Análisis de biología computacional - BT1013

Instituto Tecnológico de Estudios Superiores de Monterrey

Desirée Espinosa Contreras - A01425162

Dulce Nahomi Bucio Rivas - A01425284

Introducción

La pandemia de COVID-19, provocada por el virus SARS-CoV-2, tuvo su inicio en el año 2020, cuando la Organización Mundial de la Salud declaró la epidemia originalmente situada en China como una emergencia de salud internacional, para posteriormente ser declarada como pandemia. En la actualidad, la pandemia continúa en un estado activo, habiendo alcanzado una cifra estimada de 676,609,955 casos de coronavirus a nivel mundial, con un total de 6,881,955 lamentables muertes, según la Universidad Johns Hopkins. Asimismo, en México, se estima un total de 7,483,444 casos diagnosticados y 333,188 muertes por COVID-19.

Propuesta

Para el proyecto, haremos un análisis de las secuencias del virus en México de febrero a diciembre de 2020, y de febrero a diciembre de 2021.

Objetivo

Encontrar los cambios que experimentó el SARS-CoV-2 en su estructura durante estos periodos de tiempo. Analizaremos principalmente los genes S, M, E y N, los cuales han demostrado tener un papel importante en la construcción de la estructura del virus, especialmente el gen S.

Hipótesis

Encontraremos una cantidad relevante de mutaciones en el gen S, la cual podría implicarse que dichas mutaciones estuvieron involucradas en el aumento de la transmisibilidad del virus y/o en la efectividad de las vacunas. Además, podríamos obtener información valiosa sobre la evolución del virus.

Marco Teórico

Para nuestro enfoque de estudio es importante definir qué es una variante, la cual es un genoma viral (código genético) que puede incluir una o más mutaciones. La primera variación detectada fue la Alfa B.1.1.7, la cual fue observada por primera vez en el suroeste de Inglaterra y se caracteriza por ser entre 30% y 70% más contagiosa que el virus original.

Existen variables características descubiertas en distintas regiones del mundo que fueron principalmente bajo observación, tales son:
Variante Alfa B.1.1.7: Descubierta en Reino Unido, fue la variable dominante a nivel mundial hasta el año 2021 con una frecuencia mayor al 80% en regiones tales como Europa y América del Norte.

Variante Beta B.1.351: Detectada en Sudáfrica, siendo mayoritariamente dominante en esta región pero en otras no rebasando una frecuencia del 20%. Caracterizada únicamente por tener mayor carga vírica pero no se encontró ni se demostró nada más acerca de su transmisibilidad.

Variante Gamma P1: Procedente de la Amazonas Brasileña se convirtió en la más dominante en América del Sur mientras que en otras regiones la frecuencia no supera el 10%, siendo caracterizada por ser más contagiosa.

Variante DELTA B.1.617.2: Encontrada en la India, esta variante dio señales de ser más contagiosa y resistente a vacunas y tratamientos. Fue la variante responsable del colapso del sistema sanitario en ese país, así como de su propagación por 26 países de la región europea de la OMS.

Desarrollo

Iniciaremos el análisis de las secuencias. Tras leer las secuencias seleccionadas con el período de tiempo dado, haremos un análisis de cómo mutó cada una de las iniciales, o sea las del 2020 y las del 2021.

```
length(mx_2020)
```

```
## [1] 60
```

```
length(mx_2021)
```

```
## [1] 2400
```

Consideremos que los 11 genes del SARS-COV-2 son representados en los archivos dados como el ORF1ab separado, por lo que tenemos un vector de 12 genes por cada secuencia. Entonces, tenemos que analizar 5 secuencias del 2020 contra 200 del 2021.

