Mutation Data Clustering and Analysis of Mutation Clusters

BMEG 310 Final Report Code

Laura Ing (55616957), Benjamin Green (99276917), Aadesh Mehra (39288733)

```r
library(tidyverse)
library(ggplot2)
library(ggbiplot)
library(RColorBrewer)
library("survival")
library("survminer")
library(DESeq2)
library("AnnotationDbi")
library("org.Hs.eg.db")
library(pathview)
library(gage)
library(gageData)
```

```r
# Load Data
data.clinical <- read.delim("data/data_clinical_patient.txt",
                            sep = "\t", header = TRUE, comment.char = "#")
data.mutation <- read.delim("data/data_mutations.txt",
                            sep = "\t", header = TRUE, comment.char = "#")
data.expression <- read.delim("data/RNAseq_BRCA.csv",
                              sep = ",", header = TRUE, comment.char = "#")
```

```r
# ------------- GET LIST OF ALL PATIENTS WITH FULL DATA -------------

# Build the patient-mutation matrix
mutation.patients <- unique(substr(data.mutation$Tumor_Sample_Barcode, 1, 12))
clinical.patients <- data.clinical$PATIENT_ID
expression.patients <- gsub("\\.", "-",
                            substr(colnames(data.expression), 1, 12)[-1])

# Get the patients that we have full data for
unique.patients.full.data <- Reduce(intersect,
                                    list(mutation.patients,
                                         clinical.patients,
                                         expression.patients))
```

```r
# ------------- CLEAN MUTATION DATA -------------

# Make a new column with the cleaned patient ID (same as other datasets)
data.mutation$Tumor_Sample_Barcode_Cleaned <- substr(
  data.mutation$Tumor_Sample_Barcode, 1, 12)

# Filter out low impact and modifier mutations
important_mutations <- data.mutation[which(data.mutation$IMPACT
                                           %in% c("HIGH", "MODERATE")), ]

# Get the important mutations for full-data patients
important_mutations_full_data <- important_mutations[which(
  important_mutations$Tumor_Sample_Barcode_Cleaned
  %in% unique.patients.full.data), ]

# Make feature matrix of mutated genes for each patient
feature_mat <- table(important_mutations_full_data$Tumor_Sample_Barcode_Cleaned, important_mutations_ful

# Turn into binary matrix where 1 indicates mutation and 0 indicates no mutation
feature_mat[feature_mat > 1] <- 1
```

```
# ------------- REDUCE DIMENSIONS OF MUTATION DATA -------------

# Filter for the top ~20 mutated genes
quantile(colSums(feature_mat), probs = 0.999)
```

```
##  99.9%
## 51.494
```

Therefore, any genes with 51 or more mutations are in the 99.9th percentile.

```
feature_mat_filtered <- feature_mat[, colSums(feature_mat) >= 51]
str(feature_mat_filtered)
```

```
##  'table' num [1:1006, 1:17] 0 0 0 0 1 1 0 0 0 1 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:1006] "TCGA-3C-AAAU" "TCGA-3C-AALI" "TCGA-3C-AALJ" "TCGA-3C-AALK" ...
##   ..$ : chr [1:17] "CDH1" "FLG" "GATA3" "HMCN1" ...
```

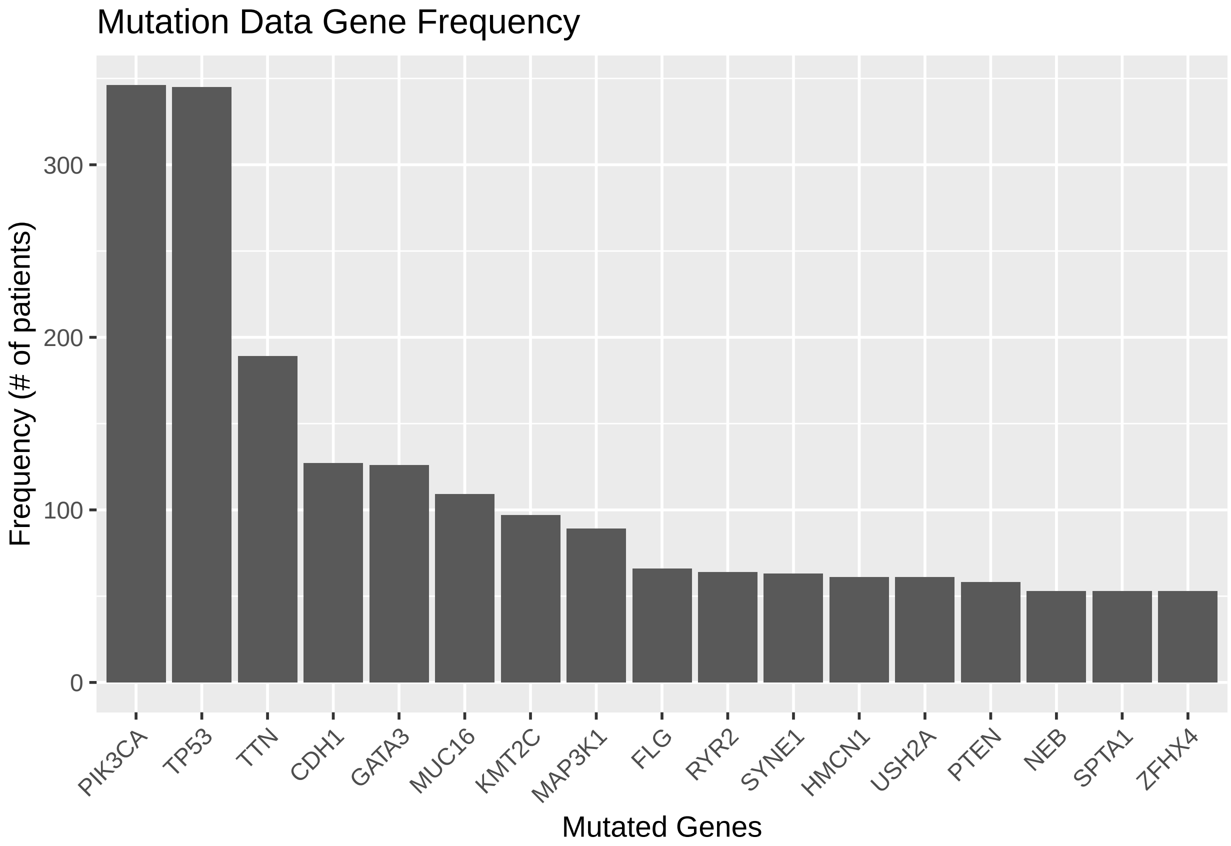
The feature matrix before filtering had 16503 genes. This matrix has only the most mutated genes (>51 patients) and only has 17 genes.

```
# ------------- CLEAN THE MUTATION FEATURE MATRIX -------------

# Convert feature matrix to data frame
feature_df_filtered <- as.data.frame(feature_mat_filtered)
feature_df_filtered <- pivot_wider(feature_df_filtered, names_from = Var2, values_from = Freq)

# Save the patients column
patient.order <- feature_df_filtered$Var1

# Remove the patients column
feature_df_filtered <- feature_df_filtered[, -1]
```

We will now visualize the most mutated genes using a bar plot.

```
# ------------- VISUALIZE MOST MUTATED GENES -------------
# Create frequency table
freq <- colSums(feature_df_filtered == 1)

# Convert frequency table to a data frame
data <- data.frame(column = names(freq), frequency = freq)
```

```
# Sort the data frame by frequency in descending order
data <- data[order(data$frequency, decreasing = TRUE), ]

# Create a bar plot to visualize the most mutated genes
ggplot(data, aes(x = column, y = frequency)) +
  geom_col() + theme(axis.text.x = element_text(angle = 45,hjust=1)) +
  scale_x_discrete(limits = data$column) +
  labs(x = "Mutated Genes", y = "Frequency (# of patients)",
       title = "Mutation Data Gene Frequency")
```

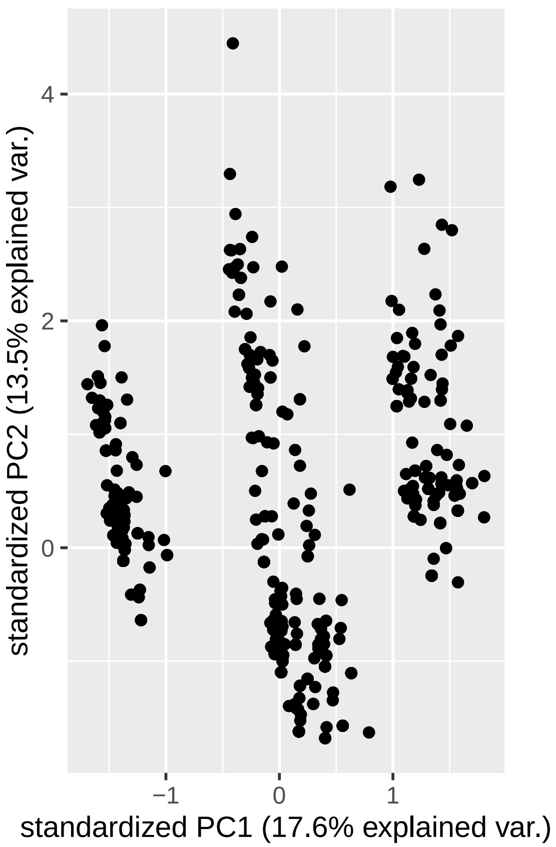


The first 8 genes are mutated in >100 patients. (>10% of the patients because there are 1006 patients in the dataset.)

