RNAseq - Differential Expression

Ahmed Sameh

2024-04-16

Importing the abundance data created from Kallisto

Creating the data DESeq matrix stored in the **dds** variable while the data is present in the **results** variable with NA values removed.

```
# Define the samples and their conditions
samples <- data.frame(
  condition = factor(c("control", "control","control","control",
                       "AD", "AD", "AD", "AD")),
  path = c(
"/Users/ahmedsameh/Downloads/Proteinea/LD/RNAseq/DEGs/data/control_output/abundance.tsv",
"/Users/ahmedsameh/Downloads/Proteinea/LD/RNAseq/DEGs/data/control_output2/abundance.tsv",
"/Users/ahmedsameh/Downloads/Proteinea/LD/RNAseq/DEGs/data/control_output3/abundance.tsv",
"/Users/ahmedsameh/Downloads/Proteinea/LD/RNAseq/DEGs/data/control_output4/abundance.tsv",
"/Users/ahmedsameh/Downloads/Proteinea/LD/RNAseq/DEGs/data/ad_output/abundance.tsv",
"/Users/ahmedsameh/Downloads/Proteinea/LD/RNAseq/DEGs/data/ad_output2/abundance.tsv",
"/Users/ahmedsameh/Downloads/Proteinea/LD/RNAseq/DEGs/data/ad_output3/abundance.tsv",
"/Users/ahmedsameh/Downloads/Proteinea/LD/RNAseq/DEGs/data/ad_output4/abundance.tsv"),
  row.names = c("control1", "control2","control3", "control4", "AD1", "AD2", "AD3", "AD4")
)

# Use tximport to read in the counts from Kallisto
files <- file.path(samples$path)
names(files) <- rownames(samples)
txi <- tximport(files, type = "kallisto", txOut = TRUE)
```

```
## Note: importing `abundance.h5` is typically faster than `abundance.tsv`
```

```
## reading in files with read_tsv
```

```
## 1 2 3 4 5 6 7 8
```

```
# Create a DESeqDataSet
dds <- DESeqDataSetFromTximport(txi, colData = samples, design = ~ condition)
```

```
##   it appears that the last variable in the design formula, 'condition',
##   has a factor level, 'control', which is not the reference level. we recommend
##   to use factor(...,levels=...) or relevel() to set this as the reference level
##   before proceeding. for more information, please see the 'Note on factor levels'
##   in vignette('DESeq2').
## using counts and average transcript lengths from tximport
```

```
# Run the DESeq pipeline
dds <- DESeq(dds)
```

```
## estimating size factors
```

```
## using 'avgTxLength' from assays(dds), correcting for library size
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## -- note: fitType='parametric', but the dispersion trend was not well captured by the
##    function: y = a/x + b, and a local regression fit was automatically substituted.
##    specify fitType='local' or 'mean' to avoid this message next time.
## final dispersion estimates
## fitting model and testing
```

```
# Extract results
results_ <- results(dds)
results <- na.omit(results_)
```

Annotation

```
ensembl <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")
t_ids = sub("\\..*", "", results@rownames)
gene_ids <- getBM(attributes = c("ensembl_transcript_id", "ensembl_gene_id"),
    filters = "ensembl_transcript_id", values = t_ids, mart = ensembl)
gene_ <- as.data.frame(gene_ids)

df_results <- as.data.frame(results)
df_results$transcript_id <- sub("\\..*", "", results@rownames)

merged_data <- merge(df_results, gene_, by.x = "transcript_id",
    by.y = "ensembl_transcript_id", all.x = TRUE)
merged_data <- na.omit(merged_data)

gene_info <- getBM(attributes = c("ensembl_gene_id", "external_gene_name"),
    filters = "ensembl_gene_id", values = gene_ids$ensembl_gene_id,
    mart = ensembl)

data_with_gene <- merge(merged_data, gene_info, by.x = "ensembl_gene_id",
    by.y = "ensembl_gene_id", all.x = TRUE)
data_with_gene <- na.omit(data_with_gene)
```

MA-Plot

need to understand still

```
ma_df <- data_with_gene
ma_df$significance <- ifelse(ma_df$padj < 0.05, "Significant", "Not Significant")
ma_df <- ma_df %>%
  arrange(padj) %>%
  mutate(rank = row_number())
# Select top 30 significant genes
top_genes <- ma_df[ma_df$rank <= 30, ]
ggplot(ma_df, aes(x = baseMean, y = log2FoldChange)) +
  geom_point(aes(color = padj < 0.05), alpha = 0.5) +  # Color code significant points
  scale_color_manual(values = c("grey", "red")) +
  geom_text(data = top_genes, aes(label = external_gene_name),
            vjust = 1.5, color = "cyan4", size = 3, check_overlap = TRUE) +
  theme_minimal() +
```

