Práctica 3
Bioinformática

José Manuel Sánchez Aquilué 759267

10 de marzo de 2021

1. Introducción.

En la siguiente memoria quedan recogidos distintos procedimientos de alineamiento de secuencias de DNA. A continuación se mostraran los resultados de las pruebas con un par de conjuntos de 25 secuencias de coronavirus. También, veremos si en esas muestras está presente o no la cepa británica. Finalmente, se realizarán las mismas pruebas con todas las secuencias de coronavirus detectadas en Aragón y observaremos como el coste temporal del problema aumenta con creces mientras la puntuación se desploma.

2. Multialineamientos de las 25 secuencias de coronavirus.

Existen diversas bibliotecas para realizar tareas de multialineamientos. A mi juicio es preferible utilizar una herramienta en local que online. Por eso, en esta actividad nos vamos a decantar por mafft y por Biopython. A mafft se le puede pasar una secuencia de referencia, lo que permite ejecutar esta tarea en una sola linea de código. Mientras que EMBOSS, otra posible alternativa, tiene una instalación un tanto compleja, por eso descarté esa opción.

2.1. Utilización de algoritmos de mafft

En un primer momento el algoritmo iba demasiado lento. Al añadir el argumento –6merpair, la distancia se calculó en base al número de seis-mers compartidos, lo que se tradujo en una gran mejora de tiempo de ejecución.

Se hicieron pruebas con distintos tipos de métodos: progresivos, refinamiento iterativo y refinamiento iterativo con puntuaciones consistentes. Cabe destacar que independientemente del método utilizado, la puntuación fue la misma y los tiempos difieren en décimas de segundos (Figuras 1 y 2).

		Aragon1	Aragon2	Todo Aragon
Métodos progresivos	FTT-NS-1	1,733s	1,500s	30,447s
	FTT-NS-2	1,766s	1,506s	30,008s
Refinamiento iterativo	2 iteraciones	1,685s	1,502s	30,722s
	1000 iteraciones	1,670s	1,508s	30,838s
Iterativo con puntuaciones consistentes	E-INS	1,523s	1,505s	30,289s
	L-INS	1,525s	1,505s	30,005s
	G-INSs	1,513s	1,507s	30,390s

Figura 1: Tiempos de ejecución en local.

Tipo de puntuación	Aragon1	Aragon2	Todo Aragon
EBI	0.9862	0.9758	0.9688
Manera habitual	28344	27070	26274

Figura 2: Puntuaciones obtenidas.

Con los alineamientos construidos, podemos proceder a comprobar si estamos ante la variante británica. Vamos a prestar especial atención a las alteraciones en la proteína Spike, en especial en la que se produce en la posición 23063. Como podemos observar en la figura 3, el conjunto de secuencias Aragón 1, tiene las mutaciones propias de la cepa británica. En la entrega se incluye imágenes del alineamiento completo con el fin de que se aprecie con mayor claridad que en las capturas aquí adjuntas. Por otro lado las secuencias que se encuentran en el conjunto Aragón 2, no presentan esos cambios (Figura 4).

2.2. Utilización de la biblioteca Biopython.

Llegados a este punto, vamos a implementar nosotros mismos un algoritmo que haga los multialineamientos. El proceso se dividirá en dos fases. En primer lugar, vamos a alinear cada una de las secuencias con la secuencia de referencia. Las secuencias de covid son demasiado largas para el modulo pairwise2. Ahora bien, la documentación de biopython [1] indica que si este método no es suficiente, podemos valernos de la clase PairwiseAligner. Con lo cual, se le dirá al alineador cuales son los criterios de puntuación.

Acabada esa fase, obtendremos 25 alineamientos de dos secuencias, a partir de estos podemos procesar el método de estrella. El primer alineamiento lo llamaremos alineamiento grande y será al que iremos añadiendo las secuencias. Para cada alineamiento, almacenamos la posición de los gaps de la secuencia de referencia. Si alguna de las secuencias del alineamiento grande (incluida la de referencia) no tenía un gap en una de esas posiciones, se le añade, al amparo del principio .ºnce a gap, always a gap". Con el fin de que todas tengan la misma longitud, se copian los gap de la secuencia de referencia del alineamiento grande en la que vayamos a añadir.

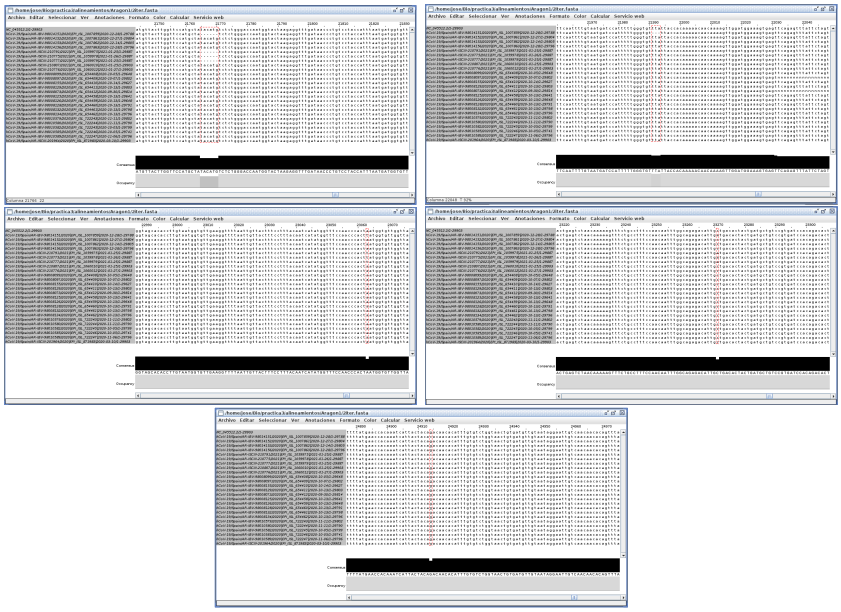

Figura 3: Mutaciones en el alineamiento de las secuencias del conjunto Aragón 1.

