There are some errors when I tried to knit the document, this file are not knitted successfully, so I just manually add them on here

---
title: "Pathway Analysis with Dataset Normalized Without Condition"
output: pdf_document
---

### Reading and Cleaning Data

```{r}
setwd("C:\\Users\\COLORFUL\\Documents\\bmeg310-1")

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,]
```

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

```{r}
#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$Tumor_Sample_Barcode)
colnames(mutation.patients) <- "Patient_ID"
mutation.patients$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]

rownames(seq.data) <- substr(rownames(seq.data),1 , 15)
```

Now clinical.data, mutation.data, and seq.data all contain common patients across all data sets.

```{r}
#BiocManager::install("DESeq2")
#install.packages("pheatmap")
#install.packages("ggplot2")
#BiocManager::install("AnnotationDbi")
#BiocManager::install("org.Hs.eg.db")
#BiocManager::install("pathview")
#BiocManager::install("gage")
#BiocManager::install("EnhancedVolcano")
library(DESeq2)
library(dplyr)
library(ComplexHeatmap)
library(ggplot2)
library(EnhancedVolcano)

dds <- DESeqDataSetFromMatrix(
```

```
    countData = seq.data,
    colData = clinical.data,
    design = ~ 1
)

#Normalization
dds <- estimateSizeFactors(dds)
normalized_counts <- counts(dds, normalized = TRUE)
log_norm_counts <- log2(normalized_counts + 1)

#Find most variable genes
gene_variance <- apply(log_norm_counts, 1, var)

# Select top 1000 most variable genes
top_genes <- names(sort(gene_variance, decreasing = TRUE)[1:1000])
filtered_data <- log_norm_counts[top_genes, ]

#Clustering
dist_matrix <- dist(t(filtered_data))
hclust_results <- hclust(dist_matrix, method = "ward.D2")
plot(hclust_results, labels = FALSE)



exp_heatmap <- Heatmap(
    filtered_data,
    name = "Expression",
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    show_column_names = FALSE,
    show_row_names = FALSE,
    show_row_dend = FALSE,
    show_column_dend = FALSE,
    heatmap_legend_param = list(
        title_gp = gpar(fontsize = 8), # Font size for the legend title
        labels_gp = gpar(fontsize = 6) # Font size for the legend labels
    )
)

png("exp_heatmap.png", width = 5, height =3, units = "in", res = 300)
draw(
    exp_heatmap,
    annotation_legend_side = "right",
    padding = unit(c(0.3, 0.3, 0.8, 0.3), "cm") # Add padding around the plot
```

```
)
grid.text(
  "Expression Heatmap of Top 1000 Most Variable Genes",
  y = unit(0.94, "npc"),
  gp = gpar(fontsize = 10)
)
dev.off()


```

```{r}
# Add cluster information to clinical data
cut_clusters <- cutree(hclust_results, k = 4)
clinical.data$Cluster <- cut_clusters[match(clinical.data$PATIENT_ID, names(cut_clusters))]

cluster1 <- clinical.data[clinical.data$Cluster == 1,]
cluster2 <- clinical.data[clinical.data$Cluster == 2,]
cluster3 <- clinical.data[clinical.data$Cluster == 3,]
cluster4 <- clinical.data[clinical.data$Cluster == 4,]

pca_res <- prcomp(t(filtered_data))
score <- pca_res$x

score = as.data.frame(score)
score$color <- clinical.data$Cluster[match(rownames(score), clinical.data$PATIENT_ID)]

table(score$color)

pca_plt <- ggplot(score, aes(x=PC1, y=PC2, color=factor(color))) +
  geom_point(size = 3) +
  #scale_color_manual(values = c("red", "blue", "green", "orange", "purple"),
                     #labels = c("Cluster 1", "Cluster 2", "Clutser 3", "Clutser 4")) +
  labs(title = "Plot of Top 2 PCA's",color = "Cluster")+
  theme(plot.title = element_text(size = 20))

pca_plt

ggsave("PCA.png", pca_plt, width = 9, height = 6)

```
```
```

```{r}
library("TCGAbiolinks")
library("survival")
library("survminer")
library("SummarizedExperiment")
library(gridExtra)

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

Surv(clinical.data$OS_MONTHS, clinical.data$deceased) ~ Cluster
fit_new = survfit(Surv(OS_MONTHS, deceased) ~ Cluster, data=clinical.data)
clus_exp <- ggsurvplot(fit_new, data=clinical.data, pval=T, risk.table=T, risk.table.col="strata",
risk.table.height=0.35, title = "Survival Analysis of Clusters", xlab = "Time (Months)")

clus_exp$plot <- clus_exp$plot +
  theme(
    plot.title = element_text(size = 21),    # Title size
    legend.text = element_text(size = 12),                    # Legend text size
    legend.title = element_text(size = 12)                    # Legend title size
  ) +
  labs(color = NULL, fill = NULL, linetype = NULL)

  clus_exp$table <- clus_exp$table +
  theme(legend.position = "none")

combined_plot_exp <- grid.arrange(clus_exp$plot, clus_exp$table, ncol = 1, heights = c(2, 1))

ggsave("clus_exp.png", combined_plot_exp, width = 8, height = 6)

