Nombre:

Bernardo de la Sierra Rábago

Luis Fernando Pérez Robles

Matrícula:

A01735821

A00832937

Materia:
Análisis de biología computacional

Maestra:

Heriberto García Coronado

Tema:

Evidencia 2

Fecha:

08 de mayo del 2022

Parte 1:

En un video, responde justificadamente las siguientes preguntas:

1. En la actualidad ¿Cuáles son las variantes de SARS-CoV-2 conocidas? Obtén tus referencias de PUBMED o cualquier otra fuente de información confiable.

2. ¿Cuáles son los coronavirus reportados en otras especies de animales que pueden ser cercanos al genoma de SARS-CoV-2? Incluye de qué especies son y las referencias de las fuentes de información confiable que consultaste para responder tu pregunta.

3. En relación con la situación actual reflexiona, ¿Qué propondrías que se deba hacer durante la contingencia del SARS-CoV-2 en comunidades de bajos recursos? Si tu vivieras en una situación de escasos recursos, ¿Qué harías? Justifica tu respuesta.

Link del video:

https://www.youtube.com/watch?v=d8nShNANvFU

Parte 2

1. Diseña un análisis filogenético del virus SARS-CoV-2, en donde incluyas de 8 a 10 genomas virales. (En nuestro caso escogimos analizar secuencia de SARS-CoV-2 2 en varios países)

Instalación de paquetes y llamado a librerías

```
library("seqinr")
library("ape")
```

```
##
## Attaching package: 'ape'
```

```
## The following objects are masked from 'package:seqinr':
##
##     as.alignment, consensus
```

```
library("Biostrings")
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get,
grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: XVector

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:ape':
##
##     complement

## The following object is masked from 'package:seqinr':
##
##     translate

## The following object is masked from 'package:base':
##
##     strsplit
```

```
library("ggplot2")
library("ggtree")
```

```
## Registered S3 method overwritten by 'ggtree':
##   method      from
##   identify.gg ggfun
```

```
## ggtree v3.2.1  For help: https://yulab-smu.top/treedata-book/
##
## If you use ggtree in published research, please cite the most
appropriate paper(s):
##
## 1. Guangchuang Yu. Using ggtree to visualize data on tree-like
structures. Current Protocols in Bioinformatics. 2020, 69:e96.
doi:10.1002/cpbi.96
## 2. Guangchuang Yu, Tommy Tsan-Yuk Lam, Huachen Zhu, Yi Guan. Two
methods for mapping and visualizing associated data on phylogeny using
ggtree. Molecular Biology and Evolution. 2018, 35(12):3041-3043.
doi:10.1093/molbev/msy194
## 3. Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk
Lam. ggtree: an R package for visualization and annotation of
phylogenetic trees with their covariates and other associated data.
Methods in Ecology and Evolution. 2017, 8(1):28-36. doi:10.1111/2041-
210X.12628
```

```
##
## Attaching package: 'ggtree'
```

```
## The following object is masked from 'package:Biostrings':
##
##     collapse
```

```
## The following object is masked from 'package:IRanges':
##
##     collapse
```

```
## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```
## The following object is masked from 'package:ape':
##
##     rotate
```

```
library("DECIPHER")
```

```
## Loading required package: RSQLite
```

```
## Loading required package: parallel
```

```
library("viridis")
```

```
## Loading required package: viridisLite
```

```
library("ade4")

##
## Attaching package: 'ade4'

## The following object is masked from 'package:Biostrings':
##
##     score

## The following object is masked from 'package:BiocGenerics':
##
##     score
```

Posicionamiento en el directorio de trabajo

```
A=getwd()
A

## [1] "C:/Users/adria/Desktop/R"

setwd("/Users/")
setwd(A)
```

Obtén las secuencias de 8 a 10 variantes de SARS-CoV-2 desde el NCBI (Enlaces a un sitio externo.) o el buscador de virus del NCBI

```
Mex1 <- read.fasta("SARSCOVMex1.fasta")
Mex2 <- read.fasta("SARSCOVMex2.fasta")
Mex3 <- read.fasta("SARSCOVMex3.fasta")
Usa1 <- read.fasta("SARSCOVUsa1.fasta")
Usa2 <- read.fasta("SARSCOVUsa2.fasta")
Usa3 <- read.fasta("SARSCOVUsa3.fasta")
Can1 <- read.fasta("SARSCOVCan1.fasta")
Can2 <- read.fasta("SARSCOVCan2.fasta")
Can3 <- read.fasta("SARSCOVCan3.fasta")
```

2. Calcula la longitud y el contenido GC de las secuencias que incluyes en tu análisis. Crea una gráfica que te permita visualizar y comparar la longitud de los genomas analizados y otra que te permita ver el contenido de GC. Muestra el código empleado para obtenerlo e incluye las gráficas que obtuviste.

