Perform Differential Expression analysis using DESeq2

Loading Required Libraries

```
library(DESeq2)
library(pheatmap)
library(ggplot2)
library(tinytex)
library(magrittr)
```

Read expression matrix into CSV file.

```
read_count_data <- function(file_path){
counts_data <- read.csv(file_path, row.names = 1)
return(counts_data)
}
expression_matrix <- read_count_data("../DESeq2/Data/pasilla_gene_exp.csv")
```

Read metadata into CSV file.

```
read_metadata <- function(file_path){
coldata <- read.csv(file_path, row.names = 1)
return(coldata)
}
meta_data <- read_metadata("../DESeq2/Data/pasilla_meta.data.csv")
```

convert condition and types columns in meta_data object to factor

```
convert_chr_to_factor <- function(){
meta_data$condition <- factor(meta_data$condition)
meta_data$type <- factor(meta_data$type)
}
convert_chr_to_factor()
```

make sure the row names in meta_data matches to the column names in expression matrix

```
all(rownames(meta_data) %in% colnames(expression_matrix))
```

```
## [1] TRUE
```

IS the columns of the expression matrix and the rows of the meta_data (information about samples) are in the same order?

```
all(rownames(meta_data) == colnames(expression_matrix))
```

```
## [1] TRUE
```

if Not, make them in the same order.

```
expression_matrix <- expression_matrix[, rownames(meta_data)]
all(rownames(meta_data) == colnames(expression_matrix))
```

Pre-filtering: removing rows with low gene counts

keep rows that have at least 10 reads total

```
pre_filter <- function(){
  # Only keep rows that have total counts above the cutoff
  keep <- expression_matrix %>% rowSums(.) >= 10
  filtered_counts <- expression_matrix[keep,]
  return(filtered_counts)
}
filtered_expression_counts <- pre_filter()
head(filtered_expression_counts,2)
```

```
##             treated1 treated2 treated3 untreated1 untreated2 untreated3
## FBgn0000008      140       88       70         92        161         76
## FBgn0000014        4        0        0          5          1          0
##             untreated4
## FBgn0000008         70
## FBgn0000014          0
```

Construct a DESeqDataSet.

```
dds <- function(){
deseqdataset <- DESeqDataSetFromMatrix(countData = filtered_expression_counts,
                                       colData = meta_data,
                                       design = ~ condition)
return(deseqdataset)
}
deseqdataset <- dds()
```

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

```
deseqdataset
```

```
## class: DESeqDataSet
## dim: 9921 7
## metadata(1): version
## assays(1): counts
## rownames(9921): FBgn0000008 FBgn0000014 ... FBgn0261574 FBgn0261575
## rowData names(0):
## colnames(7): treated1 treated2 ... untreated3 untreated4
## colData names(2): condition type
```

Differential expression analysis

```
diff_expr_analysis <- function(){
deseqdataset <- DESeq(deseqdataset)
result <- results(deseqdataset, alpha = 0.01 , lfcThreshold = 1.5)
return(result)
}
deseq_result <- diff_expr_analysis()
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
deseq_result
```

```
## log2 fold change (MLE): condition untreated vs treated
## Wald test p-value: condition untreated vs treated
## DataFrame with 9921 rows and 6 columns
##               baseMean log2FoldChange     lfcSE       stat    pvalue      padj
##              <numeric>      <numeric> <numeric>  <numeric> <numeric> <numeric>
## FBgn0000008   95.14429    -0.00227611  0.223729 -0.0101735  1.000000         1
## FBgn0000014    1.05652     0.49512076  2.143184  0.2310211  0.856368         1
## FBgn0000017 4352.55357     0.23991914  0.126337  1.8990424  1.000000         1
## FBgn0000018  418.61048     0.10467410  0.148489  0.7049281  1.000000         1
## FBgn0000024    6.40620    -0.21084650  0.689588 -0.3057574  0.975772         1
## ...                ...            ...       ...        ...       ...       ...
## FBgn0261570 3208.38861     -0.2955327  0.127350 -2.3206250  1.000000         1
## FBgn0261572    6.19719      0.9588244  0.775315  1.2366908  0.758172         1
## FBgn0261573 2240.97951     -0.0127193  0.113300 -0.1122622  1.000000         1
## FBgn0261574 4857.68037     -0.0153920  0.192567 -0.0799304  1.000000         1
## FBgn0261575   10.68252     -0.1635676  0.930911 -0.1757071  0.961411         1
```