Mutation Clustering

PCA Analysis

```
# ------------- PCA ON MUTATION DATA -------------
feature.pca <- prcomp(feature_df_filtered, center = TRUE)

# Plot the top 2 PCs and look at PCA summary
ggbiplot(feature.pca, var.axes = FALSE, ellipse = TRUE)
```



```r
summary(feature.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     0.5281 0.4636 0.4113 0.33597 0.31753 0.30566 0.29134
## Proportion of Variance 0.1757 0.1354 0.1066 0.07113 0.06354 0.05887 0.05349
## Cumulative Proportion  0.1757 0.3112 0.4178 0.48889 0.55243 0.61130 0.66479
##                            PC8     PC9    PC10    PC11   PC12    PC13    PC14
## Standard deviation     0.26705 0.26520 0.24498 0.23575 0.2264 0.22177 0.21687
## Proportion of Variance 0.04494 0.04432 0.03782 0.03502 0.0323 0.03099 0.02964
## Cumulative Proportion  0.70973 0.75405 0.79186 0.82689 0.8592 0.89018 0.91982
##                           PC15    PC16    PC17
## Standard deviation     0.21204 0.20693 0.19864
## Proportion of Variance 0.02833 0.02698 0.02486
## Cumulative Proportion  0.94815 0.97514 1.00000
```

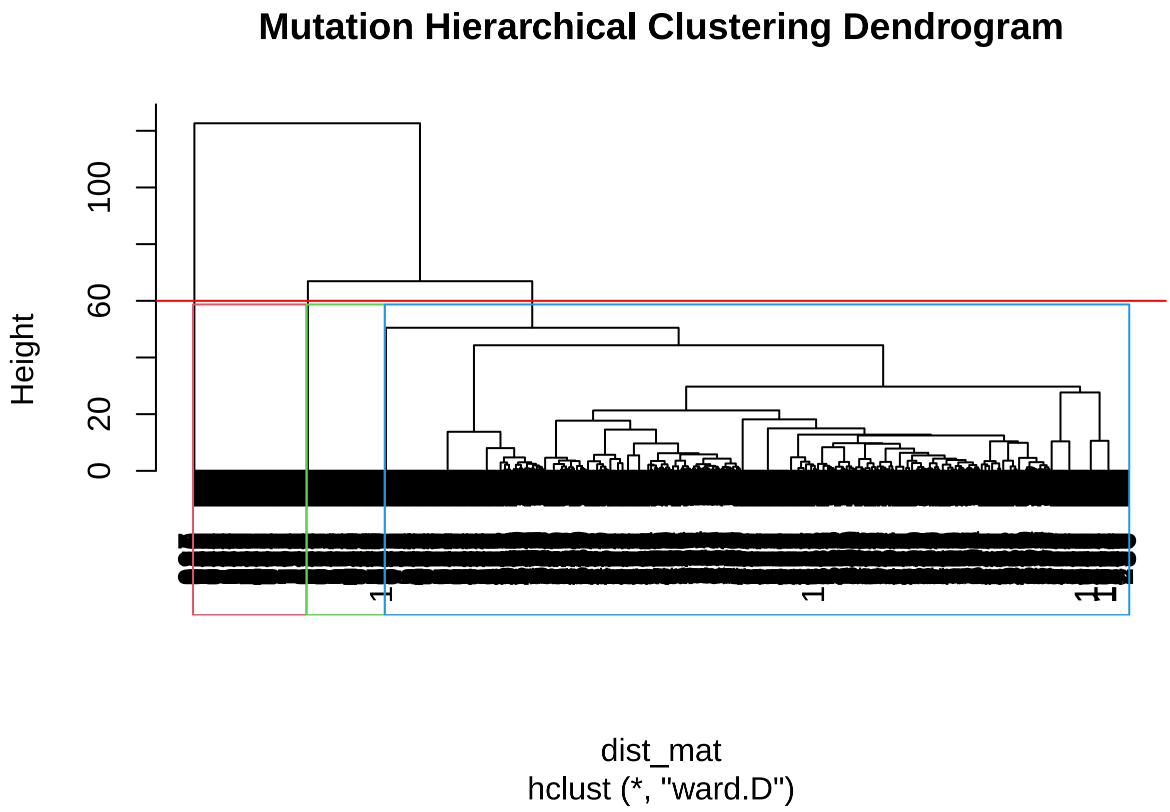
The first principal component axis seems to be separating the patients into three distinct groups. Overall, 85% of the variance can be captured in 12 PCs.

Hierarchical clustering:

We performed hierarchical clustering on the filtered data using single linkage, average linkage, complete linkage and Ward linkage methods. The Ward method yielded the most evenly distributed clusters. We chose to cut the tree at k=3 as the PCA yielded 3 distinct clusters on the first principal component axis.

```
# ------------- HIERARCHICAL CLUSTERING ON RAW MUTATION DATA -------------
# Perform hierarchical clustering
dist_mat <- dist(feature_df_filtered, method = 'binary')
hclust_ward <- hclust(dist_mat, method = 'ward.D')
```

```
# Cut tree at k=3 to create 3 clusters
cut_ward <- cutree(hclust_ward, k = 3)

# Plot a dendogram to visualize clusters
plot(hclust_ward, main = "Mutation Hierarchical Clustering Dendrogram")
rect.hclust(hclust_ward, k = 3, border = 2:6)
abline(h = 60, col = 'red')
```



```
# Look at cluster sizes
cluster.size.hc <- table(cut_ward)
cluster.size.hc
```

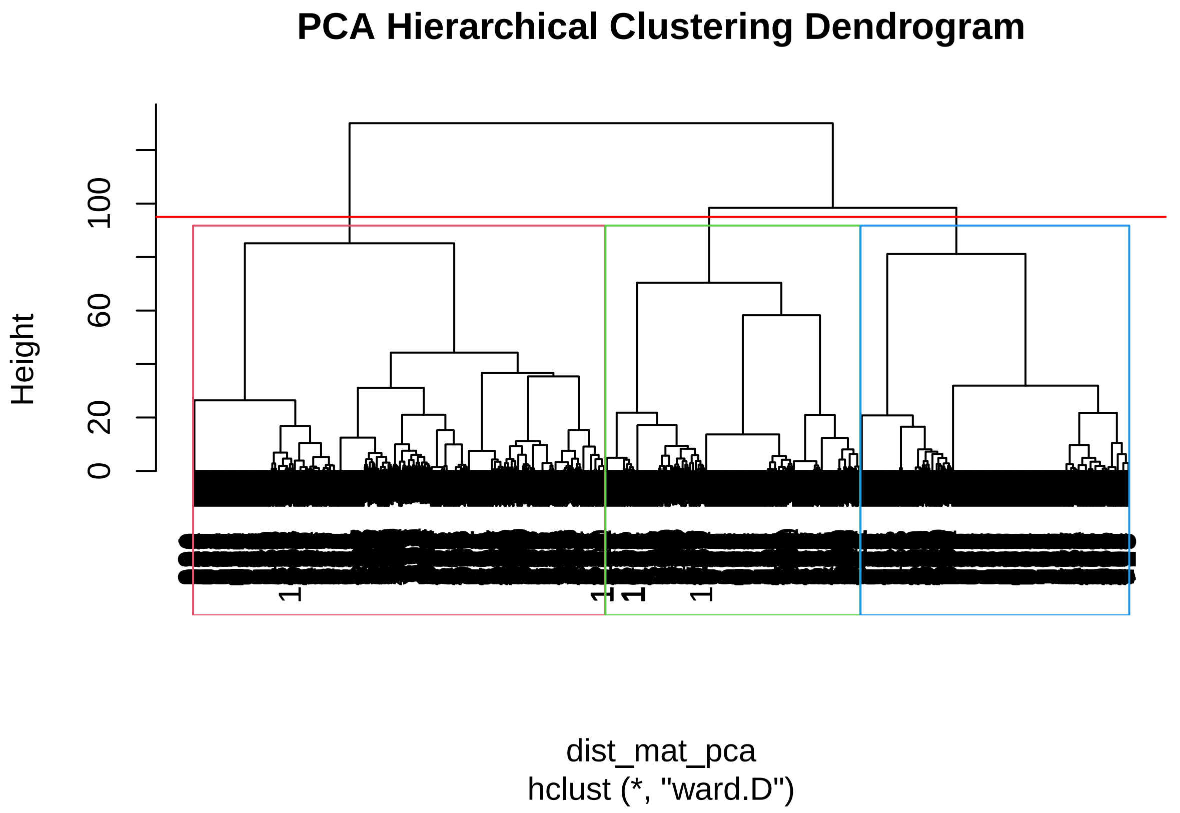
```
## cut_ward
##   1   2   3
## 800 122  84
```

Next, we performed hierarchical clustering on the PCA data. We chose to preserve 85% percent of the data as this gave us the most evenly distributed and distinct clusters. We ran the clustering using single linkage, average linkage, complete linkage and Ward linkage methods. The Ward method yielded the best clusters.

