Implementación de un metódo híbrido usando secuencias de ADN, para la obtención de probabilidades de enfemedades (diabetes tipo2, hipertensión e insuficiencia renal)

Vianney Morales Zamora, José Crispín Hernández Hernández, José Federico Ramírez Cruz

Avenida Instituto Tecnológico S/N C.P. 90300,
Apizaco, Tlaxcala, México
vimoza@hotmail.com,josechh@itapizaco.edu.mx,
Federico_ramirez@itapizaco.edu.mx

Abstrac. En este articulo se presenta un algoritmo hibrido que permite mediante la alineación de secuencias de ADN (algoritmo genético y programación dinámica), obtener una probabilidad (cadenas de Markov) de tener cierta enfermedad, como lo es diabetes tipo2, Hipertensión e insuficiencia renal. Debido a que en los últimos 3 años el número de personas con estas enfermedades han aumentado de manera considerable hasta en un 100% en nuestro país, es importante saber que probabilidad existe de tener las enfermedades anteriores, donde se utilizan secuencias de ADN, que debido a la gran cantidad de información de estas secuencias es necesario utilizar métodos híbridos que nos permitan mejorar la eficiencia de los resultados.

K**eywords:** Alineación de secuencias, algoritmo genético, programación dinámica, cadenas de Markov.

Introducción

En los últimos diez años las computadoras han tenido un papel muy importante en las áreas de la biología, y la medicina. La computación se ha enfocado al análisis de secuencias biológicas, cinco años atrás hubo muchos logros en la identificación de secuencias de genomas, como el de la mosca, y los primeros intentos en el proyecto de identificación de la secuencia del genoma humano [4][8]. Estas nuevas tecnologías y otras de alto desempeño como los arreglos de ADN (Ácido Desoxirribonucleico) y espectrografía de masas han tenido un progreso considerable [5]; y debido a que la información es demasiado grande, es necesario el uso de métodos híbridos que procesen de una manera efectiva y eficiente dicha información, como es el caso de los algoritmos genéticos y la programación dinámica para buscar la mejor alineación local óptima. Cuando se visualiza el ADN, ARN o las secuencias de proteínas como una cadena o lenguaje formal sobre alfabetos de cuatro nucleóti-

Advances in Intelligent and Information Technologies.
Research in Computing Science 50, 2010, pp. 275-288

dos o 20 amino ácidos, o como una representación gramatical y métodos de inferencia gramatical que pueden ser aplicados a varios problemas para análisis de secuencias biológicas[2].

Para obtener las probabilidades de enfermedades, primero se alinean las secuencias de ADN usando un algoritmo genético como se muestra en sección 2.1 y después el resultado será la entrada a la programación dinámica presentado en la sección 2.2, y teniendo las secuencias alineadas por pares, a continuación se obtienen las probabilidades de las enfermedades utilizando cadenas de Markov como se expresa en la sección 2.3.

Método propuesto

Algoritmo Genético para la alineación de secuencias por pares

La idea de este método es intentar generar alineaciones mediante reacomodaciones que simulan la inserción de gaps (huecos) y acciones de recombinación durante la replicación para obtener puntajes más altos para el alineamiento [10][15][18][21].

Se tomaron como entradas al algoritmo genético pares de secuencias, para los cual solo analizaremos los primeros 5 pares de secuencias. Cada par se ingreso al algoritmo genético, obteniendo como salida la mejor alineación.

1. Se toma el primer par de secuencias, y se pide el número de individuos (NI) a generar, y el número de generaciones a realizar (NG).
2. Se busca la cadena mas larga (x) del par de secuencias, se calcula la longitud de esta (L=lenght (x)), y se obtiene un factor (F=Lx0.25). Este factor da los espacios (gaps) extras que se van a usar para mover las cadenas, explícitamente se genera un número aleatorio entre 0 y F que será e l numero de gaps a insertar al inicio, y posteriormente se insertan gaps al final para hacer que las cadenas tengan la misma longitud. Así se obtienen los nuevos posibles alineamientos. Esto se realiza (Ni) veces de acuerdo con el número de individuos.
3. A continuación se saca el puntaje de cada alineamiento, teniendo en cuenta las calificaciones establecidas de acuerdo en la matriz de blosum62 que se muestra en la figura 1, cuya ecuación es la siguiente:

$$a_{ij} = \left(\frac{1}{\lambda}\right) \log\left(\frac{p_{ij}}{q_i * q_j}\right) \qquad (1)$$

