abs(result_df2$log2_fold_change) >= 1.0, ]
head(significant_genes)
```

```
          glucose_norm_rpkm_mean cellobiose_norm_rpkm_mean      p_value
YAL067C                49.894570                16.2471061 2.406964e-13
YAL062W               141.841053               331.2851169 0.000000e+00
YAL054C                18.816943                53.6567692 0.000000e+00
YAL044C              1128.504614               423.9457399 0.000000e+00
YAL039C               110.846294               388.6617995 0.000000e+00
YNCA0001W               2.994296                 0.9976067 2.933395e-07
          adjusted_p_value fold_change log2_fold_change
YAL067C       6.342515e-13   0.3256423        -1.618640
YAL062W       0.000000e+00   2.3355987         1.223792
YAL054C       0.000000e+00   2.8514152         1.511678
YAL044C       0.000000e+00   0.3756709        -1.412459
YAL039C       0.000000e+00   3.5062904         1.809945
YNCA0001W     5.804949e-07   0.3333916        -1.584710
```

Annotating the Differentially Genes

The GAF file used in the analysis was modified by removing the header (comment section) that begins with '!'.

```
gaf_data <- as.data.frame(fread("../metadata/sgd.gaf/sgd_noheader.gaf",
                                sep = "\t", quote = "", fill = TRUE))
gaf_data <- gaf_data[gaf_data[,1]=="SGD",]
gaf_data <- gaf_data[, c(3, 10,11)]
colnames(gaf_data) <- c("feature_id", "description", "synonym")
```

Cleaning up the gene synonyms and preparing to merge with DEG results

```
gaf_data$gene_id <- sub("\\|.*", "", gaf_data$synonym)
gaf_data <- gaf_data %>% distinct()
```

Annotating the significant DEGs with the additional information from the GAF file

```
significant_genes$gene_id <- rownames(significant_genes)
annotated_degs <- gaf_data %>%
  right_join(significant_genes, by = "gene_id")
annotated_degs$feature_id <- coalesce(annotated_degs$feature_id, annotated_degs$gene_id)
head(annotated_degs)
```

```
  feature_id
1       ATM1
2    YJL218W
3    YPL264C
4       SAM1
5       CYC1
6       TMT1
                                                        description
1 Mitochondrial inner membrane ATP-binding cassette (ABC) transporter
2                   Mitochondrial protein, putative acetyltransferase
3                   Endoplasmic reticulum protein of unknown function
4                                    S-adenosylmethionine synthetase
5                                            Cytochrome c, isoform 1
6                                 Trans-aconitate methyltransferase
                                                              synonym gene_id
1 YMR301C|ATP-binding cassette Fe/S cluster precursor transporter ATM1 YMR301C
2                                          YJL218W|acetyltransferase YJL218W
```

```
3                                                          YPL264C YPL264C
4                    YLR180W|ETH10|methionine adenosyltransferase SAM1 YLR180W
5                                 YJR048W|cytochrome c isoform 1 YJR048W
6                     YER175C|TAM1|trans-aconitate 3-methyltransferase YER175C
  glucose_norm_rpkm_mean cellobiose_norm_rpkm_mean     p_value adjusted_p_value
1              54.439460                  21.50717 0.000000000     0.0000000000
2              27.132087                  12.11236 0.000000000     0.0000000000
3              73.559829                  17.60086 0.000000000     0.0000000000
4            1224.673346                 167.15200 0.000000000     0.0000000000
5               9.374713                  24.66442 0.000152921     0.0002562663
6             361.766324                  81.70760 0.000000000     0.0000000000
  fold_change log2_fold_change
1   0.3950769        -1.339794
2   0.4464426        -1.163454
3   0.2392830        -2.063210
4   0.1364877        -2.873157
5   2.6307783         1.395490
6   0.2258595        -2.146502
```

Exporting the results for downstream analysis

```
write.csv(annotated_degs, "../Output/results/Annotated_degs_Baggerley.csv",
          row.names = FALSE, fileEncoding = "UTF-8")
write(annotated_degs$feature_id, "../Output/results/FeatureId.txt", ncolumns = 1)
write(annotated_degs$gene_id, "../Output/results/GeneId.txt", ncolumns = 1)
```

Extracting information about the transcription factors mentioned in the original study