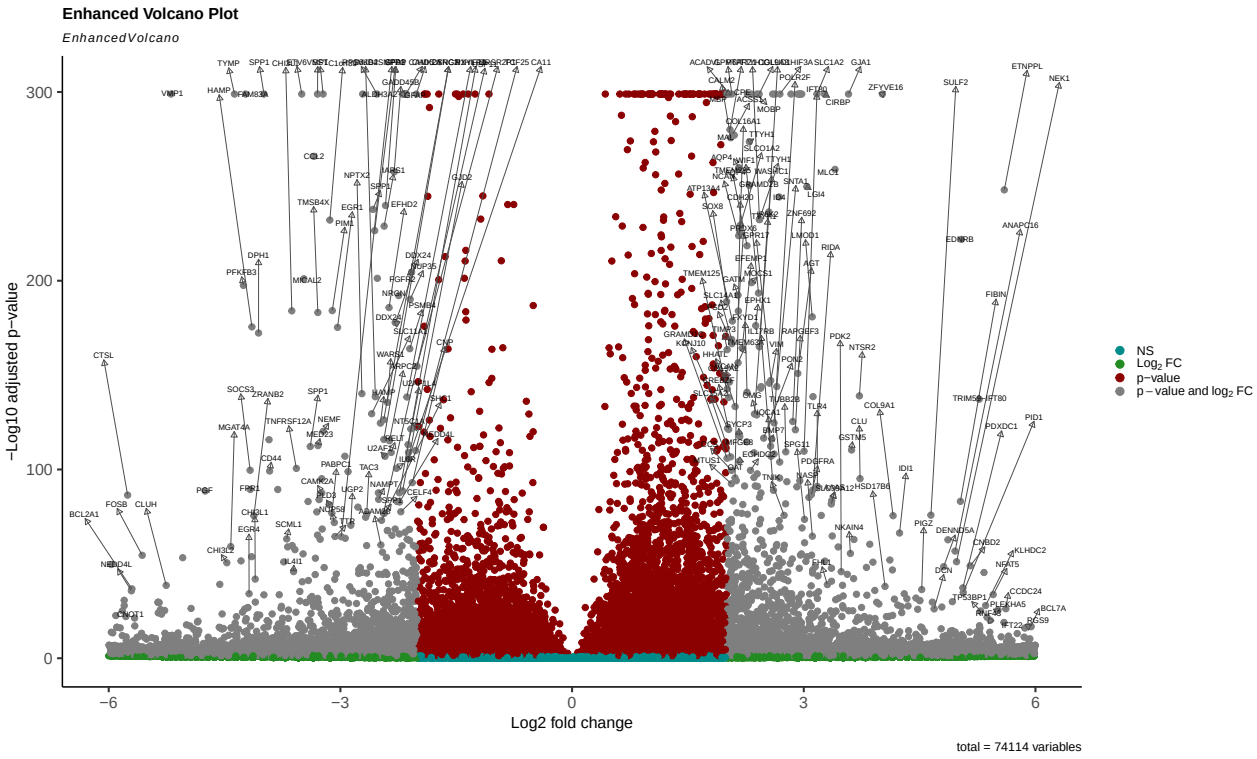
```
  labs(title = "MA Plot with Top 30 Gene Annotations",
       x = "Average Expression",
       y = "Log2 Fold Change") +
  geom_hline(yintercept = 0, linetype = "dotted") +
  theme(legend.position = "none") +
  labs(title="Enhanced MA-plot with Gene Labels",
       x="Mean of Normalized Counts",
       y="Log2 Fold Change") +
  geom_hline(yintercept=0, linetype="dashed")
```

Volcano Plot for significance visualization

```
EnhancedVolcano::EnhancedVolcano(data_with_gene, lab = data_with_gene$external_gene_name,
    x = "log2FoldChange", y = "padj", xlim = c(-6, 6), title = "Enhanced Volcano Plot",
    xlab = "Log2 fold change", ylab = "-Log10 adjusted p-value",
    pCutoff = 0.05, FCcutoff = 2, pointSize = 3, labSize = 3,
    col = c("cyan4", "forestgreen", "red4", "grey50"), colAlpha = 1,
    colCustom = NULL, shapeCustom = NULL, cutoffLineType = "blank",
    cutoffLineWidth = 0.8, cutoffLineCol = "black", legendPosition = "right",
    legendLabSize = 14, legendIconSize = 4, drawConnectors = TRUE,
    widthConnectors = 0.5, colConnectors = "grey30", gridlines.major = FALSE,
    gridlines.minor = FALSE, border = "partial")
```

