Implementación de la tesis: "Multiple Sequence Alignment using Particle Swarm Optimization"

Vicente Machaca Arceda[1]

[1]*Universidad Nacional de San Agustín de Arequipa. Email: vicente.machaca.a@gmail.com*

Abstract

Viral subtyping play a major role for the appropriate diagnosis and treatment of illness. Actually, the most used method for viral subtyping relies on BLAST method that look for similar genomes in big data sets on the Web. The major disadvantages is that we need a big data set where to look for, also we expose the privacy sample genome consulted. An alternative emerged with the use of machine learning models that take the viral sample genome and predict the subtyping. Several methods have been proposed for viral subtyping, based on machine learning models, in this study we compared the two most relevant based on k-mer frequency, Kameris proposed by Solis-Reyes *et al.* (2018) and Castor-KRFE proposed by Lebatteux, Remita, and Diallo (2019). Both have the same results when we avoid their dimensionality reduction and feature elimination, but when not, Kameris slightly outperform Castor-KRFE. Moreover, Castor-KRFE could get a small feature vector for $k > 5$ (in k-mer).

Keywords: HIV, Polyomavirus, genome, viral subtyping, k-mer, machine learning.

1 Introducción

El area de Bioinformática ha tenido un auge en las ultimas decadas. Por ejemplo, el proyecto: *The Human Genome Project* (HGP) que inicio en 1990 y fue completado el 2003, donde participarón varios paises como EEUU, Reino Unido, Japón, Francia, Alemania, España y China (NIH 2021); el proyecto tenia como objetivo secuenciar todo el genoma humano, el cual resulto ser una secuencia de aproximadamente 3.2 billones de pares de bases (Archibald 2018). En la actualidad exixten diversas areas de investigación como: la prediccón de estructura de proteinas, predicción de la función de una proteina a partir de estructuras de redes de proteinas, descubrimiento de medicamentos, predicción de enfermedades a partir del genoma, analisis de virus, etc. Es tan grande este campo de estudio que incluso se ha dividido en otras areas como *metagenomics*, *proteomics*, *chemical informatics*, etc.

1.1 Problema

Uno de los tantos problemas que existen en bioinformática, es el alineamiento multiple de secuencias. Se puede definir el alineamiento de secuencias como un método que permite determinar el grado de similitud entre dos o más secuencias (Xiong 2006), además el método puede insertar *gaps* dentro de las secuencias de consulta, con el objetivo de lograr la mayor cantidad de bases alineadas. Por ejemplo, en la Figura 1, se muestra el resultado luego de alinear varias secuencias de aminoacidos. Como podemos ver, se ha isertado *gaps* (-) para así maximizar la cantidad de aminoacidos que coinciden en la misma posición.

El alineamiento de secuencias puede dividirse en dos grupos: *pair-wise sequence alignment* (PSA) y *multiple sequence alignment* (MSA) (Xiong 2006). La diferencia radica en que el primero alinea solo dos secuencias y el segundo puede alinear dos a mas secuencias. Además, el problema

Multiple Sequence Alignment using Particle Swarm Optimization (2009)

Thesis: Master Thesis

Thesis author
Fabien Bernard

Report author
V.E. Machaca Arceda

Figura. 1. Ejemplo de *Multiple Sequence Alignment* (MSA). El método ha insertado *gaps* (-) en las secuecnias necesarias para maximizar la cantidad de letras (aminoacidos) que concidan en la misma posición.

de MSA es considerado un problema NP completo (Wang and Jiang 1994). Debido a esto es que se han planteado heuristicas que logran obtener una solución local; el algoritmo mas utilizado es CLUSTAL, fue propuesto por Higgins and Sharp (1988), este algoritmo ha sido mejorado con CLUSTALV (Higgins, Bleasby, and Fuchs 1992), CLUSTALW (Thompson, Higgins, and Gibson 1994) y CLUSTALX (Jeanmougin *et al.* 1998). Otro algoritmo importante es MUSCLE (Edgar 2004).

El problema de los algoritmos mencionados anteriormente, es que a pesar de ser heurísticas, el tiempo de procesamiento es muy alto. Cada segundo los datos genómicos crecen exponencialmente (Archibald 2018) y los algoritmos utilizados para buscar información en estas bases de datos, son basados en alineamiento. Entonces, mientras mas crecen los datos, mas lentos se vuelven estos algoritmos (Zablocki *et al.* 2009).

1.2 Objetivo

Zablocki *et al.* (2009), proponen en su tesis aplicar *Particle Swarm Optimization* (PSO) como alternativa a CLUSTAL para solucionar el problema de MSA.

