Universidad de Zaragoza

Bioinformática

Memoria del trabajo de prácticas

Alineamientos y árboles filogenéticos con secuencias de SARS-COV-2 de Aragón

Autor:

Adrián Martín Marcos 756524

Zaragoza, España

Curso 2020 – 2021

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Introducción

En esta última práctica de la asignatura se ha retomado el análisis de secuencias de SARS-CoV-2, pero esta vez con otras técnicas, como el estudio del índice de conservación de las posiciones y el uso de árboles filogenéticos.

Para ello, se han desarrollado nuevos scripts y se han mejorado los implementados en las anteriores sesiones. Además, se han usado las herramientas software vistas durante el semestre para cada tipo de problema, como es el caso de Clustal Omega y MAFFT para el alineamiento de secuencias, y MEGA para la reconstrucción de árboles filogenéticos, a la cual hay que sumarle RAxML y FastTree, que se utilizarán por primera vez.

El objetivo, al igual que en la práctica 3, es buscar mutaciones relevantes con respecto a la secuencia de referencia de Wuhan, y más concretamente, aquellas compatibles con la variante británica del coronavirus.

Las secuencias utilizadas han sido dadas como material de la práctica y tienen como origen el repositorio GISAID. Se corresponden con secuencias de SARS-CoV-2 obtenidas en Aragón entre los meses de marzo, abril y de septiembre a diciembre de 2020, y en enero y febrero de 2021.

Índice

Introducción **I**

1. Multialineamiento **1**

1.1. MAFFT . . . 1

1.1.1. Método automático . . . 1

1.1.2. Método progresivo . . . 1

1.2. Método de la estrella . . . 2

1.3. Clustal Omega . . . 2

1.3.1. Comando por defecto . . . 2

1.3.2. Opción de profiling con la secuencia de referencia . . . 2

1.4. Análisis de los resultados . . . 3

1.4.1. Criterios de calidad . . . 3

1.4.2. Comparación de los multialineamientos . . . 3

2. Detección de mutaciones **4**

2.1. Detección de posibles casos de cepa británica . . . 4

2.2. Cálculo de los índices de conservación . . . 4

2.2.1. Implementación . . . 4

2.3. Análisis de los índices de conservación . . . 5

2.3.1. Uso del script . . . 5

2.3.2. Conclusiones . . . 6

3. Construcción de una filogenia **7**

3.1. Métodos basados en distancia: UPGMA y vecinos cercanos . . . 7

3.2. Métodos basados en secuencias: máxima parsimonia . . . 7

3.3. Métodos basados en secuencias: máxima verosimilitud . . . 7

3.3.1. Herramienta FastTree . . . 7

3.3.2. Herramienta RAxML . . . 8

3.4. Comparación de los resultados . . . 9

3.4.1. Análisis del mejor árbol . . . 9

Referencias **12**

1. Multialineamiento

En este primer apartado de la práctica se ha realizado el alineamiento de todas las secuencias dadas, usando la de Wuhan como referencia.

Como paso previo, se ha añadido a las secuencias de Aragón la secuencia de referencia de la cepa británica, con el fin de poder hacer análisis más precisos en los posteriores apartados:

```
cat gisaid_hcov-19_2021_03_23_B117.fasta \
    gisaid_hcov-19_2021_03_23_aragonHC.fasta \
    > secuenciaB117secuenciasAragon.fasta
```

A continuación, se comentan las pruebas realizadas con las distintas herramientas, así como los resultados obtenidos.

1.1. MAFFT

MAFFT es una herramienta para crear multialineamientos usando la transformada de Fourier, y que ya se estudió en detalle en la práctica 3. Por ello, en esta ocasión, se ha tratado de reproducir las pruebas que se hicieron entonces.

1.1.1. Método automático

En primer lugar, se ha usado el modo auto, con el cual MAFFT selecciona automáticamente una estrategia de alineamiento en base al tamaño de las secuencias.

