Identificación de Secuencias Reguladoras Mediante Agrupamiento

por

Dulce María García Ordaz

Tesis sometida como requisito parcial para obtener el grado de

MAESTRO EN CIENCIAS EN LA ESPECIALIDAD DE

CIENCIAS COMPUTACIONALES en el Instituto Nacional de

Astrofísica, Óptica y Electrónica

Supervisada por:

Dr. Jesús Antonio González Bernal

Dr. Aurelio López López

Febrero 2011, Tonantzintla, Puebla

Agradecimientos

A mis asesores, los doctores Jesús Gonzalez Bernal y Aurelio Lopez Lopez, por su apoyo y orientación.

A mis sinodales Dr. Miguel Arias Estrada, Dr. Ariel Carrasco Ochoa y Dr. Luis Villaseñor Pineda por sus observaciones y comentarios acertados.

Al INAOE, por todas las facilidades proporcionadas durante mi estancia académica.

A mi padres, que siempre me dieron su apoyo y cariño.

A mi amigos Marco, Claudia, Betanzos y Rodolfo que han estado a mi lado animándome y alegrándome la vida.

A CONACYT, por el apoyo económico a través de la beca No. 224394.

Resumen

En los últimos años ha aumentado la cantidad de organismos de los que se ha obtenido su secuencia de ADN. La gran cantidad de datos que existen contrasta con el poco conocimiento que se tiene de las funciones del ADN. Las secuencias reguladoras son un tipo de secuencia dentro del ADN que se encarga de activar o desactivar a los genes y se encuentran en regiones cercanas a éstos. Las secuencias reguladoras son patrones inexactos, y pueden ser hallados mediante métodos computacionales. Las herramientas que existen actualmente para el descubrimiento de secuencias reguladoras se encuentran limitados por diversos factores. Algunos de estos factores son el número de secuencias de entrada y la longitud de las secuencias reguladoras que pueden descubrir. Un punto importante es que hasta ahora no existe algún método capaz de identificar todas las secuencias reguladoras que existan en el genoma, o en un subconjunto de genes. Se ha encontrado que todas las herramientas existentes suelen encontrar por lo menos una secuencia que las demás no [16], lo que provoca que los métodos de descubrimiento sean complementarios entre si. En este trabajo se propone un método para la identificación de secuencias reguladoras. Éste método está basado en un algoritmo de agrupamiento jerárquico divisivo para identificar los patrones que posteriormente serán evaluados para determinar si son o no candidatos a secuencias reguladoras. Se decidió utilizar un método de agrupamiento debido al tamaño de la bases de datos, por ejemplo el organismo Bacillus Subtilis, con el que se evaluó este método, cuenta con más de 4400 genes. Los resultados muestran que el método es capaz de identificar estas secuencias con una precisión cercana al promedio de los métodos existentes, con la ventaja de que propone el tamaño de las secuencias.

Abstract

In recent years the number of organisms, which has been obtained its DNA sequence, has increased . The large amount of data that exists contrast with the limited knowledge we have of the functions of DNA. The regulatory sequences are short sequences of DNA wich turn on or turn off the genes. These sequences are found in regions close to genes. The regulatory sequences are inexact patterns, and can be found using computational methods. The currently existing tools for the discovery of regulatory sequences are limited by several factors. Some of these factors are the number of sequences and the length of the regulatory sequences that can be discovered. An important point is that so far there isn't a method capable of identifying all regulatory sequences that exist in the genome, or in subset of genes. And it was found that all existing tools tend to find at least one sequence that others methods do not find [16], what causes discovery methods are complementary to each other. This paper proposes a method for identifying regulatory sequences. This method is based on a divisive hierarchical clustering algorithm to identify patterns which are then evaluated to determine candidates for regulatory sequences. We decided to use a clustering method due to the size of databases, such as Bacillus subtilis organism, which was evaluated with this method, has more than 4400 genes. The results show that the method is able to identify these sequences.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Bases

Toda la información genética de los seres vivos se encuentra contenida en el ADN. El ADN consta de dos tiras largas intercruzadas que forman una doble hélice (ver figura 1.1). Cada tira, a su vez, está constituida por un conjunto de moléculas llamadas nucleótidos. Estos nuecleótidos son: Adenina, Guanina, Citosina y Tiamina, pero se acostumbra abreviar cada uno de ellos con la primera letra de su nombre. A los nucleótidos también se les conoce como bases, y debido a que existe una correspondencia entre cada par de elementos de cada tira, también se les llama pares de bases. La longitud de una secuencia de ADN, se mide en pares de bases. Los nucleótidos se complementan entre si. En la hélice, la Adenina siempre va unida a la Tiamina, y la Citosina a la Guanina.

Un gen es una sección del ADN. La función de los genes es producir proteínas o aminoácidos. Las proteínas y los aminoácidos sirven para crear las células que forman a los seres vivos. Cuando un gen está produciendo una proteína o aminoácido se dice que está expresado (encendido o activado). Todas las células de un organismo contienen exactamente el mismo ADN, pero el tipo de célula que será depende de qué genes sean

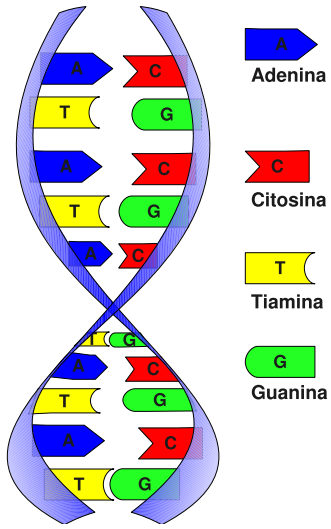

Figura 1.1: El ADN está formado por dos tiras de nucleótidos entrelazadas

los que estén expresados, así, una célula formará parte de un músculo, de un ojo, etc. El proceso de expresión de un gen está constituido por dos pasos: Transcripción y Traducción, ver figura 1.2. Durante la transcripción se hace una copia del gen en la forma de una molécula de ARN (ARN significa Ácido Ribonucleico, esta es una molécula con una forma muy parecida al ADN, pero en ella se sustituye la Tiamina (T) por la base Uracilo (U)). La Transcripción inicia cuando un Factor de Transcripción (TF), que es un tipo de proteína, se enlaza a un TFBS (Sitio de Enlace del Factor de Transcripción), figura 1.3. En la Traducción se codifica el ARN en una proteína. Los TFBS son un tipo de secuencias reguladoras.

