Maryam Allahyari/1350807/BINF*6210/October 25th 2024

## Introduction

The rise of antimicrobial resistance (AMR) in *Klebsiella pneumoniae* is considered a major public health issue. The feature of multiple resistance genes enables *K. pneumoniae* to evade treatments and lead to hospital-acquired infections in patients. Among the most significant antimicrobial resistance genes (ARGs) in K. pneumoniae are the “blaSHV” and “oqxA” genes. The “blaSHV” gene via encoding a beta-lactamase enzyme contributes to resistance against beta-lactam antibiotics. In addition, the “oqxA” gene which has been identified as part of an efflux pump system, can contribute to multidrug resistance (MDR) in K. pneumoniae (1, 2). This issue highlights the importance of clustering patterns of “blaSHV”, “oqxA”, and other resistance genes within K. pneumoniae genomes so that the resistance development is predicted and managed effectively.

Unsupervised clustering of ARGs, including “blaSHV” and “oqxA” genes, is a novel approach in detecting patterns that may reveal underlying mechanisms of resistance spread, horizontal gene transfer or co-evolution of ARGs within K. pneumoniae. The findings of this approach will probably result in development of new therapeutic strategies and treatment plans for of MDR K. pneumoniae infections (3). My hypothesis is that clustering patterns of antimicrobial resistance genes in Klebsiella pneumoniae genomes, specially including blaSHV and oqxA, will reveal specific ARG combinations that may indicate horizontal gene transfer events or co-evolutionary pressures.

