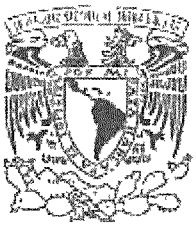

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA DE SISTEMAS – INVESTIGACIÓN DE OPERACIONES

APLICACIÓN DE LA HEURÍSTICA DE COMPOSICIÓN MUSICAL EN LA SOLUCIÓN DEL PROBLEMA DE ALINEAMIENTO MÚLTIPLE DE SECUENCIAS

TESIS
QUE PARA OPTAR POR EL GRADO DE
MAESTRO EN INGENIERÍA

PRESENTA
JULIO CESAR FRANCO NAVA

TUTOR PRINCIPAL
DRA MARÍA ELENA LÁRRAGA RAMÍREZ
INSTITUTO DE INGENIERÍA.

MÉXICO, D F JUNIO 2013

JURADO ASIGNADO:

Presidente	DR.JOSÉ JESÚS ACOSTA FLORES.
Secretario.	DRA IDALIA FLORES DE LA MOTA
Vocal.	DRA MARÍA ELENA LÁRRAGA RAMÍREZ.
1 er Suplente	M.I. FRANCISCA IRENE SOLER ANGUIANO.
2 d o. Suplente	DR ABEL CAMACHO GALVÁN

CIUDAD UNIVERSITARIA, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, MÉXICO, D F

TUTOR DE TESIS:

DRA MARÍA ELENA LÁRRAGA RAMÍREZ

FIRMA

Agradecimientos.

A la Universidad Nacional Autónoma de México (UNAM), y al posgrado en Ingeniería por permitirme formar parte de esta gran institución , por haberme albergado en sus aulas y por darme la oportunidad de continuar con mi formación académica y crecimiento como ser humano.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por su apoyo, que fue parte indispensable para poderme integrar de tiempo completo al programa de posgrado en ingeniería.

A los profesores: Dra. Ma. Elena Lárraga Ramírez , al Dr Erick Rincón García y al M.en I. Román Mora Gutierrez por su disposición, tiempo y asesoría en la elaboración del presente trabajo, por sus sugerencias, por compartir su experiencia, su amistad, así como su paciencia para aclarar mis dudas, y alentarme en todo momento para la culminación de este documento.

Al comité tutorial por la atención brindada y las obervaciones hechas.

A los profesores de la maestría por transmitirme parte de sus conocimientos que son un tesoro invaluable.

A mi madre, Esperanza Nava Arellano por ser pilar fundamental en mi vida, por su amor, comprensión, confianza y apoyo incondicional para realizar este sueño.

A mis hermanos Manuel, David, Eduardo, Andrés, Susana, Martha, Miguel y Rocío por su ejemplo y por su compañía y consejos.

A mis compañeros de maestría y amigos en general por su ánimo, alegria, energía y palabras de aliento.

A todos ellos, gracias de corazón.

Julio César.

Tabla de Contenido

Índice de tablas

Índice de figuras

Resumen

El presente trabajo plantea el empleo de la heurística de Composición Musical para la solución del problema de alineamiento múltiple de secuencias(AMS) , la validación de los resultados fue llevada a cabo haciendo uso de la metodología de Julie Thomson lo que permitió comparar los resultados obtenidos contra las soluciones que arrojan los métodos previamente desarrollados en la materia.

Para ello fue necesario adecuar la heurística de reciente aparición a un problema discreto de naturaleza binaria, ya que hasta donde se tiene conocimiento unicamente se ha aplicado a problemas enteros mixtos.

De esta forma, la implementación reportada en este trabajo es un punto de partida en la aplicación del algoritmo de Composición Musical en problemas de índole discreta, que muestra una nueva forma de abordar el tema desde el punto de vista de las sociedades artificiales . Los resultados obtenidos son satisfactorios pero se considera que podrían perfeccionarse en trabajos futuros.

Abstract.

This thesis method proposed a solving application to the multiple sequences alignment problem (AMS as its acronym in Spanish) right through Heuristic Musical Composition established method that have been used to solve continuous problem in real numbers and in this case was applied to a discrete binary problem.

The outcomes validation were done by Julie Thomson´s methodolgy whereby it let us to compare the previous results that were reported by other authors against of this thesis outcomes.

However the results have been obtained with this method could be applied for another researchers as a starting point that allowing them apply the Musical Composition established method but now to solve discrete problems,as new option to address the Multiple Sequences Alignment problem from the standpoint of artificial societies.

Introducción

En la actualidad, la investigación científica se abre camino haciendo uso de herramientas computacionales. En lo que respecta a la investigación médica, la necesidad de replicar biomoléculas que combatan enfermedades que aún no cuentan con un tratamiento terapéutico, ha promovido que varias ramas del conocimiento como las matemáticas y la computación apoyen a la biología de manera profunda en la búsqueda de soluciones. Así surge la bioinformática, la cual ha contribuido en gran medida al desarrollo de vacunas, medicinas y al entendimiento tanto de los mecanismos de evolución del genoma, como de la estructura de redes de interacción de las proteínas.

Recientemente se ha hecho apremiante encontrar patrones de similitud y diferencias entre las biomoléculas, estudiando con ahínco tanto el tratamiento de las secuencias en las que se encuentran ordenados sus componentes como la determinación de ancestros comunes (Attwood y Parry-Smith, 2002). Especialistas en la materia han desarrollado herramientas computacionales, que pueden clasificarse en métodos exactos y en heurísticos. Como parte de los métodos heurísticos están los algoritmos bioinspirados (imitan el comportamiento de varios fenómenos en la naturaleza como , por ejemplo, el proceso que siguen las hormigas para encontrar y seguir de manera ordenada la ruta más corta a su comida) y como subconjunto de éstos, se encuentran los algoritmos sociales: de reciente aparición , reducido tiempo de procesamiento y vasto campo de aplicación.

Este trabajo presenta la aplicación de un algoritmo social para encontrar solución a uno de los problemas de la bioinformática: el alineamiento múltiple de secuencias (ó AMS , como se mencionará de aquí en adelante). Su alcance contempla tal aplicación a una muestra estratificada de los conjuntos de secuencias que conforman la referencia 1 de BALIBASE (base de datos de biosecuencias) y a uno de los conjuntos de secuencias de la referencia 4. Para la validación de los resultados se empleó el método Julie Thomson para finalmente, mediante una normalización de los datos obtenidos , ubicar los resultados arrojados por el algoritmo de Composición musical dentro de una escala de comparación frente a los resultados de otros métodos desarrollados previamente. Finalmente, se aplicó el método Wilcoxon y el Bootstrap para determinar la eficiencia del método empleado en comparación con otros citados en BALIBASE.

El trabajo se presenta organizado de la siguiente manera: En el capítulo 1 del trabajo se describe el problema y se abordan los conceptos básicos necesarios para comprender la contribución de la bioinformática al estudio de las secuencias. En el capítulo 2, se muestra una breve reseña de los estudios que se han levado a cabo para resolver el problema del alineamiento de secuencias y se describen las características del algoritmo social que se pretende emplear . El capítulo 3 describe la manera en la que se resolverá el problema de alineamiento múltiple de secuencias con el algoritmo de Composición Musical, así como los resultados obtenidos. Finalmente, en el capítulo 4 cita las conclusiones de este trabajo y posibles sugerencias para trabajos futuros.

Objetivo General

El objetivo del presente trabajo es encontrar soluciones al problema del alineamiento múltiple de secuencias mediante el uso del algoritmo social "Composición Musical", cuyo empleo representa una innovación en el tratamiento de dicho problema.

Objetivos Específicos

1. Aplicar el algoritmo de Composición Musical al problema de alineamiento multiple de secuencias.

2. Comparar la eficiencia del uso de la heurística mediante el método Julie Thomson.

Capítulo 1
Generalidades.

Este capítulo aporta elementos que sirven como introducción a la comprensión del problema del AMS, desde donde surge la necesidad de abordar el tema, descripción del problema, conceptos básicos tanto biológicos como matemáticos, hasta características del problema del AMS. El objetivo es adentrar al lector en los conceptos relacionados con la materia.

1.1. Bioinformática.

El término Bioinformática significa tecnología de la información aplicada a la gestión y análisis de datos biológicos (Attwood y Parry-Smith, 2002), a continuación de dará una breve reseña. En el siglo XIX, el monje Austriaco Gregor Mendel inició una revolución genética dando a la ciencia su primera herramienta para entender cómo se heredan las características en los individuos. En 1953 la identificación del Ácido Desoxirribonucleíco (ADN) como material genético encaminó esta revolución en una nueva dirección, que permitió a los investigadores correlacionar directamente los cambios en los genes con el combate contra las enfermedades. Sin embargo, fue hasta la segunda mitad de la década de los 90's, cuando gracias a la aplicación de la tecnología en el estudio de secuencias de ADN, el manejo de la información genética realmente tomó auge; lo que permitió revelar la secuencia completa de los nucleótidos o genomas enteros. No obstante el avance de las investigaciones realizadas en este campo de estudio, hoy en día hay muchas preguntas aún por resolver que impiden establecer estrategias para el combate a las enfermedades, un ejemplo de ello, son las mutaciones que sufren los agentes que las provocan.

Como una herramienta que ayude a la comparación entre cadenas de ADN , el presente trabajo abordará el AMS, el cual se entiende como el problema de comparar la similitud entre tres o más secuencias generalmente protéicas, de ADN o de Ácido Ribonucléico (ARN) (Chan y otros, 1992) y que se aplica para encontrar subregiones altamente conservadas (patrones) de n conjuntos de biosecuencias e inferir la historia evolutiva de un conjunto de grupos de individuos emparentados a través de la asociación de sus secuencias biológicas (Mora-Gutierrez, 2009).

En este capítulo explicaremos muchos de los conceptos arriba citados con el fin de proporcionar las herramientas necesarias para la comprensión del problema del AMS.

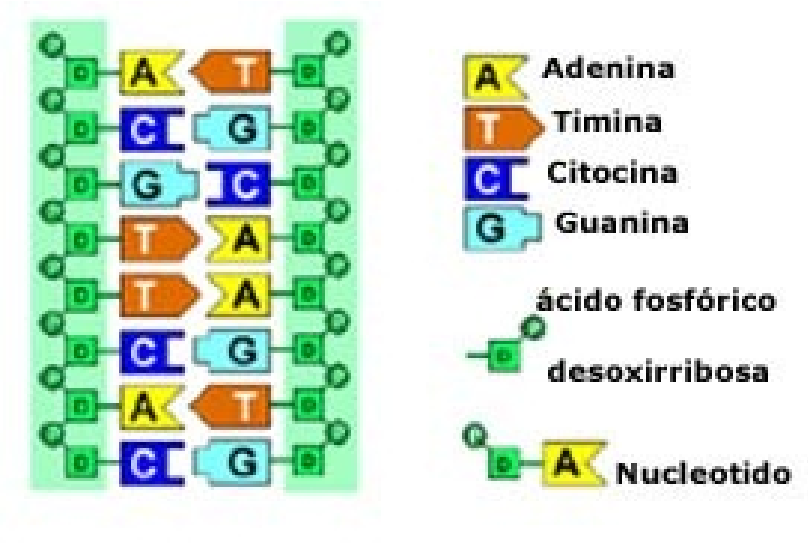

Figura 1.1: Estructura del ADN

1.2. Conceptos Básicos

Con la finalidad de inducir al lector en el problema de Alineamiento de Secuencias Múltiples, sobre el que se enfoca este trabajo, en lo siguiente se definen algunos conceptos relacionados con el mismo.

Las secuencias de proteínas y nucleótidos juegan un papel importante en la bioinformática. Se define el concepto de cadena como una sucesión organizada de elementos que pueden ser símbolos o letras de un alfabeto y que es representada simplemente por una concatenación de tales elementos (Chan y otros, 1992). En la literatura, secuencia es sinónimo de cadena (Sankoff y otros, 1982) (Chan y otros, 1992).

Un ensamble de secuencias es un conjunto de secuencias cuyos elementos que los componen vienen del mismo alfabeto (Chan y otros, 1992) y , en un ensamble, dos secuencias normalmente no son idénticas debido al proceso de substituciones, inserciones o eliminaciones de los elementos de una, dos o ambas secuencias (Chan y otros, 1992).

Una Biosecuencia se refiere a la representación simbólica de las cadenas de nucleótidos (Adenina (A), Guanina (G), Timina (T), Citosina (C) y Uracilo (U)) o protéicas.

Los nucleótidos son bases nitrogenadas que conforman al Ácido Desoxirribonucleico o ADN (A,C,G,T) y al ácido ribonucleíco o ARN (A,C, G, U).(véase figura 1.1). La disposición y secuenciación de los nucleótidos determina la codificación de la información biológica. A la unión de tres nucleótidos se le denomina codones y la unión de dos codones forma una proteína. El orden y disposición de los aminoácidos se rige por el código genético (Mora-Gutierrez, 2009).

a) Alineamiento 1 de S_1, S_2 y S_3.

S_1	a	b	c	d
S_2	b	c	d	-
S_3	a	b	-	a

b) Alineamiento 2 de S_1, S_2 y S_3.

S_1	a	b	c	d
S_2	-	b	c	d
S_3	a	b	-	a

Figura 1.2: Ejemplos de Alineamientos

La secuenciación lineal de los aminoácidos contiene la información necesaria para generar una molécula protéica con una estructura tridimensional particular. A esta secuencia se le llama estructura primaria de la proteína y cada posición es denominada residuo (Mora-Gutierrez, 2009).

El Alineamiento de Secuencias se refiere a la comparación lineal de secuencias amoniácidas (o de ácidos nucléicos) en la que se introducen inserciones para hacer que posiciones equivalentes en secuencias adyacentes se situen en el registro correcto. Los alineamientos son las bases de los métodos de análisis de secuencias y se emplean para resaltar la presencia de motivos conservados (Attwood y Parry-Smith, 2002).

La comparación entre secuencias de nucleótidos se usa cuando se desea determinar el grado de similitud entre individuos (estudios filogenéticos, genética de poblaciones, etc),lo que permite identificar genes variantes entre los individuos de una familia de secuencias (Mora-Gutierrez, 2009) (véase figura 1.2). En ella se muestran 2 alineamientos para las secuencias $S_1, S_2 y S_3$. El mejor es el el alineamiento b) ya que tiene 1 columna con todas las coincidencias.

Las secuencias biológicas se pueden clasificar en homólogas y análogas. Son homólogas

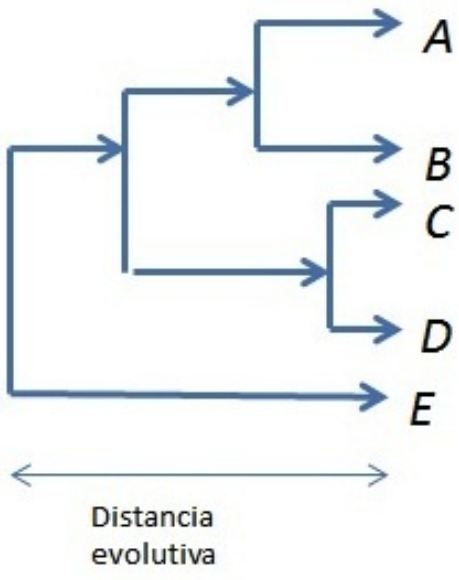

Figura 1.3: Ejemplo de análisis filogenético de Mutaciones

si se relacionan evolutivamente por un ancestro común, es decir, han evolucionado desde la misma posición ancestral. Y son secuencias análogas si se relacionan funcionalmente pero no tienen un ancestro en común (Mora-Gutierrez, 2009).

La comparación de secuencias de amoniácidos es apropiado para buscar homólogos (Mora-Gutierrez, 2009).

Debido al proceso evolutivo, con frecuencia las secuencias biológicas de individuos de una misma especie no coinciden en todos sus eslabones, dando origen a las llamadas mutaciones.

Una Mutación es el conjunto de las substituciones, inserciones o eliminaciones de elementos en las macromoléculas (Chan y otros, 1992). (véase ejemplo figura 1.3)

La Similitud entre secuencias se calcula mediante la asignación de una calificación que mide el grado de coincidencia en los caracteres (*match*) de un conjunto de secuencias, y una calificación de penalización cuando dicha coincidencia no se cumple (*mismatch*) (Najarian y otros, 2009). Es importante resaltar que, al momento de analizar la similitud de un par de secuencias deben considerarse los gaps, llamados también regiones INDEL, derivados de inserciones (desfasamientos) o residuos que han sido eliminados. Las regiones INDEL son espacios que se insertan en cada una de las cadenas al momento de alinearlas ente ellas y que, de manera generalizada, estos espacios son llenos con el caracter - (guión) (Najarian y otros, 2009).

La identificación de gaps es importante ya que, a menor cantidad de gaps, existirán calificaciones más altas e incluso, en algunos casos, se deberán determinar penalizaciones al gap para prevenir los desfasamientos de secuencias. Las penalizaciones por este desfasamiento deben ser ponderadas de acuerdo a la magnitud de los caracteres involucrados y a la repercusión que tales desfasamientos puedan tener en la comparación.

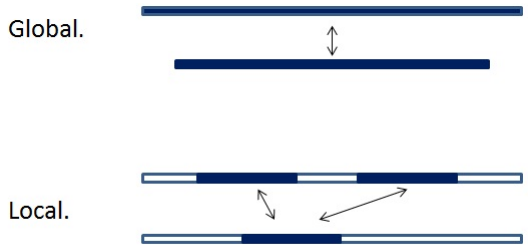

Figura 1.4: Alineamiento Local y Global

Varios métodos se han desarrollado para el análisis del alineamiento de secuencias. Como punto de partida se ha tomado el análisis de sólo dos secuencias y se denomina éste como Alineamiento de Pares de Secuencias. Este concepto se extiende para el análisis de más de dos cadenas, llamado Alineamiento Múltiple de Secuencias (AMS).

