Final Project

Dhairya Aggarwal (51529386)

2024-12-08

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 4.3.3
```

```
library(DESeq2)
```

```
## Warning: package 'DESeq2' was built under R version 4.3.3

## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## 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, saveRDS, setdiff,
##     sort, table, tapply, union, unique, unsplit, which.max, which.min

##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Warning: package 'GenomeInfoDb' was built under R version 4.3.3

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

## Warning: package 'matrixStats' was built under R version 4.3.3

##
## 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(clusterProfiler)
```

```
## Warning: package 'clusterProfiler' was built under R version 4.3.3

##

## clusterProfiler v4.10.1  For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
## If you use clusterProfiler in published research, please cite:
## T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu, X Bo, and G Yu.

##
## Attaching package: 'clusterProfiler'

## The following object is masked from 'package:IRanges':
##
##     slice

## The following object is masked from 'package:S4Vectors':
##
##     rename

## The following object is masked from 'package:stats':
##
##     filter
```

```
library(org.Hs.eg.db)
```

```
## Loading required package: AnnotationDbi
```

```
##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:clusterProfiler':
##
##     select

##
```

```
library(survival)
```

```
## Warning: package 'survival' was built under R version 4.3.3
```

```
library(survminer)
```

```
## Warning: package 'survminer' was built under R version 4.3.3

## Loading required package: ggpubr

## Warning: package 'ggpubr' was built under R version 4.3.3

##
## Attaching package: 'survminer'

## The following object is masked from 'package:survival':
##
##     myeloma
```

```
raw.clinical.patients <- read.table("data_clinical_patient.txt", sep = "\t",
                                    header = TRUE)
raw.data.mutations <- read.table("data_mutations.txt", sep = "\t",
                                 header = TRUE)
raw.data.RNAseq <- read.csv("RNAseq_BRCA.csv", row.names=1)

#Filter data where you only have 0 or 1 read count across all samples.
raw.data.RNAseq <- raw.data.RNAseq[rowSums(raw.data.RNAseq)>1,]
```

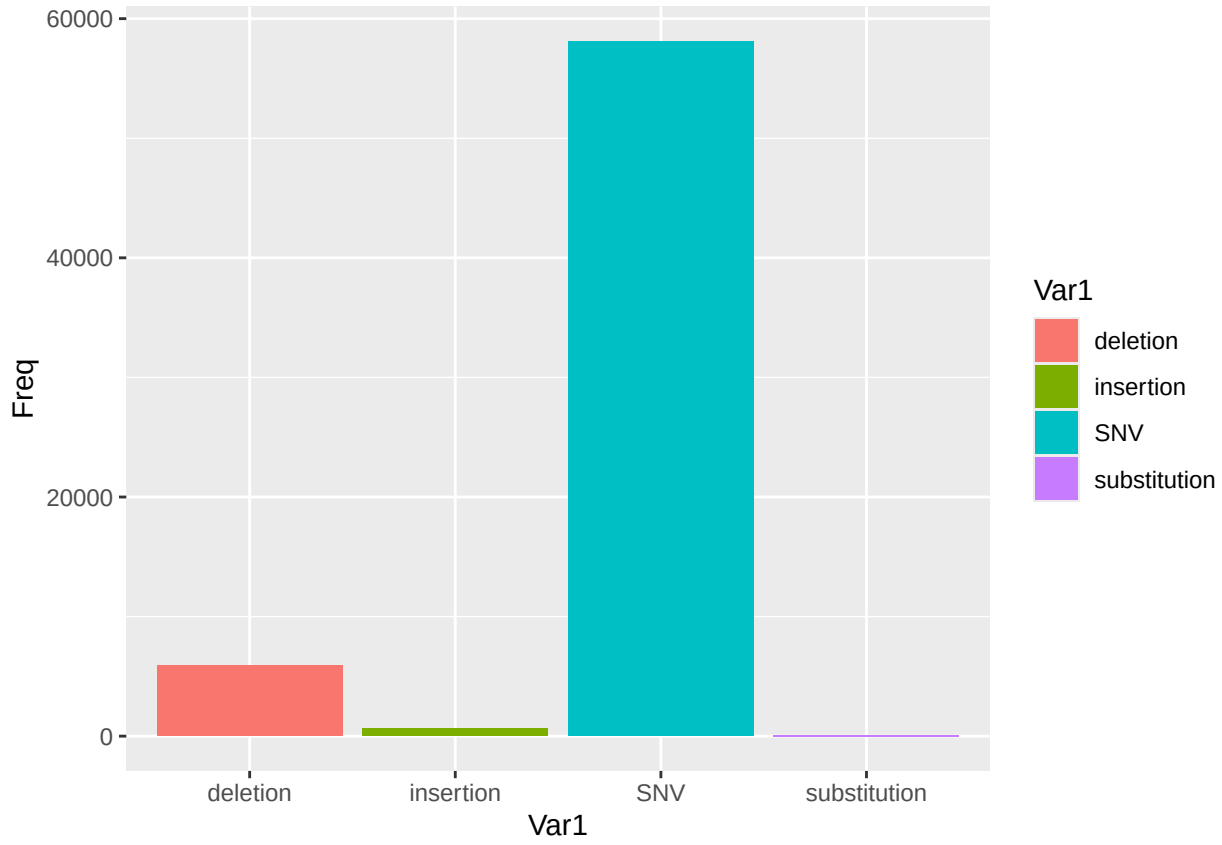
```
colnames(raw.data.RNAseq) <- make.unique(sapply(colnames(raw.data.RNAseq), function(name) {
  segments <- strsplit(name, "\\.")[[1]]
  paste(segments[1:3], collapse = "-")
}))
```