```
tf_list <- c('MET32', 'MET28', 'THI2', 'MIG2', 'UGA3', 'SIP4',
             'MIG3', 'HMS2', 'KAR4', 'MAL13', 'YAP5', 'DAL80','ADR1',
             'USV1', 'CAT8', 'GSM1', 'XBP1', 'SUT1', 'HAP4')
tf_dge <- annotated_degs %>% filter(feature_id %in% tf_list)

#write.csv(tf_dge, "../Output/results/TranscriptionFactors_degs.csv",
#          row.names = FALSE, fileEncoding = "UTF-8")
```

Data Visualisation

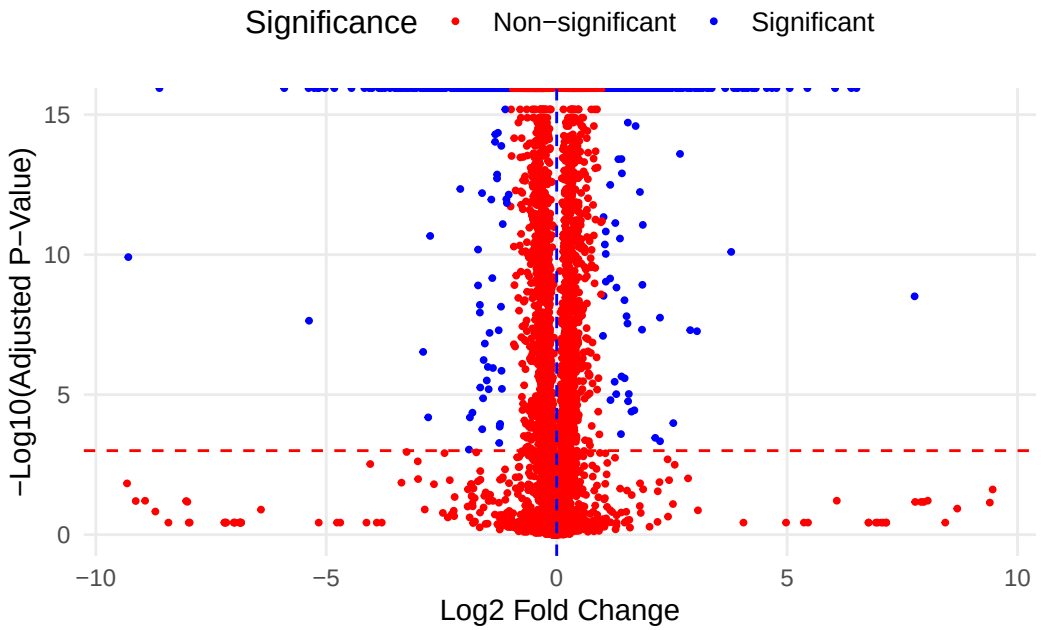
Volcano plot for all differentially expressed genes

```
# Creating a new column in the data frame to indicate significance
result_df2$Significance <- ifelse(result_df2$adjusted_p_value <= 0.001 &
                                    abs(result_df2$log2_fold_change) >= 1.0,
                                  "Significant", "Non-significant")

# Create the volcano plot with the updated 'Significance' column
volcano_plot <- ggplot(result_df2, aes(x = log2_fold_change,
                                       y = -log10(adjusted_p_value))) +
  geom_point(size = 0.7, aes(color = Significance)) +
  geom_hline(yintercept = -log10(0.001), color = "red", linetype = "dashed") +
  geom_vline(xintercept = 0, color = "blue", linetype = "dashed") +
  labs(x = "Log2 Fold Change", y = "-Log10(Adjusted P-Value)",
       color = "Significance") +

  # Custom color scale for significance
  scale_color_manual(values = c("Non-significant" = "red", "Significant" = "blue")) +
  theme_minimal() +
  theme(
    legend.position = "top",
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10),
    panel.grid.minor = element_blank()
  )

