FilogeneticaR_ActividadColaborativa

Andrea Catalina Fernandez Mena - A01197705

04/23/2021

Caso de estudio:

A Novel Coronavirus from Patients with Pneumonia in China, 2019

“In December 2019, a cluster of patients with pneumonia of unknown cause was linked to a seafood wholesale market in Wuhan, China. A previously unknown betacoronavirus was discovered through the use of unbiased sequencing in samples from patients with pneumonia. Human airway epithelial cells were used to isolate a novel coronavirus, named 2019-nCoV, which formed a clade within the subgenus sarbecovirus, Orthocoronavirinae subfamily. Different from both MERS-CoV and SARS-CoV, 2019-nCoV is the seventh member of the family of coronaviruses that infect humans. Enhanced surveillance and further investigation are ongoing. (Funded by the National Key Research and Development Program of China and the National Major Project for Control and Prevention of Infectious Disease in China.)”

El trabajo de China Novel Coronavirus Investigating and Research Team (https://www.nejm.org/doi/full/10.1056/NEJMoa2001017)

Cargaremos las sección azul de SARS-COV, MERS-COV y SARS-COV2 con algunas variantes: 1. “AY508724” SARS coronavirus NS-1, complete genome 2. “AY485277” SARS coronavirus Sino1-11, complete genome 3. “AY390556” SARS coronavirus GZ02, complete genome 4. “AY278489” SARS coronavirus GD01, complete genome 5. “MN908947” Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome 6. “MN985325” Severe acute respiratory syndrome coronavirus 2 isolate 2019-nCoV/USA-WA1/2020, complete genome 7. “MT292571” Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/ESP/Valencia12/2020, complete genome 8. “JX869059” Human betacoronavirus 2c EMC/2012, complete genome

```
virus <- c("JX869059", "AY508724", "MN908947", "AY390556", "AY278489", "MN985325","AY485
277","MT292571")
```

1. Carga las librerías necesarias:

```
library(ape) #ya
library(Biostrings) # Ya, biocmanager
```

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

```
## Warning: package 'BiocGenerics' was built under R version 4.0.5
```

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

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

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## 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 object is masked from 'package:base':
##
##     expand.grid
```

```
## Loading required package: IRanges
```

```
## Loading required package: XVector
```

```
##
## Attaching package: 'Biostrings'
```

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

```r
library(seqinr) # ya
```

```
##
## Attaching package: 'seqinr'
```

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

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

```r
library(adegenet) # ya
```

```
## Loading required package: ade4
```

```
##
## Attaching package: 'ade4'
```

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

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

```
## Registered S3 method overwritten by 'spdep':
##   method   from
##   plot.mst ape
```

```
##
##    /// adegenet 2.1.3 is loaded ////////////
##
##    > overview: '?adegenet'
##    > tutorials/doc/questions: 'adegenetWeb()'
##    > bug reports/feature requests: adegenetIssues()
```

```r
library(ggtree) # ya, biocmanager
```

```
## Registered S3 method overwritten by 'treeio':
##   method     from
##   root.phylo ape
```

```
## ggtree v2.4.1  For help: https://yulab-smu.top/treedata-book/
##
## If you use ggtree in published research, please cite the most appropriate paper(s):
##
##  [36m- [39m Guangchuang Yu. Using ggtree to visualize data on tree-like structures. Cu
rrent Protocols in Bioinformatics, 2020, 69:e96. doi:10.1002/cpbi.96
##  [36m- [39m 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
##  [36m- [39m Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam. ggt
ree: an R package for visualization and annotation of phylogenetic trees with their cova
riates and other associated data. Methods in Ecology and Evolution 2017, 8(1):28-36. do
i: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) # ya, biocmanager
```

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

```
library(viridis) # ya
```

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

```
library(ggplot2) # ya
library(geiger) # ya
library(phytools) # ya
```

```
## Loading required package: maps
```

```
library(phangorn) # ya
```

```
##
## Attaching package: 'phangorn'
```

```
## The following object is masked from 'package:adegenet':
##
##     AICc
```

2. Obtén las secuencias:

```
virus_sequences <- read.GenBank(virus)
```

3. Estructura del DNABin:

```
str(virus_sequences)
```

```
## List of 8
##  $ JX869059: raw [1:30119] 48 88 18 18 ...
##  $ AY508724: raw [1:29732] 18 88 28 28 ...
##  $ MN908947: raw [1:29903] 88 18 18 88 ...
##  $ AY390556: raw [1:29760] 88 18 88 18 ...
##  $ AY278489: raw [1:29757] 18 88 28 28 ...
##  $ MN985325: raw [1:29882] 88 18 18 88 ...
##  $ AY485277: raw [1:29741] 88 18 88 18 ...
##  $ MT292571: raw [1:29782] 88 48 88 18 ...
##  - attr(*, "class")= chr "DNAbin"
##  - attr(*, "description")= chr [1:8] "JX869059.2 Human betacoronavirus 2c EMC/2012, c
omplete genome" "AY508724.1 SARS coronavirus NS-1, complete genome" "MN908947.3 Severe a
cute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome" "AY390556.1
SARS coronavirus GZ02, complete genome" ...
##  - attr(*, "species")= chr [1:8] "Human_betacoronavirus_2c_EMC/2012" "SARS_coronaviru
s_NS-1" "Severe_acute_respiratory_syndrome_coronavirus_2" "SARS_coronavirus_GZ02" ...
```

