# Report on Assessment Test

Name – Aditya naman soni

## A. Introduction

This report presents the results of a differential expression analysis, exploratory, and gene ontology analysis performed on provided counts and target identifiers. The analysis aimed to identify and explore differentially expressed genes between different experimental conditions. This analysis summarizes the relationship and significance of genes among various conditions. This information will give insights into diverse biological conditions. I used RStudio version 4.3.2 for this analysis.

## B. Data preparation

I used given raw counts and target identifier files to conduct the analysis. The raw count file contains counts across three replicates of each sample. The target identifier file contains the details of sample (n =48) characteristics, including replicate ID, sample ID, identifier, and file name.

### Differential expression (DE) analysis

#### Making a unique combination of identifiers

To prepare a master expression file, DE must perform between each combination of 48 identifiers. The total combination of unique identifiers was 1128.

$$C(n,k) = k!\,(n-k)!/n!$$

where,

*n* = the total number of unique identifiers
*k* = the number of identifiers per combination

Next, I used a double for{} loop to combine two unique identifiers. This loop resembles the combination formula and generates unique combinations. Furthermore, I used this loop to subset the unique identifier combination data from the collapsed dataset. Subsequently, this subset dataset was used for differential analysis. Next, I stored the generated results in a list with the element name of the identifier combination.

### Differential analysis was done in the following seven steps, which are given below

a. Convert matrix data into DESeq dataset
b. Replicates collapse by Identifier group
c. Set reference level to the identifier
d. Removal of low counts (< 2, total number of samples in a combination of identifiers) to clean data and improve robustness.
e. Perform DESeq analysis to identify significant differentially expressed genes.
f. Generate results and summary to simplify the interpretation
g. Store result data in a structured list

## C. Summarizing DEGs results

Names of all unique combinations stored in a vector. This vector is used in a loop to save a summary of individual combination differential analysis results in "Master Summary file.txt" utilizing *sink()* function.

### Saving filtered results in an Excel sheet

I prepared a structured list to ensure organized storage of DESeq results of combinations. Then, I created a vector  to store names of identifier combinations. Utilizing this vector, a loop was created to subset the data from the list containing DESeq results for filtered results.

I utilized dplyr::filter() to filter the data from the result based on padj < 0.05 and log fold change > 2. This filtered data was sorted and appended to individual sheets in the workbook (file name = DEG_workbook.xlsx). For feasible navigation and access, the names of identifier combinations were assigned to the respective sheet.

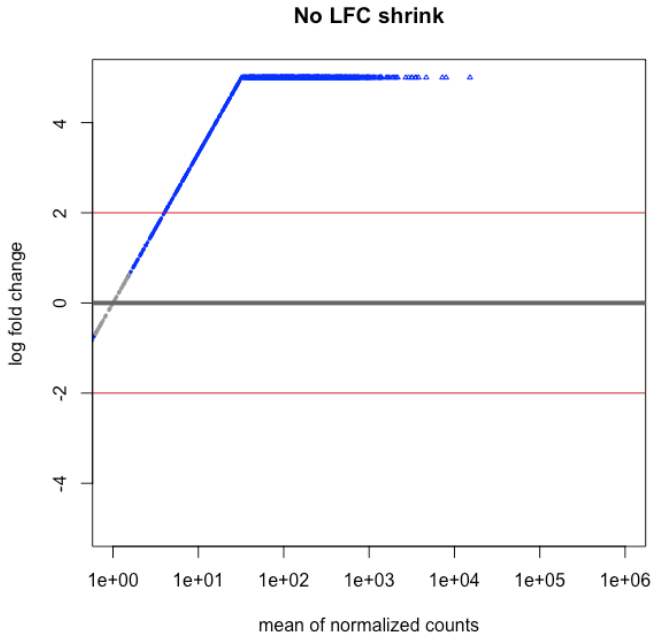
## D. Generate plots

Informative plots were generated to visualize the magnitude and significance of DE genes.

### MA plot

This plot was generated to understand the relationship between the mean of normalized counts and logFC. No LFC shrinkage was used to prepare this plot.