```
#Unique Patients in each data set
unique.clinical <- as.data.frame(unique(raw.clinical.patients$PATIENT_ID))
unique.mutations <- as.data.frame(unique
                                  (raw.data.mutations$Tumor_Sample_Barcode))
unique.RNA <- as.data.frame(colnames(raw.data.RNAseq[,1:length(raw.data.RNAseq)]))

#Addition patient ID's to Mutation data
mutation.patients <- as.data.frame(raw.data.mutations$Tumor_Sample_Barcode)
```

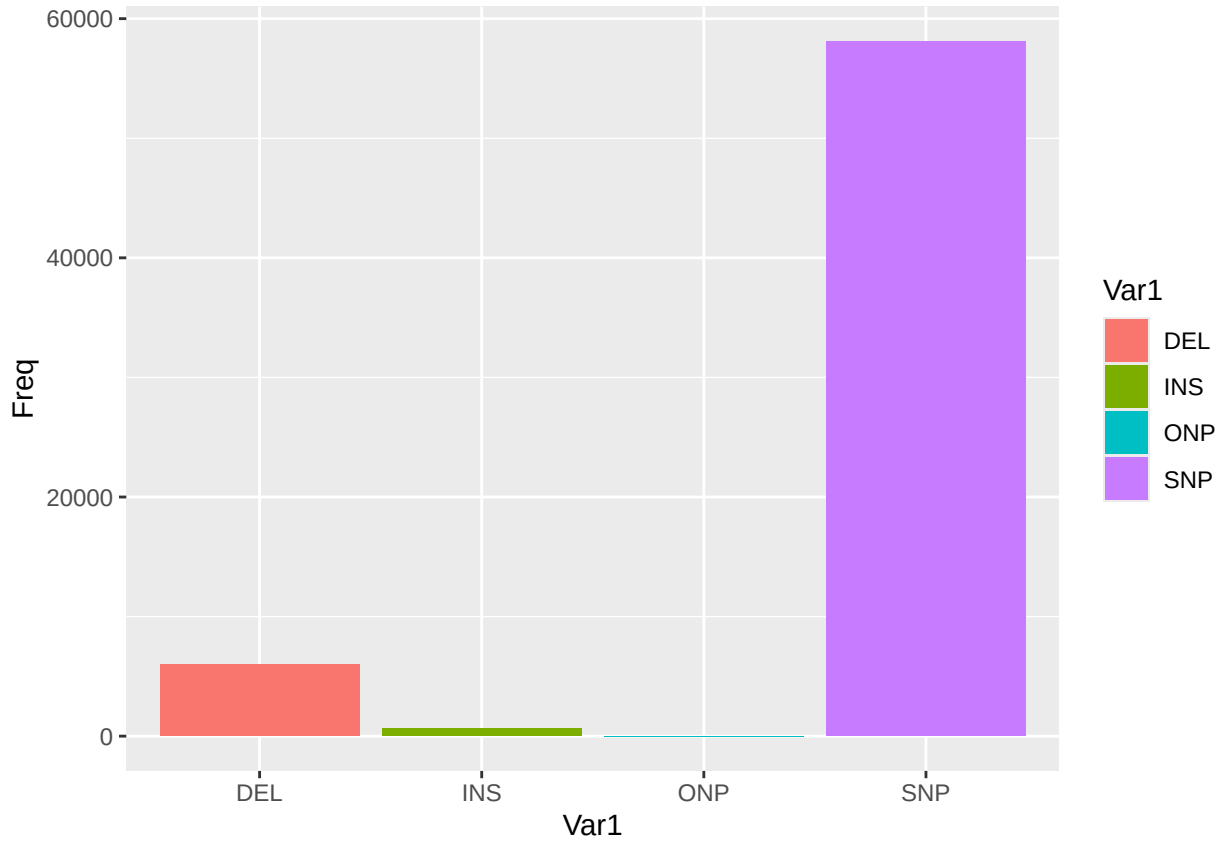
```r
colnames(mutation.patients) <- "Patient_ID"
mutation.patients$Patient_ID <- substr(mutation.patients$Patient_ID, 1, 12)
raw.data.mutations <- cbind(mutation.patients, raw.data.mutations)

colnames(unique.clinical) <- "Patient_ID"
colnames(unique.mutations) <- "Patient_ID"
colnames(unique.RNA) <- "Patient_ID"
unique.mutations$Patient_ID <- substr(unique.mutations$Patient_ID, 1, 12)