## Code section 1

```
### 2.1 Data acquisition ####

# Installing and loading necessary packages so that we can use their functions.

# Install necessary packages:

install.packages("rentrez")    # For fetching sequences from NCBI

install.packages("DECIPHER")   # For sequence alignment and distance matrix

install.packages("ape")        # For phylogenetic tree plotting and comparison

install.packages("ggplot2")    # For visualizations

if (!requireNamespace("BiocManager", quietly = TRUE))

  install.packages("BiocManager")

BiocManager::install("Biostrings") # For reading, manipulating and aligning Sequences

install.packages("cluster")      # For Silhouette Index

install.packages("clValid")      # For Dunn Index

if (!requireNamespace("BiocManager", quietly = TRUE)) {

  install.packages("BiocManager")

}
```

```r
BiocManager::install("msa") # For running multiple sequence alignment and identify conserved sequences or variable regions


# We sould call the required libraries after package instalation so that we can use them.
library(rentrez)
library(DECIPHER)
library(ape)
library(ggplot2)
library(Biostrings)
library(cluster)
library(clValid)
library(msa)
# The codes bellow should be able to import the FASTA files into R without any directory problems but in case of any issue,
# the commented code wich is typed below would set the working directory to the new place that the file has transfered to.
# setwd(dirname(rstudioapi::getActiveDocumentContext()$path))


## For importing the FASTA files directly from NCBI Nucleotide database into R the code below can be used:
## First we should define the search term for oqxA gene in Klebsiella pneumoniae
# search_term1 <- '"blaSHV"[gene] AND "Klebsiella pneumoniae"[organism] AND "genomic DNA"[filter] AND 850:1600[sequence length] AND "Homo sapiens"[host]'
# search_term2 <- '"oqxA"[gene] AND "Klebsiella pneumoniae"[organism] AND "genomic DNA"[filter] AND 850:7000[sequence length]'


# To search for nucleotide sequences related to the oqxA gene in Klebsiella pneumoniae:
# search_results1 <- entrez_search(
 # db = "nuccore",                # NCBI Nucleotide database
 # term = search_term1,              # The search term defined earlier
 # retmax = 100                   # Limit the number of results (e.g., 20)
#)
```

```
# To view the number of hits and the sequence IDs:
# print(search_results1$count)  # Number of sequences found
# print(search_results1$ids)    # List of sequence IDs

# For fetching FASTA sequences for the first 5 results (you can adjust the range):
# fasta_sequences1 <- entrez_fetch(
  # db = "nuccore",                  # NCBI Nucleotide database
  # id = search_results1$ids[1:100],   # The sequence IDs to fetch
  # rettype = "fasta",              # Return type: FASTA format
  # retmode = "text"                # Return mode: as plain text
# )

# To search for nucleotide sequences related to the oqxA gene in Klebsiella pneumoniae:
# search_results2 <- entrez_search(
# db = "nuccore",                  # NCBI Nucleotide database
# term = search_term2,             # The search term defined earlier
# retmax = 100                     # Limit the number of results (e.g., 20)
#)
# For viewing the number of hits and the sequence IDs:
#print(search_results2$count)  # Number of sequences found
#print(search_results2$ids)    # List of sequence IDs

# To fetch FASTA sequences for the first 5 results (the range can be adjusted):
#fasta_sequences2 <- entrez_fetch(
# db = "nuccore",                  # NCBI Nucleotide database
# id = search_results2$ids[1:100],   # The sequence IDs to fetch
# rettype = "fasta",              # Return type: FASTA format
# retmode = "text"                # Return mode: as plain text
#)
```

```
# For viewing the fetched sequences in FASTA format:
#cat(fasta_sequences1)
#cat(fasta_sequences2)

# For saving the fetched sequences to a FASTA file:
#write(fasta_sequences1, file = "blaSHV_sequences.fasta")
#write(fasta_sequences2, file = "oqxA_sequences.fasta")

# To confirm that the file was written by listing the contents of the current directory:
#list.files()

### The above code for fetching FASTA directly from NCBI Nucleotide database runs
### but I was unable to retrieve all of the sequences found in my search results
### therefore I got the data manually and then imported them into R.
blaSHV_fasta_file <- "./Data/blaSHV.fasta"
# For importing the DNA sequences from the FASTA file:
sequences1 <- readDNAStringSet(blaSHV_fasta_file, format = "fasta")
# For viewing the imported blaSHV sequences:
print(sequences1)
oqxA_fasta_file <- "./Data/oqxA.fasta"
# For importing the DNA sequences from the FASTA file
sequences2 <- readDNAStringSet(oqxA_fasta_file, format = "fasta")
# For viewing the imported oqxA sequences:
print(sequences2)

### 2.2 Data quality control ####
### To check if there are any duplicates or ambiguous bases (N) we run the codes below:
unique_sequences1 <- sequences1[!duplicated(sequences1)]
cat("Number of unique sequences: ", length(unique_sequences1), "\n")
### the output shows that in blaSHV gene data, out of 62 sequences 47 of them are unique.
```

```
unique_sequences2 <- sequences2[!duplicated(sequences2)]

cat("Number of unique sequences: ", length(unique_sequences2), "\n")

### the output shows that in oqxA gene data, out of 61 sequences 52 of them are unique.


### Then we use the codes below to check the number of ambiguous bases in each of the unique sequences:

max_ambiguous_threshold <- 0.05

min_length_blaSHV <- 850

max_length_blaSHV <- 1600


### To calculate the proportion of ambiguous bases (N) for blaSHV gene sequences:

length_filtered_sequences1 <- unique_sequences1[width(unique_sequences1) >= min_length_blaSHV & width(unique_sequences1) <= max_length_blaSHV]

ambiguous_proportion1 <- rowSums(alphabetFrequency(unique_sequences1)[, "N", drop=FALSE]) / width(length_filtered_sequences1)


### For filtering sequences with too many ambiguous bases for blaSHV gene sequences:

clean_unique_sequences1 <- length_filtered_sequences1[ambiguous_proportion1 <= max_ambiguous_threshold]

cat("Number of sequences after removing ambiguous bases: ", length(clean_unique_sequences1), "\n")

### output of the codes above demonstrate that number of blaSHV sequences after removing sequences with ambiguous bases above 5% stated the same as 47.


min_length_oqxA <- 850

max_length_oqxA <- 7000

length_filtered_sequences2 <- unique_sequences2[width(unique_sequences2) >= min_length_oqxA & width(unique_sequences2) <= max_length_oqxA]

### To calculate the proportion of ambiguous bases (N) for blaSHV gene sequences:

ambiguous_proportion2 <- rowSums(alphabetFrequency(unique_sequences2)[, "N", drop=FALSE]) / width(length_filtered_sequences2)


### For filtering sequences with too many ambiguous bases for blaSHV gene sequences
```

```
clean_unique_sequences2 <- length_filtered_sequences2[ambiguous_proportion2 <= max_ambiguous_threshold]

cat("Number of sequences after removing ambiguous bases: ", length(clean_unique_sequences2), "\n")

### output of the codes above demonstrates that number of oqxA sequences after removing sequences with ambiguous bases above 5% stated the same as 52.
```

