Clustering de proteínas

Gabriel Ramirez-Vilchis

2025-02-20

Introducción

Desarrollo

Para realizar el análisis de clustering de proteínas, se realizó un BLAST de proteína-proteína del archivo FASTA desde línea de comandos, haciendo uso del servidor *chaac* de la Licenciatura en Ciencias Genómicas de la UNAM.

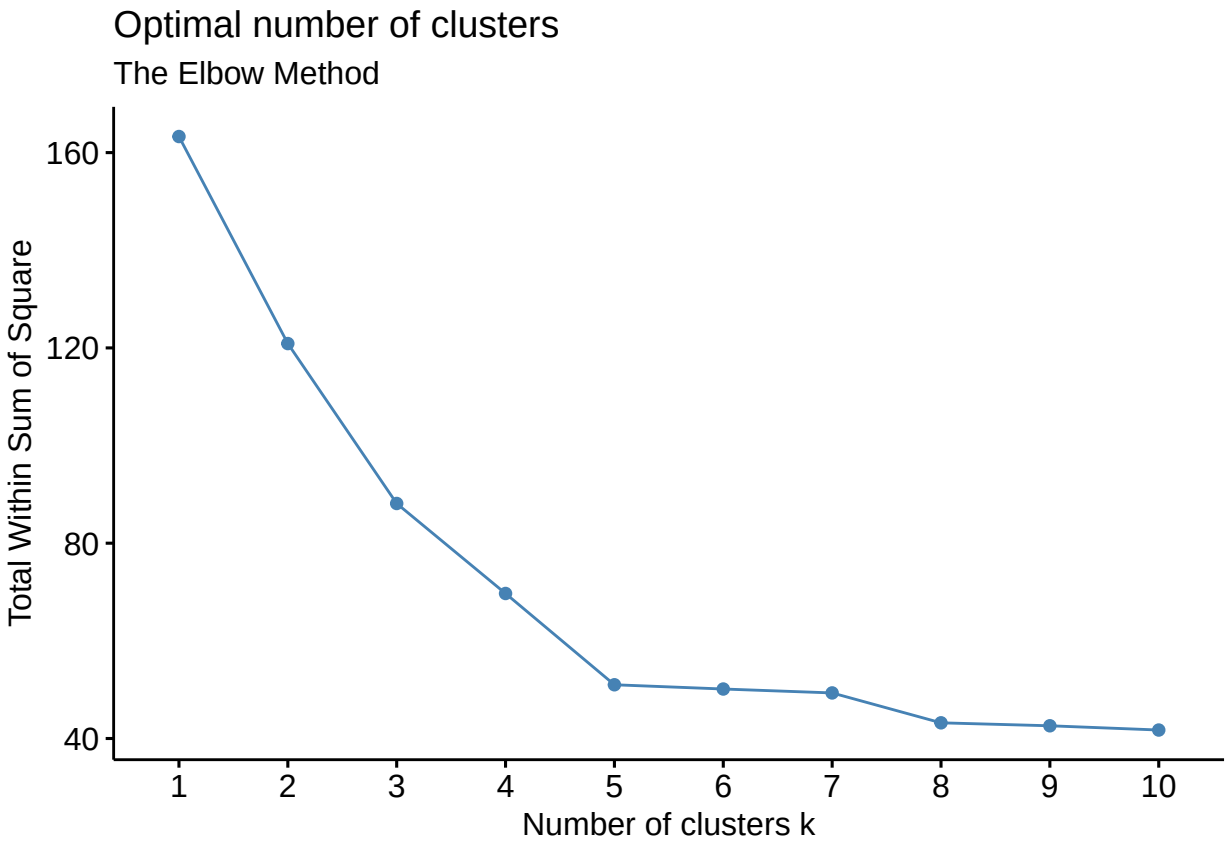
```
# Create BLAST database
makeblastdb -in ABC.faa -dbtype prot -out ABC_blastdb

# Execute BlastP
blastp -query ABC.faa -db ABC_blastdb -outfmt 7 -max_hsps 1 -use_sw_tback -out ABC.blastp
```