#Finding common patients
common_patient_ids <- Reduce(intersect, list(
  unique.clinical$Patient_ID,
  unique.mutations$Patient_ID,
  unique.RNA$Patient_ID
))

#3 data sets with all 975 common patients
clinical.data <- raw.clinical.patients[raw.clinical.patients$PATIENT_ID
                                       %in% common_patient_ids, ]
mutation.data <- raw.data.mutations[raw.data.mutations$Patient_ID
                                    %in% common_patient_ids, ]
seq.data <- raw.data.RNAseq[,names(raw.data.RNAseq)
                            %in% clinical.data$PATIENT_ID]
```

```r
hugo <- as.data.frame(table(mutation.data$Hugo_Symbol))
var.class <- as.data.frame(table(mutation.data$Variant_Classification))
```

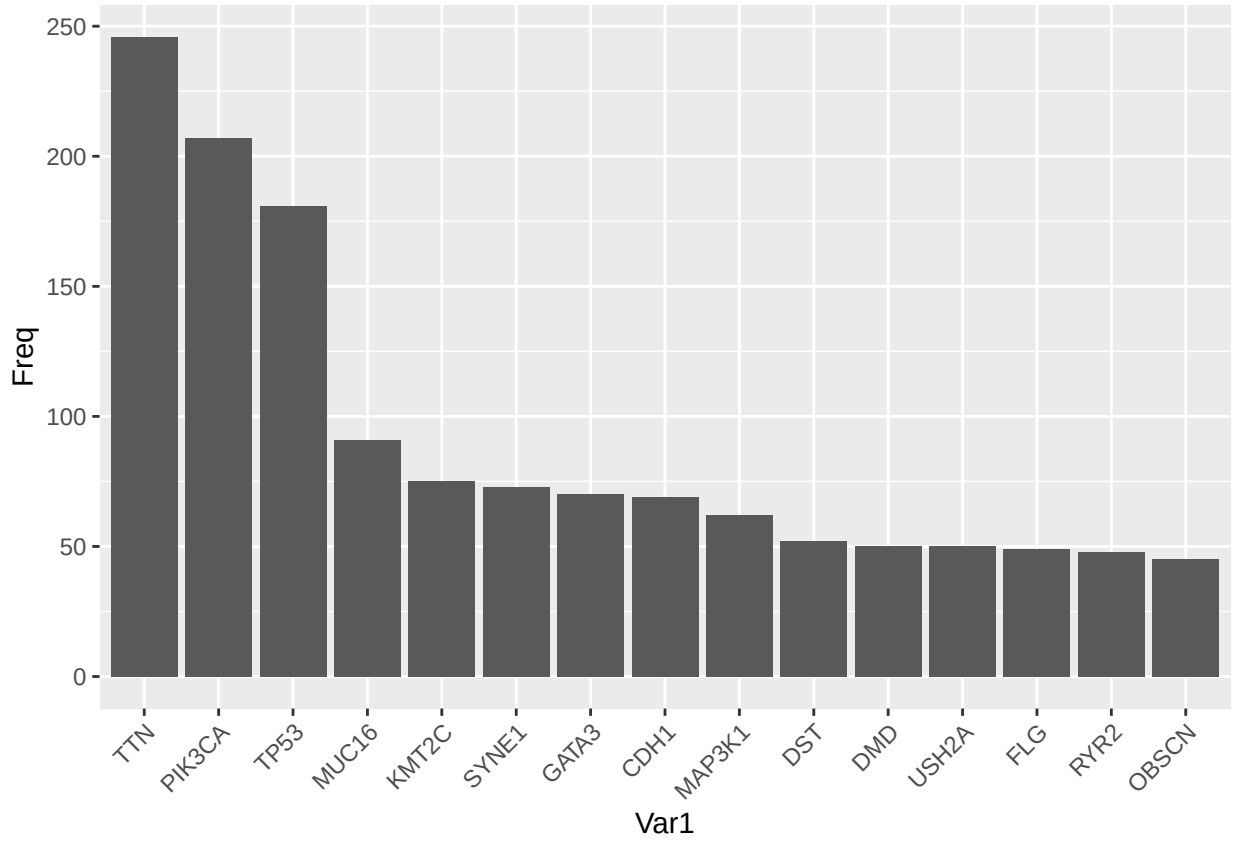
```r
var.class2 <- as.data.frame(table(mutation.data$VARIANT_CLASS))
ggplot(data=var.class2, aes(x=Var1, y=Freq))+
  geom_col(aes(fill=Var1))
```




```
var.type <- as.data.frame(table(mutation.data$Variant_Type))
ggplot(data=var.type, aes(x=Var1, y=Freq))+
  geom_col( aes(fill=Var1))
```