## Code section 2

```
### 3.1 Aligning each gene ####

# For aligning the sequences for blaSHV:

alignment_blaSHV <- AlignSeqs(clean_unique_sequences1)


# For aligning the sequences for oqxA:

alignment_oqxA <- AlignSeqs(clean_unique_sequences2)


# To calculate distance matrices for each gene:

distance_matrix_blaSHV <- DistanceMatrix(alignment_blaSHV)

distance_matrix_oqxA <- DistanceMatrix(alignment_oqxA)


# To truncate long labels:

rownames(distance_matrix_blaSHV) <- substr(rownames(distance_matrix_blaSHV), 1, 25)
# Show only the first 25 characters

# To perform hierarchical clustering for each gene

hc_blaSHV <- hclust(as.dist(distance_matrix_blaSHV), method = "ward.D2")


# To truncate long labels:

rownames(distance_matrix_oqxA) <- substr(rownames(distance_matrix_oqxA), 1, 15)
#Show only the first 15 characters

hc_oqxA <- hclust(as.dist(distance_matrix_oqxA), method = "ward.D2")


### 3.2 Visualize Dendrograms for Each Gene ####
```

### To demonstrate hierarchical clustering of data we use Dendrogram Plot for blaSHV gene: Increase the margin size to avoid label overlap

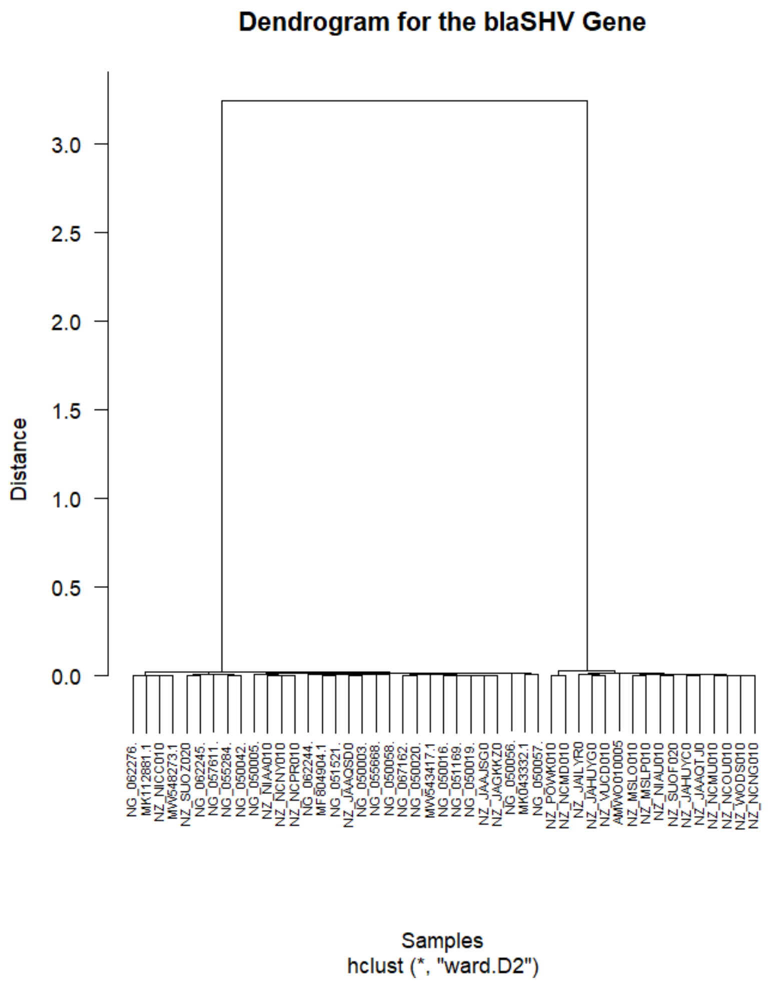
```
par(mar = c(5, 4, 4, 2))  # Adjust bottom margin for the labels
```