```
attributes(virus_sequences)
```

```
## $names
## [1] "JX869059" "AY508724" "MN908947" "AY390556" "AY278489" "MN985325" "AY485277"
## [8] "MT292571"
##
## $class
## [1] "DNAbin"
##
## $description
## [1] "JX869059.2 Human betacoronavirus 2c EMC/2012, complete genome"
## [2] "AY508724.1 SARS coronavirus NS-1, complete genome"
## [3] "MN908947.3 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, c
omplete genome"
## [4] "AY390556.1 SARS coronavirus GZ02, complete genome"
## [5] "AY278489.2 SARS coronavirus GD01, complete genome"
## [6] "MN985325.1 Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/hu
man/USA/WA-CDC-WA1/2020, complete genome"
## [7] "AY485277.1 SARS coronavirus Sino1-11, complete genome"
## [8] "MT292571.1 Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/hu
man/ESP/Valencia12/2020, complete genome"
##
## $species
## [1] "Human_betacoronavirus_2c_EMC/2012"
## [2] "SARS_coronavirus_NS-1"
## [3] "Severe_acute_respiratory_syndrome_coronavirus_2"
## [4] "SARS_coronavirus_GZ02"
## [5] "SARS_coronavirus_GD01"
## [6] "Severe_acute_respiratory_syndrome_coronavirus_2"
## [7] "SARS_coronavirus_Sino1-11"
## [8] "Severe_acute_respiratory_syndrome_coronavirus_2"
```

```
names(virus_sequences)
```

```
## [1] "JX869059" "AY508724" "MN908947" "AY390556" "AY278489" "MN985325" "AY485277"
## [8] "MT292571"
```

```
attr(virus_sequences, "species")
```

```
## [1] "Human_betacoronavirus_2c_EMC/2012"
## [2] "SARS_coronavirus_NS-1"
## [3] "Severe_acute_respiratory_syndrome_coronavirus_2"
## [4] "SARS_coronavirus_GZ02"
## [5] "SARS_coronavirus_GD01"
## [6] "Severe_acute_respiratory_syndrome_coronavirus_2"
## [7] "SARS_coronavirus_Sino1-11"
## [8] "Severe_acute_respiratory_syndrome_coronavirus_2"
```

4. Concentraremos en un archivo todas las secuencias:

```
write.dna(virus_sequences,  file ="virus_seqs.fasta", format = "fasta", append =
FALSE, nbcol = 6, colsep = " ", colw = 10)
```

5. Cargamos la secuencias:

```
virus_seq_not_align <- readDNAStringSet("virus_seqs.fasta", format = "fasta")
```

```
## Warning in .Call2("fasta_index", filexp_list, nrec, skip, seek.first.rec, :
## reading FASTA file virus_seqs.fasta: ignored 19891 invalid one-letter sequence
## codes
```

```
virus_seq_not_align
```

```
## DNAStringSet object of length 8:
##     width seq                                               names
## [1] 30119 GATTTAAGTGAATAGCTTGGCTA...AGATGATTTGCAAAAAAAAAAAA JX869059
## [2] 29732 TACCCAGGAAAAGCCAACCAACC...ACAAAAAAAAAAAAAAAAAAAAA AY508724
## [3] 29903 ATTAAAGGTTTATACCTTCCCAG...AAAAAAAAAAAAAAAAAAAAAAA MN908947
## [4] 29760 ATATTAGGTTTTTACCTACCCAG...AGCTTCTTAGGAGAATGACAAAA AY390556
## [5] 29757 TACCCAGGAAAAGCCAACCAACC...AATGACAAAAAAAAAAAAAAAAA AY278489
## [6] 29882 ATTAAAGGTTTATACCTTCCCAG...AGGAGAATGACAAAAAAAAAAAA MN985325
## [7] 29741 ATATTAGGTTTTTACCTACCCAG...AATGACAAAAAAAAAAAAAAAAA AY485277
## [8] 29782 AGATCTGTTCTCTAAACGAACTT...AATTAATTTTAGTAGTGCTATCC MT292571
```

6. Alineamiento de las secuencias: Un alineamiento múltiple de secuencias es un alineamiento de más de dos secuencias. Estas secuencias, como en el caso de los alieamientos por parejas pueden ser ADN, ARN o proteína. Las aplicaciones más habituales de los alineamientos múltiples son:

- la reconstrucción filogenética,
- el análisis estructural de proteínas,
- la búsqueda de dominios conservados y
- la búsqueda de regiones conservadas en promotores. En todos los casos los algoritmos de alineamiento múltiple asumen que las secuencias que estamos alineando descienden de un antepasado común y lo que intentamos hacer es alinear las posiciones homólogas.

```
virus_seq_not_align <- OrientNucleotides(virus_seq_not_align)
```

```
## ============================================================================
==============================================================================
======================
##
## Time difference of 0.34 secs
```

```
virus_seq_align <- AlignSeqs(virus_seq_not_align)
```

```
## Determining distance matrix based on shared 11-mers:
## ================================================================================
##
## Time difference of 0.93 secs
##
## Clustering into groups by similarity:
## ================================================================================
##
## Time difference of 0.02 secs
##
## Aligning Sequences:
## ================================================================================
##
## Time difference of 35.31 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 0.01 secs
##
## Realigning Sequences:
## ================================================================================
##
## Time difference of 0.03 secs
##
## Alignment converged - skipping remaining iteration.
```

7. Visualizar el resultado del alineamiento:

```
BrowseSeqs(virus_seq_align, highlight=0)
```

8. Guardar el resultado:

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

9. Obtener el nuevo archivo:

```
virus_aligned <- read.alignment("virus_seq_align.fasta", format = "fasta")
```

10. Crear una matriz de distancia:

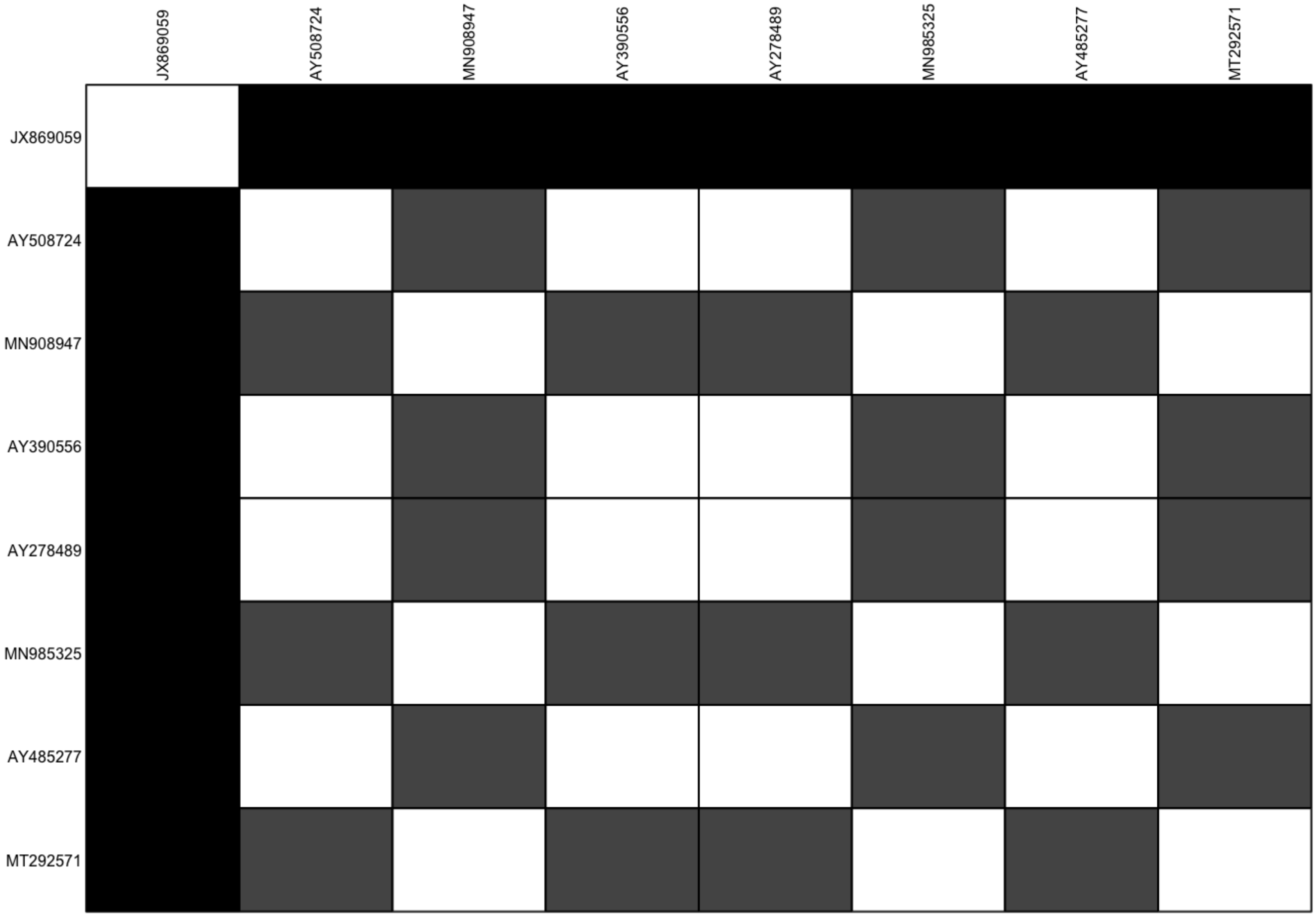
```
matriz_distancia <- dist.alignment(virus_aligned, matrix = "similarity")

matriz_distancia
```

```
##            JX869059   AY508724   MN908947   AY390556   AY278489   MN985325
## AY508724 0.68076930
## MN908947 0.67969320 0.45043503
## AY390556 0.68073819 0.04019136 0.45022264
## AY278489 0.68078171 0.04262006 0.45076862 0.03068169
## MN985325 0.67966775 0.45050336 0.01001972 0.45022264 0.45080657
## AY485277 0.68064111 0.02657962 0.45032932 0.03890661 0.04301505 0.45036726
## MT292571 0.67982185 0.45108226 0.01533106 0.45071147 0.45138535 0.01419380
##            AY485277
## AY508724
## MN908947
## AY390556
## AY278489
## MN985325
## AY485277
## MT292571 0.45091773
```

11. Visualiza la matriz de distancia: donde sombras más oscuras de gris significan una mayor distancia

```
temp <- as.data.frame(as.matrix(matriz_distancia))
table.paint(temp, cleg=0, clabel.row=.5, clabel.col=.5) + scale_color_viridis()
```