Las secuencias reguladoras se encuentran en las regiones de ADN que están entre gen y gen, llamadas regiones intergénicas. Las secuencias reguladoras normalmente aparecen en las cercanías de los genes que regulan, a una distancia variable de entre 20 y 200 pares de bases hacia arriba. A las regiones donde se encuentran las secuencias reguladoras se les llama regiones reguladoras o regiones promotoras. Las secuencias reguladoras son secuencias cortas de ADN, entre 5 y 30 pares de bases (pb) [18]. Las secuencias regula-

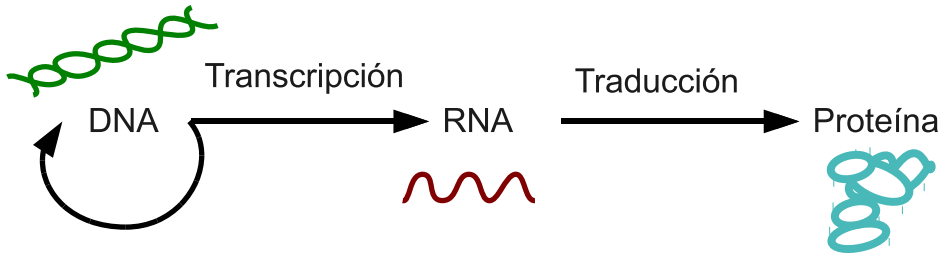

Figura 1.2: Expresión Génética

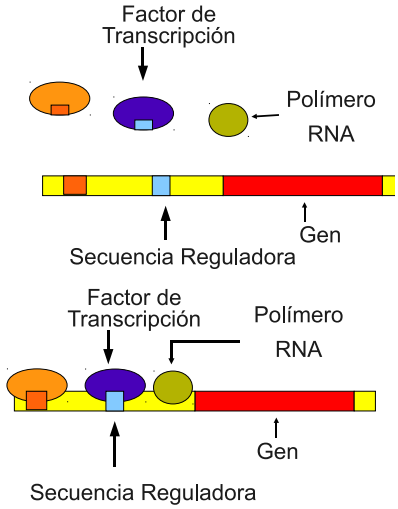

Figura 1.3: Transcripción

doras ligadas a un mismo Factor de Transcripción comparten características comunes, las cuales pueden ser representadas mediante una *secuencia consenso* que es básicamente, una secuencia formada por los nucléotidos que tienen la mayor frecuencia de aparición para cada posición.

La identificación de los elementos reguladores es crucial en el entendimiento de los mecanismos biológicos. Los biólogos pueden realizar la ubicación de estos elementos mediante experimentos in vitro. Sin embargo, estos experimentos implican el uso de costosos recursos y es un proceso que toma mucho tiempo. Es posible aplicar métodos computacio-

nales para la identificación de estas secuencias.

1.2. Definición del problema

Computacionalmente el problema de la identificación de secuencias reguladoras puede definirse como la búsqueda de fragmentos cortos, que mantienen un patrón, en un conjunto de secuencias genómicas. Estos fragmentos serán las secuencias reguladoras. Sin embargo, surgen algunos retos tales como:

- Cuando el ADN se replica se provocan mutaciones. Por lo que los patrones que se buscan no serán exactos. Pueden contener substituciones, inserciones, o borrados.
- Las secuencias se encuentran distribuidas aleatoriamente.
- Las secuencias reguladoras son muy cortas miden, entre 5 y 30 pb. Y se les encuentra en las regiones no codificantes del genoma, y estas regiones pueden llegar a representar más del 95 % del tamaño total de este.

Algunos problemas que se presentan al tratar de reconocer las secuencias reguladoras son:

- **Falta de conocimiento de las propiedades de las secuencias reguladoras.-** Las propiedades y reglas con las que se comportan las secuencias reguladoras en el ADN no se encuentran bien definidas.
- **TFBS degenerados (Sitios de Enlace de los Factores de Transcripción, Transcription Factors Binding Sites).-** Los factores de transcripción tienen una baja especificidad con los sitios donde se enlazan y esos sitios suelen ser cortos e imprecisos.
- **Falta de entendimiento de la evolución de la regulación transcripcional.-** Aún no se comprenden completamente las reglas por las qué se rige la evolución de los

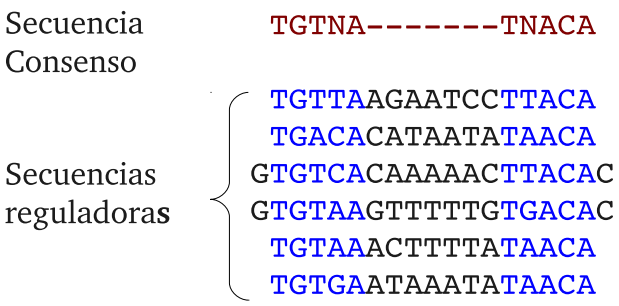

Figura 1.4: Ejemplo de Secuencias Reguladoras

elementos reguladores. Esto afecta porque no se sabe qué cambios pueden ocurrir entre las secuencias reguladoras de los organismos.

- **Secuencias reguladoras sin una estructura regular.-** La composición y organización de las secuencias reguladoras tiene grandes variaciones, y se encuentran dispersos en el ADN de forma desigual.

En la figura 1.4 se presentan algunas secuencias reguladoras, así como su consenso, que se forma con las letras de los nucleótidos que tienen la mayor aparición en determinada posición. En el caso de que no haya ningún nucleótido con una aparición mayor que los demás se indica con una N. Los son llamados brechas (*gap*), y significan que la longitud de este segmento de la secuencia puede variar. Éstas secuencias están ligadas a un factor de transcripción (TF) llamado *GlnR*. Como se puede observar, las secuencias reguladoras tienen diferentes tamaños, no son exactamente iguales, e incluso en las áreas de las orillas donde existe un mayor parecido, existen algunas variaciones.

Se han propuesto varios métodos para resolver el problema de la identificación de las secuencias reguladoras. Sin embargo este continua siendo un problema abierto puesto que aun no existe un método que sea capaz de predecir todas las secuencias reguladoras. Mas aún, se ha encontrado que los métodos propuestos son complementarios, ya que todos suelen identificar por lo menos una secuencia que los demás no. Dentro de las fallas que aún presentan estos métodos se pueden mencionar: el gran número de falsos positivos que

obtienen. Muchos de los métodos computacionales propuestos hasta ahora dependen del conocimiento previo, ya que comparan una posible nueva secuencia reguladora con un conjunto de secuencias reguladoras ya conocidas. Sin embargo esta solución puede llevar a resultados sesgados y tienen problemas para encontrar secuencias reguladoras con características diferentes a las ya conocidas. Aunque se sabe que las secuencias miden entre 5 y 30 bp, no se conoce de antemano el tamaño exacto, y, aunque las secuencias pertenezcan a un mismo patrón, no necesariamente serán de la misma longitud, por esto, determinar el tamaño de las secuencias reguladoras es otro de los retos en esta búsqueda, pocos algoritmos propuestos ofrecen una solución a este problema [21], y queda a consideración del usuario el tamaño que tendrán las secuencias.

En general, los métodos computacionales generados hasta ahora consisten en dos fases, en la primera se buscan secuencias candidatas a reguladoras. En la segunda fase se hace un agrupamiento de estas secuencias obtenidas para disminuir el número de soluciones candidatas.
Los métodos computacionales deben solucionar tres aspectos del problema:

1. Representación de las secuencias. Se debe definir un modelo computacional para representar a estas secuencias. Este modelo debe permitir la comparación entre ellas. Las más usadas son las cadenas y las matrices de pesos, ambas representaciones serán explicadas en la siguiente sección. Para este trabajo se ha decidido sacar provecho de las dos representaciones, y se hace uso de ellas en diferentes etapas del método.