Existen dos métodos diferentes para la alineación de pares de secuencias:

- El Alineamiento Global, busca el mejor alineamiento entre dos secuencias de manera general, aunque algunas subsecuencias con alta calificación pueden encontrarse desfasadas.
- El Alineamiento Local, busca la calificación más alta de las subsecuencias. Es decir, busca la distancia mínima entrecadenas buscando el máximo número de subcadenas existentes entre las secuencias a comparar. (véase figura 1.4)

(Najarian y otros, 2009).

El alineamiento de pares de secuencias puede ser procesado en tiempo y espacio cuadrático usando para ello la programación dinámica (DP). Más adelante se verá que la aproximación Hirschberg, también llamada divide y vencerás, reduce la complejidad del espacio de cuadrático a lineal duplicando el número de operaciones (Schmidt, 2010).

La DP lleva a una complejidad de tiempo exponencial para los Alineamientos Múltiples de Secuencias, es por ello que las heurísticas son usadas con más frecuencia en esos casos.

A continuación se definen de manera formal los conceptos arriba citados de alfabeto, secuencia y alineamiento, y se explican, de manera profunda, diferentes métodos de calificación y alineamiento. Ello con el fin de describir de manera más profunda el Alineamiento de Secuencias.

1.3. Alineamiento de Secuencias.

Se iniciará por definir formalmente el concepto de secuencia y alfabeto para posteriormente conocer las características de los alineamientos de pares de secuencias y llegar así al

Alineamiento Múltiple.

Considere una secuencia S de longitud l que cubre el alfabeto Σ. Usaremos la siguiente notación:

1. $S[i \ldots .j]$ denota la subcadena de S que comienza en la posición i y termina en la posición j, esto es, $S = S[0 \ldots .l-1]$.
2. $S[i]$ denota la letra de S en la posición i.
3. $|S|$ denota la longitud de la cadena S, esto es, $|S| = l$
4. La cadena de longitud cero es llamada cadena vacía y es denotada como ε.
5. El símbolo de gap es denotado como $-$, donde $-$ no pertenece a Σ

El alfabeto usado frecuentemente en bioinformática es el alfabeto DNA con cuatro nucleótidos (por ejemplo, $\Sigma = A, C, G, T$) y el alfabeto de proteínas con 20 aminoácidos estándar (por ejemplo: $\Sigma = A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y$).

A continuación se muestra la codificación de aminoácidos en el alfabeto.

Alanina (A), Arginina (R), Asparagina (N), Ácido Aspártico (D), Cisteína (C), Ácido glutámico (E), Glutamina (Q), Glicina (G), Histidina (H), Isoleucina (I), Leucina (L), Metionina (M), Fenilalanina (F), Prolina (P), Serina (S), Treonina (T), Triptófano (W), Triosina (Y), Valina (V) (Arenas˜Dáz, 2009).

1.3.1. Similitud.

Encontrar las diferencias entre secuencias es frecuentemente equivalente a encontrar similitudes entre ellas (Pevzner, 2000). Así, la similitud de secuencias puede entenderse también como el grado de proximidad existente entre ellas. Un conjunto de biosecuencias son similares por las siguientes razones:

1. Filogenéticas. Secuencias homólogas.
2. Funcionales. Generadas por convergencia evolutiva.
3. Limitaciones físicas. Por ejemplo los dominios trans membrana tienen que ser hidrófobos a pesar de que tienen funciones muy distintas.
4. Presencia de secuencias repetidas. Su contenido informativo es bajo y su interpretación puede conducir a errores graves.

En la teoría de similitud los conceptos de homología y semi-homología contribuyen a determinar cuán parecidas son las cadenas.

La homología es la relación existente entre dos individuos (o partes orgánicas) diferentes cuando sus determinantes genéticos tienen el mismo origen evolutivo. Y se utiliza para

describir el porcentaje estimado de la similitud (porcentaje de posiciones idénticas de las secuencias en la comparación) (Leluk, 2000).

La semi-homología implica la posibilidad de sustitución de un residuo x por otro residuo en un punto de mutación de los codones , de forma que las subsecuencias puedan transformarse una en la otra (Mora-Gutierrez, 2009). Recordemos que las posiciones alineadas que no corresponden entre las secuencias indican las mutaciones a partir del ancestro común, en cualquiera de las secuencias participantes.

1.3.1.1. Métodos de cuantificación.

Los métodos de cuantificación tienen como objetivo comparar dos biosecuecnias. Se refieren a determinar el grado de similitud entre ellas.

Los métodos más utilizados son los siguientes:

Matrices de puntos.

Es una representación gráfica con el objeto de comparar dos biosecuencias donde se pone de manifiesto las regiones de similaridad entre ambas, las cuales pueden ser apreciadas a simple vista por la detección de patrones. La idea base de estas comparaciones es usar dos secuencias como coordenadas de una gráfica bidimensional y comparar cada una de las posiciones donde exista similitud. Es una técnica cualitativa sencilla, pero consume mucho tiempo para análisis a gran escala. La complejidad del algoritmo empleado es de $O(l_a * l_b)$, donde l son las longitudes de las secuencias (Mora-Gutierrez, 2009). (véase figura 1.5) .

Ahora bien, considerando que la distancia es el inverso de la similitud, entonces la similitud también puede ser medida en términos de los métodos de cuantificación de distancia.

Distancia de Hamming. Es la forma más sencilla de cuantificar distancia. En este método se determina el número de posiciones donde las secuencias son diferentes mediante el análisis simultáneo de un par de secuencias. La definición se describe a continuación:

Dadas las secuencias $S_a = [S_{a_1}, S_{a_2}, \ldots S_{a_l}]$ y,$S_b = [S_{b_1}, S_{b_2}, \ldots S_{b_l}]$, ambas con una longitud l y si $S_a \in mathboxA$ y $S_b \in mathboxA$, entonces se define la distancia entre S_a y S_b como $d(S_a, S_b)$ como el número de componentes tales que $S_{a_i} \neq S_{b_i}$ para $1 \leq i \leq l$ (Grimaldi, 1998).

$$d(S_a, S_b) = \sum_{i=1}^{l} d(S_{a_i}, S_{b_i}) \tag{1.1}$$

donde: $d(S_a, S_b)$: Distancia entre la secuencia S_a y la S_b
. $d(S_{a_i}, S_{b_i})$: Distancia entre el $i - ésimo$ eslabón de ambas secuencias.

$$d(S_{a_i}, S_{b_i}) = \begin{cases} 1 & \text{si } S_{a_i} = S_{b_i} \\ 0 & \text{si } S_{a_i} \neq S_{b_i} \end{cases} \tag{1.2}$$

La distancia de Hamming sólo permite el reemplazo de caracteres asociado a un costo, sin embargo solo se define para secuencias con la misma longitud (Mora-Gutierrez, 2009).

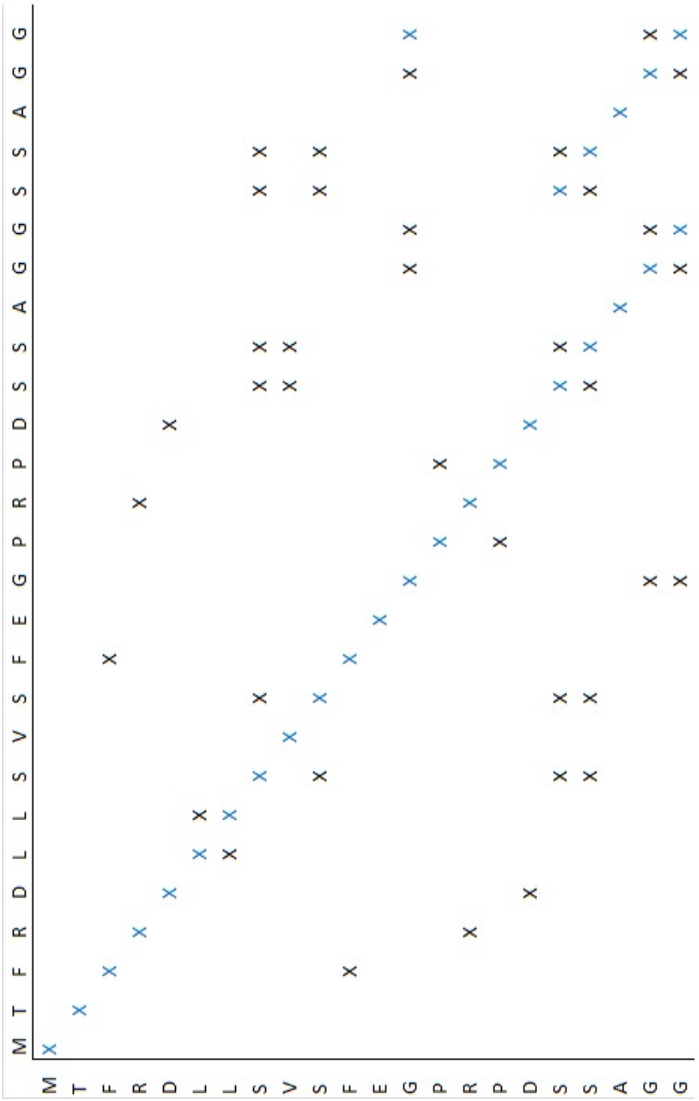

Figura 1.5: Ejemplo de matriz de punto

Distancia de Levenshtein.

En 1966 Levenshtein introdujo la noción de edición de distancia entre las cadenas como el número mínimo de operaciones elementales necesarias para transformar una cadena en otra, donde las operaciones elementales son inserción, eliminación o sustitución de un símbolo por otro (Pevzner, 2000). Es una métrica de distancia parecida a la de Hamming, sin embargo Levenshtein permite evaluar secuencias de longitud igual o diferente, e introduce el concepto de costo mínimo asociado con las operaciones necesarias para transformar una secuencia $S_a = [S_{a_1}, S_{a_2}, \ldots . S_{a_{l_a}}]$ en la secuencia $S_b = [S_{b_1}, S_{b_2}, \ldots . S_{b_{l_b}}]$ (Lee y otros, 2007), esto contribuye a que, si se ordenan las secuencias en una matriz, se permita el desplazamiento de éstas con el objeto de buscar su alineamiento.

La distancia de Levenshtein se fundamenta en el concepto de homomorfismo que involucra la existencia de una función que permita transformar una cadena en otra preservando sus características, a continuación se describe dicho concepto.

Homomorfismo. Sean $S_a = [S_{a1}, S_{a2}, \ldots S_{al_a}]$ y $S_b = [S_{b1}, S_{b2}, \ldots S_{bl_b}]$ secuencias de una familia. Un homomorfismo h de S_a en S_b es una función $h : S_a \rightarrow S_b$ tal que:

El homomorfismo preserva las relaciones y funciones de las cadenas (Enderton, 1987),y (Grimaldi, 1998).

La complejidad del algoritmo de Levenshtein es de $O(l_a * l_b)$.

Distancia Indel.

Si las operaciones de edición de Levenshtein se limitan sólo a inserciones y eliminaciones (no sustituciones), entonces el problema de edición de distancia es equivalente al problema de subsecuencia común más larga denotado como LCS. Dadas dos secuencias $S_a = S_{a1}, \ldots ., S_{al_a}$ y $S = S_b, \ldots S_{bl_b}$, una subsecuencia común de S_a y S_b de longitud k es una secuencia de índices $1 \leq i_1 < \ldots . < i_k \leq l_a$ y $1 \leq j_1 < \ldots < j_k \leq l_b$ tal que $S_{ai_t} = S_{bj_t}$ para $1 \leq t \leq k$.

Se denotará a $LCS(S_a, S_b)$ como la longitud de la subsecuencia común mas larga (LCS) de S_a y S_b. Entonces, como ejemplo, $LCS(A\mathbf{TCT}GA\mathbf{T}, \mathbf{TGC}A\mathbf{TA}) = 4$, ya que las letras que conforman el LCS son $TCTA$. Claramente $n + m - 2LCS(S_a, S_b)$ es el número mínimo de inserciones y eliminaciones necesarias para transformar S_a en S_b (Pevzner, 2000).

Entonces, de manera general, la complejidad del algoritmo de LCS es de $O(l_a * l_b)$ (Mora-Gutierrez, 2009).

Distancia de Damerau (Damerau- Levenshtein).

Este método incorpora, al método de Levenshtein, la operación de transposición de caracteres adyacentes y busca el número mínimo de operaciones (transponer, sustituir, insertar y eliminar) que se llevan a cabo para transformar una secuencia S_a en otra S_b. La complejidad del algoritmo de Damerau es de $O(l_a * l_b)$.

Para una matriz de alineamiento denotada como M,el número de columnas de dicha matriz corresponde a la longitud $\hat{l}$ de las secuencias después de la inserción de los espacios vacios.

El valor del alineamiento (valor de la función objetivo) en una matriz M se denota por $V(M)$ y se define utilizando alguna métrica de distancia de la manera siguiente:

$$V(M) = \sum_{i=1}^{\hat{l}} d(S_{ai}, S_{bi}) \tag{1.3}$$

donde: $d(S_{a_i}, S_{b_i})$ es una métrica de comparación de elementos en la $i - ésima$ columna de la matriz de alineamiento (Gusfieeld, 1993).

Alineamiento óptimo. Un alineamiento M con un $V(M)$ es óptimo si y sólo si no existe otro alineamiento M' con un valor $V(M')$ que mejore el valor de la función objetivo.

A continuación se señalan una serie de características de las funciones de distancia:

1. Distancia entre secuencias iguales. $d(S_a, S_b) = 0 \leftrightarrow S_a = S_b$ (Lee y otros, 2007).

2. La distancia entre la secuencia S_a y S_b es cero sí y solo si ambas secuencias son iguales ($S_a = S_b$), es decir, ambas secuencias contienen los mismos elementos en la misma posición, por tanto en ellas, no haría falta ninguna operación para transformarlas.

3. En los problemas de distancia entre secuencia se busca el mímino valor de diferencias entre las cadenas del conjunto (Mora-Gutierrez, 2009)

4. Cota de distancia. Dadas las secuencias $S_a = [S_{a1}, S_{a2}, \ldots S_{al_a}]$ y $S_b = [S_{b1}, S_{b2}, \ldots S_{bl_b}]$ de longitud l_a y l_b, respectivamente, donde $l_a \neq l_b$, la distancia entre ambas secuencias quedará contenida dentro del siguiente rango:

$$0 \leq d(S_a, S_b) \leq max(l_a, l_b)$$

donde: $d(s_a, s_b)$ es la distancia entre el par de secuencias. $max(l_a, lb)$ es la longitud de la secuencia más grande (Mora-Gutierrez, 2009).

Matrices de sustitución. Son herramientas de comparación utilizadas para secuencias de aminoácidos, y se basan en un análisis de la probabilidad de sustitución de un caracter por otro en la secuencia s_a para generar una secuencia s_b (Mora-Gutierrez, 2009).

Dentro de la construcción de las matrices de sustitución se implican los principios de mínima mutación, homología y semi-homología (Mora-Gutierrez, 2009).

Principio de mínima mutación. En el proceso evolutivo se involucra lo menos posible la ocurrencia de mutaciones idénticas, para individuos de diferentes líneas evolutivas (Sankoff y otros, 1982).

Matrices de datos de mutación (MDM) o matrices de puntuación.

Los algoritmos de alineamiento utilizan estas matrices para calificar cada coincidencia entre las secuencias. Estas matrices aceptan los principios de mutación (cambio puntual o generalizado) (Dayhoff y otros, 1983) y consideran la periodicidad de que un caracter cambie por otro caracter en una secuencia S_a en función del tiempo.

Es decir, en las matrices de puntuación se considera a las diferencias entre secuencias como el resultado de la divergencia evolutiva de los individuos a través del tiempo, ya que la

similitud entre secuencias puede ser una pista para identificar origenes evolutivos comunes o funciones comunes (Pevzner, 2000).

Los datos de entrada para el análisis estadístico pueden provenir de dos fuentes: a) la comparación de todos los segmentos de una secuencia S_a contra todos los segmentos de otra secuencia S_b. b) el mejor de ambas secuencias (Dayhoff y otros, 1983).

Las matrices PAM y BLOSUM son ejemplos de las matrices de sustitución. Las matrices PAM comparan especies cercanamente relacionadas mientras las matrices BLOSUM se usan para alineamientos de las proteínas evolutivamente divergentes (Mora-Gutierrez, 2009).

Matrices PAM.

Se basan en el concepto de mutación puntual aceptada (PAM: Point Accepted Mutation). Fueron desarrolladas por Dayhoff, y proponen un esquema de puntuación evolutiva para el alineamiento de los elementos de la S_a con los elementos de la S_b utilizando información estadística considerando el cociente de la probabilidad de sustitución de un aminoácido por otro respecto a la probabilidad de que los dos aminoácidos se presenten por azar. Por ejemplo, una distancia evolutiva de 1 PAM emplea la probabilidad de que un residuo mute un lapso de tiempo en el que se acepta una mutuación puntual por cada 100 residuos. Si se desea considerar matrices de mutuación a intervalos más grandes de distancia evolutiva, se multiplica repetidamente la matriz por sí misma, así, la matriz PAM 250 proporciona la puntuación por similitud equivalente para la persistencia de un 20 por ciento de coincidencias entre dos secuencias.

A mayor distancia evolutiva, la frecuencia de los cambios conservados crece y las coincidencias entre ambos elementos disminuye. Asemejando un proceso de diversificación evolutiva (Mora-Gutierrez, 2009).

Como al comparar una secuencia con otra, posición a posición, se multiplican las probabilidades de cada posición para calcular la puntuación para todo el alineamiento, es más conveniente emplear los logaritmos de las probabilidades. Así, la puntuación de cada pareja de elementos se calcula con el logaritmo del cociente de la probabilidad de ocurrencia del par y la probabilidad esperada basada en la frecuencia de cada aminoácido.(véase figura 1.6)

Matrices BLOSUM.

Las matrices de substitución PAM están condicionadas a secuencias con un mínimo de identidad del 85 por ciento y en la práctica la mayor parte de las secuencias no cumplen con esta condición, por tanto se han desarrollado otras matrices de sustitución. Una de ellas fue desarrollada por Altschul en 1991 y es la llamada BLOSUM, la cual deriva de un conjunto de matrices de sustitución a partir de bloques de secuencias alineadas en la base de datos BLOCKS. Usando las BLOSUM, en cada agrupación, se calcula la contribución media de cada posición, es decir, cualquier sesgo potencialmente introducido al contar contribuciones múltiples de pares de residuos idénticos es eliminado mediante el agrupamiento de segmentos de secuencia según un porcentaje mínimo de identidad (Attwood y Parry-Smith, 2002).