Calculo de longitud y contenido de GC

```
size <- function(dna){ # funcion que calcula el tamaño de la secuencia
  print("Número de bases: ")
  return (length(dna))
}

# Este código calcula el porcentaje GC en la secuencia
```

```r
cuenta_GC<- function(dna){ # funcion que calcula el porcentaje de GC y
  #Contadores
  g <- 0
  c <- 0
  # Bucle de repetición y checa cuantas veces salen las letras
  for( i in 1:length(dna)){
     if(dna[i] == "g")
      g <- g + 1
     else if(dna[i] == "c")
      c <- c+1
  }
  # Impresión de la información
  print("Porcentaje de Citosina: ")
  print((c/length(dna))*100)
  print("Porcentaje de Guanina: ")
  print((g/length(dna))*100)
  print("Porcentaje de GC: ")
  print((c/length(dna))*100+(g/length(dna))*100)
}

complementaria <- function(dna){ # Funcion que crea la hebra
complementaria cambiando las letras
  cadena <- c()# Vector vacío que guarda la nueva hebra
  # Ciclo de repetición donde se cambian las letras
  for( i in 1:length(dna)){
    if(dna[i] == "a")
      cadena[i] <- "t"
    else if(dna[i] == "t")
      cadena[i] <- "a"
    if(dna[i] == "c")
      cadena[i] <- "g"
    else if(dna[i] == "g")
      cadena[i] <- "c"
  }
  print("La secuencia complementaria es: ")
  return (cadena)
}

print("Información de Sars covid 2 en México de Oaxaca");

## [1] "Información de Sars covid 2 en México de Oaxaca"

size(Mex1[[1]])

## [1] "Número de bases: "

## [1] 29903

cuenta_GC(Mex1[[1]])
```

```
## [1] "Porcentaje de Citosina: "
## [1] 18.34599
## [1] "Porcentaje de Guanina: "
## [1] 19.60673
## [1] "Porcentaje de GC: "
## [1] 37.95271
```

```
head(complementaria(Mex1[[1]]))
```

```
## [1] "La secuencia complementaria es: "
```

```
## [1] "t" "a" "a" "t" "t" "t"
```

```
print("Información de Sars covid 2 en México de Ciudad de México");
```

```
## [1] "Información de Sars covid 2 en México de Ciudad de México"
```

```
size(Mex2[[1]])
```

```
## [1] "Número de bases: "
```

```
## [1] 29903
```

```
cuenta_GC(Mex2[[1]])
```

```
## [1] "Porcentaje de Citosina: "
## [1] 17.91459
## [1] "Porcentaje de Guanina: "
## [1] 19.17199
## [1] "Porcentaje de GC: "
## [1] 37.08658
```

```
head(complementaria(Mex2[[1]]))
```

```
## [1] "La secuencia complementaria es: "
```

```
## [1] "t" "a" "a" "t" "t" "t"
```

```
print("Información de Sars covid 2 en México de Baja California");
```

```
## [1] "Información de Sars covid 2 en México de Baja California"
```

```
size(Mex3[[1]])
```

```
## [1] "Número de bases: "
```

```
## [1] 29903
```

```
cuenta_GC(Mex3[[1]])
```

```
## [1] "Porcentaje de Citosina: "
## [1] 18.34264
## [1] "Porcentaje de Guanina: "
## [1] 19.61342
```

```
## [1] "Porcentaje de GC: "
## [1] 37.95606
```

```
head(complementaria(Mex3[[1]]))
```

```
## [1] "La secuencia complementaria es: "
```

```
## [1] "t" "a" "a" "t" "t" "t"
```

```
print("Información de Sars covid 2 en Estados Unidos de Delaware");
```

```
## [1] "Información de Sars covid 2 en Estados Unidos de Delaware"
```

```
size(Usa1[[1]])
```

```
## [1] "Número de bases: "
```

```
## [1] 29940
```

```
cuenta_GC(Usa1[[1]])
```

```
## [1] "Porcentaje de Citosina: "
## [1] 18.33333
## [1] "Porcentaje de Guanina: "
## [1] 19.57582
## [1] "Porcentaje de GC: "
## [1] 37.90915
```

```
head(complementaria(Usa1[[1]]))
```

```
## [1] "La secuencia complementaria es: "
```

```
## [1] "t" "a" "a" "t" "t" "t"
```

```
print("Información de Sars covid 2 en Estados Unidos de Florida");
```

```
## [1] "Información de Sars covid 2 en Estados Unidos de Florida"
```

```
size(Usa2[[1]])
```

```
## [1] "Número de bases: "
```

```
## [1] 29933
```

```
cuenta_GC(Usa2[[1]])
```

```
## [1] "Porcentaje de Citosina: "
## [1] 18.08372
## [1] "Porcentaje de Guanina: "
## [1] 19.38997
## [1] "Porcentaje de GC: "
## [1] 37.47369
```

```
head(complementaria(Usa2[[1]]))
```

```
## [1] "La secuencia complementaria es: "

## [1] "t" "t" "t" "c" "c" "a"

print("Información de Sars covid 2 en Estados Unidos de Washington");

## [1] "Información de Sars covid 2 en Estados Unidos de Washington"

size(Usa3[[1]])

## [1] "Número de bases: "

## [1] 29921

cuenta_GC(Usa3[[1]])

## [1] "Porcentaje de Citosina: "
## [1] 18.33495
## [1] "Porcentaje de Guanina: "
## [1] 19.6083
## [1] "Porcentaje de GC: "
## [1] 37.94325

head(complementaria(Usa3[[1]]))

## [1] "La secuencia complementaria es: "

## [1] "g" "a" "c" "c" "t" "t"

print("Información de Sars covid 2 en Canada de Ontario");

## [1] "Información de Sars covid 2 en Canada de Ontario"

size(Can1[[1]])

## [1] "Número de bases: "

## [1] 29853

cuenta_GC(Can1[[1]])

## [1] "Porcentaje de Citosina: "
## [1] 17.73691
## [1] "Porcentaje de Guanina: "
## [1] 19.02656
## [1] "Porcentaje de GC: "
## [1] 36.76347

head(complementaria(Can1[[1]]))

## [1] "La secuencia complementaria es: "

## [1] "g" NA  "c" "c" "a" "t"

print("Información de Sars covid 2 en Canada de Alberta");
```

```
## [1] "Información de Sars covid 2 en Canada de Alberta"

size(Can2[[1]])

## [1] "Número de bases: "

## [1] 29834

cuenta_GC(Can2[[1]])

## [1] "Porcentaje de Citosina: "
## [1] 18.16049
## [1] "Porcentaje de Guanina: "
## [1] 19.44761
## [1] "Porcentaje de GC: "
## [1] 37.6081

head(complementaria(Can2[[1]]))

## [1] "La secuencia complementaria es: "

## [1] "g" "a" "a" "c" "a" "t"

print("Información de Sars covid 2 en Canada de British Columbia");

## [1] "Información de Sars covid 2 en Canada de British Columbia"

size(Can3[[1]])

## [1] "Número de bases: "

## [1] 29782

cuenta_GC(Can3[[1]])

## [1] "Porcentaje de Citosina: "
## [1] 18.35001
## [1] "Porcentaje de Guanina: "
## [1] 19.63266
## [1] "Porcentaje de GC: "
## [1] 37.98267

head(complementaria(Can3[[1]]))

## [1] "La secuencia complementaria es: "

## [1] "t" "c" "t" "a" "g" "a"
```

