Trabajo de prácticas

Práctica 5 - Bioinformática

Árboles filogenéticos mediante RAxML y FastTree

19 Abril, 2021

Autor: Álvaro García Díaz 760704

Índice

Introducción 3

Multialineamiento 3

Mutaciones 4

Árboles filogenéticos 8

Referencias 11

Introducción

En la última práctica, se va a emplear todo lo aprendido en las anteriores para estudiar la **variabilidad** del SARS-CoV-2 y obtener su árbol filogenético para observar la evolución del virus con una serie de secuencias. Para conseguir este propósito, es necesario alinear todas las secuencias.

Los virus pueden sufrir **mutaciones** que provocan cambios en su genoma, por lo que se va a estudiar una serie de rangos de posiciones en los que ciertos genes (ORF1A, ORF3A, N) o la proteína Spike pueden encontrarse y calcular la frecuencia de cada nucleótido en cada posición y la medida de conservación para cada posición con el fin de determinar si una nueva mutación en estas posiciones puede provocar negativamente al individuo infectado.

Multialineamiento

Para realizar los **alineamientos**, tras los resultados obtenidos en la práctica 3, se va a emplear MAFFT [1] con la opción --auto ya que ofrecía una buena puntuación, y Clustal Omega [2] (debido a que solo permite ficheros de hasta 4Mb y el fichero con todas las secuencias ocupa algo más de 12Mb, se va a dividir el fichero original en 4 con la secuencia de referencia) con los argumentos por defecto con posterior utilización de MAFFT con la opción --merge [3]. La opción de emplear EMBOSS [4] (opción con mejor puntuación en la práctica 3) es inviable debido a que es necesario ejecutar la versión Web con cada secuencia (más de 300 secuencias) con la de referencia.

```
mafft --thread 8 --threadtb 5 --threadit 0 --reorder --auto input > output
```

Fig 1. Comando para alinear secuencias en MAFFT

```
mafft --merge table --reorder --auto input
```

Fig 2. Comando para unir secuencias en MAFFT

Para comprobar cuál ofrece un mejor alineamiento, se ejecutará el código desarrollado en Python en la práctica 3 para calcular la puntuación media mediante la nomenclatura habitual (1 la coincidencia, -3 la diferencia, -5 el primer gap y -2 los siguientes gaps cuando haya varios seguidos). El mejor alineamiento es el que menor puntuación media posea.

En el caso del multialineamiento mediante MAFFT, su **puntuación media** fue de 29196, mientras que para Clustal Omega fue de 29211, por lo tanto, en los siguientes apartados se ha continuado con el alineamiento obtenido mediante **MAFFT**. También, respecto a la longitud de las secuencias, las obtenidas mediante MAFFT eran de menor tamaño.

Mutaciones

Con el alineamiento obtenido en el punto anterior, se ha procedido a detectar todas las **mutaciones** que tienen las secuencias mediante el cálculo de la frecuencia de cada nucleótido para cada posición del alineamiento y del **índice de conservación**. Con estas medidas, se puede conocer que si una mutación ocurre en una posición, si afecta negativamente más o menos al individuo infectado.

Para realizar este propósito, se ha creado un programa (*conservation.py*) en Python al que se le introduce el alineamiento y calcula las dos medidas mencionadas anteriormente. Posteriormente, permite consultar para una determinada posición la frecuencia y el índice de conservación. También ofrece la posibilidad de devolver en un fichero **.CSV** todas las frecuencias y los índices de conservación calculados o devolver solamente las frecuencias e índices de conservación en un rango de posiciones en los que se encuentran los genes ORF1A, ORF3A y N y la proteína Spike. Este .CSV puede estar en orden descendente atendiendo a los datos (frecuencia y conservación) o ascendente atendiendo a las posiciones (por defecto).