2. Algoritmo de búsqueda.- Debe proponerse un algoritmo que realice la búsqueda de estos elementos en las secuencias genómicas. Esta búsqueda puede ser exhaustiva, heurística, *greedy*, por medios de alineamientos múltiples, muestreos, o algoritmos genéticos. Se ha decidido tratar este problema como un problema de agrupamiento. Esta idea se propone porque ya que las secuencias reguladoras son patrones y están sobrerepresentadas en el genoma, entonces es posible que estas lleguen a formar un

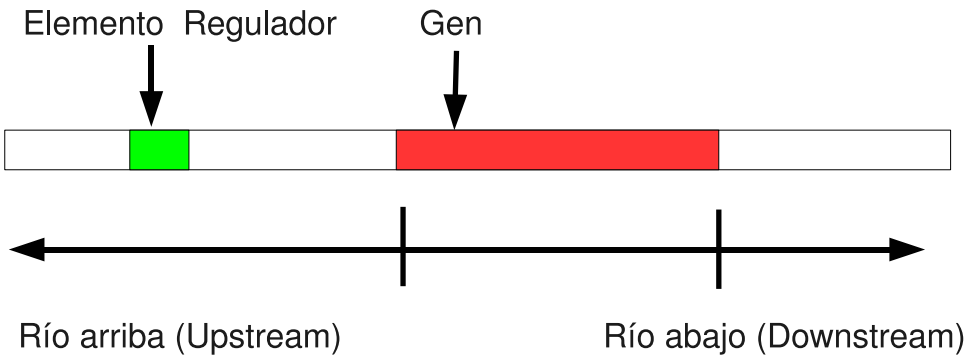

Figura 1.5: Ubicación de los elementos reguladores

grupo.

3. Una función de evaluación.- Una vez halladas las posibles secuencias reguladoras se debe proporcionar una forma de evaluar la conservación de estas con respecto al resto del genoma, y así, obtener las que tengan mayor probabilidad de ser verdaderas secuencias reguladoras. Para este trabajo se utilizó la entropía cruzada, el contenido de información y la función MAPscore propuesta en [20].

Finalmente, todos los métodos deben lidiar con los problemas del tiempo para encontrar las secuencias, la cantidad de memoria que ocupan, y deben tratar de reducir el número de falsos positivos que obtienen.

1.2.1. Objetivos

El objetivo de este trabajo es desarrollar un método para identificar secuencias reguladoras. Este método no estará sujeto al conocimiento previo existente, como por ejemplo el uso de secuencias reguladoras conocidas. El método también propondrá automáticamente el tamaño de las secuencias. El método propuesto estará basado en una técnica de agrupamiento jerárquico.

1.2.2. Objetivo General

Diseñar un método basado en agrupamiento para encontrar ”secuencias reguladoras”.

1.2.3. Objetivos Específicos

Para poder alcanzar el objetivo general, se debe lograr una serie de objetivos específicos que ayudarán a conseguirlo.

Definir una representación de los datos. Existen diferentes representaciones para los datos que se van a manejar. Se debe elegir una representación eficiente para el método que se usará.

Definir una medida de similitud entre secuencias. Las secuencias de ADN y en general las cadenas de caracteres pueden ser comparadas mediante diversas medidas de similitud. La elección de una medida de similitud es importante en este trabajo ya que, al estar basado en un algoritmo de agrupamiento, los resultados se verán afectados por la medida empleada.

Analizar algunos algoritmos de agrupamiento para la tarea. Se analizaron algunos algoritmo de agrupamiento para seleccionar uno que sea adecuado para llevar a cabo la tarea.

Adaptar un algoritmo de agrupamiento. Cuando se haya elegido un algoritmo de agrupamiento, éste se adaptará para que trabaje con la representación de datos, y la medida de similitud formulada.

Validar el método en una colección de datos. Finalmente el método diseñado se evaluará en una colección de datos.

Organización de la Tesis

La tesis está organizada de la siguiente manera: En el capitulo 2 se encuentra una revisión de los conceptos básicos. En esta sección se mencionarán las formas más comunes

de representar las secuencias reguladoras, se hará también una breve revisión de algunas medidas de similitud usadas en la comparación de cadenas, y por último se describe en qué consiste la técnica de agrupamiento, poniendo especial atención en el Agrupamiento Jerárquico Divisivo.

En el capitulo 3 se presenta una revisión del trabajo previo que existe en el área. El capítulo se divide en dos secciones, una en donde se detallan los métodos enumerativos y la otra para los métodos probabilistas. En este capítulo se explica por que el método propuesto se considera probabilista.

En el capítulo 4 se detalla el método propuesto para solucionar el problema planteado. Se expone la forma en que los datos son preparados para analizarse, se explican las funciones de similitud usadas en el método, y se detalla el procedimiento de agrupamiento.

En el capítulo 5 se encuentran los resultados de los experimentos realizados para validar el método. Se da una explicación breve de los datos utilizados para evaluar el desempeño del método propuesto. Y se presentan los resultados de evaluar el método en las bases de datos.

Finalmente el capítulo 6 contiene las conclusiones a las que se llegó después de haber realizados los experimentos con el método propuesto, igualmente se ofrecen sugerencias para el trabajo futuro.

Capítulo 2

Marco Teórico

2.1. Representación de los Elementos Reguladores

Existen diversas formas de representar a las secuencias reguladoras. Las dos principales formas son mediante cadenas, y matrices de pesos.

2.1.1. Representaciones basadas en cadena

Supóngase que se cuenta con un conjunto de secuencias alineadas. Para representar a este conjunto de secuencias, se define una cadena *consenso* que está formada por la subsecuencia de letras (nucleótidos) que tienen en común las secuencias alineadas. Si se tiene varias de estas secuencias consenso, se pueden representar de una forma más compacta mediante una *cadena consenso degenerada*, la cual consiste en substituir los nucleótidos en los que difieren las secuencias consenso, por códigos IUPAC. Este código consiste en un conjunto de letras con las que se sustituye a un determinado conjunto de nucleótidos. Por ejemplo, las secuencias AGAGAGTGTG, y GGAGAGTGTG son iguales excepto por el primer elemento. Entonces, de acuerdo a la tabla de los códigos IUPAC (tabla 2.3) cuando se tiene una A y una G, pueden ser substituido por una R. En las tablas 2.1 y 2.2 se pueden observar dos alineaciones de secuencias. Los consensos de los grupos 1 y 2 se forman con las letras con mayor repetición en cada posición. El consenso dege-

- - C T C A C A C A C G T G G G A C T A G C
- T T T C C A G C A C G T G G G G C G G A -
- - T T A T G G C A C G T G C G A A T A A -
G A T C G C T G C A C G T G G C C C G A - -
T A A T T T G G C A T G T G C G A T C T C -
- - - A C G T C C A C G T G G A A C T A T -
- - - T T T A T C A C G T G A C A C T T T T
- - - - - - - - G C A C G T G G G - - - - - -

Tabla 2.1: Grupo 1. El Consenso de este grupos es GCACGTGGG

T A A A T T A G C A C G T T T T C G C - - - -
- - A A T A C G C A C G T T T T T A A T C T A
- - - T T A C G C A C G T T G G T G C T G - -
- - - T T A C C C G C A C G C T T A A T A T -
- - - - - - - - G C A C G T T T T - - - - - -

Tabla 2.2: Grupo 2. El Consenso de este grupos es GCACGTTTT

nerado GCACGTKKK se obtiene a partir del alineamiento de las secuencias consenso GCACGTGGG y GCACGTTTT obtenidas de los grupos 1 y 2. Como se puede observar están formados por los mismos nucleótidos excepto por los últimos tres, entonces, revisando nuevamente la tabla con los códigos IUPAC, se observa que una T y una G se pueden sustituir por una K, obteniendo así el consenso degenerado GCACGTKKK.