```
## NULL
```

12. Creación del árbol con el paquete ape:

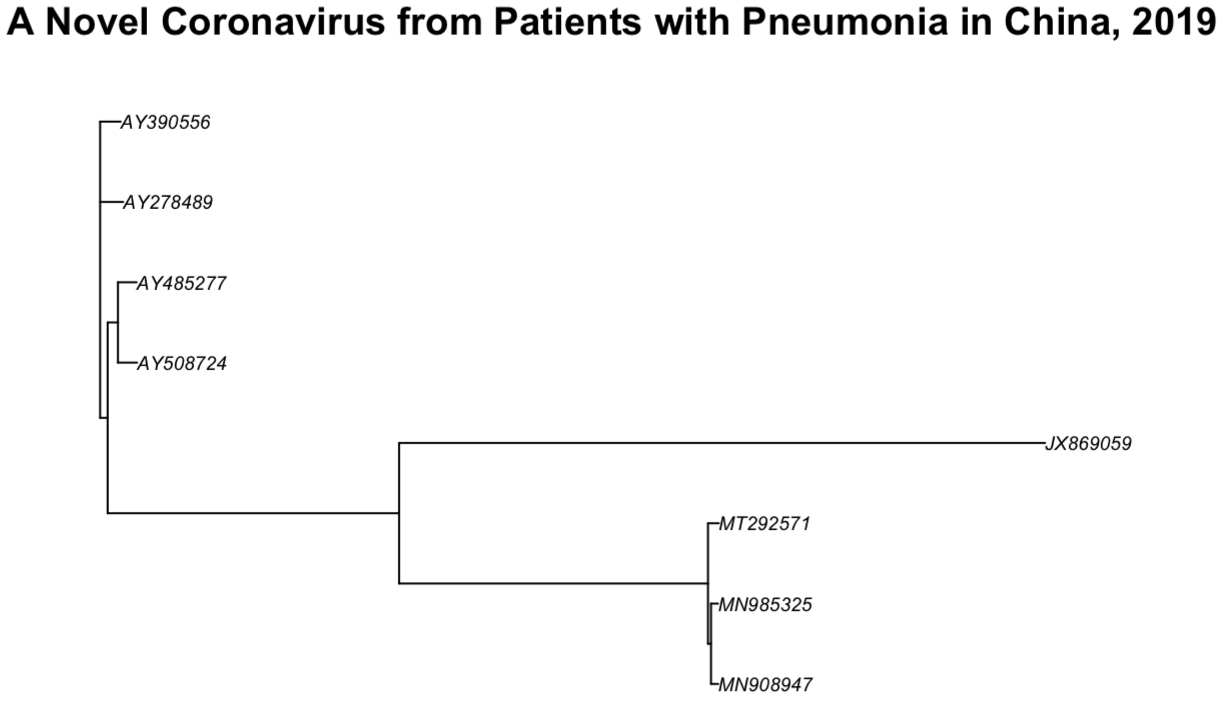
```
virus_tree <- nj(matriz_distancia)
class(virus_tree)
```

```
## [1] "phylo"
```

```
virus_tree <- ladderize(virus_tree)
```

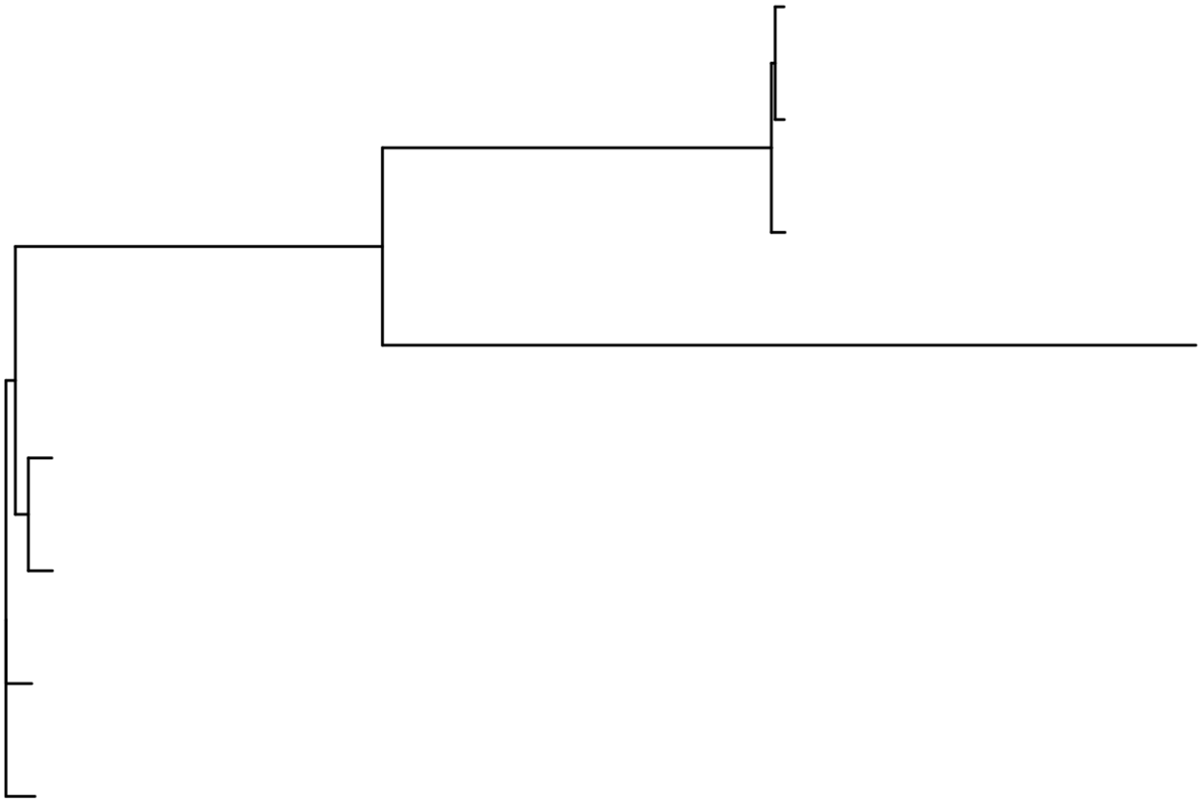
13. Plot del árbol:

```
plot(virus_tree, cex = 0.6)
title("A Novel Coronavirus from Patients with Pneumonia in China, 2019")
```

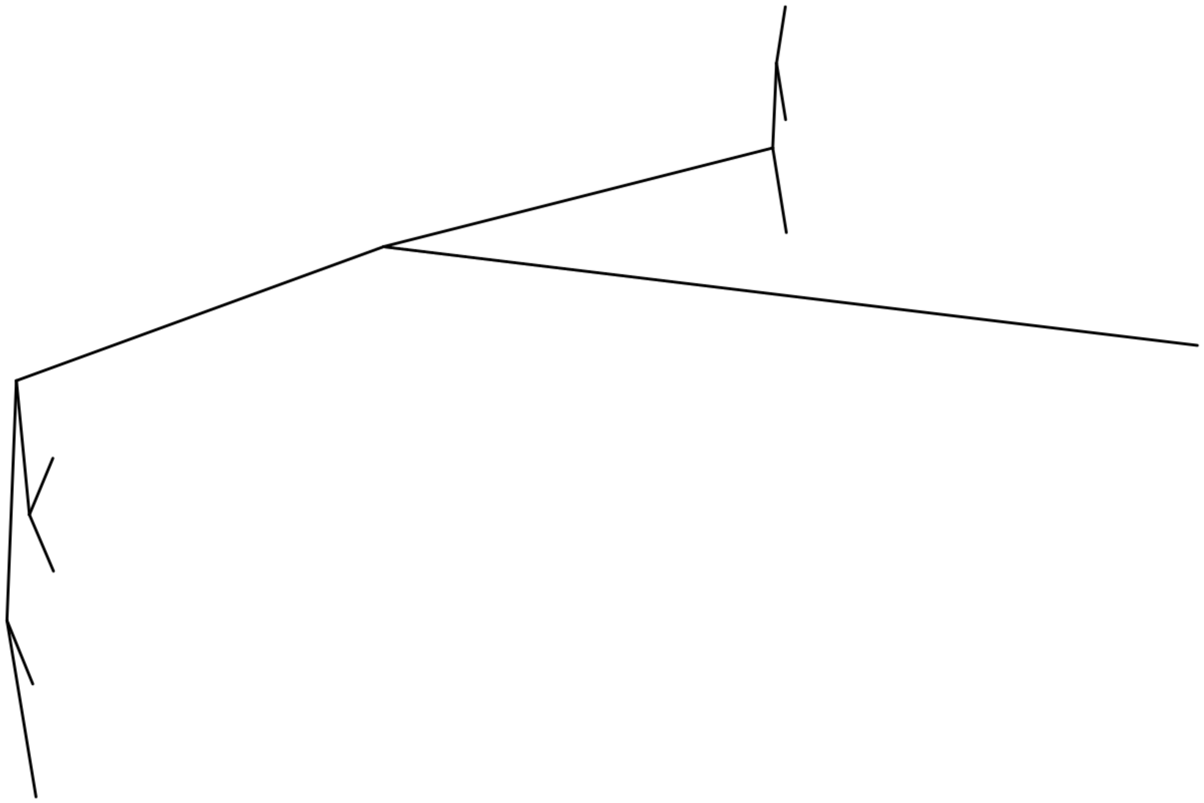

A Novel Coronavirus from Patients with Pneumonia in China, 2019

14. Plot utilizando ggtree que es parte de ggplot:

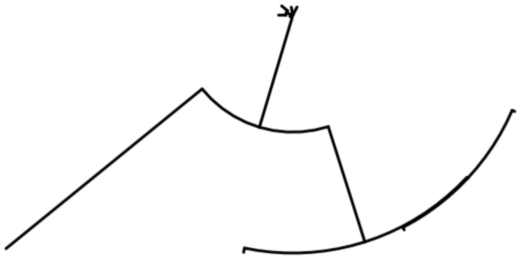
```
ggtree(virus_tree)
```



```
ggtree(virus_tree, layout="slanted")
```

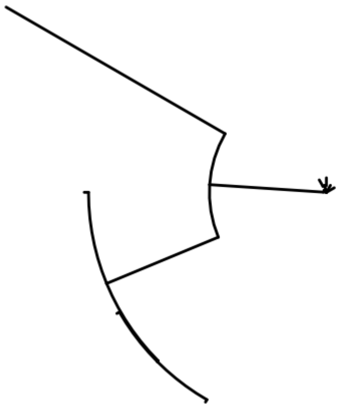


```
ggtree(virus_tree, layout="circular")
```

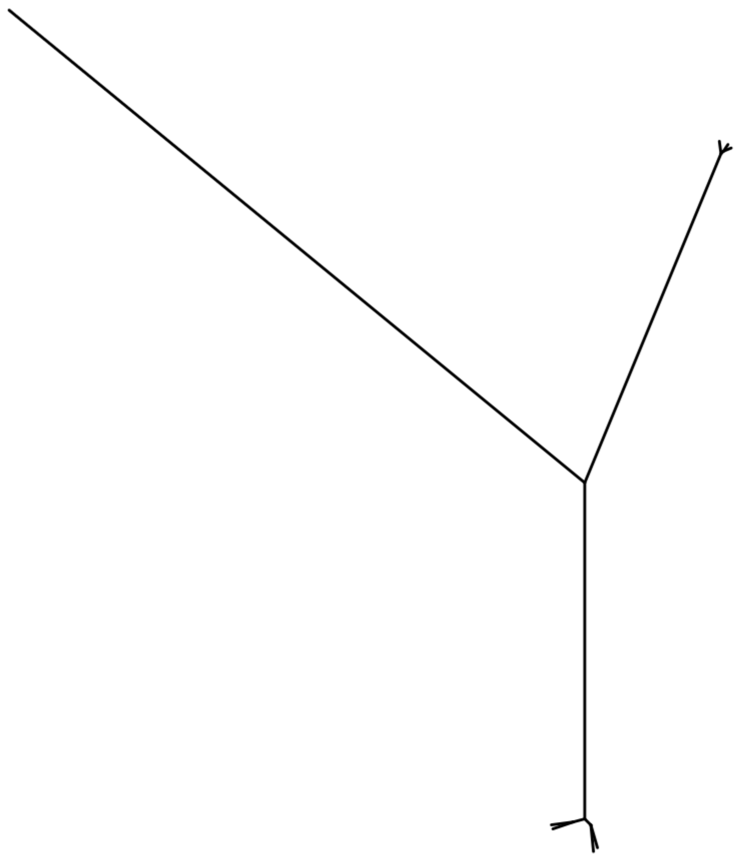