Usaremos la siguiente serie de funciones para encontrar las mutaciones necesarias:

```
aminoacido = function(codon) {
  aminoacid = switch(
    codon,
    "GCU" = "A", "GCC" = "A", "GCA" = "A", "GCG" = "A",
    "CGU" = "R", "CGC" = "R", "CGA" = "R", "CGG" = "R",
    "AGA" = "R", "AGG" = "R", "AAU" = "N", "AAC" = "N",
    "GAU" = "D", "GAC" = "D", "UGU" = "C", "UGC" = "C",
    "CAA" = "Q", "CAG" = "Q", "GAA" = "E", "GAG" = "E",
    "GGU" = "G", "GGC" = "G", "GGA" = "G", "GGG" = "G",
    "CAU" = "H", "CAC" = "H", "AUU" = "I", "AUC" = "I",
    "AUA" = "I", "UUA" = "L", "UUG" = "L", "CUU" = "L",
    "CUC" = "L", "CUA" = "L", "CUG" = "L", "AAA" = "K",
    "AAG" = "K", "AUG" = "M", "UUU" = "F", "UUC" = "F",
    "CCU" = "P", "CCC" = "P", "CCA" = "P", "CCG" = "P",
    "UCU" = "S", "UCC" = "S", "UCA" = "S", "UCG" = "S",
    "AGU" = "S", "AGC" = "S", "ACU" = "T", "ACC" = "T",
    "ACA" = "T", "ACG" = "T", "UGG" = "W", "UAU" = "Y",
    "UAC" = "Y", "GUU" = "V", "GUC" = "V", "GUA" = "V",
    "GUG" = "V"
  )
  return (aminoacid)
}
```

Ésta primera función llamada “aminoacido” devuelve la abreviación establecida de los distintos aminoácidos según una entrada dada. Esta función es posteriormente utilizada en la siguiente función que muestra las mutaciones para indicar el cambio de proteína.

```
mutaciones = function(arnOriGenX, arnMexaGenX, inicioOri, gen){
  # Obtener las posiciones donde hay nucleótidos diferentes
  diff = which(arnOriGenX != arnMexaGenX)
  for (x in diff){
    # Formar un string con el cambio de nucleótido
    muta = paste(arnOriGenX[x],"to",arnMexaGenX[x], sep="")
    # Calcular el inicio del codón
    inicioCodon = x - (x-1)%%3
    # Calcular la posición global
    posGlobal = inicioCodon + inicioOri
    # Calcular el número del codón
    numCodon = as.integer((x-1)/3+1)
    # Calcular el inicio del codón
    indiceCodon = x - (x%%3) + 1
    # Concatena los nucleotidos para crear el codon a comparar
    codonOri = paste(arnOriGenX[indiceCodon], arnOriGenX[indiceCodon+1], arnOriGenX[indiceCodon+2], sep
    codonMex = paste(arnMexaGenX[indiceCodon], arnMexaGenX[indiceCodon+1], arnMexaGenX[indiceCodon+2],
    codon = paste(codonOri,"to",codonMex, sep="") #Compara el cambio
    # Aplica la función "aminoácido" para obtener la abreviación
    aminoOri = aminoacido(codonOri)
    aminoMex = aminoacido(codonMex)
    # Resume el cambio con el aminoácido obtenido
    amino = paste(aminoOri, numCodon, aminoMex, sep="")
    # Agrega los datos al data frame
    obs = list(muta, posGlobal, codon, amino, gen)
    mutations[nrow(mutations)+1, ] = obs
  }
  return(mutations)
}
```

Después, con esta función "mutaciones", se extraen las distintas observaciones necesarias para el análisis, como la posición de inicio de cada gen, que se obtiene a partir de las anotaciones del archivo que contiene la secuencia original del virus. Otras de las observaciones que devuelve esta función es la posición donde se da el cambio de nucleótido, y con ello se obtiene la posición del codón, así como las mutaciones con la abreviatura correspondiente. Además de devolver dichas observaciones, indica también a qué gen corresponde cada una de las anteriormente mencionadas.