Donde p_{ij} es la probabilidad de que dos aminoácidos *i* y *j* reemplacen uno al otro en una secuencia homóloga, mientras que q_i y q_j son las probabilidades últimas de encontrar los aminoácidos *i* y *j* en cualquier secuencia de proteína de forma aleatoria. El factor λ es un factor de escala para asegurar que, tras su aplicación y la de un

necesario redondeo al entero más cercano, la matriz contenga valores enteros dispersos y fácilmente tratables.

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Fig. 1**.** Matriz Blosum62

Utilizando este puntaje se selecciona la mitad de alineamientos para ser utilizados posteriormente en el proceso de selección y reproducción.

4. A continuación la otra mitad de los alineamientos pasan por un proceso de mutación, se obtiene el tamaño total de las secuencias(TT) tomando en cuenta los espacios anteriormente insertados, al igual se obtiene la diferencia del número de variables máximo (Nmax) y el menor numero de variables (Nmin) del par de alineamientos, y se generan números aleatorios entre 0 y ese rango (A=random(Nmax-Nmin)) que indican el numero de gaps a insertar, en seguida de acuerdo al tamaño total de las secuencias, se genera por cada gap a insertar un numero aleatorio entre 1 y (TT) que indican la posición donde será insertado cada espacios (A) realizándose este proceso con los pares de secuencias.

Y finalmente obtenemos la puntuación de las nuevas alineaciones conforme a la matriz de blosum62 expresada en la figura 1 y se compara el puntaje obtenido con los alineamientos anteriores; y si el puntaje del nuevo alineamiento es mayor, se seleccionan las nuevas alineaciones, de lo contrario se repite el proceso hasta que exista alguna mejora.

Con el proceso anterior se busca darle una mayor probabilidad a los alineamientos “fuertes” de ser elegidos.

5. Después tanto la primera mitad de los mejores alineamientos como la segunda mitad de los alineamientos mutados se juntan, y luego son calificados y reordenados. Que serán utilizados en el proceso de reproducción.
6. Ahora se realiza el proceso de reproducción basado en el método de torneo en el cual se selecciona aleatoriamente parejas de alineamientos; aquí el que tenga un mejor puntaje va a ser elegido como padre, de cada par de secuencias. Obteniendo los padres para el proceso de reproducción.
7. A partir de estos alineamientos se genera un número aleatorio dentro del rango de uno hasta la longitud de la cadena más corta (Nmin), ese será el punto de corte para cada par de secuencias. Este numero aleatorio (AC), contara solo cada carácter sin contar los espacios.
8. Después se intercambia la primera parte de la matriz superior con la segunda de la inferior y viceversa, y se obtiene la puntuación de cada hijo.
9. **S**e elije el hijo que tengan la mejor calificación y es agregado a la nueva generación.

Y de acuerdo al número de la población (NI), se realiza el proceso 7, 8 y 9.

Se hace mención que en el proceso de reproducción se eliminan las columnas llenas de gaps, para evitar que la secuencia crezca exponencialmente.

Una vez que se ha generado el número de hijos suficientes para cumplir con el total de la población (NI), se retorna al paso tres para seguir con el procedimiento hasta completar el número solicitado de generaciones (NG).