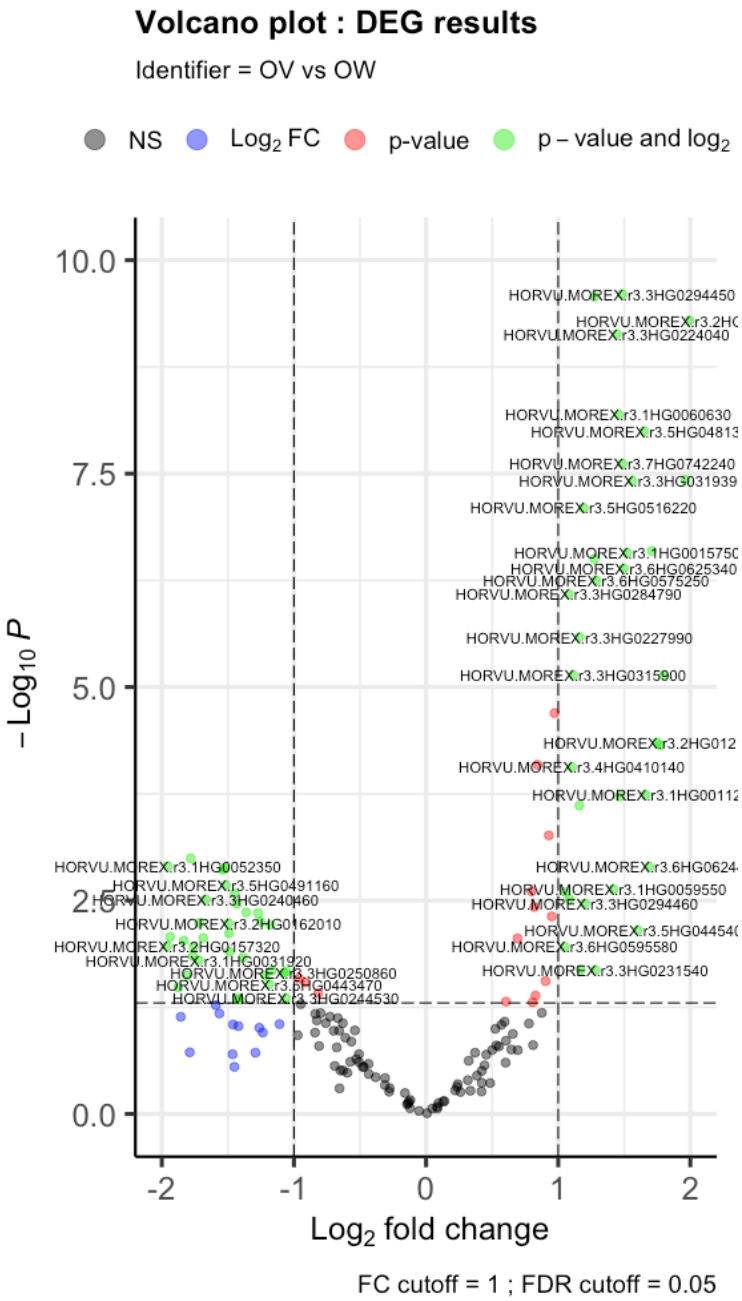
a. Function used = DESeq2::plotMA()
b. DE result of identifiers OV and OW combination was utilized to generate this plot.
c. Axis limit was set :  x-axis  = mean of normalized counts (range = 1 to 1e6); y-axis = logFC (range = -5 to 5)
d. A horizontal straight line was added to distinguish the plot by logFC = 2



Volcano plot

This plot was generated to understand the relationship between logFC and padj.

a. Function used = EnhancedVolcano:: EnhancedVolcano()
b. DE result identifiers OV and OW were used to make a volcano plot.
c. The thresholds used were logFC =1 and padj = 0.05 to highlight the gene (color set to green).



