Metodología y algoritmo para la construcción de una matriz de sustitución generalizada para alfabetos arbitrarios que describen secuencias biológicas

YERMINSON DONEY GONZALEZ MUÑOZ

200843846

yerminson.gonzalez@correounivalle.com

Irene Tischer, Ph.D

irenetischer@gmail.com

Facultad de Ingeniería

Escuela de Ingeniería de Sistemas y Computación

Programa Académico de Ingeniería de Sistemas

Santiago de Cali, Junio 5 de 2015

Índice

Índice de figuras

Índice de cuadros

Algoritmos

Agradecimientos

Primero que todo quisiera agradecer a Dufay Muñoz, mi madre, quien me ha apoyado siempre para salir adelante y quien confía en que pueda lograr todas las cosas que me proponga en la vida, seguido me gustaría agradecer a mi familia por su constante apoyo en todo sentido y que siempre han estado ahí a pesar de las dificultades, a Cristian Leonardo Rios y a Maria Andrea Cruz quienes fueron personas muy importantes a lo largo de mi carrera universitaria y con quienes compartí y aprendí muchísimo durante este recorrido académico y con quienes hasta el día de hoy preservo una bonita amistad. Agradezco también a Julian Rodriguez, Cristhian David Marin, Carlos Andres Delgado, Paola Garcia , Karen Martinez y Sandra Mayorga que siempre estuvieron pendientes y con las mejores intenciones de que el trabajo de grado se pudiera concluir satisfactoriamente y agradezco especialmente a Karen ya que me ayudo a superar las desmotivaciones que llegue a tener relacionadas con este trabajo de grado me ayudo a ver mas allá de los problemas que en algún momento todos podamos llegar a tener.

También agradezco a mi directora Irene Tischer quien a pesar de las dificultades me apoyo en la realización de este trabajo de grado y quien siempre estuvo pendiente para solucionar problemas que pudieran surgir en el camino.

Finalmente agradezco a todas las personas que de alguna forma mostraron su apoyo en determinado momento dándome ánimos para seguir y salir adelante en el desarrollo de este trabajo de grado.

Resumen

En este documento se presenta una propuesta para desarrollar e implementar una metodología para la construcción de una matriz de sustitución generalizada para alfabetos arbitrarios que describen secuencias biológicas. El desarrollo de esta propuesta se realiza mediante el seguimiento de un conjunto de pasos que van desde el entendimiento conceptual del problema, el desarrollo de la metodología, y la implementación de los algoritmos correspondientes que permitan la construcción de una matriz de sustitución generalizada, la cual se utilizará en un algoritmo de alineamiento que sirva para probar algunos casos de estudio. A partir de la base de datos BLOCKS se obtuvieron nuevas bases de datos que representan tuplas y tripletas formadas a partir de las secuencias de proteínas pertenecientes a cada uno de los bloques. Mediante el uso del algoritmo BLOSUM se obtuvieron matrices de sustitución generalizadas mostrando así que es posible desarrollar una metodología y construirlas mientras se tenga claridad sobre el alfabeto que se va a utilizar.

Palabras clave: Aminoácidos, Base de datos, BLOSUM, PAM, Matriz de sustitución**,** Alineamiento**,** *log ratios.*

1. INTRODUCCIÓN

Las matrices de sustitución han sido una herramienta muy importante en la alineación de secuencias, ya que permiten obtener mejores alineamientos a partir de la información que contienen. Las matrices de sustitución relacionadas con proteínas se crearon a partir de diferentes modelos como el de Dayhoff -de donde nacen las matrices PAM- y el modelo de Henikoff and Henikoff -de donde nacen las matrices BLOSUM-. Estos modelos nos muestran una matriz donde se puntúa qué tan bueno es sustituir un aminoácido por otro. Esta información es muy importante ya que permite establecer las relaciones que pueden existir entre los aminoácidos que de algún modo se han podido inferir mediante el estudio de grandes grupos secuencias de proteínas, por ejemplo diferentes grupos de familias de proteínas, ya sea realizando alineamientos o estableciendo relaciones a partir de propiedades comunes que pueden compartir ciertos aminoácidos.

En el desarrollo de este trabajo se profundiza en la creación de una metodología para la construcción de una matriz de sustitución generalizada, que pueda ser utilizada en algoritmos de alineamiento con el fin de obtener un acercamiento menos específico al que se tiene normalmente usando solo aminoácidos.

Para lograr esto, hemos decidido primero realizar un estudio sobre las diferentes matrices de sustitución utilizadas por los algoritmos de alineamiento en la actualidad, establecer las características generales que involucran la creación de una matriz de sustitución y las propiedades que describe cada matriz resultante de acuerdo al grupo de proteínas y la manera en la que fue construida, seleccionar la implementación de las matrices de sustitución actuales que favorezcan la creación de un algoritmo que se pueda usar de manera generalizada para construir nuevas matrices con un alfabeto generalizado, y finalmente ejecutar pruebas con las matrices de sustitución generalizadas obtenidas para los diferentes alfabetos, aplicando un algoritmo de alineamiento que permita validar la información obtenida y que describe la metodología.

De esta manera se presenta una metodología orientada a la construcción de matrices de sustitución generalizadas basadas en familias de proteínas que están representadas en la base de datos BLOCKS.

2. OBJETIVOS

2.1. Objetivo General

Desarrollar una metodología que permita construir una matriz de sustitución generalizada para alfabetos arbitrarios que describen secuencias biológicas e implementar los algoritmos subyacentes.

2.2. Objetivos Específicos

1. Estudiar las matrices de sustitución usadas actualmente para establecer los elementos básicos en la construcción de una matriz de sustitución que permitan su generalización.
2. Diseñar una metodología para la construcción de una matriz de sustitución generalizada, así como los algoritmos que permitan su construcción utilizando alfabetos arbitrarios.
3. Desarrollar un algoritmo para el alineamiento de secuencias en alfabetos arbitrarios utilizando matrices de sustitución generalizadas en su desarrollo.
4. Analizar la metodología y los resultados de los algoritmos desarrollados mediante su aplicación en diferentes casos de estudio.

3. MARCO TEÓRICO

3.1. Ácidos nucleicos y nucleótidos

Los ácidos nucleicos son un tipo de macromoléculas presentes en todas las células y virus. Las funciones de los ácidos nucleicos son el almacenamiento y la expresión de información genética. El ácido desoxirribonucleico (ADN) codifica la información que la célula necesita para fabricar proteínas, mientras que el ARN participa en la síntesis de proteínas, función para la que puede presentar diversas formas moleculares.

Los genes contienen la información como una secuencia específica de nucleótidos que son encontrados en el ADN de las moléculas. En las moléculas de ADN, solo se encuentran cuatro bases diferentes que son: guanina, adenina, timina y citosina. En el ácido ribonucléico (ARN), la base nitrogenada uracilo ocupa el lugar de la timina; además la estructura de las cadenas es diferente, el ADN forma una doble hélice mientras que el ARN forma solo una cadena. Un nucleótido está formado por una base nitrogenada unida a un grupo fosfato y a una pentosa o molécula de azúcar (desoxirribosa en el caso del ADN y ribosa en el caso del ARN). El elemento que permite diferenciar un nucleótido de otro es la base nitrogenada que contiene. Las bases nitrogenadas se pueden clasificar en dos grandes grupos, purinas (formados por adenina y guanina) y pirimidinas (formados por citosina, timina y uracilo), y al representar la estructura de cada una de ellas se pueden encontrar diferencias que permiten distinguir fácilmente entre un nucleótido y otro. [1]

3.2. Proteínas

Las proteínas son una clase importante de moléculas que se encuentran en todas las células vivas. Una proteína se compone de una o más cadenas largas de aminoácidos cuya secuencia corresponde a la secuencia de ADN del gen que la codifica. Las proteínas desempeñan gran variedad de funciones en la célula, incluidas estructurales (citoesqueleto), mecánicas (músculo), bioquímicas (enzimas), y de señalización celular (hormonas). Las proteínas son también parte esencial de la dieta. Las proteínas desempeñan un papel fundamental para la vida y son las biomoléculas más versátiles y diversas, imprescindibles en el crecimiento del organismo. Las proteínas pueden ser estudiadas de acuerdo a las estructuras en su nivel de organización. La estructura primaria es la secuencia o cadena de aminoácidos que describe la proteína, la estructura secundaria se produce cuando la secuencia de aminoácidos están unidas por enlaces de hidrógeno formando hélices alfa y hojas plegadas beta, la estructura terciaria que se produce cuando ciertas atracciones están presentes entre las hélices alfa y las hojas plegadas beta, y finalmente la estructura cuaternaria cuando la proteína está conformada por más de una cadena de aminoácidos.

Los aminoácidos son los bloques de construcción de las proteínas, las cuales se forman a partir de estructuras que contienen aminoácidos que poseen variedad en el tamaño, forma y propiedades químicas. Cada aminoácido está compuesto esencialmente por un grupo amino (-NH2), un carbono alfa, un grupo carboxilo (-COOH) y una cadena (habitualmente denominada cadena lateral o radical R) de estructura variable que determina la identidad y las propiedades de cada uno de los diferentes aminoácidos. Existen cientos de radicales que forman cientos de aminoácidos, pero solo 20 aminoácidos forman parte de las proteínas y tienen codones (que es una secuencia de tres nucleótidos de ADN o ARN que corresponden con un aminoácido especifico o una señal de parada dentro del código genético, a partir de este se da origen a los 20 aminoácidos) específicos en el código genético.

Los aminoácidos están agrupados en tres categorías: Los aminoácidos hidrofóbicos, los cuales tienen cadenas laterales compuestas principal o enteramente de carbono e hidrógeno, tienen pocas probabilidades de formar puentes de hidrógeno con las moléculas de agua, los aminoácidos polares, los cuales contienen oxígeno y/o nitrógeno en sus cadenas laterales y fácilmente pueden formar puentes de hidrógeno con el agua, y los aminoácidos cargados que llevan una carga positiva o negativa en el pH biológico.[1]

3.3. Matrices de Sustitución

Una matriz de sustitución o puntuación describe el ritmo en el que un carácter cambia a otro con el tiempo dentro de una secuencia. Usualmente, las matrices de sustitución se ven en el contexto de alineamiento de secuencias de aminoácidos o ADN, en el cual la similitud entre las secuencias depende del tiempo de divergencia y de los ritmos de sustitución según se presenta en la matriz. Este tipo de matrices son más usuales en los alineamientos de secuencias de aminoácidos (proteínas) que en los de nucleótidos ya que para secuencias de nucleótidos las matrices utilizadas son relativamente simples.

Las matrices de sustitución más utilizadas son las matrices PAM y las BLOSUM. Ambas son matrices de *log-odd ratios* que se diferencian principalmente en los fundamentos para el cálculo inicial de las probabilidades de sustitución entre aminoácidos. [1]

3.3.1. PAM

Las primeras matrices de alineamientos de proteínas fueron las matrices PAM, las cuales recogen alineamientos de 71 grupos de secuencias de proteínas estrechamente relacionadas. Las siglas PAM provienen de su nombre en inglés *point accepted mutation* (mutación puntual aceptada). El valor de una determinada celda representa la probabilidad de la sustitución de un aminoácido por otro, conocida como mutación puntual. Puesto que la matriz se calcula observando diferencias entre proteínas muy cercanas evolutivamente (con, al menos, un 85 % de similitud), las sustituciones en cuestión no tienen efecto sobre la función de la proteína, por lo que se trata de mutaciones aceptadas en el proceso evolutivo.