La orden por línea de comandos es la siguiente:

```
time mafft --auto \
           --addfragments secuencias/secuenciaB117secuenciasAragon.fasta \
           secuencias/RefSeqWuhan.fasta > alineamiento_mafft-auto-addfragments.fasta
```

Tras solo diez segundos de ejecución, el alineamiento se había completado.

1.1.2. Método progresivo

Al igual que en la práctica 3, lo siguiente que se ha probado son métodos progresivos, pues es la familia de técnicas más rápida entre los posibles algoritmos a utilizar en MAFFT. Concretamente, y al igual que entonces, se ha usado la opción FFT-NS-2 para alinear las secuencias.

La orden por línea de comandos es la siguiente:

```
time mafft --retree 2 \
           --addfragments secuencias/secuenciaB117secuenciasAragon.fasta \
           secuencias/RefSeqWuhan.fasta > alineamiento_mafft-retree2-addfragments.fasta
```

No obstante, en esta ocasión la herramienta ha superado las dos horas de ejecución sin terminar el alineamiento. Por ello, se ha decidido detenerla.

Al igual que sucedió en la práctica 3, no se han llegado a probar más opciones de MAFFT, pues se sabe a priori que su tiempo de ejecución es superior al de los métodos progresivos.

1.2. Método de la estrella

Después se ha probado con el script desarrollado en la práctica 3 que implementa el método de la estrella. Éste ha sido mejorado con respecto a la entrega, eliminando ciertas instrucciones que eran innecesarias y haciendo que el código se compile con el máximo nivel de optimización posible. Además, ahora se ejecuta con cuatro hilos en vez de con tres, pues se ha podido ver que tras los cambios da mejores resultados.

Las órdenes por línea de comandos son las siguientes:

```
export JULIA_NUM_THREADS=4
./estrellaMSA.jl -r secuencias/RefSeqWuhan.fasta \
                 secuencias/secuenciaB117secuenciasAragon.fasta \
                 -o alineamiento-estrella.fasta
```

Tras algo más de un cuarto de hora, las secuencias estaban alineadas.

1.3. Clustal Omega

Clustal Omega es una herramienta para crear multialineamientos que utiliza técnicas de clustering y alineamiento progresivo, y que ya se estudió en la práctica 2. En la práctica 3 se volvió a usar para compararla con MAFFT, y restultó ser la perdedora con diferencia. No obstante, se produjo un error por mi parte, y es que entonces no ajusté el número de hilos de la herramienta al número de procesadores lógicos de la máquina, por lo que en esta ocasión se ha vueto a usar con la configuración adecuada, con la esperanza de que mejoren sus resultados.

1.3.1. Comando por defecto

En primer lugar, se prueba con la configuración por defecto, pues en la práctica 2 fue con la que se obtuvo uno de los mejores tiempos de ejecución.

Dado que con esta configuración Clustal Omega trata de realizar un alineamiento de todas las secuencias con todas, se ha tenido que preparar un fichero de entrada especial en el que la secuencia de referencia de Wuhan está incluida como una más, situada concretamente la primera de ellas. Las órdenes por línea de comandos necesarias para realizar esta prueba son las siguientes:

```
cat RefSeqWuhan.fasta secuenciaB117secuenciasAragon.fasta > clustalo_input.fasta
clustalo -i clustalo_input.fasta \
         -o alineamiento_clustalo-default.fasta \
         -v --threads=$(nproc)
```

El resultado ha sido el mismo que en la práctica 3, y a las dos horas se ha detenido la ejecución de la herramienta, ya que todavía no había terminado.