Top 10 differentail expressed genes

```
ordered_result <- deseq_result[order(deseq_result$padj, decreasing = FALSE), ]
top10 <- head(ordered_result, n=10)
top10
```

```
## log2 fold change (MLE): condition untreated vs treated
## Wald test p-value: condition untreated vs treated
## DataFrame with 10 rows and 6 columns
##              baseMean log2FoldChange     lfcSE      stat      pvalue
##             <numeric>      <numeric> <numeric> <numeric>   <numeric>
## FBgn0039155  730.5677        4.61874 0.1691240  27.30980 3.10421e-76
## FBgn0003360 4342.8321        3.17954 0.1435677  22.14663 6.47858e-32
## FBgn0039827  261.9112        4.16243 0.2325942  17.89566 1.22195e-30
## FBgn0025111 1501.4479       -2.89995 0.1273576 -22.77011 2.08216e-28
## FBgn0034736  225.8707        3.51132 0.2147628  16.34976 3.79153e-21
## FBgn0034434  114.6233        3.64248 0.2783459  13.08616 6.95482e-15
## FBgn0035085  638.2193        2.56024 0.1378126  18.57771 7.16579e-15
## FBgn0029167 3706.0240        2.19691 0.0979154  22.43684 5.49538e-13
## FBgn0085359   68.6061        4.91772 0.4949550   9.93570 2.50805e-12
## FBgn0024288   58.8495        4.58583 0.4647472   9.86737 1.57042e-11
##                    padj
##               <numeric>
## FBgn0039155 3.07938e-72
## FBgn0003360 3.21338e-28
## FBgn0039827 4.04057e-27
## FBgn0025111 5.16376e-25
## FBgn0034736 7.52239e-18
## FBgn0034434 1.01549e-11
## FBgn0035085 1.01549e-11
## FBgn0029167 6.81427e-10
## FBgn0085359 2.76443e-09
## FBgn0024288 1.55785e-08
```

Explore results

```
summary(deseq_result)
```

```
##
## out of 9921 with nonzero total read count
## adjusted p-value < 0.01
## LFC > 1.50 (up)    : 11, 0.11%
## LFC < -1.50 (down) : 7, 0.071%
## outliers [1]       : 1, 0.01%
## low counts [2]     : 0, 0%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Write results to CSV file

```
write_sig_genes <- function(out_path){
write.csv(ordered_result, file = out_path)
}
write_sig_genes("../DESeq2//Output/Significant genes.csv")
```

Visualizing the results

PCA plot

```
pca_plot <- function(){
normalized = normTransform(deseqdataset)
jpeg("../DESeq2/Output/PCA.jpeg")
p <- plotPCA(normalized, intgroup=c("condition","type"))
print(p)
dev.off()
}
pca_plot()
```

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

## pdf
##   2
```

MA-plot

```
ma_plot <- function(){
jpeg("../DESeq2/Output/MAplot.jpeg")
plotMA(deseq_result)
dev.off()
}
ma_plot()
```

```
## pdf
##   2
```

Plot counts

Here we specify the gene which had the smallest padj from the results table

```
plot_counts <- function(){
jpeg("../DESeq2/Output/plot_count.jpeg")
plotCounts(deseqdataset, gene = which.min(deseq_result$padj), intgroup = "condition")
dev.off()
}
plot_counts()
```

```
## pdf
##   2
```

Heatmap

```
heatmap <- function(){
select <- order(rowMeans(counts(deseqdataset)),
                decreasing = FALSE)[1:20]
df <- as.data.frame(colData(deseqdataset)[,c("condition","type")])
jpeg("../DESeq2/Output/Heatmap.jpeg")
pheatmap::pheatmap(assay(deseqdataset)[select,], cluster_rows = FALSE,
                   show_rownames = FALSE, cluster_cols = FALSE,
                   annotation_col = df)
dev.off()
}
heatmap()
```

```
## jpeg
##    3
```

Volcano plot

```
vplcano_plot <- function(){
result.df <- as.data.frame(deseq_result)
result.df$diffexpressed <- "NO"
result.df$diffexpressed[result.df$log2FoldChange > 1.5 &
                        result.df$padj < 0.01] <- "UP"
result.df$diffexpressed[result.df$log2FoldChange < -1.5 &
                        result.df$padj < 0.01] <- "DOWN"
jpeg("../DESeq2/Output/Volcano.jpeg")
g <- ggplot(data = result.df, aes(x = log2FoldChange,
                                  y = -log10(pvalue),
                                  col = diffexpressed))+
  geom_point()+
  theme_minimal()+
  geom_vline(xintercept = c(-1.5, 1.5), col = "black", linetype = 'dashed') +
  geom_hline(yintercept = -log10(0.01), col = "black", linetype = 'dashed') +
  scale_color_manual(values = c("#00AFBB", "grey", "#FFDB6D"),
  labels = c("Downregulated", "Not significant", "Upregulated"))
print(g)
dev.off()
}
vplcano_plot()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (‘geom_point()‘).
```

```
## pdf
##   2
```