Estas secuencias de proteínas fueron agrupadas basadas en reconstrucción filogenética usando *maximum parsimony* (máxima parsimonia), que es un método que nos permite crear un árbol filogenético minimizando el total de pasos evolutivos requeridos para la creación del árbol con la información dada. Las matrices PAM fueron después derivadas basadas en la divergencia evolutiva entre las secuencias del mismo grupo. Una unidad (PAM1) nos indica que la frecuencia de mutación corresponde solo al 1 % de los aminoácidos dentro de la secuencia. Para construir una matriz de sustitución PAM1, un grupo de secuencias estrechamente relacionadas con frecuencias de mutación iguales a una unidad PAM son escogidas y utilizadas para realizar el calculo de la matriz.

Una matriz de sustitución puede ser derivada a partir de esta matriz (PAM1) elevándola a una determinada potencia por lo que la matriz (PAM250) resultaría elevando a la 250 potencia la matriz (PAM1) obteniendo así una aproximación con los mismos datos que se generó la matriz PAM1 pero que nos permite representar secuencias mas divergentes debido a que la unidad PAM cambia al realizar el cálculo de la potencia $PAM_{250} = (PAM_1)^{250}$ (ver cap.5 para más detalles).

3.3.2. BLOSUM

La matriz BLOSUM (matriz de sustitución de bloques de aminoácidos) es una matriz de sustitución utilizada para el alineamiento de secuencias de proteínas. Se usa para puntuar alineamientos entre secuencias de proteínas evolutivamente divergentes. Fue introducida en 1992 en un artículo de S. Henikoff y J. G. Henikoff, quienes recorrieron la base de datos BLOCKS analizando regiones muy conservadas de familias de proteínas (sin *gaps* en el alineamiento de secuencias) y comprobaron las frecuencias relativas de aparición de los aminoácidos y las probabilidades de sustitución entre ellos, calcularon una puntuación de log-probabilidad para cada una de las 210 posibles sustituciones de los 20 aminoácidos estándar.

Una particularidad de las matrices BLOSUM es que se basan en alineamientos observados y no son extrapoladas a partir de comparaciones de proteínas cercanamente relacionadas. Existen conjuntos de matrices BLOSUM que utilizan diferentes bases de datos de alineamientos y se nombran con números. Las BLOSUM seguidas de un número alto están diseñadas para comparar secuencias cercanamente relacionadas, mientras que las BLOSUM con número bajo están diseñadas para comparar secuencias relacionadas de forma distante.

A cada posible identidad o sustitución se le asigna una puntuación basada en las frecuencias observadas en el alineamiento de proteínas relacionadas. Se da una puntuación positiva a las sustituciones más probables, mientras que corresponde una puntuación negativa para sustituciones menos probables. El número que acompaña a la matriz BLOSUM indica el porcentaje de similitud que existe entre las secuencias que se tomaron como base para la construcción de la matriz, por tal motivo entre mayor sea el número, más relación hay entre las secuencias que fueron consideradas para la construcción de la matriz. BLOSUM 62 es la matriz calculada usando las sustituciones observadas entre proteínas que tienen, como mínimo, el 62 % de identidad en la secuencia, y se ha convertido en el estándar de la mayoría de los programas que utilizan este tipo de matrices (ver cap.5 para más detalles). [2]

3.4. Alineamiento de Secuencias

La comparación de secuencias se encuentra en el corazón del análisis bioinformático siendo un primer paso importante hacia el análisis estructural y funcional de secuencias recién determinadas. Debido a la generación exponencial de nuevas secuencias biológicas, la comparación de secuencias se ha convertido en un proceso esencial para extraer inferencias funcionales y evolutivas de una nueva proteína a partir de bases de datos de proteínas existentes. El principal proceso en este tipo de comparación es el alineamiento de secuencias, en el cual las secuencias se comparan mediante la búsqueda de patrones de caracteres comunes y se establecen 1-1 correspondencias entre secuencias relacionadas.

Los principales usos relacionados con algoritmos de alineamiento de secuencias están relacionados con la búsqueda de residuos comunes entre dos secuencias, la búsqueda de secuencias similares dentro de una base de datos, y la identificación de características comunes basadas en los residuos dentro de un grupo de secuencias (alineamiento múltiple). Su objetivo general es encontrar el mejor emparejamiento, tal que haya máxima correspondencia entre residuos. Para ello, una secuencia se desplaza respecto a la otra para encontrar la posición donde se encuentran las coincidencias máximas. El alineamiento de pares puede ser local, en donde se encuentran pequeños segmentos pertenecientes a las secuencias que son muy similares, o global, en donde se tiene en cuenta la totalidad de las secuencias para el establecimiento de similitudes.

El alineamiento global puede presentar problemas si se aplica a secuencias divergentes o de longitudes variables ya que falla en el reconocimiento de regiones locales muy similares entre las dos secuencias,

razón por la cual no se obtienen resultados óptimos. La alineación local, por otra parte, no asume que las dos secuencias en cuestión tienen similitud en toda la longitud. Esto sólo encuentra regiones con el más alto nivel de similitud entre las dos secuencias y alinea estas regiones sin considerar el resto de las regiones de la secuencia. Este enfoque se puede utilizar para la alineación de secuencias más divergentes con el objetivo de buscar patrones de ADN conservados o secuencias de proteínas. Las dos secuencias a ser alineadas pueden ser de diferentes longitudes. Los módulos similares resultantes del alineamiento local se conocen como dominios o motivos.

4. ESTADO DEL ARTE

Las matrices de sustitución son muy usadas actualmente en alineamiento de secuencias de nucleótidos o proteínas, y es así como muchos estudios e investigaciones se centran principalmente en la construcción de una matriz de sustitución para un grupo de datos específicos, con lo que los resultados son aplicables y funcionan bien bajo ese rango de valores. Por tal motivo existen distintas matrices de sustitución que son generadas a partir de diferentes alfabetos y describen secuencias biológicas de manera muy específica. A continuación mostramos una breve descripción de los trabajos que más se asemejan al trabajo que se realizará y que se basan en la construcción de matrices de sustitución usando metodologías específicas de acuerdo a cada uno de los casos de estudio.

4.1. Amino acid similarity matrices based on force fields

En este trabajo proponen un método general para derivar matrices de sustitución de aminoácidos a partir de campos de fuerza de baja resolución. A diferencia de los métodos populares para la obtención de matrices de sustitución, el método planteado no utiliza argumentos evolutivos, alineamiento de secuencias o estructuras. La estrategia utilizada fue mutar los residuos y recolectar la contribución al total de la puntuación/energía total, de manera que el promedio de los valores de la puntuación/energía total para cada posición dentro de un conjunto de proteínas resulta en una matriz de sustitución.

Los autores muestran cómo se puede obtener matrices de sustitución basadas en campos de fuerza que actúan sobre estructuras de tres dimensiones. La matriz obtenida permitirá realizar comparaciones entre diferentes campos de fuerza sin importar la diferencia que pueda existir entre los diferentes campos. Estas matrices, debido a la manera en que son calculadas, no tienen una relación con un proceso evolutivo o un proceso de alineamiento de secuencias que permita generar la relación entre cada uno de los aminoácidos; ya que están fundamentadas en propiedades de los aminoácidos como es el caso de los campos de fuerza.

Como parte de los resultados varios ejemplos de matrices de sustitución asimétricas han sido calculados a partir de campos de fuerza, usando diferentes acercamientos, y el rendimiento obtenido para cada una de las matrices fue comparado con respecto a las matrices de sustitución convencionales. [3]

4.2. Fold-specific substitution matrices for protein classification

En este trabajo se describe un método para organizar proteínas en una familia que permita reducir un orden de magnitud en el número de parámetros utilizados para la construcción de una matriz de sustitución. La base usada como medida de similitud en las matrices de sustitución convencionales es el *log-odd ratio*. Esta característica se adaptó para poder crear un atributo de diferenciación que

conllevó a la definición de las matrices de sustitución CLASSUM(*Class Attribute Substitution Matrices*) con características similares a la matriz de sustitución BLOSUM. Con lo que se pueden clasificar proteínas dentro de una familia usando menos parámetros que los que serán utilizados por métodos de clasificación como redes neuronales, modelos ocultos de Markov o matrices de puntuación de posición específica.

El método utilizado en este trabajo fue aplicado para clasificar secuencias jerárquicamente en subgrupos *lambda* y *kappa* de la súperfamilia de las inmunoglobulinas. Las posiciones que confieren una clase fueron identificadas basadas en el grado de variabilidad de un aminoácido en una posición. Donde las matrices CLASSUM tienen mejores resultados en la clasificación comparados a los resultados de la matriz de sustitución BLOSUM 62.

Los resultados obtenidos sugieren que matrices de sustitución derivadas de información específica de una familia pueden mejorar la resolución de métodos automáticos que usan matrices de sustitución para búsqueda y clasificación de proteínas. [4]

4.3. The construction of amino acid substitution matrices for the comparison of proteins with non-standard compositions

Las matrices de sustitución de aminoácidos juegan papel importante en los métodos de alineamientos de proteínas. Las matrices estándar de *log-odd* probabilidades como la PAM y BLOSUM son construidas a partir de largos conjuntos de alineamientos, teniendo implícitamente antecedentes de las frecuencias de los aminoácidos.

Estas matrices de sustitución han sido usadas para comparar proteínas con diferencias muy marcadas en la composición de los aminoácidos como lo son las proteínas de transmembrana o proteínas de organismos con composiciones de nucleótidos fuertemente sesgadas. Las matrices convencionales no son ideales para este tipo de comparaciones, por lo que se construyó una matriz que puede ser utilizada en un ambiente donde no hay una composición estándar, obteniendo mejores resultados mediante el ajuste matemático de las matrices ya existentes.

El ajuste composicional de las matrices de sustitución de aminoácidos en general tiende a incrementar la significación estadística de los alineamientos de proteínas con composiciones no estándar y frecuentemente mejora la exactitud de estos alineamientos también.[5]

4.4. A Protein Structural Alphabet and Its Substitution Matrix CLESUM

Los autores plantean una nueva matriz de sustitución al usar un modelo mixto para la distribución de 3 ángulos pseudoenlazados formados por átomos de carbono alfa de 4 residuos consecutivos, los estados de la estructura local son discretizados como 17 letras conformadas de un alfabeto estructural de proteínas. Se construye una matriz de sustitución usando estas letras y basados en los alineamientos estructurales de la base de datos FSSP, que es una base de datos de estructuras alineadas mediante un proceso exhaustivo de todos contra todos, comparando la estructura 3D de las proteínas en el PDB(*Protein Data Bank*).

La mayoría de métodos de predicción de estructura local usan tres estados de la estructura secundaria: hélices, hojas y giros. Sin embargo una estructura secundaria podría variar significativamente en sus estructuras 3D. Al restringir las conformaciones de los residuos locales a estados manejables se puede

discretizar la conformación de una proteína para convertir la estructura 3D de la cadena principal a una secuencia 1D con los estados discretos que describen la estructura de la proteína parecidos a los aminoácidos.

La popular matriz BLOSUM se deriva de un gran conjunto de patrones de aminoacidos conservados sin gaps que representan varias familias de proteinas. Las frecuencias de sustituciones de los aminoacidos es contada en los alineamientos de secuencias. Estas frecuencias son divididas por la frecuencia esperada de encontrar los aminoacidos juntos en un alineamiento por coincidencia. La proporción de conteos es la probabilidad de que se presente la sustitucion de un aminoacido por otro, es decir , un *odds score*. Las entradas de las matrices BLOSUM son logartimos de los *odds score* en base 2 y multiplicados por un factor de escala de 2.

La matriz de sustitución construida se derivó de la misma manera que la matriz BLOSUM cambiando cierta informacion con la intención de mostrar más detalles por ejemplo el factor de escala utilizado fue de 20 en lugar de 2 que es el factor utilizado para mostrar los resultados finales en la matriz; la matriz construida y utilizada para comparar secuencias teniendo en cuenta el alfabeto estructural propuesto se denomino CLESUM. [6]

4.5. A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence

En el reconocimiento de pliegue de proteínas, una secuencia de aminoácidos de prueba es comparado con una biblioteca de pliegues representativos de estructura conocida para identificar un homólogo estructural. En caso donde la secuencia de aminoácidos de prueba y su homólogo tengan una clara similitud como secuencias, las matrices de sustitución tradicionales han sido usadas para predecir la similitud estructural.