Again, we cut the tree at k=3 as our PCA yielded 3 clusters. One of the clusters is very large, and the other two are small. Next, we tried hierarchical clustering on the PC-transformed data.

```
# ------------- HIERARCHICAL CLUSTERING ON PCA MUTATION DATA -------------
# Use top 12 PCs to preserve ~85% of the variance
dist_mat_pca <- dist(feature.pca$x[,1:12], method = 'euclidean')

# Perform hierarchical clustering
hclust_pca <- hclust(dist_mat_pca, method = 'ward.D')
```

```
# Cut tree at k=3 to create 3 clusters
cut_pca <- cutree(hclust_pca, k=3)

# Plot a dendogram to visualize clusters
plot(hclust_pca, main = "PCA Hierarchical Clustering Dendrogram")
rect.hclust(hclust_pca,k=3, border = 2:6)
abline(h=95, col = 'red')
```

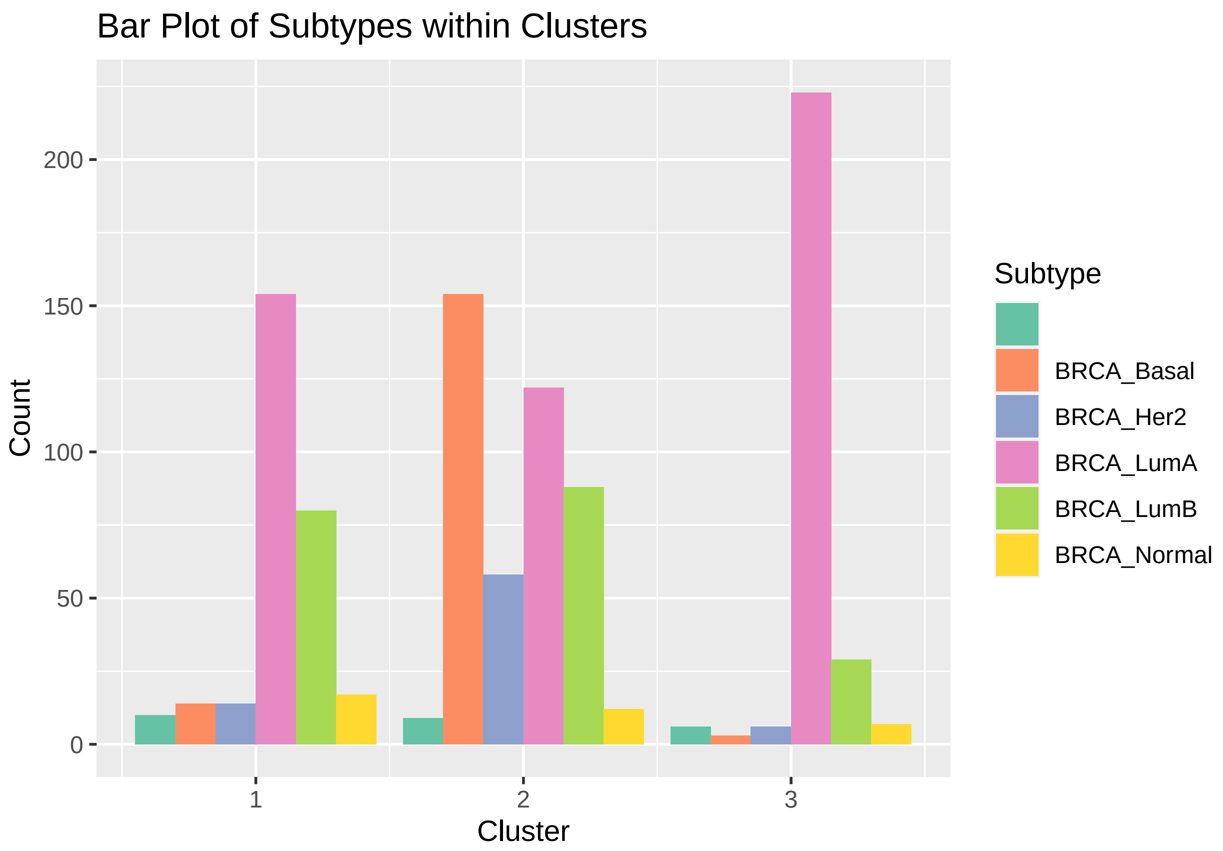


```
# Look at cluster sizes
cut_pca <- as.data.frame(cut_pca)
colnames(cut_pca) <- c("cluster")
cluster.size.hc.pca <- table(cut_pca)
cluster.size.hc.pca
```

```
## cluster
##   1   2   3
## 289 443 274
```

The hierarchical clustering on our PCA data yielded better, more distinct and more evenly distributed clusters therefore, we will continue our analysis using these 3 clusters.

```r
# ------------- VISUALIZE CLUSTERS VERSUS TUMOUR-SUBTYPES -------------
# Find indices of all patients in the clinical data that have been clustered
# (1006 patients)
idx <- which(data.clinical$PATIENT_ID %in% patient.order)

# Create a data frame containing the patient ID, their BRCA subtype and their
# cluster assignment
clust.sub.data <- data.frame(Patient = data.clinical$PATIENT_ID[idx],
                             Subtype = data.clinical$SUBTYPE[idx],
                             Clusters = cut_pca)

# Visualize the BRCA suptypes within each cluster using a bar plot
ggplot(clust.sub.data, aes(x = cut_pca$cluster, fill = Subtype)) +
  geom_bar(position = "dodge") +
  labs(title = "Bar Plot of Subtypes within Clusters",
       x = "Cluster",
       y = "Count") +
  scale_fill_brewer(palette = "Set2", name = "Subtype")
```

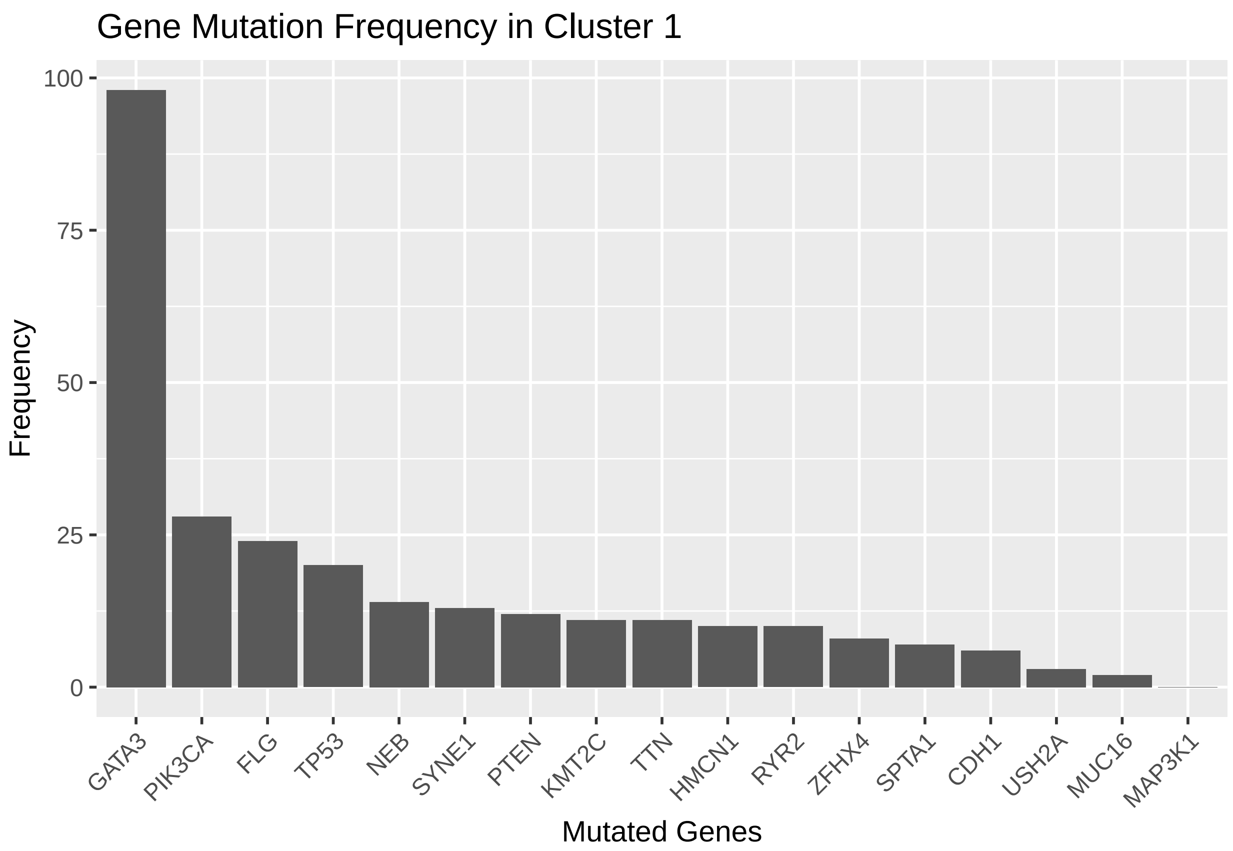


Cluster 1 has luminal A as the most prominent subtype with luminal B as the next. This is similar to cluster 3 however the luminal A subtype prominence is much more distinct in cluster 3. Cluster 2 contains a more even distribution of the basal-like, luminal A and luminal B subtypes with slight HER2 prominence as well. Basal-like and Her2 subtypes are almost exclusively found in cluster 2.