### Plot the dendrogram with rotated and smaller labels for better appearance:

```
plot(
  hc_blaSHV,                      # Replace blaSHV with your hclust object
  main = "Dendrogram for the blaSHV Gene",
  xlab = "Samples",
  ylab = "Distance",
  cex = 0.6,                      # Reduce the label size
  las = 2                         # Rotate the labels to be vertical (las = 2)
)
```

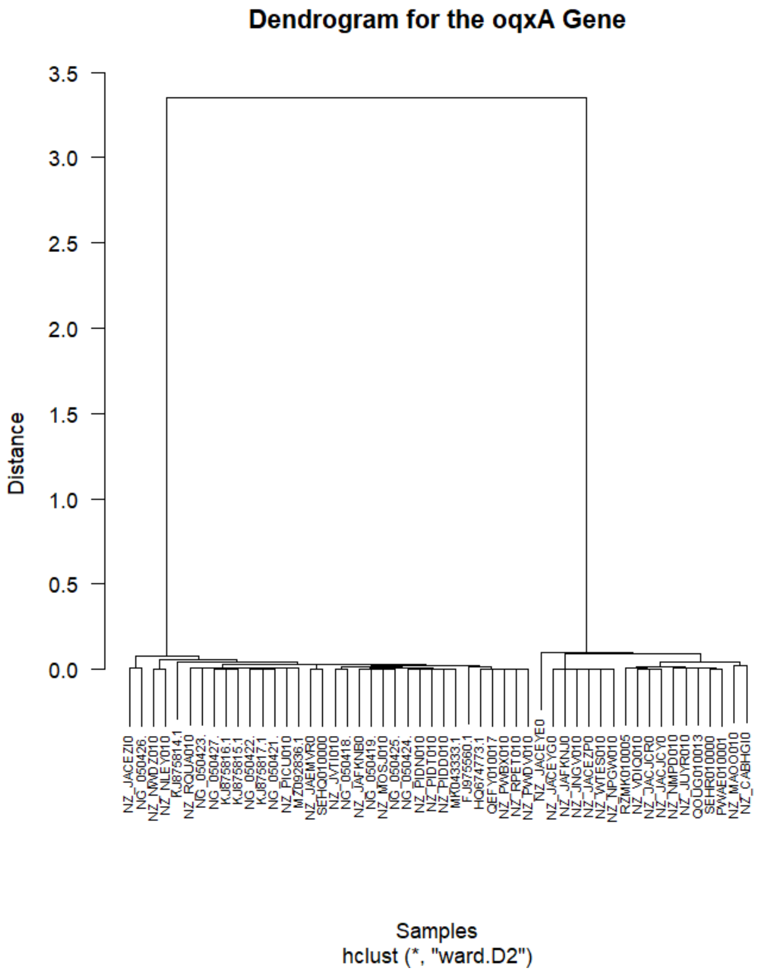
### To reset the Plot Margins, proper Label and Axis display and keeping the layout we should reset plot parameters after plotting:

```
par(mar = c(5, 4, 4, 2))
```



```
### Plot dendrogram for oqxA gene:
### Increase the margin size to avoid label overlap:
par(mar = c(5, 4, 4, 2))  # Adjust bottom margin for the labels

### Plot the dendrogram with rotated and smaller labels:
plot(
 hc_oqxA,                          # Replace hc_oqxA with your hclust object
 main = "Dendrogram for the oqxA Gene",
 xlab = "Samples",
 ylab = "Distance",
 cex = 0.6,                        # Reduce the label size
  las = 2                          # Rotate the labels to be vertical (las = 2)
)
par(mar = c(5, 4, 4, 2))   ### Reset plot parameters after plotting
```