```
ggtree(virus_tree, layout="fan", open.angle=120)
```

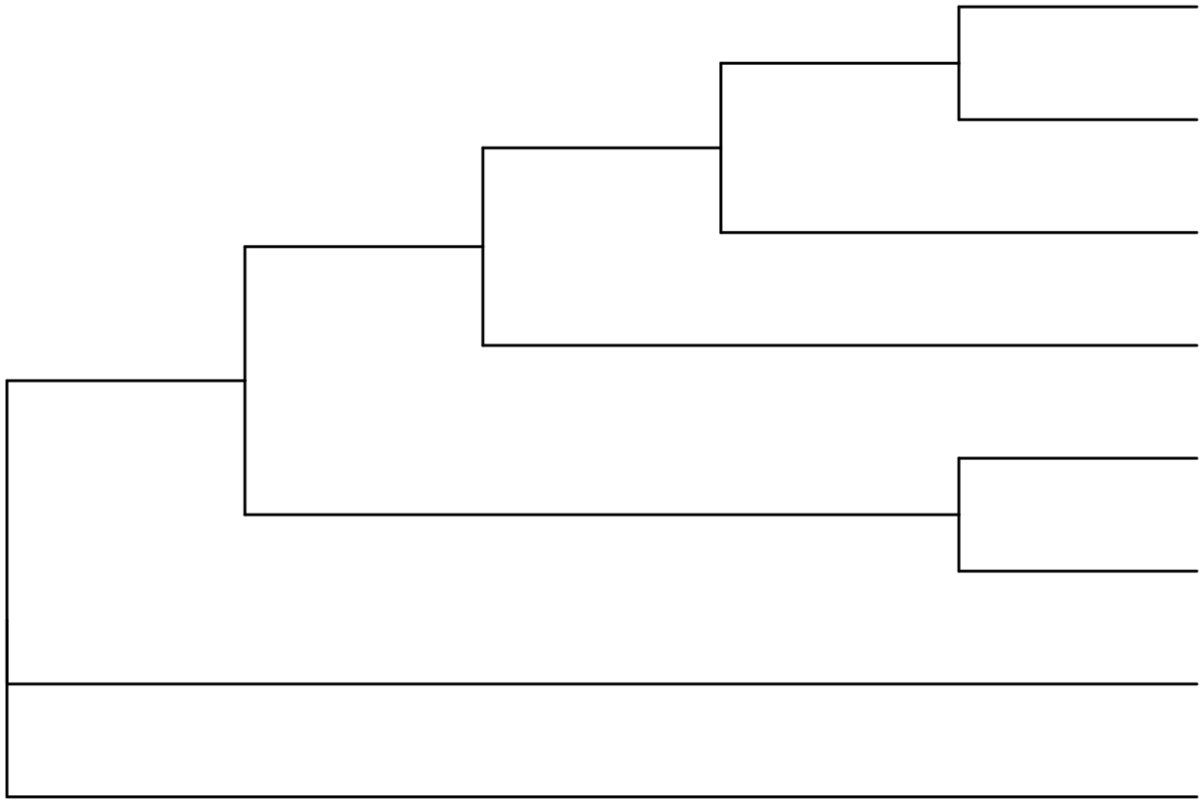
```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```



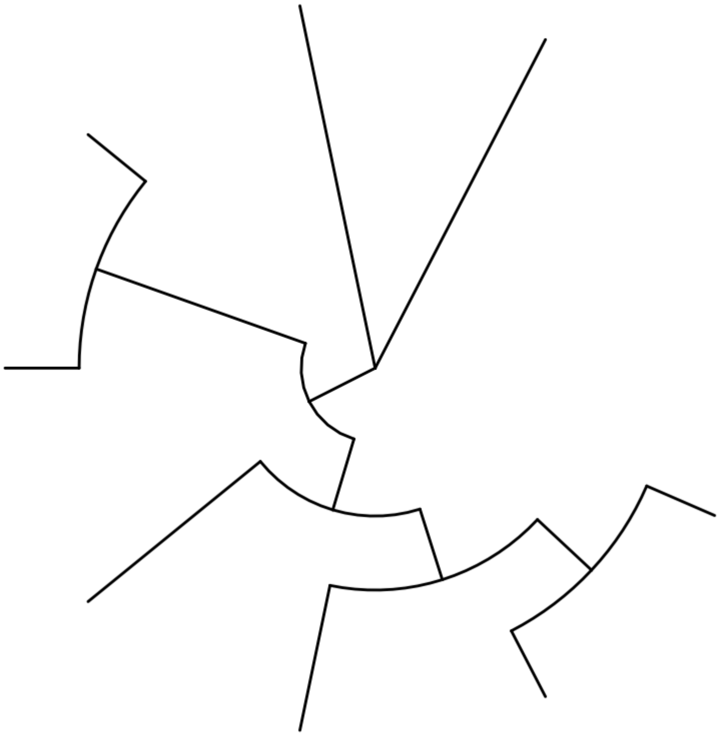
```
ggtree(virus_tree, layout="equal_angle")
```