Creación de la tabla para poder visualizar los gráficos

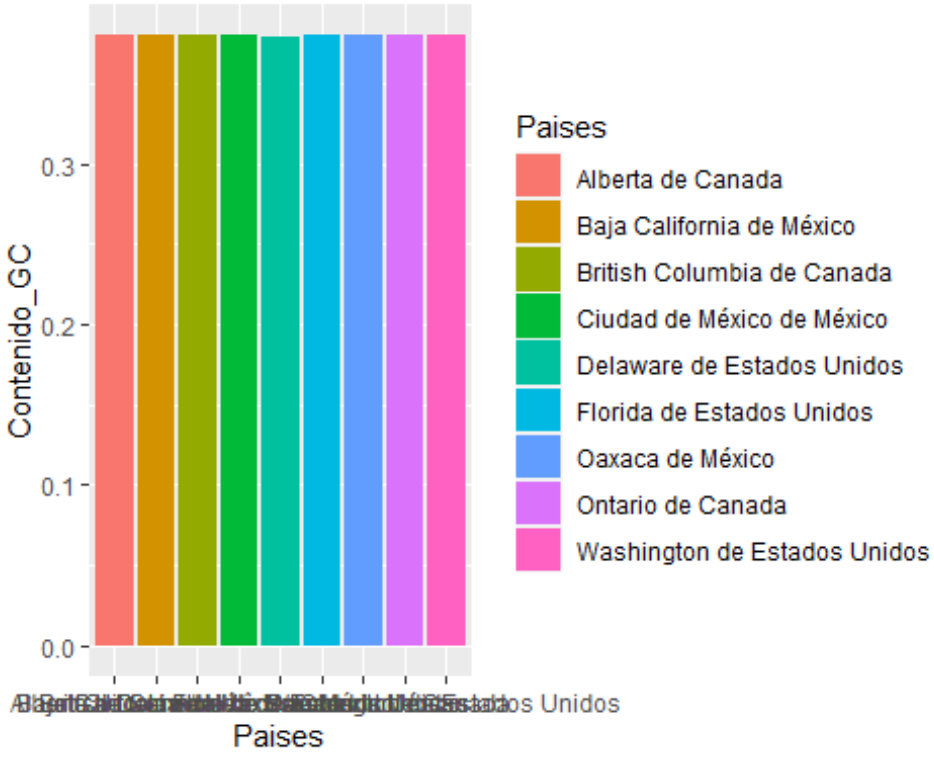
```
# install.packages("tidyverse") aca se comenta porque ya instale el
paquete
tabla_ev01<- data.frame(Paises= 1:9,Longitud = 1:9, Contenido_GC = 1:9);
rownames(tabla_ev01) <- c("1","2","3","4","5","6","7","8","9")
tabla_ev01[1,1 ] <- "Oaxaca de México"
```

```
tabla_ev01[2,1] <- "Ciudad de México de México"
tabla_ev01[3,1 ] <- "Baja California de México"
tabla_ev01[4,1] <- "Delaware de Estados Unidos"
tabla_ev01[5,1 ] <- "Florida de Estados Unidos"
tabla_ev01[6,1 ] <- "Washington de Estados Unidos"
tabla_ev01[7,1 ] <- "Ontario de Canada"
tabla_ev01[8,1 ] <- "Alberta de Canada"
tabla_ev01[9,1 ] <- "British Columbia de Canada"

tabla_ev01[1,2 ] <- length(Mex1[[1]])
tabla_ev01[2,2] <-  length(Mex2[[1]])
tabla_ev01[3,2 ] <- length(Mex3[[1]])
tabla_ev01[4,2] <- length(Usa1[[1]])
tabla_ev01[5,2 ] <- length(Usa2[[1]])
tabla_ev01[6,2 ] <- length(Usa3[[1]])
tabla_ev01[7,2 ] <- length(Can1[[1]])
tabla_ev01[8,2 ] <- length(Can2[[1]])
tabla_ev01[9,2 ] <- length(Can3[[1]])

tabla_ev01[1,3 ] <- GC(Mex1[[1]])
tabla_ev01[2,3] <-  GC(Mex2[[1]])
tabla_ev01[3,3 ] <- GC(Mex3[[1]])
tabla_ev01[4,3] <- GC(Usa1[[1]])
tabla_ev01[5,3 ] <- GC(Usa2[[1]])
tabla_ev01[6,3 ] <- GC(Usa3[[1]])
tabla_ev01[7,3 ] <- GC(Can1[[1]])
tabla_ev01[8,3 ] <- GC(Can2[[1]])
tabla_ev01[9,3 ] <- GC(Can3[[1]])

tabla_ev01
```

