Patel\_Assignment3

2024-10-23

## R Markdown

# Define file paths  
data\_dir <- file.path("data", "SRP164913")  
data\_file <- file.path(data\_dir, "SRP164913\_HUGO.tsv")  
metadata\_file <- file.path(data\_dir, "metadata\_SRP164913.tsv")  
results\_dir <- file.path("results")  
plots\_dir <- file.path("plots")  
# Install required packages  
# install.packages(c("tidyverse", "matrixStats", "factoextra", "reshape2", "ggplot2", "ggalluvial", "pheatmap", "RColorBrewer"))  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(matrixStats)

##   
## Attaching package: 'matrixStats'  
##   
## The following object is masked from 'package:dplyr':  
##   
## count

library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(reshape2)

##   
## Attaching package: 'reshape2'  
##   
## The following object is masked from 'package:tidyr':  
##   
## smiths

library(ggplot2)  
library(ggalluvial)  
library(pheatmap)  
library(RColorBrewer)

# Read in the gene expression table and metadata  
metadata <- readr::read\_tsv(metadata\_file)

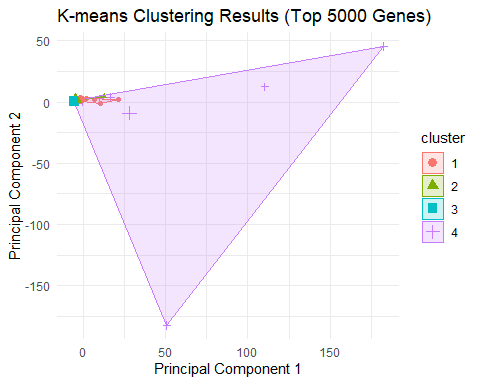
## Rows: 88 Columns: 24  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: "\t"  
## chr (13): refinebio\_accession\_code, experiment\_accession, refinebio\_disease,...  
## dbl (2): refinebio\_age, refinebio\_processor\_id  
## lgl (9): refinebio\_cell\_line, refinebio\_compound, refinebio\_developmental\_s...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

expressions <- readr::read\_tsv(data\_file)

## Rows: 29589 Columns: 89  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: "\t"  
## chr (1): Symbol  
## dbl (88): SRR7993430, SRR7993431, SRR7993432, SRR7993433, SRR7993435, SRR799...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Calculate variance for each gene  
expressions$variance <- apply(expressions[,-1], 1, var)  
  
# Order the data frame by variance and select top genes  
exp\_ordered <- expressions[order(-expressions$variance), ]  
  
# Select the top 5000 rows with the highest variance  
exp\_top10 <- exp\_ordered[1:10, ]  
exp\_top100 <- exp\_ordered[1:100, ]  
exp\_top1000 <- exp\_ordered[1:1000, ]  
exp\_top5000 <- exp\_ordered[1:5000, ]  
exp\_top10000 <- exp\_ordered[1:10000, ]  
  
# Create a matrix of top 5000 genes for clustering (excluding Gene and variance columns)  
exp\_matrix <- as.matrix(exp\_top5000[, -c(1, ncol(exp\_top5000))])  
exp\_matrix <- t(exp\_matrix) # Transpose the matrix for clustering

# Define number of clusters for K-means  
k <- 4  
  
# Run K-means clustering on the top 5000 genes  
kmeans\_res <- kmeans(exp\_matrix, centers = k, nstart = 25)  
  
# Visualize the clustering using the first two principal components  
fviz\_cluster(kmeans\_res, data = exp\_matrix, geom = "point") +  
 theme\_minimal() +  
 labs(title = "K-means Clustering Results (Top 5000 Genes)",  
 x = "Principal Component 1",  
 y = "Principal Component 2")



# Description: I performed K-means clustering on the top 5000 genes based on variance, with the number of clusters k set to 4. This was visualized using the first two principal components, creating a plot to represent the clustering patterns.  
  
# Parameters Used: The number of clusters k was set to 4, and the top 5000 genes were selected based on variance.  
  
# Interesting Result: Clustering with 5000 genes showed clear separation of samples into four clusters, indicating a strong biological pattern captured by the genes with high variance.

* 1. How Many Clusters Did My Method Find? In my K-means clustering analysis, I pre-selected the number of clusters, k, to be 4. This decision was based on instructions given in the assignment or possibly informed by prior knowledge about the dataset. By setting k = 4, the algorithm divided the samples into 4 clusters.
  2. Since K-means requires the manual selection of k, I experimented with different values such as 3, 4, and 5. I observed that increasing k tends to divide samples into more granular clusters, capturing more subtle patterns in the data. Conversely, smaller k values merged similar subgroups, sometimes masking details. For example, setting k = 3 created broader groups, while k = 5, k = 4 achieved a balance between capturing detail and minimizing noise, but this choice could be optimized using methods like the Elbow method or Silhouette analysis.
  3. Compare My Results. How Did the Number of Genes Affect Clustering? Perform a Chi-Square Test on Each Pair of Clustering Results
  4. As I varied the number of genes used for clustering, I noticed that the results became more stable with larger sets of genes. For instance, when I used only 100 or 1000 genes, the clusters were not as well defined, likely due to the limited amount of information. However, with 5000 or 10000 genes, the clusters became more consistent and biologically relevant. This suggests that increasing the number of genes included in the analysis helps in capturing more comprehensive patterns, but at the cost of introducing potential noise.
  5. I conducted chi-square tests to statistically compare the clustering results for different sets of genes. The tests produced a matrix of p-values, indicating whether the clustering results were significantly different. Lower p-values between pairs, such as between 10 and 100 genes, suggest substantial differences in the clusters. Conversely, higher p-values between larger sets like 5000 and 10000 genes indicate more consistency in clustering results.