2.1.2. Representaciones basadas en matrices

La representación basada en matrices se construye a partir de un conjunto de subsecuencias. Las filas de la matriz representan a cada uno de los nucleótidos, y cada columna una posición de la secuencia. Se puede definir varios tipos de matrices. Supóngase que se han alineado las secuencias de abajo tabla 2.4, entonces definimos las siguientes matrices:

Código	Descripción
M	A o C
R	A o G
W	A o T
S	C o G
Y	C o T
K	G o T
V	A o C o G
H	A o C o T
D	A o G o T
B	C o G o T
N	A o C o G o T

Tabla 2.3: Código IUPAC

2.1.2.1. Matriz de ocurrencias

Una matriz de ocurrencias representa solo el conteo de los nucleótidos en cada posición, es decir, las columnas corresponden a la posición y las filas a los nucleótidos.

2.1.2.2. Matriz de Frecuencias

La matriz de Frecuencias se obtiene al aplicar la fórmula 2.1 a cada una de los elementos de la matriz de ocurrencias, con ella se genera la tabla 2.6:

$$f_{i,j} = \frac{n_{i,j}}{\sum_{i=1}^{A} n_{i,j} + k} \tag{2.1}$$

Donde

A Tamaño del alfabeto

Secuencia												Identificador
A	A	A	C	A	C	G	T	G	G	G	G	1
C	G	A	C	A	C	G	T	G	C	G	A	2
C	T	T	C	A	C	G	T	G	G	G	C	3
G	G	T	C	A	T	G	T	G	C	G	G	4
T	A	C	C	A	C	G	T	G	G	A	C	5
T	A	C	C	A	C	G	T	G	T	T	A	6
T	C	T	C	A	C	G	T	T	T	T	T	7
T	C	C	C	A	C	G	T	T	G	G	T	8

Tabla 2.4: Secuencias Alineadas

Ocurrencias

Pos	1	2	3	4	5	6	7	8	9	10	11	12
A	1	3	2	0	8	0	0	0	0	0	1	2
C	2	2	3	8	0	8	0	0	0	2	0	2
G	1	2	3	0	0	0	8	0	5	4	5	2
T	4	1	0	0	0	0	0	8	3	2	2	2
Sum	8	8	8	8	8	8	8	8	8	8	8	8

Tabla 2.5: Matriz de ocurrencias

$n_{i,j}$ Ocurrencias de la base i en la columna j de la matriz

k Pseudo peso

En algunas ocasiones un nucleótido podría no aparecer en el muestreo. Esto a veces es provocado por un muestreo pobre. Con la frecuencia $n_{i,j}$ igual a cero se generaría una división entre cero. Para corregir esto se introduce el pseudopeso

Frecuencias

Pos	1	2	3	4	5	6	7	8	9	10	11	12
A	0.13	0.38	0.25	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.13	0.25
C	0.25	0.25	0.38	1.0	0.0	1.0	0.0	0.0	0.0	0.25	0.0	0.25
G	0.13	0.25	0.38	0.0	0.0	0.0	1.0	0.0	0.63	0.5	0.63	0.25
T	0.5	0.13	0.0	0.0	0.0	0.0	0.0	1.0	0.38	0.25	0.25	0.25
suma	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Tabla 2.6: Matriz de Frecuencias

2.1.2.3. Matriz de Pesos de Posición

Con una matriz de pesos de Posición se trata de capturar información sobre qué significa que un nucleótido esté ubicado en determinada posición. Por este se ocupa el *background* para integrar a la formula la frecuencia con la que una base determinada suele aparecer en una posición. Una matriz de pesos de posición se obtiene mediante la siguiente fórmula:

$$W_{i,j} = \ln\left(\frac{f_{i,j}}{p_i}\right) \tag{2.2}$$

$$f_{i,j} = \frac{n_{i,j} + p_i k}{\sum_{i=1}^{A} n_{i,j} + k} \tag{2.3}$$

Donde

Pesos

Pos	1	2	3	4	5	6	7	8	9	10	11	12
A	-0.79	0.13	-0.23	-2.20	1.05	-2.20	-2.20	-2.20	-2.20	-2.20	-0.79	-0.23
C	0.32	0.32	0.70	1.65	-2.20	1.65	-2.20	-2.20	-2.20	0.32	-2.20	0.32
G	-0.29	0.32	0.70	-2.20	-2.20	-2.20	1.65	-2.20	1.19	0.97	1.19	0.32
T	0.39	-0.79	-2.20	-2.20	-2.20	-2.20	-2.20	1.05	0.13	-0.23	-0.23	-0.23
Sum	-0.37	-0.02	-1.02	-4.94	-5.55	-4.94	-4.94	-5.55	-3.08	-1.13	-2.03	0.186

Tabla 2.7: Matriz de Pesos de Posiciones

A = Tamaño del alfabeto

$n_{i,j}$ = Ocurrencias de la base i en la columna j de la matriz

p_i = Probabilidad apriori para la base i

$f_{i,j}$ = Frecuencia relativa de la base i en la posición j

k = Pseudo peso

2.1.3. Representación visual

A veces resulta complicado interpretar de inmediato las representaciones con cadenas o mediante matrices. Por esto, es necesario contar con una representación más sencilla de entender. Una representación visual ofrece una forma rápida de obtener información.

2.1.3.1. Sequence Logo

Ésta es la forma visual más utilizada para representar un conjunto de secuencias. Está basada en el concepto de contenido de información. Para obtener esta representación se calcula el contenido de información para cada columna de la Matriz de Pesos de Posición. Este valor corresponde a qué tan bien conservados se encuentran los elementos

en esa posición. Una columna tiene el máximo contenido de Información cuando contiene un solo nucleótido y su valor alcanza los dos bits, y tiene el menor contenido de información cuando contiene los cuatro nucleótidos en cantidades iguales, en este caso su contenido de infomación es de cero bits. El contenido de información para una matriz de frecuencias F se calcula mediante la siguiente fórmula:

$$IC_c = 2 + \sum_{n=A,C,G,T} p(f_{n,c}) log_2 p(f_{n,c}) \tag{2.4}$$

donde

- $F = f_{n,c}$ es el valor del nucleótido n, en la columna c, para la matriz de frecuencias.
- $p(f_{n,c}) = \frac{f_{n,c}}{N}$ donde N es el número de secuencias.

Un Sequence Logo, se representa como un conjunto de pilas de letras que corresponden a cada columna de F. La altura de la columna representa el contenido de información, mientras que la altura individual de cada una de las letras que forman la columna es proporcional a la distribución de su conteo en la columna. En la figura 2.1, se presenta el Sequence Logo, correspondiente a las secuencias de los ejemplos anteriores, como se puede observar las pilas de las posiciones 4, 5, 7, y 8 son las que posen una altura máxima, debido a que poseen un contenido de información de 2bits, al encontrarse completamente conservados. En las demás posiciones las pilas no alcanzan la altura máxima debido a que no se encuentran completamente conservados, por ejemplo, en la posición 9 se observa que los nucléotidos G y T se repiten un número parecido de veces ya que el tamaño de estas letras es casi el mismo, mientras que el IC es de aproximadamente 1 bit.