A continuación se muestra un ejemplo resultado de alinear con el algoritmo genético, la figura 2 muestra las alineaciones de entrada y la figura 3 muestra la salida obtenida con el algoritmo genético:

```
TCACCTGGGTGTGTGGGTGCCGTTCCAGGCTGTCAGAGCTCGCGTGGGGGTGTGGGTGCTGCTCCAGGCT
TTCGGAGCTCACCTGGGGGTGCAGGGTGCTGTTCCAGGCTGTCAGATGCTCACCTGGGGGTGTGGTTGCT

GCTCCAGGCTGTCAGATGCTCACTTGGGGGTGCAGCGTGCTGTTCCAGGCTGTCAGATGCTAACCTGGGG
TTGTGAGAGCTGTTGCAGGCTGTCAGATGCTCACTGGGGGTGTGCGTGCTGCTCCAGCCTGTCAGATGCT

CACCTGGGGGTGTGGGTGCTGTTCCAGGATATCAGATGCTCACCTGGGGGTGTGGGTGCTGCTCCAGGCT
GTCGGATGCTCACCTGGGGGTGTCGGTGCTGTTCCAGGCTGTTGGATGCTCACCTGGGGGTGTGGTTGCT

GCTCCAGGCTGTCAGGTGCTCACTTGGGGGTGCAGCGTGCTGTTCCGGGCTGTCAGATGCTCACCTGGGG
TTGTGAGAGCTGTTCCAGGCTGTCAGATGCTCACCTGGGGGTGTGGGTGCTGTTCCAGGCTGTCAGATGC

TCACCTGTTGGTGTGGGTGCTGCTTCAGGCTGTCAGATGCTCACCTGGAGTGTGGGTGCTGCTCCAGGCT
GTCTGATCCTCACTCGCGGGTGCAGGGTTCTGTTCCAGGCTGTCAGATGCTTGCCTGGGGGTGTGGTTGC
```

Fig. 2. Ejemplo de la secuencia de entrada

Solución obtenida:

```
puntuacion1 =  91.88
Alineacion1 =
---------------TCACCTGGGTGTG-TGGGTGCCGTTCCAGGCTGTCAGA-GCTCGCGTGGGGGTGTGGGTGCTGCTCCAGGCT
           |||||||||||:||||||||||||||||||||||||||:|||||||||| ||| |  |||
TTCGGAGCTCACCTGGGGGTGCAGGGTGCTGTTCCAGGCTGTCAGATGCTCACCTGGGGGTGT--G-G-T--T----GCT

puntuacion2 =  87
Alineacion2 =
---------------GC---T-C-CAGGCTGTCAGATGCTCACTTGGGGGTGCAGCGTGCTGTTCCAGGCTGTCAGATGCTAACCTGGGG
             ||  |   |||||||||||||||||||||||||||||||||   ||| ||||||||||||||| ||
TTGTGAGAGCTGTTGCAGGCTGTCAGATGCTCAC-TGGGGGTG-TGCGTGCTGCTCCAGCCTGTCAGATG-----CT----

puntuacion3 =  88.777
Alineacion3 =
-----------------CACCTGGGGGTGTGGGTGCTGTTCCAGGATATCAGATGCTCACCTGGGGGTGTGGGTGCTGCTCCAGGCT
              |||||||||||||||||||||||||||:| :|||||||||||||||||||| ||| |  |||
GTCGGATGCTCACCTGGGGGTGTCGGTGCTGTTCCAGGCTGTTGGATGCTCACCTGGGGGTGT--G-G-T--T----GCT

puntuacion4 =  88.787
Alineacion4 =
---------------GC-----TCCAGGCTGTCAGGTGCTCACTTGGGGGTGCAGCGTGCTGTTCCGGGCTGTCAGATGCTCACCTGGGG
             ||     |||||||||||||:|||||||||||:| |||||||||||:|||||||||||  ||||||||     |
TTGTGAGAGCTGTTCCAGGCTGTCAGATGCTCACCTGGGGGTGTGG-GTGCTGTTCCAGGCTGTCAGATG-----C-----

puntuacion5 =  77.99
Alineacion5 =
--TC---A----CCT-GTTGGTGTGGGTGCTGCTTCAGGCTGTCAGATGCTCACCTGGAGTGTGGGTGCTGCTCCAGGCT
  ||   |  |||  | ||||              ||| ||||||||||||||||||:|||||||||||| || : ||
GTCTGATCCTCACTCGCGGGTGCAGGGTTCTGTTCCAGGCTGTCAGATGCTTGCCTGG-G-G-GTGTG--G-T--T-GC-
```

Fig .3. Secuencia alineada y sus puntajes de alineación, obtenidas con el algoritmo genético. La alineación obtenida se introduce como entrada en la programación dinámica.