print(volcano_plot)
```



Saving the plot as jpg for preparing reports

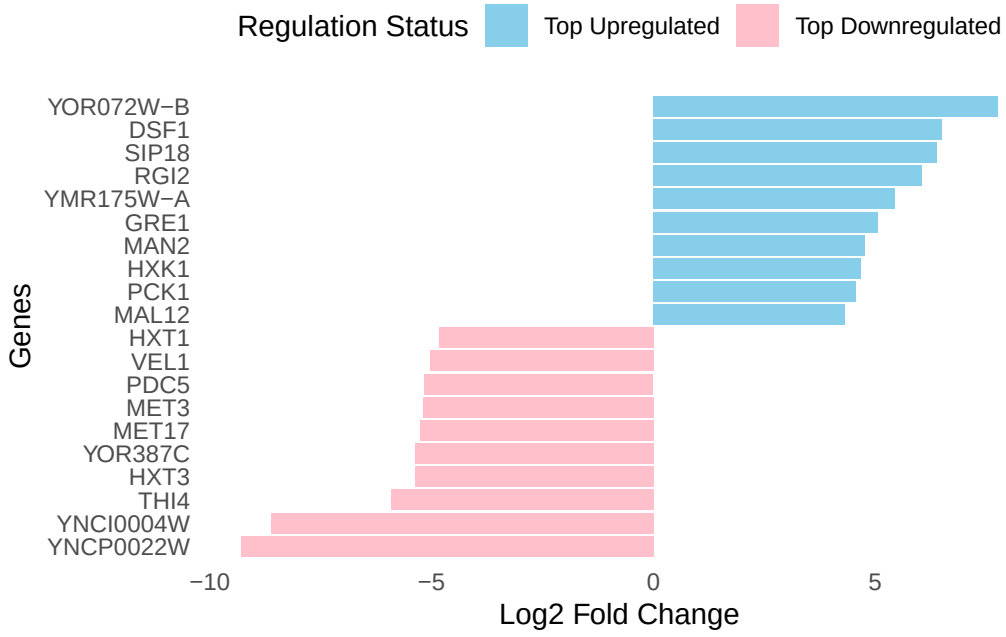
```
ggsave(
  filename = "../Output/plots/VolcanoPlot_Baggerley.jpg",
  plot = volcano_plot,
  width = 6, height = 4,
  dpi = 300
)
```

Visualising the top 10 upregulated and downregulated differentially expressed genes

```
# sorting the annotated DEGs based on their log2(fold change) in ascending order
deg_sorted <- annotated_degs[order(-annotated_degs$log2_fold_change), ]
top_upregulated <- head(deg_sorted, 10)
top_downregulated <- tail(deg_sorted, 10)
top_genes <- rbind(top_upregulated, top_downregulated)

# Making a bar
top10 <- ggplot(top_genes, aes(x = reorder(feature_id, log2_fold_change),
                      y = log2_fold_change,
                      fill = log2_fold_change < 0)) +
```

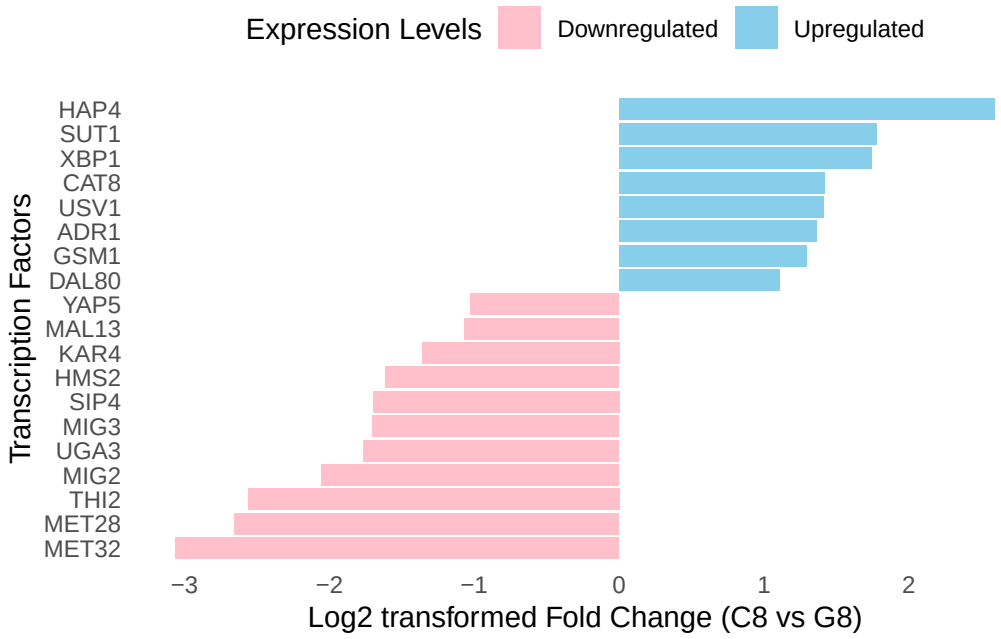
```
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("skyblue", "pink"),
                    labels = c("Top Upregulated", "Top Downregulated")) +
  labs(x = "Genes",
       y = "Log2 Fold Change",
       fill = "Regulation Status") +
  theme_minimal() +
  theme(
    panel.grid.major = element_blank(), # Remove major gridlines
    panel.grid.minor = element_blank(), # Remove minor gridlines
    legend.position = "top" # Move legend to the top
  ) +
  coord_flip()  # Flip for better readability
print(top10)
```




