Proyecto Final

Genómica Computacional

Hernández Uriostegui David
Suaste Morales Noé Abraham

03/06/23

1. Introducción

Desde el año 2019 la humanidad se ha visto atacada por el coronavirus de tipo 2 causante del síndrome drespiratorio agudo severo **SARS-CoV-2** (del inglés *severe acute respiratory syndrome coronavirus 2*). Este virus fue el que ocasionó la epidemia, y que durante ésta, manifestó distintas variantes gracias a mutaciones que se convirtieron rápidamente en objeto de estudio para el desarrollo de vacunas y medicinas.

Desde el primer brote detectado en Wuhan, China en diciembre de 2019 de acuerdo con la Organización Mundial de la Salud, se han confirmado más de 767 millones de casos y más de 6.9 millones de muertes debido al virus (1). Estos datos reflejan la importancia que debe tener el estudio del virus, particularmente de las variantes de relevancia médica.

Las variantes se clasifican de acuerdo a su tasa de transmisión y a la gravedad de la enfermedad que ocasionan. También su linaje es un factor importante para clasificar de manera correcta y así mismo lo es la mutación dentro de las variantes particulares, dentro de las variantes más importantes y en las cuáles nos basaremos para el análisis, están las variantes: alfa, beta, gamma, delta y ómicron.

2. Pregunta de investigación

¿Qué tanto difieren las secuencias de las variantes de SARS-CoV-2 con respecto a la secuencia original?

3. Objetivo

El objetivo de éste trabajo es determinar las diferencias y similitudes entre las principales variantes de importancia médica del SARS-CoV-2 frente a la secuencia original.

4. Métodos

- Clustel Omega. Se utilizará este método para alinear las múltiples secuencias y obtener la matriz de porcentaje de identidad.
- BLAST. Se usa para realizar una alineamiento local entre la secuencia de referencia con las secuencias de las variantes.
- BioPython. Se utiliza para poder comparar como funciona los alineamientos globales los cuales están basados en el algoritmo `Needleman-Wunsch`.

5. Resultados

Primeramente, utilizando las siguientes secuencias y alineando mediante Clustel Omega:

- Secuencia variante alfa.
- Secuencia variante beta.
- Secuencia variante delta.
- Secuencia variante gamma.
- Secuencia variante ómicron.

pudimos obtener resultados que podemos visualizarlos en el siguiente árbol filogénetico 1:

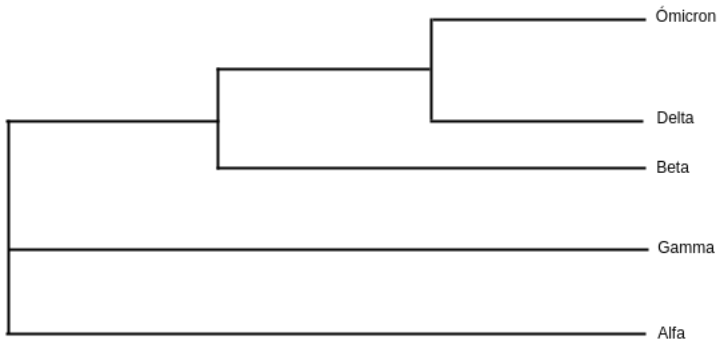


Figura 1: Árbol filogenético de las 5 principales variantes

Mediante el mismo método pero también considerando la secuencia original, pudimos obtener la matriz de porcentaje de identidad representada en la tabla 1:

	original	alfa	beta	delta	gamma	ómicron
original	100	99.47	97.05	98.24	95.94	95.89
alfa		100	97.36	98.54	96.29	96.21
beta			100	96.10	93.87	93.78
delta				100	96.50	95.09
gamma					100	92.70
ómicron						100

Tabla 1: Matriz de porcentaje de identidad de las principales variantes

Siguiendo con **BLAST**, sabemos a partir de la Práctica 3 que esta herramienta usa el algoritmo de **Smith–Waterman**, el cual es un algoritmo de alineamiento global.
Estos fueron los resultados obtenidos:

7ZBCZVZZ114-Alignment-Descriptions

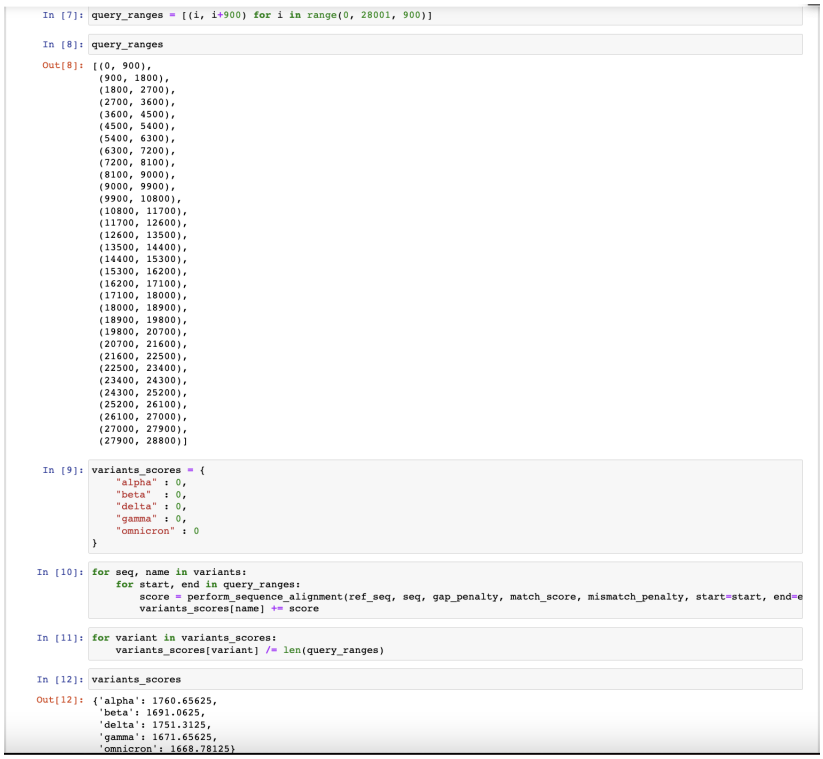
Description	Scientific Name	Common Name	Taxid	Max Score	Total Score	Query Cover	E value	Per. ident	Acc. Len	Accession
Alpha			0	54674	54674	99%	0.0	99.82	29884	Query_8401
Delta			0	38566	54010	98%	0.0	99.82	29890	Query_8403
Gamma			0	27497	52769	96%	0.0	99.87	29898	Query_8404
Omnicron			0	25874	52505	96%	0.0	99.35	29866	Query_8405
Beta			0	21339	53348	97%	0.0	99.79	29879	Query_8402

Cómo podemos observar del `Query cover` y `Per. Ident` indican, que en efecto, las variantes son similares a la secuencia original, siendo la más divergente la variante *Omnicron*.

Por último, para poder realizar una mejor comparativa entre las tecnologías disponibles para realizar alineamientos. Se decidió hacer uso de la biblioteca de Python: `Biopython`.

Para el análisis, usamos la implementación del algoritmo de alineamiento global, el cual se basa en el algoritmo `Needleman-Wunsch`.
Cabe mencionar, que las secuencias son bastante grandes lo cual querer comparar todas contra la secuencia original hacía que nuestras computadoras se volvieran bastante lentas y que no terminaran de correr el algoritmo. Por lo cual decidimos ir evaluando por subregiones en intervalos de 900 (se escogió intervalos de 900, ya que era el número más grande que nuestras maquínas podían usar sin que el algoritmo tardara una cantidad de tiempo bastante alta). Es decir comparamos la subregión (0, 900) de la secuencia original, contra la subregión (0, 900) de *alpha*, *beta*, *gamma*, *delta* y *omnicron*.