Formateo de los datos

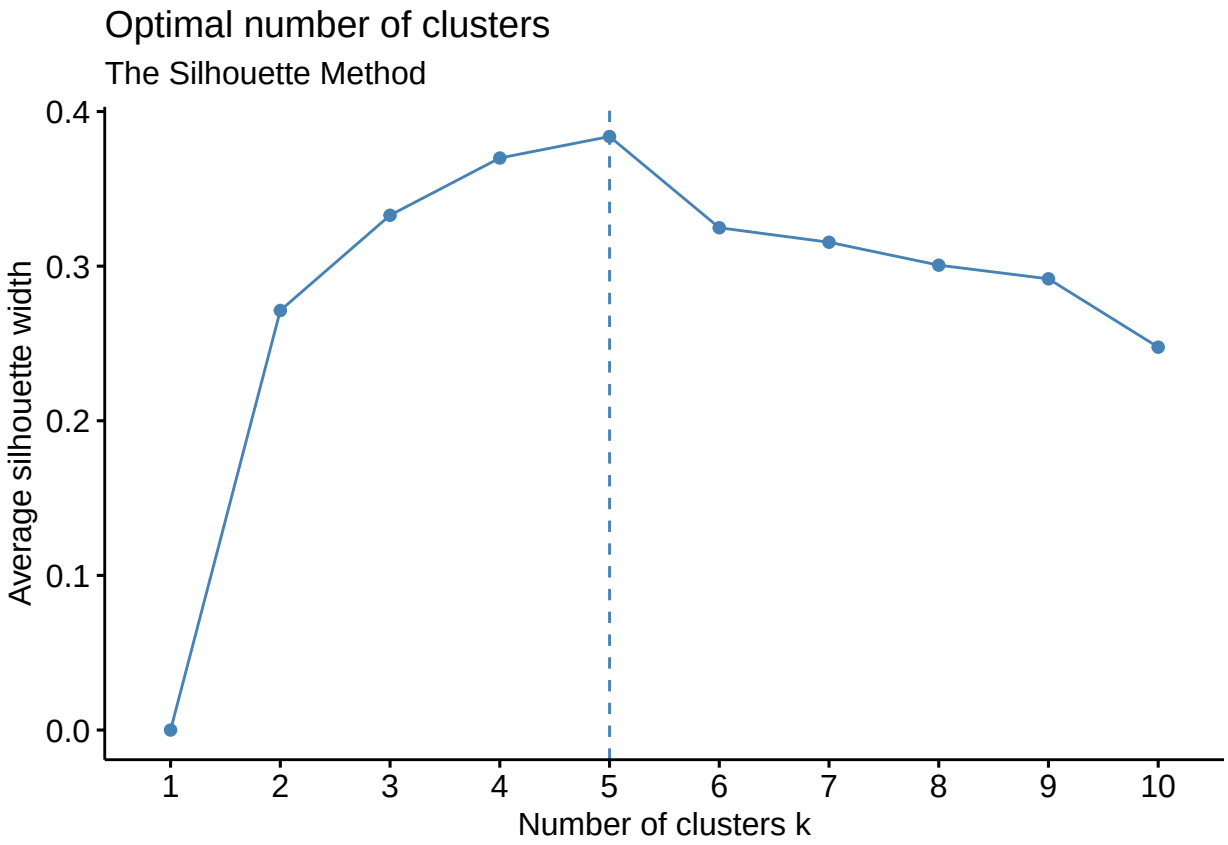
Posteriormente, se realizó un análisis de clustering de proteínas haciendo uso de R. Con este fin se cargaron los datos del archivo de salida del BLAST y se generó una matriz de disimilitud, usando los bit scores obtenidos con BLAST.

```
# Import libraries
library(cluster)
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(factoextra))
suppressPackageStartupMessages(library(dendextend))
suppressPackageStartupMessages(library(ape))
suppressPackageStartupMessages(library(corrplot))

# Read the data
data <- read.table("data/ABC.blastp", sep = "\t", header = FALSE, comment.char = "#")

# Assign names to the columns
colnames(data) <- c("query",
                    "subject",
                    "identity",
                    "alignment_length",
                    "mismatches",
                    "gap_opens",
                    "q_start",
```