En este trabajo se presenta el caso en que la secuencia de prueba es secuencialmente distante de su homólogo. A partir de una base de datos de 119 pares estructurales en donde cada uno de los pares comparten un pliegue similar pero tienen una identidad de secuencia menor al 30 %. De acuerdo a la informacion de los pares estructurales y de los pliegues homologos se ha desarrollado una matriz de sustitución de 5 dimensiones (7 x 3 x 2 x 7 x 3) que nace a partir de la combinacion de 7 clases de residuos, 3 clases de estructura secundaria y en donde cada pliegue homologo se define a su vez por 7 clases de resiudo, 2 clases de estructura secundaria y 2 clases enterradas(*burial*) que lleva una representación 3D a 1D de una proteína llamada H3P2 que cuenta con un total de 882 valores que representan el pliegue e información de la estructura secundaria de los pares estructurales estudiados.

Las 7 clases de residuos utilizados provienen de agrupar los 20 aminoacidos teniendo en cuenta los valores en la matriz de sustitución PAM250 descriptas a continuacion: citosina, triptófano, básico , aromático, hidrofóbico, pequeño y polar. Las 3 clases de estructura secundaria se definieron a partir de la base de datos DSSP descriptas a continuación: helice, hoja y giro. Las 2 clases enterradas(burial) que fueron usadas corresponden a enterrada o expuesta que fueron asignados para cada una de las 7 clases de resiudos.

Para realizar la prueba de la matriz de sustitución obtenida se realizó una validación cruzada para comparar la matriz H3P2 con las matrices: GONNET, PAM250, BLOSUM62 y una matriz de sustitución únicamente de estructura secundaria. Los resultados mostraron que para secuencias distantemente relacionadas la matriz H3P2 detectó más estructuras homologas y con más alta confiabilidad que las otras matrices de sustitución basados en la relación sensibilidad, las secuencias resultantes realmente eran secuencias homologas de la secuencia usada en la prueba, contra especificidad, las secuencias que fueron descartadas o no tenidas en cuenta en rel resultado para una secuencia dada realmente no eran homologas.[7]

4.6. A substitution matrix for structural alphabet based on structural alignment of homologous proteins and its applications

En el desarrollo del trabajo de utilizaron 16 Protein Blocks(PBs) que fueron construidos a partir de de los ángulos diedros formados a partir de 5 residuos consecutivos y que fueron tomados a partir de pentapetidos de estructuras de proteínas conocidas. Para realizar la asignación de un PB a una región pentapetida se toman los valores observados y los valores ideales para los valores de los ángulos y se seleccionaron los valores con el rmsda(root mean square derivation on angular values , la media cuadrática aplicada a ángulos que forman el pentapetido de donde se calculan los PBs) menor con lo que tenemos el valor del PB correcto para un pentapetido. A partir de esto y teniendo como base de datos de filogenia y un conjunto de alineamientos de estructuras de proteínas homologas se creo una matriz de sustitución de 16x16. Cada uno de los PBs representa 8 valores para cada uno de los ángulos en grados celsius con lo que cada elemento de alfabeto es representado por un vector de 8 elementos. Se usaron las letras desde la a hasta la p teniendo así un total de 16 elementos que conforman la matriz de sustitución.

El trabajo muestra como usando una representación 1D de una una estructura 3D combinada con una matriz de sustitución y programación dinámica simple se logro localizar regiones de similitud estructural, resaltar un cambio sutil en las regiones de similitud estructural e identificar regiones donde no hay similitud estructural, también realizaron comparaciones utilizando diferentes algoritmos de alineamiento obteniendo buenos resultados para la identificación de regiones de alta similitud estructural, los autores dan a entender que la parte complicada se encuentra en definir correctamente el alfabeto 1D basado en propiedades estructurales que pueden variar entre distintas estructuras de proteínas.[8]

4.7. Amino acid substitution matrices for protein conformation identification

En este trabajo usando las bases de datos PDB_SELECT, que contiene secuencias de aminoácidos, y DSSP, que contiene estructuras secundarias de proteínas, se creó una base de bloques de conformación de estructura, los cuales representan la relación secuencia-estructura. Los miembros en un bloque son idénticos en conformación y altamente similares en secuencia. Para la creación de cada uno de los bloques se siguen las siguientes reglas: 1) Las secuencias tienen la misma estructura secundaria y 2) Una secuencia dentro del bloque está relacionada al menos con otra secuencia dentro del bloque.

A partir de esta base de datos BLOCKS se derivó una matriz de sustitución de aminoácidos de conformación específica llamada CBSM60. La matriz muestra un mejorado rendimiento en la búsqueda de conformación de segmentos y en la detención de homólogos cuando se comparó con la matriz BLOSUM 62.

Los autores en el proceso de construcción de la base de datos BLOCKS realizaron un proceso iterativo. Para ello utilizaron la matriz de sustitución BLOSUM 62 para la búsqueda de similitudes entre las secuencias para la creación de los bloques y después de obtener la primera base de datos y construir la matriz CBSM60; esta fue utilizada nuevamente para la creación de una nueva versión de la base de datos y con ello una nueva versión de la matriz CBSM60 y el proceso se repitió hasta que una convergencia final fue encontrada. [9]

4.8. Computing Substitution Matrices for Genomic Comparative Analysis

En este trabajo se presenta un novedoso algoritmo para construir una matriz de sustitución de nucleótidos, el método está fundamentado en la teoría de la información. Básicamente, el algoritmo usa

compresión iterativamente(se representan las secuencias como datos que van a ser enviados mediante una canal de comunicación y son comprimidos entre mayor sea la relación entre las dos secuencias más corto es el mensaje que va a ser transmitido) y construye la matriz de sustitución a partir del alineamiento, luego se aplica nuevamente la matriz encontrada en el alineamiento, para hallar uno mejor, hasta alcanzar un punto de convergencia donde el alineamiento no se pueda mejorar.

Hasta donde se sabe este es el primer algoritmo que realiza el cálculo de una matriz de sustitución para secuencias del tamaño de un genoma sin asumir previamente nada o usar datos de alineamientos previos. Los autores incorporan el proceso alineamiento de secuencias usando la matriz de sustitución calculada en un proceso de *expectation maximisation.* El método ha sido aplicado sobre información real con distancias filogenéticas y composición de nucleótidos diferentes que engañarian a métodos estadísticos clásicos. [10]

5. SELECCIÓN DE LA MATRIZ

Dentro de la primera parte de este trabajo se busca realizar un estudio de las matrices de sustitución que son utilizadas actualmente con el fin de encontrar la más adecuada a partir de la cual se pueda desarrollar la metodología que se quiere proponer. Para ello se tomaron como principales referencias la matriz PAM y la matriz BLOSUM. Mostramos las características fundamentales, y de manera muy general cómo son construidas, para finalmente establecer una comparación entre ambas matrices y seleccionar la más adecuada como base para realizar matrices de sustitución generalizadas utilizando alfabetos arbitrarios.

5.1. Revisión y comparación de las matrices de sustitución más utilizadas actualmente y elección del algoritmo a utilizar

A partir de la información encontrada en la literatura y la definición de muchos algoritmos de alineamiento que se utilizan actualmente se realizó una revisión de cada una de las matrices que es considerada como útil y viable para su utilización hoy en día y que permita caracterizar de manera correcta la probabilidad de que un residuo cambie a otro, ya sea basado en información evolutiva como es el caso de PAM, o de alineamientos previos de muchas secuencias de proteínas pertenecientes a una base de datos como es el caso de BLOSUM. Se evalúan las características principales y se analiza que opción es más conveniente para la construcción de la metodología que busca generalizar la manera en que se construyen las matrices de sustitución para que puedan abarcar diferentes alfabetos y no meramente aminoácidos y nucleótidos.

5.2. Comparación de las matrices utilizadas actualmente

Las matrices de sustitución juegan un papel muy importante en la bioinformática ya que al momento de comparar secuencias se debe tener en cuenta la influencia que puede tener la evolución y cómo se pueden presentar diferentes tipos de cambios durante el proceso. Para las proteínas, en el proceso de sustitución se considera más aceptable el reemplazo de aminoácidos similares que aquellos que son divergentes con respecto a sus características fisicoquímicas .

Existen varias maneras de construir matrices de sustitución para proteínas. Una de ellas es basada en las propiedades de cada aminoácido, como lo son el residuo hidrofóbico, la carga de electronegatividad,

y el tamaño, viendo así la importancia de conocer las propiedades fisicoquímicas de los aminoácidos. Otra manera se basa en el código genético, con lo que se busca el número necesario de sustituciones para pasar de un codón a otro, teniendo en cuenta las dos secuencias alineadas, la matriz de sustitución es construida mediante la observación de la proporción de sustitución real entre varios aminoácidos en la naturaleza, por lo que si una sustitución entre dos aminoácidos es observada frecuentemente, esta será puntuada positivamente. Esta es una idea muy sencilla, sin embargo se basa en algo que ocurre en la naturaleza.

Por esta razón la segunda manera es la más utilizada, dado que es más intuitiva y se ajusta mejor a lo que realmente puede estar ocurriendo entre diferentes secuencias de proteínas. Lo que se desea conocer es si dos secuencias son homólogas, es decir, si están evolutivamente relacionadas o no lo están; por lo que se busca una puntuación para el alineamiento que refleje la razón con la que un residuo puede cambiar a otro teniendo en cuenta alineamientos realizados anteriormente. Por este motivo y basados en alineamientos sin *gaps* se han desarrollado dos matrices muy utilizadas para puntuar alineamientos de secuencias en bioinformática, la matriz PAM y la matriz BLOSUM. Las matrices buscan puntuar de manera diferente cada una de las sustituciones que se puede dar de un aminoácido a otro teniendo en cuenta un análisis inicial sobre una gran cantidad de secuencias que estaban de cierto modo relacionadas y que sirven como base para construir estas matrices.

Para la construcción de una matriz de sustitución para proteínas necesitamos alrededor de 400 valores, teniendo en cuenta los pares de 20 aminoácidos. Durante la construcción de la matriz sólo estamos teniendo en cuenta que ocurra el reemplazo de un aminoácido por otro, es decir, la dirección de la sustitución no influye, obteniendo que A —> B es igual a B —> A. El resultado será una matriz simétrica que nos facilita el total de operaciones que se realizan para la construcción de una matriz de sustitución donde un total de 210 valores es suficiente para la construcción de la matriz de sustitución.

A continuación se muestra la idea básica detrás de la construcción de una matriz de sustitución para proteínas basada en el ritmo de sustituciones observadas:

Considera el más simple de los alineamientos, es decir, un alineamiento global sin *gaps* de dos secuencias X y Y, de longitud n.

M probabilidad de que los residuos alineados tengan un ancestro en común

R probabilidad de que los residuos fueron alineados por azar

En la puntuación de este alineamiento se evalúa :

$$\frac{Pr(X, Y|M)}{Pr(X, Y|R)}$$

El numerador indica la probabilidad de que haya una sustitución dentro de la secuencia X a la secuencia Y dado que las secuencias tienen un ancestro en común (M). El denominador indica la probabilidad de que haya una sustitución de X a Y dado que las secuencias son alineadas por azar (R).

La matriz de sustitución que puntuará un alineamiento mediante la estimación de esta proporción para cada uno de los pares de aminoácidos que se encuentran en las dos secuencias alineadas.

Sea q_a la frecuencia del aminoácido a.