Programación dinámica para la alineación de secuencias

La Programación Dinámica (PD) es una alternativa de descomposición en que se resuelven subproblemas más pequeños y luego se juntan asumiendo que en cada etapa futura se tomaran decisiones correctas [13].

Los pasos del diagrama se definen a continuación:

1.-Se introducen en pares las secuencias para nuevamente alinearlas.
2.- Se elabora su matriz de encajes, como se muestra en la figura 4 y si existe coincidencia de letras se asigna a la coincidencia un 1 y si no hay se coloca un cero.

		1	2	3	4	5	6	7	8	9	10	
		A	T	C	C	G	C	T	T	A	C	A
1	C	0	0	1	1	0	1	0	0	0	1	
2	T	0	1	0	0	0	0	1	1	0	0	
3	C	0	0	1	1	0	1	0	0	0	1	
4	C	0	0	1	1	0	1	0	0	0	1	
5	T	0	1	0	0	0	0	1	1	0	0	
6	A	1	0	0	0	0	0	0	0	1	0	
B	7	G	0	0	0	0	1	0	0	0	0	0

Fig. 4. Matriz de encajes

El camino que se eligió para este problema fue el diagonal como se muestra la figura 5, en el que se puede ir seleccionando un paso a la derecha y así mismo

hacia abajo, formando un camino diagonal a partir de la parte superior izquierda, como se muestra a continuación.

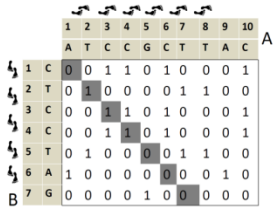

Fig. 5. Movimiento en diagonal

Buscar el mejor alineamiento entre 2 secuencias es buscar el camino que obtenga la mejor puntuación. La estrategia de la programación dinámica es dividir el problema global en subproblemas. Así se busca primero el mejor camino que empieza en cada una de las casillas. Empezamos por la última fila. En esta fila la puntuación del mejor camino para cada casilla coincide con la puntuación de encaje. En las casillas de la penúltima fila el mejor camino es de un sólo paso. Para computar las casillas del resto de las filas también basta con dar un sólo paso porque cada casilla a la que nos podemos mover lleva ya acumulada la mejor puntuación que se puede obtener pasando por ella. Así el problema de elegir el mejor camino que empieza en una casilla se reduce a elegir la siguiente casilla con mejor puntuación.

Se elije la máxima puntuación de las casillas marcadas en gris y se le suma a la puntuación de encaje tal como se muestra en la figura 6, y el resultado de este proceso se muestra en la figura 7.

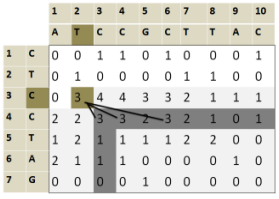

Fig.6. Obtención de puntajes máximos

Así se van a ir generando los caminos de alineación haciendo un recorrido en diagonal, y una vez obtenidos las posibles soluciones, se elije la mayor puntuación como se presenta en la figura 8, y esto se realiza con cada par de secuencias. Aplicando el ejemplo anterior de secuencias de ADN, las siguientes alineaciones obtenidas se ilustran en la figura 9.

		1	2	3	4	5	6	7	8	9	10
		A	T	C	C	G	C	T	T	A	C
1	C	5	4	5	4	3	4	2	1	1	1
2	T	4	5	4	3	3	2	3	2	1	0
3	C	3	3	4	4	3	3	2	1	1	1
4	C	2	2	3	3	2	3	2	1	0	1
5	T	1	2	1	1	1	2	2	0	0	
6	A	2	1	1	1	0	0	0	0	1	0
7	G	0	0	0	0	1	0	0	0	0	0

Fig. 7. Resultados de puntajes máximos

		1	2	3	4	5	6	7	8	9	10
		A	T	C	C	G	C	T	T	A	C
1	C	5	4	5	4	3	4	2	1	1	1
2	T	4	5	4	3	3	2	3	2	1	0
3	C	3	3	4	4	3	3	2	1	1	1
4	C	2	2	3	3	2	3	2	1	0	1
5	T	1	2	1	1	1	2	2	0	0	
6	A	2	1	1	1	0	0	0	0	1	0
7	G	0	0	0	0	1	0	0	0	0	0

Fig. 8. Alineamientos que alcanzan el máximo puntaje

En la figura 10 se presentan los resultados obtenidos de introducir las secuencias en un algoritmo genético y después en la programación dinámica.