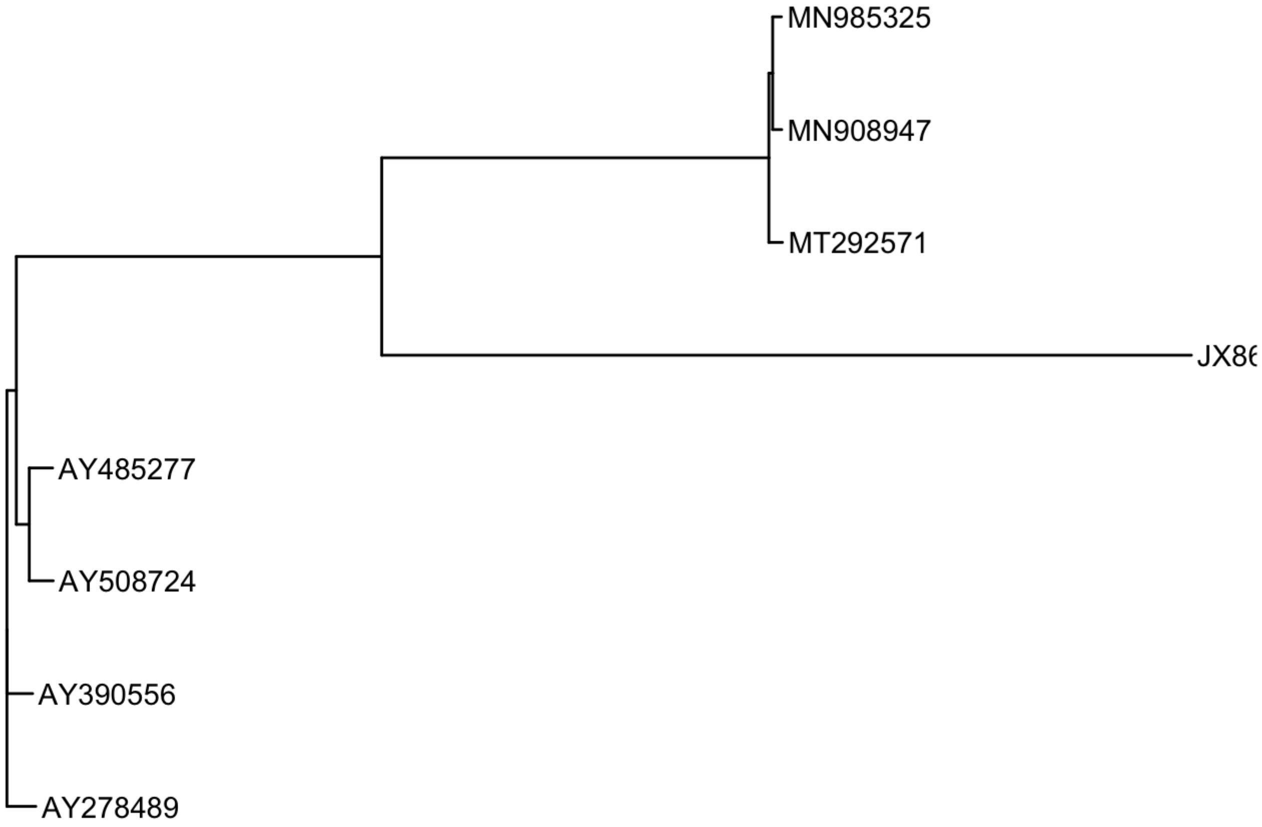
```
ggtree(virus_tree, branch.length='none')
```



```
ggtree(virus_tree, branch.length='none', layout='circular')
```



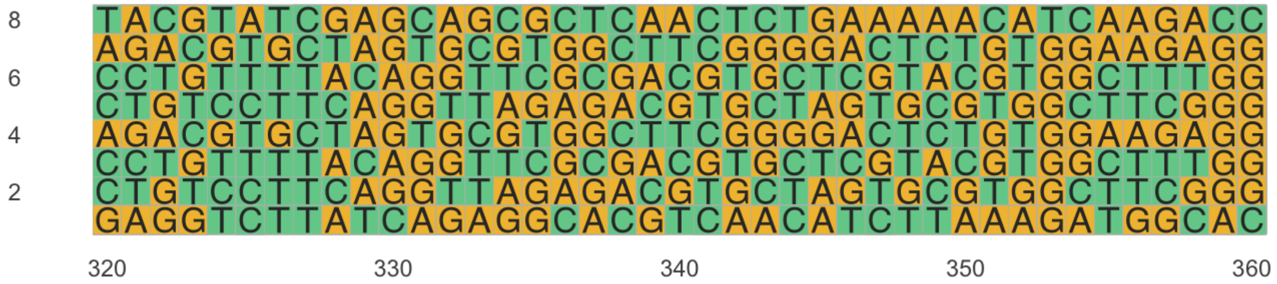
```
#Ahora si, el bueno
ggtree(virus_tree ) + geom_tiplab()
```



15. Visualiza el alineamiento de las secuencias:

```
library(ggmsa)
ggmsa(virus_seq_not_align, 320, 360, color = "Chemistry_AA")
```

```
## Warning in rbind(c("G", "A", "T", "T", "T", "A", "A", "G", "T", "G", "A", :
## number of columns of result is not a multiple of vector length (arg 2)
```