Sea q_{xi} y q_{yi} los residuos en una posición específica de ambas secuencias X y Y.

Considera el caso donde el alineamiento de X y Y es aleatorio:

$$Pr(X, Y|R) = \prod_i^n q_{xi} \prod_i^n q_{yi}$$

Sea P_{ab} la probabilidad de que a y b sean derivados de un ancestro común.

El caso donde el alineamiento es debido a un ancestro común es:

$$Pr(X, Y|M) = \prod_i^n P_{xiyi}$$

Esto como resultado de las observaciones relacionadas.

El log-odds ratio de estas alternativas está dada por:

$$\frac{Pr(X, Y|M)}{Pr(X, Y|R)} = \frac{\prod_i^n P_{xiyi}}{\prod_i^n q_{xi} \prod_i^n q_{yi}} = \frac{\prod_i^n P_{xiyi}}{\prod_i^n q_{xi} q_{yi}}$$

Como vemos tenemos una multiplicatoria donde aplicando log probabilidades:

Primero tenemos que

$$\frac{Pr(X, Y|M)}{Pr(X, Y|R)} = \frac{\log(\prod_i^n P_{xiyi})}{\log(\prod_i^n q_{xi} \prod_i^n q_{yi})} = \frac{\log(\sum_i^n P_{xiyi})}{\log(\sum_i^n q_{xi} q_{yi})}$$

Finalmente tenemos que:

$$log \frac{Pr(X, Y|M)}{Pr(X, Y|R)} = log \sum_i^n \frac{P_{xiyi}}{q_{xi} q_{yi}}$$

Aplicando *log* probabilidades podemos realizar un mejor análisis ya que estamos sumando sobre las razones de cada una de las posibles sustituciones y no sobre las multiplicaciones, lo que nos podría generar un mayor error debido a que al realizar multiplicaciones de estos valores puede haber una perdida de precisión y desborde en las cantidades al finalizar los cálculos.

5.3. BLOSUM

Se basa en alineamientos de secuencias de proteínas evolutivamente divergentes. Para ello se tomó la base de datos BLOCKS, que es una base de datos de bloques de secuencias correspondientes a familias conservadas de proteínas que están alineadas y caracterizadas por unos atributos específicos (ver cap. 6 para mas información) y se analizaron regiones muy conservadas de familias de proteínas, luego se comprobaron las frecuencias relativas de la aparición de aminoácidos y la probabilidad de sustitución entre ellos. Las matrices BLOSUM se describen de la siguiente manera: BLOSUM seguidas de un número, por ejemplo BLOSUM62. El número que acompaña el nombre de la matriz indica el porcentaje de identidad entre las secuencias de proteínas que conforman un bloque dentro de BLOCKS que

son utilizadas para la construcción de la matriz. Por lo tanto BLOSUM62 une a todas las secuencias proteínas dentro de un bloque cuyo porcentaje de similitud es $>= 62\,\%$. El número hace referencia al mínimo porcentaje de similitud entre las secuencias utilizadas dentro de los bloques usados para construir la matriz de sustitución. Un número alto indica que usando esta matriz se comparan secuencias estrechamente relacionadas, mientras que un número bajo indica que usando esta matriz se comparan secuencias relacionadas de manera distante.

Las puntuaciones dentro de la matriz BLOSUM corresponden a *log odd* probabilidades. Cada posición de la matriz está representada de la siguiente manera, en términos generales:

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

P_{ij} es la probabilidad de que dos aminoácidos i y j reemplacen uno a otro en una secuencia homóloga.

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 de proporcionalidad para obtener puntuaciones que difieren después del redondeo para pares distintos de aminos.

En la matriz puntuaciones positivas indican sustituciones conservadas y las puntuaciones negativas indican sustituciones no conservadas.

A continuación se muestra de manera general como se realiza el cálculo de la matriz BLOSUM:

- Agrupar varias secuencias en un cluster siempre que el porcentaje de similitud sea mayor o igual que el valor L % definido para la matriz BLOSUML, es decir, se cumpla con esta similitud a lo largo de todas las secuencias utilizadas para la construcción de la matriz de sustitución.
- Contar el número de sustituciones dentro de un mismo cluster.
- Estimar la frecuencia basado en los conteos.

5.4. PAM

Esta matriz es derivada de alineamientos globales de secuencias cercanamente relacionadas, es motivada por la evolución. En si la matriz PAM es una matriz usada para buscar relaciones entre secuencias divergentes que fue construida a partir de secuencias estrechamente relacionadas.

El modelo de evolución de las matrices PAM tiene las siguientes características:

- Cada posición cambia independientemente del resto.
- La probabilidad de mutación es la misma en cada posición.
- La evolución no recuerda (sin memoria).

Ahora definiremos la medida de distancia PAM : Sean S_1 y S_2 dos secuencias de proteínas con $|S_1| = |S_2|$. Decimos que S_1 y S_2 están a una distancia x PAM, si S_1, muy probablemente fue producida a partir de S_2 con x mutaciones por 100 aminoácidos. Hay que diferenciar la unidad de medida PAM con el porcentaje de identidad de una secuencia. Por ejemplo la matriz PAM250 tiene en cuenta secuencias con el 45 % de identidad, debido a que la unidad PAM mide la cantidad de residuos que fueron cambiados para llegar de una secuencia a otra es por esta razón que una unidad PAM pequeña nos muestra una estrecha relación entre las secuencias comparadas ya que el porcentaje de aminoácidos cambiados para llegar de un aminoácido a otro es muy pequeño. Pares de secuencias con una medida PAM > 250 probablemente no son homólogos ya que para estos pares el porcentaje de identidad de secuencia es menor al 45 %.

El cálculo de las matrices PAM se basó en árboles filogenéticos que se construyeron a partir de 71 familias de proteínas estrechamente relacionadas, las proteínas a ser estudiadas fueron seleccionadas sobre la base que tuvieran alta similitud con sus predecesores.

Las puntuaciones dentro de la matriz PAM corresponden a log probabilidades. El valor de una determinada celda representa la probabilidad de la sustitución de un aminoácido por otro, a esto lo denominamos mutación puntual, así, la matriz se calcula observando diferencias en las proteínas muy cercanas evolutivamente, por lo que las sustituciones en cuestión no tienen efecto sobre la función de la proteína. Cada posición de la matriz está representada de la siguiente manera en términos generales:

$$M_x\left(i,j\right) = \log\left(\frac{f_{ij}}{f_i f_j}\right)$$

f_{ij}es la probabilidad de que dos aminoácidos i y j reemplacen uno a otro en una secuencia homóloga. Proporción de mutación.

f_i y f_j son las frecuencias relativas de los aminoácidos i y j en todas las secuencias de proteinas tenidas en cuenta para la construcción de la matriz.

Las frecuencias de las mutaciones de PAMX dependen linealmente de las frecuencias de las mutaciones de PAM1 de modo que PAMX es calculada al repetir la multiplicación de matrices de PAM1 por ella misma.

$$M_x = (M_1)^x$$

Debido a que en nuestra generalización buscamos tener una visión general, es oportuno realizar una profundización sobre su implementación, principalmente en los datos que son usados en el inicio para construir los clusters con el determinado porcentaje de identidad.

5.5. Elección de la matriz a utilizar y el algoritmo

5.5.1. Comparación entre BLOSUM y PAM

Debido a que en este trabajo deseamos mostrar una metodología que permita la construcción de matrices de sustitución generalizadas y evaluando entre las matrices PAM y BLOSUM se puede comprobar que resulta más coherente la utilización de las matrices BLOSUM, ya que la idea es generalizar y poder llevar acabo estudios con diferentes grupos de proteínas, donde los resultados estarán estrechamente relacionados con las fuentes utilizadas para la construcción de las matrices, basándose en alineamientos de secuencias de proteínas existentes

PAM	BLOSUM
Calculada a partir de alineamientos globales	Cálculo a partir de alineamientos locales
Secuencias de proteínas usadas en alineamiento tienen $> 99\,\%$ de identidad	Puede seleccionar un nivel de similitud entre las secuencias usadas en el análisis
La matriz más utilizada es la PAM250	La matriz más usada es BLOSUM62
Las matrices son extrapolaciones matemáticas de la matriz PAM1	Cada matriz es el resultado del análisis de un alineamiento de bloques conservados (análisis real)
Es posible elaborar un modelo evolutivo y así generar nuevas matrices a partir de la primera	No permite generar un modelo evolutivo
	Permite detectar las mejores secuencias con relación biológica ya que se basa en alineamiento de secuencias existentes y no en valores extrapolados.

Tabla 1: Comparación entre la matriz BLOSUM y PAM

BLOSUM80	BLOSUM62	BLOSUM45
PAM1	PAM120	PAM250
Menos divergente	$<$ ——————— $>$	Más divergente

Tabla 2: Relación de similitud entre la matriz PAM y la matriz BLOSUM

6. GENERALIZACIÓN DEL ALGORITMO (Usando una base de datos BLOCKS generalizada)

6.1. Base de datos BLOCKS

La base de datos BLOCKS fue construida a partir de las bases de datos Prosite y Swiss-Prot. El proceso se realiza en dos pasos, primero se corre un algoritmo de *motif* para cada uno de los grupos de proteínas, formando unos bloques parciales, luego estos bloques son ajustados dando lugar a los mejores bloques que permiten representar las características conservadas de las diferentes secuencias de proteínas. Estos dos pasos son repetidos para cada grupo de proteínas y los resultados son concatenados para dar lugar la base de datos BLOCKS.[11]

La base de datos BLOCKS, que actualmente se encuentra disponible en el servidor FTP del NCBI, y que corresponde a la ultima versión desarrollada (la 14.3 construida en el 2007) con un total de 29068 bloques que representan 5900 grupos de proteínas documentados en InterPro 14.0 dirigidas a SWISS-PROT 51.3 y TrEMBL 34.3 obtenidas del servidor InterPro.

6.1.1. Creación de la base de datos para alfabetos específicos

Algoritmo 1 Traducción de la base de datos BLOCKS

```
Require: BLOCKS /* Base de datos BLOCKS*/
Ensure: BLOCKSTraducido
  BLOCKSTraducido = BLOCKS
  while BLOCKSTraducido has lines do
    linea ← BLOCKSTraducido.lineaActual
    if contieneSecuencia(linea) then
      BLOCKSTraducido.lineaActual ← traducir(linea)
    else if contieneWidth(linea) then
      BLOCKSTraducido.lineaActual ←actualizarWidth(linea)
    end if
  end while
```

Algoritmo 2 Traducir para tuplas

```
Require: linea /* Cadena que representa una entrada */
Ensure: lineaTraducida
  partes ← linea.split(" ")
  secuencia ← partes[3]
  for i from 0 to lenght(secuencia)/2 do
    lineaTraducida ← lineaTraducida +secuencia[i*2] + secuencia[i*2 +1]
  end for
  return lineaTraducida
```

Algoritmo 3 ActualizarWidth para tuplas

```
Require: linea /* Cadena que representa una entrada */
Ensure: lineaActualizada
  partes ← linea.split(" ")
  secuencia ← partes[2]
  for i from 0 to lenght(secuencia) do
    if i < 6 then
      lineaActualizada ← lineaActualizada +secuencia[i]
    else
      valor ← valor +secuencia[i]
    end if
  end for
  valorNumerico ←valor.toNum()
  lineaActualizada ← lineaActualizada +valorNumerico/2
  return lineaActualizada
```