```
sample.name <- as.data.frame(table(mutation.data$Tumor_Sample_Barcode))
hugo <- as.data.frame(table(mutation.data$Hugo_Symbol))

hugo.ordered <- hugo[order(-hugo$Freq),]
ggplot(data=hugo.ordered[1:15,], aes(x=Var1, y=Freq))+
  geom_col()+
  theme(axis.text.x = element_text(angle = 45,hjust=1))+
  scale_x_discrete(limits = hugo.ordered[1:15,]$Var1)
```




```
cnv_events = unique(mutation.data$Variant_Classification)
oncomat = reshape2::dcast(
  data = mutation.data,
  formula = Hugo_Symbol ~ Tumor_Sample_Barcode,
  fun.aggregate = function(x, cnv = cnv_events) {
    x = as.character(x)
    xad = x[x %in% cnv]
    xvc = x[!x %in% cnv]

    if (length(xvc) > 0) {
      xvc = ifelse(test = length(xvc) > 1,
                   yes = 'Multi_Hit',
                   no = xvc)
    }

    x = ifelse(
      test = length(xad) > 0,
      yes = paste(xad, xvc, sep = ';'),
      no = xvc
    )
    x = gsub(pattern = ';$',
             replacement = '',
             x = x)
    x = gsub(pattern = '^;',
             replacement = '',
             x = x)
```

```
    return(x)
  },
  value.var = 'Variant_Classification',
  fill = '',
  drop = FALSE
)

#Code adopted from Tutorial 4 "Introduction to Mutation Analysis".
```

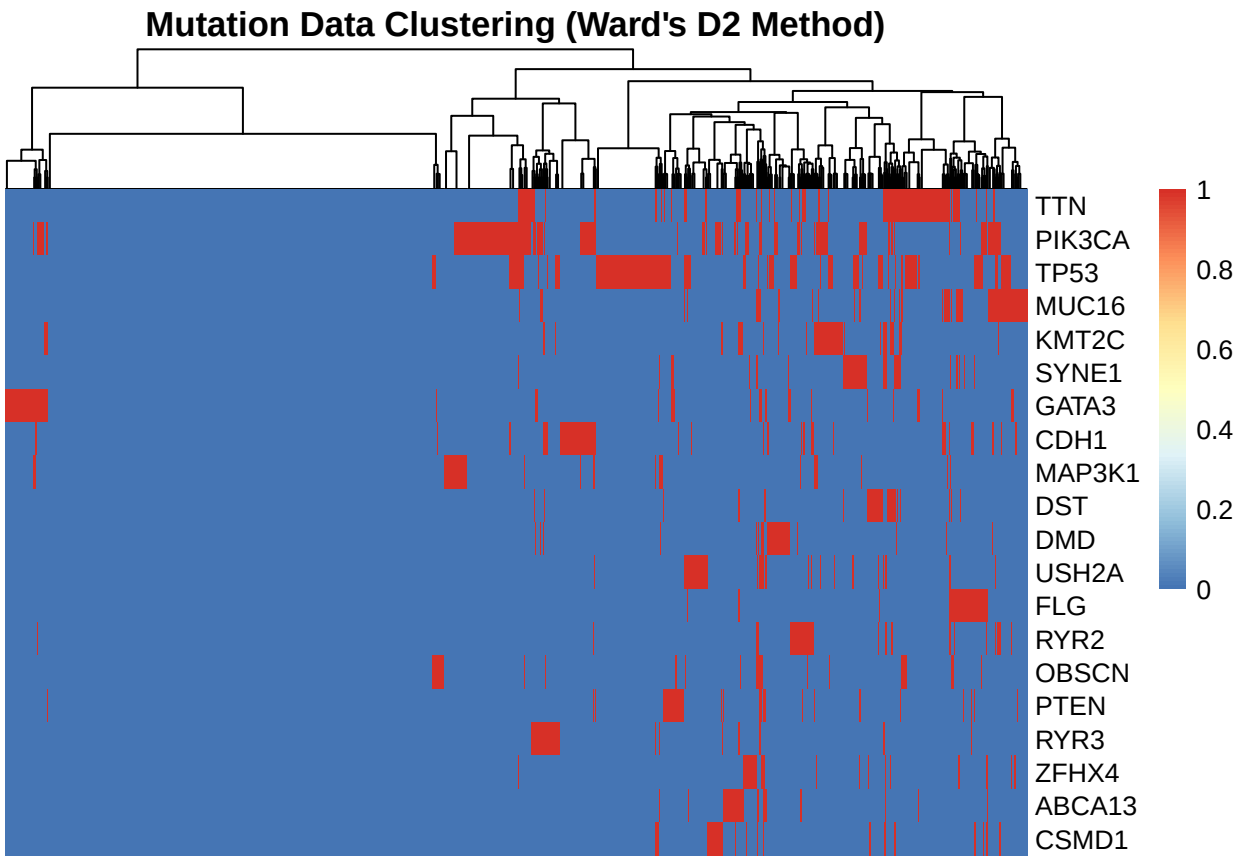
```
rownames(oncomat) = oncomat$Hugo_Symbol
oncomat <- oncomat[,-1]
```