Otra diferencia significativa es que las matrices PAM asumen un proceso markoviano para la sustitución de aminoácidos (Leluk, 1998). Mientras que las matrices BLOSUM no se basan en ningún modelo explícito de evolución y consideran secuencias de proteínas empiricamente relacionadas que comparten un antepasado común.

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	12																			
S	0	2																		
T	-2	1	3																	
P	-3	1	0	6																
A	-2	1	1	1	2															
G	-3	1	0	-1	1	5														
N	-4	1	0	-1	0	0	2													
D	-5	0	0	-1	0	1	2	4												
E	-5	0	0	-1	0	0	1	3	4											
Q	-5	-1	-1	0	0	-1	1	2	2	4										
H	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6				
V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W	-8	-3	-5	-6	-6	-7	-4	-7	-7	-3	3	2	-3	-4	-5	-2	-6	0	0	17

Figura 1.6: Ejemplo de Matriz 250PAM

1.3.2. Alineamiento.

El alineamiento es un procedimiento de comparación de dos o más secuencias, lo cual se logra al determinar una serie de caracteres individuales o patrones de caracteres que se encuentran en el mismo orden en el conjunto de secuencias.

El caso más sencillo del AMS es el alineamiento de dos cadenas, por tanto se iniciará con su estudio para, posteriormente, extender su aplicación al problema del Alineamiento Múltiple de Secuencias.

En términos generales el alineamiento de S_a y S_b se logra al insertar espacios vacíos (guiones) en una u otra secuencias según convenga, de tal forma que se logra el mayor número de coincidencias y diferencias mínimas entre los caracteres (Mora-Gutierrez, 2009).

El alineamiento de dos secuencias dadas $S_a = [S_{a1}, S_{a2}, \ldots S_{al_a}]$ y $S_b = [S_{b1}, S_{b2}, \ldots S_{bl_b}]$ implica determinar una matriz M_{2*x} tal que $x \geq Max(l_a, l_b)$, cuyos elementos en la matriz M pertenecen al conjunto $\beta = A \cup -$ y ninguna columna de M consta completamente de espacios vacios. Al eliminar todos los guiones del primer y segundo renglón de M son iguales a S_a y S_b respectivamente (Lee y otros, 2007).

Dado el número exponencial de posibles alineamientos, se vuelve necesario encontrar una función de evaluación para identificar el mejor alineamiento. Éste es el objetivo de la matriz resultante.

Alineamiento. Definición: Dada una familia de x-secuencias ($S_1 = [b_1^1, b_2^1, \ldots . b_{l_1}^1], S_2 = [b_1^2, b_2^2, \ldots . b_{l_2}^2], \ldots . S_x = [b_1^x, b_2^x, \ldots . b_{l_x}^x]$), se obtiene un alineamiento $M = [\hat{S_1}, \hat{S_2}, \ldots \hat{S_x}]$ de secuencias con caracteres sobre β, donde todas las secuencias de M tienen la misma longitud y cada $\hat{S_a}$ se obtiene apartir de insertar guiones (celdas vacías) a S_a (Manthey, 2003). Además los elementos de M deben satisfacer la ecuación siguiente:

$$\bigcup_{i=max\{l_1,l_2,\ldots l_x\}}^{l_1+l_2+\cdots+l_x} \prod_{j-1}^{x} beta^i \tag{1.4}$$

Donde: $\beta : A \cup \{-\}$. l_i:longitud de la secuencia i. x: número de secuencias a comparar.

Sujeta a :

$$\text{máx}\{l_1, l_2, \ldots, l_x\} \leq \hat{l} \leq \{l_1 + l_2 + \cdots + l_x\} \tag{1.5}$$

(Mora-Gutierrez, 2009).

Un alineamiento de secuencias se obtiene insertando guiones en cada secuencia, si es necesario, de forma que: a) las secuencias resultantes obtengan la misma longitud b) cada columna tenga por lo menos un caracter diferente al vacío.

Los alineamientos de secuencias pueden clasificarse por:

1.- Número de secuencias analizadas: a) Alineamiento de un par de secuencias: en él solo se analizan dos secuencias b) Alineamiento Múltiple: se analizan tres o mas secuencias y el resultado es una secuencia consenso, ésta es una secuencia media.

2.- Nivel de análisis: a) Alineamiento global: consiste en buscar la distancia que representa el costo mínimo por mutaciones considerando la totalidad de los elementos de cada secuencia.

b) Alineamiento local: consiste en buscar la distancia mínima entre cadenas identificando el máximo número de subcadenas existentes entre las secuencias a comparar.

De forma general, los patrones pueden ser divididos en determinísticos o probabilísticos.

Los determinísticos se definen como el ajuste para un determinado patrón, generalmente son modelos binarios y se dividen en:

1. Los Oligos: Se reconocen como la función que corresponde a 1 si ambos caracteres en el i-ésimo lugar de las secuencias comparadas son iguales y 0 en otro caso.

2. Expresiones regulares: retorna 1 si existe una subsecuencia idéntica a la expresión regular dada. Estos modelos son usados en el descubrimiento de motivos para composición de símbolos exactos, símbolos ambiguos, espacios fijos o flexibles.

3. Expresiones discordantes: estos modelos evalúan el número de discordancias (distancia de Hamming) entre una subsecuencia y la subsecuencia consenso (Mora-Gutierrez, 2009).

En los patrones probabilísticos, para cada secuencia se genera una probabilidad de ocurrencia a partir de un modelo (matrices MDM). Dichos patrones ofrecen la ventaja de representar de forma implícita las reglas de discriminación (Mora-Gutierrez, 2009).

En la detección de patrones, existen tres diferentes niveles establecidos según su grado de dificultad.

1. Detección de patrones para una secuencia dada: dicho patrón se debe describir por medio de un algoritmo.

2. Esquemas. Es un modelo de un patrón, se establece la estructura de los elementos invariantes, los cuales se ubican dentro del conjunto de datos.

3. Probar patrones. En un conjunto de datos secuenciados de los que se desconoce el patrón son comparados con un patrón x y se verifica el grado de adaptación de los datos. Esta categoría se reconoce como aprendizaje no supervisado.

.

En la siguiente sección se ocuparan las ideas anteriores para describir las formas de expresión del AMS como un problema de optimización.

1.3.2.1. Características del Alineamiento Múltiple de secuencias.

Función objetivo.

Dentro de la literatura no se ha generalizado el uso de una función única para medir la calidad del alineamiento en x-secuencias (Ma y otros, 2007) y (Gusfieeld, 1993). Las funciones más conocidas son: a) funciones de puntaje (penalización por espacios vacios y suma por pares de las diferencias) b) alineamiento por templete (alineamiento por secuencia de consenso) y c) alineamiento por secuencia media (Mora-Gutierrez, 2009).

Un rasgo común a todas las funciones objetivo es determinar el grado de similitud entre las secuencias.

Funciones de puntaje.

Penalización por espacios vacíos.

Es un problema NP-Duro. Donde se considera un costo asociado a la introducción o expansión de espacios vacíos en las secuencias, en el cual resulta más costoso introducir un nuevo espacio vacío que solo agrandar uno ya existente. El objeto de esta función objetivo es minimizar los costos asociados a los guiones.

Función objetivo penalización por espacios vacios.

$$\text{mín}\, z : \sum_{\substack{a=1 \\ a \neq b}}^{x} \sum_{b=1}^{x} f(S_a, S_b) \tag{1.6}$$

donde:

x: Número de secuencias. S_a, S_b: Son un par de secuencias cualesquiera del conjunto de interés. $f(S_a, S_b)$: Es la función asociada a los costos por introducir y por extender los espacios vacíos, dichos costos deben ser positivos.

Suma por pares de las diferencias.

Es una función de puntaje y se le conoce en la literatura como SP-Score (por sus siglas en inglés) Elías demostró que el problema de AMS es NP-Duro para toda formulación con una función SP-diferencias (Manthey, 2003).

El alineamiento de dos cadenas S_a y S_b es una matriz de dos filas tal que la primera fila contiene los caracteres de S_a en orden con espaciones insertados y en la segunda fila los caracteres de S_b de igual manera. La calificación del alineamiento está definido como la suma de las calificaciones de sus columnas. La calificación de las columnas es positiva para las letras que coinciden y negativa para letras diferentes (Pevzner, 2000).

Ahora bien, para un Alineamiento Múltiple de un conjunto $mathboxF$ de x secuencias, M es la matriz $mathboxFx\hat{l}$ donde la i-ésima fila contiene a la cadena i-ésima con espacios insertados. La suma por pares de la diferencia para una Alineación Múltiple es la suma de las distancias para todo par de hileras en la alineación. Por lo que el problema es encontrar un mínimo en la alineación (Elías, 2003).

$$\text{mín}\, z : \sum_{\substack{a \neq b \\ a,b \in mathboxF}}^{x} \sum_{j=1}^{\hat{l}} d(S_{a,j}, S_{b,j}) \tag{1.7}$$

donde: x: Número de secuencias. $\hat{l}$: Longitud de las secuencias después de insertar las celdas vacías. $d(S_{a,j}, S_{b,j})$: Distancia entre un par de secuencias.

Función objetivo de la suma por pares de las diferencias.

$$\text{mín}\, z : \sum_{\substack{a=1 \\ a \neq b}}^{x} \sum_{b=1}^{x} d(S_a, S_b) \tag{1.8}$$

La suma de las parejas de un Alineamiento Múltiple es la suma de las puntuaciones de todos los pares implicados, es decir, es la suma de las celdas de la matriz de distancia excluyendo los elementos de la diagonal

.

Aproximación por templetes. Utiliza la idea de subcadenas comunes a todas las secuencias de un conjunto, las cuales se han preservado a través del proceso evolutivo de los organismos (Mora-Gutierrez, 2009).

Alineamento por consenso.

Es un problema NP-Duro para cualquier esquema arbitrario y para una función objetivo donde las diferencias tienen un costo de 1 y las coincidencias un costo de 0 (Elias, 2003). (véase figura 1.7)

Función objetivo para el alineamiento por consenso.

$$\text{mín } z : \sum_{\substack{b=1 \\ S_b \in x}}^{x} D(S_\star, S_b) \tag{1.9}$$

Donde: x: Número de secuencias. $D(s_*, s_b)$: Distancia entre el esquema y una secuencia del conjunto. El esquema debe cumplir con la desigualdad del triángulo que implica que dadas las cadenas S_*, S_a, S_b entonces:

$$d(S_a, S_b) + d(S_a, S_\star) \geq d(S_b, S_\star) \tag{1.10}$$

(Lee y otros, 2007).

	1	2	3	4	5	6	7	8	9	10
S_1	Y	D	G	G	A	V	-	E	A	L
S_2	Y	D	G	G	-	-	-	E	A	L
S_3	F	E	G	G	I	L	V	E	A	L
S_4	F	D	-	G	I	L	V	Q	A	V
S_5	Y	E	G	G	A	V	V	Q	A	L
	y	d	G	G	A/I	V/L	V	e	A	I

Figura 1.7: Ejemplo de Alineamiento por consenso

Alineamiento por secuencia media. Uno de los principales problemas en la construcción del esquema exacto para un conjunto de secuencias, se debe a la exigencia de examinar todas las secuencias de la familia de interés. Para evitar lo anterior, se utiliza la secuencia media que es una cadena de la familia y que satisface la desigualdad del triángulo, además de minimizar la distancia entre las secuencias (Mora-Gutierrez, 2009).

Secuencia media. Dada una familia de secuencias $N = [S_1, S_2, \ldots . S_x]$ la secuencia media es aquella $S_c \in N$ que satisface la siguiente relación:

$$\text{mín } z : \sum_{\substack{b=1 \\ S_c \in N \setminus s_b}}^{x} D(S_c, S_b) \tag{1.11}$$

Donde: $D(S_c, S_b)$: Distancia entre un esquema arbitrario (Secuencia media) y una secuencia del conjunto. x: Número de secuencias.

Dado que la secuencia media es un esquema arbitrario del conjunto, esta función objetivo induce a un problema NP-Duro.

Existe un problema con el alineamiento por secuencia media, ya que al imponer un esquema arbitrario a todas las secuencias no puede garantizarse el alineamiento óptimo.

Error por secuencia media.

$$\frac{V(M_c)}{V(M^*)} \leq \frac{2(x-1)}{x-2} \tag{1.12}$$

Donde: $V(M_c)$: Valor del alineamiento por secuencia media. $V(M^*)$: Valor óptimo del alineamiento. x: Número de secuencia del conjunto que debe ser mayor a 2.

Restricciones del problema.

En la definición del alineamiento, distancia y valor de alineamiento se incluyen restricciones explicitas para las secuencias del conjunto, las cuales son:

1.- Todas las secuencias que se van a alinear deben pertenecer a una familia (Definición de familia de secuencias)

$$\forall a \ S_a \in N \ \forall S_a = 1, 2, \ldots x \tag{1.13}$$

2.- La longitud de las cadenas después de agregar los espacios vacíos debe ser la misma para todas las que integren la matriz M.

$$\forall \hat{S_a} \, y \, \hat{S_b} \ \in M \ \hat{l_a} = \hat{b_b} \tag{1.14}$$

3.- La distancia entre cualquier par de secuencias es no negativa.

$$\forall S_a \, y \, S_b \ d(S_a, S_b) \geq 0 \tag{1.15}$$

4.- La distancia entre cualquier par de secuencias es simétrica.

$$\forall S_a \, y \, S_b \ d(S_a, S_b) = d(S_b, S_a) \tag{1.16}$$

5.- La distancia para dos secuencias iguales es 0, si y solo si ambas secuencias son idénticas.

$$d(S_a, S_b) = 0 \leftrightarrow S_a = S_b \tag{1.17}$$

(Lee y otros, 2007).

6.- Al seleccionar la distancia entre dos secuencias debe satisfacerse la desigualdad del triángulo.

$$\forall S_a, S_b \, y \, S_c \ d(S_a, S_b) + d(S_b, S_c) \geq d(S_a, S_c) \tag{1.18}$$

7.- El procedimiento para determinar la métrica de distancias dentro de un experimento, debe ser el mismo para todas las secuencias.

Pueden agregarse restricciones que representen los conocimientos biológicos con que se cuenten, es decir, si se conoce que dentro del conjunto existen subconjuntos estrechamente relacionados biológicamente (géneros, especies, etc.), se imponen restricciones de pertenencia a subgrupos. Con lo anterior se consigue acotar mejor el universo de posibilidades.

Capítulo 2
Estado del arte del Alineamiento Múltiple de Secuencias.

Tal como se citó en el capítulo anterior, se han desarrollado varios métodos de solución del Alineamiento Múltiple de Secuencias y se ha dicho que conforme aumenta el número de secuencias ,crece el tiempo de procesamiento del alineamiento con los métodos exactos.

El presente capítulo describe el estado del arte del alineamiento múltiple de secuencias, lo cual permitirá dar a conocer los esfuerzos que los expertos en la materia han hecho por resolver este problema.

Se iniciará con una breve revisión de los métodos de optimización para dos secuencias ya que los métodos heurísticos empleados se basan en los métodos de comparación de dos secuencias como un proceso elemental (Chan y otros, 1992).

2.1. Comparación de dos secuencias por programación dinámica.

(Waterman, 1984) y (Davison, 1985) presentaron una revisión de los métodos de comparación de dos secuencias. El método más comunmente empleado es el de aproximación matemática mediante programación dinámica(Chan y otros, 1992). Generalmente, dichos métodos tienen un tiempo de procesamiento de O(nm) donde n y m son las longitudes de las secuencias a ser comparadas (Chan y otros, 1992).

A continuación un cuadro con las aportaciones más representativas:

Otros autores también contribuyeron , como (Sankoff, 1972), (Sankoff y Sellers, 1973), (Reichert y otros, 1973), (Wong y otros, 1974), (Cohen y otros, 1975), (Waterman y otros, 1976), (Gotoh, 1982), (Taylor, 1984) (véase figura 2.1).

2.2. Métodos de Comparación de Secuencias Múltiples.

A continuación se describirán los métodos desarrollados para la Alineación múltiple de secuencias, de acuerdo a la forma que se emplea para su solución (véase figura 2.2).

Autor	Año	Característica	Complejidad Computacional.
Needlean & Wunsch	1970	Maximiza el número de coincidencias menos el número de inserciones o eliminaciones.	
Sellers	1974	Este algortimo asigna pesos no negativos a las sustituciones, inserciones o eliminaciones y considera el alineamiento óptimo como la suma	
Fitch & Smith	1983	Proponen que en un alineamiento óptimo debe pesar tanto el gap como la longitud (y no que el peso de las penalizaciones de un gap sean	Para Walterman $O(nm^2+n^2m)$, para Gotoh O(nm).
Fickett	1984	Construye solo una banda (alrededor de la diagonal principal) de la matriz de la programación dinámica en lugar de la	
Altschul & Erickson	1986	Mejoró el tiempo de Gotoh.	O(nm)
Millers & Myers	1988	Proponen una extensión del algoritmo de Waterman.	$O(n^2 logn)$

Figura 2.1: Métodos de programación dinámica en la solución del AMS

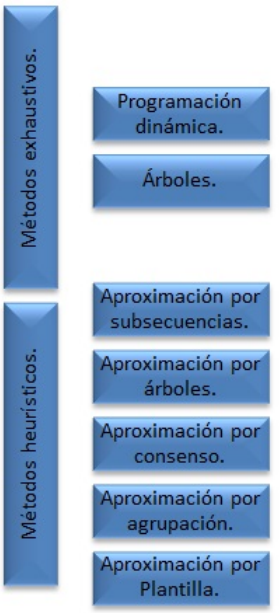

Figura 2.2: Estado del arte de los Métodos de solución del AMS

2.2.1. Métodos Exhaustivos.

Los métodos exhaustivos de comparación de secuencias múltiples garantizan un alineamiento óptimo. Algunos de ellos usan heurísticas para limitar la búsqueda de tal alineamiento.