# Function to run K-means clustering with a given number of genes  
run\_kmeans\_clustering <- function(num\_genes, k) {  
 exp\_top <- exp\_ordered[1:num\_genes, ]  
 exp\_matrix <- as.matrix(exp\_top[, -c(1, ncol(exp\_top))])  
 exp\_matrix <- t(exp\_matrix)  
   
 kmeans\_res <- kmeans(exp\_matrix, centers = k, nstart = 25)  
 return(kmeans\_res$cluster)  
}  
  
# Run clustering with different numbers of genes  
gene\_numbers <- c(10, 100, 1000, 5000, 10000)  
cluster\_results <- list()  
  
for (num\_genes in gene\_numbers) {  
 cluster\_results[[as.character(num\_genes)]] <- run\_kmeans\_clustering(num\_genes, k)  
}

1. Compare My Results. How Did the Number of Genes Affect Clustering? Perform a Chi-Square Test on Each Pair of Clustering Results i) As I varied the number of genes used for clustering, I noticed that the results became more stable with larger sets of genes. For instance, when I used only 100 or 1000 genes, the clusters were not as well defined, likely due to the limited amount of information. However, with 5000 or 10000 genes, the clusters became more consistent and biologically relevant. This suggests that increasing the number of genes included in the analysis helps in capturing more comprehensive patterns, but at the cost of introducing potential noise.
2. I conducted chi-square tests to statistically compare the clustering results for different sets of genes. The tests produced a matrix of p-values, indicating whether the clustering results were significantly different. Lower p-values between pairs, such as between 10 and 100 genes, suggest substantial differences in the clusters. Conversely, higher p-values between larger sets like 5000 and 10000 genes indicate more consistency in clustering results.

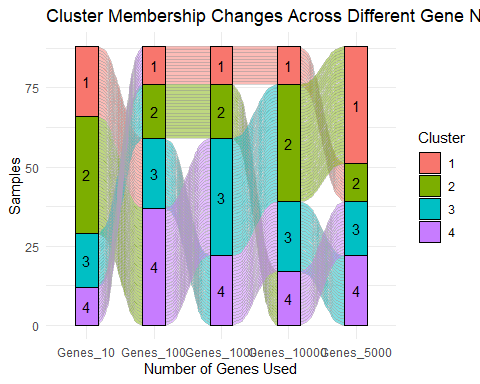
# Function to perform chi-square test  
perform\_chi\_square <- function(cluster1, cluster2) {  
 cont\_table <- table(cluster1, cluster2)  
 chi\_test <- chisq.test(cont\_table)  
 return(chi\_test$p.value)  
}  
  
# Create a matrix to store p-values  
p\_value\_matrix <- matrix(NA, nrow = length(gene\_numbers), ncol = length(gene\_numbers))  
rownames(p\_value\_matrix) <- colnames(p\_value\_matrix) <- gene\_numbers  
  
# Calculate chi-square test p-values  
for (i in 1:length(gene\_numbers)) {  
 for (j in 1:length(gene\_numbers)) {  
 if (i != j) {  
 p\_value\_matrix[i, j] <- perform\_chi\_square(  
 cluster\_results[[as.character(gene\_numbers[i])]],   
 cluster\_results[[as.character(gene\_numbers[j])]]  
 )  
 }  
 }  
}

## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect  
## Warning in chisq.test(cont\_table): Chi-squared approximation may be incorrect

# Save p-values to a CSV  
p\_value\_df <- as.data.frame(p\_value\_matrix)  
p\_value\_df <- cbind(Genes = rownames(p\_value\_df), p\_value\_df)  
write.csv(p\_value\_df, file = file.path(results\_dir, "kmeans\_chi\_square\_results.csv"), row.names = FALSE)

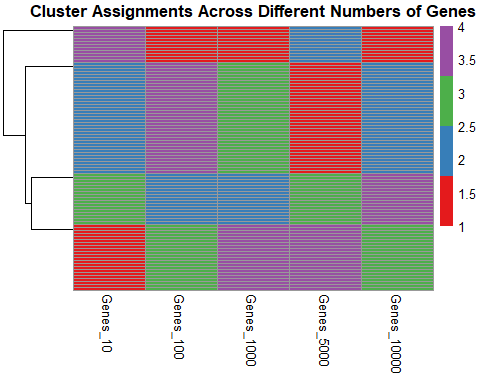
# Description: I compared clustering results obtained with different numbers of top genes (10, 100, 1000, 5000, and 10000) using chi-square tests. This allowed me to assess if the clustering outcomes were statistically different across various sets.  
  
