Universidad del Rosario

MACC Matemáticas Aplicadas y Ciencias de la Computación

Algoritmos Y Estructuras De Datos

Proyecto Segunda Entrega

Juan L. Ávila M. / Estefania Laverde B. / Juan P. Sierra U. *Fecha: 25 de Abril del 2020*

TÍTULO:

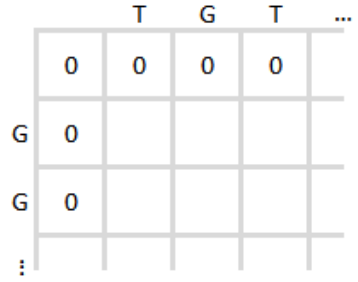
Análisis de mutaciones moleculares.

OBJETIVOS ALCANZADOS:

- **Cambios hechos al enunciado del problema:**
  - Se ha determinado partir de una cadena de ADN proveniente de una individuo sano para que se pueda comparar con las demás cadenas y señalar la similitud entre ellas. Según el resultado de la comparación se decide si el trozo de cadena presenta mutaciones moleculares significativas o no.
  - Con un diccionario en el que se contienen las diversas enfermedades dada la mutación del código genético, se da un resultado de la posible enfermedad genética.
- **Avance realizado en el diseño e implementación de la herramienta computacional:**
  - Hasta el momento hemos terminado de implementar el algoritmo de Smith - Waterman. Este recibe como entrada 2 cadenas de código genético y un número que indica la penalización por cada casilla diferente. Este algoritmo, no necesariamente se preocupa porque una cadena esté contenida en la otra, sino que el objetivo es buscar las subcadenas que más se parecen.
  - Para la búsqueda de la respectiva enfermedad se ha decidido implementar Hash Tables.
- **Retos enfrentados:**
  - Búsqueda y entendimiento del algoritmo más óptimo para encontrar la similitud entre cadenas.
  - Entender el funcionamiento de la base de datos del National Center for Biotechnology Information Search database para poder extraer la información relevante al proyecto.

ALGORITMOS Y ESTRUCTURAS DE DATOS:

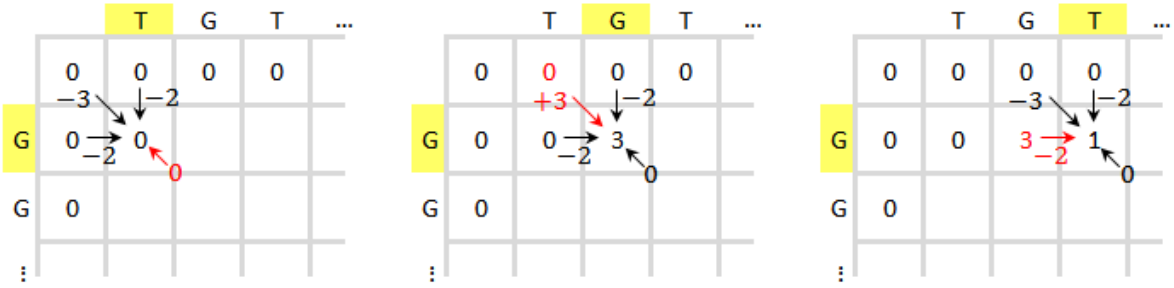
Como lo expusimos anteriormente, hemos acoplado el algoritmo de Smith - Waterman. Para ello, utilizamos un arreglo dinámico de arreglos dinámicos para emular una matriz. Este algoritmo que recibe como parámetros dos cadenas A y B , y un número de penalización P. Sea M y N la longitud de las cadenas A y B respectivamente. Primero, se crea una matriz de tamaño (m+1,n+1) ya que la primera fila y la primera columna se rellenan con 0 como se muestra a continuación.



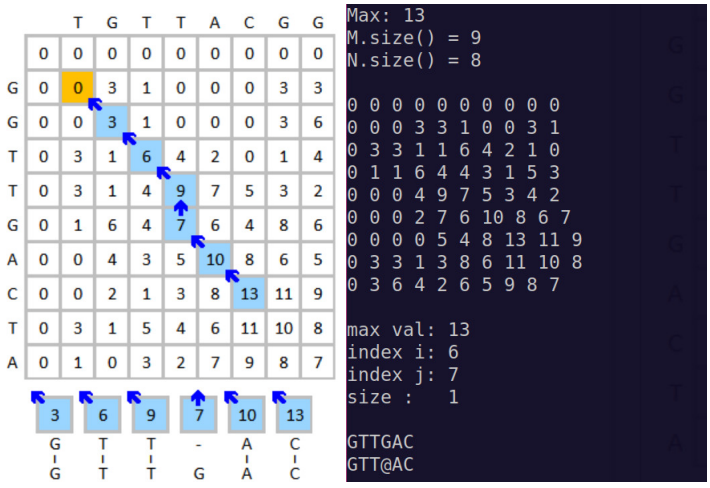
Para rellenar cada cuadrícula de la matriz se usa la siguiente ecuación:

$$H_{ij} = \text{máx} \begin{cases} H_{i-1,j-1} + s\left(a_i, b_j\right) \\ \text{máx}_{k \geq 1} \left\{H_{i-k,j} - P_k\right\}, \quad (1 \leq i \leq n, 1 \leq j \leq m) \\ \text{máx}_{l \geq 1} \left\{H_{i,j-l} - P_i\right\} \\ 0 \end{cases}$$

Donde H, es la matriz, $s(a_1, b_1)$ es el puntaje que tienen 2 carácteres iguales. En nuestro caso, si son iguales $s = 3$, de lo contrario $s = -3$. P, como habíamos dicho es la penalización asignada. Estos son algunos ejemplos de relleno de la matriz, dado $P = 2$:



Una vez tenemos la matriz completa, debemos encontrar, las subcadenas que más se parecen. Para eso empezamos a recorrer la matriz desde la casilla con el mayor puntaje (si son varias se hace el proceso en cada una de ellas) y vamos avanzando a la casilla de la izquierda, de la diagonal izquierda o de la derecha que tenga más puntaje hasta llegar a la casilla 0. Aquí una representación y la solución que nuestra implementación retorna:




```
Max: 13
M.size() = 9
N.size() = 8

0 0 0 0 0 0 0 0 0
0 0 0 3 3 1 0 0 3 1
0 3 3 1 1 6 4 2 1 0
0 1 1 6 4 4 3 1 5 3
0 0 0 4 9 7 5 3 4 2
0 0 0 2 7 6 10 8 6 7
0 0 0 0 5 4 8 13 11 9
0 3 3 1 3 8 6 11 10 8
0 3 6 4 2 6 5 9 8 7

max val: 13
index i: 6
index j: 7
size :   1

GTTGAC
GTT@AC
```

NUEVOS OBJETIVOS:

- Relacionar los resultados del algoritmo de comparación con sus respectivas enfermedades, ya que no todas las mutaciones desencadenan en una enfermedad.
- Diseñar un algoritmo que utilizando el que ya hemos implementado, pueda comparar la cadena de una persona con la de otras que presentan enfermedades y encontrar niveles de similitud que nos puedan llevar a identificar alguna enfermedad en ella.

REPOSITORIO:
https://github.com/JuanPab3/Molecular-mutations-analysis