2.2.1.1. Aproximación por Programación Dinámica.

Esta aproximación se refiere a la comparación simultánea de N secuencias usando matrices N-dimensionales de programación dinámica (Waterman y otros, 1976). Por ejemplo, el algoritmo (Needleman y Wunsch, 1970) se extiende directamente a la comparación de tres secuencias usando una matriz tridimensional (Jue y otros, 1980) . (Murata y otros, 1985) que reduce el tiempo de procesamiento de Needleman y Wunsch de $O(n^5)$ a $O(n^3)$ al igual que (Freedman, 1984). Ambos algoritmos usan gaps de peso (penalizaciones) que son independientes de la longitud del gap. Esta desventaja es erradicada por (Gotoh, 1982) usando funciones de pesos de gaps lineales obteniendo el mismo tiempo de procesamiento (Chan y otros, 1992).

El algoritmo (Murata y otros, 1985) y (Gotoh, 1982) explícitamente especifican que los pesos de las comparaciones simultáneas de los tres residuos es la suma de los pesos de las comparaciones de pares de los tres residuos. Tal criterio se extiende para evaluar el alineamiento de N secuencias, por ejemplo, el costo de alinear N secuencias es la suma de los costos de alineación de $N(N-1)/2$ pares de secuencias. Esa medida de costos es llamado Suma de pares (Lipman y otros, 1989).

El algoritmo de (Fickett, 1984) para dos secuencias, busca la ruta óptima con una banda de la matriz bidimensional definida por un límite superior de costo en la ruta óptima. Posteriomente, basado en la medición de suma de pares, (Carrillo y Lipman, 1988) propusieron una estrategia para la alineación de N secuencias similar al algoritmo de (Chan y otros, 1992).

El algoritmo de (Lipman y otros, 1989) adopta las estrategias de Carrillo y Lipman usando un procedimiento heurístico para obtener un alineamiento inicial basado en que el límite superior de un costo de alineamiento de pares se determina previamente. De manera alternativa, los usuarios pueden especificar cualquier conjunto de límites. El algoritomo tiene dos características especiales. Uno es el uso de un costo de gap quasinatual como lo propuso (Altschul, 1989), y el otro es la opción de usar o la suma de pares ponderada o un alineamiento de pares sin ponderar (Chan y otros, 1992).

En la suma de pares ponderados, diferentes pesos se asignan a los costos de alineamiento de pares, siguiendo los métodos propuestos por (Altschul, 1989). El propósito es descontar la dominancia de un conjunto de secuencias muy similares en el alineamiento múltiple de secuencias. El algoritmo de Lipman puede alinear de seis a ocho secuencias de 200 o 300 residuos con mejores resultados que los tres métodos antes mencionados(Chan y otros, 1992).

2.2.1.2. Aproximación de árbol.

Un ensamble de secuencias está conformado por secuencias de entrada en un árbol de N nodos terminales. Ahora bien, cuando se cuenta con el árbol que describe la relación ancestral entre dicho ensamble, se puede obtener una aproximación realista al problema de alineamiento

construyendo la secuencia ancestral (en los M nodos internos del árbol) y entonces alinear las secuencias de entrada y las secuencias ancestrales siguiendo las relaciones de incidencia (topología) del árbol).

Si un costo es asignado a cada rama del árbol la cual es igual a la distancia entre las dos secuencias de las dos terminaciones de las ramas, entonces el problema de alineamiento se convierte en un problema que busca encontrar el costo mínimo (suma total mínima de los costos de ramificaciones) del árbol (Fitch, 1971), (Hartigan, 1973)). Un algoritmo que reconstruye las secuencias ancestrales y da un alineamiento relacionando las $N + M$ secuencias basadas en la minimización del costo de los árboles, se desarrolló por (Sankoff, 1975). Desafortunadamente tiene una complejidad de $O(M(2n)^N)$ donde n es la longitud de las secuencias (Chan y otros, 1992).

La estrategia de (Carrillo y Lipman, 1988) está basada en la evaluación de los costos de alineamiento por la suma de pares. Esta estrategia fue examinada por (Altschul y Lipman, 1989) para las aplicaciones al caso donde un alineamiento es evaluado por el costo del árbol. En tales aplicaciones, la estrategia involucra el problema de encontrar un conjunto de pesos no negativos que optimizarían los enlaces de costos del alineamiento de pares (Chan y otros, 1992).

2.2.2. Métodos Heurísticos.

Los métodos heurísticos pretenden encontrar, en un tiempo razonable, buenos alineamientos que no necesariamente sean los óptimos. Los métodos heurísticos existentes de comparación de secuencias múltiples pueden ser clasificados en cinco diferentes aproximaciones:

1. Aproximación por subsecuencias
2. Aproximación de árbol.
3. Aproximación por secuencia de concenso.
4. Aproximación por agrupación.
5. Aproximación por plantilla.

2.2.2.1. Aproximación por subsecuencias.

Una subsecuencia se refiere a un segmento (Johnson y Doolitle, 1986), (Bacon y Anderson, 1986),), una región (Martínez, 1983), o un patrón de consenso (Waterman, 1984) . Cuatro diferentes métodos se describen a continuación:

1.- (Johnson y Doolitle, 1986) proponen un método para la alineación de tres o más secuencias de proteínas por la comparación de segmentos seleccionados de las secuencias. Al inicio se establece una ventana para limitar el número de comparaciones de segmentos. La ventana entonces se mueve a la posición actual tal que se compara el nuevo segmento y se alinean los residuos siguientes (Chan y otros, 1992).

Si se pretendiera la alineación de tres secuecnias, la calificación de las comparaciones simultáneas de los tres segmentos es la suma de las tres calificaciones obtenidas de las comparaciones de pares de los segmentos. Se resta una penalización de la calificación cuando todos los tres segmentos no son contiguos. Esta aproximación se puede extender a la comparación de cuatro o cinco secuencias. La complejidad del tiempo es $O(N(n-W)W_{N-1})$, donde N es el número de las secuencias a comparar y n es la longitud de las secuencias (Waterman, 1986) (Chan y otros, 1992).

2.- Este método define una region R que se representa por una triple indicación (w, i, j) donde hay una palabra de coincidencia w que empieza en la posición i de una secuencia y en la posición j de otra secuencia. Para obtener una lista de regiones de coincidencia para la comparación de dos secuencias, se puede concatenar las dos secuencias en una secuencia simple S y entonces ordenar S según la repetición de palabras. Para encontrar regiones de manera más rápida, se desarrollaron métodos basados en el concepto de desmenuzamiento (o hashing, en inglés) (Dumey, 1956), (Duman y Ninio, 1982)). También puede usarse el algoritmo de (Karlin y otros, 1983) para encontrar repeticiones exactas en un conjunto de secuencias. Dada una lista de regiones, el alineamiento óptimo de dos secuencias se obtiene reuniendo las piezas de la mejor region tal que el peso total de los elementos de no coincidencia entre dos regiones sucesivas sea el mínimo (Martínez, 1983). Algoritmos generales son conocidos para correr en $O(L^2)$ tiempo donde L denta la lista de regiones (Waterman, 1984). El método de alineación de dos secuencias se puede extender al alineamiento de secuencias múltiples. El programa implementado se llama MALIGN que puede también encontrar los alineamientos cercanos al óptimo y provee una media para aleatorizar las secuencias dadas para probar la significancia estadística de una alineación (Sobel y Martínez, 1986). Sin embargo, esta aproximación es demasiado rigida ya que la misma palabra debería ser encontrada en todas las secuencias (Chan y otros, 1992).

3.- (Waterman, 1984) propone encontrar patrones de concenso que ocurran imperfectamente sobre una frecuencia preestablecida. Un patrón de concenso es una k-letra ($K \geq 2$) que es común en por lo menos un porcentaje preestablecido (β) de secuencias. La complejidad de este método es $O(NW^2nB)$, donde N es el número de secuencias, n es la longitud de la secuencia y B es una función de 4^k en el caso de ácidos nucléicos ó 20^4 en el caso de proteínas. En la última mejora, se consideran gaps en las palabra (Waterman y Jones, 1990) (Chan y otros, 1992).

4.- (Bacon y Anderson, 1986) proponen un algoritmo basado en la alineación de los segmentos donde se juzga la significancia de las calificaciones de los alineamientos usando diferentes modelos estadísticos. La similitud entre un par de segmentos se define aquí como la suma de las similitudes individuales de los residuos en las posiciones relativas correspondientes al inicio de cada segmento. El número de posibles alineaciones de segmentos es muy grande para tres secuencias, y para mejorar la velocidad, se determinan dos variables con el fin de almacenar la historia de los detalles. Una pregunta concerniente a este algoritmo es si el almacenamiento de tan pocas alineaciones guiaría la omisión de mejores alineamientos que sean débiles al principio pero que sean fuertes en las siguientes. Otra limitación es que los gaps no son considerados ya que los gaps incrementarían sustancialmente el número de posibles alineamientos y complicarían la representación de ellos. Además, este método no produce un alineamiento completo (Chan y otros, 1992).

2.2.2.2. Aproximación por árboles.

Un árbol es una gráfica acílica cuyas hojas representan un conjunto de muestras y la relación de incidencia de los gaps representa la taxonomía o las relaciones filogenéticas entre las muestras. La búsqueda de los alineamientos óptimos, haciendo uso de la informacion de un árbol, aún requiere tiempo exponencial (Sankoff, 1975). Para esto, varias heurísticas se proponen para obtener alineamientos en tiempo razonable incluso basados en el uso de un árbol. Tres diferentes métodos se describen a continuación:

1.- (Sankoff y Cedergren, 1983) dieron una heurística para alinear N secuencias que estén relacionadas por un árbol filogenético representando la historia evolutiva de las secuencias. Tomando como punto de partida del estudio de (Sankoff, 1975), ahora él mismo y (Sankoff y Cedergren, 1983) mejoraron la rapidez descomponiendo el árbol en conjuntos de árboles sobrepuestos para encontrar el alineamiento en tiempo razonable por repetición de la aplicación N=3 del método exahustivo a los subárboles en un orden apropiado (Chan y otros, 1992).

2.- Si la secuencia está relacionada a un árbol binario, el método heurísico basado en (Waterman y Perlwitz, 1984) puede alinear N secuencias de longitud n en tiempo $O(Nn^2)$. El método inicia con la construcción de una secuencia promedio apartir de dos secuencias originales de entrada que estén relacionadas por un nodo del árbol. Se asigna un peso basado en el número de secuencias originales que deriva de una secuencia contruída. El proceso de construcción de la secuencia continúa siguiendo las relaciones de incidencia del árbol sobre la ruta donde una secuencia promedio final se deriva. El alineamiento promedio se obtiene entonces por alineación de cada secuencia original con la secuencia promedio final (Chan y otros, 1992).

3.- (Hein, 1989) introdujo el concepto de secuencia gráfica al alineamiento de secuencias múltiples y la reconstrucción de secuencias ancestrales cuando se dá la filogenética de las secuencias de entrada . Sin embargo la filogenia se restringe a árboles que incluyen solo bifurcaciones (Chan y otros, 1992).

2.2.2.3. Aproximación de secuencias de concenso.

Cuatro diferentes métodos se incluyen en esta aproximación. La derivación de las secuencias de concenso de un conjunto de secuencias se presenta a continuación:

1.- Cuando se da un alineamiento de un grupo de secuencias relacionadas se puede usar este método que es subjetivo. (Patthy, 1987) simula esta aproximación por un procedimiento controlado de alineamiento iterativo que pondera características clave de una familia de proteínas y esto forza el alineamiento de residuos conservados. El procedimeinto básicamente determina secuencias de consenso que incorpora características a las secuencias relacionadas. La secuencia de consenso se determina como sigue:

En un alineamiento de un grupo de secuencias, se asigna una letra x a la columna donde haya aminoácidos presentes en más de k fracciones de la secuencia mientras que se asigna g a la columna donde el aminoácido este presente en k o menos que k fracciones de la secuencia (Chan y otros, 1992).

2.- Multan (Bains, 1986) es un programa que puede alinear un número largo de secuencias de ácidos nucléicos. Primero, una de las secuencias es escogida como la secuencia de consenso

original. Las otras secuencias se alinean una a una con esta secuencia de consenso para generar un alineamiento basado en la creación de una nueva secuencia de consenso. Este proceso es aceptable, pero (Bains, 1986) reporta que en algunos casos se bloquea por situaciones cíclicas, esto sucede cuando un largo número de secuencias fallidamente divergentes o un menor número de secuencias muy divergentes son alineadas (Chan y otros, 1992).

3.- SEQCMP (Krishnan y otros, 1986) es un programa que encuentra una secuencia consenso para un conjunto de secuencias de ácidos nucléicos. El principio empleado es la generación de una matriz de puntos para cada par de secuencias y todas las matrices punto generadas se superponen en otra para identificar la matriz punto que tienen en común y, de ello, se puede obtener la secuencia consenso. Este proceso puede representar un problema de almacenaje. Pero, cuando las secuencias son de diferentes longitudes, se dificulta obtener la secuencia consenso (Chan y otros, 1992).

4.- El método de (Higgins y Sharp, 1988) pretende ser un mejor método mediante la calificación del alineamiento de pares de secuencias usando el algoritmo de (Wilbur y Lipman, 1984) , la construcción de un árbol filogenético basado en el método UPGMA (un procedimiento de agrupamiento jerárquico que usa la medición de distancia agrupada-promedio como una estrategia de un ordenamiento) (Sneath y Sokal, 1973) y finalmente un alineamiento de secuencias de árboles usando como base el árbol del paso 2 (Chan y otros, 1992).

2.2.2.4. Aproximación por Agrupación.

Esta aproximación pretende construir un árbol filogenético para la alineación de las secuencias u ordenar las secuencias en un orden particular basado en cuáles de las secuencias se alinean una por una. Los métodos de (Patthy, 1987) y (Higgins y Sharp, 1988) incluyen esta aproximación. Ocho métodos se citan a continuación:

1.- GENALIGN (Martínez, 1988) es un programa para alineamiento múltiple de secuencias evolucionado a partir de MALIGN (Sobel y Martínez, 1986).

GENALIGN tiene la opción de encontrar los alineamientos de pares ya sea por un nuevo método que es considerado como una mejora en el método de regiones (Martínez, 1983) o por el algoritmo de (Needleman y Wunsch, 1970). En el nuevo método basado en regiones, un primer alineamiento se basa en una región de longitud específica mínima. El alineamiento se redefine entonces reduciendo recursivamente tal región (Chan y otros, 1992).

2.-Pueden alcanzarse alineamientos eficientes de secuencias multiples cuando un árbol filogenético relacionado con las secuencias está disponible. (Hogeweg y Hesper, 1984) argumentan que el alineamiento de conjuntos de secuencias y la construcción de los árboles filogenéticos pueden ser tratados conjuntamente. En su método, un árbol es usado para alinear secuencias y los alinemiamientos obtenidos se usan para ajustar el árbol. Muchos cuestionamientos existen acerca de este método. Por ejemplo, no está claro si se alcanza una convergencia o un árbol/alineamiento estable (Chan y otros, 1992).

3.- El método de (Barton y Sternberg, 1987) ordena las secuencias en una cadena específica y posteriomente las demás secuencias se alinean una por una. (Chan y otros, 1992).

4.- El método de (Feng y Doolitle, 1987) es, de hecho, un método de construcción de árbol basado en el alineamiento progresivo de las secuencias. Este comienza con el alineamiento de dos secuencias, sea A y B, las cuales tienen la calificación de similitud más alta. Si C es

más cercano a AB, entonces se evaluan la calificación del alineamiento basado en el orden ABC y la calificación del alineamiento basado en el orden BAC. Posteriormente se escoge el alineamiento que tiene la calificación más alta. Si ABC tiene calificacion más alta y D está cerca de él, entonces el alineamiento basado en el orden $ABCD$ se compara con el alineamiento basado en el orden $ABDC$ y se escoge el que tiene la calificacion más alta. Notar que C y D pueden ser un subgrupo de secuencias prealineadas por un procedimiento simple. Desafortumadamente, no está claro cómo se compara un subgrupo de secuencias y cómo se alinea con otro subgrupo de secuencias (Chan y otros, 1992).

5.- El método de (Subbiah y Harrison, 1989) usa un proceso cíclico para afinar el alineamiento de un ensamble de secuencias. Esto básicamente desplaza el ensamble en varios pares de subgrupos de secuencias y entonces compara cada subgrupo con sus complementos. Este método emplea un tiempo computacional de $O(N!)$ y $N = 10$ ya es considerado alto para ello (Chan y otros, 1992).

6.- El método de (Taylor, 1988) intenta encontrar un buen alineamiento promedio y un buen tiempo en el que se lleve a cabo. Construye un árbol usando un procedimiento que es similar al (Feng y Doolitle, 1987) y entonces alinea las secuencias siguiendo la topología de un árbol. Sin embargo, requiere que la comparación entre dos alineamientos se lleve a cabo comparando una columna de un alineamiento con una columna de otro alineamiento usando programación dinámica (Chan y otros, 1992).

7.- El método de (Corpet, 1988) adopta el procedimiento de agrupamiento jerárquico para construir un árbol para el alineamiento de un ensamble de secuencias de proteínas basado en la calificación de similitud máxima entre las secuencias o grupos de secuencias. La comparación entre dos grupos de secuencias alineadas se lleva a cabo por programación dinámica tal como lo hace (Taylor, 1988) excepto que la calificación promedio de similitud la usa en vez del peso promedio (Chan y otros, 1992).

8.- (You, 1983), (Wong, 1987) y (Wong, 1990) proponen el uso del procedimiento de agrupamiento jerárquico para obtener el alineamiento de un ensamble de secuencias. Sus conceptos se originan de la aproximacion aleatoria gráfica para reconocimiento de estructuras patrones (You, 1983), (Wong y You, 1985) (Chan y otros, 1992).