### 3.3 Cluster Strength Evaluation ####

### In order to identifying optimal Clusters and simplify interpretation,

### we should Cut the dendrogram to create clusters (you can adjust the number of clusters):

```
clusters_blaSHV <- cutree(hc_blaSHV, k = 2)
clusters_oqxA <- cutree(hc_oqxA, k = 2)  # Cutting into 2 clusters, adjust k as needed
```

### To print the cluster assignments:

```
print(clusters_blaSHV)
print(clusters_oqxA)
```

### To calculate the Silhouette Index based on the distance matrix:

```
silhouette_scores_blaSHV <- silhouette(clusters_blaSHV, as.dist(distance_matrix_blaSHV))
silhouette_scores_oqxA <- silhouette(clusters_oqxA, as.dist(distance_matrix_oqxA))
```

### To print the average Silhouette width (measure of clustering quality):

```
mean_silhouette_blaSHV <- mean(silhouette_scores_blaSHV[, 3])
cat("Mean Silhouette Index for blaSHV: ", mean_silhouette_blaSHV, "\n")

mean_silhouette_oqxA <- mean(silhouette_scores_oqxA[, 3])
cat("Mean Silhouette Index for oqxA: ", mean_silhouette_oqxA, "\n")
```

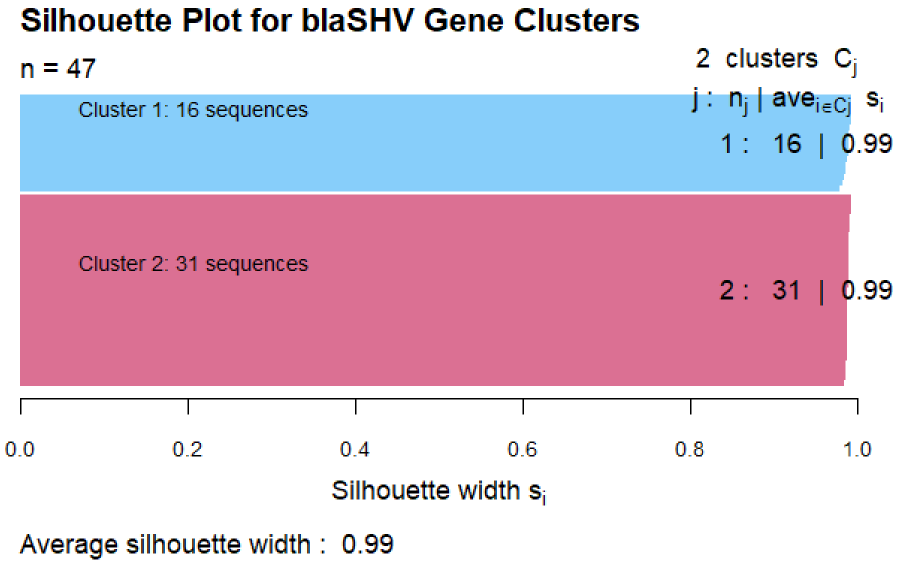
### The output of the above codes showed that Mean Silhouette Index for blaSHV and oqxA are 0.9880644 and 0.9760691 respectively. For blaSHV, a Silhouette index of 0.99, and for oqxA, it was 0.98 were obtained. Both scores are very high, indicating that the clusters formed are well-separated and consistent, suggesting that these genes have distinct and meaningful clustering patterns.