1.3.2. Opción de profiling con la secuencia de referencia

Tras los resultados obtenidos en la prueba anterior, se decidió volver a probar la configuración usada en la práctica 3, que era:

```
clustalo --profile1 RefSeqWuhan.fasta \
         -i secuenciaB117secuenciasAragon.fasta \
```

```
-o alineamiento_clustalo-profile.fasta \
-v --threads=$(nproc)
```

En esta ocasión, contradiciendo lo que se esperaba, la herramienta terminó la fase de construcción de k-tuplas antes de las dos horas de ejecución, por lo que se decidió dejarle más tiempo para ver si lograba acabar el alineamiento. No obstante, tras una hora más en la que solo había avanzado un 10 % de la fase que estaba aplicando, se decidió interrumpir el proceso.

1.4. Análisis de los resultados

Una vez concluidas las pruebas, se procede a analizar los resultados obtenidos en cada una de ellas.

1.4.1. Criterios de calidad

Por una parte, y al igual que en la práctica 3, se ha usado la métrica del *sum of pairs* de cada multialineamiento para evaluarlo. Para ello se han utilizado los scripts *obtenerMatrizPuntuacion.jl* y *calcularSP.jl*, que ya se entregaron en la práctica 3, solo que ahora están ligeramente modificados para ser más eficientes (el cálculo de las distancias está paralelizado, y en ambos scripts se aplica un nivel de optimización 3 en lugar de dejarlo por defecto en 2).

Además, se ha implementado un script adicional que evalúa la longitud de la secuencia más larga antes del alineamiento con respecto a la longitud final de éste. La semántica con la que interpretar los resultados es que a una menor diferencia entre el alineamiento y la secuencia inicialmente más larga, de mayor calidad será, pues menos gaps ha sido necesario introducir. Para realizar los cálculos necesarios se ha desarrollado el script *calcularLongitudesMSA.jl*, que muestra ambas longitudes y la diferencia entre ellas.

1.4.2. Comparación de los multialineamientos

A continuación, se muestra una tabla con el tiempo de ejecución y las métricas mencionadas para cada una de las configuraciones probadas de las herramientas:

	Tiempo ejecución	**Sum of Pairs (SP)**	**Diferencia de longitud**
MAFFT auto	14 s	29578.995	6
MAFFT retree	$>$ 2 h	undefined	undefined
Método de estrella	16 min 12 s	29578.018	9
Clustal Omega default	$>$ 3 horas	undefined	undefined
Clustal Omega profile	$>$ 2 horas	undefined	undefined

Tabla 1: Tabla comparativa de los programas de alineamiento probados

Si bien los resultados del alineamiento en estrella son prácticamente iguales a los de MAFFT, se usará el alineamiento obtenido con MAFFT en los siguientes apartados.

2. Detección de mutaciones

En este segundo apartado se ha llevado a cabo un análisis de las secuencias alineadas, con el fin de determinar cuáles de sus posiciones son las que más variabilidad presentan.

2.1. Detección de posibles casos de cepa británica

Como ya se ha comentado en la introducción, uno de los objetivos de la búsqueda de mutaciones es tratar de inferir si existen secuencias que puedan pertenecer a la variante británica. Por ello, para tener una visión preliminar, se usa el script *detectarB117.jl*, implementado en la práctica 3:

```
./detectarB117.jl -a alineamiento.fasta > B117.txt
```

Como extraer conclusiones a partir de la salida del script era muy complicado (pues muestra el análisis de las mutaciones en posiciones concretas, y no un resumen total), se ha implementado un script en bash, *analizarResultadosB117.bash*, que procesa esa salida para mostrar un resultado más conciso, indicando cuántas de las mutaciones compatibles con la variante británica están presentes en cada una de las secuencias.

El resultado es que las secuencias *1007859|2020-12-28*, *1225128|2021-01-27*, *1225154|2021-01-28*, *1261469|2021-02-12* y *1261470|2021-02-12* presentan todas las mutaciones que se corresponden con la variante británica, y las secuencias *1059975|2021-01-25*, *1059978|2021-01-26* y *1059979|2021-01-25* cuentan con 18 de las 19.