```
oncomat.ordered <- oncomat[order(-hugo$Freq),]
```

```
mat <- oncomat.ordered
mat[mat!=""]=1
mat[mat==""]=0
mat <- apply(mat, 2 ,as.numeric)
mat <- as.matrix(mat)
rownames(mat)  <-  row.names(oncomat.ordered)
```

```
reduce.mat <- mat[1:20,]
```

```
mutation_dist <- dist(t(reduce.mat))

mutation_hclust <- hclust(mutation_dist, method = "ward.D2")

pheatmap(
  reduce.mat,
  cluster_rows = FALSE,
  cluster_cols = mutation_hclust,
  show_colnames = FALSE,
  main = "Mutation Data Clustering (Ward's D2 Method)"
)
```




```
patient_clusters <- cutree(mutation_hclust, k = 2)

cluster_assignments <- data.frame(
  Sample = colnames(reduce.mat),
  Cluster = as.factor(patient_clusters)
)

print(table(cluster_assignments$Cluster))
```

```
##
##   1   2
## 419 556
```

```
colnames(reduce.mat) <- sub("-\\d+$", "", colnames(reduce.mat))

# assign clusters to clinical data
clinical.data$Cluster <- patient_clusters[match(clinical.data$PATIENT_ID, colnames(reduce.mat))]
```

```
matched_patients <- clinical.data$PATIENT_ID[clinical.data$PATIENT_ID %in% colnames(seq.data)]

expression.data.filtered <- seq.data[, matched_patients]

clusters <- clinical.data$Cluster[match(matched_patients, clinical.data$PATIENT_ID)]
```

```
dds <- DESeqDataSetFromMatrix(
  countData = as.matrix(expression.data.filtered),
  colData = data.frame(Cluster = clusters),
  design = ~ Cluster
)
```

```
##   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
```

```
dds <- DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 12335 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