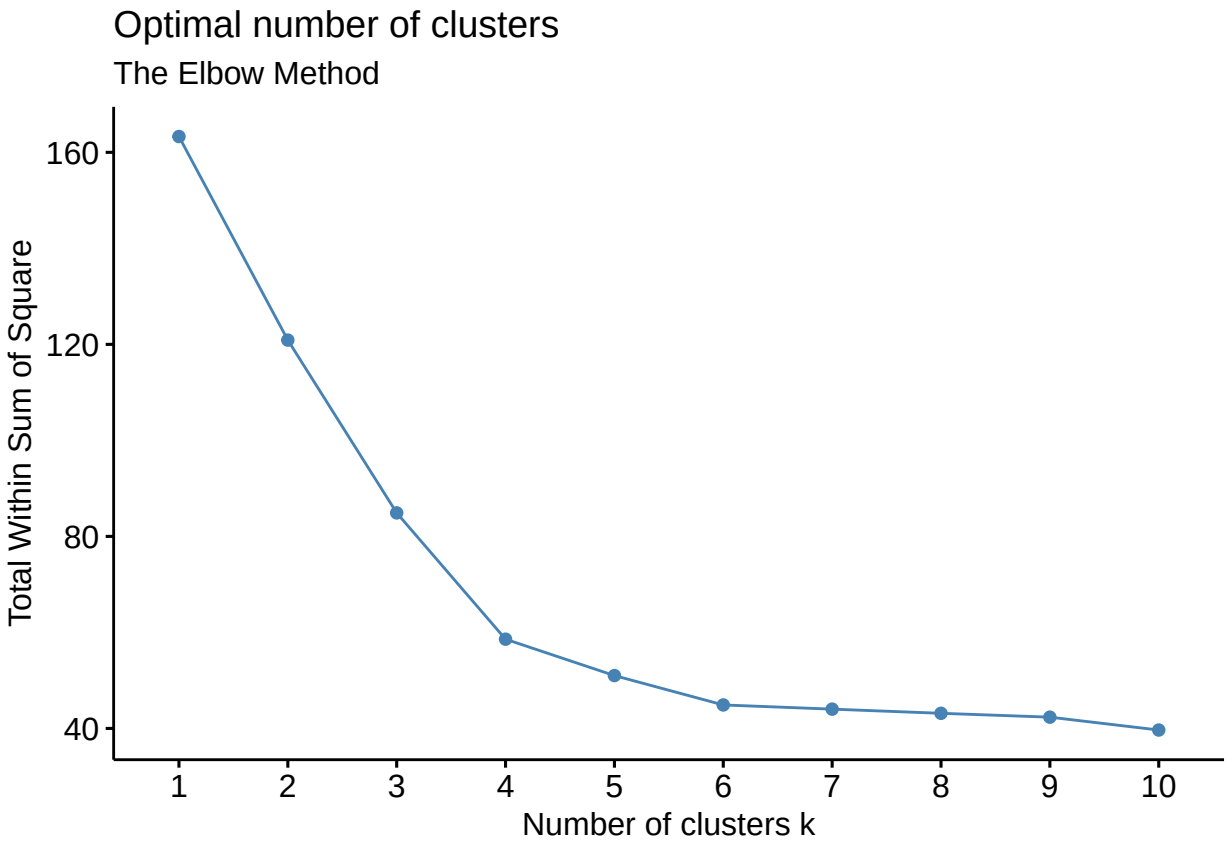
```
                    "q_end",
                    "s_start",
                    "s_end",
                    "evalue",
                    "bit_score")

# Calculate the normalized similarity
similarity <- select(data, query, subject, bit_score)
similarity <- mutate(similarity, normalized_bit_score = bit_score / max(data$bit_score))

# Regularize diagonal in matrix
for(row in 1:nrow(similarity)) {
  if(similarity[row, "query"] == similarity[row, "subject"]) {
    similarity[row, "normalized_bit_score"] <- 1
  }
}

# Calculate dissimilarity
dissimilarity <- mutate(similarity, dissimilarity = 1 - normalized_bit_score)

# Create a dissimilarity matrix
dissimilarity_matrix <- dissimilarity %>%
  select(query, subject, dissimilarity) %>%
  spread(key = subject, value = dissimilarity) %>%
  column_to_rownames(var = "query")

fviz_nbclust(dissimilarity_matrix, FUN = hcut, hc_method = "single", method = "wss", k.max = 10) +
  labs(subtitle = "The Elbow Method")
```