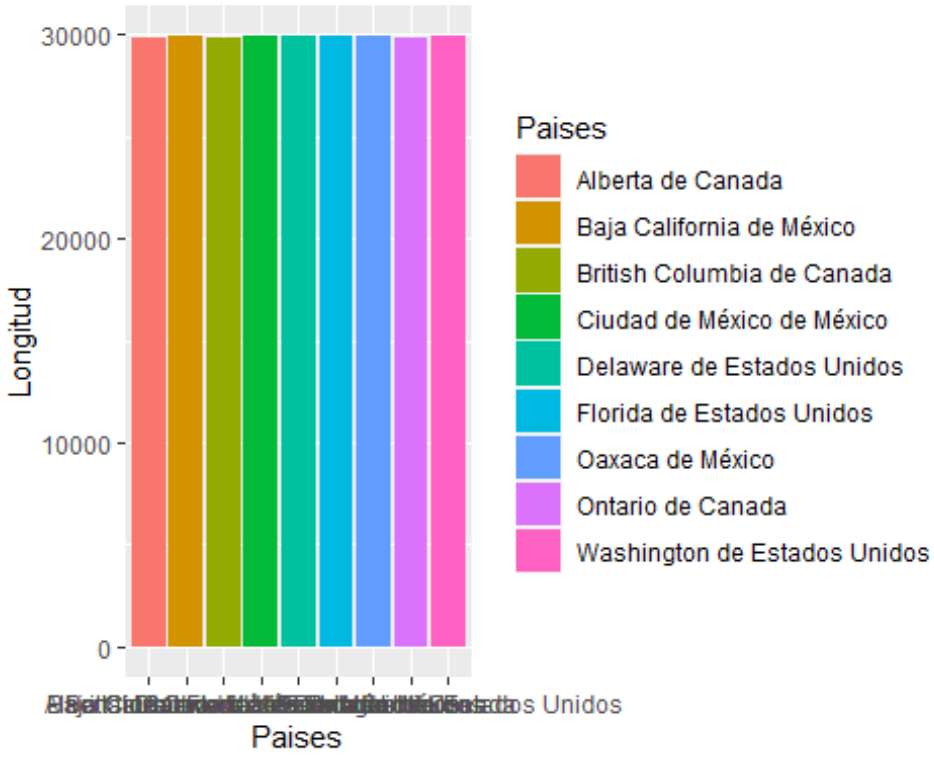
```
##                         Paises Longitud Contenido_GC
## 1             Oaxaca de México    29903    0.3795271
## 2   Ciudad de México de México    29903    0.3800679
## 3    Baja California de México    29903    0.3795606
## 4   Delaware de Estados Unidos    29940    0.3790915
## 5    Florida de Estados Unidos    29933    0.3797353
## 6 Washington de Estados Unidos    29921    0.3794325
## 7            Ontario de Canada    29853    0.3793378
## 8            Alberta de Canada    29834    0.3794643
## 9   British Columbia de Canada    29782    0.3798522
```

Creación de graficas de longitud y GC

```
# Graficamos  el contenido de Guanina Citosina y la Longitud
library(ggplot2)
# Graficamos  el contenido de Guanina Citosina y la Longitud
plot_tabla_ev01_GC <-ggplot(tabla_ev01, aes(x=Paises, y=Contenido_GC,
fill=Paises))+geom_bar(stat = "identity")
plot_tabla_ev01_GC
```