```
ggsave(
  filename = "../Output/plots/Top_DEG_Baggerley.jpg",
  plot = top10,
  width = 6, height = 4,
  dpi = 300
)
```

Preparing a bar plot to compare the fold change of differentially expressed transcription factors.

```
tf_fc_plot <- ggplot(tf_dge, aes(x = reorder(feature_id, fold_change),
                     y = log2_fold_change,
                     fill = log2_fold_change > 0)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("pink", "skyblue"),
                    labels = c("Downregulated", "Upregulated")) +
  labs(
    x = "Transcription Factors",
    y = "Log2 transformed Fold Change (C8 vs G8)",
    fill = "Expression Levels"
  ) +
  theme_minimal() +
  theme(
    panel.grid.major = element_blank(), # Remove major gridlines
    panel.grid.minor = element_blank(), # Remove minor gridlines
    legend.position = "top" # Move legend to the top
  ) +
  coord_flip()  # Flip for better readability
print(tf_fc_plot)
```



Saving the barplot

```
ggsave(
  filename = "../Output/plots/Fold_change_tf_Baggerley.jpg",
  plot = tf_fc_plot,
  width = 6, height = 4,
  dpi = 300
)
```

Comparing with the DeSeq results

Reading the DeSEQ results file

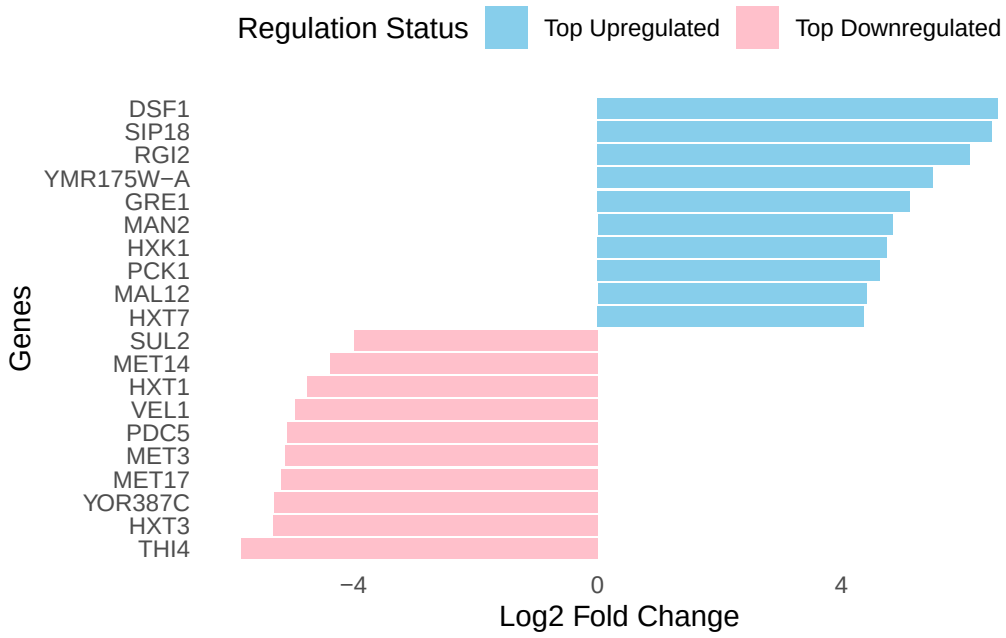
```
deseq <- read.csv("../Output/results/sig_DEG_DeSEQ.csv", header=TRUE)
deseq_annotated <- gaf_data %>%
  right_join(deseq, by = "gene_id")
deseq_annotated$log2FoldChange <- (-1)*deseq_annotated$log2FoldChange
```

Making the Bar plot for the Top upregulated and Downregulated genes