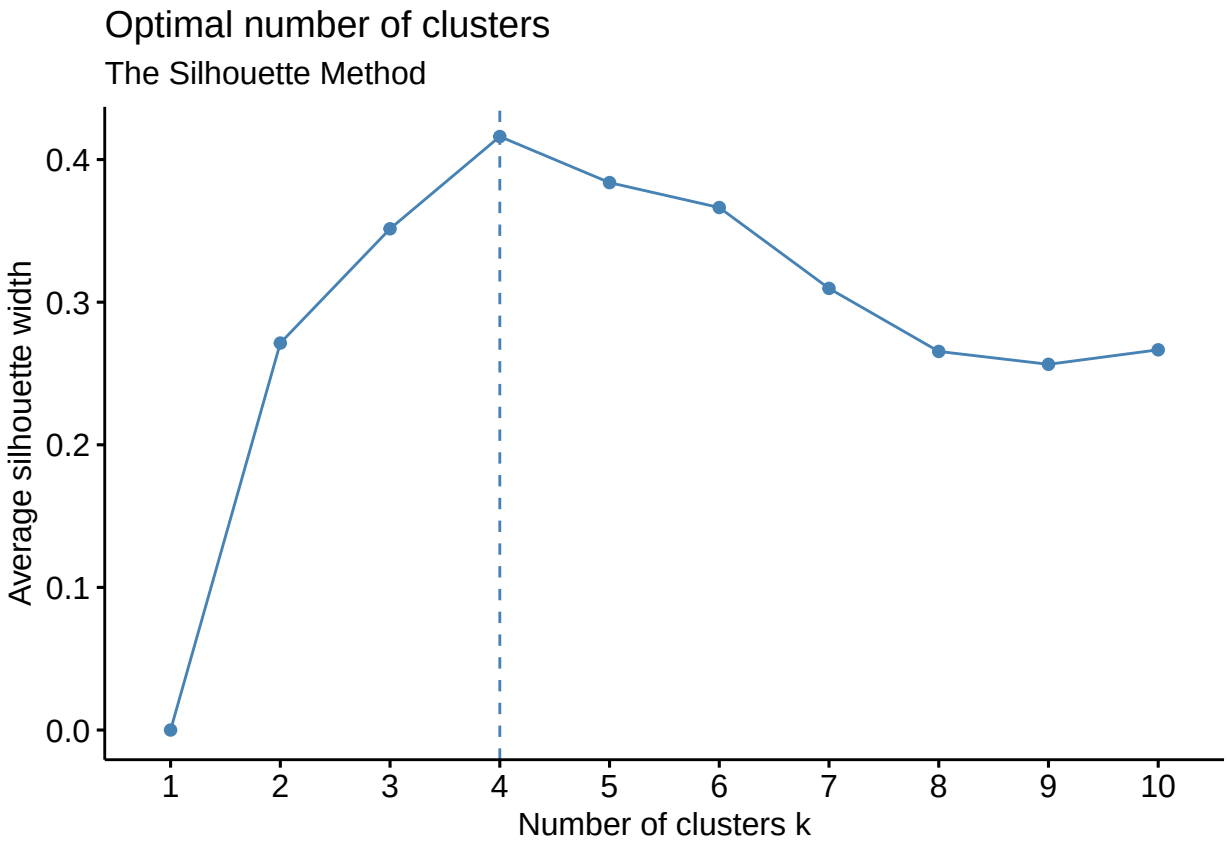

```
fviz_nbclust(dissimilarity_matrix, FUN = hcut, hc_method = "single", method = "silhouette", k.max = 10)
  labs(subtitle = "The Silhouette Method")
```