2.2. Reconocimiento de Patrones

El problema de identificación de elementos reguladores puede ser visto como un problema de Reconocimiento de Patrones debido a las características de las secuencias reguladoras. El descubrimiento de patrones en secuencias consiste en, dado un conjunto de

Figura 2.1: Sequence Logo

secuencias, encontrar patrones desconocidos que sean frecuentes, inesperados, o interesantes de acuerdo a cierto criterio. Un patrón puede ser representado como una expresión regular o como una matriz probabilista de pesos. El problema del descubrimiento de patrones puede ser dividido en tres supbroblemas:

1. Elegir el lenguaje apropiado para describir los patrones
2. Elegir la función de evaluación para comparar los patrones
3. Diseñar un algoritmo para identificar los patrones con los mejores puntajes

2.2.1. Tipos de patrones

Los tipos de patrones se pueden dividir en dos grandes grupos:

- Patrones Deterministas
- Patrones Probabilistas

En los patrones deterministas un patrón coincide o no con con alguna cadena, mientras que los patrones probabilistas son usualmente modelos probabilistas que asignan cada secuencia una probabilidad de que sean generados por el modelo.

Patrones Deterministas. Los patrones deterministas son una secuencias de caracteres en un alfabeto Σ, algunas de sus variantes son las siguientes:

Carácter Ambiguo. Un carácter ambiguo es un carácter que pertenece a un subconjunto de Σ. Es decir, un carácter ambiguo puede coincidir con cualquier elemento de este subconjunto. Normalmente los patrones de este tipo se representan encerrando en corchetes a los elementos del subconjunto, por ejemplo C [C,G] T. Los códigos IUPAC descritos anteriormente pueden usarse para representar un patrón de este tipo. Para el ejemplo anterior, de acuerdo al código correspondiente en IUPAC para C ó G, se puede sustituir [C,G] por S, entonces se puede representar el patrón como CSG.

Carácter Irrelevante. Este carácter puede ser emparejado con cualquier elemento. De acuerdo a la codificación IUPAC, se puede usar N para emparejar con cualquier nucleótido. A la secuencia de uno o varios caracteres irrelevantes se le llama brecha (gap).

Brecha flexible. Una brecha flexible se refiere a una brecha de longitud variable

Patrones Probabilistas. Matrices de Pesos de Posiciones. Son la forma más simple de una patrón probabilista. Una matriz de pesos de posición no contiene brechas.

2.2.2. Medidas de similitud

Las medidas de similitud se necesitan para comparar dos datos y saber qué tanto parecido existe entre ellos. El tipo de medidas de similitud depende del tipo de datos que se desean comparar. No se puede utilizar el mismo tipo de medidas para comparar datos numéricos que datos secuenciales, cada tipo de datos necesita sus propias medidas. Para este trabajo nos interesa saber qué tan parecidas son diferentes secuencias de ADN, las cuales son representados como cadenas de letras, para esto, se hace una revisión de algunas medidas de similitud que existen para medir la similitud entre cadenas.

Las medidas de similitud de cadenas se dividen en tres tipos:

Basadas en Palabra Esta medida considera a una cadena como un multiconjunto (o bolsa), de palabras .

Funciones de Distancia	d(S,T)
Manhattan	$\sum_{w \in L} \lvert\phi_w(S) - \phi_w(T)\rvert$
Canberra	$\sum_{w \in L} \frac{\lvert\phi_w(S) - \phi_w(T)\rvert}{\phi_w(S) + \phi_w(T)}$
Mankowski	$\sqrt[k]{\sum_{w \in L} \lvert\phi_w(S) - \phi_w(T)\rvert^k}$
Hamming	$\sum_{w \in L} sgn \lvert\phi_w(S) - \phi_w(T)\rvert$
Chebyshev	$max_{w \in L} \lvert\phi_w(S) - \phi_w(T)\rvert$
Euclidiana	$\sqrt{\sum_{w \epsilon L} (S_w - T_w)^2}$

Tabla 2.8: Funciones de Distancia

La Divergencia de Kullback-Leiber también conocido como entropía relativa, es una medida de similitud basada en palabra. Esta medida es usada en este trabajo para comparar la similitud entre dos grupos.

$$KLD(X, Y) = \frac{1}{2}\left(\sum_{a \epsilon A} X_a \log\left(\frac{X_a}{Y_a}\right) + \sum_{a \epsilon A} Y_a \log\left(\frac{Y_a}{X_a}\right)\right) \tag{2.5}$$

Basadas en Carácter Estas medidas cuentan el número de caracteres que tienen en común un par de cadenas. Estas medidas a su vez pueden dividirse en dos: las funciones de distancia, y las funciones de similitud. En las funciones de distancia, dado un par de cadenas, S y T, se les asigna un número real r, mientras más pequeño sea este número mayor será la similitud entre S y T. Mientras que en las funciones de similitud, mientras más grande sea el número, mayor será el parecido. Sea Σ de tamaño N. El contenido de una secuencia S puede modelarse como el conjunto de todas las secuencias traslapadas w tomadas de un lenguaje finito $L \subset \Sigma^*$. Y sea la función $\Phi_w(S)$ la frecuencia de w en S. En la tabla 2.8 se presentan algunas funciones de distancia.

Híbridas Estas medidas combinan los métodos basados en palabra y en carácter.

Estas funciones de similitud proporcionan diversas maneras de comparar secuencias.

La elección de una de estas funciones depende del problema que se va a resolver. Para el problema de identificación de secuencias reguladoras, se utilizará la técnica de agrupamiento. está depende significativamente de la medida de similitud utilizada, con diferentes medidas se pueden obtener diferentes resultados.

2.3. Agrupamiento

Los algoritmos de agrupamiento se encargan de organizar instancias en grupos significativos de acuerdo a sus características, de manera que los datos en el mismo grupo tengan una gran similitud, y tengan una baja similitud con datos de otros grupos. Sin embargo, obtener grupos significativos es algo vago, ya que depende de la aplicación. En este trabajo se decidió utilizar una técnica de agrupamiento por dos razones principales, para disminuir el espacio de búsqueda, y para encontrar patrones biológicamente interesantes.

2.3.1. Tipos de Agrupamiento

Existen diferentes tipos de agrupamientos: particional, jerárquico, exclusivo, difuso, total y parcial [5]. El agrupamiento particional consiste en una división de las instancias, mientras que en el agrupamiento jerárquico los grupos contienen subgrupos de sus elementos. En el agrupamiento exclusivo, los datos solo pueden pertenecer a un solo grupo, en contraste con el agrupamiento difuso, en el que todos los datos pertenecen a todos los grupos en cierto grado dado por una membresía que va de 0, cuando no pertenecen en nada, a 1, cuando pertenecen totalmente. Por último, se encuentran los agrupamientos totales, en que todos los elementos son asignados a un grupo, y el agrupamiento parcial, en el que es posible que algunos elementos queden sin ser asignados.

En este trabajo se utilizarán dos tipos de agrupamiento: Agrupamiento Jerárquico, y K-mean. En la siguiente sección se describen estos algoritmos.