2.2. Cálculo de los índices de conservación

Con el fin de estudiar de manera analítica la variabilidad de las bases en cada una de las posiciones, se calcula el índice de conservación, que es una medida de cuánto cambia el valor de una misma posición en las secuencias sobre las que se calcula.

2.2.1. Implementación

Se ha decidido que el script siempre calcule el índice de conservación de todas las posiciones, es decir, sin dar la posibilidad de definir rangos concretos sobre los que hacerlo. El motivo es que se ha implementado otro script que a partir de todos los índices de conservación, permite llevar a cabo un análisis con los de aquellas posiciones que resulten de interés (véase el siguiente apartado). De esa forma, separando el cálculo de los índices y su análisis en scripts diferentes, se evita tener que recalcularlos cada vez que se requiera de ellos.

La implementación se basa en el algoritmo descrito en el guión de prácticas, y supone aplicar el método de entropía que se describe en el artículo dado como referencia [1].

No se han hecho optimizaciones más allá del uso intensivo de funciones base de Julia siempre que era posible (p.ej., la función *sum*). No obstante, el bucle principal de la función *indicesConservacion* es fácilmente paralelizable, pero se ha considerado que con un tiempo de ejecución de 10 segundos, como es el caso, no merece la pena tratar de hacer mejoras.

Una vez implementado el script, se ha comprobado que su salida era correcta utilizando la herramienta JalView. Ésta permite visualizar un diagrama de barras que mide el consenso de los valores de cada posición de un multialineamiento, por lo que se ha comprobado que las posiciones en las

que el índice de conservación no era cero coincidían con aquellas en las que el consenso no era absoluto.

2.3. Análisis de los índices de conservación

Con el fin de estudiar los índices de conservación calculados, se ha implementado otro script, *analizarIndicesConservacion.jl*, que permite obtener los valores máximos de índice de conservación entre aquellas posiciones que se especifiquen. Además permite realizar una gráfica de barras que muestra los valores de índice de conservación para cada posición del intervalo, marcando con puntos los valores máximos mostrados.

2.3.1. Uso del script

El script tiene dos parámetros obligatorios: el número de valores máximos a mostrar y el fichero con los índices de conservación.

Para restringir los cálculos a un rango concreto de posiciones, se puede usar el flag *-r*, que permite introducir las posiciones a usar como extremos del intervalo. Adicionalmente, existen flags que predefinen las posiciones correspondientes a los genes del SARS-CoV-2 que se propone estudiar (*–ORF1A*, *–Spike*, *–ORF3A* y *–N*). En caso de no especificarse ninguna opción relativa a las posiciones a analizar, los cálculos se aplican a todas ellas.

Opcionalmente, se puede usar la opción *-p* (o *–plot*), que especifica que se muestre la gráfica descrita antes.

A modo de ejemplo, se muestra el resultado de la siguiente ejecución del script:

```
./analizarIndicesConservacion.jl -m 19 indiceConservacion-mafft.csv -p
```

Se ha buscado que la gráfica resultante se pareciese a la gráfica *Diversity* del sitio web NextStrain (la figura [6] muestra una comparativa de ambas1).