2 Metodología aplicada

En esta sección describimos la metodología utilizada por Zablocki *et al.* (2009) para solucionar el problema de MSA utilizando PSO.

2.1 Representación de las partículas

Un de los primeros pasos para solucionar un problema utilizando PSO, es la representación de cada particula. Zablocki *et al.* (2009) propone utilizar la posición de cada *gap* como un vector. Por ejemplo en al Figura 2, tenemos la representación del la partícula *leader* y una partícula cualquiera (ambas representan una posible solución al problema). Ahora en la FIgura 3, tenemos otra forma de representar dichas partículas, en este caso solo estamos considerando la posición de cada *gap* insertado. La opción mas facil de controlar es la correspondiente a la Figura 3.

2.2 Movimiento de las partículas

Según el algoritmo de PSO, cada partícula debe acercarse al *leader* en cada iteración. Este acercamiento será implementado haciendo un *crossover*. El *crossover* permitirá que la nueva partícula (partícula en movimiento) tenga información de ambas partículas (*leader* y la partícula en movimiento).

Para aplicar un crossover entre dos partículas, debemos calcular primero la distancia entre ellas aplicando la formula de la Ecuación 1. Luego debemos calcular el punto de cruce de ambas

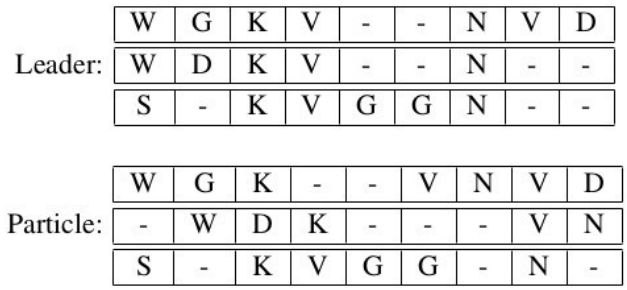

Figura. 2. Ejemplo de la partícula *leader* y una partícula cualquiera. Ambas representan una posible solución a un problema de alineamiento de 3 secuencias de aminoacidos.

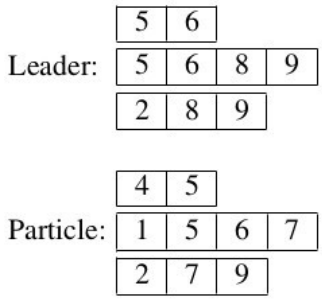

Figura. 3. Ejemplo de la partícula *leader* y una partícula cualquiera. En este caso solo estamos registrando la posición de cada *gap* insertado.

partículas, para esto aplicamos la formula de la Ecuación 2, en este caso $length$ representa la longitud de las secuencias.

$$distance = \frac{matchingGaps}{totalGaps} \tag{1}$$

$$crossPoint = rand(1, distance * length) \tag{2}$$

Por ejemplo, dada las partículas de la Figura 3 y suponiendo que hemos obtenido un *crossPoint = 5*, utilizando la Ecuación 2. La partícula en movimiento se desplazaría y su nueva representación sería el resultado de aplicar un *crossover*. En la Figura 4 mostramos el resultado de aplicar el *crossover*. Para lograr esto, por cada secuencia del leader insertamos sus gaps en la nueva partícula que son menores o iguales al $crossPoint$, luego completamos insertando los *gaps* de la partícula en movimiento que tengan *gaps* mayores al $crossPoint$.

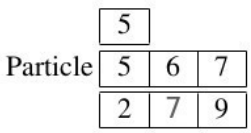

Figura. 4. Resultado del movimiento de una partícula luego de aplicar *crossover*.

3 Experimentos

En esta sección detallaremos las bases de datos utilizadas y los parametros del algoritmo PSO para poder replicar los resultados.

3.1 Bases de datos

Zablocki *et al.* (2009) propone un conjunto de 7 bases de datos. S1, S2, S3, S4, S5, S6 y S7 que el construyo a partir de un conjunto de secuencias de ADN. En la Tabla 1, presentamos el *ascension code* de cada secuencia utilizada. Ademas, el autor propuso un conjunto pequeño de secuencias para hacer pruebas rapidas, a este conjunto lo llamo S8. En nuestro caso, al ser una tesis un poco antigua, algunas secuencias ya no estaban disponibles en NCBI, y solo logramos obtener las secuencias de S6, S7 y S8.