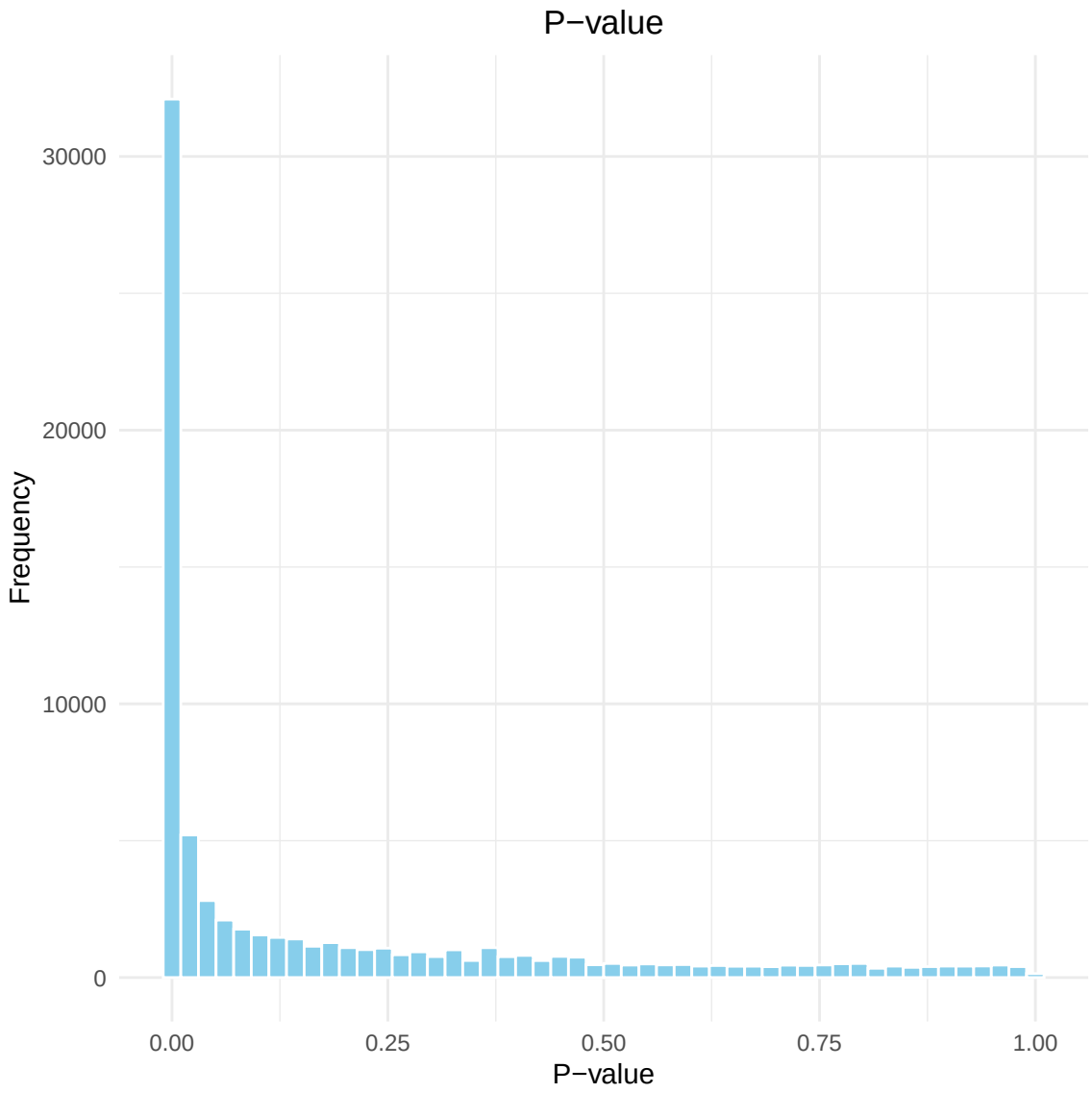
```
## Warning: One or more p-values is 0. Converting to 10^-1 * current lowest
## non-zero p-value...
```

```
## Warning: ggrepel: 12930 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

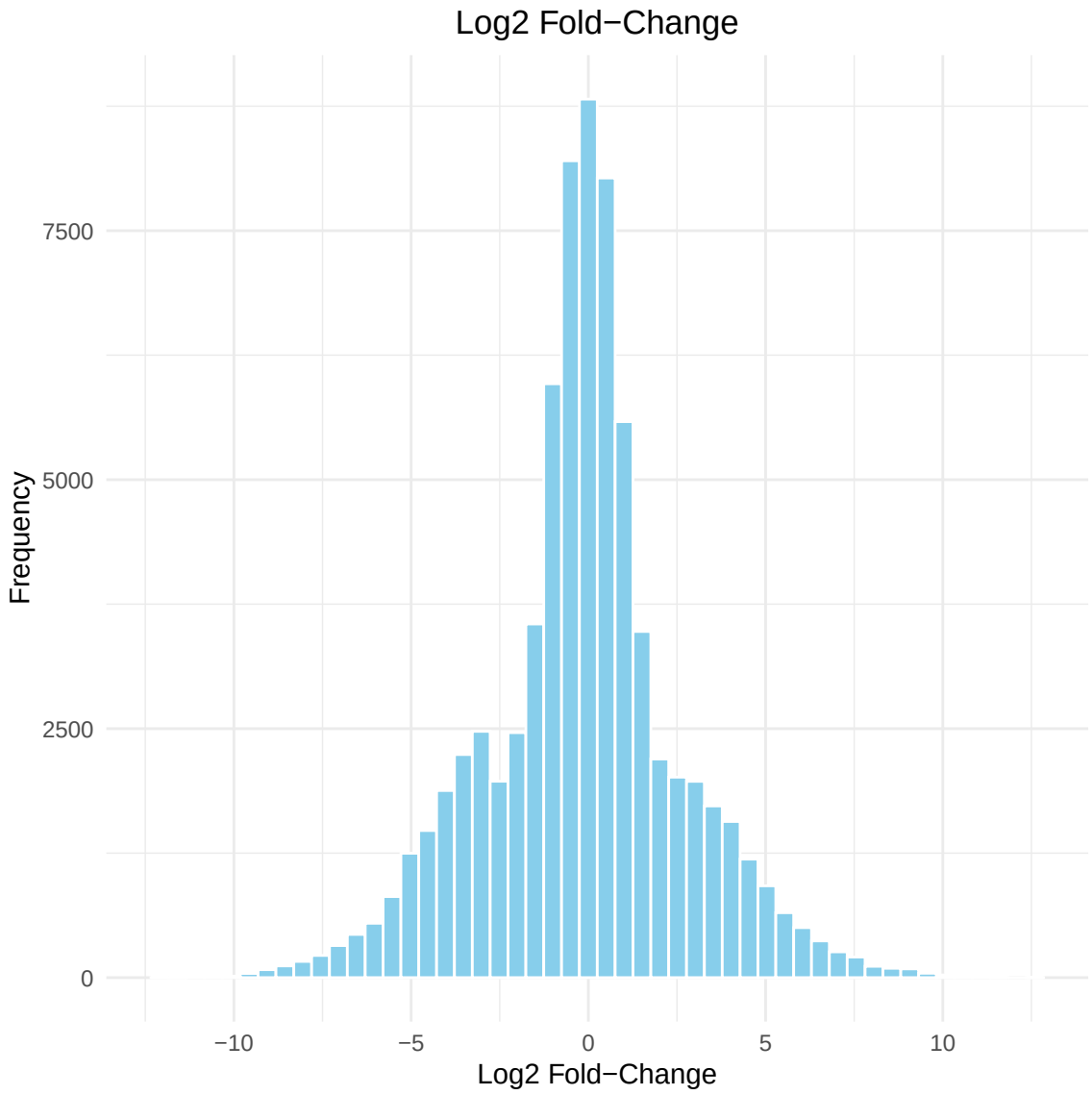


Histogram