6.2. BLOSUM y su algoritmo

6.2.1. Descripción del algoritmo

Algoritmo 4 BLOSUM parte 1

```
Require: AAS /* Tamaño del alfabeto */, MAXSEQS /* Cantidad maxima de secuencias dentro
  de un bloque*/, MAX_MERGE_WIDTH /* Ancho maximo de un bloque*/, BLOCKS /* Base
  de datos BLOCKS*/, MinimunStrenght /* Minima fuerza de un bloque*/, MaximunStrenght /*
  Maxima fuerza de un bloque*/, ClusterPercentage /* El porcentaje con el que se agruparan las
  secuencias dentro de un bloque que finalmente seran tenidas en cuentas para el conteo*/, Scale /*
  Escala sobre la cual se reportaran los valores de la matriz resultante*/
Ensure: sij[AAS,AAS]/* Matriz de sustitucion obtenida con el % de clustering pasado como entrada*/
  , tij[AAS,AAS]
  Counts[AAS][AAS]
  sij[AAS][AAS]
  tij[AAS][AAS]
  for i from 0 to AAS do
    FAaPairs[i] ← 0
    for j from 0 to AAS do
      Counts[i][j] ←0
      sij[i][j] ←0
      tij[i][j] ←0
    end for
  end for
  totpairs ← 0
  totdiag ← 0
  totoffd ← 0
  for row from 0 to AAS do
    for col from 0 to row+1 do
      if row == col then
        totdiag += Counts[row][col]
      else
        totoffd += Counts[row][col] + Counts[col][row]
      end if
    end for
  end for
```

Algoritmo 5 BLOSUM parte 2

```
ftotpairs = totdiag + totoffd
for col from 0 to AAS do
  for row from 0 to AAS do
    if col == row then
      FAaPairs[row] += Counts[row][col]
    else
      FAaPairs[row] += (Counts[row][col] + Counts[col][row]) / 2.0
    end if
  end for
end for
for row from 0 to AAS do
  for col from 0 to row +1 do
    if row == col then
      fij = Counts[row][col]
    else
      fij = (Counts[row][col] + Counts[col][row])/ 2.0;
    end if
    fifj = FAaPairs[row] * FAaPairs[col]
    if fifj > 0.000000001 then
      oij = ftotpairs * fij / fifj
    else
      oij = 0.0
    end if
    if oij > 0.000000001 then
      s = log(oij)
    else
      s = -20.0
    end if
    dtemp = 10.0 * s / log(10.0)
    tij[row][col] = round(dtemp)
    s /= log(2.0)
    sij[row][col] = s
  end for
end for
```

Algoritmo 6 read_dat

```
Require: BLOCKS
Ensure: totblk
  totblk ← 0
  line ← “”
  while BLOCKS has more lines do
    line ←BLOCKS.line
    if line contains “AC” then
      values = line.split(“ ”)
      Block.ac = values[1]
    else if line contains “BL” then
      values = line.split(“=”)
      Block.strength ← values[4].toNum()
      if Block.strength >= minStrength && Block.strength <= maxStrength then
        fill_block(BLOCKS)
        cluster_seqs()
        count_cluster()
        totblk ←totblk + 1
      end if
    end if
  end while
  return totblk
```

Algoritmo 7 fill_block

```
Require: BLOCKS, Block
Ensure: Block
  Block.nseq = Block.width = 0
  Block.totdiag = Block.totoffd = Block.wtot = 0
  cluster = ncluster = 0
  line = “”
  done = false
  while !done do
    line = BLOCKS.line
    if len(line) == 1 then
      if ncluster > 0 then
        for n from 0 to Block.nseq do
          if Block.cluster[n] == cluster then
            Block.ncluster[n] = ncluster
          end if
        end for
      end if
      cluster++
      ncluster = 0
    else if len(line) > 1 then
      if line == “\\” then
        done = true
      else if len(line) > 20 then
        values = line.split(“ ”)
        sequence = values[3]
        for i from 0 to len_alphabet(sequence) do
          value = extract_alphabet(sequence,i)
          Block.aa[Block.nseq][i] = alphabet_to_num(value);
        end for
        if values[4] != “” then
          Block.weight[Block.nseq] = values[4].toNum()
          Block.wtot += Block.weight[Block.nseq]
        else
          Block.weight[Block.nseq] = 0;
        end if
        Block.cluster[Block.nseq] = cluster
        ncluster++
        Block.width = i
        Block.nseq++
      end if
    end if
  end while
  if ncluster > 0 then
    for n from 0 to Block.nseq do
      if Block.cluster[n] == cluster then
        Block.ncluster[n] = ncluster
      end if
    end for
  end if
```

Algoritmo 8 cluster_seqs Parte 1

Algoritmo que permite calcular los clusters de acuerdo al valor proporcionado como parametro y se ajustan los clusters para todas las secuencias del bloque

Require: Block /* Estructura block que contiene toda la información relacionada a un bloque */, AAS /* Cantidad de elementos dentro del alfabeto */, npair /* Total de pares que se pueden formar entre todas las secuencias pertenecientes a un bloque Block.nseq * (Block.nseq - 1) / 2 */ , Cluster

Ensure: pairs[npair] /* Arreglo de estructuras pair*/

```
threshold = Cluster * Block.width / 100
for s1 from 0 to Block.nseq - 1 do
  l1 = 0
  for s2 from s1+1 to Block.nseq do
    l2 = 0
    px = INDEX(Block.nseq, s1, s2)
    pairs[px].score = 0
    pairs[px].cluster = -1
    for i from 0 to Block.width do
      i1 = l1 + i
      i2 = l2 + i
      if i1 >= 0 && i1 < Block.width && i2 >= 0 && i2 < Block.width && Block.aa[s1][i1] ==
      Block.aa[s2][i2] then
        pairs[px].score += 1
      end if
      pairs[px].cluster = -1
    end for
  end for
end for
for s1 from 0 to Block.nseq do
  Block.cluster[s1] = -1
  Block.ncluster[s1] = 1
  nclus[s1] = 0
end for
...
```

Algoritmo 9 cluster_seqs Parte 2

```
...
for s1 from 0 to Block.nseq - 1 do
  for s2 from s1+1 to Block.nseq do
    px = INDEX(Block.nseq, s1, s2)
    if pairs[px].score >= threshold then
      if Block.cluster[s1] < 0 then
        if Block.cluster[s2] < 0 then
          Block.cluster[s1] = clus++
          Block.cluster[s2] = Block.cluster[s1]
        else
          Block.cluster[s1] = Block.cluster[s2];
        end if
      else if Block.cluster[s1] >= 0 && Block.cluster[s2] < 0 then
        Block.cluster[s2] = Block.cluster[s1]
      else if Block.cluster[s1] >= 0 && Block.cluster[s2] >= 0 then
        minclus = Block.cluster[s1]
        oldclus = Block.cluster[s2]
        if Block.cluster[s2] < Block.cluster[s1] then
          minclus = Block.cluster[s2]
          oldclus = Block.cluster[s1]
        end if
        for i1 from 0 to Block.nseq do
          if Block.cluster[i1] == oldclus then
            Block.cluster[i1] = minclus
          end if
        end for
      end if
    end if
  end for
end for
for s1 from 0 to Block.nseq do
  if Block.cluster[s1] >= 0 then
    nclus[Block.cluster[s1]]++
  end if
end for
for s1 from 0 to Block.nseq do
  if Block.cluster[s1] < 0 then
    Block.cluster[s1] = clus++
    Block.ncluster[s1] = 1
  else
    Block.ncluster[s1] = nclus[Block.cluster[s1]]
  end if
end for
Block.nclus = clus
```

Algoritmo 10 count_cluster

Algoritmo que permite el conteo de los diferentes clusters que se encuentran dentro de un bloque

```
Require: Block , AAS
Ensure: Counts[AAS][AAS] , Block.totdiag , Block.totoffd
  for col from 0 to Block.width do
    for seq1 from 0 to Block.nseq do
      for seq2 from seq1+1 to Block.nseq do
        if Block.aa[seq1][col] >= 0 && Block.aa[seq1][col] < AAS && Block.aa[seq2][col] >= 0 &&
        Block.aa[seq2][col] < AAS && Block.cluster[seq1] ! = Block.cluster[seq2] then
          weight = 1/Block.ncluster[seq2];
          weight *= 1 / Block.ncluster[seq1];
          Counts[Block.aa[seq1][col]][Block.aa[seq2][col] += weight
          if Block.aa[seq1][col] == Block.aa[seq2][col] then
            Block.totdiag += weight
          else
            Block.totoffd += weight
          end if
        end if
      end for
    end for
  end for
```

El algoritmo inicial sobre el cual se desarrollaron las matrices BLOSUM remonta al año 1991. Esta base se ha mantenido con algunas modificaciones que se han ido realizando durante los años siguientes a su creación, permitiendo corregir problemas en dicho algoritmo pero conservando su base y usando diferentes bases de datos como entradas que corresponden a las distintas versiones de BLOCKS.

La base de datos BLOCKS contiene múltiples alineamientos de regiones conservadas en familias de proteínas. La base de datos BLOCKS está basada en entradas de las siguientes bases de datos InterPro(contiene familia de proteinas, dominios y sitios funcionales) con secuencias de SWISS_PROT(proteinas anotadas manualmente y revisadas) y TrEMBL(proteinas anotadas automaticamente y no revisadas), también tiene referencias a PROSITE(Contiene entradas documentadas que describen dominios de proteinas, familias y sitios funcionales), PRINTS(Consiste en un compendio de *fingerprints* que es un grupo de motivos conservados para caracterizar una familia de proteinas), SMART(Contiene proteinas documentadas en SWISS_PROT y SP_TrEMBL), PFAM(Contiene una colección de familias de proteinas) y ProDOM(Contiene dominios de familias de proteinas) . La base de datos fue construida usando el sistema PROTOMAT, que mediante el uso de un algoritmo de MOTIFS y un proceso interactivo se construyeron todos los bloques que conforman la base de datos que describen cada una de las familias.

Inicialmente se planteaba construir los bloques usando el sistema PROTOMAT, pero debido a la complejidad y al cambio que ha habido en el tiempo en cada una de las bases de datos se ha decidido trabajar con la última versión de BLOCKS que corresponde a la versión 14.3 del 2007.

Para el desarrollo de este trabajo de grado hemos tomado como base un algoritmo que ha sido revisado y que ha corregido algunos problemas que presentaba el algoritmo original puede ser encontrado en la siguiente URL (http://web.mit.edu/bamel/blosum/revised_blosum.c) para correr correctamente el algoritmo solo se ajustaron los valores del ancho del bloque que describe la cantidad máxima de secuencias que puede contener un bloque, irónicamente los resultados que arroja la versión corregida son peores cuando hablamos de búsquedas en bases de datos usando como base las matrices de sustitución arrojadas por el algoritmo. Es por esta razón que basados en los alcances de este proyecto se decide

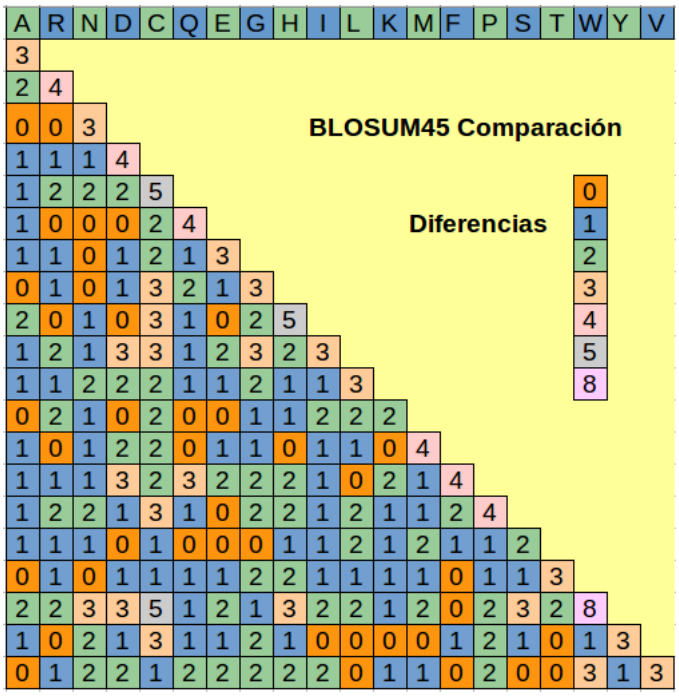

Figura 1: Comparación BLOSUM 45 calculada con la matriz BLOSUM 45 utilizada actualmente

realizar tres pruebas para verificar que el algoritmo revisado, a pesar de tener problemas con las matrices cuando son utilizadas para búsquedas, conserva la misma información que requerimos para el proceso de generalización.