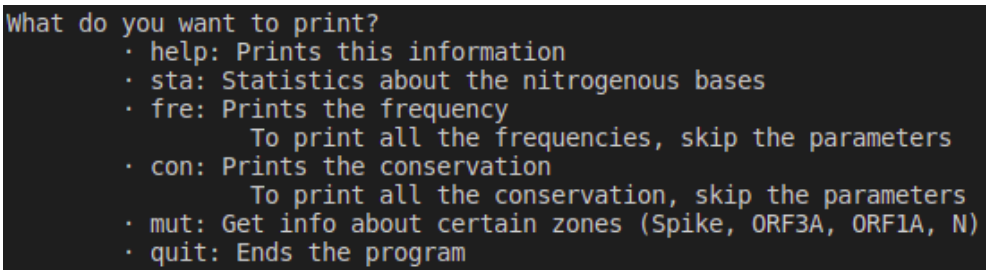

Fig 3. Interfaz del programa desarrollado para calcular las frecuencias y los índices de conservación

	A	B
1	Position	Conservation
2	24920	0.7079702340554762
3	28896	0.7025024028289916
4	24512	0.6931471805599453
5	23715	0.6931471805599453
6	23277	0.6931471805599453
7	28295	0.6931357338918793
8	5986	0.6930752938992721
9	28293	0.6930745993293921
10	28294	0.6930441574024588

Fig 4. Fragmento del .CSV que contiene todos los índices de conservación en orden descendente atendiendo a los datos

	A	B	C	D	E
1	Position	Adenine	Guanine	Cytosine	Thymine
2	1	1.0	0.0	0.0	0.0
3	2	0.0	0.0	0.0	1.0
4	3	0.0	0.0	0.0	1.0
5	4	1.0	0.0	0.0	0.0
6	5	1.0	0.0	0.0	0.0

Fig 5. Fragmento del .CSV que contiene todas las frecuencias por orden ascendente atendiendo a la posición

En el caso de querer observar la incidencia de los **genes** ORF1A, ORF3A o N o la **proteína** Spike, siempre se devolverán dos .CSV, uno con la frecuencia y otro con el índice de conservación.

```
Enter a valid option: m
        · 1: ORF1A Gene: 266-13483
        · 2: Spike protein: 21563-25384
        · 3: ORF3A Gene: 25393-26220
        · 4: N Gene: 28274-29533
Select one option (1-4):
```

Fig 6. Selección del gen o de la proteína a analizar junto al rango de posiciones que se van a observar

El código solo requiere de la instalación de la biblioteca numPy (mediante *pip3 install numpy*) y de una versión de Python >= 3.0. Precisa de entre 4 y 5 segundos para leer el alineamiento y realizar los cálculos necesarios antes de mostrar la interfaz.

Si una mutación ocurre en una posición que contenga poco índice de conservación, afectará de forma más negativa que si ocurriese en otra con alto índice de conservación. Por ejemplo, si la mutación tiene lugar en la posición 28896, cuyo índice de conservación es superior a 0.7, afectaría menos que si fuera en la posición 1, que posee como índice de conservación 0.0. Esto se debe a que en la posición 1, siempre se encuentra la Adenina, mientras que en la posición 28896 puede variar entre Adenina (0.2%), Guanina (44.5%) y Citosina (55.3%).

```
Enter a valid option: s
Adenine: Always in 8795 positions, dominates in 164 positions and appears in 144 positions.
Guanine: Always in 5564 positions, dominates in 297 positions and appears in 150 positions.
Cytosine: Always in 4916 positions, dominates in 574 positions and appears in 177 positions.
Thymine: Always in 9433 positions, dominates in 170 positions and appears in 743 positions.
```

Fig 7. Estadísticas de cada base nitrogenada

Otro detalle a destacar es que la **Timina** es la base nitrogenada que más se repite (se encuentra siempre en 9433 posiciones, predomina sobre 170 y aparece en otras 743).

En el caso de **evaluar** un gen o una proteína, además de devolver en .CSV tanto la frecuencia como el índice de conservación, también se muestran las estadísticas de las apariciones de cada base nitrogenada. En todos los casos prevalece la TImina, menos en el caso del gen N, que se trata de la Adenina.

```
>hCoV-19/Spain/CT-HUGTiPR006BX5B10/2021|EPI_ISL_1208495|2021-02-15
Mutation at column 25582, in the reference was t but it was c
Mutation at column 25645, in the reference was t but it was a
Mutation at column 25646, in the reference was g but it was t
Mutation at column 25793, in the reference was g but it was t
Total: 4

>hCoV-19/Spain/CT-HUVH-19243/2021|EPI_ISL_1080839|2021-02-09
Mutation at column 26068, in the reference was c but it was a
Total: 1
```