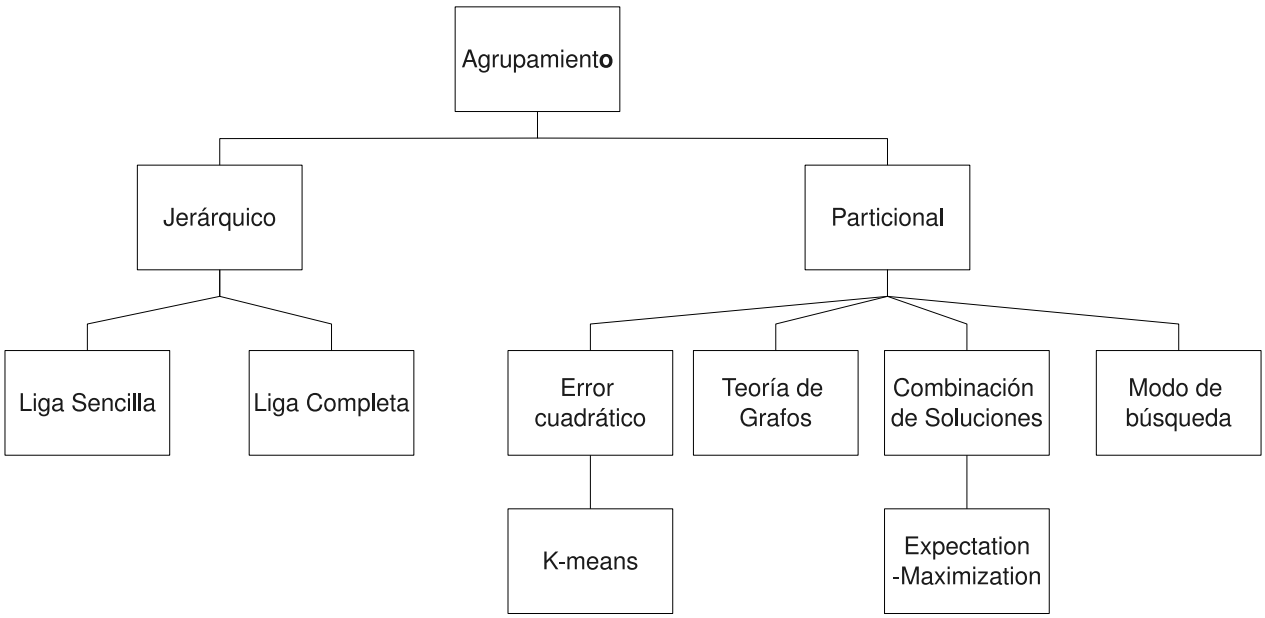

Figura 2.2: Tipos de Agrupamiento [5]

2.3.2. K-means

El algoritmo k-means es un algoritmo de agrupamiento particional que divide un conjunto de instancias en k grupos. La característica principal de este algoritmo es que cada grupo está representado por un centroide. Este algoritmo comienza con la elección de k centroides iniciales. Después, cada instancia es asignada al centroide con el que mantiene la menor distancia, y con estos conjuntos de instancias se forman k grupos. Posteriormente se recalcula cada centroide de los grupos, basándose en los elementos actuales del grupo. Se repite la asignación y actualización del centroide hasta que no haya un cambio en él. A continuación se presentan los pasos del algoritmo.

Algoritmo 1 Algoritmo Básico k-means

1: Seleccionar k puntos como centroides iniciales
2: Formar k grupos asignando cada instancia al centroide más cercano
3: Recalcular los centroides con los nuevos elementos
4: Repetir los pasos 2 y 3 hasta que no haya cambio en el centroide

Las principales desventajas del algoritmo k-means, son las siguientes:

- Se debe conocer el número de grupos
- El resultado depende de los centros iniciales
- Grupos vacíos

En el problema del descubrimiento de secuencias se desconoce el número de grupos que se pueden encontrar, por lo que será necesario desarrollar una estrategia para determinar el número de grupos.

2.3.3. Agrupamiento Jerárquico

En el agrupamiento jerárquico se construye una jerarquía de grupos. Los grupos contienen subgrupos, o hijos. Existen dos estrategias para crear la jerarquía, ascendente (bottom-up), y descendente (top-down). A los algoritmos que utilizan la estrategia ascendente, se les conoce también como aglomerantes. Se empieza con un grupo para cada dato, y estos grupos se van uniendo con más elementos de acuerdo a un criterio de similitud, hasta tener un solo grupo que los contenga a todos los elementos. Los algoritmos divisivos utilizan la estrategia descendente, y estos comienzan con un solo grupo, y lo van dividiendo en subgrupos hasta alcanzar cierto criterio. Algunas ventajas de los algoritmos jerárquicos son las siguientes [5] :

1. El nivel de granularidad es flexible.
2. No debe conocerse el número de grupos de antemano.

Debido a que este tipo de algoritmos no dependen del tipo de atributos y puede usarse cualquier tipo de medida de similitud se pueden aplicar a este problema, ya que podrá adaptarse al tipo de datos que se usaran en este trabajo. Además, se puede aprovechar que su granularidad es flexible para superar el problema del desconocimiento del número de grupos.

Su principal desventaja es que su criterio de terminación es vago.

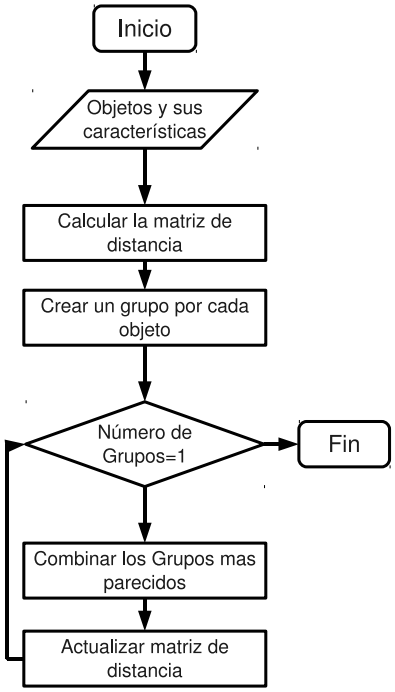

Figura 2.3: Algoritmo de Agrupamiento Jerárquico Aglomerante

En la figura 2.3 se presenta los pasos del algoritmo de agrupamiento jerárquico aglomerante. Estos son los siguientes:

Algoritmo 2 Algoritmo de Agrupamiento Jerárquico Aglomerante

1: Calcular la matriz de cercanía entre cada grupo
2: Combinar los grupos que estén más cercanos
3: Recalcular la matriz de cercanía
4: Repetir los pasos 2 y 3 hasta que se forme un solo grupo

2.3.4. Agrupamiento Jerárquico Divisivo

En el agrupamiento jerárquico divisivo se inicia con un solo grupo. Y en cada iteración se van dividiendo los grupos hasta alcanzar cierto criterio. Hay dos formas de dividir los

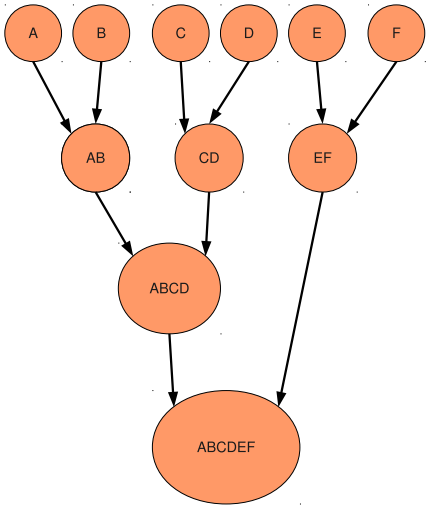

Figura 2.4: Agrupamiento Jerárquico Aglomerante

grupos:

- Polythetic: Usa todas las variables para realizar las divisiones sucesivas.
- Monothetic: Solo usa una variable para realizar las divisiones.