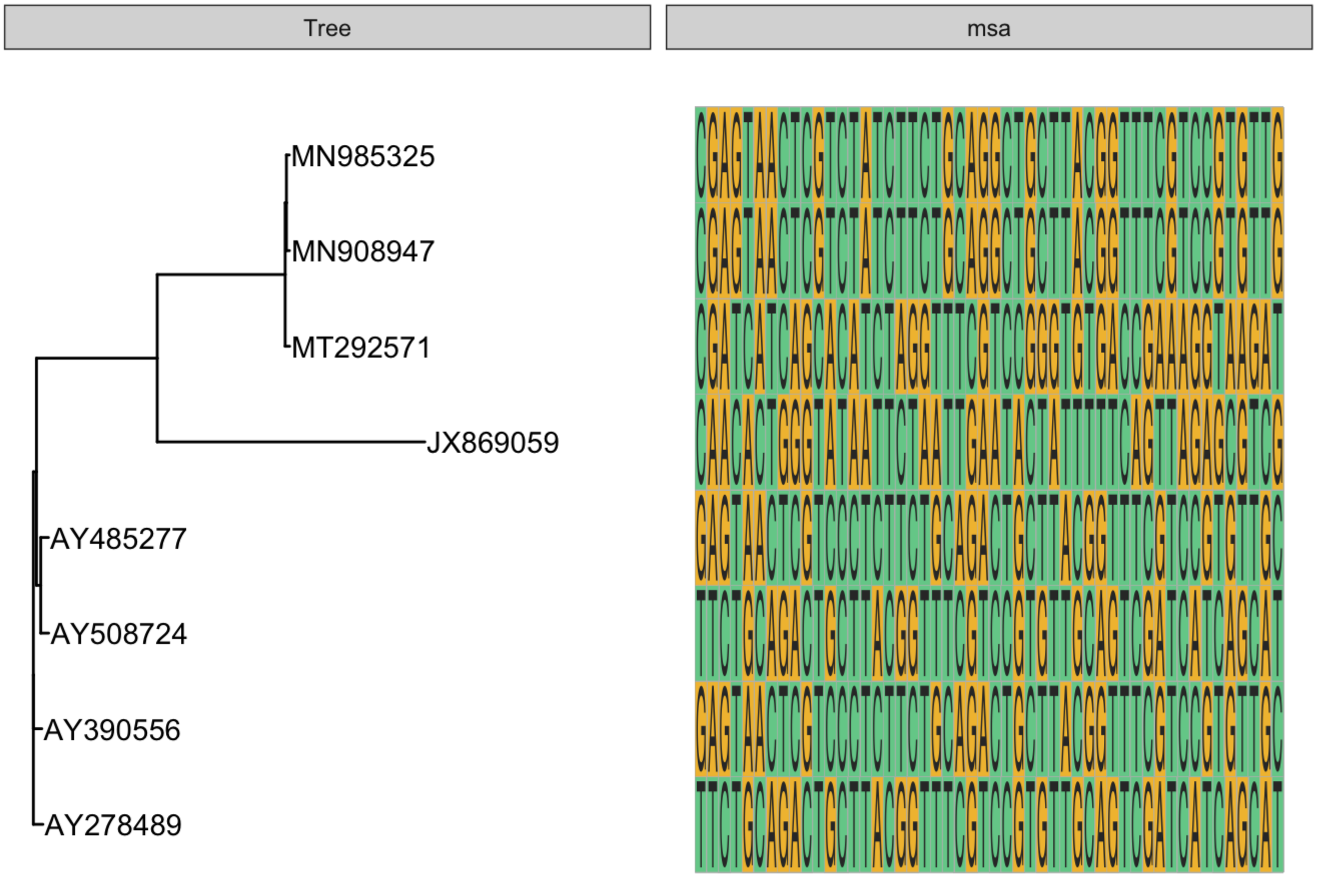
16. Combina el árbol filogenético con el alineamiento de las secuencias:

```
plot_virus <- ggtree(virus_tree ) + geom_tiplab()

data = tidy_msa(virus_seq_not_align, 164, 213)
```

```
## Warning in rbind(c("G", "A", "T", "T", "T", "A", "A", "G", "T", "G", "A", :
## number of columns of result is not a multiple of vector length (arg 2)
```

```
plot_virus + geom_facet(geom = geom_msa, data = data,  panel = 'msa', color = "Chemistry
_AA") +
    xlim_tree(1)
```



Citar paquetes de R

Citar los paquetes, módulos y softwares que usaste para tu análisis es importante, tanto desde una perspectiva de reproducibilidad (las rutinas estadísticas a menudo se implementan de diferentes maneras por diferentes paquetes, lo que podría explicar ligeras discrepancias en los resultados. Decir “Hice esto usando esta función de ese paquete, versión 1.2.3”es una forma de protegerse al ser claro acerca de lo que ha encontrado haciendo lo que ha hecho), pero también de reconocer el trabajo y el tiempo que las personas dedicaron a crear herramientas para otros (a veces a expensas de su investigación propia).

citation(“packagename”)

```
citation("dplyr")
```

```
##
## To cite package 'dplyr' in publications use:
##
##   Hadley Wickham, Romain François, Lionel Henry and Kirill Müller
##   (2021). dplyr: A Grammar of Data Manipulation. R package version
##   1.0.5. https://CRAN.R-project.org/package=dplyr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {dplyr: A Grammar of Data Manipulation},
##     author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller},
##     year = {2021},
##     note = {R package version 1.0.5},
##     url = {https://CRAN.R-project.org/package=dplyr},
##   }
```

Conclusión:

1. En esta sesión revisamos el proceso para crear árboles filogenéticos utilizando secuencias de ADN de diferentes virus, es momento de trabajar en la evidencia 2 y finalizar la materia.
2. No olvides citar todos los paquetes que usas en tu trabajo final utilizando citation.

##Fuentes de paquetes instalados ## For adegenet install.packages("adegenet")

DECIPHER

if (!requireNamespace("BiocManager", quietly=TRUE)) install.packages("BiocManager") BiocManager::install("DECIPHER")

for viridis

install.packages("viridis") library(viridis)

Other librarys installed

BiocManager::install("Biostrings") BiocManager::install("ggtree") BiocManager::install("DECIPHER")