Fig 9. Ejemplo de fichero de salida del nuevo programa

```
Total number of differences: 256

Statistics:
Adenine: differences between reference sequence: 25
Guanine: differences between reference sequence: 17
Cytosine: differences between reference sequence: 32
Thymine: differences between reference sequence: 182
```

Fig 10. Final del fichero que devuelve el nuevo programa

Para evaluar más a fondo las mutaciones provocadas por los genes ORF1A, ORF3A y N y por la proteína Spike, se ha implementado un segundo programa (*mutation.py*) en Python, al cual hay que pasarle solamente el alineamiento obtenido y devuelve un fichero para cada gen y proteína estudiado que contendrá todas las diferencias existentes entre cada secuencia con la secuencia de referencia. Al estar presentes en ciertas posiciones, solo se miran estas. Al final del fichero, se encuentra el número total de diferencias que ha habido y unas estadísticas acerca de las diferencias entre las bases nitrogenadas.

En este caso, vuelve a destacar la **Timina** que es la base nitrogenada con la que hay más diferencias con la secuencia de referencia en todos los genes y en la proteína.

A continuación, se muestran una serie de capturas del uso del primer programa:

```
Enter a valid option: f
Enter a column (between 1 and 29916 or empty to print all): 1
Frequency at the column 1 and base Adenine: 1.0
Frequency at the column 1 and base Guanine: 0.0
Frequency at the column 1 and base Cytosine: 0.0
Frequency at the column 1 and base Thymine: 0.0
```

Fig 11. Frecuencia en una determinada posición (1)

```
Enter a valid option: c
Enter a column (between 1 and 29916 or empty to print all): 28896
Conservation at column 28896: 0.7025024028289916
```

Fig 12. Índice de conservación en una determinada posición (28896)

```
Enter a valid option: f
Enter a column (between 1 and 29916 or empty to print all):

Going to print all the frequencies...
Data in descending order? (y | n): y
Writing into the file frequency-D.csv all the frequencies...
```

Fig 13. Frecuencia en fichero .CSV en orden descendente atendiendo a la frecuencia

```
Enter a valid option: c
Enter a column (between 1 and 29916 or empty to print all):

Going to print all the conservation...
Data in descending order? (y | n): n
Writing into the file conservation.csv all the conservation...
```

Fig 14. índice de conservación en fichero .CSV en orden ascendente atendiendo a la posición

```
Enter a valid option: m
        · 1: ORF1A Gene: 266-13483
        · 2: Spike protein: 21563-25384
        · 3: ORF3A Gene: 25393-26220
        · 4: N Gene: 28274-29533
Select one option (1-4): 4
Data in descending order? (y | n): y
Adenine: Always in 387 positions, dominates in 19 positions and appears in 15 positions.
Guanine: Always in 244 positions, dominates in 30 positions and appears in 20 positions.
Cytosine: Always in 273 positions, dominates in 42 positions and appears in 15 positions.
Thymine: Always in 254 positions, dominates in 11 positions and appears in 57 positions.
Writing into the file N-D-con.csv all the conservation related to the gene passed...
Writing into the file N-D-fre.csv all the frequencies related to the gene passed...
```

Fig 15. Frecuencia e índice de conservación en .CSV para el gen N en orden descendente atendiendo a los datos

Árboles filogenéticos

Para finalizar, se van a emplear las herramientas RAxML y FastTree para construir árboles filogenéticos. Estas herramientas emplean el método de **máxima verosimilitud** basándose en modelos evolutivos para inferir la distribución del conjunto de secuencias del árbol.