```
ggplot(data_with_gene, aes(x = pvalue)) + geom_histogram(bins = 50,
    fill = "skyblue", color = "white") + theme_minimal() + labs(title = "P-value",
    x = "P-value", y = "Frequency") + theme(plot.title = element_text(hjust = 0.5))
```

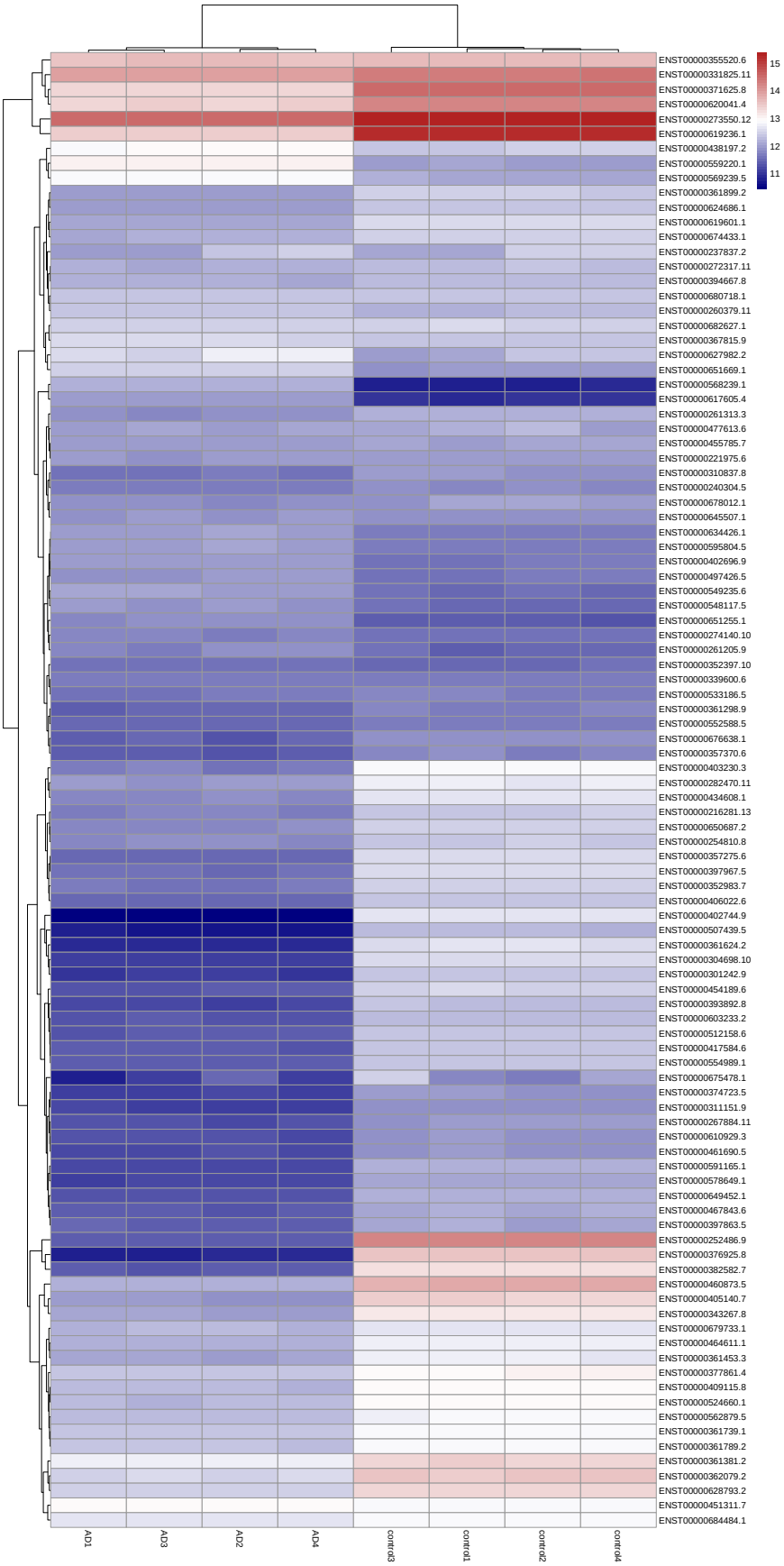