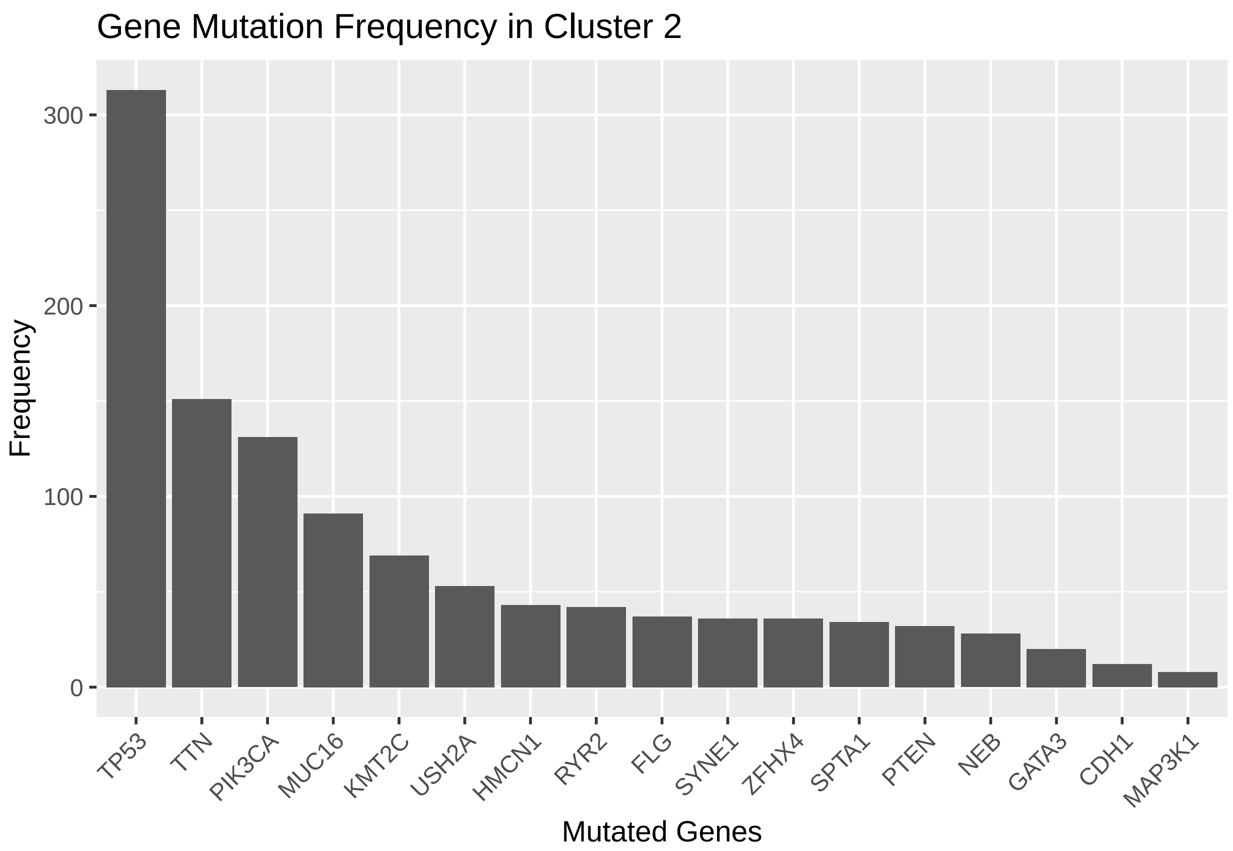
Next, we looked at the most frequently mutated genes in each cluster to observe any concordance with current literature on the different BRCA subtypes.

```r
# ------------- VISUALIZE MUTATION FREQS IN CLUSTERS -------------
# This function takes the cluster number as an input and returns a frequency
# table data frame of mutated genes in the cluster
freqData <- function(clust) {
  cluster <- feature_df_filtered[which(cut_pca == clust),]

  # Create frequency table
  freq <- colSums(cluster == 1)

  # Convert frequency table to a data frame
  data <- data.frame(column = names(freq), frequency = freq)

  # Sort the data frame by frequency in descending order
  data <- data[order(data$frequency, decreasing = TRUE), ]
}
```

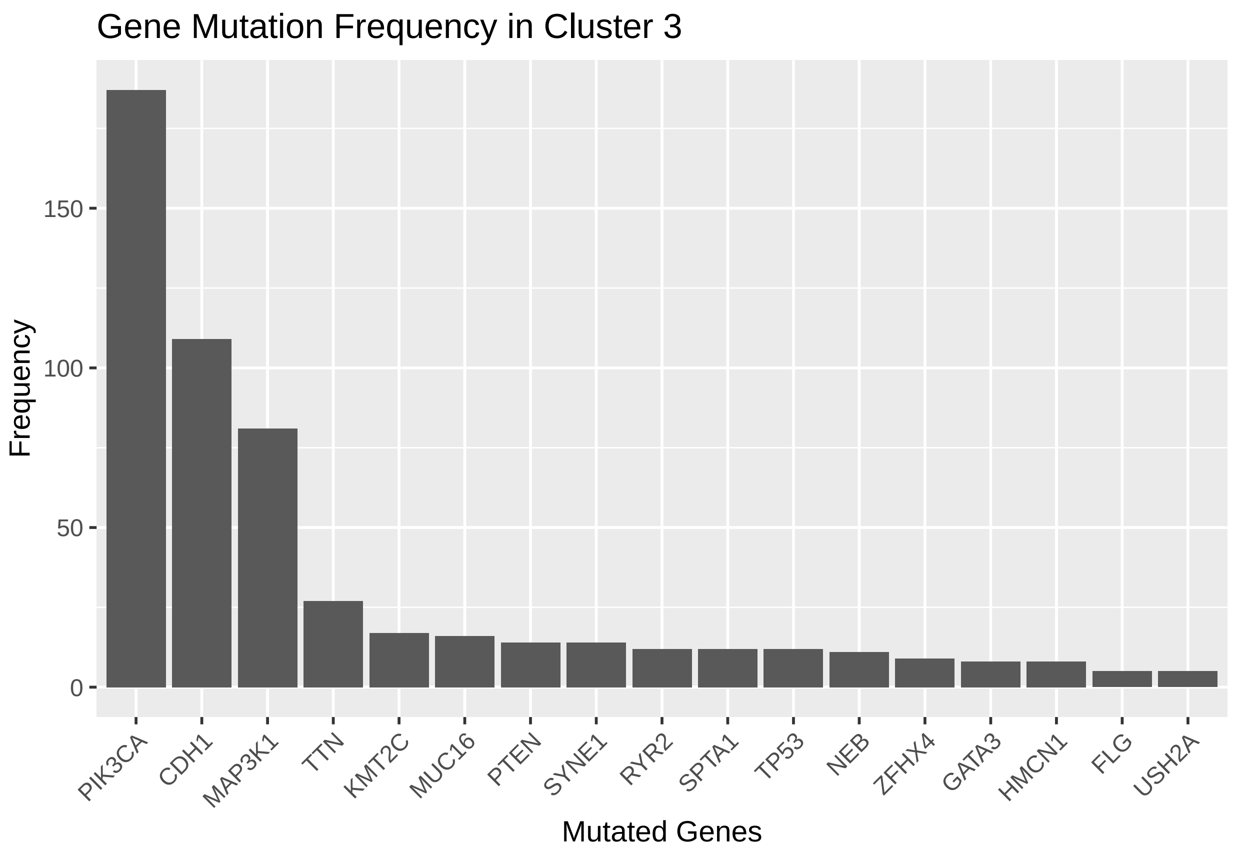
```r
# Get mutation frequency data for each cluster
clust.data1 <- freqData(1)
clust.data2 <- freqData(2)
clust.data3 <- freqData(3)

# Create bar plots to visualize the most frequently mutated gene in each cluster
ggplot(clust.data1, aes(x = column, y = frequency)) +
  geom_col() + theme(axis.text.x = element_text(angle = 45,hjust=1)) +
  scale_x_discrete(limits = clust.data1$column) +
  labs(x = "Mutated Genes", y = "Frequency",
       title = "Gene Mutation Frequency in Cluster 1")
```



```r
ggplot(clust.data2, aes(x = column, y = frequency)) +
  geom_col() + theme(axis.text.x = element_text(angle = 45,hjust=1)) +
  scale_x_discrete(limits = clust.data2$column) +
  labs(x = "Mutated Genes", y = "Frequency",
       title = "Gene Mutation Frequency in Cluster 2")
```

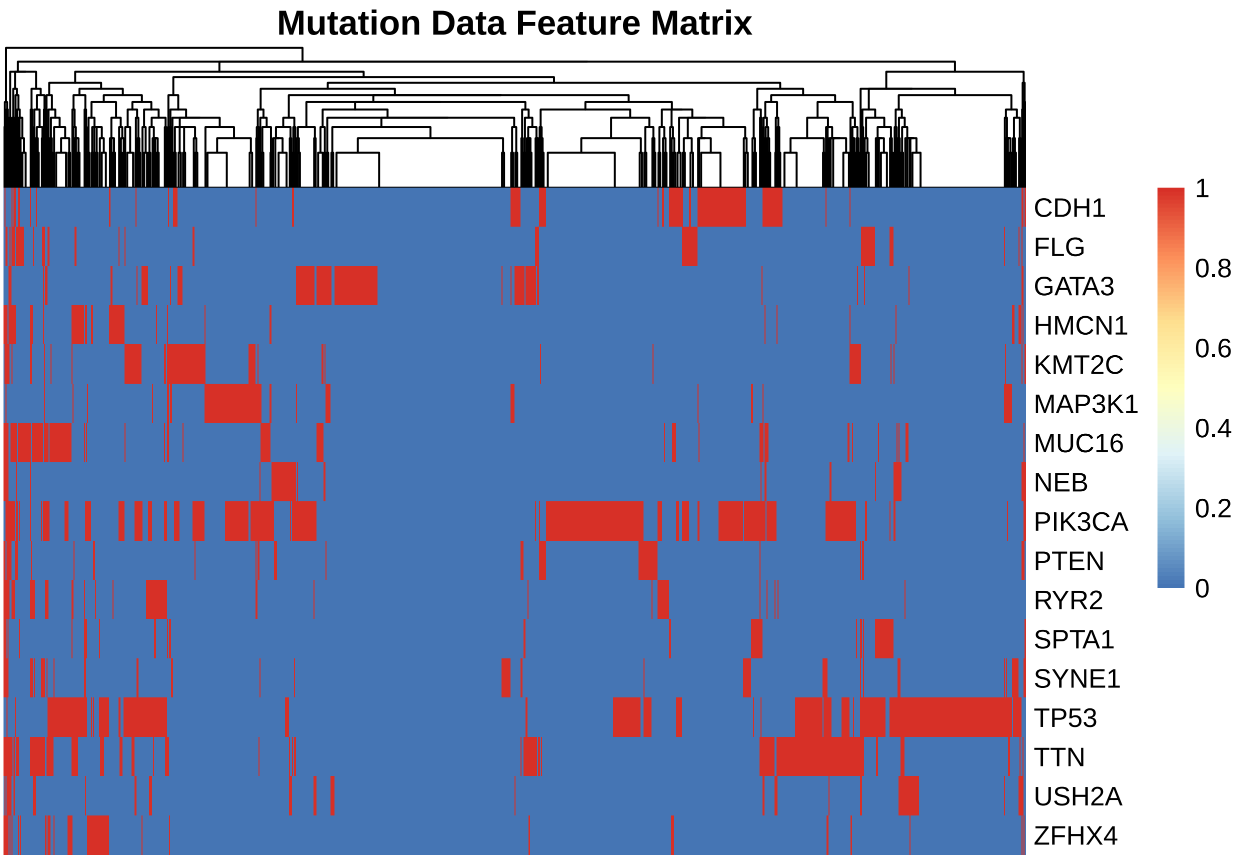


```r
ggplot(clust.data3, aes(x = column, y = frequency)) +
  geom_col() + theme(axis.text.x = element_text(angle = 45,hjust=1)) +
  scale_x_discrete(limits = clust.data3$column) +
  labs(x = "Mutated Genes", y = "Frequency",
       title = "Gene Mutation Frequency in Cluster 3")
```