Se van a realizar una serie de pruebas con diferentes argumentos para obtener el árbol filogenético con mejor puntuación. Durante estas pruebas, se ha detectado que RAxML obtenía mejores puntuaciones a coste de un mayor tiempo en comparación con FastTree.

```
fasttree -nt -gtr alignment < outputTree
```

Fig 16. Comando para la salida del mejor resultado para FastTree

```
ML-NNI round 1: LogLk = -62599.621 NNIs 100 max delta 19.60 Time 82.46s (max delta 19.597)
GTR Frequencies: 0.2987 0.1835 0.1963 0.3215ep 12 of 12
GTR rates(ac ag at cg ct gt) 0.3699 1.0401 0.1279 0.5203 3.7081 1.0000
Switched to using 20 rate categories (CAT approximation)20 of 20
Rate categories were divided by 0.641 so that average rate = 1.0
CAT-based log-likelihoods may not be comparable across runs
Use -gamma for approximate but comparable Gamma(20) log-likelihoods
ML-NNI round 2: LogLk = -58872.639 NNIs 62 max delta 3.35 Time 264.94es (max delta 3.346)
ML-NNI round 3: LogLk = -58862.518 NNIs 13 max delta 6.70 Time 282.03es (max delta 6.702)
ML-NNI round 4: LogLk = -58862.451 NNIs 3 max delta 0.00 Time 292.99es (max delta 0.000)
Turning off heuristics for final round of ML NNIs (converged)
ML-NNI round 5: LogLk = -58859.232 NNIs 6 max delta 1.92 Time 336.11 (final)delta 1.923)
Optimize all lengths: LogLk = -58859.147 Time 347.21
Total time: 449.08 seconds Unique: 418/418 Bad splits: 0/415rnal splits
```

Fig 17. Salida de FastTree con puntuación de -58859.147

El mejor árbol obtenido ha sido mediante **RAxML** con una puntuación -52494.365 mediante el uso de **bootstrap** y 10 iteraciones (mejor puntuación obtenida en la 3º). El comando empleado ha sido el siguiente:

```
raxmlHPC-SSE3 -s alignment -n output -m GTRCAT -p 555 -b 555 -N 10
```

Fig 18. Comando para la salida del mejor resultado para RAxML

```
Bootstrap[0]: Time 258.796930 seconds, bootstrap likelihood -54080.045473, best rearrangement setting 25
Bootstrap[1]: Time 227.038124 seconds, bootstrap likelihood -52576.366251, best rearrangement setting 25
Bootstrap[2]: Time 321.939031 seconds, bootstrap likelihood -52494.365134, best rearrangement setting 20
Bootstrap[3]: Time 264.690011 seconds, bootstrap likelihood -53930.293378, best rearrangement setting 25
Bootstrap[4]: Time 213.665338 seconds, bootstrap likelihood -53154.793248, best rearrangement setting 10
Bootstrap[5]: Time 235.240600 seconds, bootstrap likelihood -53472.911090, best rearrangement setting 25
Bootstrap[6]: Time 429.211743 seconds, bootstrap likelihood -53355.013198, best rearrangement setting 15
Bootstrap[7]: Time 307.256089 seconds, bootstrap likelihood -53663.050218, best rearrangement setting 25
Bootstrap[8]: Time 266.648385 seconds, bootstrap likelihood -52886.532702, best rearrangement setting 20
Bootstrap[9]: Time 218.190462 seconds, bootstrap likelihood -53876.879651, best rearrangement setting 25
```