```
ggplot(data_with_gene, aes(x = log2FoldChange)) + geom_histogram(bins = 50,
    fill = "skyblue", color = "white") + theme_minimal() + labs(title = "Log2 Fold-Change",
    x = "Log2 Fold-Change", y = "Frequency") + theme(plot.title = element_text(hjust = 0.5))
```



Heatmap

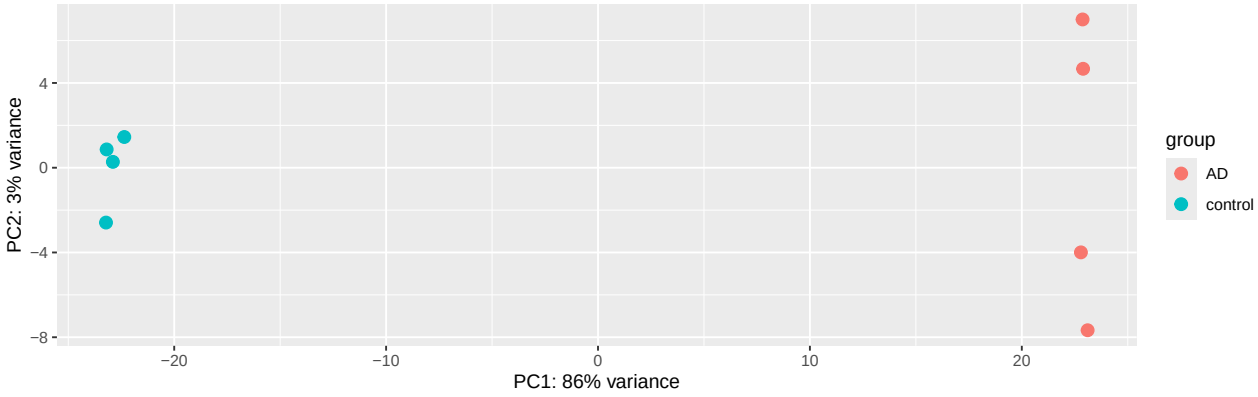
```
vsd <- vst(dds, blind = TRUE)
select_genes <- head(order(rowMeans(counts(dds, normalized = TRUE)),
    decreasing = TRUE), 100)
mat <- assay(vsd)[select_genes, ]