Cluster 2 has TP53 as its most frequent mutated gene. As basal-like is the most prominent subtype in cluster 2, this result is consistent with literature stating that TP53 is the most mutated and other mutations that were significant in luminal subtypes, such as CDH1 and MAP3K1, are absent [2]. Cluster 3 has PIK3CA as the most frequently mutated gene followed by CDH1 and MAP3K1. This is mostly consistent with literature about the luminal A subtype except for prominent mutations in GATA3, TP53 and MAP2K4 are missing in this cluster [2]. The most frequently mutated genes in cluster 1 do not correspond well with any specific subtypes based on literature [2]. Although it makes sense that this group has a high concentration of luminal A and B subtypes as GATA3 has been found to be frequently mutated in luminal-like breast cancers [8].

```
# ------------- VISUALIZE ONCOPLOT -------------
pheatmap_mat <- t(feature_df_filtered)

library(pheatmap)
reduce.mat <- pheatmap_mat
res <- pheatmap(reduce.mat,
         cluster_rows = F,
         show_colnames=FALSE,
         main = "Mutation Data Feature Matrix")
```



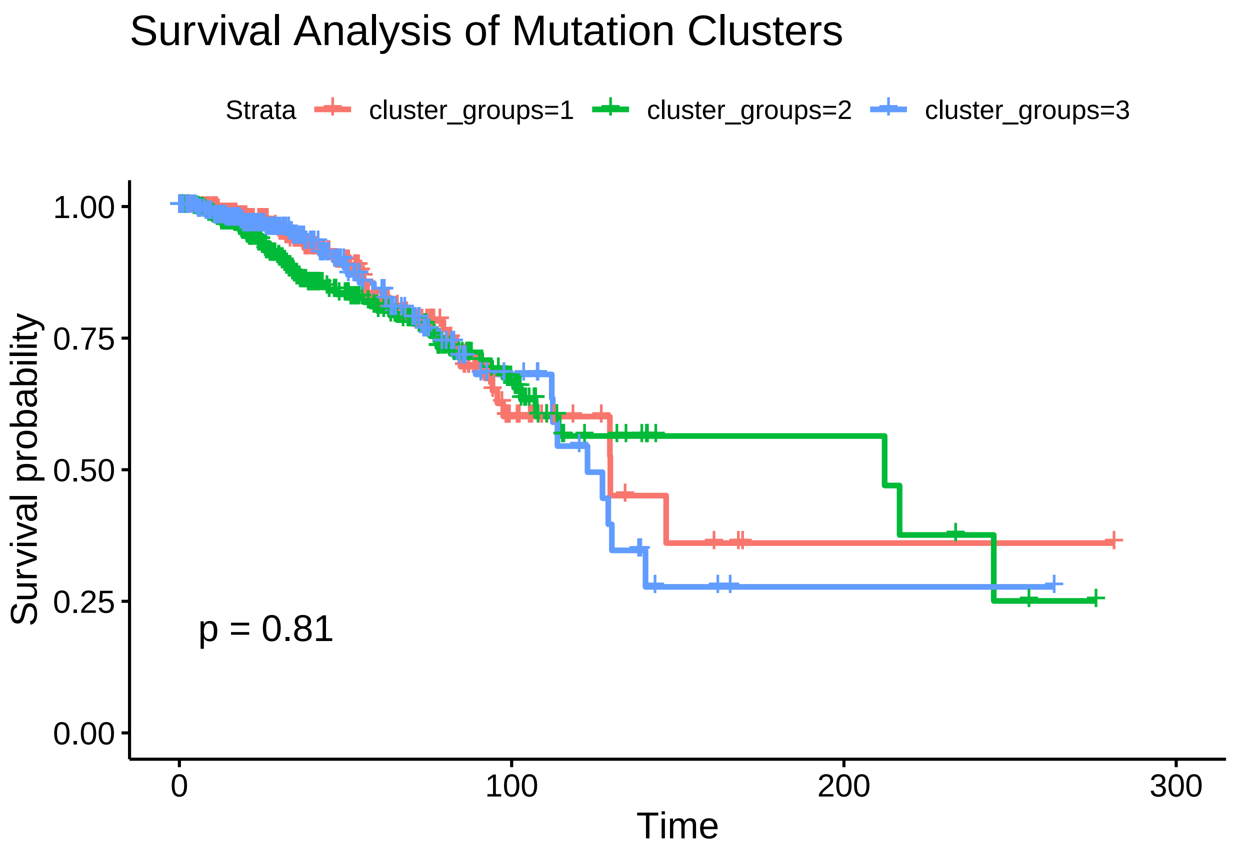
Survival Analysis on Mutation Clusters

```r
# ------------- FORMAT SURVIVAL DATA -------------
#Create a subset of the clinicalData containing only the patients with mutated
# genes after filtering
clinical_cleaned <- data.clinical[which(data.clinical$PATIENT_ID
                                        %in% unique.patients.full.data) ,]

#Create a data frame called survival with a vector that contains
# TRUE = dead and FALSE = alive
survival_DF <- data.frame(deceased = clinical_cleaned$OS_STATUS == "1:DECEASED")

#Create a column of months to death from diagnosis
indices <- which(clinical_cleaned$OS_STATUS == "1:DECEASED")

#Set all to NA first since patients who are not dead should not have
# an OS_months value
survival_DF$months_to_death = rep(NA, length(unique.patients.full.data))

for(i in indices) {
  survival_DF$months_to_death[i] = clinical_cleaned$OS_MONTHS[i]
}
```

```r
# ------------- SURVIVAL ANALYSIS -------------
survival_DF$progression_free = clinical_cleaned$PFS_MONTHS

# create an "overall survival" variable that is equal to days_to_death
# for dead patients, and to progression free disease for patients who
# are still alive
survival_DF$overall_survival = ifelse(survival_DF$deceased,
                                   survival_DF$months_to_death,
                                   survival_DF$progression_free )

#Create a vector within survival dataframe containing cluster groups
# for labelling
survival_DF$cluster_groups <- cut_pca$cluster

#Now that the survival time has been tagged with the censoring, we can add the
# categorical independent variable `cluster groups`, and effectively create a
# formula

Surv(survival_DF$overall_survival,
     survival_DF$deceased) ~ survival_DF$cluster_groups
```

```
## Surv(survival_DF$overall_survival, survival_DF$deceased) ~ survival_DF$cluster_groups
```

```r
fit = survfit(Surv(overall_survival,
                   deceased) ~ cluster_groups, data=survival_DF)
ggsurvplot(fit, data=survival_DF, pval=T,
           title = "Survival Analysis of Mutation Clusters")
```



No significant survival analysis difference was found between the three mutation analysis clusters.

