Aplicación Web para el Análisis de Información Virológica transmitidas por vectores

Alex Steve Chung Alvarez
Rommel Yoshimar Condori Muñoz

Julio 2020

Abstract

In the present work, a help system web application will be developed for the analysis of genetic information of 4 types of viruses and their variants, using bioinformatics tools such as BLAST and MUSCLE, for the construction of phylogenetic trees. The virus to be analyzed are: Chikungunya Virus, Dengue Virus, Ebola Virus and Zika Virus

Resumen

En el presente trabajo, se desarrollará una aplicación web de sistema de ayuda para el análisis de información genética de 4 tipos de virus y sus variantes, usando herramientas de bioinformática como BLAST y MUSCLE, para la construcción de árboles filogenéticos. Los virus a analizar son: Chikungunya Virus, Dengue Virus, West Nile Virus y Zika Virus.

1 Descripción del proyecto

Se desarrollará una aplicación web para el procesamiento de información genética de los virus citados anteriormente, como también analizar la relación evolutiva con otras variantes a partir de la construcción de árboles filogenéticos.
Para dicho propósito, se usarán la librería Biopython la cual incluye las herramientas de bioinformática que se usarán (BLAST, MUSCLE), y el framework Flask para la aplicación web.

1.1 Objetivos

1. Elegir los diversos virus presentados a analizar para el proyecto.
2. Recopilar genes y proteínas de los virus escogidos.
3. Encontrar secuencias homólogas para los genes escogidos.
4. Alinear secuencias encontradas.
5. Generar el árbol filogenético de cada gen.
6. Construir un software interactivo para visualizar los árboles filogenéticos.

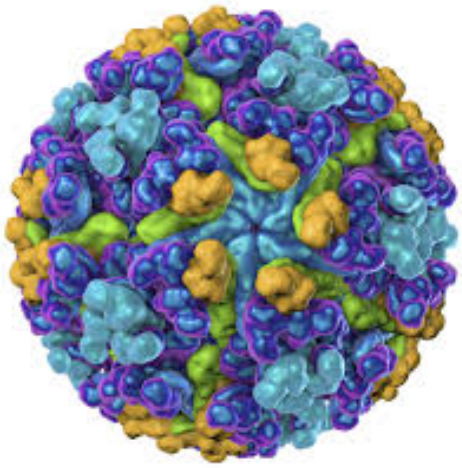

Figura 1: Chikungunya Virus

Figura 2: Dengue Virus

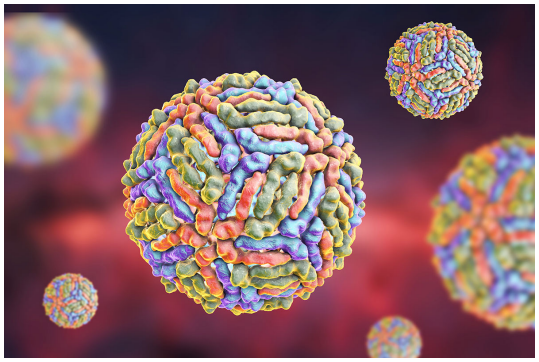

Figura 3: West Nile Virus

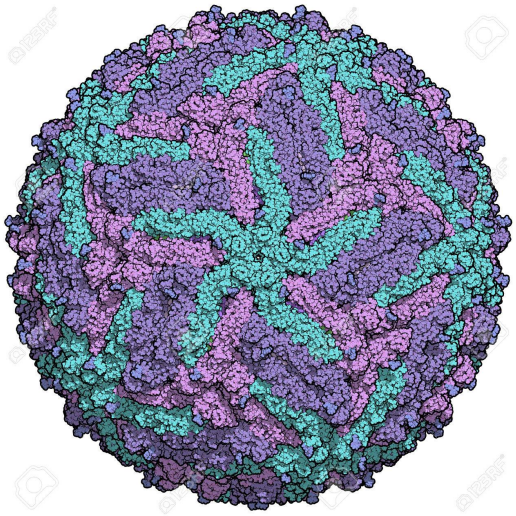

Figura 4: Zika Virus

1.2 Cronograma

Entrega 1 (Lun 21 Julio)

1. Desarrollo de primera versión de propuesta.
2. Definición de cronograma y entregables.

Entrega 2 (Lun 10 Agosto)

1. Presentación de la versión final de propuesta.
2. Ajuste de cronograma.
3. Elección de especies para el análisis.
4. Recopilación de genes y proteínas de virus elegidos.

Entrega 3 (Lun 31 Agosto)

1. Buscar secuencias homólogas de las secuencias usando BLAST.
2. Tratamiento de secuencias homólogas encontradas.
3. Alineamiento múltiple de secuencias usando MUSCLE.
4. Visualización de la información genética procesada en la aplicación web.