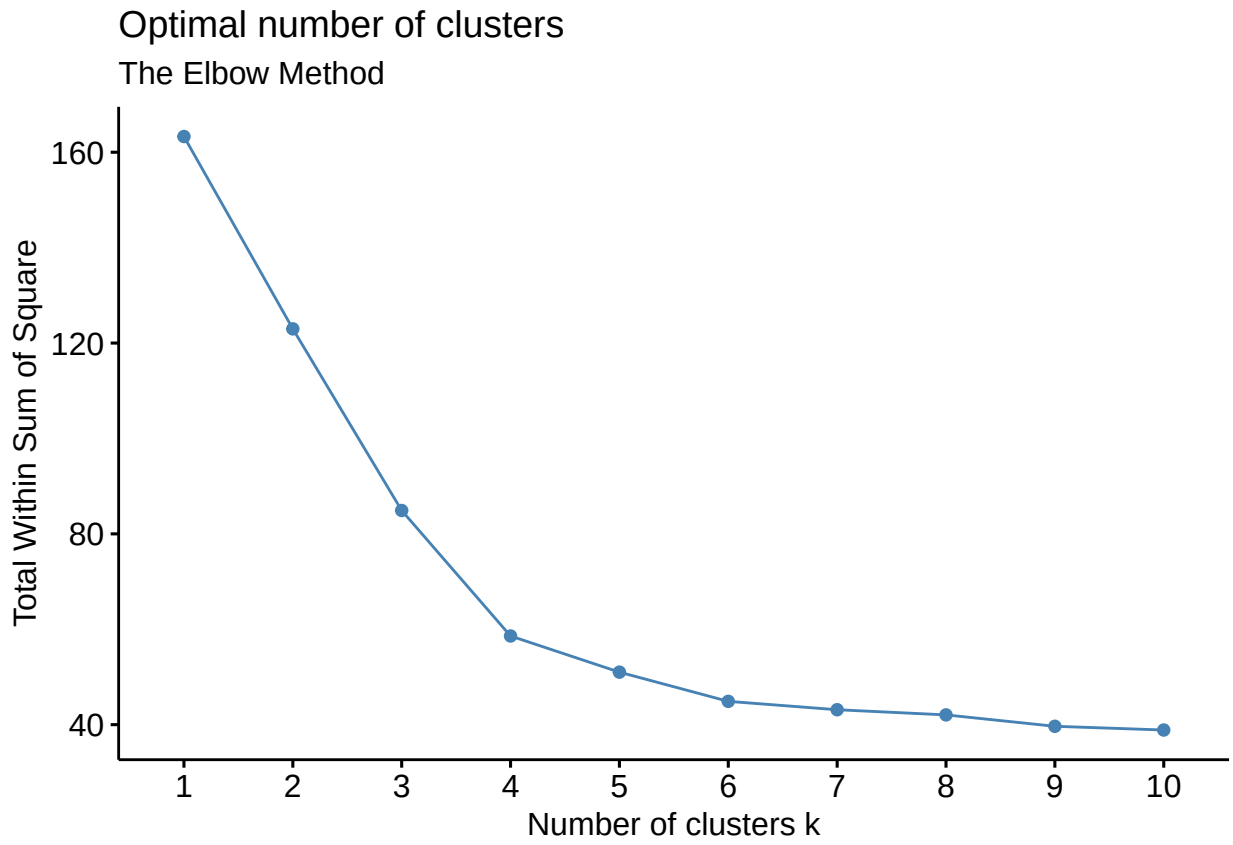

```
fviz_nbclust(dissimilarity_matrix, FUN = hcut, hc_method = "average", method = "wss", k.max = 10) +
  labs(subtitle = "The Elbow Method")
```