Algoritmo 3 Algoritmo de Agrupamiento Jerárquico Divisivo

Entrada: Las instancias a agrupar

Salida: Todas las instancias agrupadas

1: Colocar todas las instancias en un grupo
2: **mientras** número de elementos de todos los grupos sea mayor que 1 **hacer**
3: Elegir un grupo para dividir
4: Dividir grupo
5: **fin mientras**

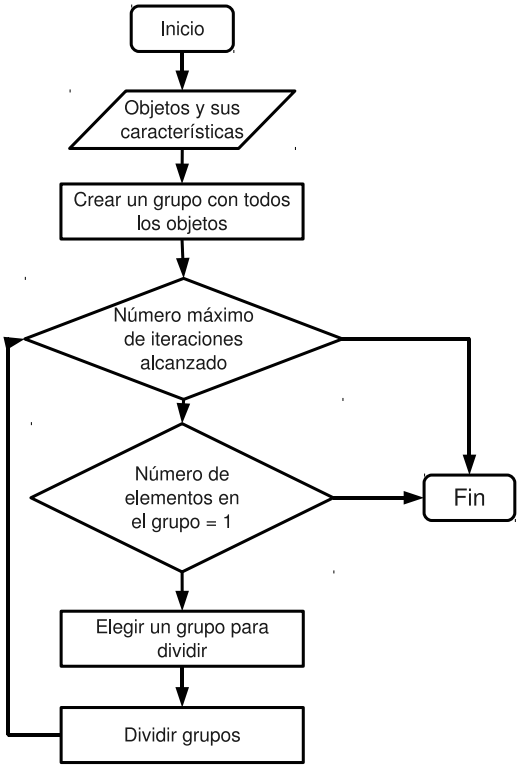

Figura 2.5: Agrupamiento Jerárquico Divisivo

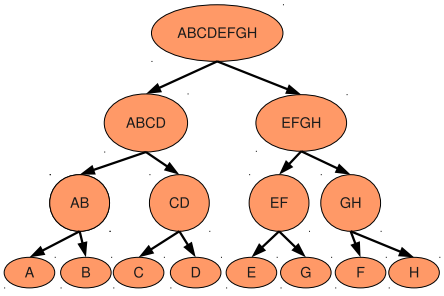

Figura 2.6: Algoritmo de Agrupamiento Jerárquico Divisivo

Capítulo 3

Trabajo relacionado

En el ADN existen muchos elementos con diferentes funciones. Se le conoce como motivo a una secuencia patrón de nucleótidos que está distribuida en el genoma y que tiene un significado biológico. Las secuencias reguladoras son un tipo especial de motivos. Se han desarrollado varios trabajos con el objetivo de encontrar diferentes tipos de motivos. Este problema se ha tratado de resolver con varios enfoques, biológicamente se pueden encontrar tres tipos de aproximaciones:

1. **Un solo gen, múltiples especies**. Esto significa que se buscará en las regiones hacia arriba de un mismo gen, pero en diferentes especies de organismos. A esto se le conoce como huella filogenética (phylogenetic footprinting).

2. **Múltiples genes, una sola especie** La búsqueda se realiza usando los genes coregulados de un mismo organismo.

3. **Múltiples genes, múltiples especies** Es una combinación de los dos enfoques anteriores, es decir, se busca en los genes coregulados, pero también se usa la huella filogenética.

En este trabajo se usará el segundo enfoque. Lo que significa que buscará las secuencias reguladoras para un solo organismo.

Los distintos algoritmos que se han desarrollado para dar una solución a este problema pueden dividirse en dos clases, los métodos basados en palabras y los métodos probabilistas.

3.1. Métodos Basados en Palabra.

Los métodos basando en palabra, también conocidos como métodos enumerativos, enumeran exhaustivamente todas las posibles palabras de cierta longitud que pueden formarse, después calculan su significancia estadística para elegir los motivos que serán propuestos. La complejidad computacional de los métodos basados en palabra es de: $O(NMA^eL^e)$ donde N es el número de secuencias, *M* es su longitud, A es el tamaño del alfabeto, L es la longitud del motivo, y *e* es el número de errores permitidos en un emparejamiento [12]. Estos métodos garantizan optimalidad global, pero solo son apropiados para motivos cortos [2]. Una desventaja de estos métodos es que a menudo producen muchos resultados espurios (falsos positivos). Muchos de estos métodos utilizan diversas técnicas de indexado [17] para acelerar las búsquedas.

En 2004 Pavesi et al. presentaron Weeder Web [14], que es una interfaz web al algoritmo de descubrimiento automático de motivos en un conjunto de regiones reguladoras del ADN Weeder. El algoritmo Weeder se basa en que la búsqueda exhaustiva puede ser acelerada significativamente si las secuencias de entrada son preprocesadas y organizadas. Para hacer esta organización ocupan la estructura de datos de árbol sufijo.

Pavesi et al., hicieron modificaciones a su anterior algoritmo [14] y en 2007 introdujeron [15], este algoritmo toma como entrada una secuencia de referencia S, y cualquier número de secuencias homólogas (que provienen de organismos con un ancestro común), después ejecuta los siguientes pasos:

1. Cada oligo (secuencias de nucleótidos normalmente menor a 20bp) de cierto tamaño de la secuencias de referencia es comparada con la secuencia homóloga

2. Las coincidencias encontradas que no exceden cierto umbral de substitución son

calificadas con una medida que toma en cuenta la conservación de la secuencias, y se almacena la que obtiene el puntaje más alto.

3. Los puntajes de los oligos son transformados en puntajes relativos de acuerdo a los puntajes promedio obtenidos por oligos del mismo tamaño.

4. Se fusionan los oligos con mayor puntaje.

En 2006 Lawrence y Ajay, presentaron MaMF [3], este algoritmo recibe un conjunto de promotores y un motivo de entrada de longitud *l*. El objetivo de MaMF es maximizar el valor de una función de evaluación que da un mayor puntaje a los motivos que están mejor conservados a lo largo de los promotores y que están poco representados en el genoma objetivo. La salida de MaMF es una lista de motivos ordenada de acuerdo a su puntaje. Este algoritmo es determinista, y depende de una estrategia de indexado para optimizar su resultado. Esta consiste en crear un índice en que todos los n-mers (secuencias de n nulceótidos) encontrados por una secuencia de entrada, con lo que se consigue identificar ubicaciones dentro de una secuencia que contiene un nmer dado en un tiempo constante. Dados los índices de dos secuencias y un n-mer, se pueden identificar todos los alineamientos entre las dos secuencias que comparten ese nmer. Para esto se crea una tabla para todos los alineamientos de las secuencias de longitud *l* que comparten un n/-mer. Se enumeran todos los pares de secuencias de la tabla y se evalúan, guardando las 1000 secuencias con el puntaje más alto para ser usadas como semillas en el paso de generación de motivos. Este paso utiliza una estrategia voraz en la que se construyen motivos a partir de las semillas obtenidas, e iterativamente, se añaden secuencias al motivo siempre y cuando maximicen el puntaje del mismo. Esta iteración continua hasta alcanzar un umbral N, el número de secuencias de entrada.