Para realizar estas pruebas tomamos como base de datos BLOCKS 14.3, que es la ultima base de datos generada por el algoritmo PROTOMAT , y el algoritmo BLOSUM revisado. La base de datos BLOCKS utilizada es la misma que se encuentra disponible en la siguiente (`ftp://ftp.ddbj.nig.ac.jp/mirror_database/blocks/unix/current/`) y el algoritmo BLOSUM . Teniendo como base el algoritmo y la base de datos se realizaron pequeños ajustes para que el algoritmo pudiera procesar la base de datos y construimos las matrices de sustitución BLOSUM45, BLOSUM62 y BLOSUM80, y después de obtener estas matrices comparamos los resultados arrojados con las versiones que son manejadas actualmente como estándar de las mismas matrices. Seguido a esto hicimos una comparación viendo qué diferencias había entre la versión antigua y la nueva versión de cada una de las matrices de sustitución.

6.2.2. Validación del algoritmo

Para validar el algoritmo generalizado se aplica al alfabeto de aminoácidos directamente y se comparan las matrices de sustitución obtenidas por este algoritmo con las matrices BLOSUM actuales.

A continuación se presentan tres imágenes correspondientes a la comparación de cada una de las matrices en donde cada valor en la matriz tiene un color asociado para reconocer a simple vista qué tan divergente es la matriz arrojada de su antepasado.

Como podemos ver en cada una de las imágenes, las diferencias no van más allá de 3 unidades en la mayoría de las matrices. Exceptuando la matriz BLOSUM45, que presenta valores más divergentes,

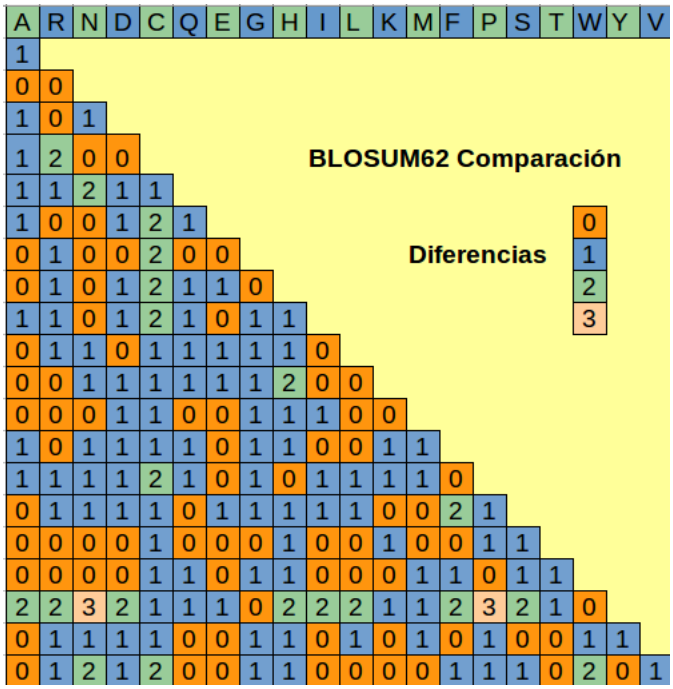

Figura 2: Comparación BLOSUM 45 calculada con la matriz BLOSUM 45 utilizada actualmente

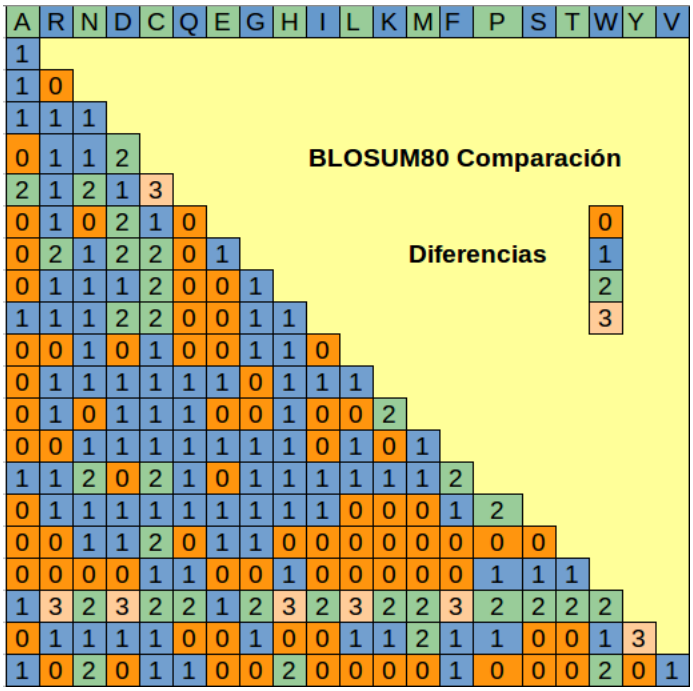

Figura 3: Comparación BLOSUM 45 calculada con la matriz BLOSUM 45 utilizada actualmente

cabe resaltar que cada uno de estos valores están determinados por los parámetros con los cuales se corrió el algoritmo y la base de datos utilizada para las matrices originales (BLOCKS 5.0). Para estas nuevas matrices se usó la base de datos BLOCKS 14.3, que es la que se tendrá de base para realizar todo el desarrollo durante el trabajo de grado. (ver cap. para mas información)

A partir de estos resultados podemos ver que es altamente viable usar el algoritmo revisado como base para las implementaciones de los algoritmos generalizados de las matrices de sustitución.

7. METODOLOGÍA PROPUESTA

A continuación se describe de manera detallada la metodología propuesta en este trabajo de grado y con la cual partiendo de un alfabeto bien definido y una base de datos de proteínas con un proceso definido de traducción se puede lograr obtener una matriz de sustitución tomando como base el algoritmo BLOSUM.

7.1. Definición de la metodología

Para el desarrollo de la metodología se plantearon 5 pasos, en donde cada uno de ellos es importante para que se complete correctamente el proceso de desarrollo de una matriz de sustitución. Los pasos consisten en:

1. Identificar la base de datos BLOCKS a utilizar, que es nuestro punto de partida
2. Ajustar la base de datos BLOCKS pasando de proteínas al alfabeto arbitrario que se va a trabajar
3. Se adapta adecuadamente el algoritmo para que funcione con el alfabeto arbitrario utilizado y la representación numérica de cada uno de los alfabetos que sea fácilmente utilizado para la construcción de la matriz de sustitución.
4. Se ejecuta el algoritmo, cuyo tiempo de ejecución puede variar dependiendo de la capacidad de cómputo.
5. Se obtienen los resultados correspondientes a la matriz de sustitución y que representan el alfabeto que fue inicialmente definido.

7.2. Traducción al alfabeto definido

Durante el desarrollo de la metodología nos encontramos con un problema muy grande y está relacionado con la fuente de los datos del algoritmo escogido y su implementación, dado que es fuertemente independiente de la estructura de bloques que está definida en la base datos, donde se agrupan regiones conservadas de diferentes familias de proteínas. Lo ideal sería contar con la base de datos correspondiente al alfabeto que se va a utilizar debido a la complejidad del problema y como medida de emergencia. Por la dependencia que existe del algoritmo BLOSUM a la base de datos BLOCKS se decidió realizar un proceso de traducción de las secuencias de proteínas a los diferentes alfabetos tomando como supuesto que si la secuencias están relacionadas como secuencias de aminoácidos también lo estarán en un nuevo alfabeto creado a partir de la secuencia de aminoácidos.

Teniendo en cuenta esta medida de emergencia se continuó con el desarrollo de la metodología que se basa en los bloques de proteínas provenientes de la base de datos BLOCKS, pero que finalmente son traducidos y utilizados en un alfabeto arbitrario.

7.3. Nueva base de datos BLOCKS en el alfabeto definido

La nueva base de datos conserva las mismas características de la base de datos BLOCKS solo que dentro de cada bloque específicamente para cada una de las secuencias se hace una traducción de la secuencia que es presentaba en aminoácidos al alfabeto especifico, es decir que tenemos claridad sobre como realizar el ajuste de una secuencia de proteínas al alfabeto que vamos a utilizar para realizar la nueva matriz de sustitución.

A partir del proceso de traducción podemos tener una nueva base de datos BLOCKS que corresponda al alfabeto arbitrario. Esta base puede presentar bloques más pequeños, más grandes, o del mismo tamaño dependiendo de la definición del alfabeto con el que se vaya a trabajar para el desarrollo de la metodología. Como parte de las pruebas se utilizó un alfabeto de tuplas de aminoácidos y otro de tripletas de aminoácidos con los cuales las nuevas base de datos BLOCKS generadas tenían un valor múltiplo de 2 en la longitud de los bloques para las tuplas, y un valor múltiplo de 3 en la longitud de los bloques para las tripletas.

En el caso de manejar un alfabeto reducido o extendido, que puede ser traducido a partir de las secuencias de aminoácidos que tenemos inicialmente y que son los datos que tenemos como base para la construcción de la metodología, la nueva base de datos verá reflejada las características y los residuos correspondientes al alfabeto.

7.4. Configuración del algoritmo BLOSUM para la utilización del nuevo alfabeto definido

Dependiendo de la naturaleza del alfabeto que se esté utilizando para la construcción de la matriz de sustitución es necesario realizar ajustes relacionados con la cantidad de elementos que contendrá la matriz de sustitución, esto lo podemos conocer a partir de la cantidad de residuos que conforman el alfabeto con el que nos encontramos trabajando.

De esta manera, si tenemos un alfabeto que corresponde a los 20 aminoácidos utilizados normalmente en las secuencias de proteínas, debemos informar al algoritmo que este es el tamaño del alfabeto, así como la representación numérica de cada uno de los residuos que conforman el alfabeto. En el caso de los aminoácidos serían valores numéricos de 0 a 19 si tomamos como punto de partida 0. En el caso de tuplas de aminoácidos, donde tenemos un total de 400 combinaciones que se pueden formar a partir de los 20 aminoácidos, tendríamos valores numéricos que van desde 0 a 399. Con esto podemos dimensionar, y dependiendo del alfabeto que utilicemos, obtendremos una matriz de sustitución diferente con dimensiones que pueden ser muy grandes o pequeñas, todo de acuerdo a la cantidad de residuos dentro del alfabeto.

7.5. Ejecución del algoritmo utilizando los parámetros adecuados

Para la ejecución del algoritmo tenemos como entrada básicamente la base de datos BLOCKS de donde se leerá la información, el valor mínimo y máximo para la fuerza del bloque (*strength*) con lo que se filtran bloques que no estén dentro de este rango, un valor para el *clustering* u otros parámetros relacionados con la manera de agrupar las secuencias dento de un mismo bloque, y finalmente un valor para la entropía. Cada uno de estos parámetros es fundamental para la ejecución del algoritmo y nos permite, después de realizar todos los conteos, cálculos y ajustes, tener como salida un conjunto de matrices que nos muestran diferentes valores de una misma idea, es decir, las matrices involucradas en la construcción de la matriz de sustitución para el alfabeto arbitrario que se ha utilizado.