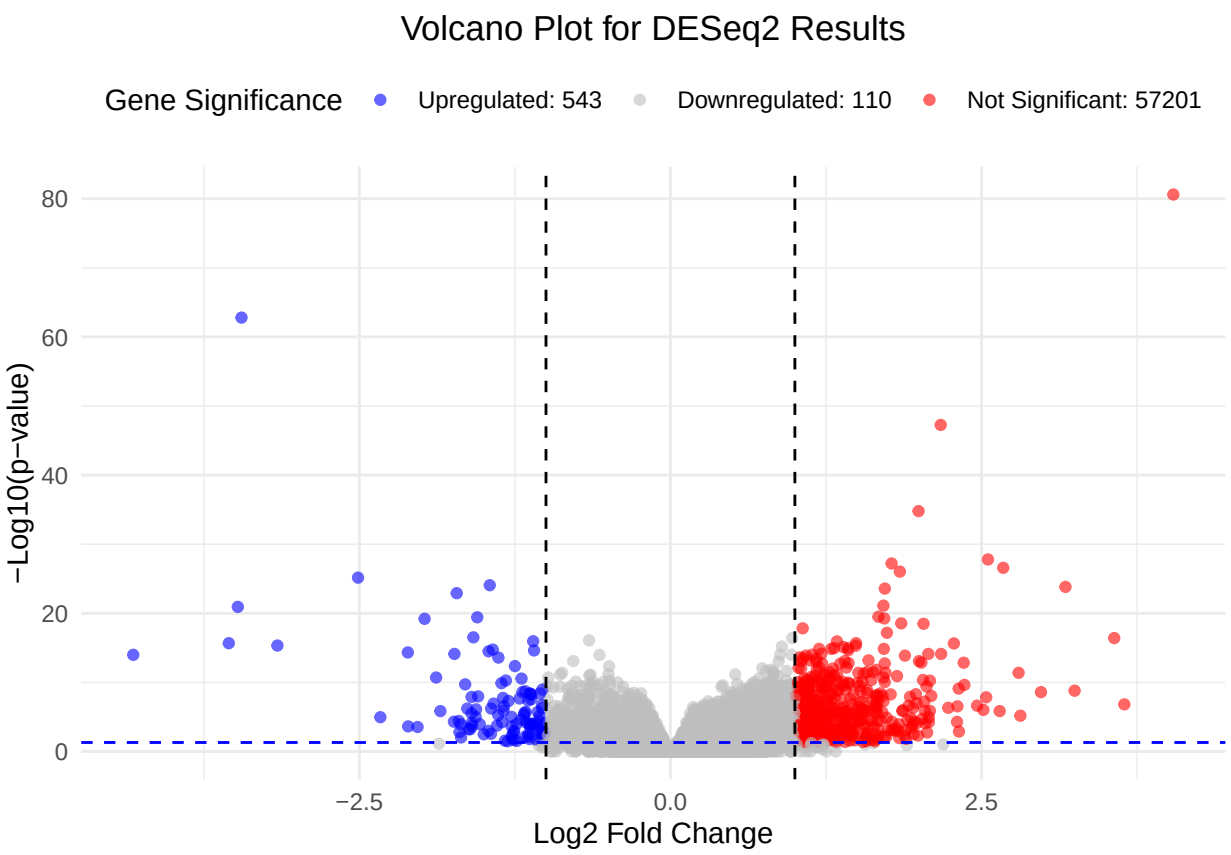
```
de_results <- results(dds, alpha = 0.05)

sig_genes <- de_results[!is.na(de_results$padj) & de_results$padj < 0.05, ]
sig_gene_list <- rownames(sig_genes)
head(sig_genes)
```

```
## log2 fold change (MLE): Cluster
## Wald test p-value: Cluster
## DataFrame with 6 rows and 6 columns
##                      baseMean log2FoldChange     lfcSE      stat      pvalue
##                     <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000000005.6     73.8716       0.443881 0.1613514   2.75102 5.94103e-03
## ENSG00000000971.16  3306.4337       0.176021 0.0702326   2.50626 1.22015e-02
## ENSG00000001617.12  6224.6316      -0.206319 0.0559788  -3.68567 2.28102e-04
## ENSG00000001626.16    45.9049       0.612524 0.1310456   4.67413 2.95205e-06
## ENSG00000002079.14    14.4727       0.233275 0.0907763   2.56978 1.01763e-02
```

```
## ENSG00000002586.20 11030.5343       0.188390 0.0537037   3.50796 4.51565e-04
##                           padj
##                      <numeric>
## ENSG00000000005.6  2.80307e-02
## ENSG00000000971.16 4.81250e-02
## ENSG00000001617.12 2.33606e-03
## ENSG00000001626.16 7.55345e-05
## ENSG00000002079.14 4.19578e-02
## ENSG00000002586.20 3.92543e-03
```

```
volcano_data <- as.data.frame(de_results)
volcano_data$Gene <- rownames(volcano_data)

volcano_data$padj[is.na(volcano_data$padj)] <- 1
volcano_data$log2FoldChange[is.na(volcano_data$log2FoldChange)] <- 0

padj_threshold <- 0.05
log2Fold_threshold <- 1


volcano_data$Significance <- ifelse(
  volcano_data$log2FoldChange >= log2Fold_threshold & volcano_data$padj <= padj_threshold, "Upregulated
  ifelse(volcano_data$log2FoldChange <= -log2Fold_threshold & volcano_data$padj <= padj_threshold, "Dow
)

up_genes <- sum(volcano_data$Significance == "Upregulated")
down_genes <- sum(volcano_data$Significance == "Downregulated")
not_significant_genes <- sum(volcano_data$Significance == "Not Significant")

legend_labels <- c(
  paste0("Upregulated: ", up_genes),
  paste0("Downregulated: ", down_genes),
  paste0("Not Significant: ", not_significant_genes)
)

# Define plot colors
plot_colors <- c(
  "Upregulated" = "red",
  "Downregulated" = "blue",
  "Not Significant" = "gray"
)

ggplot(volcano_data, aes(x = log2FoldChange, y = -log10(padj))) +
  geom_point(aes(color = Significance), alpha = 0.6, size = 1.5) +
  scale_color_manual(values = plot_colors, labels = legend_labels) +
  theme_minimal() +
  labs(
    title = "Volcano Plot for DESeq2 Results",
    x = "Log2 Fold Change",
    y = "-Log10(p-value)",
    color = "Gene Significance"
  ) +
  theme(
    legend.position = "top",
```

```
    plot.title = element_text(hjust = 0.5)
  ) +
  geom_hline(yintercept = -log10(padj_threshold), linetype = "dashed", color = "blue") +
  geom_vline(xintercept = c(-log2Fold_threshold, log2Fold_threshold), linetype = "dashed", color = "bla
```




```
rownames(de_results)  <- gsub("\\..*", "", rownames(de_results))
head(rownames(de_results))
```

```
## [1] "ENSG00000000003" "ENSG00000000005" "ENSG00000000419" "ENSG00000000457"
## [5] "ENSG00000000460" "ENSG00000000938"
```

```
converted_genes <- bitr(
  rownames(de_results),
  fromType = "ENSEMBL",
  toType = "ENTREZID",
  OrgDb = org.Hs.eg.db
)
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
## Warning in bitr(rownames(de_results), fromType = "ENSEMBL", toType =
## "ENTREZID", : 39.52% of input gene IDs are fail to map...
```