2.2.2.5. Aproximación por Plantilla.

Una plantilla (o template, en inglés) es una secuencia de concenso de un segmento de alineamiento que corresponde a una parte de la estuctura secundaria de la molécula. Las secuencias relacionadas son alineadas con las plantillas o son incluídas en el alineamiento original una por una. Las plantillas son modificadas para incluir la variedad de residuos en estas secuencias durante el proceso de alineamiento. La incorporación de relaciones jerárquicas entre los elementos de la secuencia primaria, los elementos de alto orden y los elementos más espaciados en una descripcion de patrones, permiten expresar posteriormente la estructura de una secuencia de máximo orden(super-secundario). ARIADNE (Webster y otros, 1987); (Lathrop y otros, 1987)) es un sistema experto para la inferencia de estructuras similares de orden alto, dada una descripción patrón y una predicción de estructura secundaria de las secuencias.

Un perfil es una tabla de calificación de posiciones específicas generada de un grupo de

secuencias alineadas que se basa en la tabla de comparación dando la calificación de las comparaciones entre dos residuos de aminoácidos (Chan y otros, 1992).

Como conclusión de los métodos ya desarrollados, cabe destacar que algunos de los principios que emplean cada uno de ellos sirvieron de base para formular el Algoritmo de Composición Musical aplicado al problema del AMS, sin embargo la complejidad de éste es menor, siendo ella $O(nlogn * l^2)$ (véase figura 2.3).

A continuación se describe el algoritmo principal de Composición Musical.

2.3. Algoritmo de optimización inspirado en la Composición Musical.

Los algoritmos heurísticos son una alternativa atractiva que ha sido usada para encontrar soluciones de alta calidad en los problemas de optimización. Su diseño y adaptación involucra analogías de conceptos como creatividad y conocimiento ,por ejemplo: procesos biológicos y sistemas (algoritmos genéticos), y memoria humana (redes neuronales).

El Método de Composición Musical o MMC es una nueva metaheurística de reciente aparición. Es un algoritmo bioinspirado del que forman parte las sociedades artificiales, definidas éstas como modelos multiagente donde cada agente experimenta el proceso de aprendizaje mediante la interacción con su medio.

El algoritmo de composición musical usa un sistema creativo dinámico para crear una obra musical. Este método simula el proceso creativo de composición musical con agentes (compositores) con la capacidad de crear y cambiar sus obras intercambiando información entre ellos y el ambiente, usando su propio conocimiento para mejorar su trabajo. El diseño de esta metaheurística aplica escencialmente 3 ideas: las principales características de las sociedades artificiales, el proceso creativo de composición musical y la optimización.

Este método fue desarrollado por (Mora-Gutiérrez y otros, 2011) y será empleado en el presente trabajo para resolver el problema de Alineamiento Múltiple de Secuencias. La presente sección describe el algoritmo de Composición Musical.

2.3.1. Antecedentes.

Los algoritmos sociales son un subconjunto de sistemas evolutivos, ejemplos de ellos son: a) Algoritmos culturales (Reynolds., 1994) para el modelaje social evolutivo y el aprendizaje; b) Algoritmo de optimización de colonia de hormigas. (Dorigo y otros, 196) que es una metaheurística inspirada por sistemas naturales de colonias de hormigas reales que simula el comportamiento de éstas para encontrar la ruta más corta a su fuente de comida; c) Sistemas inmunes artificiales que explotan las características de los sistemas inmunes de aprendizaje y memoria para resolver problemas de optimización; d) Algoritmos de Sociedad y civilizaciones (Ray y Liew, 2003) que emplean las interacciones inter e intrasociales en un conjunto de individuos y en un modelo de civilización; etc (Mora-Gutiérrez y otros, 2011).

Una sociedad artificial, como se dijo arriba, está compuesta por varios individuos que interactúan entre sí. Lo que se pretende con ellas es:

Tipo de algoritmo	Nombre del algoritmo	Número de secuencias que alinea simultáneamente	Complejidad.
Exhaustivos.	Matriz de puntos	2	$O(l_a * l_b)$
	Distancia de Levenshtein	2	$O(l_a * l_b)$
	Distancia Indel	2	$O(l_a * l_b)$
	Distancia de Damerau	2	$O(l_a * l_b)$
	Needleman y Wunsch	2	$O(l_a * l_b)$
	Murata	3	$O(l_a * l_b * l_c)$
	Gotoh	2	$O(l_a * l_b)$
	Gotoh y Altschul	2	$O(l_a * l_b)$
	Carrillo y Lipman	Hasta 10	$O(l^n)$
	Smith y Waterman	2	$O(l_a * l_b)$
	Fredman	3	$O(l^3)$
Progresivos.	Sankoff	5	$O(n_i(2l^n)$
	Sankoff y Cedergren	3	$O(n_i(2l^n)$
	Fitch	15	$O(n_i(2l^n)$
	Johnson y Doolitle	3 o más	$O(n(l-l_r)l_r^{n-1})$
	Karlin	2 o más	$O(\|l^2\|)$
	Waterman y Jones	2 o más	$O(n*l_r^2*l*B)$
	Waterman y Perlwitz		$O(nl^2)$
	Barton y Sternberg		$O(n!)$
	Subbiah y Harrison		$O(n!)$
	Clustal		$O(n^2l^2)$
	T-Cofee		$O(n^2l^2)+(O^3)$
Iterativos.	Algoritmo de Viterebi		$O(n^3P)$
	Busqueda armónica con recocido simulado		$O(n^2+l^2)$
Algoritmo Social	Composición Musical		$O(nlogn+l^2)$

Figura 2.3: Comparativo de la complejidad de los algoritmos para solucionar el AMS

1. Permitir el flujo de información entre los agentes.

2. Permitir la actualización de la información que genera cada uno de ellos debido a la decisión de cambiar su información previa e

3. Intercambiar información entre los individuos participantes

todo lo anterior con el fin de mejorar el resultado de cada individuo y encontrar una mejor solución en conjunto.

La composición músical es el proceso artistico de crear e innovar una obra a través de procesos recursivos en un sistema creativo. La creatividad de los compositores resulta entre conexiones de ideas disconjuntas (de Bono 1993) y puede ser producida por momentos de genialidad o de un procesamiento de razonamiento recursivo de un pensamiento, llamado "trabajo duro"(Jacob, 1996). Se puede entender la creatividad distinguiendo dos niveles diferentes: el personal y el socio-cultural (Liu., 2000) (Mora-Gutiérrez y otros, 2011).

Estas ideas pueden ser usadas para modelar, simular o replicar la creatividad usando la computadora. El modelo de creactividad artificial se ha usado en la música desde el inicio del proceso de la composición musical (Jacob, 1996), algunos ejemplos son: Algoritmos genéticos y Composición musical asistida por computadora (Horner y Goldberg, 1991); Genjam (Biles, 1994); Composición y algoritmos genéticos (Jacob, 1995); etc (Mora-Gutiérrez y otros, 2011).

2.3.2. Descripción del método.

El MMC considera la idea de que la composición musical puede ser considerada como un algoritmo , ya que el proceso emplea reglas, principios y un número finito de pasos para crear música original de un estilo particular (Cope, 2000) (Mora-Gutiérrez y otros, 2011).

En el MMC cada solución es llamada tono y es representada por un vector n-dimensional

$$tune = [x_1 x_2 .....x_n] \tag{2.1}$$

La estructura básica es presentada en el algoritmo 1 y consiste de seis pasos:

1. Inicialización del proceso de optimización (desde la línea 1 a la 4).

2. Extracción de información entre los agentes (linea 6 y 7).

3. Generación de un nuevo tono por cada agente (linea 9 y 10).

4. Actualización de la obra de cada agente (de la linea 11 a la 13).

5. Construcción de un conjunto de soluciones (linea 15).

6. Repetición mientras el criterio de paro no es satisfecho (desde la linea 5 a la 16).

(Mora-Gutiérrez y otros, 2011)

(Mora-Gutiérrez y otros, 2011)

Algoritmo 1: Algoritmo basico del MMC

Input: Parámetros del método MMC y la información acerca de las instancias a resolver

Output: Todos los mejores tonos generados por los compositores

Creación de una sociedad artificial con las reglas de interaccion entre agentes.
for *cada individuo en la sociedad* **do**
    Iniciar aleatoriamente una obra(considerando los limites minimo y maximo de cada variable).
end
repeat
    Actualización de la sociedad artificial.
    Intercambio de informacion entre agentes.
    for *cada individuo en la sociedad* **do**
        Actualizacion de la matriz de conocimiento.
        Generacion y evaluacion de un nuevo tono ($x_{\star,new}$)
        if $x_{\star,new}$ *es mejor que el peor tono* ($x_{x-worst}$) *en la obra de cada? individuo* **then**
            Reemplazar $x_{x-worst}$ con $x_{\star,new}$ en la obra
        end
    end
    Construccion del conjunto de soluciones
until *hasta que el criterio de terminacion se satisfaga*;

Tabla 2.1: Características de los parámetros del algoritmo MMC

Descripción	Parámetros
Número máximo de arreglos (máx $\_arrangement$)	máx $\_arrangement \in \mathtt{N}$
Factor de genialidad de innovacion (ifg)	$ifg \in [0 \ , \ 1]$
Factor de genialidad de cambio (cfg)	$cfg \in [0 \ , \ 1]$
Factor de intercambio entre agentes ($fcla$)	$fcla \in [0 \ , \ 1]$
Número de compositores Nc	$Nc \in \mathbb{R} \setminus (-\infty \ , \ 2]$ ya que una sociedad es un grupo de personas que interactuan entre ellas, entonces deben ser al menos dos agentes
Número de acordes que forman la obra Ns	$Ns \in \mathbb{R} \setminus (-\infty \ , \ 3]$ Por definición, la armonía requiere por lo menos tres acordes ejecutandose simultáneamente
s	r

2.3.2.1. Fase de Inicialización del proceso de optimización.

En esta fase se alimentan las caracterísicas de la situación a resolver y el valor de los parámetros usados , los cuales se muestran en la tabla 2.1.

(Mora-Gutiérrez y otros, 2011)

El número de evaluaciones Ne es:

$$Ne = Nc * \text{máx } \_arreglo \tag{2.2}$$

(Mora-Gutiérrez y otros, 2011)

Usando la información alimentada en esta fase el algoritmo genera, para cada compositor, una calificación $P_{\star,\star,i}$ que es usada como memoria. Las calificaciones se generan de manera aleatoria y se estructuran como lo muestra la ecuación:

$$P_{\star,\star,i} = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{Ns,1} & x_{Ns,2} & \dots & x_{Ns,n} \end{pmatrix} \tag{2.3}$$

donde $P_{\star,\star,i}$ es la calificación del $i-esimo$ compositor y $x_{j,l}$ es el $l-esima$ variable de decisión del $j-esimo$ tono (Mora-Gutiérrez y otros, 2011)

Algoritmo 2: Generacion de un conjunto inicial de calificaciones

Input: n, Nc, Ns, x_l^U para todo $l = 1, 2, \ldots, n$ y x_l^L para todo $l = 1, 2, \ldots, n$
Output: $P_{\star,\star,i}$

```
for i = 1 : Nc do
    for j = 1 : Ns do
        for l = 1 : n do
            P_{⋆,⋆,i} = x_l^L + (rand ∗ (x_l^U − x_l^L))
        end
    end
end
```

Para producir las calificaciones para cada agente, se usa el algoritmo 2.
donde:$rand \sim U[0, 1]$ (Mora-Gutiérrez y otros, 2011)

2.3.2.2. Extracción de información entre agentes.

En esta fase, los compositores extraen la información usando la política de interacción: el compositor i intercambia un tono con el compositor k, sí y solo sí, hay un link entre ellos y el peor tono del compositor k es mejor que el peor tono el compositor i. Esta fase se subdivide en dos fases: a)actualización de los links entre los compositores y b) extracción de información (Mora-Gutiérrez y otros, 2011).

Actualización de los links entre los compositores.
El objetivo de esta subfase en generar un cambio en la red social en el tiempo t con respecto a la red en el tiempo $t-1$. Ver fig 2. Para esta actividad, el MMC usa el algoritomo 3.

donde v es el $v - esimo$ arreglo. (Mora-Gutiérrez y otros, 2011)
Después de la actualización de los links entre los compositores , el algoritmo MMC ejecutará la siguiente subfase.

Extracción de información.
El objetivo de esta subfase es proveer a cada compositor de la información proveniente de su ambiente. Para este propósito el algoritmo MMC usa el procedimiento mostrado en Algoritmo 4.
(Mora-Gutiérrez y otros, 2011)

2.3.2.3. Generación de un nuevo tono.

En esta fase cada compositor creará un nuevo tono usando su propio conocimiento. Esta fase se divide en dos subfases: a)Construcción del antecedente y b) La creación de un nuevo tono (Mora-Gutiérrez y otros, 2011)
a) Construcción del antecedente.

Algoritmo 3: Actualizacion de links entre compositores

Input: v, Nc, $fcla$, sociedad artificial previa
Output: Actualización de la sociedad artificial

```
if v = 1 then
    for i = 1 : Nc do
        for k = i + 1 : Nc do
            if rand < 0.5 then
                Creación de un link entre el compositor i y el compositor k .
            end
        end
    end
else
    for i = 1 : Nc do
        if rand < fcla then
            Elección aleatoria de un compositor k , tal que i ≠ k.
            Cambiar la relación entre ambos compositores.
        end
    end
end
Checar que el agente tiene por lo menos un link.
```

Algoritmo 4: Intercambio de informacion del medio ambiente

Input: $P_{\star,\star,i}$, Nc y función objetivo de la instancia ($f(x)$)
Output: Matriz de conocimiento del ambiente del $i-th$ compositor ($SC_{\star,\star,i}$)

```
for i = 1 : Nc do
    x_{i-worst} ← vector con el peor valor de f(x) en P_{⋆,⋆,i}.
    for k = 1 : Nc ∧ k ≠ i do
        x_{k-worst} ← vector con el peor valor de f(x) en P_{⋆,⋆,k}.
        if hay un link entre el compositor i y el compositor k then
            if f(x_{i-worst}) es peor que f(x_{k-worst}) then
                El compositor i aleatoriamente toma un tono de P_{⋆,⋆,k} y adiciona esta
                información a ISC_{⋆,⋆,i}.
            end
        end
    end
end
```

En esta subfase el algoritmo crea una matriz que representa el antecedente para cada compositor $KM_{\star,\star,i}$. Esta matriz contiene el conocimiento del compositor i y la información que él adquirió del medio. El algoritmo MMC que se usa en esta rutina se muestra en el Algoritmo 5.

Algoritmo 5: Construcción y ponderación del conocimiento previo

Input: $P_{\star,\star,i}$, $ISC_{\star,\star,i}$, Nc y $f(x)$
Output: Matriz de conocimiento ponderado del $i-th$ compositor ($fitness(KM_{j',\star,i})$

```
for i = 1 : Nc do
    KM_{⋆,⋆,i} = P_{⋆,⋆,i} ⋃ ISC_{⋆,⋆,i}.
    for k = 1 : Nc ∧ k ≠ i do
        x_{k−worst} ← Vector con el peor valor de f(x) en P_{⋆,⋆,k}.
        a_i = Σ_{j'=i}^{r} f(KM_{j',⋆,i}).
        for j' = 1 : r do
            fitness(KM_{j',⋆,i}) = (a_i − f(KM_{j',⋆,i})) / (a_i * (Nc − 1)).
        end
    end
end
```

(Mora-Gutiérrez y otros, 2011)

b) Creación de un nuevo tono. En esta subfase cada compositor creará un nuevo tono usando sus antecedentes y sus ideas innovadoras. El algoritmo MMC usado se muestra en el Algoritmo 6.

(Mora-Gutiérrez y otros, 2011)

2.3.2.4. Actualización de la obra de cada agente.

Actualización del $P_{\star,\star,i}$. En esta fase, cada compositor decide si reemplaza el peor tono de su memoria para generar un nuevo tono, esta decisión se basa en el valor de la función objetivo obtenida. El algoritmo empleado es el mostrado en el algoritmo 7.

(Mora-Gutiérrez y otros, 2011)

2.3.2.5. Construcción de un conjunto de soluciones.

En esta fase el MMC tma el mejor tono de cada compositor. El algoritmo se muestra en el Algoritmo 8.

(Mora-Gutiérrez y otros, 2011)

En la siguiente sección se presenta una breve descripción de la base de datos Balibase.

Balibase es una herramienta que en el presente trabajo fue empleada como fuente de datos para la validar la eficiencia de la aplicación del método de Composición Musical en la solución del Alineamiento Múltiple de Secuencias. Balibase proprociona una lista de secuencias biológicas y los resultados de evaluarlas con diferentes métodos.