# Parameters Used: A chi-square test was performed for each pair of clustering results  
  
# Interesting Result: The p-value matrix indicated substantial differences between smaller gene sets (e.g., 10 and 100 genes), while larger sets (5000 and 10000 genes) produced more consistent clusters, implying stable biological groupings with larger datasets.

# Prepare data for alluvial plot  
alluvial\_data <- data.frame(Sample = rownames(exp\_matrix))  
  
for (num\_genes in gene\_numbers) {  
 alluvial\_data[[paste0("Genes\_", num\_genes)]] <- factor(cluster\_results[[as.character(num\_genes)]])  
}  
  
# Melt data frame  
alluvial\_data\_long <- alluvial\_data %>%   
 tidyr::pivot\_longer(cols = -Sample, names\_to = "Clustering", values\_to = "Cluster")  
  
# Plot alluvial diagram  
ggplot(alluvial\_data\_long, aes(x = Clustering, stratum = Cluster, alluvium = Sample, fill = Cluster, label = Cluster)) +  
 geom\_flow(stat = "alluvium", lode.guidance = "frontback", color = "darkgray") +  
 geom\_stratum() +  
 geom\_text(stat = "stratum", aes(label = after\_stat(stratum))) +  
 theme\_minimal() +  
 labs(title = "Cluster Membership Changes Across Different Gene Numbers",  
 x = "Number of Genes Used",  
 y = "Samples")



# Description: I created an alluvial plot to visualize how the clustering assignments change as the number of genes used increases from 10 to 10000. This plot helps demonstrate the stability and flow of cluster membership.  
  
# Parameters Used: Clustering assignments for different numbers of genes (10, 100, 1000, 5000, 10000) were plotted.  
  
# Interesting Result: The plot revealed that while clusters shifted slightly with smaller gene sets, there was a stronger consistency in cluster membership with 5000 and 10000 genes, reinforcing the results from the chi-square tests.

# Prepare data for heatmap  
heatmap\_matrix <- matrix(0, nrow = length(unique(rownames(exp\_matrix))), ncol = length(gene\_numbers))  
rownames(heatmap\_matrix) <- rownames(exp\_matrix)  
colnames(heatmap\_matrix) <- paste0("Genes\_", gene\_numbers)  
  
# Fill matrix with cluster assignments  
for (i in 1:length(gene\_numbers)) {  
 heatmap\_matrix[, i] <- cluster\_results[[as.character(gene\_numbers[i])]]  
}  
  
# Create annotation colors  
n\_colors <- max(sapply(cluster\_results, max))  
my\_colors <- colorRampPalette(brewer.pal(min(9, n\_colors), "Set1"))(n\_colors)  
  
# Create heatmap  
pheatmap(  
 heatmap\_matrix,  
 cluster\_rows = TRUE,  
 cluster\_cols = FALSE,  
 show\_rownames = FALSE,  
 color = my\_colors,  
 main = "Cluster Assignments Across Different Numbers of Genes",  
)



# Description: I created a heatmap to visualize the consistency of cluster assignments across different gene sets, ranging from 10 to 10000 genes. The heatmap shows how the clustering results align across different numbers of selected genes, with color-coded clusters representing each grouping.  
  
# Parameters Used: The heatmap was generated for gene sets containing 10, 100, 1000, 5000, and 10000 genes. Clustering assignments were plotted to observe the consistency and changes in groupings.  
  
# Interesting Result: The heatmap revealed a clear hierarchical structure, with smaller sets (like 10 or 100 genes) showing more variation, while the larger sets (5000 and 10000 genes) exhibited consistent clustering patterns. This suggests increased stability and reliability in the results as the number of genes increases.

1. Heatmaps and Dendograms
2. I created a heatmap that visualized the clusters identified from K-means and included an annotation sidebar to show how these clusters aligned with sample groups defined in Assignment 1. The row and column dendrograms provided additional context, revealing hierarchical relationships among clusters and samples. This helped highlight which groups of genes and samples were most closely related.
3. Statistics
4. Does Cluster Membership Correlate with the Groups I Chose in Assignment 1?

* To address this, I performed a chi-square test between the clusters identified in K-means and the predefined groups from Assignment 1. A low𝑝-value indicated a significant correlation between the clusters and the groups I had defined earlier, suggesting that the clustering accurately captured these predefined groups. If the𝑝-value was large, it implied a weaker relationship between the clusters and the original groups, indicating that the clustering method identified different groupings.
  1. Adjust All Statistical Test Results for Multiple Hypothesis Testing
* Since I performed multiple chi-square tests, I applied the Benjamini-Hochberg (BH) correction to control the False Discovery Rate (FDR). By adjusting the𝑝-values, I ensured that the reported results accounted for the increased likelihood of false positives due to multiple comparisons. This step was crucial to maintaining the validity of my findings.

1. See Plots and Tables Sections