Expression Analysis on Mutation Clusters

```
# ------------- CLEAN AND FORMAT EXPRESSION DATA -------------
# Set gene IDs as rownames
rownames(data.expression) <- data.expression[, 1]
# Remove excess gene IDs column
data.expression <- data.expression[, -1]

# Clean up patient tags for expression data
patient.ids <- colnames(data.expression)
patient.ids.shortened <- substr(patient.ids, 1, 12)
patient.ids.shortened.sub <- gsub("\\.", "-", patient.ids.shortened)
colnames(data.expression) <- patient.ids.shortened.sub

# Get the data from patients with full data
data.expression <- data.expression[, unique.patients.full.data]
```

```
# Filter out genes that only have 0 or 1 read counts accross all samples
counts <- data.expression[rowSums(data.expression) > 1, ]
counts.df <- as.data.frame(counts)
```

```r
# Make the study design data frame
# This dataframe just has the cluster assignments
cluster.assignments <- cut_pca$cluster
cluster.assignments.df <- as.data.frame(factor(cluster.assignments))

rownames(cluster.assignments.df) <- colnames(counts.df)
colnames(cluster.assignments.df) <- c("cluster")
```

```r
# ------------- DIFFERENTIAL EXPRESSION ON MUTATION CLUSTERS -------------
dds = DESeqDataSetFromMatrix(countData=counts,
                              colData=cluster.assignments.df,
                              design=~cluster)
```

```r
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 12184 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

```r
dds
```

```
## class: DESeqDataSet
## dim: 57664 1006
## metadata(1): version
## assays(6): counts mu ... replaceCounts replaceCooks
## rownames(57664): ENSG00000000003.15 ENSG00000000005.6 ...
##   ENSG00000288674.1 ENSG00000288675.1
## rowData names(27): baseMean baseVar ... maxCooks replace
## colnames(1006): TCGA-3C-AAAU TCGA-3C-AALI ... TCGA-Z7-A8R5 TCGA-Z7-A8R6
## colData names(3): cluster sizeFactor replaceable
```

We have decided to focus on comparing clusters 2 and 3 in differential expression and pathway analysis. Please see the 'methods' and 'discussion' section for why we made this decision.

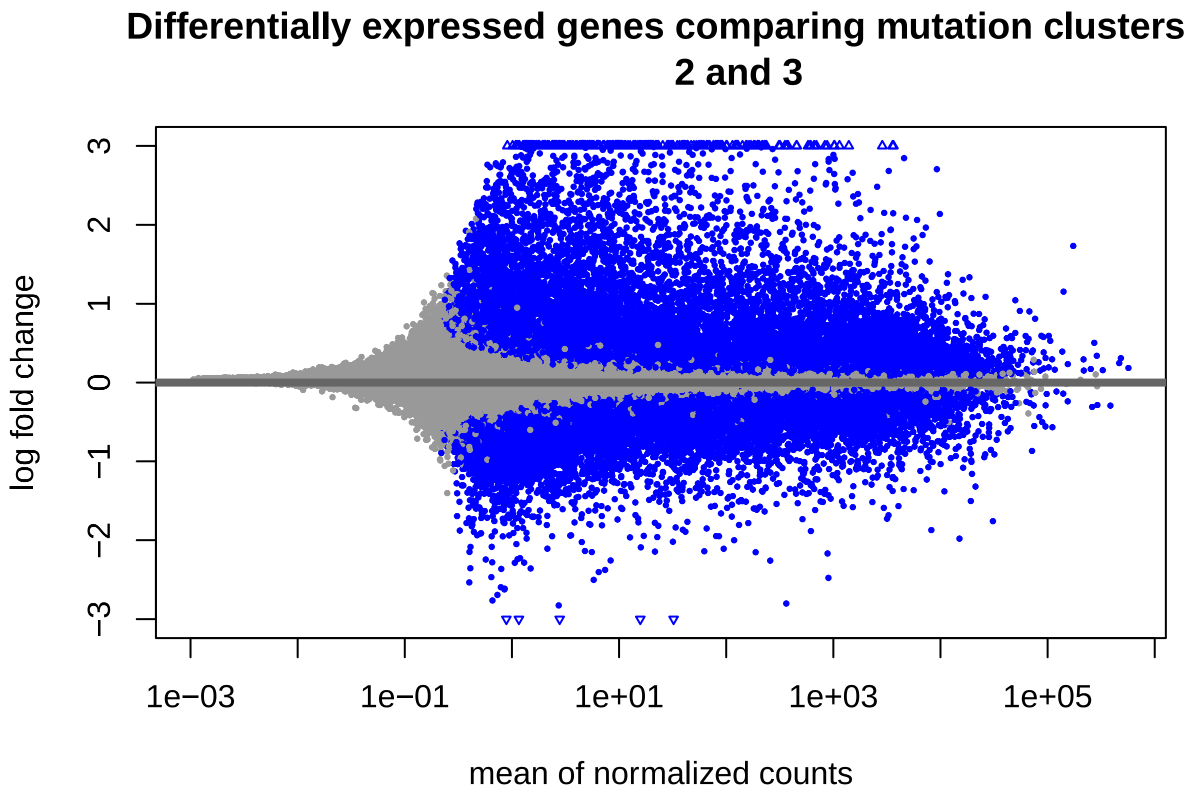
```r
# We are going to analyze clusters 2 and 3
res <- results(dds, contrast = c("cluster", 2, 3))
res
```

```
## log2 fold change (MLE): cluster 2 vs 3
## Wald test p-value: cluster 2 vs 3
## DataFrame with 57664 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE      stat      pvalue
##                    <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000000003.15  3126.1978       0.154127 0.0745979   2.06610 3.88190e-02
## ENSG00000000005.6    77.9416      -0.650491 0.1933116  -3.36499 7.65471e-04
## ENSG00000000419.13  2377.3111       0.471165 0.0425712  11.06771 1.79934e-28
## ENSG00000000457.14  1580.9323      -0.176608 0.0438166  -4.03063 5.56282e-05
## ENSG00000000460.17   719.0585       0.487270 0.0597967   8.14877 3.67644e-16
## ...                       ...            ...       ...       ...         ...
## ENSG00000288667.1    0.219751     -0.5638561 0.4653078 -1.211792 0.225592166
## ENSG00000288669.1    0.154189     -0.0689671 0.4638630 -0.148680 0.881806228
## ENSG00000288670.1  425.309403     -0.0365657 0.0521244 -0.701508 0.482986041
## ENSG00000288674.1    8.084830     -0.2191389 0.0748251 -2.928681 0.003404039
## ENSG00000288675.1   33.146202      0.3094663 0.0848159  3.648682 0.000263589
##                           padj
##                      <numeric>
## ENSG00000000003.15 6.57374e-02
## ENSG00000000005.6  1.87350e-03
## ENSG00000000419.13 6.79560e-27
## ENSG00000000457.14 1.67533e-04
## ENSG00000000460.17 4.68296e-15
## ...                        ...
## ENSG00000288667.1  0.305535880
## ENSG00000288669.1           NA
## ENSG00000288670.1  0.573088337
## ENSG00000288674.1  0.007339708
## ENSG00000288675.1  0.000703589
```

```r
summary(res)
```

```
##
## out of 57664 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 14700, 25%
## LFC < 0 (down)     : 12193, 21%
## outliers [1]       : 0, 0%
## low counts [2]     : 14534, 25%
## (mean count < 0)
## [1] see ’cooksCutoff’ argument of ?results
## [2] see ’independentFiltering’ argument of ?results
```

```r
# ------------- VISUALIZE DE RESULTS ON MUTATION CLUSTERS -------------
plotMA(res, ylim = c(-3, 3),
       main = "Differentially expressed genes comparing mutation clusters
               2 and 3")
```



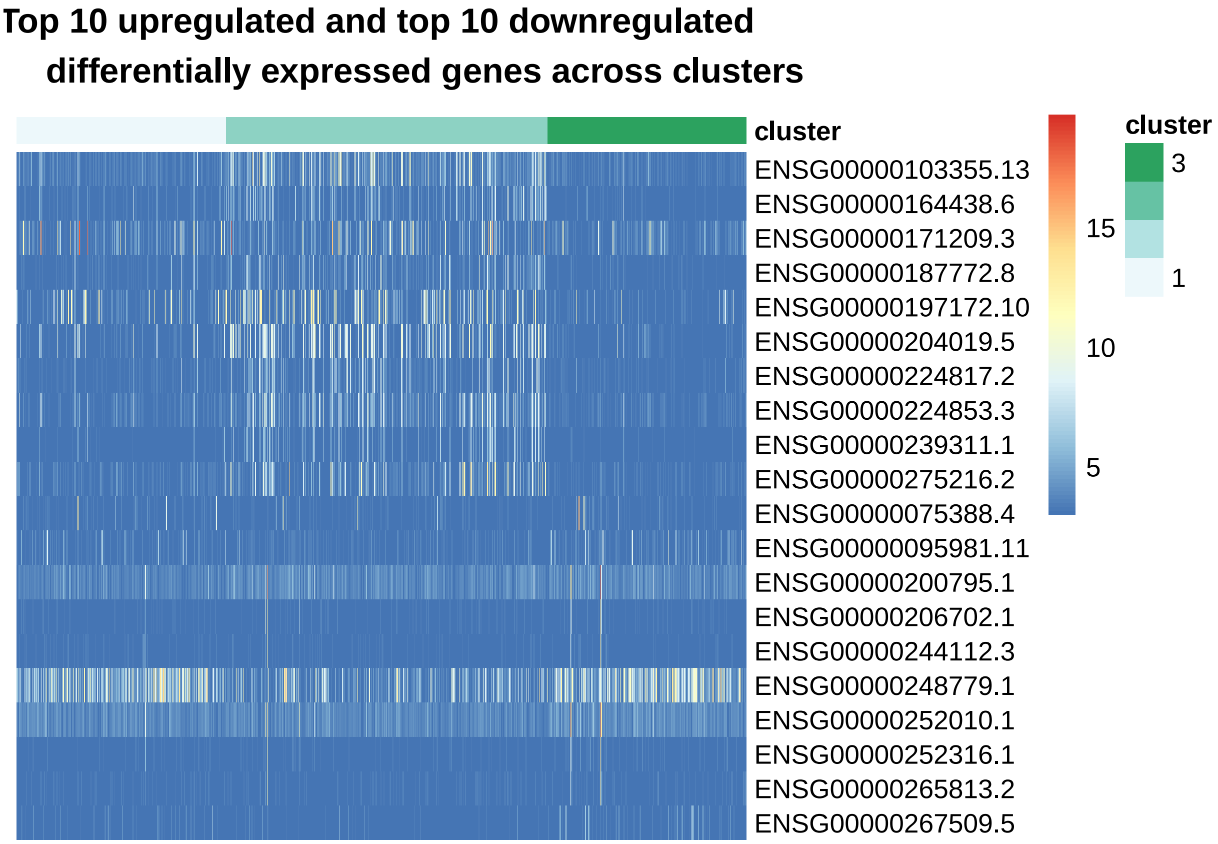
```
# Variance stabilizing transform the DESeq results
vst <- vst(dds, blind = FALSE)
```