Fig 19. Resultado de las iteraciones con bootstrap con RAxML

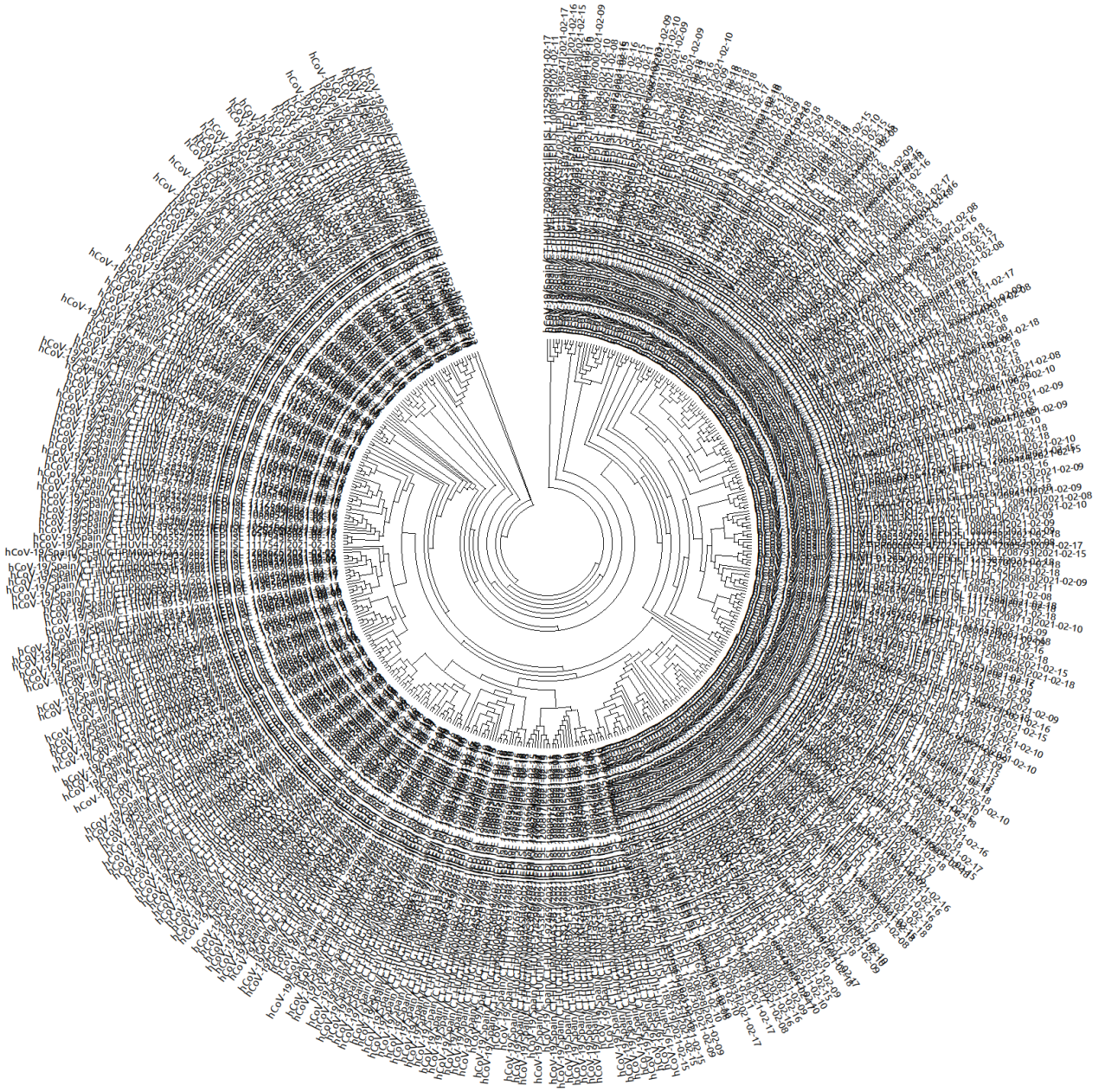

Fig 20. Árbol filogenético obtenido en la 3º iteración con forma circular visualizado mediante MEGA