```

```{r}
# Expression analysis on all genes with clusters as conditon

clinical.data$Cluster <- factor(clinical.data$Cluster, levels = c(2, 1, 3, 4))

levels(clinical.data$Cluster)

dds_clus <- DESeqDataSetFromMatrix(
  countData = seq.data,
  colData = clinical.data,
  design = ~ Cluster
)
```

```
dds_clus <- DESeq(dds_clus)

norm_count_clus <- counts(dds_clus, normalized = TRUE)

log_norm_counts_clus <- log2(norm_count_clus + 1)

#Find most variable genes
gene_variance_clus <- apply(log_norm_counts_clus, 1, var)

# Select top 1000 most variable genes
top_genes_clus <- names(sort(gene_variance_clus, decreasing = TRUE)[1:1000])

res <- results(dds_clus, contrast = c("Cluster", "4","1"))

resultsNames(dds_clus)

# Sort results by adjusted p-value (FDR)
res <- res[order(res$padj), ]

summary(res)

res.05 <- results(dds_clus, alpha = 0.05)
table(res.05$padj < 0.05)

resLFC1 <- results(dds_clus, lfcThreshold=1)
table(resLFC1$padj < 0.1)

res.order <- res[order(res$pvalue),]
summary(res.order)

# View significant DE genes
sig_genes <- sort(subset(res, padj < 0.05), decreasing = TRUE)[1:1000,]
head(sig_genes)

plotCounts(dds_clus, gene=which.min(res$padj), intgroup="Cluster")
```

```{r}
library(EnhancedVolcano)

EnhancedVolcano(
   res,
   lab = rownames(res),
   x = "log2FoldChange",
```

```
    y = "padj",
    pCutoff = 0.05,
    FCcutoff = 1,
    title = "DE Analysis: Cluster 1 vs Cluster 2",
    subtitle = "Differentially Expressed Genes"
)

```

```{r}
de_gene_counts <- log_norm_counts[rownames(sig_genes), ]

Heatmap(
    de_gene_counts,
    name = "Expression",
    cluster_rows = FALSE,
    cluster_columns = TRUE,
    show_column_names = FALSE,
    show_row_names = FALSE
)
```

```{r}

resSig <- subset(res, padj < 0.05)
# Get the indices for top 20 upregulated and downregulated genes
genes.top.upreg <- order(resSig$log2FoldChange,decreasing = TRUE)[1:20]
genes.top.downreg <- order(resSig$log2FoldChange,decreasing = FALSE)[1:20]

# Bind the two lists of genes into a single vector
genes.top <- c(genes.top.upreg, genes.top.downreg)


# Variance stabilizing transformation
vsd <- vst(dds_clus)

pca_res <- prcomp(t(assay(vsd)), scale. = TRUE)
score <- pca_res$x

score = as.data.frame(score)
score$color <- as.factor(clinical.data$Cluster)


ggplot(score, aes(x=PC1, y=PC2,    color=color)) +
```

```
  geom_point(size = 4)
```
```{r}
sampleDists = dist(t(assay(vsd)),upper = TRUE)

annot_col = data.frame(clinical.data$Cluster)
row.names(annot_col) <- rownames(clinical.data)

sampleDistMatrix = as.matrix( sampleDists )
rownames(sampleDistMatrix) = colnames(seq.data)
colnames(sampleDistMatrix) = colnames(seq.data)

pheatmap(sampleDistMatrix,
         clustering_distance_rows = sampleDists,
         clustering_distance_cols = sampleDists,
         cluster_rows=FALSE, show_rownames=FALSE,
         show_colnames = FALSE,
         cluster_cols=FALSE,
         annotation_col=annot_col)
```

```{r}
mutation.patients <- mutation.data[mutation.data$Hugo_Symbol %in% c("TP53", "PIK3CA", "TTN"), ]

mutation.patients$Cluster_gene <- ifelse(mutation.patients$Hugo_Symbol == "TP53", 1,
                                         ifelse(mutation.patients$Hugo_Symbol == "TTN", 2,
                                         ifelse(mutation.patients$Hugo_Symbol == "PIK3CA", 3, "Other")))

common <- intersect(mutation.patients$Patient_ID, clinical.data$PATIENT_ID)

# Use aggregate to ensure one patient can be assigned a single cluster
patient_clusters <- aggregate(Cluster_gene ~ Patient_ID, data = mutation.patients, FUN = function(x) x[1])

# Merge clusters into clinical data
clinical.data <- merge(clinical.data, patient_clusters, by.x = "PATIENT_ID", by.y = "Patient_ID", all.x = TRUE)

# Fill non-mutated patients with a default cluster, e.g., 3
clinical.data$Cluster_gene[is.na(clinical.data$Cluster_gene)] <- "Other"
```

```
clinical.data$Patient_Status <- ifelse(clinical.data$PATIENT_ID %in% common, 1, 2)

clinical.data$tp <- clinical.data$PATIENT_ID[ifelse((clinical.data$PATIENT_ID), 2, 1)] # Change groups

table(clinical.data$Patient_Status)

genes <- as.data.frame(mutation.patients$Gene)
```