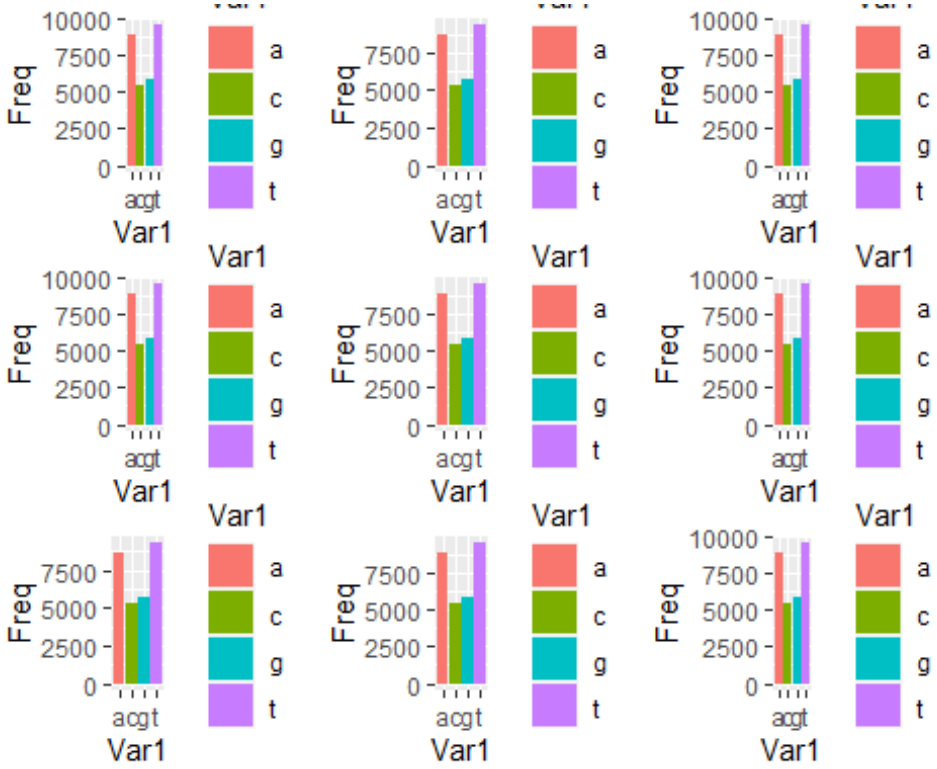



```
plot_tabla_ev01_length <-ggplot(tabla_ev01, aes(x=Paises, y=Longitud,
fill=Paises))+geom_bar(stat = "identity")
plot_tabla_ev01_length
```



3. Crea una gráfica en donde se observe el número de bases (A, G, C y T) de ADN que componen a los genomas virales utilizados en tu análisis. Muestra el código empleado para obtenerlo y la imagen de las gráficas obtenidas.

```
count_Mex1 <- as.data.frame(count(Mex1[[1]],1))
count_Mex2 <- as.data.frame(count(Mex2[[1]],1))
count_Mex3 <- as.data.frame(count(Mex3[[1]],1))
count_Usa1 <- as.data.frame(count(Usa1[[1]],1))
count_Usa2 <- as.data.frame(count(Usa2[[1]],1))
count_Usa3 <- as.data.frame(count(Usa3[[1]],1))
count_Can1 <- as.data.frame(count(Can1[[1]],1))
count_Can2 <- as.data.frame(count(Can2[[1]],1))
count_Can3 <- as.data.frame(count(Can3[[1]],1))

plot_count_Mex1 <- ggplot(count_Mex1,
aes(x=Var1,y=Freq,fill=Var1))+geom_bar(stat = "identity")
plot_count_Mex2 <- ggplot(count_Mex2,
aes(x=Var1,y=Freq,fill=Var1))+geom_bar(stat = "identity")
plot_count_Mex3 <- ggplot(count_Mex3,
aes(x=Var1,y=Freq,fill=Var1))+geom_bar(stat = "identity")
plot_count_Usa1 <- ggplot(count_Usa1,
aes(x=Var1,y=Freq,fill=Var1))+geom_bar(stat = "identity")
plot_count_Usa2 <- ggplot(count_Usa2,
aes(x=Var1,y=Freq,fill=Var1))+geom_bar(stat = "identity")
plot_count_Usa3 <- ggplot(count_Usa3,
aes(x=Var1,y=Freq,fill=Var1))+geom_bar(stat = "identity")
plot_count_Can1 <- ggplot(count_Can1,
aes(x=Var1,y=Freq,fill=Var1))+geom_bar(stat = "identity")
plot_count_Can2 <- ggplot(count_Can2,
aes(x=Var1,y=Freq,fill=Var1))+geom_bar(stat = "identity")
plot_count_Can3 <- ggplot(count_Can3,
aes(x=Var1,y=Freq,fill=Var1))+geom_bar(stat = "identity")

library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:BiocGenerics':
##
##     combine

grid.arrange(plot_count_Mex1,plot_count_Mex2,plot_count_Mex3,plot_count_U
sa1,plot_count_Usa2,plot_count_Usa3,plot_count_Can1,plot_count_Can2,plot_
count_Can3,ncol=3)
```