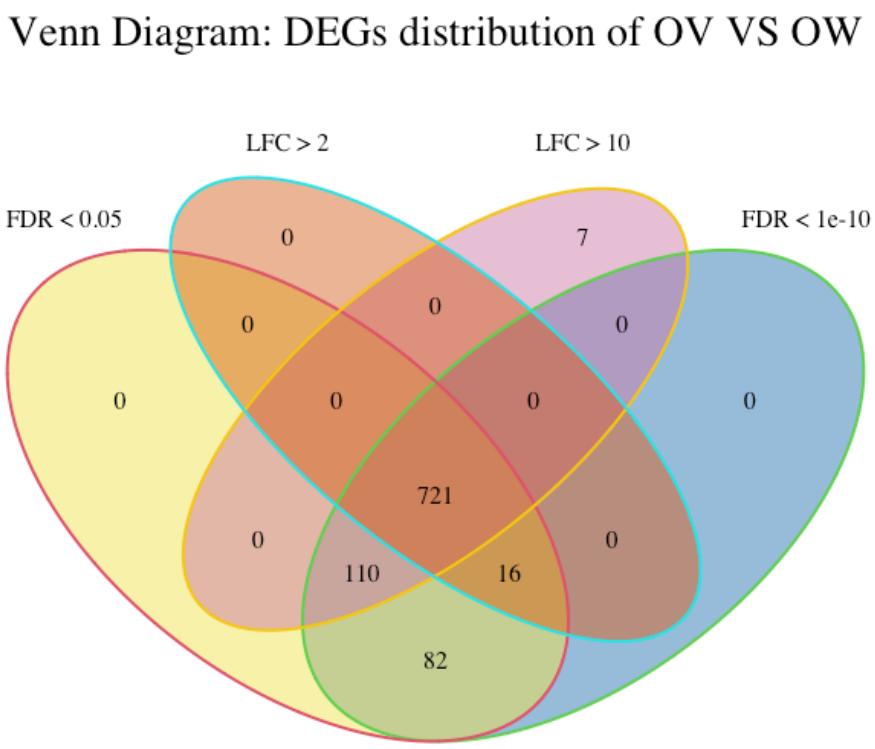

Venn diagram

This diagram was made to gain insights into overlapping numbers of genes under different thresholds. Data preparation was done on multiple sets based on padj and logFC.

i. Dplyr::filter() was used to filter data and prepare multiple sets.

ii. 4 sets were made .i.e., set1 = padj < 0.1, set2 = padj < 1e-10, set3 = logFC > 2, set4 = logFC > 10
iii. These sets of filtered data were stored in a list

b. VennDiagram:: Venn.diagram() was applied on the list containing sets to create the diagram




# E. Downstream analysis

## Gene ontology

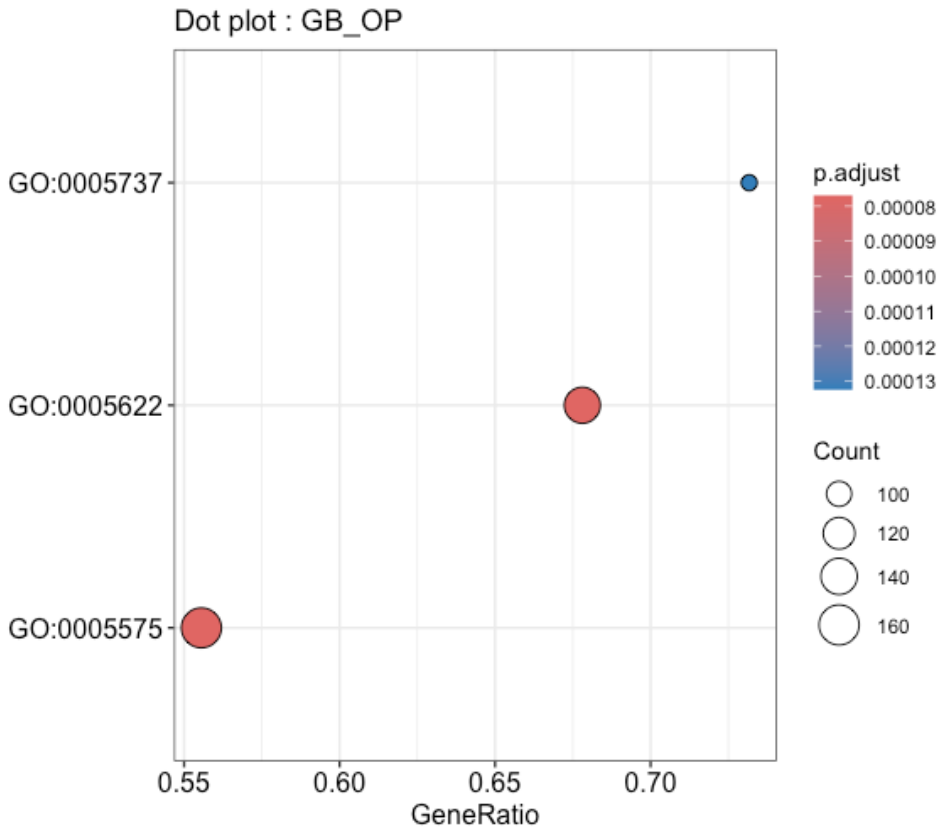
GO analysis was done to know about the functional role of the DE genes.