Nótese que la función "aminoácidos" devuelve la abreviación establecida de los distintos aminoácidos según una entrada dada. Esta función es posteriormente util

Para usar la función en todas las secuencias aplicamos el siguiente ciclo:

```
# Iteramos el conjunto de genes obtenidos del 2020
for (k in seq(1, length(mx_2020))) {
  arnMexa2020 = as.vector(mx_2020[[k]])
  # Convertimos a ARN
  arnMexa2020[arnMexa2020=="t"] = "u"
  arnMexa2020 = toupper(arnMexa2020)
  # Iteramos el conjunto de genes obtenidos del 2021
  for (j in seq(1,length(mx_2021))){
    # Solamente hacemos la comparación si son el mismo gen
    if (k%%12 == j%%12){
      arnMexa2021 = as.vector(mx_2021[[j]])
      arnMexa2021[arnMexa2021 == "t"] = "u"
```

```
      arnMexa2021 = toupper(arnMexa2021)
      # Esto lo usamos para que tome las anotaciones del gen 12 del original
      vect = j%%12
      if (vect == 0){
        vect = 12
      }
      if (j==2) next
      # Extraemos las anotaciones y características que nos importan
      # como el número de inicio del gen y su nombre
      anotaciones = attr(original[[vect]], "Annot")
      atributos = unlist(strsplit(anotaciones,"\\[|\\]|:|=|\\.|\\("));
      geneName = atributos[which(atributos=="gene")+1]
      if (length(which(atributos=="join"))>0) inicioGen = as.integer(atributos[which(atributos=="join")+
      else inicioGen = as.integer(atributos[which(atributos=="location")+1])
      # Usamos la función "mutaciones" con los parámetros adquiridos en este ciclo
      mutations = mutaciones(arnMexa2020, arnMexa2021, inicioGen, geneName)
    }
  }
}
```

Donde empezamos iterando a las muestras del 2020 para comparar todas y cada una de ellas con las del 2021 que correspondan al mismo lugar módulo 12 para así comparar cada gen correspondientemente. Cabe resaltar que se cambió el nucleotido "t" por "u" para convertir la secuencia en ARN. Obtuvimos una serie de datos de todas las mutaciones de la siguiente longitud:

```
str(mutations)
```

```
## 'data.frame':    158501 obs. of  5 variables:
##  $ mutation   : chr  "AtoU" "GtoU" "UtoC" "UtoC" ...
##  $ pos_global : num  1302 8178 12786 18486 20931 ...
##  $ cambioCodon: chr  "GAGtoGAG" "CAAtoCAA" "AUAtoACA" "AUAtoAUA" ...
##  $ cambioAmino: chr  "E346E" "Q2638Q" "I4174T" "I6074I" ...
##  $ gen        : chr  "ORF1ab" "ORF1ab" "ORF1ab" "ORF1ab" ...
```

Observemos que existen muchas observaciones, entre las cuales muchas de ellas pueden aparecer la mínima cantidad de veces y pueden ser insignificantes para su estudio posterior, por lo que vamos a filtrar entre los que aparecieron más de 10 veces.