Algoritmo 6: Creación de un nuevo tono

Input: $KM_{\star,\star,i}$, ifg, n, Nc, $f(x)$, $fitness(KM_{j',\star,i})$, x_l^U para todo $l = 1, 2, \ldots, n$ y x_l^L para todo $l = 1, 2, \ldots, n$

Output: Un nuevo tono $(x_{\star,new})$

```
for i = 1 : Nc do
    if rand < (1 − ifg) then
        for l = 1 : n do
            x_l^max ← maximo valor de x_l en KM_⋆,l,i.
            x_l^min ← minimo valor de x_l en KM_⋆,l,i.
            KM_j,l,i ← toma aleatoriamente el j tono de KM_⋆,l,i, considerando
            fitness(KM_j,⋆,i).
            KM_j',l,i ← toma aleatoriamente el j' tono de KM_⋆,l,i, considerando
            fitness(KM_j',⋆,i).
            if rand < (1 − cfg) then
                x_l,new = KM_j,l,i + (rand ∗ (KM_j',l,i − KM_j,l,i)).
            else
                if rand < 0.5 then
                    x_l,new = x_l^min + (rand ∗ (KM_j,l,i − x_l^min)).
                else
                    x_l,new = x_l^max − (rand ∗ (x_l^max − KM_j,l,i)).
                end
            end
        end
    else
        for l = 1 : n do
            x_l,new = x_l^U − (rand ∗ (x_l^U − x_l^L))
        end
    end
    Determinación del valor de la función objetivo de x_⋆,new (f(x_⋆,new)).
end
```

Algoritmo 7: Actualización de la obra del $i - th$ compositor

Input: $P_{\star,\star,i}$, $f(x)$ y Nc

Output: matriz $P_{\star,\star,i}$ actualizada

```
for i = 1 : Nc do
    x_x−worst ← elemento en P_⋆,⋆,i con el peor valor del fitness.
    if f(x_x−worst) es peor que f(x_⋆,new) then
        Reemplazar x_x−worst con x_⋆,new en P_⋆,⋆,i .
    end
end
```

Algoritmo 8: Construcción del conjunto de soluciones

Input: $P_{\star,\star,i}$, $f(x)$ y Nc

Output: Conjunto con la mejor solución encontrada por los compositores $(S\star, \star)$

```
for i = 1 : Nc do
    Si, ⋆ ← elemento en P_{⋆,⋆,i} con el mejor valor de f(x).
end
```

2.4. Balibase.

Balibase es una base de datos de alineamientos múltiples de secuencias, específicamente diseñada para la evaluación y comparación de programas que resuelven este problema. Los alineamientos estan clasificados por longitud de secuencia, similitud y presencia de inserciones y extensiones N/C terminales. (de la pagina web BALIBASE I).

Actualmente existen tres versiones de BALIBASE. A continuación se citan las caraterísticas de cada una de ellas.

BALIBASE ver 1. Es una colección de 142 alineamientos protéicos referenciales, que contiene más de 1000 secuencias (Notredame 2000; Thomson, Plewnlak y Poch 1999 and Wong y Li 2004).

Los alineamientos están divididos en cuatro conjuntos de referencias jerárquicas. La referencia 1 provee las bases para la construcción de los conjuntos posteriores. El conjunto de referencias de BALIBASE se clasifican en :

Referencia 1. Contiene un conjunto de alineamientos de secuencias equidistantes (por lo menos 6), esto es, tienen una longitud similar sin extensiones o inserciones largas.

Referencia 2. Alinea por arriba de tres secuencias (identicas en menos del 25 por ciento) de la referencia 1 con una familia de por lo menos 15 secuencias cercanamente relacionadas.

Referencia 3. Contiene arriba de 4 subgrupos con menos del 25 por ciento de similitud de sus residuos entre las secuencias de los diferentes grupos. Los alineamientos son construídos adicionando miembros de famlias homólogas a la secuencia relacionada con mayor distancia en la referencia 1.

Referencia 4. Está dividida en dos subcategorias conteniendo alineamientos de por encima de 20 secuencias incluyendo extensiones N/C terminales de arriba de 400 residuos , e inserciones de arriba de 100 residuos.

BALIBASE ver 2. Esa version separa la referencia 4 en dos: a)una incluyendo solo las extensiones N/C terminales y b)incluyendo insersiones internas.

BALIBASE 2.0 incluyen tres conjuntos de referencias (6 a 8) que conteniendo 26 familias de proteínas , en esta versión se representan 1100 secuencias. Las caraterísticas de las secuencias agregadas son:

Referencia 6. Contiene un conjunto de alineamientos proteícos construidos a partir de secuencias con alto grado de regiones repetidas.

Referencia 7. Incluye casos de alineamientos de proteínas trans membranas.

Referencia 8. Colección de alineamientos caracterizados por la permitación circular de las secuencias.

BALIBASE ver 3.

En esta versión se incluye un nuevo conjunto denominado referencia 9. En este se incluyen tres subconjuntos cuyas característica es la existencia de motivos lineales en las secuencias del alineamiento. (Perrodou, Chica, Poch, Gibson and Thompson,2000).

A continuación se describen los métodos por medio de los cuales se determinará la eficiencia de la aplicación del algoritmo de Composición Musical contra los otros métodos de los que BALIBASE ofrece información.

2.5. Método Wilcoxon y Bootstrap.

Método Bootstrap.

Cuando se presenta el valor de una estimación puntual, suele ser necesario dar alguna idea de su precisión.El error estándar es la medida de precisión más usal. Si $\hat{\theta}$ es un estimador de θ, el error estándar de $\hat{\theta}$ es justamente la desviación estándar de $\hat{\theta}$, ó

$$\sigma_{\hat{\theta}} = \sqrt{V(\hat{\theta})} \tag{2.4}$$

Si $\sigma_{\hat{\theta}}$ involucra cualesquiera parámetros desconocidos, entonces si sustituimos estimaciones de estos parámetros en la ecuación anterior, obtendremos el error estándar de $\hat{\theta}$, digamos $\hat{\sigma}_{\hat{\theta}}$. Un error estándar pequeño implica que se ha presentado na estimación relativamente precisa.

Cuando la distribución de $\hat{\theta}$ es desconocida o complicada, puede ser difícil estimar el error estándar de $\hat{\theta}$ usando la teoría de la estadística estándar. En este caso, se puede usar una técnica intensiva de cálculo llamada Bootstrap.

Suponga que el error estándar de $\hat{\theta}$ se denota por $\sigma_{\hat{\theta}}$. Además suponga que la función de densidad de probabilidad está dada por $f(x;\theta)$. A partir de estos datos, se puede construir fácilmente la estimación bootstrap de $\sigma_{\hat{\theta}}$.

1. Dada una muestra aleatoria de $f(x;\hat{\theta})$, $x_1, x_2, ...x_n$, estime θ denotado por $\hat{\theta}$.

2. Usando la estimación $\hat{\theta}$, genere una muestra de tamaño n de la distribución $f(x;\hat{\theta})$. Ésta es la muestra bootstrap.

3. Usando una muestra bootstrap, estime θ. Ésta estimación se denota por $\hat{\theta}_i^*$.

4. Genere muestras bootstrap B para obtener estimaciones bootstrap, $\hat{\theta}_i^*$ para $i = 1, 2, ...B$ (con frecuencia se usa $B = 100o200$).

5. Sea $\overline{\theta^*} = \sum_{i=1}^{B}/B$ la representación de la muestra de las estmaciones botstrap.

6. El error bootstrap estándar de $\overline{\theta^*}$ se encuentra con la fórmula usual de desviación estándar.

$$\S_{\hat{\theta}} \tag{2.5}$$

En los textos sobre estadística, con frecuencia se reemplaza $B - 1$ por B; para valores grandes de B, sin embargo, se obtiene una pequeña diferencia práctica en la estimación (Hines y otros, 2006).

Capítulo 3
Aplicación del Algoritmo de Composición Musical al Problema de Alineamiento Múltiple de Secuencias.

En el presente capítulo se describe el procedimiento propuesto para resolver el problema del AMS empleando la metaheurística de Composición Musical.

Este capítulo muestra también la manera en la que se eligieron los datos que sirvieron de materia prima para el presente trabajo, cómo se validaron los resultados del presente trabajo por la metodología de Julie Thomson y cómo se compararon éstos con los obtenidos por otros métodos.

3.1. Descripción del método.

Actualmente el problema del AMS ha sido tratado por medio de muchos métodos, tanto heurísticos como exactos tal como se ha mostrado en el capítulo anterior.

Ahora bien, el Método de Composición Musical , debido a su reciente aparición, se ha empleado más en la solución de problemas de caracter contínuo, y nunca para resolver el problema del AMS. Cabe destacar que paralelamente a la solución del AMS por medio del algoritmo de Composición Musical, los investigadores creadores de éste, trabajan en otro problema discreto llamado diseño de zonas.

Lo anterior demuestra la importancia que el presente trabajo representa por la aportación de información en lo que a la aplicacion del algoritmo de Composición Musical se refiere, considerando la naturaleza discreta, binaria y restricta del AMS.

Así mismo, se propone esta metaheurística para resolver el problema del AMS ya que es muy costoso aplicar los algoritmos exactos existentes debido a que el problema es de orden exponencial.

Por último es importante resaltar que, para resolver el problema de AMS, se emplearon penalizaciones para la generación de soluciones factibles. Esto comprende la adecuación llevada a cabo, para solucionar el AMS, en comparación con el trabajo inicial del uso del algoritmo de Composición musical en problemas contínuos.

3.1.1. Funcionamiento del método aplicado.

La función objetivo es la minimizar la suma de pares de diferencias considerando las restricciones del problema (ver Capítulo 1, sección Alineamiento y Patrones). Así, la esencia de

la función objetivo es encontrar la máxima similitud (o mínima distancia) entre las secuencias de cada juego con el mínimo de columnas después de inserción de espacios vacíos.

Ahora bien, cada individuo será llamado compositor y se asocia con una matriz de información que funciona como una memoria denominada partitura. Esta memoria está constituida por un conjunto de soluciones llamadas tonos o melodías y al considerar éstas como un todo, conforman una partitura. Así entonces, el compositor n, se asocia a la partitura n. Ahora bien, el resultado final del algoritmo es una partitura global obtenida tomando la mejor melodía de cada compositor (o de cada partitura).

3.1.1.1. Descripción de las etapas del Algoritmo de Composición Musical.

El proceso de Composición Musical está compuesto de varias etapas

.

Inicialización

La primera de ellas es la etapa de inicialización, donde se introducen los parámetros y la instancia a resolver. Dichos parámetros son :

1. Número de compositores.
2. El factor de genialidad de cambio fcg[0,1]. Que es el factor que indica si un cambio se establece entre dos compositores.
3. El factor de genialidad de innovación fgi[0,1]. Que es el factor que indica si hay un cambio en la melodía de un compositor respecto a su memoria previa.
4. El factor de intercambio entre agentes fcla[0,1]. Que es el factor que indica si se realiza un cambio en los vínculos de los compositores (ya que esto influye si se da o no el intercambio de información entre cualesquiera dos compositores)
   .

En esta etapa se crea la partitura inicial de cada compositor, que es una matriz de matrices. Esta matriz está formada de 3 soluciones aleatorias, esto es, tres matrices donde en cada una de ellas se ordenó el mismo conjunto de secuencias a alinear y se insertaron guiones de manera aleatoria (resultando así tres soluciones de la alineación). Éstas matrices pueden tener longitudes diferentes que oscilan entre el valor de la longitud de la secuencia más larga de la familia y la suma de todas las secuencias contenidas en ella.

Posteriormente y por medio del proceso de suma de pares, se evalua la matriz total creada y cada una de las melodías que la conforman. Ello permite asignar una calificación a cada melodía para poder compararlas más adelante

.

Adquisición de conocimiento de los vecinos.

Cada compositor adquiere conocimiento de los demas agentes comparando sus melodías con las de sus vecinos, esto es, ya que se evaluaron éstas por medio de la cuantificación de distancias (método de suma de pares), se pregunta cada compositor si la peor melodía de él es mejor que la peor de cada uno de sus vecinos. Si esto es afirmativo, no adquiere conocimiento de él, de lo contrario incorpora cualquier melodía de la memoria inicial de cada vecino a su

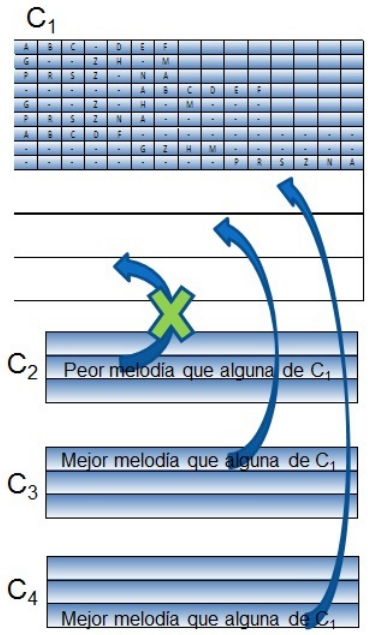

Figura 3.1: Ejemplo de creación de matriz de conocimiento de un compositor.

memoria inicial. De esta manera puede adquirir tantas melodias desde 0, hasta el número máximo de vecinos existentes.

Creación y evaluación de la matriz de conocimimiento de cada compositor.
La matriz de conocimiento de cada compositor se forma uniendo:

1. La partitura del compositor (que se creó como memoria inicialmente de manera aleatoria) y

2. Conocimiento que adquiere de sus vecinos.

(véase figura 3.1).

Posteriormente , se evalua la matriz de conocimiento y como resultado de ésta, se forma una nueva matriz tomando solo tres soluciones (melodías) de ella. Estas soluciones son:

a) La mejor de cada compositor.

b) Una tomada de manera probabilística. Esta se obtiene de una matriz de conocimiento reducida donde se elimina el mejor y, de ella la probabilidad de elegir una solución depende su calidad o aptitud. El detalle de lo anterior se cita a continuación: Se suman por compositor todos los valores obtenidos de cada melodía que conforma su partitura reducida y se pondera cada una de éstas (en términos de porcentaje). Posteriormente su calidad se evalua mediante el método Montecarlo, el cual genera un número aleatorio entre 0 y 1 con distribución uniforme y se compara este valor contra la columna de valores ponderados de suma de pares de las melodías. La melodía elegida como de mejor calidad será aquella que, considerando el

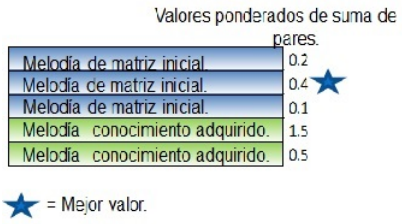

Figura 3.2: Ejemplo de obtención de matrices solución a), b) y c).

acumulado hasta llegar a él de la columna de valores ponderados de las melodías, sea mayor o igual al aleatorio.

c) Una tomada de manera aleatoria. Esta se obtiene NO de una matriz de conocimiento reducida, sino de la matriz completa. En este caso todas las soluciones tienen la misma probablidad de ser elegidas, sin importar su calidad.Esto es, se genra un número aleatorio uniforme discreto entre 1 y el número de melodías de la matriz de conocimiento de cada compositor y se elige la que corresponde a dicho número.

(véase figura 3.2).

Lo anterior permite formar un subespacio conformado por tres puntos que acotan el espacio total para generar una nueva melodía.

Para no quedar sólo en el espacio acotado formado por las tres soluciones arriba citadas, el factor de genialidad de cambio permite abrir las opciones de movimiento libre en una dimensión.

El factor de genialidad de cambio denota cambios puntuales. Por ejemplo: Si este valor es 0.5, la mitad de los puntos en una melodía se seleccionarán en base a la información previa y la otra mitad se generará aleatoriamente.

Ahora bien, el factor de genialidad de innovación es el factor que indica un cambio global , esto es, si se debe o no considerar la información previa usando todas las dimensiones. Indica si es necesario generar la melodía de la nada o no. Por ejemplo: si el valor de éste es 0.5, entonces la mitad de las melodías se generarán aleatoriamente.

Factibilización de la matriz generada (ya acotado el espacio total) por cada compositor.
Para este proceso se propone inicialmente convertir dicha matriz en una matriz binaria según el siguiente criterio:

$$x_{i,j,k,m} = \begin{cases} 1 & \text{si el elemento } i \text{ de la secuencia } j\text{está en la } k-\text{ésima columna}\forall\text{m} \\ 0 & \text{si en caso contrario} \end{cases} \tag{3.1}$$

donde x es cada una de las posiciones ocupadas hasta ahora por letras y m es cada una de las melodías.

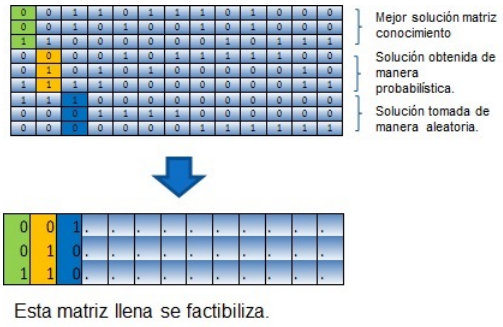

Figura 3.3: Ejemplo de obtención de matriz de cada compositor a factibilizar.

Una vez terminado este proceso se obtiene una sola matriz por compositor, el tamaño será de (número máximo de secuencias)x(número máximo de columnas del alineamiento más largo). Las columnas de la matriz única que se busca, se formarán trayendo los mismos valores de las columnas que se elijan, de manera aleatoria, de las tres matrices obtenidas cuando se buscó acotar el espacio total (véase figura 3.3).

El proceso siguiente, que es la factibilización como tal, se lleva a cabo para comprobar que la longitud de las secuencias sea igual a la suma de los valores binarios de cada hilera. Para ello, se eliminan las columnas que tienen solo 0´s y se agregan 1´s de manera aleatoria en las filas donde la suma de los valores binarios no corresponden al número de elementos (o letras) contenidas en las secuencias iniciales. Todo lo anterior disminuye el tamaño del alineamiento.

Ahora bien, para poder comparar aptitudes del alineamiento obtenido (biológicas, químicas, etc), se convierte la matriz binaria obtenida a una donde sus elementos sean los caracteres de las secuencias originales, claro, ahora posicionándolos el mismo orden, en las ubicaciones donde hay 1´s.

Proceso de fitness

Una vez obtenida la nueva solución (melodía) a partir de las tres soluciones, se compara ésta mediante el proceso de fittnes, en donde se evaluan tres atributos de la nueva solución (evaluación de tres funciones de manera simultánea). Dicha comparación se lleva a cabo poniendo frente a frente la peor del compositor (*Speor*) y la nueva solución (*Snueva*) según lo que dicte el atributo de una matriz de referencia tomada de la literatura.

Lo anterior se realiza por niveles, esto es, se compara el primero de los atributos (grado de alineación según la suma de pares por cada columna) respecto a los dos elementos arriba citados (*Speor* y *Snueva*). Se pregunta si el atributo tiene el mismo valor para ambos, si es así, se pasa al siguiente nivel (comparación del atributo de semejanza biológica) y se hace la misma pregunta. Si tienen el mismo valor se pasa al tercer nivel (atributo de semejanza química). Si en alguno de los niveles de comparación de atributos *Snueva* es mejor que la *Speor*, entonces *Snueva* se agrega a la matriz de conocimiento del compositor y *Speor* se desecha. En caso contrario *Snueva* se desecha y la matriz de conocimiento permanece igual.