En materia de coste temporal, observamos que el algoritmo tiene un orden de $O(\frac{numSeq^2}{2} * lenSeq)$.

Los resultados obtenidos, tanto temporales (Figura 5) como de puntuación (Figura 6) muestran un mejor rendimiento en el algoritmo de mafft con diferencia. Una causa a la que se achaca esta baja puntuación es la forma en la que el algoritmo trata las secuencias de Ns. Al no poderse emparejar con el resto, obligan al algoritmo a añadir demasiados gaps y, por ende, aumenta la longitud de las secuencias (Figura 7). Si fuesen unas pocas secuencias las que contienen largas cadenas de Ns, se podrían excluir del proceso y comparar los resultados. Sin embargo, en la visualización se aprecia que no se trata de un caso aislado, sino que esas cadenas están presentes en casi todas las secuencias. Otra posible solución sería cambiar la matriz de sustitución para indicar que los caracteres N actúen como comodín. Si bien puedes cargar distintos tipos de matrices de sustitución comúnmente utilizadas por biólogos (BENNER22, BENNER6, BLOSUM50, BLOSUM62...), no se ha conseguido localizar la opción que permite personalizar la matriz.

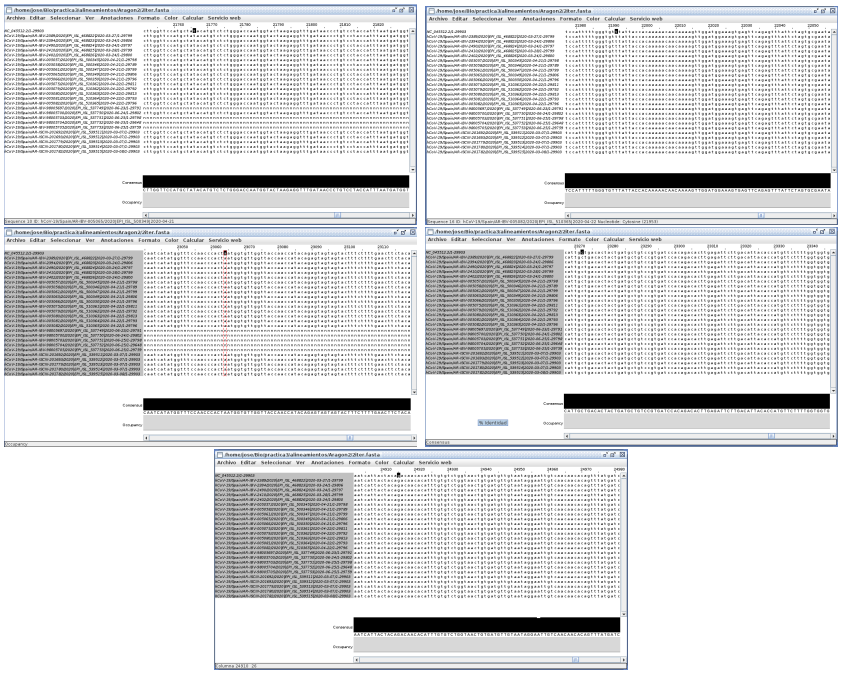

Figura 4: Mutaciones en el alineamiento de las secuencias del conjunto Aragón 1.

	Aragon1	Aragon2	Todo Aragon
Tiempos	6 min 57 seg	7 min 3 seg	2 horas 6 minutos

Figura 5: Tiempos de ejecución en local.

	Aragon1	Aragon2	Todo Aragon
EBI	0.8965	0.8282	0.4517
Manera habitual	20883	13518	-49231

Figura 6: Puntuaciones obtenidas.

Figura 7: Zonas críticas en los alineamientos (izquierda con mafft, derecha con biopython).

3. Multialineamientos de todas secuencias de coronavirus de Aragón.

En este apartado se comprueba esencialmente como al aumentar el número de secuencias el coste temporal de los algoritmos aumenta con creces (Figura 1 y 5). No solo se nota a la hora de ejecutar los algoritmos de multialineamiento; cuando se efectúa el calculo de la puntuación con el script que implementé, cuyo orden es también de $O(\frac{numSeq^2}{2} * lenSeq)$, el tiempo de ejecución también se dispara. Supongo que Biopython, o alguna otra librería, tendrá algún método implementado para calcular la puntuación y seguramente sea más eficiente que cualquier algoritmo que haga yo. No obstante, para esta práctica era suficiente con mi código.

En el caso del algoritmo del mafft, se percibe un ligero descenso en la puntuación respecto a los otros multialineamientos. Mientras que en biopython ese desplome es abismal, llegando a tomar una puntuación negativa.

Cuando se trabaja con 500 secuencias del orden de 30.000 pares de longitud, es más complicado visualizar y detectar las mutaciones significativas.

Referencias

[1] Iddo Friedberg Thomas Hamelryck Michiel de Hoon Peter Cock Tiago Antao Eric Talevich Bartek Wilczyński Jeff Chang, Brad Chapman. *Biopython Tutorial and Cookbook*. 2020.