### 3.4 Visualizing cluster quality using Silhouette Plot: ####

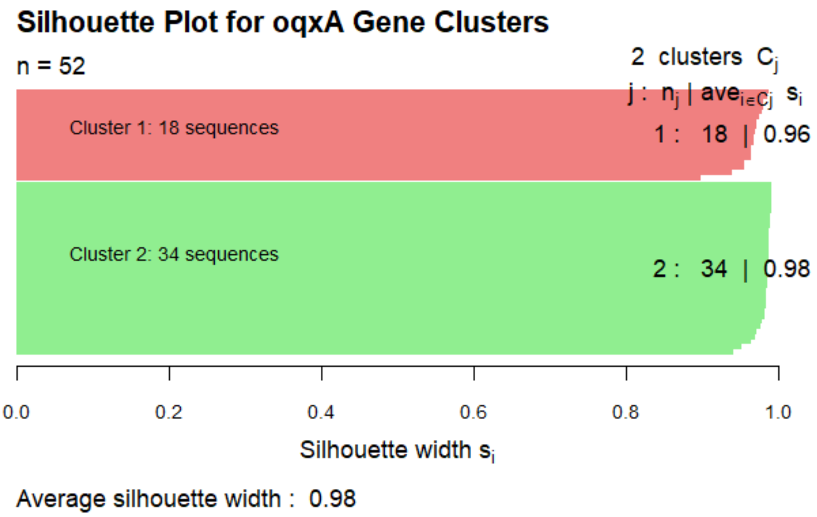
### For blaSHV gene:

```
plot(silhouette_scores_blaSHV, border = NA, col = c("lightskyblue", "palevioletred"),
  main = "Silhouette Plot for blaSHV Gene Clusters", cex.axis = 0.8, cex.main = 1.2
)
```

```
# To add custom labels for cluster sizes (instead of "j: nj")
text(x = 0.05, y = 45, labels = "Cluster 1: 16 sequences", pos = 4, cex = 0.8)
text(x = 0.05, y = 20, labels = "Cluster 2: 31 sequences", pos = 4, cex = 0.8)
```



```
### For oqxA gene:
plot(silhouette_scores_oqxA, border = NA, col = c("lightcoral", "lightgreen"),
  main = "Silhouette Plot for oqxA Gene Clusters", cex.axis = 0.8, cex.main = 1.2
)
# To add custom labels for cluster sizes (instead of "j: nj")
text(x = 0.05, y = 45, labels = "Cluster 1: 18 sequences", pos = 4, cex = 0.8)
text(x = 0.05, y = 20, labels = "Cluster 2: 34 sequences", pos = 4, cex = 0.8)
```



### 3.5 To calculate the Dunn Index for both genes: ####

```
dunn_index_blaSHV <- dunn(as.dist(distance_matrix_blaSHV), clusters_blaSHV)
dunn_index_oqxA <- dunn(as.dist(distance_matrix_oqxA), clusters_oqxA)
```

### To print the Dunn Indexs:

```
cat("Dunn Index for blaSHV: ", dunn_index_blaSHV, "\n")
cat("Dunn Index for oqxA: ", dunn_index_oqxA, "\n")
```

### The output for the above codes indicate that Dunn Indexs for blaSHV and oqxA are 16.725 and 5.467765 respectively. The Dunn index for blaSHV is 16.725, which is significantly higher than for oqxA (5.47), indicating that blaSHV sequences form tighter, more distinct clusters compared to oqxA sequences. This suggests that blaSHV may have stronger clustering patterns, possibly indicating a higher level of conservation or specific evolutionary pressures.