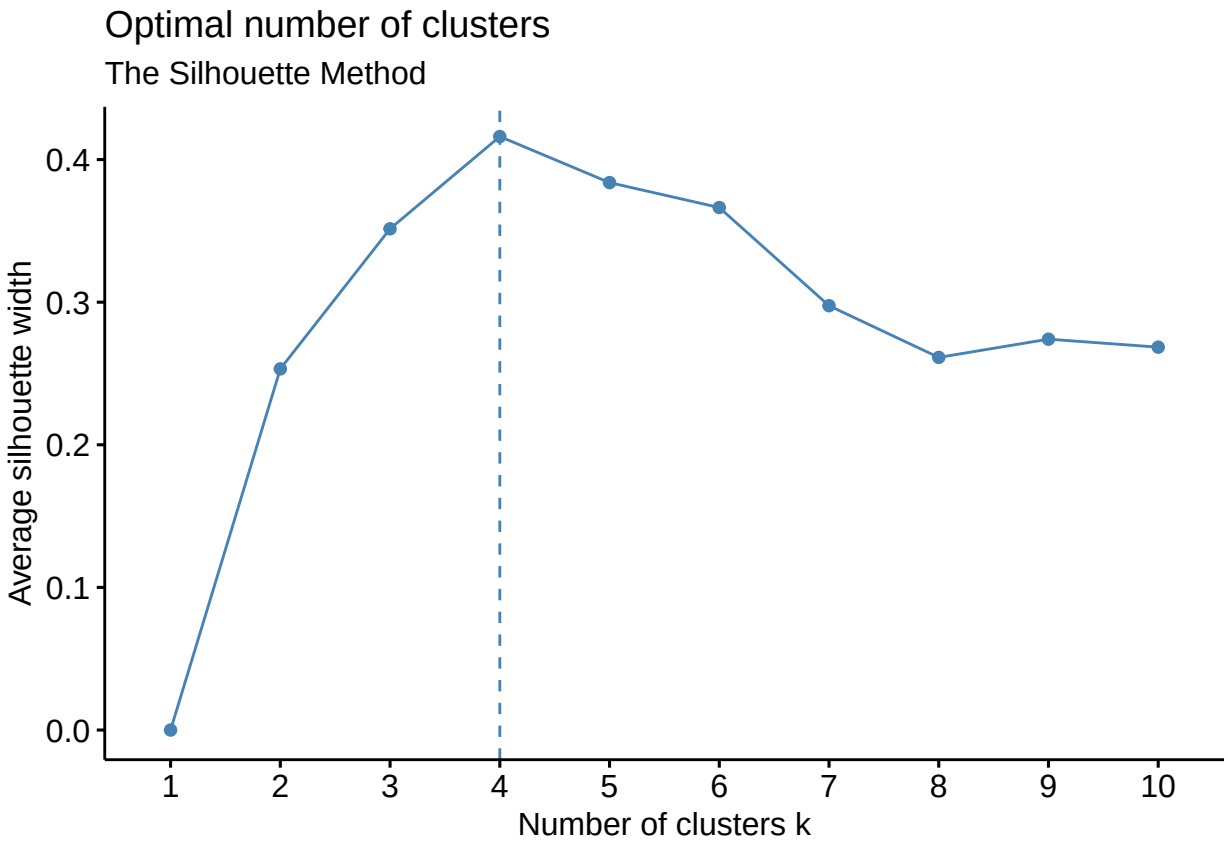

```
fviz_nbclust(dissimilarity_matrix, FUN = hcut, hc_method = "average", method = "silhouette", k.max = 10)
  labs(subtitle = "The Silhouette Method")
```



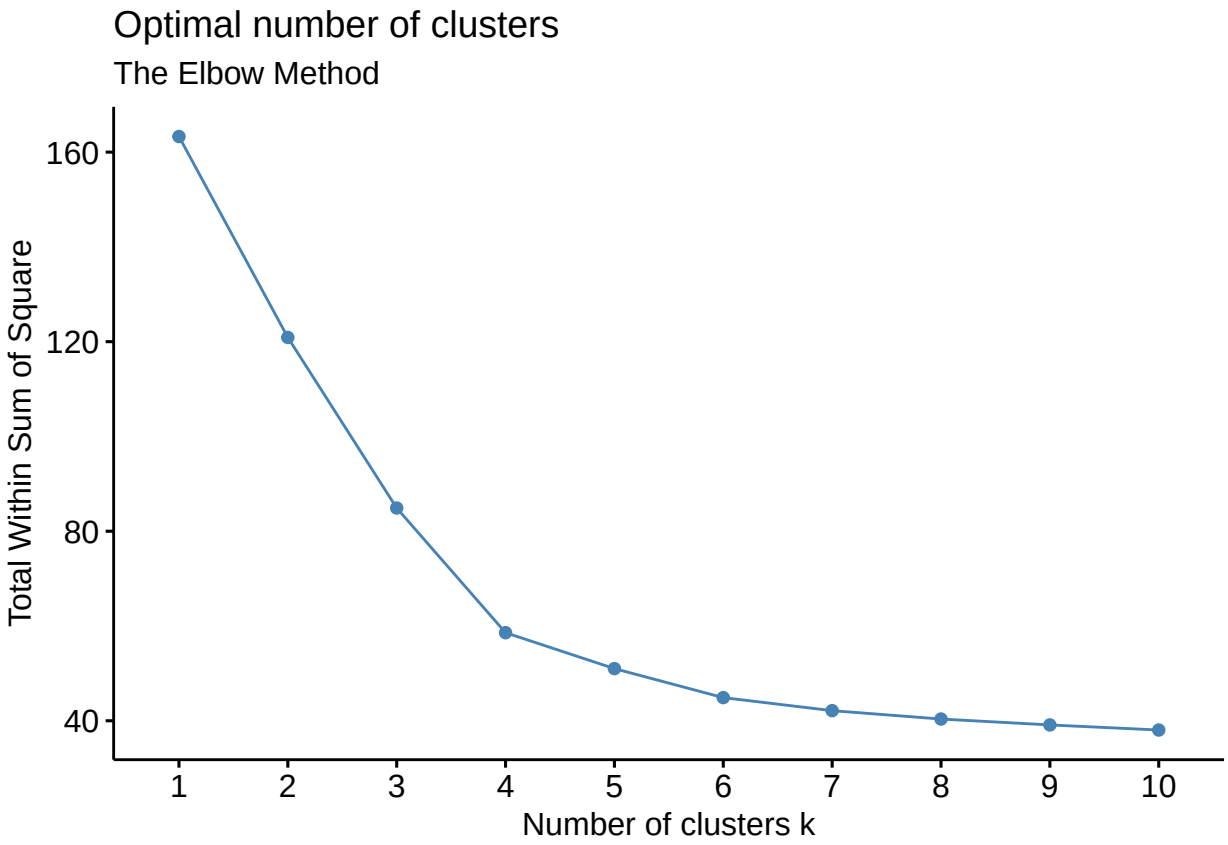

```
fviz_nbclust(dissimilarity_matrix, FUN = hcut, hc_method = "complete", method = "wss", k.max = 10) +
  labs(subtitle = "The Elbow Method")
```