De esta manera lo que se hacía era ir sumando los resultados obtenidos de correr el algoritmo en esas regiones por cada una de las variantes, y al final. El resultado de cada variante dividirlo entre el número de subregiones evaluadas.



```
In [7]: query_ranges = [(i, i+900) for i in range(0, 28001, 900)]

In [8]: query_ranges

Out[8]: [(0, 900),
         (900, 1800),
         (1800, 2700),
         (2700, 3600),
         (3600, 4500),
         (4500, 5400),
         (5400, 6300),
         (6300, 7200),
         (7200, 8100),
         (8100, 9000),
         (9000, 9900),
         (9900, 10800),
         (10800, 11700),
         (11700, 12600),
         (12600, 13500),
         (13500, 14400),
         (14400, 15300),
         (15300, 16200),
         (16200, 17100),
         (17100, 18000),
         (18000, 18900),
         (18900, 19800),
         (19800, 20700),
         (20700, 21600),
         (21600, 22500),
         (22500, 23400),
         (23400, 24300),
         (24300, 25200),
         (25200, 26100),
         (26100, 27000),
         (27000, 27900),
         (27900, 28800)]

In [9]: variants_scores = {
            "alpha" : 0,
            "beta"  : 0,
            "delta" : 0,
            "gamma" : 0,
            "omnicron" : 0
        }

In [10]: for seq, name in variants:
             for start, end in query_ranges:
                 score = perform_sequence_alignment(ref_seq, seq, gap_penalty, match_score, mismatch_penalty, start=start, end=e
                 variants_scores[name] += score

In [11]: for variant in variants_scores:
             variants_scores[variant] /= len(query_ranges)

In [12]: variants_scores

Out[12]: {'alpha': 1760.65625,
          'beta': 1691.0625,
          'delta': 1751.3125,
          'gamma': 1671.65625,
          'omnicron': 1668.78125}
```

De igual manera, en este caso podemos observar que el *Omnicron* es la variante con menor similitud entre todas las demás.

Lo que se procedió a hacer es comparar cada una de las variantes contra las demás y también comprobar que el método de alineamiento que usa `BioPython` es en efecto el de

`Needleman-Wunsch`. Al final, se comprobó que esto último es cierto.

6. Discusión

Del árbol filogenético 1 podemos deducir que la variante ómicron ha sido la que más mutaciones ha tenido con respecto a las demás variantes de importancia médica. Además, sabemos que la variante ómicron es de las más predominantes en diversos países(2). Esto nos invita a pensar que las mutaciones efectivamente ayudaron a que el virus encontrara una manera más eficaz de infectar.

Gracias a la matriz de porcentaje de identidad1 podemos observar que tenemos variantes bastante similares a la original como la alfa y en menor medida la delta y por el contrario tenemos algunas que no se asemejan tanto a la secuencia original como lo son la gamma y la ómicron.

7. Conclusión

Podemos apreciar que las variantes que más mutaciones conllevaron son aquellas que actualmente tienen un alto grado de casos en el mundo como la variante ómicron y la variante delta(2). También pudimos ver que no necesariamente se necesitaba diferir mucho de la secuencia original para ser una variante altamente contagiosa como lo es la variante delta. Sin embargo la variante ómicron si presenta significantes cambios con respecto a la variante original y ha sido probada altamente contagiosa aunque menos letal.

Referencias

[1] Organización Mundial de la Salud. (2023). COVID-19: Información general. Recuperado de https://www.who.int/emergencies/diseases/novel-coronavirus-2019

[2] Covariants. (2023) Casos estimados por variante. Recuperado de https://covariants.org/cases

[3] Farhud, D. D., and Mojahed, N. (2022). SARS-COV-2 Notable Mutations and Variants: A Review Article. Iranian journal of public health, 51(7), 1494–1501. https://doi.org/10.18502/ijph.v51i7.10083

[4] Mittal, A. (2021, February 23). Sequence Alignment and the Needleman-Wunsch Algorithm. Analytics Vidhya. https://medium.com/analytics-vidhya/sequence-alignment-and-the-needleman-wunsch-algorithm-710c7b1a23a4

[5] Bio.pairwise2 module — Biopython 1.75 documentation. (n.d.). Biopython.org. https://biopython.org/docs/1.75/api/Bio.pairwise2.html