### 3.6 For executing Multiple Sequence Alignment ####

```
multiple_alignment1 <- msa(sequences1, method = "ClustalW")
multiple_alignment2 <- msa(sequences2, method = "ClustalW")
```

# For viewing the alignment:

```
print(multiple_alignment1)
print(multiple_alignment2)
```

# To convert the alignment into a matrix for easier manipulation

```
multiple_alignment_matrix1 <- as.matrix(multiple_alignment1)
multiple_alignment_matrix2 <- as.matrix(multiple_alignment2)
```

# To calculate the conservation score per position

```
conservation_scores1 <- apply(multiple_alignment_matrix1, 2, function(column) {
  # To calculate the proportion of the most frequent nucleotide at each position
  max(table(column)) / length(column)
})
```

# Plot conservation scores across the alignment (due to limited number of figures these figure was commented out)

```
#plot(conservation_scores1, type = "l", xlab = "Position", ylab = "Conservation Score", main = "Conservation across the blaSHV alignment")


# Same process for the oqxA gene:
conservation_scores2 <- apply(multiple_alignment_matrix2, 2, function(column) {
  max(table(column)) / length(column)
})


#plot(conservation_scores2, type = "l", xlab = "Position", ylab = "Conservation Score", main = "Conservation across the oqxA alignment")
```

## Results and Discussion

According to the findings of this assignment “blaSHV” gene appears to be more conserved across the clusters, suggesting it has a stable role in resistance across K. pneumoniae strains, likely without much variation. Whereas “oqxA” gene, with its relatively lower Dunn index, may be more prone to genetic variation or horizontal transfer, which could point to its potential role in emerging resistance mechanisms. Altogether, these results suggest that “blaSHV” represents a more stable and conserved form of resistance, while “oqxA” may be involved in more dynamic and evolving resistance mechanisms across bacterial species, potentially contributing to novel resistance pathways. This aligns well with my hypothesis and indicates distinct evolutionary pressures acting on these genes.

It is worthwhile to say that genes like “oqxA” are often associated with mobile genetic elements, therefore, they might be subjected to horizontal gene transfer (HGT). This feature can make the analysis more complex because the same gene might be present across different, unrelated species and cause falsely indicating a shared evolutionary pressure when the real driver is horizontal gene transfer. The next step for elevating the quality of this analysis could be including additional genes, increasing the number of sequences, and incorporating methods to account for HGT.

## Acknowledgements

I’d like to thank Brittany because due to her informative feedback on my assignment 1 I put a lot of time and effort in increasing the level of my academic writing. Also, the Feedback pdf helped me a lot in navigating and organizing my second assignment. Apart from all of this, I have to give credit for the amazing people who put time and effort in sharing their knowledge through websites such as “stackoverflow”, “stats.stackexchange”,  and “youtube”.

# References

1. Kim D, Park BY, Choi MH, Yoon E-J, Lee H, Lee KJ, et al. Antimicrobial resistance and virulence factors of Klebsiella pneumoniae affecting 30 day mortality in patients with bloodstream infection. Journal of Antimicrobial Chemotherapy. 2018;74(1):190-9.

2. Tsang KK, Lam MMC, Wick RR, Wyres KL, Bachman M, Baker S, et al. Diversity, functional classification and genotyping of SHV β-lactamases in <em>Klebsiella pneumoniae</em>. bioRxiv. 2024:2024.04.05.587953.

3. Wyres KL, Wick RR, Judd LM, Froumine R, Tokolyi A, Gorrie CL, et al. Distinct evolutionary dynamics of horizontal gene transfer in drug resistant and virulent clones of Klebsiella pneumoniae. PLoS Genet. 2019;15(4):e1008114.

Websites:

https://stats.stackexchange.com/questions/82326/how-to-interpret-the-dendrogram-of-a-hierarchical-cluster-analysis

https://www.kaggle.com/code/berkayalan/unsupervised-learning-clustering-complete-guide

https://datatab.net/tutorial/hierarchical-cluster-analysis

https://www.r-project.org/other-docs.html

Packages:

https://cran.r-project.org/web/packages/rentrez/vignettes/rentrez_tutorial.html

https://cran.r-project.org/web/packages/ape/ape.pdf

https://bioconductor.github.io/BiocManager/

https://www.rdocumentation.org/packages/cluster/versions/2.1.6

https://www.bioconductor.org/packages/release/bioc/html/DECIPHER.html

https://cran.r-project.org/web/packages/cluster/cluster.pdf

https://cran.r-project.org/web/packages/clValid/vignettes/clValid.pdf