pheatmap(mat, cluster_rows = TRUE, show_rownames = TRUE, cluster_cols = TRUE,
    show_colnames = TRUE, color = colorRampPalette(c("navy",
        "white", "firebrick"))(50))
```


15
14
13
12
11
ENST00000355520.6
ENST00000331825.11
ENST00000371625.8
ENST00000620041.4
ENST00000273550.12
ENST00000619236.1
ENST00000438197.2
ENST00000559220.1
ENST00000569239.5
ENST00000361899.2
ENST00000624686.1
ENST00000619601.1
ENST00000674433.1
ENST00000237837.2
ENST00000272317.11
ENST00000394667.8
ENST00000680718.1
ENST00000260379.11
ENST00000682627.1
ENST00000367815.9
ENST00000627982.2
ENST00000651669.1
ENST00000568239.1
ENST00000617605.4
ENST00000261313.3
ENST00000477613.6
ENST00000455785.7
ENST00000221975.6
ENST00000310837.8
ENST00000240304.5
ENST00000678012.1
ENST00000645507.1
ENST00000634426.1
ENST00000595804.5
ENST00000402696.9
ENST00000497426.5
ENST00000549235.6
ENST00000548117.5
ENST00000651255.1
ENST00000274140.10
ENST00000261205.9
ENST00000352397.10
ENST00000339600.6
ENST00000533186.5
ENST00000361298.9
ENST00000552588.5
ENST00000676638.1
ENST00000357370.6
ENST00000403230.3
ENST00000282470.11
ENST00000434608.1
ENST00000216281.13
ENST00000650687.2
ENST00000254810.8
ENST00000357275.6
ENST00000397967.5
ENST00000352983.7
ENST00000406022.6
ENST00000402744.9
ENST00000507439.5
ENST00000361624.2
ENST00000304698.10
ENST00000301242.9
ENST00000454189.6
ENST00000393892.8
ENST00000603233.2
ENST00000512158.6
ENST00000417584.6
ENST00000554989.1
ENST00000675478.1
ENST00000374723.5
ENST00000311151.9
ENST00000267884.11
ENST00000610929.3
ENST00000461690.5
ENST00000591165.1
ENST00000578649.1
ENST00000649452.1
ENST00000467843.6
ENST00000397863.5
ENST00000252486.9
ENST00000376925.8
ENST00000382582.7
ENST00000460873.5
ENST00000405140.7
ENST00000343267.8
ENST00000679733.1
ENST00000464611.1
ENST00000361453.3
ENST00000377861.4
ENST00000409115.8
ENST00000524660.1
ENST00000562879.5
ENST00000361739.1
ENST00000361789.2
ENST00000361381.2
ENST00000362079.2
ENST00000628793.2
ENST00000451311.7
ENST00000684484.1
AD1
AD3
AD2
AD4
control3
control1
control2
control4

PCA

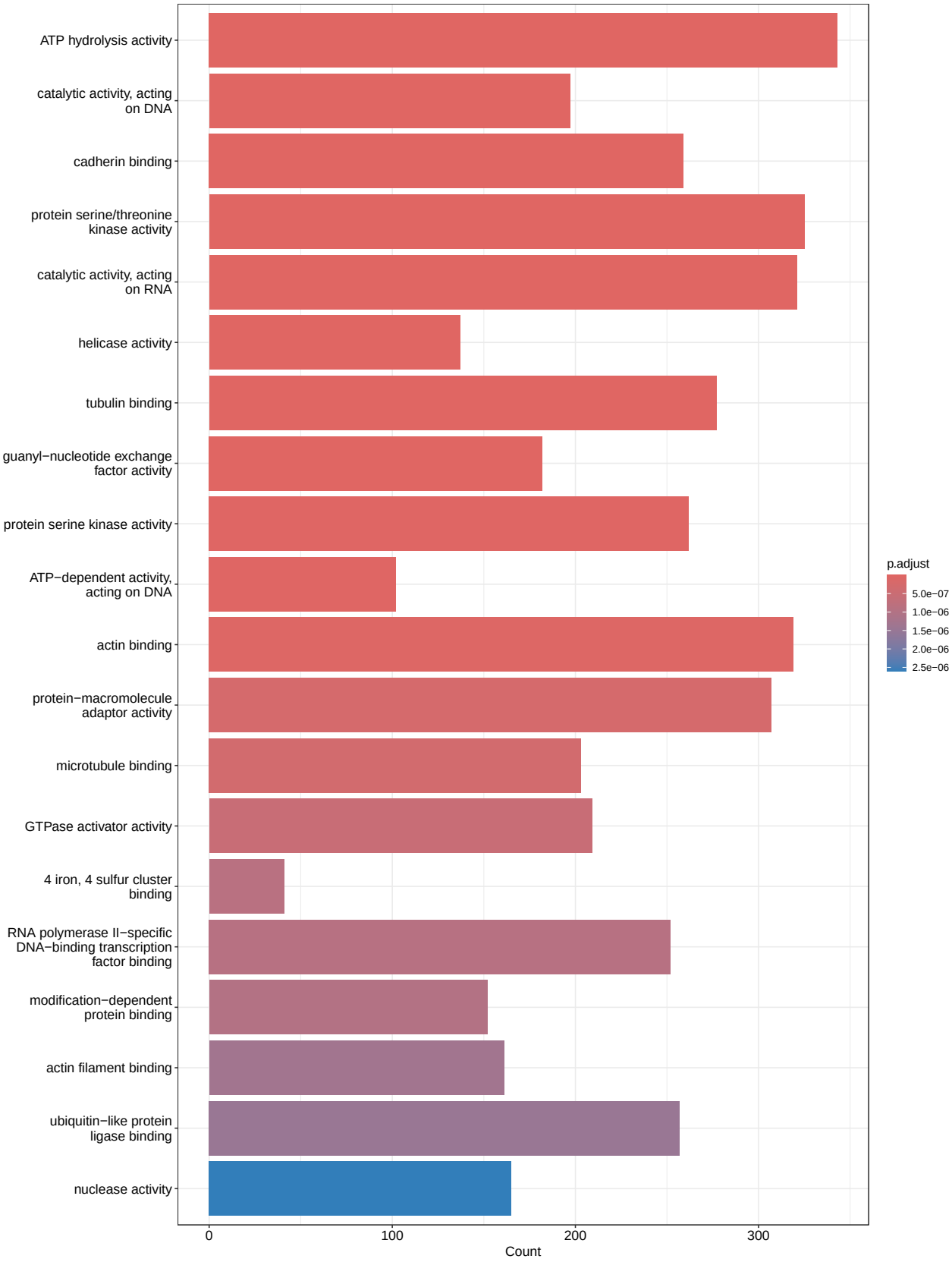
```
plotPCA(vsd, intgroup = c("condition"))
```