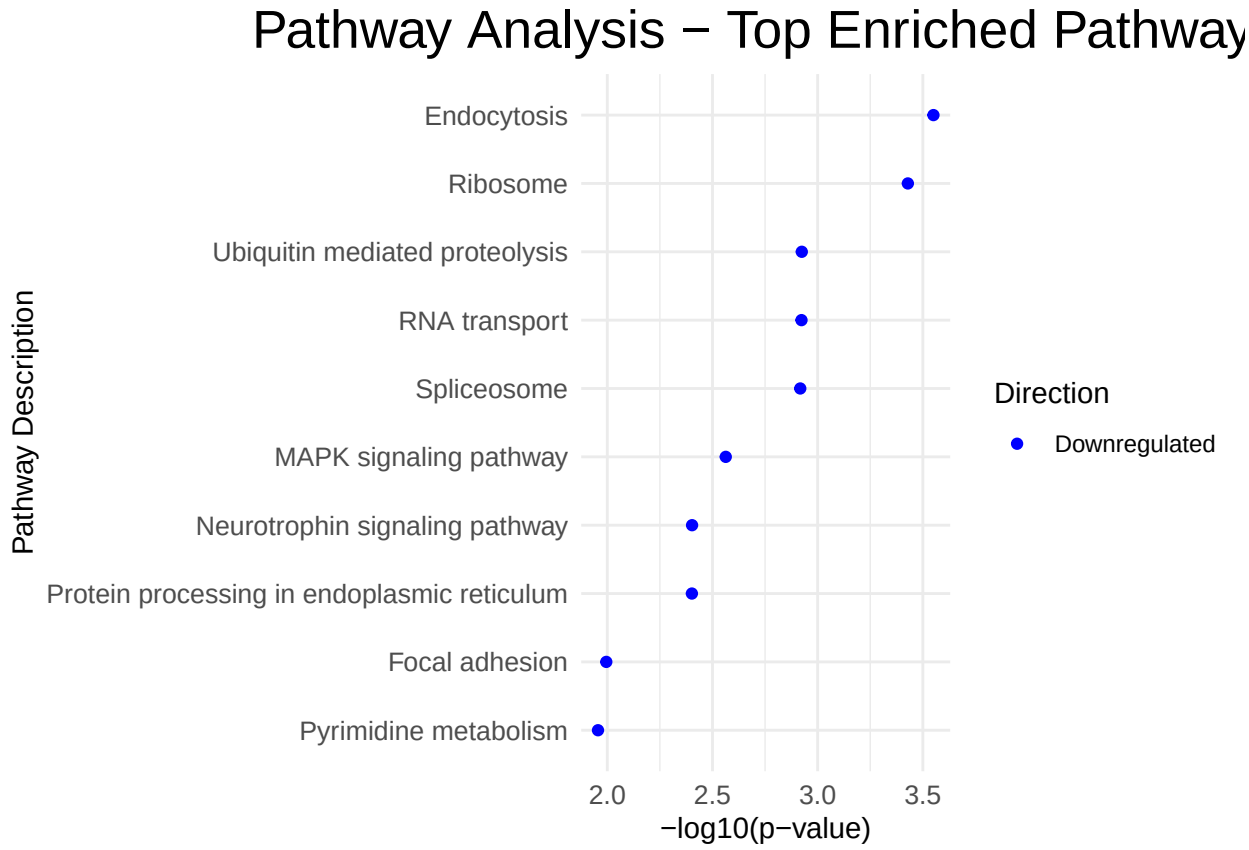
```r
converted_genes <- na.omit(converted_genes)

foldchanges <- de_results$log2FoldChange
names(foldchanges) <- converted_genes$ENTREZID[match(rownames(de_results), converted_genes$ENSEMBL)]

foldchanges <- na.omit(foldchanges)
```

```r
library(gage)
library(gageData)

data(kegg.sets.hs)
data(sigmet.idx.hs)

kegg.sets.filtered <- kegg.sets.hs[sigmet.idx.hs]

keggres <- gage(
  foldchanges,
  gsets = kegg.sets.filtered,
  same.dir = TRUE
)
```

```r
upregulated_ids <- substr(rownames(keggres$greater)[1:5], 1, 8)
downregulated_ids <- substr(rownames(keggres$less)[1:5], 1, 8)
```

```r
keggres_summary <- rbind(
  data.frame(Pathway = rownames(keggres$greater), keggres$greater, Direction = "Upregulated"),
  data.frame(Pathway = rownames(keggres$less), keggres$less, Direction = "Downregulated")
)

keggres_summary <- keggres_summary[!is.na(keggres_summary$p.val) & keggres_summary$p.val < 0.05, ]

keggres_summary$Pathway_ID <- substr(keggres_summary$Pathway, 1, 8)
keggres_summary$Description <- substr(keggres_summary$Pathway, 10, nchar(keggres_summary$Pathway))

keggres_summary <- keggres_summary[order(keggres_summary$p.val), ]

# Select top pathways for plotting (top 10)
plot_data <- keggres_summary[1:10, ]

ggplot(plot_data, aes(x = -log10(p.val), y = reorder(Description, -log10(p.val)), color = Direction)) +
  geom_point() +
  scale_color_manual(
    values = c("Upregulated" = "red", "Downregulated" = "blue"),
    name = "Direction"
  ) +
  theme_minimal() +
  labs(
    title = "Pathway Analysis - Top Enriched Pathways",
    x = "-log10(p-value)",
    y = "Pathway Description"
  ) +
  theme(
    axis.text.y = element_text(size = 10),
```