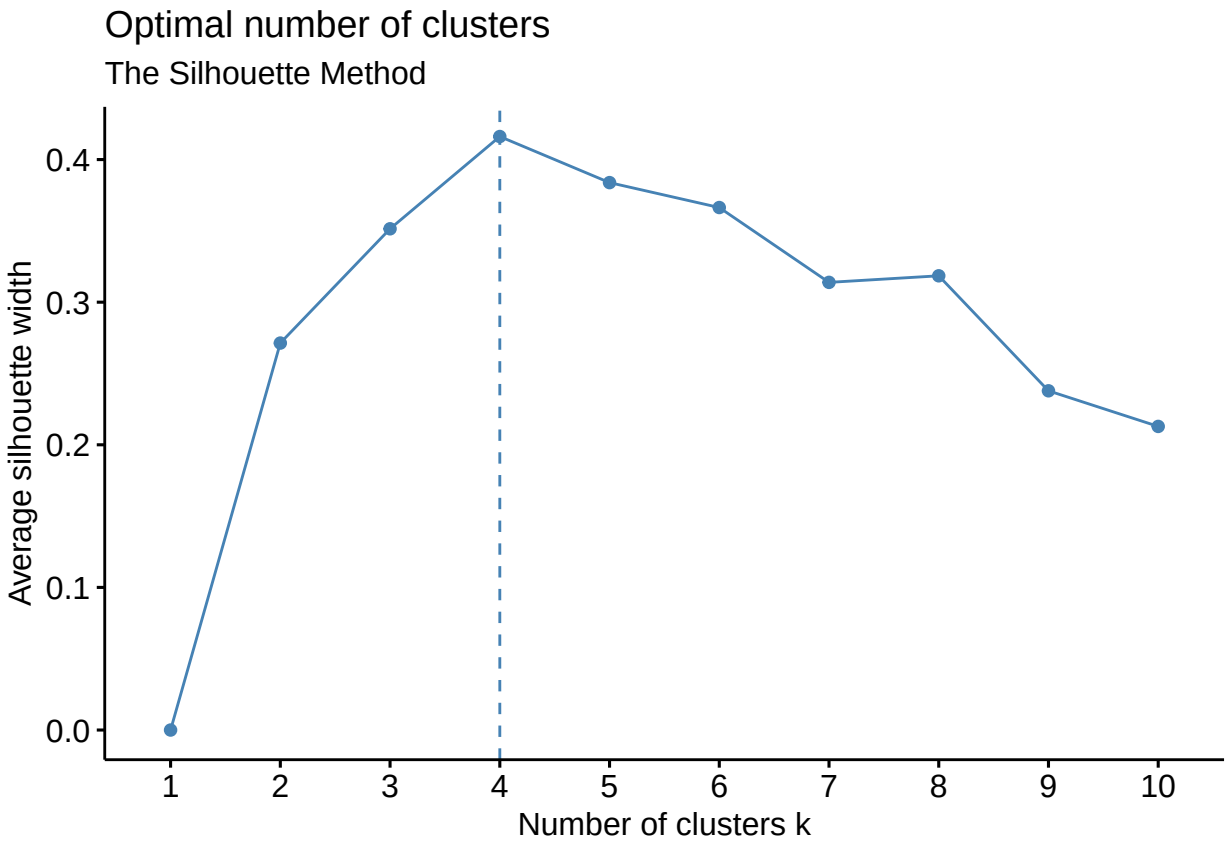

```
fviz_nbclust(dissimilarity_matrix, FUN = hcut, hc_method = "complete", method = "silhouette", k.max = 1
  labs(subtitle = "The Silhouette Method")
```