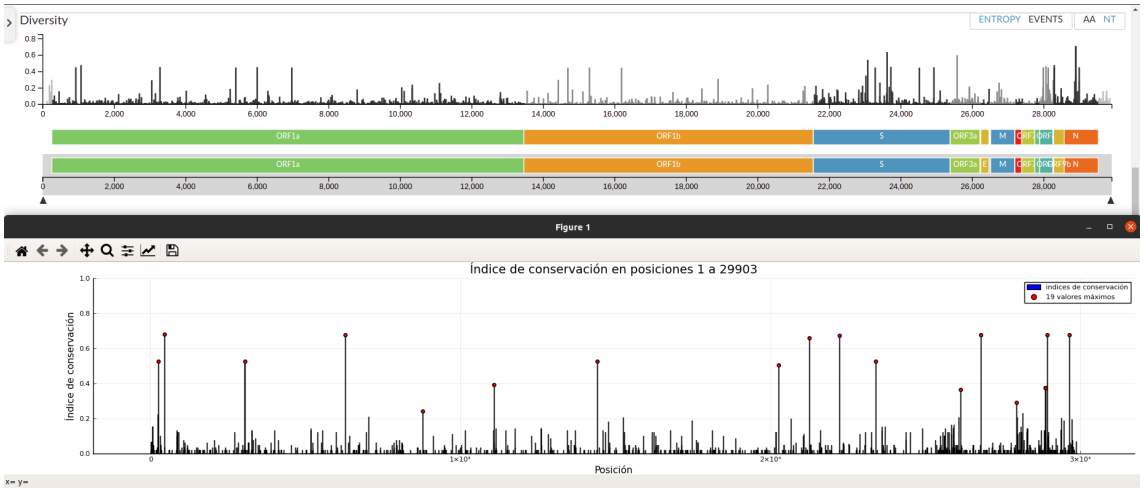

Figura 1: Captura de la gráfica generada (parte inferior) superpuesta con la gráfica Diversity de NextStrain

Para una información más detallada del uso, véase la ayuda del script (flag *-h*).

2.3.2. Conclusiones

Los análisis realizados se han basado en el estudio de aquellas posiciones en las que se encuentran las mutaciones que caracterizan la cepa británica.

Por ello, en el estudio de cada gen, se ha considerado un número de valores máximos igual o mayor al número de posiciones que analiza en él el script *detectarB117.jl*, con el fin de determinar si las posiciones con una mayor variabilidad se corresponden o no con las de las mutaciones de la cepa británica.

En el caso del análisis del gen Spike, el uso que se ha hecho del script ha sido el siguiente:

```
./analizarIndicesConservacion.jl -m 13 \
                                 indicesConservacion.csv \
                                 --Spike
```

Una vez obtenidos los valores máximos, se han comparado con las posiciones de las mutaciones de la variante B117, y se tiene lo siguiente:

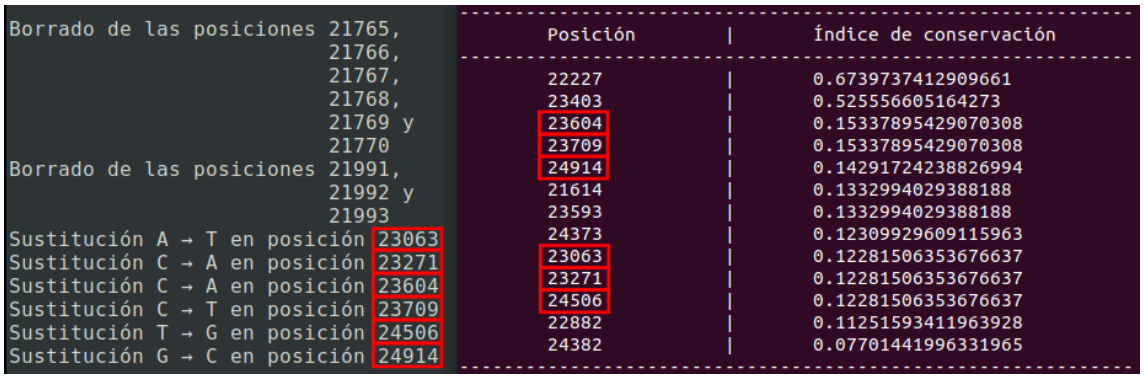

Figura 2: Captura de la salida del script comparada con la lista de mutaciones de la variante británica en el gen Spike

Se han recuadrado en rojo aquellas posiciones de más variabilidad que se corresponden con mutaciones de esta cepa.

Es llamativo que no aparezca ninguna de las posiciones relacionadas con mutaciones de borrado, pero hay que tener en cuenta que el cálculo del índice de conservación omite todos los valores que no sean A, C, G o T, por lo que con esta técnica no se puede tener información con respecto a ellas.