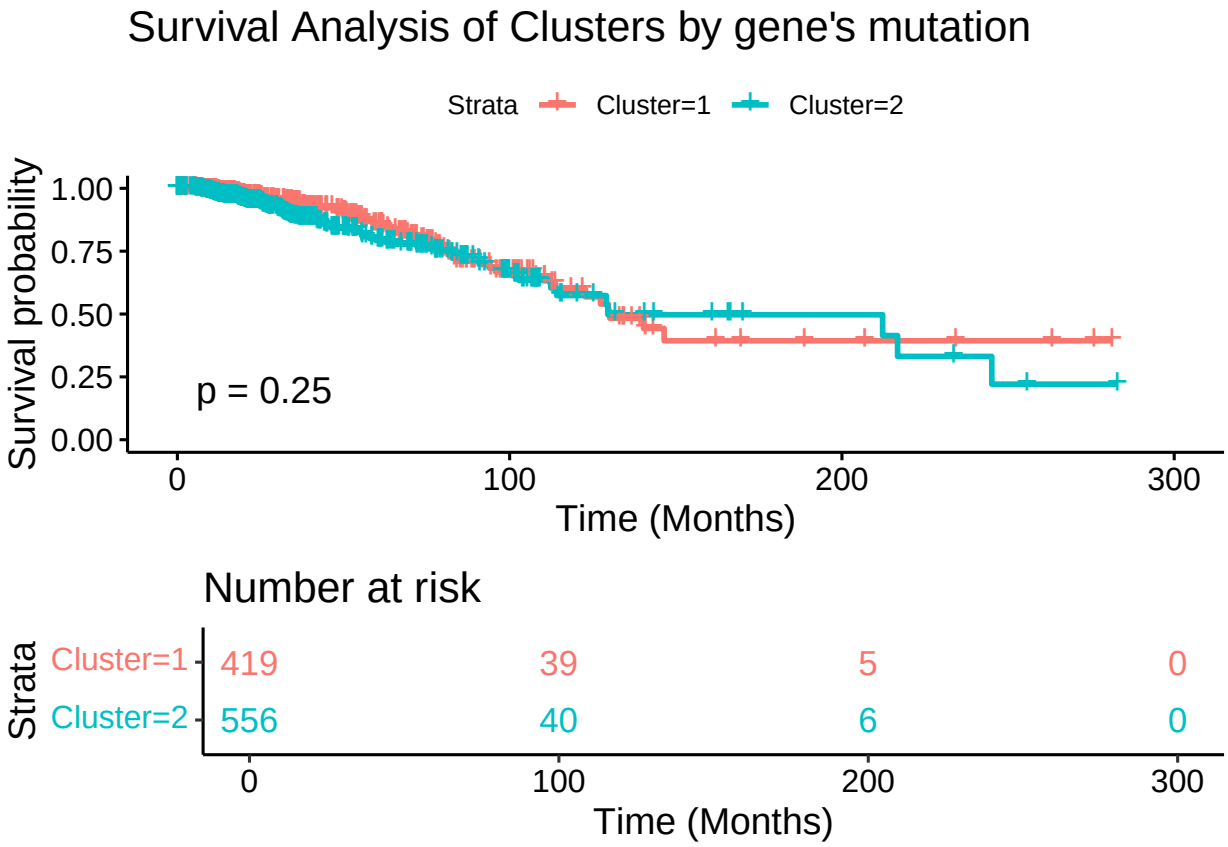
```
    axis.text.x = element_text(size = 10),
    plot.title = element_text(hjust = 0.5, size = 20)
  )
```




```
clinical.data$deceased = clinical.data$OS_STATUS == "1:DECEASED"

Surv(clinical.data$OS_MONTHS, clinical.data$deceased) ~ Cluster
```

```
## Surv(clinical.data$OS_MONTHS, clinical.data$deceased) ~ Cluster
```

```
fit_new = survfit(Surv(OS_MONTHS, deceased) ~ Cluster, data=clinical.data)
ggsurvplot(fit_new, data=clinical.data, pval=T, risk.table=T, risk.table.col="strata", risk.table.heigh
```


Survival Analysis of Clusters by gene's mutation
Strata
Cluster=1
Cluster=2
Survival probability
1.00
0.75
0.50
0.25
0.00
p = 0.25
0
100
200
300
Time (Months)

Number at risk
Strata
Cluster=1
Cluster=2
419
39
5
0
556
40
6
0
0
100
200
300
Time (Months)