```
## using ntop=500 top features by variance
```

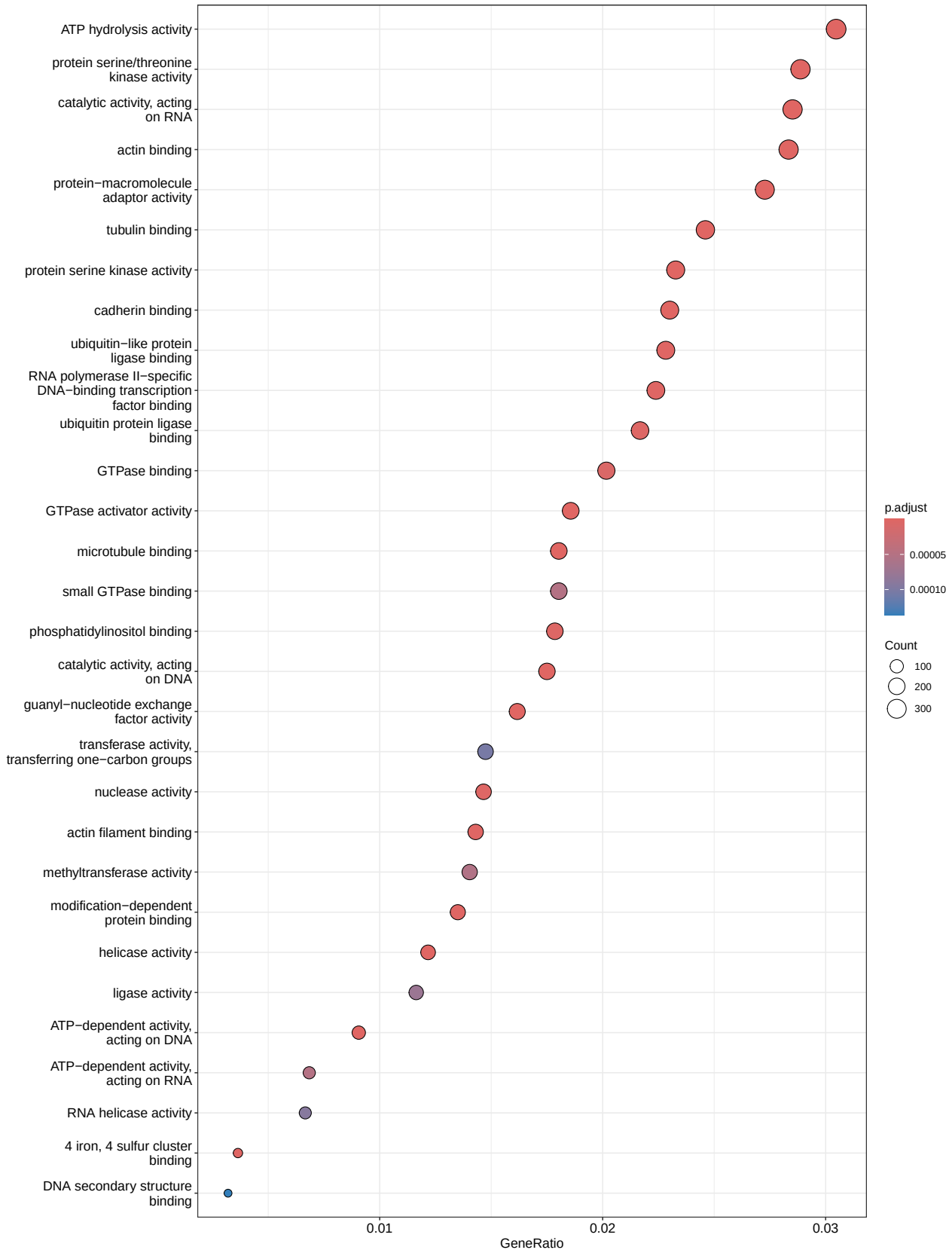


GO-Enrichement