```
# sorting the annotated DEGs based on their log2(fold change) in ascending order
deseq_sorted <- deseq_annotated[order(-deseq_annotated$log2FoldChange), ]
top_upregulated <- head(deseq_sorted, 10)
top_downregulated <- tail(deseq_sorted, 10)
top_genes <- rbind(top_upregulated, top_downregulated)

# Making a bar
top10 <- ggplot(top_genes, aes(x = reorder(feature_id, log2FoldChange),
                      y = log2FoldChange,
                      fill = log2FoldChange < 0)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("skyblue", "pink"),
                    labels = c("Top Upregulated", "Top Downregulated")) +
  labs(x = "Genes",
       y = "Log2 Fold Change",
       fill = "Regulation Status") +
  theme_minimal() +
  theme(
    panel.grid.major = element_blank(), # Remove major gridlines
    panel.grid.minor = element_blank(), # Remove minor gridlines
```

```
    legend.position = "top" # Move legend to the top
  ) +
  coord_flip()  # Flip for better readability
print(top10)
```



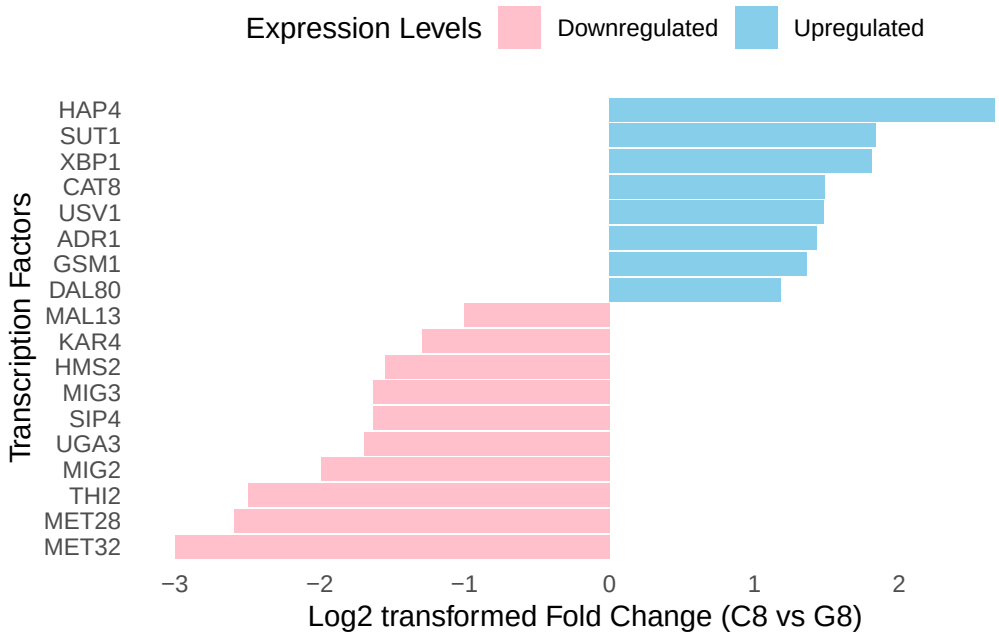

```
ggsave(
  filename = "../Output/plots/Top_DEG_DESeq.jpg",
  plot = top10,
  width = 6, height = 4,
  dpi = 300
)
```

Visualising the Expression profile of the differentially Transcription factors

```
tf_deseq <- deseq_annotated %>% filter(feature_id %in% tf_list)

tf_fc_plot <- ggplot(tf_deseq, aes(x = reorder(feature_id, log2FoldChange),
                    y = log2FoldChange,
                    fill = log2FoldChange > 0)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("pink", "skyblue"),
                    labels = c("Downregulated", "Upregulated")) +
```

```
  labs(
    x = "Transcription Factors",
    y = "Log2 transformed Fold Change (C8 vs G8)",
    fill = "Expression Levels"
  ) +
  theme_minimal() +
  theme(
    panel.grid.major = element_blank(), # Remove major gridlines
    panel.grid.minor = element_blank(), # Remove minor gridlines
    legend.position = "top" # Move legend to the top
  ) +
  coord_flip()  # Flip for better readability
print(tf_fc_plot)
```




```
ggsave(
  filename = "../Output/plots/Fold_change_tf_DESeq.jpg",
  plot = tf_fc_plot,
  width = 6, height = 4,
  dpi = 300
)
```