Con lo anterior podemos comparar nuestros resultados, resaltando la mejora en la alineación usando un algoritmo genético y programación dinámica, como se muestra en la Tabla 1

Por lo tanto el método propuesto de usar un algoritmo genético y después programación dinámica, nos permite tener una mejor alineación de secuencias, como se mostro en la tabla anterior. Y ya teniendo las secuencias alineadas, ahora el siguiente paso es buscar patrones que nos permitan identificar enfermedades y poder tener una probabilidad de contraer alguna enfermedad y así poder prevenir problemas futuros. Para la solución de dicho problema se hace uso de los modelos de Markov.

Obtención de probabilidades utilizando cadenas de Markov

Uno de los aspectos fundamentales en la Bioinformática, es el diseño de modelos matemáticos que interpreten los sesgos de las secuencias biológicas e identifiquen patrones de las mismas. Uno de los modelos más utilizados para este propósito, lo son las cadenas de Markov.

Una cadena de Markov es una secuencia de variables aleatorias X (p) donde la distribución de probabilidad de cada X (i) depende de una cantidad de k valores precedentes X (i-1), X (i-2),.., X(i-k) .

Cada uno de los nucleótidos de una secuencia puede diferenciarse por su base correspondiente; por tal motivo es usual hablar indistintamente de nucleótido o de base.

```
Alineacion1 =

--------TCACCTGGGTGTG-TGGGTGCCGTTCCAGGCTGTCAGA-GCTCGCGTGGGGGTGTGGGTGCTGCTCCAGGCT
        ||||||||| ||| :|||||| |||||||||||||||| ||||:| |||||||||  | | |  |    |||
TTCGGAGCTCACCTGGGGGTGCAGGGTGCTGTTCCAGGCTGTCAGATGCTCACCTGGGGGTGT--G-G-T--T----GCT


puntuacion1 =

   92.3333


Alineacion2 =

--------GC--TC-CAGGCTGTCAGATGCTCACTTGGGGGTGCAGCGTGCTGTTCCAGGCTGTCAGATGCTAACCTGGGG
        ||  |  |||||||||||||||||||| |||||||| :|||||||| ||||| ||||||||||     ||
TTGTGAGAGCTGTTGCAGGCTGTCAGATGCTCAC-TGGGGGTG-TGCGTGCTGCTCCAGCCTGTCAGATG-----CT----


puntuacion2 =

   88.6667

Alineacion3 =

----------CACCTGGGGGTGTGGGTGCTGTTCCAGGATATCAGATGCTCACCTGGGGGTGTGGGTGCTGCTCCAGGCT
          |||||||||||||| |||||||||||||| |:| :|||||||||||||||||||||  | | |  |    |||
GTCGGATGCTCACCTGGGGGTGTCGGTGCTGTTCCAGGCTGTTGGATGCTCACCTGGGGGTGT--G-G-T--T----GCT


puntuacion3 =

    89


Alineacion4 =

--------GC---TCCAGGCTGTCAGGTGCTCACTTGGGGGTGCAGCGTGCTGTTCCGGGCTGTCAGATGCTCACCTGGGG
        ||   |||||||||||||:|||||||| |||||||| :| ||||||||||:||||||||||||     |
TTGTGAGAGCTGTTCCAGGCTGTCAGATGCTCACCTGGGGGTGTGG-GTGCTGTTCCAGGCTGTCAGATG-----C-----


puntuacion4 =

    89
Alineacion5 =

-TC--A----C-CT-GTTGGTG-TGGGTGCTGCTTCAGGCTGTCAGATGCTCACCTGGAGTGTGGGTGCTGCTCCAGGCT
 ||  |    | || |  |||| :|||| ||| | |||||||||||||||| :||||| | | | |||  | |  : ||
GTCTGATCCTCACTCGCGGGTGCAGGGTTCTGTTCCAGGCTGTCAGATGCTTGCCTGG-G-G-GTGTG--G-T--T-GC-


puntuacion5 =

    78
```

Fig.9. Resultados obtenidos únicamente con la PD

Fig. 10. Resultados obtenidos con el algoritmo genético (AG) y programación dinámica (PD).