4. Obtén las secuencias de los genomas de los virus elegidos, según la investigación que hayas decidido realizar, con la función read.GenBank. Muestra el código empleado para obtenerlo.

```
# Aquí obtenemos los genomas virales
# Generamos un vector con los números de accesión de nuestros genomas virales de interés
corona_virus <- c("OL790194.1","OK435439.1","MZ363983.1","MZ472103.1","OK439973.1","MT461607.1","ON112487.1","OM366050.1","MW309426.1")
# Recuperamos las secuencias y los datos de los genomas con la funcion read.GenBank
secuencia_de_virus <- read.GenBank(corona_virus)
```

5. Realiza el alineamiento de los genomas virales y visualiza el resultado de tu alineamiento en tu navegador. Muestra el código empleado para realizar lo anterior e incluye dos imágenes con el resultado del alineamiento, una de los primeros 150 nucleótidos y otra de los nucleótidos 500 al 650.

```
# Concentramos en un archivo las secuencias del genoma
write.dna(secuencia_de_virus, file="coronavirus_seqs.fasta", format = "fasta")
```

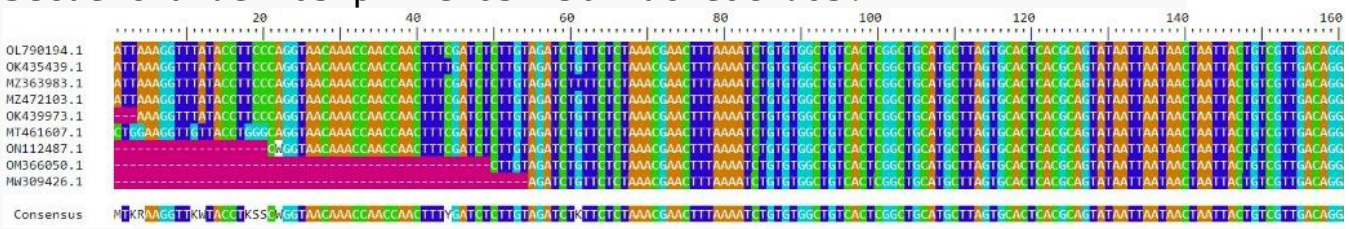
```r
# Carga las secuencias concentradas en el archivo
virus_seq_not_align <- readDNAStringSet("coronavirus_seqs.fasta",
format="fasta")

## Warning in .Call2("fasta_index", filexp_list, nrec, skip,
seek.first.rec, :
## reading FASTA file coronavirus_seqs.fasta: ignored 22416 invalid one-
letter
## sequence codes

# Orientamos los nucleotidos
virus_seq_not_align <- OrientNucleotides(virus_seq_not_align)

##
================================================================================
================================================================================
=================================================
##
## Time difference of 0.26 secs

# Realizamos el alineamiento y visualizamos eel resultado en internet
virus_seq_align <- AlignSeqs(virus_seq_not_align)

## Determining distance matrix based on shared 11-mers:
##
================================================================================
=======
##
## Time difference of 0.03 secs
##
## Clustering into groups by similarity:
##
================================================================================
=======
##
## Time difference of 0.01 secs
##
## Aligning Sequences:
##
================================================================================
=======
##
## Time difference of 1.13 secs
##
## Iteration 1 of 2:
##
## Determining distance matrix based on alignment:
##
================================================================================
=======
##
```

```
## Time difference of 0.01 secs
##
## Reclustering into groups by similarity:
##
=========================================================================
=======
##
## Time difference of 0 secs
##
## Realigning Sequences:
##
=========================================================================
=======
##
## Time difference of 1.09 secs
##
## Iteration 2 of 2:
##
## Determining distance matrix based on alignment:
##
=========================================================================
=======
##
## Time difference of 0 secs
##
## Reclustering into groups by similarity:
##
=========================================================================
=======
##
## Time difference of 0 secs
##
## Realigning Sequences:
##
=========================================================================
=======
##
## Time difference of 0.01 secs
```

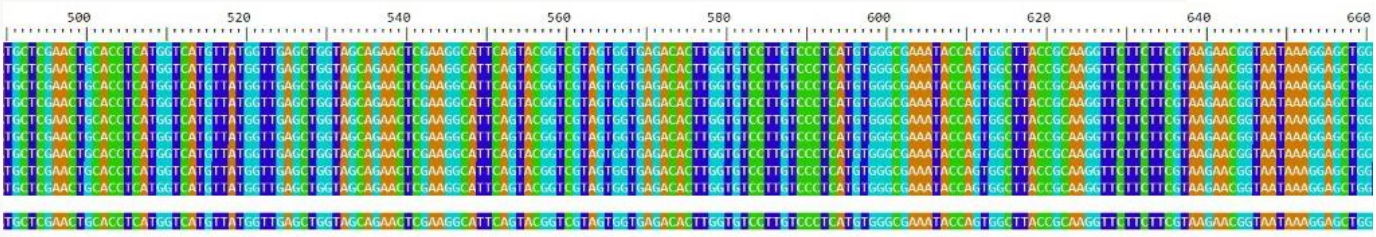
```
BrowseSeqs(virus_seq_align)
```

Imagenes:

Secuencia de los primeros 150 nucleótidos:



Secuencia de 500 a 650:



6. Genera una matriz de distancia a partir de los genomas alineados. Crea una tabla en escala de grises en la que observes de manera visual el resultado de la matriz de distancia e inclúyela en tu reporte. Muestra el código empleado para obtener lo anterior e incluye la tabla que obtuviste.