7.6. Resultado obtenido matriz de sustitución en el alfabeto definido

Después de ejecutar el algoritmo con los parámetros seleccionados obtendremos una matriz de sustitución, en la cual veremos una representación de los valores de intercambio que pueden ser encontrados para el alfabeto seleccionado. Con esto tendremos una idea de qué sustituciones son altamente calificadas y cuáles no lo son, además de ser un resultado obtenido y representado en la base de datos utilizada y que se ve mejor representado a partir de la matriz de sustitución.

Esta matriz de sustitución puede ser utilizada en diferentes aplicaciones, en nuestro caso nos basamos únicamente en los valores obtenidos que demuestran una relación entre los elementos del alfabeto que antes no se conocía y una aplicación mediante un algoritmo de alineamiento adaptado para usar la matriz de sustitución, con el cual podemos revisar algunos resultados al alinear alfabetos arbitrarios que no corresponden con alfabetos comunmente utilizados, como es el caso de los aminoácidos y nucleótidos.

8. GENERALIZACIÓN DEL ALGORITMO DE ALINEAMIENTO

8.1. Needleman Wunsch

Es un algoritmo de alineamiento global basado en programación dinámica y fue uno de los primeros utilizados para el alineamiento de secuencias biológicas. La idea esencial detrás del algoritmo es dividir un problema muy grande en pequeños subproblemas que pueden ser resueltos fácilmente y que finalmente dan como resultado una solución óptima al problema grande planteado inicialmente.[12]

Para realizar la puntación de los diferentes residuos, que en su momento pueden estar alineados, el algoritmo usa una función de puntuación basada en los residuos que pertenecen al alfabeto, de esta manera una matriz de sustitución puede ser utilizada fácilmente, cumpliendo esta función y permitiendo obtener resultados más variables ya que es un elemento que podemos variar dependiendo de las pruebas que necesitemos realizar.

Para la realización del alineamiento global de un alfabeto arbitrario tomaremos ambas secuencias donde cada posición corresponde a un residuo del alfabeto y que describe de manera particular la secuencia que se está alineando en el momento. Teniendo esto para ambas secuencias y una representación matricial de su alineamiento mediante programación dinámica se puede obtener un valor óptimo para el alineamiento global de ambas secuencias.

A continuación se muestra la idea básica detrás de la resolución de los subproblemas que resuelven el problema general:

Dadas dos secuencias $X(x_1, ..., x_n)$ y $Y(y_1, ..., y_m)$, para realizar el alineamiento de ambas secuencias se construye una matriz $F(i, j)$ de dimensiones nxm en donde la parte de las columnas corresponde a una de las secuencias y cada columna es un residuo perteneciente a esta secuencia. Igualmente para cada una de las filas, la función de puntuación $s(a, b)$ y el valor d que representa la puntuación para los *gaps* que pueden ser utilizados durante el alineamiento.

$$F(i,j) = max \begin{cases} F(i-1, j-1) + s(xi, yi) & D \\ F(i-1, j) - d & A \\ F(i, j-1) - d & I \end{cases}$$

Para la inicialización de la matriz se tienen en cuenta los siguientes valores:

$F(0,0) = 0$ (1)

$F(i,0) = -d * i$ (2)

$F(0,j) = -d * j$ (3)

Donde $F(i,j)$ es el valor para la matriz en una posición específica i y j. Después de obtener los valores de la matriz se realiza un proceso de escritura del alineamiento utilizando los valores de D, A e I que son almacenados en un matriz auxiliar representando D (Diagonal), A (Arriba) y I (Izquierda).

Basados en esta información podemos entender cómo el proceso de generalización del algoritmo de alineamiento puede ser llevado a cabo para diferentes alfabetos mientras se tenga una clara distinción entre los residuos y que sean representados correctamente en la matriz de sustitución.

Este algoritmo se propone como una manera de probar las matrices de sustitución que fueron generadas mediante la metodología siguiendo los pasos aquí descriptos y como vemos el algoritmo de alineamiento también se puede generalizar teniendo la posibilidad de realizar alineamientos para alfabetos y tripletas si tomamos como base las matrices generadas a partir de este trabajo de grado.

Algoritmo 11 Needleman - Wunsch Parte 1

Require: N /* Tamaño del alfabeto utilizado */ , L1 /* Tamaño de la secuencia 1*/ , L2 /* Tamaño de la secuencia 2 */ , d /* la penalización para gap*/, A[L1+1] /* Secuencia 1*/ , B[L2+1] /* Secuencia 2*/

Ensure: F[L1+1][L2+1] /* Matriz de puntuación */, B[L1+1][L2+1] /* Matriz con el recorrido para poder renconstruirlo al final del algoritmo */ , AlineamientoA, AlineamientoB

```
F(0,0) ← 1
for i from 0 to L1+1 do
  F(i,0) ← -d * i;
end for
for j from 0 to L2+1 do
  F(0,j) ← -d * j;
end for
for i from 1 to L1+1 do
  for j from 1 to L2+1 do
    Acierto ← F(i-1,j-1) + S(T(A(i)),T(B(j)))
    Eliminar ←F(i-1, j) + d
    Insertar ← F(i, j-1) + d
    F(i,j) ←max(Acierto,Eliminar,Insertar)
    if F(i,j) == Acierto then
      B(i,j) == "\\"
    else if F(i,j) == Eliminar then
      B(i,j) == "|"
    else
      B(i,j) == "-"
    end if
  end for
end for
```

Algoritmo 12 Needleman - Wunsch Parte 2

```
AlineamientoA <- ""
AlineamientoB <- ""
i ← length(A) - 1;
j ← length(B) - 1;
while i > 0 and j > 0 do
  if B(i,j) == "\\" then
    AlineamientoA ← A(i) + AlineamientoA
    AlignmentB ← Bj + AlignmentB
    i ← i - 1
    j ← j - 1
  else if if B(i,j) == "|" then
    AlineamientoA ← Ai + AlineamientoA
    AlineamientoB ← "-" + AlineamientoB
    i ← i - 1
  else
    AlineamientoA <- "-" + AlineamientoA
    AlineamientoB <-Bj + AlineamientoB
    j <- j - 1
  end if
end while
while i > 0 do
  AlineamientoA <- A(i) + AlineamientoA
  AlineamientoB <- "-" + AlineamientoB
end while
while j > 0 do
  AlineamientoA <- "-" + AlineamientoA
  AlineamientoB <- Bj + AlineamientoB
end while
return F,B,AlineamientoA, AlineamientoB
```

9. CASOS DE ESTUDIO

Para la descripción de los resultados obtenidos mediante este trabajo se realizaron dos casos de estudio generales. Uno en el cual se muestran las matrices generadas y se evalúan los correspondientes intercambios que se presentan con alta frecuencia. Otro en cual se utiliza el algoritmo de alineamiento escogido para probar de manera general los resultados obtenidos por las matrices de sustitución, se seleccionan diferentes secuencias de proteínas pertenecientes a una misma súper familia pero de diferente familia y secuencias de diferentes súper familias. De esta forma podemos realizar una comparación de diferentes secuencias y podemos evaluar los resultados al realizar los alineamientos.

Para el caso de las matrices de sustitución donde se reviso minuciosamente las substituciones que se presentan en la matriz de sustitución se transformo toda la matriz en una matriz positiva; teniendo estos valores se genera para cada matriz un sistema de puntuación que representa que tan cerca se encuentra un valor cualquiera dentro de la matriz al valor máximo de la matriz con eso si una posición de la matriz obtiene 100 puntos quiere decir que ese valor corresponde exactamente al mismo del máximo. En cada uno de los siguientes casos de estudios se hicieron pruebas experimentales con el sistema de puntos de modo que hay variabilidad y no se ha encontrado un estándar del numero de puntos que permita afirmar que siempre servirá el mismo valor numérico , por defecto estamos usando

80 puntos pero en caso de arrojar muy pocos resultados o demasiados resultados se modifica el umbral con lo que puede haber mayor posibilidad de ver intercambios distintos a los que se dan entre residuos del mismo tipo.

9.1. Caso de estudio 1: Matriz de sustitución a nivel de aminoácidos

9.1.1. Alfabeto Aminoácidos

El alfabeto de aminoácidos consiste de 20 residuos que corresponden a los aminoácidos básicos, los cuales son generados a partir del proceso de traducción del ADN y que están descritos en el código genético. Se generaron tres matrices de sustitución para aminoácidos donde se mantenían los mismos parámetros excepto el nivel de *clustering*. A continuación se muestran las diferentes matrices generadas a partir de la metodología que resultan muy similares a las existentes actualmente con unos pequeños cambios debido a que se usaron bases de datos diferentes.

Se generaron tres matrices correspondientes a los porcentajes de similitud 52, 62 y 80, todas a partir de la misma base de datos BLOCKS, dando origen a las matrices BLOSUM 52, BlOSUM 62 y BLOSUM 80. Cada una de estas matrices es simétrica y contiene un total de 190 valores que representan el total de intercambios que se pueden dar de un aminoácido a otro. Los valores con puntuación mas alta son encontrados en la diagonal y corresponden a la conservación del mismo aminoácido dentro de la base de datos teniendo en cuenta diferentes secuencias.

Dentro de los resultados obtenidos encontramos que para cada una de las matrices se conservan los intercambios entre los aminoácidos, es decir, en la matriz BLOSUM 52 el intercambio mejor puntuado es Triptófano(W) a Triptófano(W). Este comportamiento se mantiene en las otras matrices y se mantiene de la misma manera para los demás aminoácidos a pesar de que el porcentaje de similitud sea diferente.

A continuación mostramos cada una de las matrices generadas con el algoritmo BLOSUM usando como alfabeto los 20 aminoácidos esenciales.

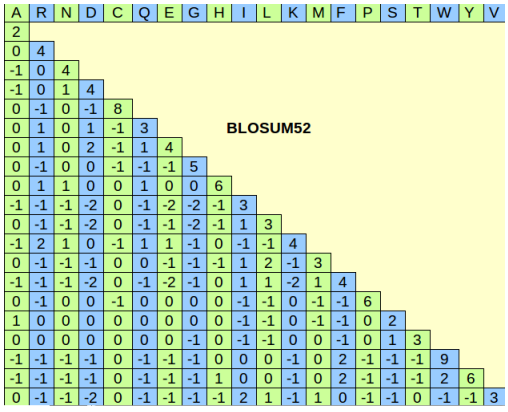

Figura 4: Matriz BLOSUM52 para Aminoácidos

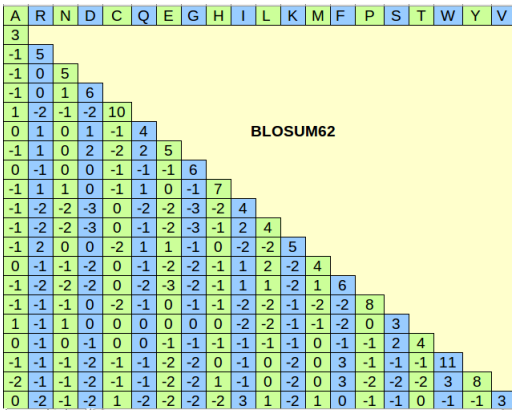

Figura 5: Matriz BLOSUM62 para Aminoácidos

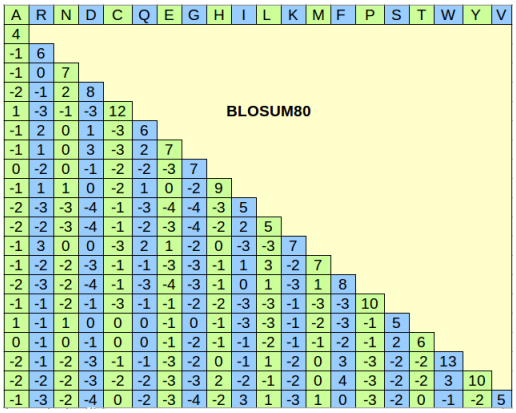

Figura 6: Matriz BLOSUM80 para Aminoácidos

9.2. Caso de estudio 2: Matriz de sustitución para pares de aminoácidos

9.2.1. Alfabeto de tuplas

El alfabeto de las tuplas de aminoácidos está conformado por 400 residuos que resultan a partir de la combinación de los 20 aminoácidos básicos. A continuación se muestran las matrices generadas a partir de la metodología.