Tabla1. Resultados en la alineación

ALGORITMO GENÉTICO	PROGRAMACIÓN DINÁMICA	AG-PD
91.88 %	92.3333 %	95 %
87 %	88.6667 %	89.888 %
88.777 %	89 %	90 %
88.787 %	89 %	89.888 %
77.99 %	78 %	78.777 %

Los modelos de cadenas de Markov aplicados a secuencias de DNA permiten calcular la probabilidad de ocurrencia de un nucleótido (es decir, de la base correspondiente a un nucleótido dado) b en la secuencia dependiendo de los k nucleótidos inmediatamente anteriores a b en la secuencia.

Los modelos de Markov se utilizan bastante en el análisis de secuencias de datos biológicos. Los más utilizados son las cadenas de Markov de orden fijo, en las cuales el contexto (es decir, la secuencia de k nucleótidos precedentes) se utiliza en cada posición donde se desea hallar la probabilidad de ocurrencia de un nucleótido dado.

Cualquier modelo de cadenas de Markov de orden fijo predice un nucleótido de la secuencia de ADN usando los nucleótidos anteriores de la secuencia; el modelo de mayor uso es el de cadenas de Markov de quinto orden.

A diferencia de un modelo simple como lo es el IDD (Independent Identically Dsitributed) donde la ocurrencia de cada nucleótido (A, T, G, C) seria independiente de los demás, en las cadenas de Markov el valor tomado de una variable es dependiente del valor tomado en el estado anterior.

En el modelo de Markov, la probabilidad de encontrar un patrón está dada por la probabilidad (P_1) del modelo de distribución de α en la primera posición y la probabilidad condicional de β en la posición i>1 (P_2), es decir,

$$\mathbf{P(x \mid M) = P_1 (X_1)} \quad \prod_{i=2.....n(x)} \quad \mathbf{P_2 (X_1|X_{1-1})} \qquad \mathbf{(2)}$$

P_2 se calcula con la siguiente fórmula:

$$P_2 (\beta|\alpha) = P (\alpha\beta)/ P (\alpha) \qquad \mathbf{(3)}$$

Para esto es necesario conocer las frecuencias de cada uno de los nucleótidos y cada uno de los pares en la secuencia dada.

Por ejemplo: Dada la siguiente secuencia de 25 nucleótidos,

AACGTCTCTATCATGCCAGGATCTG

¿Cuál sería la probabilidad de encontrar el patrón **CAAT**?

Se obtienen las frecuencias de cada uno de los nucleótidos y cada uno de los pares en la secuencia dada.

A = 6/25, C = 7/25, T = 7/25, G = 5/25

Donde el numerador en la fracción es el número de veces que existe la letra en la cadena de secuencias, y el denominador es el tamaño de la cadena, y en el caso de los pares será el tamaño total de cadena menos 1.

(AA) = 1/24 (CA) = 2/24 (TA) = 1/24
(GA) = 1/24 (CG) = 1/24 (GT) = 1/24

(TC) = 4/24 (GG) = 1/24 (AC) = 1/24
(CT) = 3/24 (AT) = 3/24 (TG) = 2/24
(GC) = 1/24 (CC) = 1/24 (AG) = 1/24

Con lo cual podemos calcular P_2 (β|α), por ejemplo del di-nucleótido AC es:

P_2 (C|A) = $P_{AC/}$ P_A = (1/24)/ (6/25) = 25/144

Por lo tanto la tanto la probabilidad de encontrar CAAT usando cadenas de Markov de primer orden es*:*

P(C) P (A|C) P (A|A) P (T|A)
P(C) P (A|C) P (A|A) P (T|A)
(7/25) (2/23)/ (7/25) (1/24)/ (6/25) (3/22)/ (6/25) =
(7/25) x (50/161) x (25/144) x (75/132) =0.00857
=0.0086

Así la probabilidad de encontrar el patrón **CAAT,** en la secuencia AACGTCTCTATCATGCCAGGATCTG, es de 0.0086.