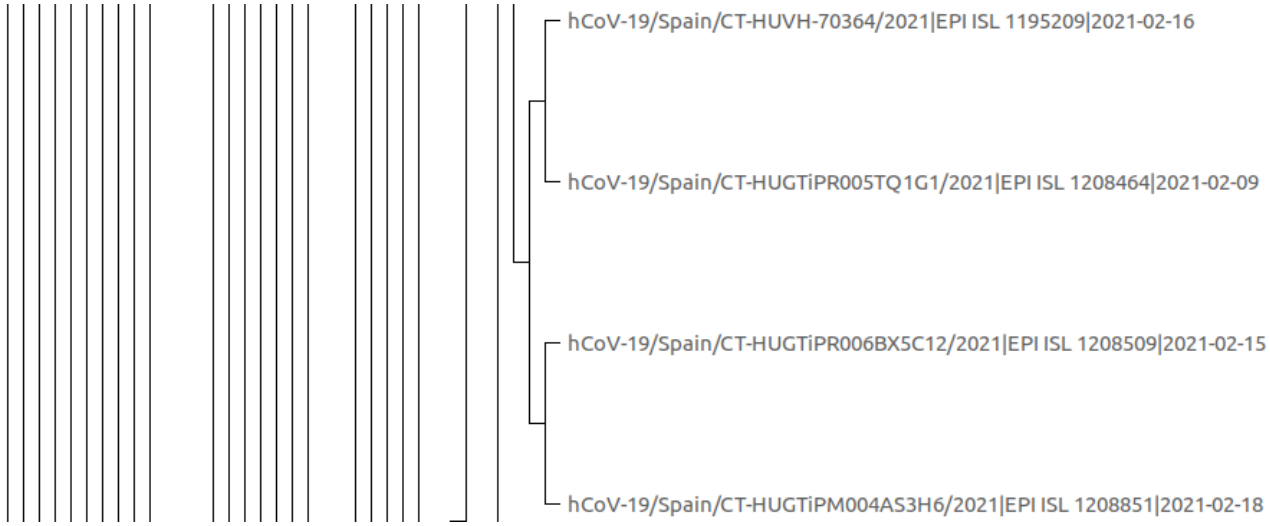

Fig 21. Fragmento del árbol filogenético obtenido en la 3º iteración visualizado mediante MEGA

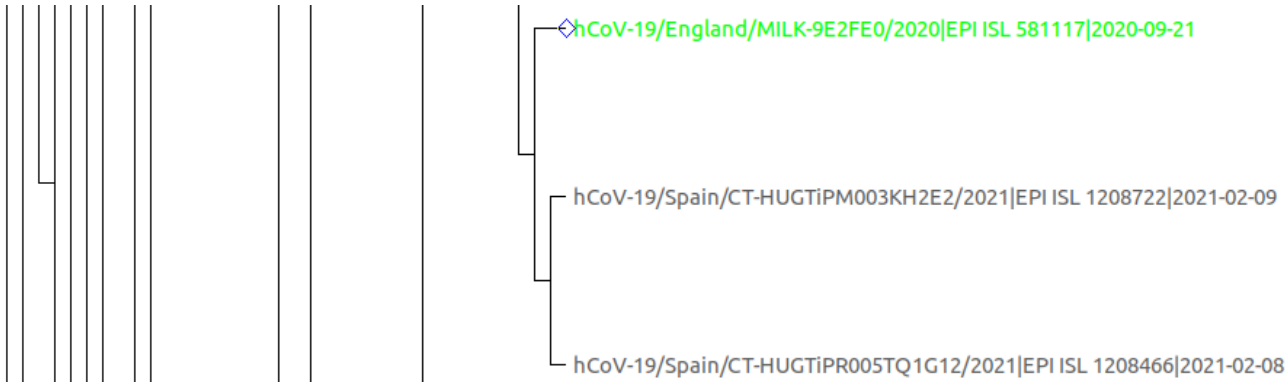

Fig 22. Fragmento del árbol filogenético obtenido en la 3º iteración visualizado mediante MEGA con la secuencia de la cepa británica

Al tratarse de más de 300 secuencias alineadas, si las secuencias contienen varias diferencias, el árbol filogenético será bastante grande. Debido a la cantidad de líneas en las dos últimas figuras, existe una gran cantidad de diferencias o mutaciones respecto a la secuencia de referencia. Tratándose del SARS-CoV-2 y de secuencias de este año, habría que realizar un análisis más a fondo del árbol filogenético y de las mutaciones para determinar si las secuencias pueden pertenecer a una variante, como puede ser la británica o la sudafricana.

Atendiendo a la Figura 22 en la que se visualiza la secuencia de la **cepa británica**, se pueden observar dos secuencias que podrían pertenecer a esta cepa, por lo que habría que realizar un análisis en profundidad de ambas.

Referencias

[1] MAFFT alignment and NJ / UPGMA phylogeny,
https://mafft.cbrc.jp/alignment/server/

[2] Clustal Omega < Multiple Sequence Alignment < EMBL-EBI,
https://www.ebi.ac.uk/Tools/msa/clustalo/

[3] Merge MSAs to single MSA using MAFFT,
https://mafft.cbrc.jp/alignment/server/merge.html

[4] EMBOSS Stretcher < Pairwise Sequence Alignment < EMBL-EBI,
https://www.ebi.ac.uk/Tools/psa/emboss_stretcher/