Los resultados para este gen son similares a los observados en los otros analizados; hay posiciones que presentan una mayor variabilidad y no se corresponden con las mutaciones de la variante británica, si bien en aquellas posiciones que se corresponden con las mutaciones de esta cepa sí que se observa variabilidad.

Una posible explicación para las posiciones que varían tanto es que los cambios en ellas tengan efectos poco significativos en la actividad del ser vivo. Eso quiere decir que el hecho de que algunas de las mutaciones de la cepa británica no aparezcan como posiciones de alta variabilidad da lugar a pensar que esos cambios puedan repercutir seriamente en el comportamiento del virus. No obstante, convendría hacer un análisis de los codones para ratificar esa hipótesis, estudiando la codificación de las proteínas a la que dan lugar las mutaciones.

3. Construcción de una filogenia

En este tercer apartado se van a usar árboles filogenéticos para analizar las secuencias. El objetivo es obtener un árbol que permita entender su historia evolutiva.

Si bien las herramientas solo pueden basarse en la similitud o diferencia entre las secuencias, el hecho de que cada una de ellas incorpore la fecha en la que fue tomada será de gran ayuda a la hora de interpretar los árboles obtenidos y valorarlos cualitativamente.

Se describen a continuación los diferentes métodos que se han probado.

3.1. Métodos basados en distancia: UPGMA y vecinos cercanos

Se ha utilizado la herramienta MEGA, estudiada en la práctica 4, para construir árboles usando métodos de distancia.

Tanto en el caso de UPGMA como de vecinos cercanos, se ha optado por la p-distancia como modelo de sustitución. Además, aunque no era necesario (pues a los métodos de distancia no les afecta el orden de las secuencias), se ha aplicado bootstrap, concretamente, generando 100 reordenaciones de las secuencias, con el fin de obtener unos resultados estadísticamente más robustos (entendiendo que en este contexto puede que sea determinante la aleatoriedad para resolver empates). Como consecuencia, se ha tomado el árbol de consenso como resultado.

3.2. Métodos basados en secuencias: máxima parsimonia

Al igual que en la práctica 4, se ha usado la herramienta MEGA para aplicar este método.

En este caso es casi obligado usar bootstrap, pues los árboles obtenidos son muy dependientes del orden inicial de las secuencias. Por ello, se han generado 100 reordenaciones y se ha tomado el árbol de consenso como resultado.

Cabe destacar que este método ha sido con diferencia el que menos tiempo de ejecución ha requerido.

3.3. Métodos basados en secuencias: máxima verosimilitud

Aunque en MEGA existe la posibilidad de usar métodos de máxima verosimilitud, se han usado en esta ocasión los programas RAxML y FastTree.

Cabe destacar que ambas herramientas han sido añadidas como paquetes a los repositorios de la versión 20 de Ubuntu, por lo que no ha sido necesario compilar ni descargar directamente ninguna de ellas.

3.3.1. Herramienta FastTree

Esta herramienta usa por defecto el modelo evolutivo CAT.

Las modelos matemáticos probados han sido los siguientes:

```
# Jones-Taylor-Thornton (por defecto)
fasttree -nt < alineamiento.fasta > MV-fasttree-jtt.nwk

# Generalized time-reversible
fasttree -gtr -nt < alineamiento.fasta > MV-fasttree-gtr.nwk

# Le and Gascuel
fasttree -lg -nt < alineamiento.fasta > MV-fasttree-lg.nwk

# Whelan and Goldman
fasttree -wag -nt < alineamiento.fasta > MV-fasttree-wag.nwk
```

Es relevante mencionar que durante su ejecución advierte de que tanto ésta como herramientas similares, basadas en métodos de máxima verosimilitud, pueden no ser apropiadas para la generación de árboles filogenéticos de secuencias tan estrechamente relacionadas, como lo son las que se le está dando como entrada.