Esto se ejecuta un número determinado de veces según el criterio de paro (número de

Nombre del conjunto de Series.	Número de Series que contiene.	Longitud de las series contenidas.
Prueba1aab.	4	67; 71; 74; 79
Prueba1aboA.	5	57; 60; 80; 49; 51
Prueba1ad2.	4	209; 203; 203; 213
Prueba1ad3.	4	424; 447; 442; 433
Prueba1adj.	4	404; 416; 410; 418
Prueba1aho1.	5	65; 66; 61; 67; 61
Prueba1amk.	5	250; 248; 247; 254; 242
Prueba1bbt3.	5	192; 192; 168; 169; 180
Prueba1csp_ref4.	6	67; 69; 72; 67; 711; 419

Figura 3.4: Conjuntos de secuencias de BALIBASE empleadas como datos

iteraciones en cada juego de secuencias). Por lo tanto, el algoritmo da como resultado un óptimo local después de haber reducido el espacio factible.

3.1.2. Datos de entrada.

BALIBASE es la base de datos pública y validada que proporciona información de secuencias que fueron ya alineadas por varios métodos.

De la versión 1 de BALIBASE se hizo un muestreo estratificado conformado por 9 juegos de secuencias que representan cerca del 10 por ciento del número de juegos de secuencias totales de la referencia 1. Se tomó de manera representativa un juego de secuencias con cada una de las características en las que se subdivide tal referencia (secuencias cortas con más del 25 por ciento de similitud, secuencias medias, entre 20 y 40 por ciento de similitud, secuencias largas con más del 35 por ciento de similitud, etc).

Así mismo se tomó un juego de secuencias de la referencia 4 (extensiones), que representa también cerca del 10 por ciento de las secuencias que contiene.

Los juegos de secuencias que se tomaron conforman el universo de datos que fueron sometidos al algoritmo y se muestran a continuación (véase figura 3.4).

Para apoyar a comprender el procesamiento de los datos, se ejemplificará el flujo del tratamiento del primer conjunto de secuencias (Prueba1aab).A continuación se muestra la información que éste contiene (véase figura 3.5).

Ahora bien, otros datos de entrada son los factores de genialidad. A continuación se explica su importancia y cómo fueron obtenidos. Recordemos que el algoritmo de composición musical usa funciones de aptitud (fitness en inglés) simulando el proceso de aprendizaje e innovación (relación de cada compositor con el ambiente y proceso personal de creatividad, respectivamente) (ver capitulo 2 sección Algoritmo de optimización inspirado en la com-

Figura 3.5: Secuencias del conjunto Prueba1aab.

posición Musical). Dichas funciones permiten acotar en mayor medida la región factible. La elección de usar una u otra en la creación de una melodía corre por cuenta de cada compositor y obedece a un proceso autoadaptativo.

Para que esto ocurra, es necesario definir factores, llamados de genialidad, inicialmente calibrados para cada una de las funciones. Los factores son ifg para la innovación y cfg para el aprendizaje (cambio de acuerdo al intercambio de información entre los agentes). Dicha calibración se llevó a cabo ejecutando varias corridas de prueba del algoritmo (2 iteraciones por cada conjunto de secuencias) con un valor inicial aleatorio entre 0 y 1 para ifg y cfg . Los valores de éstos fueron variando de corrida en corrida de manera tal que proporcionaran como resultado un número significativo de coincidencias. Finalmente los datos iniciales se determinaron como $ifg = 0.01$ y $cfg = 0.01$.

Al dato de entrada de número de compositores Nc se le asignó el valor de 5, al número de iteraciones $Max_arrragement$ un valor de 1000 y al número de pruebas para cada conjunto de secuencias Nsi el valor de 10. Todo lo anterior con el fin de conseguir resultados representativos sin invertir demasiado tiempo en la experimentación.

A continuación, (véase figura 3.6), se muestra la primera solución del conjunto de secuencias Prueba1aab:

3.1.3. Tratamiento de los resultados (conversión a formato MFS y validación).

Los resultados obtenidos son 10 soluciones posibles para cada conjunto de secuencias. Estas soluciones fueron traducidas al formato MFS mediante el uso del software GeneDoc. Posteriormente mediante Baliscore (herramienta desarrollada por Julie Thomson) se valida el resultado de cada prueba para cada uno de los conjuntos de secuencias. Baliscore asigna una calificación a cada resultado y ésta permite compararlos con los resultados obtenidos con otros métodos y almacenados en BALIBASE

.

Recursos empleados.

Los datos fueron procesados en un equipo: HP Pavilion dv2325la Notebook PC Procesador T5300 Intel Core 2 Duo 120 GB, 1024 MB, memoria 667 MHz DDR2. El Software empleado fue:

- MATLAB Ver 7.10.0.499 (R2010a) 32 bits (wm32).

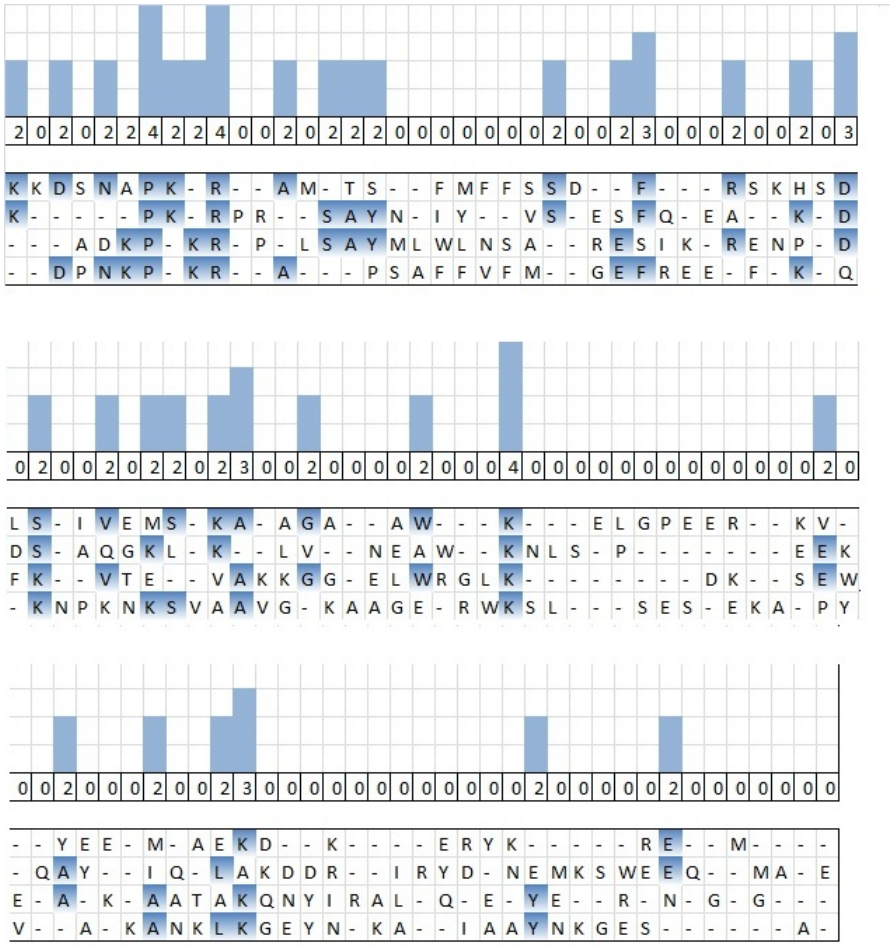

Figura 3.6: Alineamiento solucion para el conjunto de secuencias Prueba1aab.

- GeneDoc ver 2.7 (Multiple Sequence Aligment Editor and Shading Utility).
- BALISCORE.

3.1.4. Comparación de las soluciones del AMS.

Para la comparación de las soluciones del AMS se muestra la siguiente tabla, ubicando a cada método dentro de una escala normalizada entre 0 y 1, considerando a 0 como el mejor método y a 1 como la desviación máxima de éste (véase figuras 3.7 y 3.8).

Posteriormente se aplicó el método de Wilcoxon para determinar qué tan significativamente diferentes son los resultados obtenidos por el MCM contra los otros métodos con los que se comparó. El resultado es alentador, ya que prueba que los resultados por Composición Musical no son significativamente diferentes a los obtenidos específicamente con los métodos SB_PIMA, ML_PIMA, MULTAL, HMMT, Búsqueda Armónica (véase figura 3.9).

Para finalizar, y poder mostrar un rango en el que se encuentra la media cada uno de los conjuntos de secuencias, se aplicó el método de Bootstrap (véase figura 3.11), el cual generó 1000 poblaciones aleatorias a partir de cada una de las poblaciones obtenidas como resultado de la aplicacion del MCM al problema del AMS. Ello con el fin de obtener una media de medias y, finalmente un rango en que, con 95 por ciento de probabilidad, se puede encontrar ésta.

3.2. Análisis de Resultados.

Como puede apreciarse en la tabla de resultados, en el tratamiento del quinto conjunto de secuencias (Prueba1adj), el alineamiento múltiple de secuencias empleando el algoritmo de composición musical resuelve el problema. Cabe destacar que BALIBASE no reporta solución a éste.

Así mismo, para el noveno conjunto de secuencias (csp_ref4), no existiendo datos en la base de datos y considerando que un método de reciente aparición (Búsqueda de armonia) sí resuelve el problema siendo uno de los dos únicos métodos que da un valor satisfactorio para la solución), el algoritmo de Composición musical arroja un resultado 450 por ciento mejor que este último. Lo cual nos indica que es al menos mejor que los métodos con los que se comparó (PRRP, CLUSTALX, SAGA, SB_PIMA, ML_PIMA, Búsqueda Armónica, PILEUP8 y a excepción del MULTIALIGN, el cual es el mejor).

Ahora bien, haciendo un análisis global y con apoyo del análisis estádistico Wilcoxon, los resultados obtenidos por medio del MCM son significativamente iguales a los obtenidos por medio de los métodos SB_PIMA, ML_PIMA, MULTAL, HMMT, Búsqueda Armónica Y MMC.

	1aab		1aboA		1ad2	
		Normali		Normali		Normali
PRRP	1	0	0.56	0.49125418	0.943	0.07091316
CLUSTALX	1	0	0.687	0.34946036	0.984	0.01990545
SAGA	0.823	0.22317488	0.529	0.52586528	0.917	0.10325952
DIALIGN	1	0	0.359	0.71566803	0.96	0.04976362
SB_PIMA	1	0	0.312	0.76814291	0.934	0.08210998
ML_PIMA	1	0	0.312	0.76814291	0.934	0.08210998
MULTALIGN	1	0	0.703	0.33159657	0.96	0.04976362
PILEUPS	1	0	0.521	0.53479717	0.96	0.04976362
MULTAL	1	0	0.526	0.52921474	0.975	0.03110226
HMMT	0.214	0.99104779	0.181	0.91440268	0.341	0.81985569
BUSQUEDA A	0.35	0.81956878	0.10433333	1	0.272	0.90569793
PILEUP8	-	-	-	-	-	-
MCM	0.2069	1	0.1202	0.98228507	0.1962	1
optimo	1		1		1	
peor	0.2069	0	0.10433333	0.33159657	0.1962	1

	1ad3		1adj		1aho1	
		Normali				Normali
PRRP	0.972	0.03267211	-	-	0.99	0.01515611
CLUSTALX	0.968	0.03733956	-	-	0.971	0.04395271
SAGA	0.967	0.03850642	-	-	1	0
DIALIGN	0.963	0.04317386	-	-	1	0
SB_PIMA	0.962	0.04434072	-	-	1	0
ML_PIMA	0.962	0.04434072	-	-	1	0
MULTALIGN	0.979	0.02450408	-	-	0.913	0.13185814
PILEUPS	0.97	0.03500583	-	-	0.938	0.09396787
MULTAL	0.971	0.03383897	-	-	0.938	0.09396787
HMMT	0.738	0.30571762	-	-	0.789	0.31979388
BUSQUEDA A	0.23766667	0.88953714	-	-	0.49066667	0.77195109
PILEUP8	-	-	-	-	-	-
MCM	0.143	1	0.2116	0.7884	0.3402	1
optimo	1		1		1	
peor	0.143	1	0.2116	0.7884	0.3402	1

	1amk		1bbt3		csp_ref4	
		Normali		Normali		Normali
PRRP	0.986	0.01803659	0.907	0.10026954	0	1
CLUSTALX	0.989	0.01417161	0.706	0.31698113	0	1
SAGA	0.997	0.00386498	0.652	0.37520216	0	1
DIALIGN	0.993	0.00901829	0.45	0.59299191	0.889	0.111
SB_PIMA	0.987	0.01674826	0.26	0.79784367	0	1
ML_PIMA	0.987	0.01674826	0.26	0.79784367	0	1
MULTALIGN	0.991	0.01159495	0.582	0.45067385	0	1
PILEUPS	0.993	0.00901829	0.43	0.61455526	-	-
MULTAL	0.992	0.01030662	0.16	0.90566038	-	-
HMMT	0.941	0.07601134	0.128	0.94016173	-	-
BUSQUEDA A	0.39866667	0.77471442	0.16333333	0.90206649	0.001	0.999
PILEUP8	-	-	-	-	0	1
MCM	0.2238	1	0.0725	1	0.00455556	0.99544444
optimo	1		1		1	
peor	0.2238	1	0.0725	1	0	1

Figura 3.7: Comparacion de los resultados obtenidos con MMC frente a otros metodos de Balibase.

escala \ prueba	1aab	1aboA	1ad2	1ad3	1adj	1aho1	1amk	1bbt3	csp_ref4
1	HMMT y **MCM**	BUSQUEDA ARMONICA, **MCM**	**MCM**	**MCM**		**MCM**	**MCM**	**MCM**	PRRP, CLUSTALX, SAGA, SB_PIMA, ML_PIMA, MULTALIGN, BUSQUEDA ARMONICA, PILEUP8, **MCM**
0.9		HMMT	BUSQUEDA ARMONICA	BUSQUEDA ARMONICA				MULTAL, HMMT, BUSQUEDA ARMONICA.	
0.8	BUSQUEDA ARMONICA	SB_PIMA, ML_PIMA	HMMT.		**MCM**	BUSQUEDA ARMONICA	BUSQUEDA ARMONICA	SB_PIMA, ML_PIMA,	
0.7		DIALIGN							
0.6								DIALIGN, PILEUPS	
0.5		PRRP, SAGA, PILEUPS, MULTAL						MULTALIGN	
0.4								SAGA	
0.3		CLUSTALX, MULTALIGN		HMMT		HMMT		CLUSTALX	
0.2	SAGA								
0.1			PRRP, SAGA, SB_PIMA, ML_PIMA,			MULTALIGN, PILEUPS, MULTAL.	HMMT	PRRP	DIALIGN
0	PRRP, CLUSTALX, DIALIGN, SB_PIMA, ML_PIMA, MULTALIGN, PILEUPS, MULTAL		CLUSTALX, DIALIGN, MULTALIGN, PILEUPS, MULTAL	PRRP, CLSTALX, SAGA, DIALIGN, SB_PIMA, ML_PIMA, MULTALIGN, PILEUPS, MULTAL		PRRP, CLUSTALX, SAGA, DIALIGN, SB_PIMA, ML_PIMA.	PRRP, CLUSTALX, SAGA, SB_PIMA, ML_PIMA, MULTALIGN, PILEUPS, MULTAL		

Figura 3.8: Resultados finales.

Método	p	h
PRRP	0,048745372	1
CLUSTALX	0,048745372	1
SAGA	0,048745372	1
DIALIGN	0,003948992	1
SB_PIMA	0,072727272	0
ML_PIMA	0,072727272	0
MULTALIGN	0,048745372	1
PILEUPS	0,048745372	1
MULTAL	0,131139448	0
HMMT	0,421966269	0
BUSQUEDA A	0,489428218	0
PILEUP8	4.11E+09	1
MCM	1	0

Figura 3.9: Aplicacion de Prueba Wilcoxon a resultados finales.

1aab		1aboA		1ad2	
Media	0.2069	Media	0.1202	Media	0.1962
Mediana	0.2095	Mediana	0.1185	Mediana	0.181
Moda	#N/A	Moda	0.144	Moda	0.181
Desviación estándar	0.03401127	Desviación estándar	0.03269319	Desviación estándar	0.06391105
Varianza de la muestra	0.00115677	Varianza de la muestra	0.00106884	Varianza de la muestra	0.00408462
Curtosis	3.5310199	Curtosis	-1.1090838	Curtosis	1.22059154
Coeficiente de asimetría	-1.1103305	Coeficiente de asimetría	-0.0098167	Coeficiente de asimetría	0.89750838
Rango	0.134	Rango	0.098	Rango	0.228
Mínimo	0.128	Mínimo	0.073	Mínimo	0.098
Máximo	0.262	Máximo	0.171	Máximo	0.326
Suma	2.069	Suma	1.202	Suma	1.962
Cuenta	10	Cuenta	10	Cuenta	10

1ad3		1adj		1aho1	
Media	0.143	Media	0.2116	Media	0.3402
Mediana	0.1415	Mediana	0.209	Mediana	0.338
Moda	#N/A	Moda	#N/A	Moda	#N/A
Desviación estándar	0.01943079	Desviación estándar	0.02985595	Desviación estándar	0.04800185
Varianza de la muestra	0.00037756	Varianza de la muestra	0.00089138	Varianza de la muestra	0.00230418
Curtosis	-0.4414181	Curtosis	1.46197174	Curtosis	-0.1814916
Coeficiente de asimetría	0.31146906	Coeficiente de asimetría	0.83865622	Coeficiente de asimetría	-0.1466214
Rango	0.064	Rango	0.108	Rango	0.161
Mínimo	0.114	Mínimo	0.166	Mínimo	0.255
Máximo	0.178	Máximo	0.274	Máximo	0.416
Suma	1.43	Suma	2.116	Suma	3.402
Cuenta	10	Cuenta	10	Cuenta	10

1amk		1bbt3		1csp_ref4	
Media	0.2238	Media	0.0725	Media	0.00455556
Mediana	0.221	Mediana	0.0645	Mediana	0.005
Moda	#N/A	Moda	#N/A	Moda	0.005
Desviación estándar	0.03068043	Desviación estándar	0.02808024	Desviación estándar	0.00194365
Varianza de la muestra	0.00094129	Varianza de la muestra	0.0007885	Varianza de la muestra	3.7778E-06
Curtosis	-0.0936152	Curtosis	0.41984594	Curtosis	0.06617647
Coeficiente de asimetría	0.20648808	Coeficiente de asimetría	1.1471225	Coeficiente de asimetría	-0.8955036
Rango	0.104	Rango	0.083	Rango	0.006
Mínimo	0.174	Mínimo	0.045	Mínimo	0.001
Máximo	0.278	Máximo	0.128	Máximo	0.007
Suma	2.238	Suma	0.725	Suma	0.041
Cuenta	10	Cuenta	10	Cuenta	9

Figura 3.10: Estadistica descriptiva de los resultados obtenidos por MMC

	1aab	1aboA	1ad2	1ad3	1adj
limite inferior de la media	0.1104444444	0.1057777778	0.1180000000	0.1032222222	0.0995555556
limites superior de la media	0.2290000000	0.2225555556	0.2587777778	0.2331111111	0.1998888889

	1aho1	1amk	1bbt3	csp_ref4
limite inferior de la media	0.1027777778	0.1181111111	0.1191111111	0.0961111111
limites superior de la media	0.2290000000	0.2622222222	0.2648888889	0.2221111111

Figura 3.11: Aplicacion de Prueba Bootstrop a resultados finales

Capítulo 4
Conclusiones y Trabajos Futuros.