En este caso se busca la probabilidad de encontrar patrones de enfermedades en las secuencias. Las enfermedades a encontrar serán: Diabetes, Hipertensión e Insuficiencia Renal. Cuyos patrones, serán evaluados en las secuencias para obtener una probabilidad de tener dicha enfermedad, y debido a que las secuencias ya se encuentran alineadas, la identificación de patrones será de una manera más rápida.

Para el siguiente ejemplo se tomo aleatoriamente de las bases de datos de las enfermedades 10 personas. Cuyas entradas son las siguientes:

Tabla 2. Entradas de personas con enfermedad

No. De persona	Enfermedad
1	Diabetes tipo 2
2	Diabetes tipo 2
3	Diabetes tipo 2
4	Insuficiencia renal
5	Insuficiencia renal
6	Hipertensión
7	Hipertensión
8	Hipertensión
9	Diabetes tipo 2
10	Insuficiencia renal

Y los resultados de las probabilidades obtenidas con las cadenas de Markov son los siguientes:

Tabla3. Probabilidades obtenidas

No de persona	Diabetes tipo2	Insuficiencia renal	Hipertension
1	**0.87**	0,20	0,15
2	**0.84**	0,02	0,01
3	**0.80**	0,15	0,10
4	0,19	**0.80**	0,17
5	0,13	**0.85**	0,03
6	0,17	0,01	**0.85**
7	0,15	0,10	**0.90**
8	0,11	0,01	**0.89**
9	**0.91**	0,03	0,11
10	0,02	**0.90**	0,17

Por lo que obtenemos:

Tabla 4. Resultados de las probabilidades respecto a su enfermedad

No de persona	Enfermedad	Probabilidad obtenida
1	*Diabetestipo2*	0,87
2	*Diabetestipo2*	0,84
3	*Diabetestipo2*	0,80
4	*Insuficienciarenal*	0,80
5	*Insuficienciarenal*	0,85
6	*Hipertensin*	0,85
7	*Hipertensin*	0,90
8	*Hipertensin*	0,89
9	*Diabetestipo2*	0,91
10	*Insuficienciarenal*	0,90

Conclusiones

Debido a la gran cantidad de información que se procesa al manejar cadenas de ADN, se recurren a diversos métodos y técnicas computacionales o bien sistemas híbridos que nos permiten la manipulación de dicha información para analizarla y así poder obtener mayor información de las secuencias que forman el ADN del ser humano y poder desarrollar sistemas que permitan facilitar el manejo y acceso de información a especialistas en el área de la biología molecular. Considerando estos sistemas computacionales como una herramienta esencial de ayuda.

Tras haber realizado algunas pruebas con diferentes grupos de secuencias, se evidenciaron varios aspectos que hay que tomar en cuenta para obtener buenos resultados, con este algoritmo. Dado que uno de los problemas de los algoritmos genéticos en general es la existencia de los máximos locales, en algunos casos el algoritmo se ejecuta varias veces sobre el mismo conjunto de secuencias y escoger entre las diferentes corridas el mejor resultado. Por otro lado, es necesario jugar un poco con los parámetros del algoritmo tales como el número y longitud de gaps (huecos), que se pueden insertar en las secuencias mutadas, y el tamaño de la población. Esta variación en los parámetros es muy importante porque no es lo mismo trabajar con

secuencias que son muy similares entre si, a trabajar con secuencias altamente disparejas.

Con las pruebas también se pudo detector que cuando se está realizando la implementación del algoritmo, es de vital importancia evitar que las secuencias aumenten de tamaño de manera exagerada porque si no se toman tales precauciones, puede suceder que todos los alineamientos estén llenos de secuencias que son en un alto porcentaje gaps; y si este error se propaga el resultado del algoritmo puede ser muy pobre. Por lo que la propuesta es aplicarle otro filtro a esas secuencias alineadas obtenidas del algoritmo genético, pasando por la programación dinámica para volver a alinearlas, y así poder optimizar su alineación.