3.3.2. Herramienta RAxML

Esta herramienta cuenta con multitud de "saboresz opciones de ejecución. La versión empleada ha sido la AVX que cuenta con soporte a pthreads.

Los modelos evolutivos probados han sido los siguientes:

```
# Modelo CAT
raxmlHPC-PTHREADS-AVX -s alineamiento.fasta \
                      -n MV-raxml-GTRCAT.nwk \
                      -m GTRCAT -T $(nproc) -p 1

# Modelo CAT usando corrección de sesgo de verificación
raxmlHPC-PTHREADS-AVX -s alineamiento.fasta \
                      -n MV-raxml-GTRCATI.nwk \
                      -m GTRCATI -T $(nproc) -p 1

# Modelo CAT empleando proporciones de sitios invariables para la estimación
raxmlHPC-PTHREADS-AVX -s alineamiento.fasta \
                      -n MV-raxml-ASC_GTRCAT.nwk \
                      -m ASC_GTRCAT -T $(nproc) \
                      -p 1 --asc-corr=lewis

# Modelo GAMMA
raxmlHPC-PTHREADS-AVX -s alineamiento.fasta \
                      -n MV-raxml-GTRGAMMA.nwk \
                      -m GTRGAMMA -T $(nproc) -p 1

# Modelo GAMMA usando corrección de sesgo de verificación
raxmlHPC-PTHREADS-AVX -s alineamiento.fasta \
                      -n MV-raxml-GTRGAMMAI.nwk \
                      -m GTRGAMMAI -T $(nproc) -p 1

# Modelo GAMMA empleando proporciones de sitios invariables para la estimación
raxmlHPC-PTHREADS-AVX -s alineamiento.fasta \
                      -n MV-raxml-ASC_GTRGAMMA.nwk \
```

```
-m ASC_GTRGAMMA -T $(nproc) \
-p 1 --asc-corr=lewis
```

En este caso, las pruebas en las que se realizaba una estimación empleando proporciones de sitios invariables han fallado, ya que los valores de muchas de las posiciones eran iguales en todas las secuencias, y para poder seguir adelante la herramienta solicitaba eliminarlas.

3.4. Comparación de los resultados

A continuación, se muestra una tabla que recoge los resultados de tiempo de ejecución y puntuación de todas las pruebas realizadas:

	Tiempo ejecución	**Puntuación***
MEGA UPGMA	507.449 s	0.023 sbl
MAFFT Neighbor-joining	498.151 s	0.026 sbl
MEGA Maximum Parsimony	21.444 s	794 tl
Fasttree JTT CAT	102.77 s	-49900.521 lk
Fasttree GTR CAT	220.90 s	-48470.996 lk
Fasttree LG CAT	102.81 s	-49900.521 lk
Fasttree WAG CAT	102.81 s	-49900.521 lk
RAxML GTRCAT	99.226 s	-48706.82 lk
RAxML GTRCATI	117.29 s	-48691.127 lk
RAxML GTRGAMMA	100.43 s	-48696.799 lk
RAxML GTRGAMMAI	144.84 s	-48689.842 lk

Tabla 2: Tabla comparativa de los programas de reconstrucción de árboles filogenéticos probados

Nótese que las puntuaciones son solo comparables cuando se corresponen con la misma métrica. Es el caso del *likelihood* para los métodos de máxima verosimilitud, o del *sum of branch lengths* en los métodos de distancia. Es por ello que se han tomado los árboles con una mejor puntuación de cada tipo de métrica y se han comparado entre ellos cualitativamente.

En el caso de máxima verosimilitud, se tiene que a un mayor valor de *likelihood* de más calidad es el árbol. Dado que todos los valores son negativos, una puntuación mayor se corresponde con menor valor absoluto, por lo que el mejor de los árboles obtenidos atendiendo a esta métrica es el generado con Fasttree usando el modelo evolutivo CAT y el modelo matemático *generalized time-reversible*.