Entrega 4 (Lun 7 Setiembre)

1. Construcción de árboles filogenéticos.
2. Visualización de los árboles filogenéticos en la aplicación web.
3. Reporte final.

2 Algoritmos e implementación computacional

2.1 BLAST

"Basic Local Alignment Search Tool" (BLAST) es un programa de búsqueda de base de datos, el cual usa heurísticas para producir resultados rápidamente. BLAST retorna una lista de pares de segmentos de alta puntuación (HSP) entre la secuencia consultada y las secuencias de la base de datos. Sus alineamientos reportados no poseen gaps, lo cual es una razón por la que este algoritmo es rápido.

2.1.1 Algoritmo

El algoritmo de BLAST depende del tipo de secuencias comparadas: ADN o Proteínas. Consiste en 3 pasos:[1]

1. **Compilación de conjunto de cadenas de alto puntaje (o *words*).**
   Las secuencias de proximidad de una $k - word$ w incluyen todas las cadenas de longitud k, que cuando están alineadas con w tienen un score de alineamiento de por lo menos T. Estos scores se calculan usando matrices de scoring (PAM o BLOSUM).

2. **Búsqueda de *seed hits*.**
   Se usan dos métodos para el escaneo de la base de datos en búsqueda de hits. Una de ellas es organizar los words de la lista en una Tabla Hash. El otro método usa un Autómata Finito Determinista.

3. **Extensión de semillas.**
   El algoritmo se detiene cuando la puntuación cae una cierta distancia por debajo de la mejor obtenida hasta ese punto para las extensiones más cortas. Esto se hace en ambas direcciones, y se mantiene el par de segmentos de alta puntuación originados a partir de esta seed.

2.1.2 Variantes

Algunas variantes de búsquedas BLAST[2]:

- **BLASTN:** Búsqueda de secuencias Nucleótidos-Nucleótidos. Busca secuencias más distantes.
- **BLASTP:** Comparación de secuencias Proteína-Proteína. Su algoritmo es base para otras búsquedas como BLASTX y TBLASTN.
- **BLASTX:** busca una consulta de nucleótidos contra una base de datos de proteínas, traduciendo la consulta sobre la marcha.
- **TBLASTN:** busca una consulta de proteínas contra una base de datos de nucleótidos, traduciendo la base de datos sobre la marcha.

2.2 MUSCLE

"Multiple Sequence Comparison by Log-Expectation" (MUSCLE) es un software de computadora de alineamiento múltiple de secuencias de proteínas y nucleótidos. A menudo es usado como reemplazo de Clustal, ya que generalmente (pero no siempre) da mejores alineamientos de secuencias, dependiendo de la opción escogida. También es significativamente más rápido que Clustal.

2.2.1 Algoritmo

Consiste de tres etapas: Borrador progresivo, Mejora progresiva y Refinamiento. Las dos primeras etapas tienen una complejidad temporal: $O(N^2L + NL^2)$ Y una complejidad espacial $O(N^2 + NL + L^2)$. Mientras que el refinamiento adiere $O(N^3L)$ a la complejidad temporal. [3]

1. **Etapa 1: Borrador progresivo** El objetivo de la primera etapa es producir una alineación múltiple, enfatizando la velocidad sobre la precisión.
   1.1 La distancia *kmer* se calcula para cada par de secuencias de entrada, dando la matriz de distancia $D1$.
   1.2 La matriz $D1$ está agrupada por UPGMA, produciendo el árbol binario $TREE1$.
   1.3 Una alineación progresiva se construye siguiendo el orden de ramificación de $TREE1$. En cada hoja, se construye un perfil a partir de una secuencia de entrada. Los nodos en el árbol se visitan en orden de prefijo (hijos antes que sus padres). En cada nodo interno, se construye una alineación por pares de los dos perfiles secundarios, dando un nuevo perfil que se asigna a ese nodo. Esto produce una alineación múltiple de todas las secuencias de entrada, $MSA1$, en la raíz.

2. **Etapa 2: Mejora progresiva** La principal fuente de error en el borrador de la etapa progresiva es la medida aproximada de la distancia *kmer*, que resulta en un árbol subóptimo. MUSCLE por lo tanto, vuelve a estimar el árbol utilizando la distancia $Kimura$, que es más precisa pero requiere una alineación.
   2.1 La distancia de $Kimura$ para cada par de secuencias de entrada se calcula a partir de $MSA1$, dando la matriz de distancia $D2$.
   2.2 La matriz $D2$ está agrupada por UPGMA, produciendo el árbol binario $TREE2$.
   2.3 Se produce una alineación progresiva siguiendo $TREE2$ (similar a 1.3), produciendo una alineación múltiple $MSA2$. Esto se optimiza calculando alineaciones solo para subárboles cuyos

órdenes de ramificación cambiaron en relación con $TREE1$.