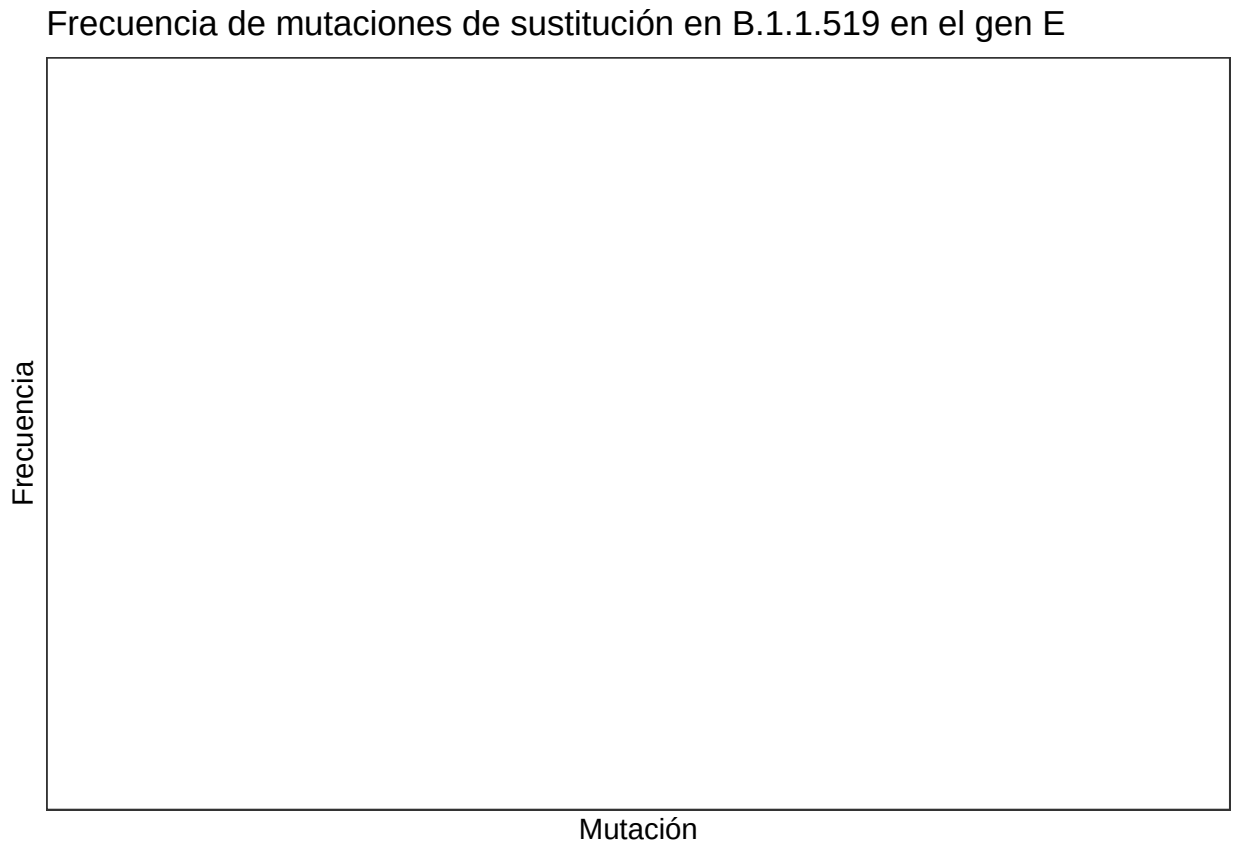
```
mutaciones_filtradas = filter(
  summarise(
    select(
      # agrupa por columna de aminoacido
      group_by(mutations, cambioAmino),
      mutation:gen
    ),
    mutation = first(mutation),
    cambioCodon = first(cambioCodon),
    gen = first(gen),
    # establece una cuenta para verificar el número de registros que tenemos
    Cuenta = n()
  ),
  Cuenta>10
)
```

Como los genes de nuestro interés son los estructurales, entonces los separaremos:

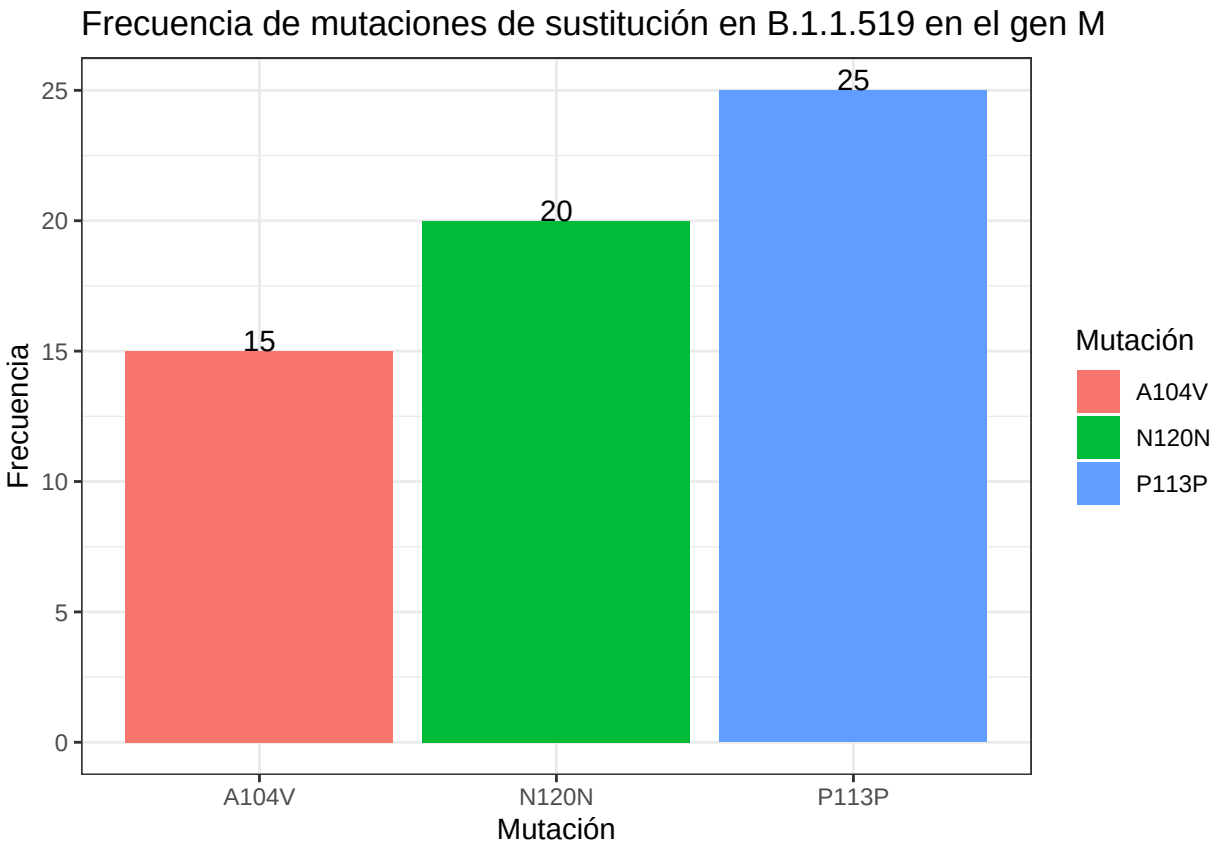
```
mutations_S = subset(mutaciones_filtradas, gen == "S")
mutations_M = subset(mutaciones_filtradas, gen == "M")
mutations_E = subset(mutaciones_filtradas, gen == "E")
mutations_N = subset(mutaciones_filtradas, gen == "N")
```

Para visualizar de mejor manera estos datos, se presentan las siguientes gráficas:

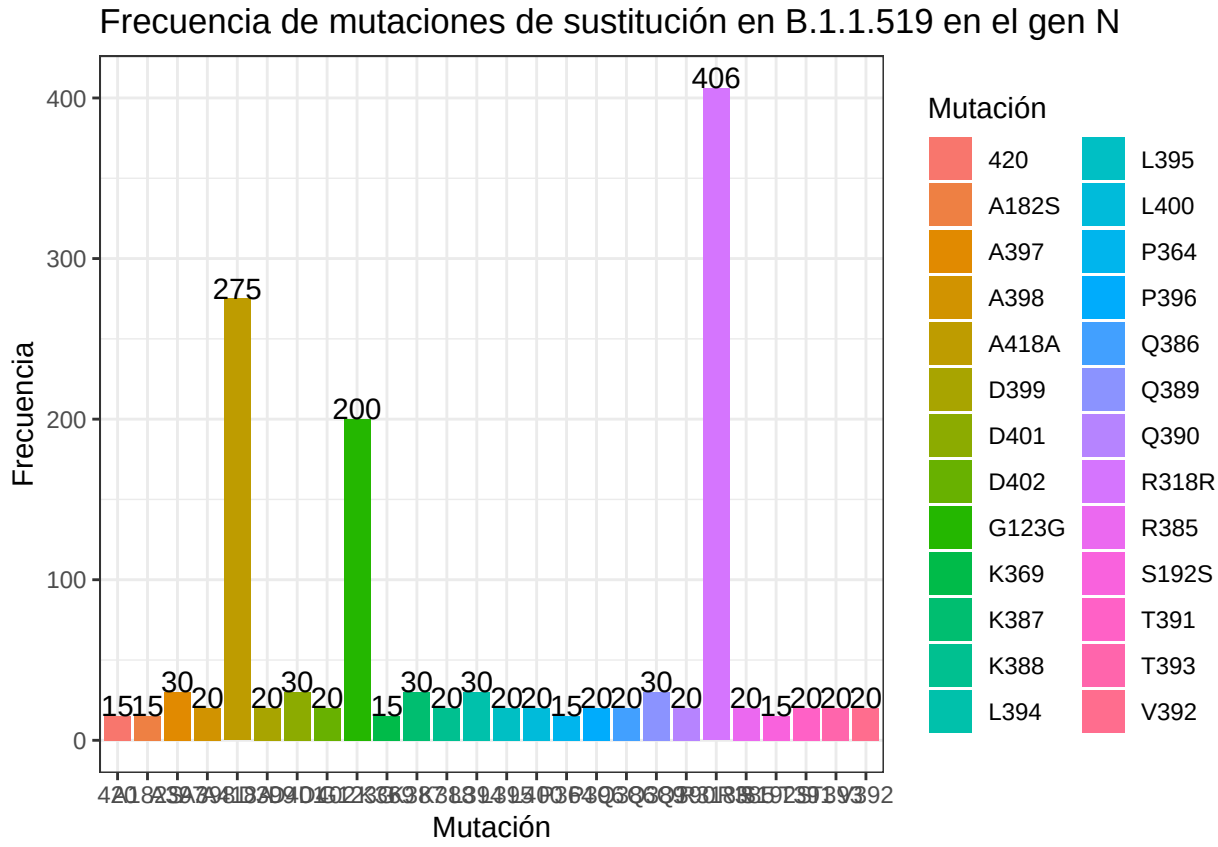
```
grafica_E
```



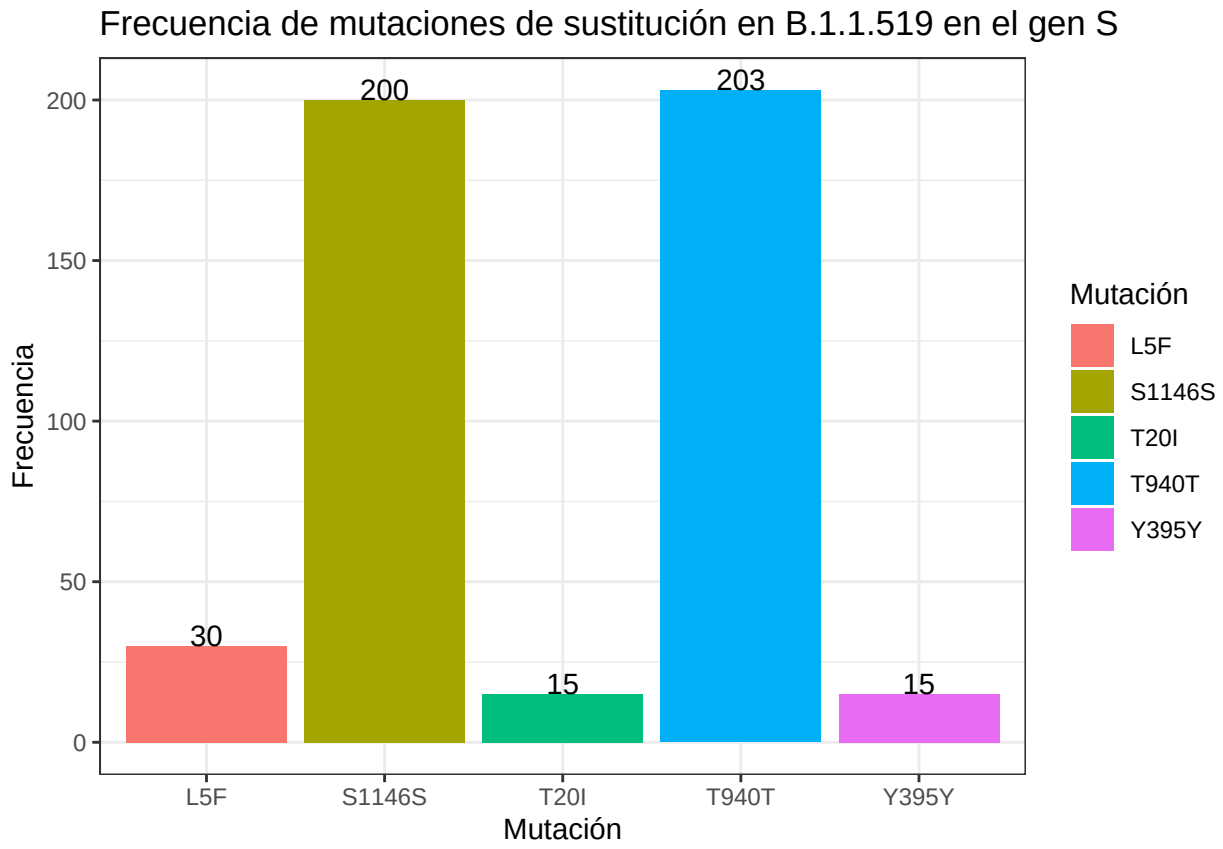

```
grafica_M
```


Frecuencia de mutaciones de sustitución en B.1.1.519 en el gen M
Frecuencia
25
20
15
10
5
0
15
20
25
A104V
N120N
P113P
Mutación
Mutación
A104V
N120N
P113P

```
grafica_N
```


Frecuencia de mutaciones de sustitución en B.1.1.519 en el gen N
Frecuencia
Mutación
0
100
200
300
400
406
275
200
Mutación
420
A182S
A397
A398
A418A
D399
D401
D402
G123G
K369
K387
K388
L394
L395
L400
P364
P396
Q386
Q389
Q390
R318R
R385
S192S
T391
T393
V392

```
grafica_S
```



Bibliografía

– Ferrer, R. (2020). Pandemia por COVID-19: el mayor reto de la historia del intensivismo. Medicina Intensiva, 44(6), 323-324. https://doi.org/10.1016/j.medin.2020.04.002
– COVID-19 Map - Johns Hopkins Coronavirus Resource Center. (s.f.). Johns Hopkins Coronavirus Resource Center. https://coronavirus.jhu.edu/map.html
– Updated working definitions and primary actions for SARSCOV2 variants. (s.f.). Retrieved April 25, 2023, from https://www.who.int/publications/m/item/historical-working-definitions-and-primary-actions-for-sars-cov-2-variants
<Pulido, S. (2021). Variantes Covid por países: Un análisis completo y experiencias en Reino Unido y ee.uu con Los Test Genéticos. Retrieved April 25, 2023, from https://economiadelasalud.com/topics/difusion/variantes-covid-por-paises-un-analisis-completo-y-experiencias-en-reino-unido-y-ee-uu-con-los-test-geneticos/ =======