Tabla. 1. Bases de datos utilizados por Zablocki *et al.* (2009). S1, S2, S3, S4, S5, S6 y S7 es el nombre que el autor definio para cada conjunto de secuencias. La segunda columna representa el *ascension code* de cada secuencia.

Dataset	Secuencias
S1	HCV2L1A10 HCV2L3A5 HCV2L3C1 HCV2L3C8 HCV2L3D4 HCV2L3E6 HCV2L3A7 HCV2L3A9 HCV2L3B2 HCV2L3B1
S2	HS06674 HS06675 HS06676 HS06677 HS06679
S3	TPAHISIN TNIHISIN TNHISIN TMIHISIN TMHISIN THHISIN TFHISIN TEHISIN TCUHISIN TCHISIN TBHISIN TAUHISIN TAHISIN TTHISIN TSHISIN TRHISIN TPYHISIN TPIHISIN TPHISIN TCAHISIN TLHISIN
S4	HI1U16764 HI1U16766 HI1U16768 HI1U16776 HI1U16778 HI1U16770 HI1U16774 HI1U16772
S5	HI1U16765 HI1U16767 HI1U16769 HI1U16771 HI1U16773 HI1U16775 HI1U16777 HI1U16779
S6	PP59651 PP59652 PP59653 PP59654 PP59655 PP59656
S7	AB023287 AB023286 AB023285 AB023284 AB023283 AB023279 AB023278 AB023276

3.2 Parametros

4 Resultados

5 Conclusiones

Supplementary Material

For supplementary material accompanying this paper, please visit https://github.com/arceda/bio-samples/tree/master/viral/viral_classification

References

Adetiba, E., J. A. Badejo, S. Thakur, V. O. Matthews, M. O. Adebiyi, and E. F. Adebiyi. 2017. "Experimental investigation of frequency chaos game representation for in silico and accurate classification of viral pathogens from genomic sequences." In *International Conference on Bioinformatics and Biomedical Engineering,* 155–164. Springer.

Archibald, J. M. 2018. *Genomics: A Very Short Introduction.* Vol. 559. Oxford University Press.

Bansiwal, A. 2014. "Analysis of Circulating Recombinant Forms (CRFs) of HIV-1 using Chaos Game Representation (CGR)." PhD diss., IISER M.

Edgar, R. C. 2004. "MUSCLE: multiple sequence alignment with high accuracy and high throughput." *Nucleic acids research* 32 (5): 1792–1797.

Higgins, D. G., A. J. Bleasby, and R. Fuchs. 1992. "CLUSTAL V: improved software for multiple sequence alignment." *Bioinformatics* 8 (2): 189–191.

Higgins, D. G., and P. M. Sharp. 1988. "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer." *Gene* 73 (1): 237–244.

Jeanmougin, F., J. D. Thompson, M. Gouy, D. G. Higgins, and T. J. Gibson. 1998. "Multiple sequence alignment with Clustal X." *Trends in biochemical sciences* 23 (10): 403–405.

Lebatteux, D., A. M. Remita, and A. B. Diallo. 2019. "Toward an alignment-free method for feature extraction and accurate classification of viral sequences." *Journal of Computational Biology* 26 (6): 519–535.

NIH. 2021. *The HUman Genome Project.* Web resource. National Human Genome Research Institute.

Pandit, A., and S. Sinha. 2010. "Using genomic signatures for HIV-1 sub-typing." *BMC bioinformatics* 11 (S1): S26.

Randhawa, G. S., M. P. Soltysiak, H. El Roz, C. P. de Souza, K. A. Hill, and L. Kari. 2020. "Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study." *bioRxiv.*

Solis-Reyes, S., M. Avino, A. Poon, and L. Kari. 2018. "An open-source k-mer based machine learning tool for fast and accurate subtyping of HIV-1 genomes." *PloS one* 13 (11).

Tanchotsrinon, W., C. Lursinsap, and Y. Poovorawan. 2015. "A high performance prediction of HPV genotypes by Chaos game representation and singular value decomposition." *BMC bioinformatics* 16 (1): 71.

Thompson, J. D., D. G. Higgins, and T. J. Gibson. 1994. "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice." *Nucleic acids research* 22 (22): 4673–4680.

Wang, L., and T. Jiang. 1994. "On the complexity of multiple sequence alignment." *Journal of computational biology* 1 (4): 337–348.

Xiong, J. 2006. *Essential bioinformatics.* Cambridge University Press.

Zablocki, F. B. R., *et al.* 2009. "Multiple sequence alignment using Particle swarm optimization." PhD diss., University of Pretoria.