3. **Etapa 3: Refinamiento**
   3.1 Se elige un borde de $TREE2$ (los bordes se visitan en orden de distancia decreciente desde la raíz).
   3.2 $TREE2$ se divide en dos subárboles eliminando el borde. Se calcula el perfil de la alineación múltiple en cada subárbol.
   3.3 Se produce una nueva alineación múltiple al realinear los dos perfiles.
   3.4 Si se mejora la puntuación SP, se mantiene la nueva alineación; de lo contrario, se descarta.

2.2.2 MUSCLE vs CLUSTALW

CLUSTALW es otra herramienta de alineamiento múltiple que incluye Biopython. Según pruebas de benchmark como BALiBASE y PREFAB [3], MUSCLE tiene mejor desempeño con menor tiempo que CLUSTALW, en el alineamiento de 49 secuencias de 240 de largo, haciéndolo en la mitad de tiempo.

2.3 Neighbor Joining

"Neighbor Joining" (NJ) es un método de agrupación de abajo hacia arriba para la creación de árboles filogenéticos. La base del método radica en el concepto de evolución mínima, a saber, que el árbol verdadero será aquel para el cual la longitud total de la rama, S, es más corta.
La técnica de Neighbor Joining es una aproximación simple, pero muy efectiva para construir árboles para grandes conjuntos de datos. El árbol resultante no está enraizado y es aditivo, una propiedad que se supone que deriva las fórmulas para su construcción.

2.3.1 Algoritmo

A partir de una matriz distancia, que indica la distancia entre cada par de taxones. El algoritmo comenzará con un árbol sin resolver, cuya topología es una "red estrella" y sigue los siguientes pasos hasta resolver el árbol[4]:
Para simplificar, se definen los siguientes terminos:

$$U_i = \sum_{k=1}^{N} d_{ik} \tag{1}$$

$$U_j = \sum_{k=1}^{N} d_{jk} \tag{2}$$

Paso 1: Con la matriz de distancias actual se calcula la matriz δ como sigue:

$$\delta_{ij} = d_{ij} - \frac{U_i + U_j}{N-2} \tag{3}$$

Paso 2: Una vez que los vecinos i y j han sido definidos desde el cálculo de δ_{ij}, el nuevo nodo Y puede ser añadido al árbol como sigue:

$$b_{iY} = \frac{1}{2}(d_{ij} + \frac{U_i - U_j}{N-2}) \tag{4}$$

y:

$$b_{jY} = d_{ij} - b_{iY} \tag{5}$$

Paso 3: Ahora se requiere las distancias desde Y hacia las otras secuencias, k, los cuales son calculados usando:

$$d_{Yk} = \frac{1}{2}(d_{ik} + d_{jk} - d_{ij}) \tag{6}$$

Para agregar más nodos, ahora repetimos el proceso, comenzando con el árbol de estrellas formado al eliminar las secuencias i y j, para dejar un árbol de estrellas con el nodo Y como una nueva hoja

2.3.2 NJ vs UPGMA

UPGMA es otro algoritmo de construcción árboles filogenéticos a partir de matrices de distancia. A diferencia de NJ, los árboles son enraizados (dendrogramas). UPGMA se considera como método menos confiable, ya que asume que el reloj molecular de todos los nodos es igual, por lo que se prefiere usar NJ. [4].

3 Resultados

1. Podemos observar los árboles filogenéticos que se muestran de acuerdo al alineamiento múltiple.

2. Obtenemos un análisis bioinformático de cinco viruses provenientes de mosquitos.

3. Podemos observar las relaciones evolutivas de dichos viruses.

4 Conclusiones

1. Se logrará el análisis de viruses provenientes de mosquitos.
2. Mediante el alineamiento podremos ver similitudes entre las secuencias.
3. Se descubrirán las relaciones evolutivas entre dichos viruses.

Referencias

[1] Joao Setubal y Joao Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Company, 1997, págs. 84-86.

[2] Thomas Madden. *"The BLAST Sequence Analysis Tool." The NCBI Handbook [Internet]. 2nd edition.* 2013. URL: `https://www.ncbi.nlm.nih.gov/books/NBK153387/` (visitado 24-10-2019).

[3] Edgar RC. "MUSCLE. Multiple sequence alignment with high accuracy and high throughput". En: *Nucleic Acids Res* 32.5 (2004), págs. 1792-7. DOI: `10.1093/nar/gkh340`. pmcid: `PMC390337`.

[4] Marketa Zvelebil y Jeremy O. Baum. *Understanding Bioinformatics*. Garland Sciencee, Taylor & Francis Group, LLC, 2008, págs. 278-285.