Se generaron tres matrices correspondientes a los porcentajes de similitud 52, 62 y 80, todas a partir de la misma base de datos BLOCKS, dando origen a las matrices BLOSUM 52, BlOSUM 62 y BLOSUM 80. Con lo que se obtuvieron matrices simétricas cada una con un total de 19.900 valores los cuales describen el intercambio de una tupla de aminoácidos a otra. La mayoría de valores positivos se encuentran en la diagonal pero exceptuando estos valores tenemos que la tupla Triptófano-Tirosina(WY) se intercambia con alta probabilidad por la tupla Triptófano-Fenilalanina(WF) es decir que presenta un total de 80 puntos o más con lo que consideramos que es intercambio que se da frecuentemente teniendo en cuenta los resultados de la matriz de sustitución, esta relación se presento dentro de las matrices BLOSUM 52 y BLOSUM 62.

Para tener un punto de vista más amplio sobre los resultados arrojados por la matriz con respecto

Matriz de sustitución	Cantidad de intercambios encontrados	Intercambios encontrados
BLOSUM52 Tuplas	5	(CH, CG) (WW, WF) (WY, WF) (WY, WW) (VW,
BLOSUM62 Tuplas	4	(TW, SW) (WY, WF) (WY, WW) (YW, WW)
BLOSUM72 Tuplas	2	(QW, NW) (WY, WF)

Tabla 3: Intercambios para tuplas teniendo un sistema de puntos de 75

Matriz	80 Puntos	85 Puntos	90 Puntos	95 Puntos
BLOSUM52	20255	1605	161	0
BLOSUM62	8979	765	15	0
BLOSUM72	10459	876	14	0

Tabla 4: Cantidad de intercambios para un rango de puntuación de 80 a 95

al intercambio de tuplas se bajó el umbral de 80 puntos a 75 puntos para poder obtener resultados relacionados con los posibles intercambios mostrados en las matrices de sustitución.

9.3. Caso de estudio 3: Matriz de sustitución para tripletas de aminoácidos

9.3.1. Alfabeto de tripletas

El alfabeto de las tripletas de aminoácidos esta conformado por 8000 residuos que resultan a partir de la combinación de los 20 aminoácidos básicos. A continuación se muestran las matrices generadas a partir de la metodología.

Se generaron tres matrices correspondientes a los porcentajes de similitud 52, 62 y 80, todas a partir de la misma base de datos BLOCKS, dando origen a las matrices BLOSUM 52, BLOSUM 62 y BLOSUM 80. Con lo que se obtuvieron matrices simétricas cada una con un total de 31.996.000 valores los cuales describen el intercambio de una tripleta de aminoácidos a otra. Los valores dentro de las matrices están bastante distribuidos, por lo que no se muestra la tendencia que se observaba en los alfabetos anteriores donde pocos valores representaban intercambios con un porcentaje mayor al 80 % y por el contrario existen una gran cantidad de intercambios. Los totales se muestran a continuación.

Como podemos ver, al tener una cantidad tan grande de valores los puntajes altos están distribuidos a lo largo de la matriz de manera que tenemos gran cantidad intercambios por lo que no podemos mostrar de manera especifica las tripletas que muestran estos valores altos de manera organizada.

A manera de ejemplo mostramos las 15 sustituciones con un puntaje mayor a 90 puntos y que corresponden a la matriz BLOSUM62. El listado de sustituciones para la BLOSUM72 y BLOSUM52 se muestran en el anexo.

9.4. Alineamiento de diferentes secuencias de proteínas utilizando las matrices de sustitución encontradas

Los algoritmos de alineamiento permiten realizar una validación clara de los resultados obtenidos en una matriz de sustitución sin embargo para los alfabetos que se trabajaron durante el desarrollo de este trabajo de grado que fueron aminoácidos, tuplas y tripletas que fueron construidos a partir de

Intercambios
(CCW, CCF)
(CCY, CCW)
(CPW, CIW)
(CVW, CIW)
(CVW, CPW)
(HWH, HWN)
(HWY, HWN)
(HYW, HCW)
(WCT, WCM)
(WTC, WSC)
(WTW, WNW)
(WWN, PWN)
(WWY, WWF)
(YPW, FPW)
(YYC, YFC)

Tabla 5: Intercambios para la matriz BLOSUM62 para tripletas con una puntuación de 90

la base de datos BLOCKS. El algoritmo de alineamiento desarrollado cubre todos estos escenarios y por ello se plantean el estudio de diferentes alineamientos. Primero alineamientos entre diferentes secuencias pertenecientes a la misma familia, luego secuencias de diferentes familias pertenecientes a la misma superfamilia y finalmente entre 2 secuencias de diferentes familias que no pertenecen a la misma superfamilia.

Los resultados se están discutiendo ya que la idea de alinear tripletas y tuplas es algo nuevo que en la teoría se puede aplicar fácilmente pero es necesario entender y poder evaluar sus aplicaciones y la manera en que puede llegar a ser verdaderamente utilizados estos alineamientos.

10. CONCLUSIONES

- El proceso de creación de una metodología implica revisión minuciosa de los procesos actuales utilizados en la construcción de matrices de sustitución con el fin de establecer criterios puntuales y generales que deben ser tenidos en cuenta a la hora de construir una matriz de sustitución y que muestre la relación que existe entre los diferentes residuos que componen el alfabeto y que pueden ser aprovechados ya que nos muestran una relación biológica existente y que está presente en muchas secuencias que son las que conforman la base de datos.
- Los algoritmos de alineamiento pueden ser ajustados para trabajar con diferentes alfabetos que describen secuencias biológicas siempre y cuando se tenga claridad sobre cada uno de los elementos que compone el alfabeto y se cuente con la posibilidad de puntuar claramente diferentes posibles alineamientos.
- Construir una matriz de sustitución que funcione de manera correcta para todas las secuencias es una tarea complicada ya que si se toma una base de datos muy general los resultados no aportarán información mas allá que la mostrada por las secuencias de la base de datos con lo que datos específicos sobre un determinado grupo no son reflejados en las matrices obtenidas.
- En los casos de estudio trabajados pudimos ver las diferencias que se consiguen cuando se pasa de un alfabeto que describe tuplas de aminoácidos a uno de tripletas de aminoácidos con lo que la cantidad de opciones para la sustitución aumenta considerablemente las posibilidades de intercambio entre un elemento del alfabeto y otro con lo que se observa una matriz muy dispersa donde es complicado encontrar un equilibrio entre todos los valores presentes ya que algunos intercambios puede que nunca se vean reflejados.

11. TRABAJOS FUTUROS

- Encontrar la manera de generar bloques fácilmente a partir de un conjunto de secuencias biológicas en un alfabeto arbitrario conservando la idea de BLOCKS sin que se dependa completamente de tener una base de datos BLOCKS como punto de partida.
- Elaborar una funcionalidad que en la aplicación del algoritmo BLOSUM teniendo como base solamente la especificación del alfabeto se pueda ajustar automáticamente el algoritmo respecto a la cantidad de elementos dentro del alfabeto como interpretar las bases de datos del nuevo alfabeto es decir una especificación de alfabeto numero con lo que se puede trabajar fácilmente a nivel algorítmico para la construir de las bases de datos.
- Estudiar la posibilidad de realizar aplicaciones mas allá de alineamientos utilizando matrices de sustitución que representan alfabetos arbitrarios de secuencias biológicas como puede ser la búsqueda de secuencias del alfabeto definido en una base de datos.

Referencias

[1] M. L. Raymer and D. E. Krane, *Fundamental Concepts of Bioinformatics*. Pearson Education, 2003.

[2] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, no. 22, pp. 10915–10919, 1992.

[3] Z. Dosztányi and a. E. Torda, "Amino acid similarity matrices based on force fields.," *Bioinformatics (Oxford, England)*, vol. 17, no. 8, pp. 686–699, 2001.

[4] R. B. Vilim, R. M. Cunningham, B. Lu, P. Kheradpour, and F. J. Stevens, "Fold-specific substitution matrices for protein classification," *Bioinformatics*, vol. 20, no. 6, pp. 847–853, 2004.

[5] Y. K. Yu and S. F. Altschul, "The construction of amino acid substitution matrices for the comparison of proteins with non-standard compositions," *Bioinformatics*, vol. 21, no. 7, pp. 902–911, 2005.

[6] W.-M. Zheng and X. Liu, "A protein structural alphabet and its substitution matrix CLESUM," p. 10, 2004.

[7] D. W. Rice and D. Eisenberg, "A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence.," *Journal of molecular biology*, vol. 267, no. 4, pp. 1026–1038, 1997.

[8] M. Tyagi, V. S. Gowri, N. Srinivasan, A. G. de Brevern, and B. Offmann, "A substitution matrix for structural alphabet based on structural alignment of homologous proteins and its applications," *Proteins: Structure, Function, and Bioinformatics*, vol. 65, no. 1, pp. 32–39, 2006.

[9] X. Liu and W.-M. Zheng, "An amino acid substitution matrix for protein conformation identification.," *Journal of bioinformatics and computational biology*, vol. 4, no. 3, pp. 769–782, 2006.

[10] M. Duc Cao, T. I. Dix, and L. Allison, "Computing substitution matrices for genomic comparative analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5476 LNAI, pp. 647–655, 2009.

[11] S. Pietrokovski, J. G. Henikoff, and S. Henikoff, "The Blocks database–a system for protein classification.," *Nucleic acids research*, vol. 24, no. 1, pp. 197–200, 1996.

[12] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, pp. 443–453, Mar. 1970.

[13] O. Bastien, S. Roy, and E. Maréchal, "Construction of non-symmetric substitution matrices derived from proteomes with biased amino acid distributions," *Comptes Rendus - Biologies*, vol. 328, no. 5, pp. 445–453, 2005.

[14] E. Benard and C. J. Michel, "A generalization of substitution evolution models of nucleotides to genetic motifs," *Journal of Theoretical Biology*, vol. 288, no. 1, pp. 73–83, 2011.

[15] F. Fabris, A. Sgarro, and A. Tossi, "Splitting the BLOSUM score into numbers of biological significance," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2007, 2007.

[16] F. Frommlet, A. Futschik, and M. Bogdan, "On the significance of sequence alignments when using multiple scoring matrices," *Bioinformatics*, vol. 20, no. 6, pp. 881–887, 2004.

[17] S. Henikoff, J. G. Henikoff, and S. Pietrokovski, "Blocks+: A non-redundant database of protein alignment blocks derived from multiple compilations," *Bioinformatics*, vol. 15, no. 6, pp. 471–479, 1999.

[18] T. Müller, S. Rahmann, and M. Rehmsmeier, "Non-symmetric score matrices and the detection of homologous transmembrane proteins.," *Bioinformatics (Oxford, England)*, vol. 17 Suppl 1, pp. S182–S189, 2001.

[19] M. P. Styczynski, K. L. Jensen, I. Rigoutsos, and G. Stephanopoulos, "BLOSUM62 miscalculations improve search performance.," *Nature biotechnology*, vol. 26, no. 3, pp. 274–275, 2008.

12. ANEXOS

Todos los resultados que por el tamaño de la matriz o el peso del archivo impedían que su visualización en el documento fuera correcta están mostrados en esta carpeta en google Drive : `https://drive.google.com/folderview?id=0B1tS0doq61W7fjU2QkNoUk5ueHdmRXphYWRjSmNyVHA0RkRYMWp4alRGTnhLcDdkZWViMlU&usp=sharing`