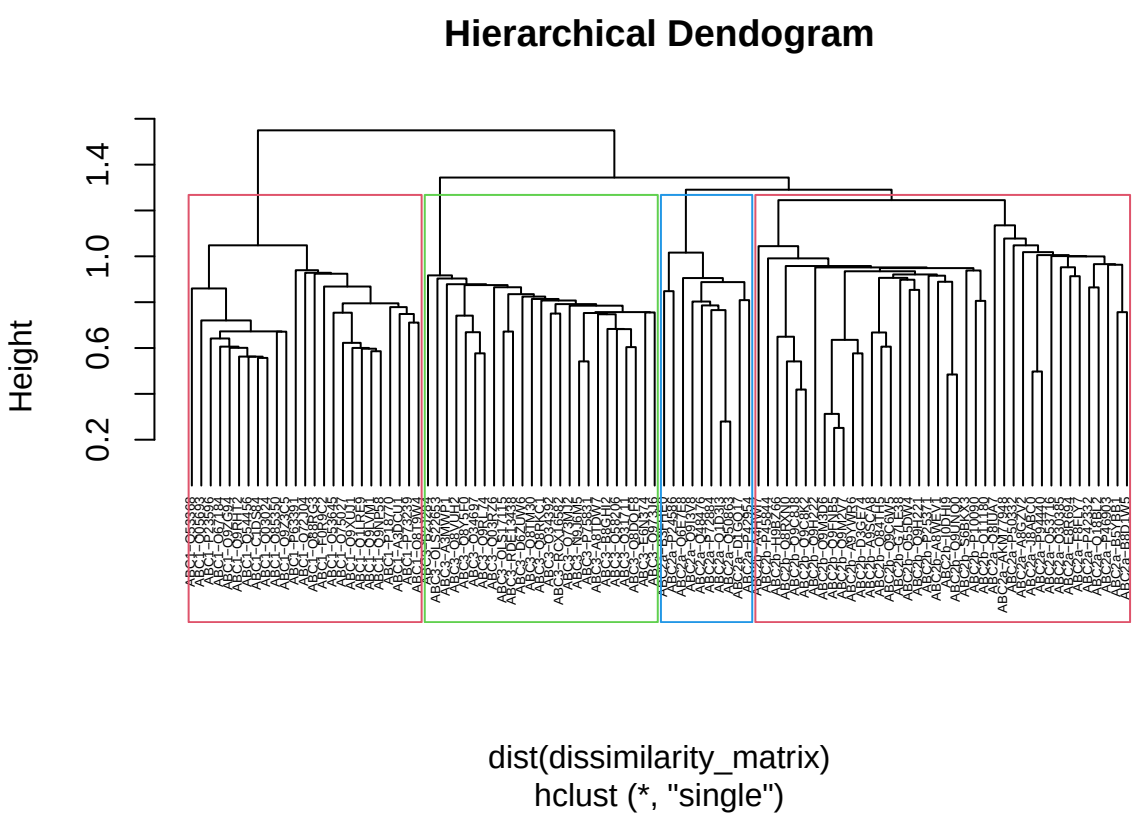

```
fviz_nbclust(dissimilarity_matrix, FUN = hcut, hc_method = "ward.D", method = "wss", k.max = 10) +
  labs(subtitle = "The Elbow Method")
```




```
fviz_nbclust(dissimilarity_matrix, FUN = hcut, hc_method = "ward.D", method = "silhouette", k.max = 10)
  labs(subtitle = "The Silhouette Method")
```




```r
hierarchical_clustering <- hclust(dist(dissimilarity_matrix), method = "single")
coeff <- coef(hierarchical_clustering)
plot(hierarchical_clustering, hang = -1, main = "Hierarchical Dendogram", cex=0.4)
cls3 <- cutree(hierarchical_clustering, k=4)

#cut the dendogram such that 3 clusters are produced
rect.hclust(hierarchical_clustering, k=4, border=2:4)
```


Hierarchical Dendogram
Height
0.2
0.6
1.0
1.4
dist(dissimilarity_matrix)
hclust (*, "single")