a. I used DE analysis result of GB + OP identifier from the saved data.
b. Subsequently, gprofiler2:: gconvert() was used to fetch and retrieve genes' gene ontology (GO) information using their ENSEMBL IDs from the gprofiler database.
c. GO IDs were saved in a vector for GSEA analysis.

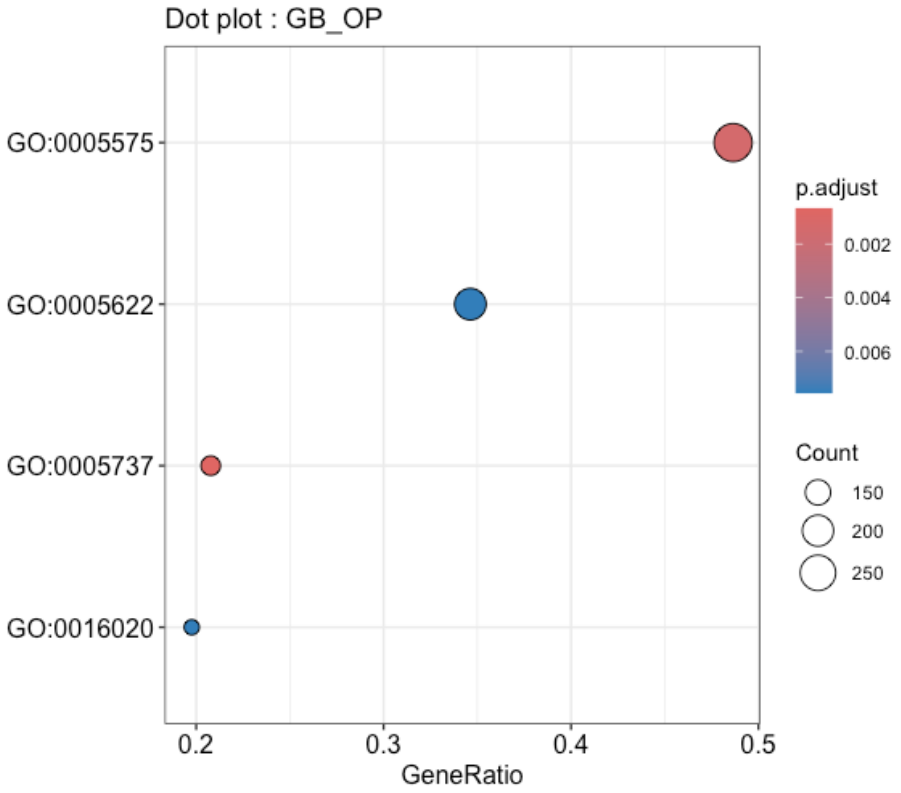
## Gene set enrichment analysis (GSEA)

An annotation database of Hordeum vulgare (non-model organism) is unavailable. Therefore, I performed universal gene set enrichment to identify the enriched gene sets.

a. clusterProfiler:: GSEA() was used for universal GSEA of the data by *fgsea*. The thresholds used for analysis are given below:
    i. The weight of each step was set to 1
    ii. Min gene set size = 15 and Max gene set size = 500
    iii. FDR = 0.05
b. A dot plot was created for the visualization of enriched gene ontologies.




In addition, I have done a universal enrichment analyzer to compare the difference between enricher() and GSEA(). I found a difference of one GO between GSEA and enricher.



KEGG pathway analysis

1. *Hordeum vulgare is unavailable in the KEGG organism list. Therefore, there is a need for some other way of pathway analysis for this species.*
2. No gene information in the pathway database was present in the biomart's "hvulgare_eg_gene" dataset.