Finalmente al tener una buena alineación de secuencias, se prosigue a identificar patrones de enfermedades en las secuencias mediante el uso de otro modelo estocástico como lo son la cadena de Markov, que nos permitirán obtener una probabilidad de tener una enfermedad y poder prever problemas graves y se puedan atender.

El inconveniente que presentan los modelos de cadenas de Markov es que el aprendizaje de algunos modelos puede producir dificultad cuando no se tienen suficientes datos de un contexto determinado. Sin embargo en este caso si contamos con los patrones a trabajar y nuestra salida requerida es una probabilidad, cuyo problema es fácil resolver con las cadenas de Markov.

Lo que se tiene es un sistema que nos puede mostrar diferentes soluciones a un problema y es ahí donde la capacidad de interpretar estas propuestas puede hacer la diferencia.

Bibliografía

1. D.J. Attwood, T.K y Parry-Smith. Introducción a la Bioinformática. Madrid, 2002.
2. Pierre Balde and Soren Brunak. Bioinformatics: The Machine Learning approach. Cataloging in publication data, 2001.
3. G. Baldi, P. y Pollastri. A machine learning strategy for protein analysis, volume 17. 2002.
4. U Bandyopadhyay, S y Maulik. Gene identification: classical and computational intelligence approaches. Man and cybernetics, Part C: Applications and Reviews, IEEE, 3:40_56, 2005.

5 M. Bertone, P. y Gerstein. Integrative data mining: new direction in bioinformatics engineering in medicine and biology. IEEE Magazine, 20:33_40, 2001.

6. Bethesda. Los riñones y la insuficiencia renal. National Kidney and Urologic Diseases Information Clearinghouse, Agosto 2009.
7. MD Bethesda. Diabetes. National Institute Health, December 2009.
8. Liaison Branch. National human genome research institute. Communications and Public Liasion Branch, December 2009.
9. T.Koike R. Lopez T.J. Gibson D.G. Higgins J.D. Thompson Chenna, H.Sugawara. Multiple sequence alignment with the clustal series of programs. IEEE, 3:37_45, 2003.
10. M; Merler S; Jurman G Furlanello, C; Serafini. Semi supervised learning for molecular profiling. IEEE Computational Biology and Bioinformatics, 2:110_118, 2005.
11. D.B; Knowles J. Handl, J; Kell. Multiobjetive optimization in bioinformatics and computational biology. IEEE ACM Transactions on Computational Biology and Bioinformatics, 4:279_292, 2007.

12. M Hawkins, J; Boden. The applicability of recurrent neural networks for biological sequence analysis computational biology and bioinformatics. IEEE ACM, 2:243_253, 2005.
13. Moulton J. Stoye, V and A.W. And efficient implementation of the divided and conquer approach to simultaneous multiple sequence alignment. Biosci, 6, 2000.
14. D.G. Higgins J.D. Thompson and T.J. Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighling, positions- specific gap penalties and weight matrix choice. Nucleic Acid Research, 22, 1999.
15. A Keedwell L, Narayanan. Discovering gene networks with a neural-genetic hybrid computational biology and bioinformatics. IEEE ACM, 2:231_242, 2007.
16. Nelson Fausto Kumar, Abul K. Hypertensive vascular disease, volume 8. Saunders (Elsevier), 2009.
17. Sverlin A Manavsky and Giorgio Valle. Cuda compatible CPU as efficient hardware accelerators for smith-waterman sequence aligment. CRIBI, 2008.
18. C. Notredame and D.G Higgins. Saga: Sequence alignment by genetic algorithm. Nucleic Acid Research, 24, 2000.
19. John L. Semmlow. Biosignal and Biomedical Image Processing, Matlab based application. 2004.
20. Albert W; Sat Sharma y Claude Kortas. Hypertension and the kidney. Department of Medicine, University of Western Ontario, Canada, February 2010.
21. H. Carrillo y D.J. Lipman. The multiple sequence alignment problems in biology. SIAM J. Appl., 48:4_7, 2000.
22. Yufeng Wang Yijuan Lu; Qi Tian; Feng Liu; Sanchez M. Interactive semi supervised learning for microarray analysis. IEEE ACM, 4:190203, 2007.