En cuanto a métodos de distancia, se tiene que a menor valor de suma de longitud, mejor es el árbol. Por ello el mejor resultado que se ha obtenido es el de UPGMA.

En el caso de máxima parsimonia, no hay con qué comparar (es el único puntuado con la métrica *tree length*), luego se toma directamente el árbol resultante.

Tras visualizar esos árboles con la herramienta online iTOOL [4], se ha llegado a la conclusión de que el que mejor explica la historia evolutiva es el construido con UPGMA.

3.4.1. Análisis del mejor árbol

Se muestra a continuación una captura del árbol UPGMA 3 que será analizado.

Se ha marcado en color rojo la secuencia de referencia de Wuhan, y en color azul aquellas que en el apartado anterior habían sido catalogadas como cepa británica.

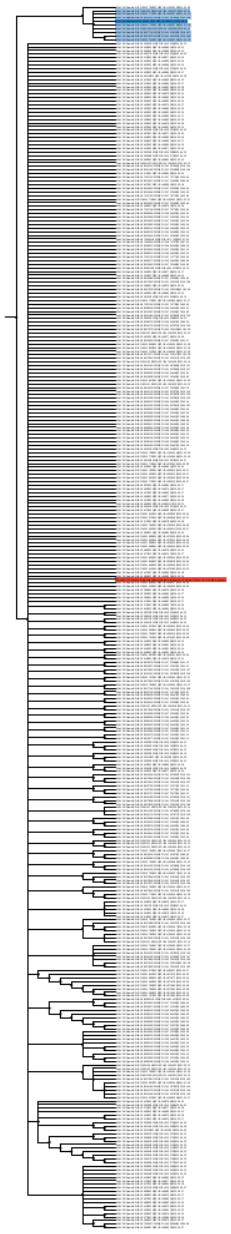

Figura 3: Captura de la visualización en la herramienta iTOL del árbol obtenido con método UPGMA (se adjunta en PDF)

En este arbol se puede ver claramente cómo las secuencias detectadas como de variante B117 aparecen completamente separadas de las que no. Cabe destacar que junto a ellas, aunque también separada, aparece otra secuencia, la *1262046|2021-01-26*. Si bien no se tienen más evidencias de ello, se piensa basándose en el árbol que se trata de un caso de otra variante del coronavirus distinta a la británica.

También se aprecia que la mayoría de las secuencias de 2020 aparecen a la misma altura que la secuencia de referencia de Wuhan.

En la parte inferior del árbol se puede ver cómo ciertas secuencias aparecen más estrechamente relacionadas entre sí. Resulta relevante que aquellas que pertenecen a una de esas mismas ramas fueron tomadas en fechas muy cercanas. Esto puede dar información de la evolución del virus en

la zona, e incluso podría servir para llevar a cabo labores de rastreo ante nuevos brotes aplicando un razonamiento análogo al del caso criminal estudiado en sesiones anteriores [3].

Referencias

[1] *Artículo sobre el índice de conservación.* URL: `http://bioinformatics.oxfordjournals.org/content/17/8/700.long` (visitado 21-04-2021).

[2] *Artículo sobre la variante B117.* URL: `https://www.biorxiv.org/content/10.1101/2020.12.14.422555v2.full.pdf` (visitado 21-04-2021).

[3] *Caso criminal en el que se usó como prueba un estudio filogenético.* URL: `https://www.pnas.org/content/99/22/14292` (visitado 21-04-2021).

[4] *Herramienta online iTOL para la visualización de árboles filogenéticos.* URL: `https://itol.embl.de/upload.cgi` (visitado 21-04-2021).

[5] *Imagen de cabecera de las páginas.* URL: `https://www.freepng.es/png-0m3sw0/` (visitado 16-02-2021).

[6] *Página con estadísticas sobre la evolución del SARS-COV-2.* URL: `https://nextstrain.org/ncov/global` (visitado 21-04-2021).