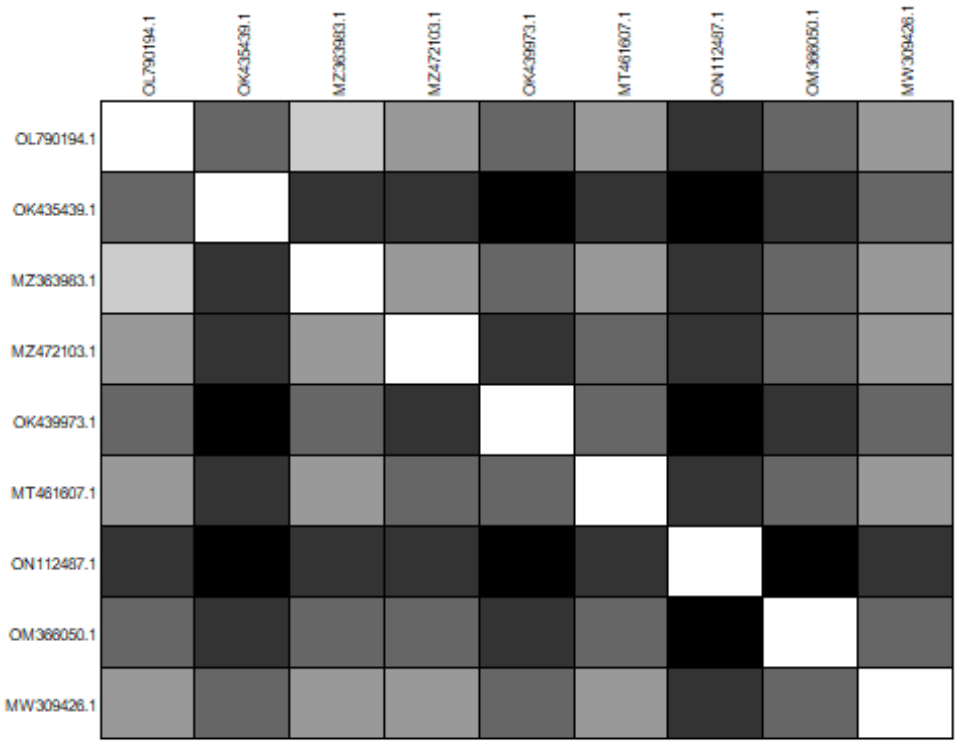
```
writeXStringSet(virus_seq_align, file="coronavirus_seq_align.fasta")

virus_aligned <- read.alignment("coronavirus_seq_align.fasta",
format="fasta")
matriz_distancia <- dist.alignment(virus_aligned, matrix = "similarity")
matriz_distancia
```

```
##             OL790194.1 OK435439.1 MZ363983.1 MZ472103.1 OK439973.1
MT461607.1
## OK435439.1 0.03883212
## MZ363983.1 0.01917957 0.04013412
## MZ472103.1 0.02520690 0.04097914 0.02712401
## OK439973.1 0.03635054 0.05069761 0.03588148 0.04074522
## MT461607.1 0.02085040 0.04261895 0.02586173 0.03059081 0.03947822
## ON112487.1 0.04439313 0.05456124 0.04554639 0.04848783 0.05552732
0.04478085
## OM366050.1 0.03289758 0.04598722 0.03537447 0.03901174 0.04646279
0.03440512
## MW309426.1 0.02168212 0.03846948 0.02389253 0.02897395 0.03912437
0.02244312
##             ON112487.1 OM366050.1
## OK435439.1
## MZ363983.1
## MZ472103.1
## OK439973.1
## MT461607.1
## ON112487.1
## OM366050.1 0.05049792
## MW309426.1 0.04366021 0.03395199
```

```
temp <- as.data.frame(as.matrix(matriz_distancia))
temp
```

```
##             OL790194.1 OK435439.1 MZ363983.1 MZ472103.1 OK439973.1
MT461607.1
## OL790194.1 0.00000000 0.03883212 0.01917957 0.02520690 0.03635054
0.02085040
## OK435439.1 0.03883212 0.00000000 0.04013412 0.04097914 0.05069761
0.04261895
## MZ363983.1 0.01917957 0.04013412 0.00000000 0.02712401 0.03588148
0.02586173
## MZ472103.1 0.02520690 0.04097914 0.02712401 0.00000000 0.04074522
0.03059081
## OK439973.1 0.03635054 0.05069761 0.03588148 0.04074522 0.00000000
0.03947822
## MT461607.1 0.02085040 0.04261895 0.02586173 0.03059081 0.03947822
0.00000000
## ON112487.1 0.04439313 0.05456124 0.04554639 0.04848783 0.05552732
0.04478085
## OM366050.1 0.03289758 0.04598722 0.03537447 0.03901174 0.04646279
0.03440512
## MW309426.1 0.02168212 0.03846948 0.02389253 0.02897395 0.03912437
0.02244312
##             ON112487.1 OM366050.1 MW309426.1
## OL790194.1 0.04439313 0.03289758 0.02168212
## OK435439.1 0.05456124 0.04598722 0.03846948
## MZ363983.1 0.04554639 0.03537447 0.02389253
## MZ472103.1 0.04848783 0.03901174 0.02897395
## OK439973.1 0.05552732 0.04646279 0.03912437
## MT461607.1 0.04478085 0.03440512 0.02244312
## ON112487.1 0.00000000 0.05049792 0.04366021
## OM366050.1 0.05049792 0.00000000 0.03395199
## MW309426.1 0.04366021 0.03395199 0.00000000
```