```
# Filter for significant genes
res.sig <- subset(res, padj < 0.0001)
# Select the top 10 over-expressed genes
de.genes.up <- rownames(res.sig)[order(res.sig$log2FoldChange,
                                       decreasing = TRUE)[1:10]]
de.genes.up <- which(rownames(res) %in% de.genes.up)
# Select the top 10 under-expressed genes
de.genes.down <- rownames(res.sig)[order(res.sig$log2FoldChange,
                                         decreasing = FALSE)[1:10]]
de.genes.down <- which(rownames(res) %in% de.genes.down)
# Bind significant genes together
de.genes <- c(de.genes.up, de.genes.down)
```

```
# Plot the log2 fold expression of the top 20 differentially expressed genes
sampleMatrix <- assay(vst)[de.genes,]
colnames(sampleMatrix) <- colnames(counts)

de.genes.cleaned <- sapply(strsplit(rownames(counts[de.genes,]),
                                    "\\."), "[", 1)
de.genes.sym = mapIds(org.Hs.eg.db,
                     keys=de.genes.cleaned,
                     column="SYMBOL",
                     keytype="ENSEMBL",
                     multiVals="first")
```

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

```
# Commented out because some genes have NA symbols
# rownames(sampleMatrix) <- de.genes.sym

cluster.order.df <- as.data.frame(sort(cut_pca$cluster, decreasing = FALSE))
colnames(cluster.order.df) <- c(cluster)

patient.order.df <- as.data.frame(list(colnames(counts), cut_pca$cluster))
colnames(patient.order.df) <- c("patient", "cluster")
patient.order.df <- patient.order.df[order(patient.order.df$cluster), ]

annotation.df <- as.data.frame(patient.order.df$cluster)
rownames(annotation.df) <- patient.order.df$patient
colnames(annotation.df) <- c("cluster")

sampleMatrix <- sampleMatrix[, patient.order.df$patient]


library(pheatmap)
pheatmap(sampleMatrix,
         cluster_rows=FALSE,
         show_rownames=TRUE,
         show_colnames = FALSE,
         cluster_cols=FALSE,
         annotation_col = annotation.df,
         main = "Top 10 upregulated and top 10 downregulated
         differentially expressed genes across clusters")
```



The above plot shows the top 10 significant upregulated genes and the top 10 significant downregulated genes (sorted by cluster at the top). The clusters are able to achieve some level of separation, as the top 10 upregulated genes in cluster 2 are more highly expressed compared to cluster 3.

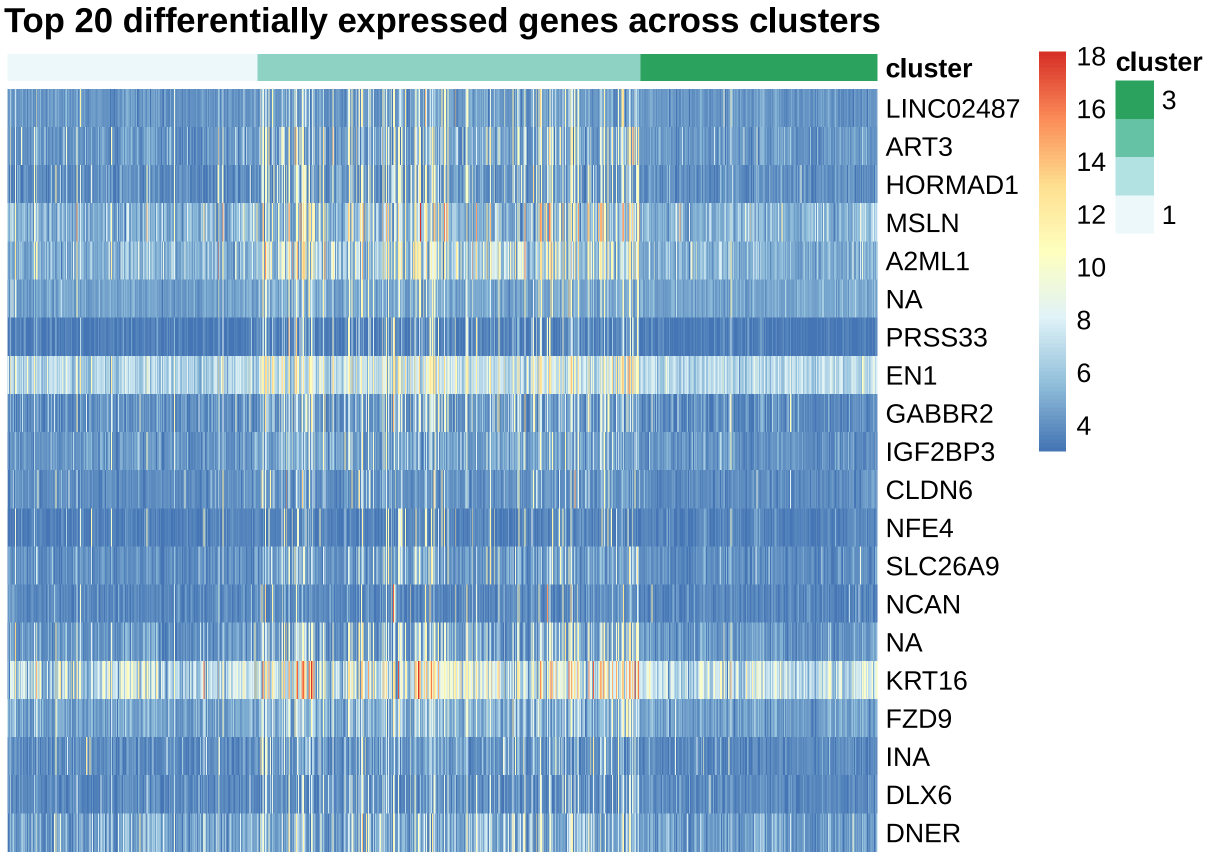
```r
de.genes <- order(res$padj, decreasing = FALSE)[1:20]
```

```r
# Plot the log2 fold expression of the top 20 differentially expressed genes
sampleMatrix <- assay(vst)[de.genes,]
colnames(sampleMatrix) <- colnames(counts)

de.genes.cleaned <- sapply(strsplit(rownames(counts[de.genes,]),
                                    "\\."), "[", 1)
de.genes.sym = mapIds(org.Hs.eg.db,
                      keys=de.genes.cleaned,
                      column="SYMBOL",
                      keytype="ENSEMBL",
                      multiVals="first")
```

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

```r
rownames(sampleMatrix) <- de.genes.sym

cluster.order.df <- as.data.frame(sort(cut_pca$cluster, decreasing = FALSE))
colnames(cluster.order.df) <- c(cluster)

patient.order.df <- as.data.frame(list(colnames(counts), cut_pca$cluster))
```

```
colnames(patient.order.df) <- c("patient", "cluster")
patient.order.df <- patient.order.df[order(patient.order.df$cluster), ]

annotation.df <- as.data.frame(patient.order.df$cluster)
rownames(annotation.df) <- patient.order.df$patient
colnames(annotation.df) <- c("cluster")

sampleMatrix <- sampleMatrix[, patient.order.df$patient]


library(pheatmap)
pheatmap(sampleMatrix,
         cluster_rows=FALSE,
         show_rownames=TRUE,
         show_colnames = FALSE,
         cluster_cols=FALSE,
         annotation_col = annotation.df,
         main = "Top 20 differentially expressed genes across clusters")
```



Above it is seen that differential expression analysis comparing clusters 2 and 3 shows that cluster 2 has significant upregulated genes, and cluster 3 is downregulated in comparison. These significant genes are not overlapping with any of the highly mutated genes.

```
# ------------- ADD GENE ANNOTATIONS -------------
gene.names.cleaned <- sapply(strsplit(row.names(res), "\\."), "[", 1)
res$symbol = mapIds(org.Hs.eg.db,
                    keys=gene.names.cleaned,
                    column="SYMBOL",
                    keytype="ENSEMBL",
                    multiVals="first")
```

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

```
res$entrez = mapIds(org.Hs.eg.db,
                     keys=gene.names.cleaned,
                     column="ENTREZID",
                     keytype="ENSEMBL",
                     multiVals="first")
```

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

```
res$name =   mapIds(org.Hs.eg.db,
                     keys=gene.names.cleaned,
                     column="GENENAME",
                     keytype="ENSEMBL",
                     multiVals="first")
```

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

```
# ------------- PATHWAY ANALYSIS ON MUTATION CLUSTERS -------------
data(kegg.sets.hs)
data(sigmet.idx.hs)

# Focus on signaling and metabolic pathways only
kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]
```

```
fold.changes <- res$log2FoldChange
names(fold.changes) <- res$entrez

# Gage pathway analysis
kegg.res = gage(fold.changes, gsets=kegg.sets.hs)
```

```
head(kegg.res$less, 6)
```

```
##                                                         p.geomean stat.mean
## hsa04710 Circadian rhythm - mammal                     0.03670579 -1.856752
## hsa00071 Fatty acid metabolism                         0.12685443 -1.149557
## hsa04614 Renin-angiotensin system                      0.13879325 -1.106012
## hsa04150 mTOR signaling pathway                        0.15093135 -1.038877
## hsa00280 Valine, leucine and isoleucine degradation 0.15131535 -1.037543
## hsa04270 Vascular smooth muscle contraction            0.20303735 -0.832351
##                                                             p.val     q.val
## hsa04710 Circadian rhythm - mammal                     0.03670579 0.9999979
## hsa00071 Fatty acid metabolism                         0.12685443 0.9999979
## hsa04614 Renin-angiotensin system                      0.13879325 0.9999979
## hsa04150 mTOR signaling pathway                        0.15093135 0.9999979
## hsa00280 Valine, leucine and isoleucine degradation 0.15131535 0.9999979
## hsa04270 Vascular smooth muscle contraction         0.20303735 0.9999979
##                                                    set.size       exp1
## hsa04710 Circadian rhythm - mammal                        22 0.03670579
## hsa00071 Fatty acid metabolism                            43 0.12685443
## hsa04614 Renin-angiotensin system                         17 0.13879325
## hsa04150 mTOR signaling pathway                           52 0.15093135
## hsa00280 Valine, leucine and isoleucine degradation       44 0.15131535
## hsa04270 Vascular smooth muscle contraction              116 0.20303735
```

Only one pathway was found as significantly differentially expressed and downregulated in cluster 2 comapared to cluster 3, and this pathway is mammal Circadian rhythm. As seen in the expression heatmaps, cluster 2 generally has a much high gene expression on average. These results make sense because almost no pathways are being overexpressed in cluster 3 compared to cluster 2.