```
sig_genes <- subset(merged_data, padj < 0.05 & abs(log2FoldChange) > 1)
sig_genes <- na.omit(sig_genes)
gene_list <- sig_genes$ensembl_gene_id

ego <- enrichGO(gene = gene_list,
                OrgDb = org.Hs.eg.db,
                keyType = "ENSEMBL", # or "SYMBOL", "ENTREZID", depending on your gene_list IDs
                ont = "MF", # "MF" for Molecular Function, "CC" for Cellular Component
                pAdjustMethod = "BH", # Benjamini & Hochberg method
                pvalueCutoff = 0.05,
                qvalueCutoff = 0.2)
barplot(ego, showCategory=20)
```


ATP hydrolysis activity
catalytic activity, acting on DNA
cadherin binding
protein serine/threonine kinase activity
catalytic activity, acting on RNA
helicase activity
tubulin binding
guanyl−nucleotide exchange factor activity
protein serine kinase activity
ATP−dependent activity, acting on DNA
actin binding
protein−macromolecule adaptor activity
microtubule binding
GTPase activator activity
4 iron, 4 sulfur cluster binding
RNA polymerase II−specific DNA−binding transcription factor binding
modification−dependent protein binding
actin filament binding
ubiquitin−like protein ligase binding
nuclease activity
0
100
200
300
Count
p.adjust
5.0e−07
1.0e−06
1.5e−06
2.0e−06
2.5e−06

```
dotplot(ego, showCategory=30)
```



```
ego_sim <- pairwise_termsim(ego)
emapplot(ego_sim)
```