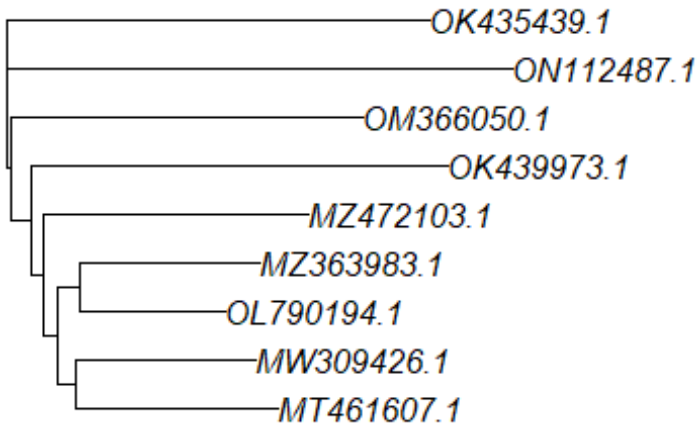
```
table.paint(temp, cleg=0, clabel.row=.5, clabel.col=.5) +
scale_color_viridis()
```




```
## NULL
```

7. Construye un árbol filogenético a partir de la matriz de distancia obtenida e incluye en el árbol los números de accesión de los genomas utilizados, sus nombres o cualquier otra leyenda que te permita indicar la ubicación de ellos en el árbol.

```
virus_tree <- nj(matriz_distancia)
virus_tree <- ladderize(virus_tree)
plot(virus_tree)
```



8. Agrega una interpretación escrita, desde el punto de vista biológico, para cada una de las gráficas incluidas en el reporte y una conclusión general de tu análisis, según el caso de estudio que realizaste. No olvides sustentar tus argumentos con fuentes de información confiables, y las lecturas que realizaste para la realización de este entregable y durante el curso.

Analizando la tabla de escala de grises podemos ver que los casos de Canadá son los que más similitudes en el genoma tienen, y 2 de estas 3 secuencias comparten lugar en el árbol filogenético, esto puede deberse a que como Canadá tuvo pocos casos, no haya tantas mutaciones en los genomas, mientras tanto los genomas de Estados Unidos y México, tienen menos similitud con los de su mismo país y en el árbol se encuentran en ramas diferentes, y algunos teniendo mayor semejanza con la del otro país como el genoma de Ciudad de México y el de Florida, debiéndose a esto a los altos casos que hay en ambos países, y a la migración que hay entre ellos. Es por eso que nuestro árbol 2 de Canadá comparten rama y el otro genoma se encuentra cercano, mientras que ninguno de los de México y Estados Unidos comparten rama por la gran cantidad de casos y mutaciones que hubo en estos países.

Y aunque parezca que hay una gran variación genética dentro de estas graficas de acuerdo con un estudio de la NCBI todas las secuencias de SARS-CoV-2 comparten por al menos 99% de la secuencia y 96% si es una variante como la (Alpha, Beta, Gamma, etc.), y que puede seguir aumentando la diferencia de los casos que vimos al ser un virus de super transmisión.

Bibliografia:

R;, C. T. K. J. M. M. L. H.-L. (n.d.). *[SARS-COV-2 infections in cats, dogs, and other animal species: Findings on infection and data from Switzerland]*. Schweizer Archiv fur Tierheilkunde. Retrieved May 8, 2022, from https://pubmed.ncbi.nlm.nih.gov/34881715/#:~:text=In%20November%202020%2C%20the%20first,been%20recorded%20in%20animals%20worldwide.

AK;, A. A. A. S. A. B. S. (n.d.). *Emerging variants of SARS-COV-2 and novel therapeutics against Coronavirus (COVID-19)*. National Center for Biotechnology Information. Retrieved May 8, 2022, from https://pubmed.ncbi.nlm.nih.gov/34033342/

Gómez-Carballa, A., Bello, X., Pardo-Seco, J., Martinón-Torres, F., & Salas, A. (2020, October). *Mapping genome variation of SARS-COV-2 worldwide highlights the impact of covid-19 super-spreaders*. Genome research. Retrieved May 8, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7605265/

FM;, C. C. G. (n.d.). *Genomic variance of the 2019-ncov coronavirus*. Journal of medical virology. Retrieved May 8, 2022, from https://pubmed.ncbi.nlm.nih.gov/32027036/#:~:text=We%20confirm%20high%20sequence%20similarity,zoonotic%20origin%20of%202019%2DnCoV.