En este trabajo se propuso una estrategía para la solución del problema del AMS que se basa en el empleo de las sociedades artificiales, específicamente el de Composición Musical que simula el proceso dinámico de la creación de una obra musical por medio del intercambio de información (aprendizaje) entre los agentes que participan, en este caso, cada compositor.

El objetivo inicialmente buscado se alcanzó dado que el Algoritmo de Composición Musical sí arrojó resultados en los conjuntos de secuencias que fueron procesados. El reto consisitió en modificar dicho algoritmo para aplicarlo al AMS bajo condiciones restrictas en un problema discreto binario.

El análisis de resultados nos indica que el algoritmo de Composición musical fue útil en la solución del alineamiento múltiple de secuencias, arrojando resultados inclusive en conjuntos de secuencias en los que BALIBASE no cuenta con datos.

El método Wilcoxon y el Bootstrap, que se aplicó a los resultados obtenidos con el AMS, permiten determinar que dichos resultados fueron significativamente iguales a los obtenidos por medio de los métodos SB_PIMA, ML_PIMA, MULTAL, HMMT, Búsqueda Armónica Y MMC, de los que BALIBASE reporta soluciones.

De hecho, cabe mencionar que en el único conjunto de secuencias que se eligió de la referencia 4, el resultado fue muy satisfactorio con el MMC, siendo 450 por ciento mejor que el segundo mejor resultado de BALIBASE registrado para su solución. Se debe recordar que las características de los conjuntos de secuencias de la referencia 4 son más complejas que las de la referencia 1. Lo cual es un indicativo de que, a primera instancia y para secuencias con un alto grado de dificultad de alineamiento, el algoritmo de Composición musical puede funcionar mejor que para secuencias menos complejas y llegar a ser una herramienta mejor que los métodos ya conocidos. Para comprobarlo podría considerarse para trabajos futuros la aplicación del algoritmo en un número mayor de conjuntos de secuencias de la referencia 4 o 5. Para comprobar lo dicho, se propone aplicar el Algoritmo de Composición Musical a un mayor número de conjuntos de secuencias de la referencia 4 o inclusive de referencias subsecuentes de BALIBASE en trabajos futuros.

Una manera de mejorar el algoritmo sería calibrar los factores de genialidad mediante la aplicación de árboles con métodos de programación dinámica u otra técnica para reducir el espacio de búsqueda inicial. Esto posiblemente podría mejorar también el tiempo de procesamiento. Otra manera de mejorar el algoritmo sería incrementar el número de iteraciones para cada conjunto de secuencias. Lo arriba mencionado no fue llevado a cabo por la limitante en tiempo para el desarrollo del presente trabajo.

Así pues, el algoritmo de Composición Musical en la aplicación de la solución del problema de alineamiento múltiple de secuencias aportó una nueva manera de abordarlo, haciendo uso del conocimiento proporcionado por los algoritmos bioinspirados.

Bibliografía

Altschul, S.F.: «Gap costs for multiple sequence alignment». *J.Theor.Biol*, 1989, **138**, pp. 297–309.

Altschul, S.F. y Lipman, D.J.: «Trees,strars, and multiple biological sequence alignment». *J.appl.Math*, 1989, **49**, pp. 197–209.

Arenas Díaz, Edgar David: *Alineamiento de múltiples secuencias genéticas usando cómputo evolutivo.* Tesina o Proyecto, Universidad Nacional Autónoma de México, 2009.

Attwood, T. K. y Parry-Smith, D.J.: *Introducción a la Bioinformática.* Prentice Hall, 2002.

Bacon, D.J. y Anderson, W.F.: «Multiple sequence alignment». *J.molec.Biol*, 1986, **191**, pp. 153–161.

Bains, W.: «A program to align multiple DNA sequences». *Nucl.Acids.Res*, 1986, **14**, pp. 159–177.

Barton, G.J. y Sternberg, M.J.E.: «A strategy for the rapid multiple alignment of protein sequences: confidence levels from tertiary structure comparisions». *J.molec.Biol*, 1987, **198**, pp. 327–337.

Biles, J.A.: «International music conference. International Computer Music Association,Aarhus». En: *Genjam:a genetic algorith for generating jazz solos.*, , 1994.

Carrillo, H. y Lipman, D.: «The multiple sequence alignment problem in biology.» *J.appl.Math*, 1988, **48**, pp. 1073–1082.

Chan, S.C.; Wong, A.K. y Chiu, D.K.: «A Survey of Multiple Sequence Comparison Methods». *Bolletin of Mathematical Biology*, 1992, **54**, pp. 563–598.

Cohen, D.N.; Reichert, T.A. y Wong, A.K.: «Matching code sequences utilizing context free quality measure». *Math.Biosci.*, 1975, **24**, pp. 25–30.

Cope, D.: *The agorithmic composer.* A-R.Editions Inc. Winsconsin, 2000.

Corpet, F.: «Multiple sequence alignment with hierarchical clustering». *Nucl.Acids.Res*, 1988, **16**, pp. 10881–10890.

Davison, D.: «Sequence similarity ("homology") searching for molecular biologists.» *Bulletin of Mathematical Biology*, 1985, **47**, pp. 437–474.

Dayhoff, M.O.; Barker, W.C. y Hunt, L.T.: «Establishing Homologies in protein sequences». *Methods in enzymology*, 1983, pp. 524–538.

Dorigo, M.; Maniezzo, V. y Colorni., A.: «Ant systems: optimization by a colony of cooperating agents.» *IEEE Trans. Syst. Man. Cybernet.*, 196, **26**, pp. 29–41.

Duman, J.P. y Ninio, J.: «Efficient algorithms for folding and comparing nucleic acid sequences». *Nucl.Acids.Res*, 1982, **10**, pp. 197–206.

Dumey, A.I.: «Indexing for rapid random-access memory». *Compt.Automat.*, 1956, **5**, pp. 6–8.

Elias, I.: «Setling the intractability of multiple alignment». *14 th Annual Int. Symph on algorithms and computation*, 2003, pp. 352–363.

Enderton, H.B.: *Una introducción matemática a la lógica*. Universidad Nacional Autónoma de México, 1987.

Feng, D.F. y Doolitle, R.F.: «Progressive sequence alignment as a prerequisite to correct phylogenetic trees». *J.molec.Evol*, 1987, **25**, pp. 351–360.

Fickett, J.W.: «Fast optimal aligment». *Nucl.Acids.Res*, 1984, **12**, pp. 175–180.

Fitch, W.M.: «Towards defining the course of evolution:minumum change for a specific tree topology». *Syst.Zool*, 1971, **20**, pp. 406–416.

Freedman, M.L.: «Algorithm for computing evlutionary similarity measures with length independ gap penalties». *Bull.Math.Biol*, 1984, **46**, pp. 553–566.

Gotoh, O.: «An improved algorithm for matching biological sequences». *J.molec.Biol*, 1982, **162**, pp. 705–708.

Grimaldi, R.P.: *Matemáticas discretas y combinatorias . Una introducción con aplicaciones.* Pearson Printice Hall, 1998.

Gusfieeld, D.: «Efficient methods for multiple sequence aligment with garantee error bonds.» *Bulletin of Mathematical Biology*, 1993, pp. 141–154.

Hartigan, J.A.: «Minimum mutation fits to a given tree». *Biometrics*, 1973, **29**, pp. 53–65.

Hein, J.: «A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences,when the phylogeny is given». *Molec.Biol.Evol*, 1989, **6**, pp. 649–668.

Higgins, D.G. y Sharp, P.M.: «CLUSTAL: a package for performing multiple sequence alignment on a microcomputer». *Gene*, 1988, **73**, pp. 237–244.

Hines, William; Montgomery, Douglas; Goldsman, David y Borror, Connie: *Probabilidad y Estadística para ingeniería*. CECSA, 2006.

Hogeweg, P. y Hesper, B.: «The alignment of sets of sequences and the construction of phyletic trees:an integrated method». *J.molec.Evol*, 1984, **20**, pp. 175–186.

Horner, A. y Goldberg, D.E.: «Genetic algorithms and computer assisted music composition». En: *ICMC91 proceedings, International Computer Music Association,Sn Francisco.*, , 1991.

Jacob, B.: «Compusing with genetic algorithms.» *International Computer Music Association,San Francisco.*, 1995, pp. 452–455.

Jacob, B.L.: «Algorithmic composition as a model of creativity.» *Organ Sound*, 1996, **1**, pp. 157–165.

Johnson, M.S. y Doolitle, R.F.: «A method for the simultaneous alignment of three or more amino acid sequences». *J.molec.Evol*, 1986, **23**, pp. 267–278.

Jue, R.A.; Woodbury, N.W. y Doolitle, R.F.: «Sequence homologies among E.Coli ribosomal proteins:evidence for evolutionary related grouping and internal duplications». *J.molec.Evol*, 1980, **15**, pp. 129–148.

Karlin, S.G.; Ghandour, G.; Ost, F.; Tavare, S. y Korn, L.J.: «New approaches for computer analysis of nucleic acid sequences». *Proc. nat. Acad. Sci.U.S.A.*, 1983, **80**, pp. 5660–5664.

Krishnan, G.; Kaul, R.K. y Jagedeeswaran, P.: «DNA sequence analysis: a procedure to find homolgies among many sequences». *Nucl.Acids.Res*, 1986, **14**, pp. 543–550.

Lathrop, R.H.; Webster, T.A. y Smith, T.F.: «ARIADNE: pattern.directed inference and hierarchical abstraction in protein structure recognition». *Comm.ACM*, 1987, **30**, pp. 909–921.

Lee, R.C.; Tseng, S.S.; Chang, R.C. y Tsai, Y.: *Introducción al diseño y análisis de algoritmos*. Mc Graw-Hill, 2007.

Leluk, J.: «A new algorith for analysis of the homologya in protein primary structure.» *Computer Chem*, 1998, pp. 123–131.

——: «Regularities in mutational variability in selected protein families and the Markovian model of aminoacid replacement». *Computers & Chemistry*, 2000, pp. 659–672.

Lipman, D.J.; Altschul, S.F. y Kececioglu, J.D.: «A tool for multiple sequence alignment». *Proc. nat. Acad. Sci.U.S.A.*, 1989, **86**, pp. 4412–4415.

Liu., Y.T.: «Creativity or novelty: Cognitive-computational versus social-cultural». *Design Stud.*, 2000, **23**, pp. 261–276.

Ma, B.; Wang, L. y Li, M.: «Near optimal multiple aligment with bonds in polynomial time». *Journal of computer and systems sciences*, 2007, pp. 997–1011.

Manthey, B.: «Non-Aproximability of weighted multiple sequence aligment». *Theoretical Computer science*, 2003, pp. 179–192.

Martínez, H.M.: «An efficient method for finding repeats in molecular sequences». *Nucl.Acids.Res*, 1983, **11**, pp. 4629–4634.

——: «A flexible multiple sequence alignment program». *Nucl.Acids.Res*, 1988, **16**, pp. 1683–1691.

Mora-Gutierrez, Roman A: *Desarrollo de un procedimiento para solucionar el problema de alineamiento múltiple de secuencias*. Tesina o Proyecto, Universidad Nacional Autónoma de México, 2009.

Mora-Gutiérrez, Roman Anselmo; Ramírez-Rodríguez, Javier y Elizondo-Cortés, Mayra: «Heurística para solucionar el problema de alineamiento múltiple de secuencias». *Revista de Matemática Teoría y Aplicaciones*, 2011, **18**, pp. 121 – 136. ISSN 1409-2433.
http://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S1409-24332011000100009&nrm=iso

Murata, M.; Richardson, J.S. y Sussman, J.L.: «Simultaneous comparison of three protein sequences». *Proc.natn.Acad.Sci.U.S.A.*, 1985, **82**, pp. 3073–3077.

Najarian, Kayvan; Eichelberger, C.N. y Gharibzadeh, Shahriar: *Systems Biology & Bioinformatics/ A Computational Approach*. CRC Press, 2009.

Needleman, S.B. y Wunsch, C.D.: «A general method applicable to the search for similarities in the amino acid sequences of two proteins.» *Journal of molecular Biology*, 1970, **48**, pp. 444–453.

Patthy, L.: «Detecting homology of distantly related proteins with consensus sequences». *J.molec.Biol*, 1987, **198**, pp. 567–577.

Pevzner, Pavel A: *Computational molecular biology: an algorithmic approach*. The MIT Press, 2000.

Ray, T. y Liew, K.M.: «Society and civilization:an optimization algorithm based on simulation of social behavior.» *IEEE Trans. Evol. Comput.*, 2003, **7**, pp. 386–396.

Reichert, T.A.; Cohen, D.N. y Wong, A.K.C.: «An application of information theory to genetic mutations and matching of polypeptide sequences». *J. Theor. Biol.*, 1973, **42**, pp. 245–261.

Reynolds., R.G.: «An introduction to cultural algorithms.» En: *Proceedings on the 3rd annual conference on evolutionary programming, World Scientific Publishing*, , 1994.

Sankoff, D.: «Matching sequences under deletion-insertion constraints». *Proc. nat. Acad. Sci.*, 1972, **68**, pp. 4–6.

——: «Minimum mutation trees of sequences». *J.appl.Math*, 1975, **78**, pp. 35–42.

Sankoff, D. y Cedergren, R.J.: *Simultaneous comparison of tree or more sequences related by a tree.* Addison Wesley, 1983.

Sankoff, D.; Cedergren, R.J. y W.McKay: «A strategy for sequences phylogeny research». *Nucleic Acid Research*, 1982, pp. 421–431.

Sankoff, D. y Sellers, P.: «Shortcuts,diversions and maximal chains in partially ordered set». *Discrete Mathematics*, 1973, **4**, pp. 278–293.

Schmidt, Bertil: *Bioinformatics/ High Perfomance Parallel Computer Architectures.* CRC, 2010.

Sneath, H.A. y Sokal, R.: *Numerical Taxonomy.* W.H.Freeman, 1973.

Sobel, E. y Martínez, H.M.: «A multiple sequence alignment program». *Nucl.Acids.Res*, 1986, **14**, pp. 363–374.

Subbiah, S. y Harrison, S.C.: «A method for multiple sequence alignment with gaps». *J.molec.Biol*, 1989, **209**, pp. 539–548.

Taylor, P.: «A fast homology program for aligning biological sequences». *Nucl.Acids.Res*, 1984, **12**, pp. 447–455.

Taylor, W.R.: «A flexible method to align large number of biological sequence». *J.molec.Evol*, 1988, **28**, pp. 161–169.

Waterman, M.S.: «General methods of sequence comparision.» *Bulletin of mathematical Biology.*, 1984, **46**, pp. 473–500.

——: «Multiple sequence alingnment by consensus». *Nucl.Acids.Res*, 1986, **14**, pp. 9095–9102.

Waterman, M.S. y Jones, R.: «Consensus methods for DNA and protein sequence alignment». *Methods Enzymol*, 1990, **183**, pp. 221–237.

Waterman, M.S. y Perlwitz, M.D.: «Line geometries for sequence comparision». *Bull.Math.Biol*, 1984, **46**, pp. 567–577.

Waterman, M.S.; Smith, T.F. y Beyer, W.A.: «Some biological sequence metrices». *Adv.Math*, 1976, **20**, pp. 367–387.

Webster, T.A.; Lathrop, R.H. y Smith, T.F.: «Prediction of a common structural domain in aminoacyl-tRNA synthetases through use of a new patern-directed inference system.» *Biochemistry*, 1987, **26**, pp. 6950–6957.

Wilbur, W. y Lipman, D.J.: «The context dependent comparision of biological sequences». *J.appl.Math*, 1984, **44**, pp. 557–567.

Wong, A.K.C.: «Structural pattern recognition: a random graph approach.» *Pattern Recognition Theory and Applications*, 1987, **F30**.

Wong, A.K.C: *Syntactic and Structural Pattern Recognition: Fundamentals,Advances and Applications.* World Scientifdic Publishing Company PteLtd, 1990.

Wong, A.K.C; Reichert, T.A.; Cohen, D.N. y Aygun, B.O.: «A generalized method for matching informational macromolecular code sequences». *Comput.Biol.Med*, 1974, **4**, pp. 43–57.

Wong, A.K.C y You, M.: «Entropy and distance of random graphs with application to structural pattern recognition». *IEEE Trans.Pattern Anal.MAchine Intell*, 1985, **7**, pp. 599–609.

You, M.: *A random graph to pattern recognition.* Tesina o Proyecto, Universty of Waterloo, 1983.