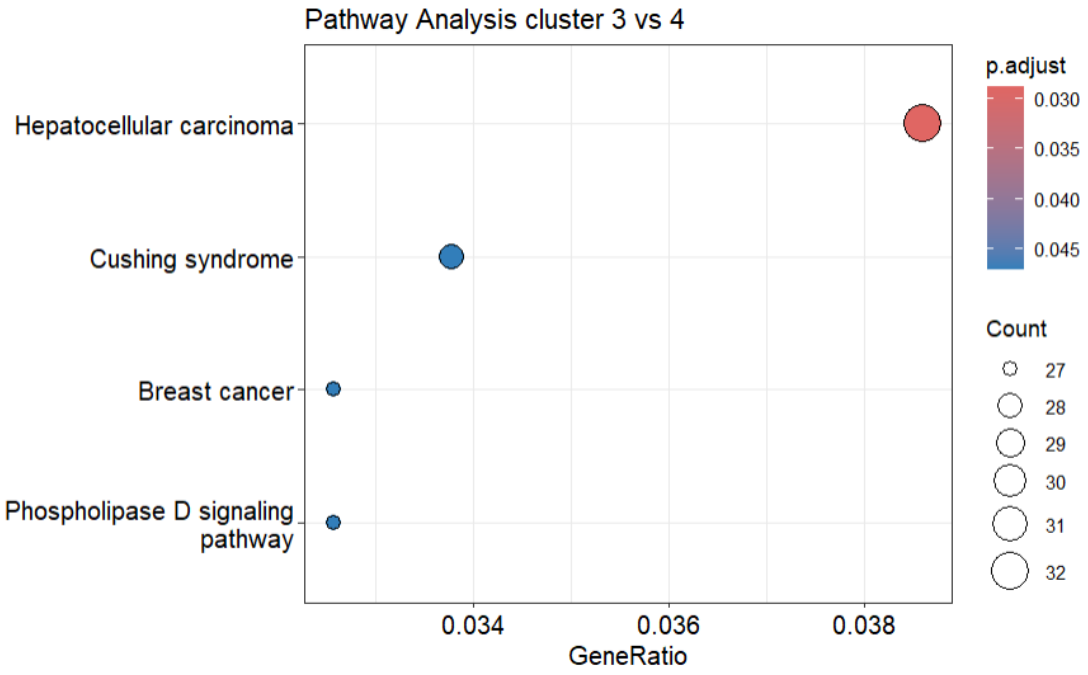
```{r}
res_Cluster34 <- results(dds_clus, contrast = c("Cluster", "3", "4"))   #Compare Cluster 3 and Cluster 4

sig_genes34 <- subset(res_Cluster34, padj < 0.05)

#pathway analysis
gene_list34 <- rownames(sig_genes34)
mapped_genes34 <- mapIds(
   org.Hs.eg.db,
   keys = gene_list34,
   column = "ENTREZID",
   keytype = "ENSEMBL",
   multiVals = "first")
mapped_genes34 <- na.omit(mapped_genes34)

kegg_results34 <- enrichKEGG(
   gene = mapped_genes34,
   organism = "hsa",
   keyType = "kegg",
   pvalueCutoff = 0.05
)

dotplot(kegg_results34, showCategory = 10) + ggtitle("Pathway Analysis cluster 3 vs 4")
```


Pathway Analysis cluster 3 vs 4
Hepatocellular carcinoma
Cushing syndrome
Breast cancer
Phospholipase D signaling pathway
0.034
0.036
0.038
GeneRatio
p.adjust
0.030
0.035
0.040
0.045
Count
27
28
29
30
31
32