```r
head(kegg.res$greater, 20)
```

```
##                                                                        p.geomean
## hsa04110 Cell cycle                                                 2.121346e-06
## hsa04650 Natural killer cell mediated cytotoxicity                  1.408658e-04
## hsa04612 Antigen processing and presentation                        1.732783e-04
## hsa03050 Proteasome                                                 1.503714e-03
## hsa03030 DNA replication                                            1.693928e-03
## hsa04145 Phagosome                                                  5.178415e-03
## hsa03008 Ribosome biogenesis in eukaryotes                          6.896681e-03
## hsa03013 RNA transport                                              1.063035e-02
## hsa04062 Chemokine signaling pathway                                1.162438e-02
## hsa04514 Cell adhesion molecules (CAMs)                             1.237288e-02
## hsa04114 Oocyte meiosis                                             1.646237e-02
## hsa00270 Cysteine and methionine metabolism                         1.857212e-02
## hsa00190 Oxidative phosphorylation                                  1.980619e-02
## hsa00520 Amino sugar and nucleotide sugar metabolism                2.211052e-02
## hsa00860 Porphyrin and chlorophyll metabolism                       2.254043e-02
## hsa00980 Metabolism of xenobiotics by cytochrome P450               2.422418e-02
## hsa00040 Pentose and glucuronate interconversions                   2.471554e-02
## hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series 2.925498e-02
## hsa04141 Protein processing in endoplasmic reticulum                3.197912e-02
## hsa04970 Salivary secretion                                         3.198568e-02
##                                                                     stat.mean
## hsa04110 Cell cycle                                                  4.704742
## hsa04650 Natural killer cell mediated cytotoxicity                   3.681698
## hsa04612 Antigen processing and presentation                         3.688754
## hsa03050 Proteasome                                                  3.081173
## hsa03030 DNA replication                                             3.070740
## hsa04145 Phagosome                                                   2.581521
## hsa03008 Ribosome biogenesis in eukaryotes                           2.506929
## hsa03013 RNA transport                                               2.318566
## hsa04062 Chemokine signaling pathway                                 2.278939
## hsa04514 Cell adhesion molecules (CAMs)                              2.258692
## hsa04114 Oocyte meiosis                                              2.146479
## hsa00270 Cysteine and methionine metabolism                          2.125827
## hsa00190 Oxidative phosphorylation                                   2.068952
## hsa00520 Amino sugar and nucleotide sugar metabolism                 2.040017
## hsa00860 Porphyrin and chlorophyll metabolism                        2.037663
## hsa00980 Metabolism of xenobiotics by cytochrome P450                1.994203
## hsa00040 Pentose and glucuronate interconversions                    2.009752
## hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series  1.937677
## hsa04141 Protein processing in endoplasmic reticulum                 1.861037
## hsa04970 Salivary secretion                                          1.866593
##                                                                            p.val
## hsa04110 Cell cycle                                                 2.121346e-06
## hsa04650 Natural killer cell mediated cytotoxicity                  1.408658e-04
## hsa04612 Antigen processing and presentation                        1.732783e-04
## hsa03050 Proteasome                                                 1.503714e-03
## hsa03030 DNA replication                                            1.693928e-03
## hsa04145 Phagosome                                                  5.178415e-03
## hsa03008 Ribosome biogenesis in eukaryotes                          6.896681e-03
## hsa03013 RNA transport                                              1.063035e-02
## hsa04062 Chemokine signaling pathway                                1.162438e-02
## hsa04514 Cell adhesion molecules (CAMs)                             1.237288e-02
## hsa04114 Oocyte meiosis                                             1.646237e-02
## hsa00270 Cysteine and methionine metabolism                         1.857212e-02
## hsa00190 Oxidative phosphorylation                                  1.980619e-02
## hsa00520 Amino sugar and nucleotide sugar metabolism                2.211052e-02
## hsa00860 Porphyrin and chlorophyll metabolism                       2.254043e-02
## hsa00980 Metabolism of xenobiotics by cytochrome P450               2.422418e-02
## hsa00040 Pentose and glucuronate interconversions                   2.471554e-02
## hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series 2.925498e-02
## hsa04141 Protein processing in endoplasmic reticulum                3.197912e-02
## hsa04970 Salivary secretion                                         3.198568e-02
##                                                                            q.val
## hsa04110 Cell cycle                                                 0.0003479008
## hsa04650 Natural killer cell mediated cytotoxicity                  0.0094725450
## hsa04612 Antigen processing and presentation                        0.0094725450
## hsa03050 Proteasome                                                 0.0555608450
## hsa03030 DNA replication                                            0.0555608450
## hsa04145 Phagosome                                                  0.1415433443
## hsa03008 Ribosome biogenesis in eukaryotes                          0.1615793937
## hsa03013 RNA transport                                              0.2029152342
## hsa04062 Chemokine signaling pathway                                0.2029152342
## hsa04514 Cell adhesion molecules (CAMs)                             0.2029152342
## hsa04114 Oocyte meiosis                                             0.2340181770
## hsa00270 Cysteine and methionine metabolism                         0.2340181770
## hsa00190 Oxidative phosphorylation                                  0.2340181770
## hsa00520 Amino sugar and nucleotide sugar metabolism                0.2340181770
## hsa00860 Porphyrin and chlorophyll metabolism                       0.2340181770
## hsa00980 Metabolism of xenobiotics by cytochrome P450               0.2340181770
## hsa00040 Pentose and glucuronate interconversions                   0.2340181770
## hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series 0.2340181770
## hsa04141 Protein processing in endoplasmic reticulum                0.2340181770
## hsa04970 Salivary secretion                                         0.2340181770
##                                                                     set.size
## hsa04110 Cell cycle                                                      124
## hsa04650 Natural killer cell mediated cytotoxicity                       131
## hsa04612 Antigen processing and presentation                              68
## hsa03050 Proteasome                                                       44
## hsa03030 DNA replication                                                  36
## hsa04145 Phagosome                                                       149
## hsa03008 Ribosome biogenesis in eukaryotes                                73
## hsa03013 RNA transport                                                   150
## hsa04062 Chemokine signaling pathway                                     186
## hsa04514 Cell adhesion molecules (CAMs)                                  129
## hsa04114 Oocyte meiosis                                                  112
## hsa00270 Cysteine and methionine metabolism                               36
## hsa00190 Oxidative phosphorylation                                       132
## hsa00520 Amino sugar and nucleotide sugar metabolism                      48
## hsa00860 Porphyrin and chlorophyll metabolism                             41
## hsa00980 Metabolism of xenobiotics by cytochrome P450                     68
## hsa00040 Pentose and glucuronate interconversions                         30
## hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series       26
## hsa04141 Protein processing in endoplasmic reticulum                     165
## hsa04970 Salivary secretion                                               89
##                                                                             exp1
## hsa04110 Cell cycle                                                 2.121346e-06
## hsa04650 Natural killer cell mediated cytotoxicity                  1.408658e-04
## hsa04612 Antigen processing and presentation                        1.732783e-04
## hsa03050 Proteasome                                                 1.503714e-03
## hsa03030 DNA replication                                            1.693928e-03
## hsa04145 Phagosome                                                  5.178415e-03
## hsa03008 Ribosome biogenesis in eukaryotes                          6.896681e-03
## hsa03013 RNA transport                                              1.063035e-02
## hsa04062 Chemokine signaling pathway                                1.162438e-02
## hsa04514 Cell adhesion molecules (CAMs)                             1.237288e-02
## hsa04114 Oocyte meiosis                                             1.646237e-02
## hsa00270 Cysteine and methionine metabolism                         1.857212e-02
## hsa00190 Oxidative phosphorylation                                  1.980619e-02
## hsa00520 Amino sugar and nucleotide sugar metabolism                2.211052e-02
## hsa00860 Porphyrin and chlorophyll metabolism                       2.254043e-02
## hsa00980 Metabolism of xenobiotics by cytochrome P450               2.422418e-02
## hsa00040 Pentose and glucuronate interconversions                   2.471554e-02
## hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series 2.925498e-02
## hsa04141 Protein processing in endoplasmic reticulum                3.197912e-02
## hsa04970 Salivary secretion                                         3.198568e-02
```

19 pathways were identified as significantly upregulated in cluster 2 compared to cluster 3. Some notable clusters include the cell cycle pathway, DNA replication, RNA transport, and certain immune cell pathways.