3.2. Métodos Probabilistas

Kon, Holloway y DeLisi presentaron en 2007 SVMotif, un algoritmo de aprendizaje computacional basado en máquinas de vectores de soporte. Con este algoritmo se trata

de identificar motivos utilizando una asociación estadística de las secuencias con las interacciones conocidas de los Factores de Transcripción. Una característica importante de este algoritmo, es que para realizar el aprendizaje utiliza ejemplos tanto negativos como positivos. Como ejemplos negativos se utilizan promotores donde se sabe que no existen enlaces con los Factores de Transcripción; mientras que los ejemplos positivos son los promotores donde si existen estos enlaces. El algoritmo trabaja de la siguiente manera, los datos de entrada consisten de vectores de características de los genes, la entrada incluye tanto ejemplos positivos como negativos. Se trata de que este conjunto de datos este balanceado. Estos datos son enviados a el clasificador SVM, el cual proporciona un vector de dirección $\tilde{w}$, pesado. Los candidatos obtenidos en el paso anterior son reducidos mediante Recursive SVM". Los k-mers obtenidos son agrupados para poder así, formar las matrices de pesos, en este caso se utilizan las matrices PWM (position weight matrix). Para elegir los mejores motivos se calculan los puntajes de las matrices, así como la entropía (exclusivamente de las columnas), y el número de secuencias que contiene cada grupo. Este algoritmo está sujeto a un tamaño específico de las secuencias por lo que no es muy bueno para determinar el tamaño adecuado de los motivos. [9]

En [20] se utiliza una representación de matriz de pesos, las subsecuencias son codificadas en una matriz binaria tal que $e(k) = [a_{i,j}]_{4Xk}, a_{i,j} = 1$ if $T_j = V_i$, y $a_{i,j} = 0$. Esta representación facilita la aplicación de la distancia de Hamming para medir la distancia entre dos secuencias. Se propone un algoritmo llamado Miscluster, la idea de este algoritmo es crear grupos a partir de una submuestra de las secuencias, para después agregar las secuencias faltantes mediante un enfoque jerárquico. Una característica importante de este algoritmo, es que cada vez que actualiza los grupos los analiza para comprobar su utilidad. Si los grupos proporcionan poca información, no seguirán siendo procesados. Debido a la naturaleza de los datos, no es posible definir un centroide para los grupos, por lo que en este trabajo en lugar de centroide se definirá un prototipo para cada grupo. Este

prototipo está definido como:

$$M = \frac{i}{p} \sum_{r=1}^{p} e(k_r) = [f(i,j)]_{4xK} \tag{3.1}$$

Donde, f(i,j) es la frecuencia del nucleótido i en la posición j. Otra característica de este algoritmo es que utiliza una evaluación para seleccionar los mejores grupos al inicio, y así evitar grupos con secuencias repetitivas como AAAAAA o CGCGCGCGC. Este rankeo se obtiene al calcular el puntaje Maximum a Posteriori, MAP, de cada uno de los grupos iniciales. Se proponen tres reglas heurísticas para el procesamiento de los grupos.

Lones et al. [11] presentaron una solución a este problema mediante un algoritmo genético. Este algoritmo realiza la búsqueda de los motivos en los promotores de genes co-expresados, es decir, los genes que son regulados por la misma secuencia reguladora. Las secuencias son representadas mediante una matriz de frecuencia de posiciones, la cual es posteriormente transformada en una matriz de pesos de posición con probabilidades logarítmicas. La función de aptitud del motivo se calcula como la diferencia entre la media del mejor puntaje de coincidencia sobre los datos de los genes co-expresados, y el mejor puntaje medio de las coincidencias sobre el conjunto de datos base. Para mantener la diversidad en la población, se utiliza un algoritmo de de agrupamiento que realiza particiones en la población, con el fin de realizar los apareamientos entre individuos de diferentes poblaciones. Para comparar los elementos a ser agrupados se utiliza la distancia Euclidiana. Esta se puede obtener gracias a que, para cada elemento, se calcula un vector de características que describen la distribución de los tetranucleótidos en la PFM. El algoritmo se puede resumir como sigue: Se inicializa la población generando aleatoriamente PFMs con frecuencias uniformemente distribuidas para cada una de las cuatro bases. Se agrupa la población con el algoritmo mencionado anteriormente. Cada grupo debe proporcionar por lo menos una solución hijo. Se crean nuevas PFMs mediante mutación y cruza. Se aparean los individuos con una mayor aptitud. Se iteran los pasos después de la inicialización.

Nimwegen et al. [19], crearon un método de agrupamiento probabilista de secuencias.

Representaron las secuencias como una matriz de pesos. Esta matriz toma en cuenta el nivel de energía del enlace de un Factor de Transcripción (TF) a un segmento de la secuencia del ADN. Con esta matriz, pueden calcular la probabilidad de que una secuencia sea un sitio de enlace (binding site) para el TF. Utilizan un muestreo de Monte Carlo para realizar los agrupamientos. Se hicieron experimentos en una base de datos donde ya se sabe que deben existir 29 grupos. El método, en sus diferentes versiones, descubre entre 16 y 29 grupos. Y su tasa de falsos positivos fue de cero [19].

Jensen, Shen y Liu [6], presentaron un trabajo para predecir genes coregulados. Para conseguirlo, combinaron métodos de filogenia, descubrimiento de motivos y agrupamiento de motivos. El agrupamiento bayesiano jerárquico fue el método elegido para inferir el agrupamiento de motivos. Los motivos fueron representados como una matriz de conteo. Implementaron el modelo mediante el algoritmo de Gibbs Sampling, el cual, iterativamente muestrea parámetros desconocidos, y decide a qué grupo debe asignarse cada motivo. El método permite que haya una variación en el tamaño de los motivos. El número de motivos encontrados varia dependiendo de si se utilizó un tamaño variable o fijo de los motivos. El método mostró buenos resultados, aunque obtiene varios grupos con poco significado biológico [6].

Middendorf, Kundaje, Shah, Freund, Wiggins, y Leslie presentan MEDUSA [13] , un método para aprender modelos de motivos de los sitios de enlace de los factores de transcripción (Transcription Factors Binding Sites) incorporando secuencias promotoras y datos de la expresión de un gen. Cada modelo de un motivo puede ser representado como una secuencia de longitud k (k-mear) un dimer, o una PSSM (Position Specific Score Matrix). MEDUSA realiza un agrupamiento de motivos jerárquico [13].

Kelarev, Kang y Steane [8] adaptaron los algoritmos k-means, y NN, para que agruparan secuencias de nucleótidos. Los puntajes obtenidos mediante alineamiento fueron utilizados como medida de similitud. Los experimentos fueron realizados con un con-

junto de datos derivado de las regiones ITS (Internal Transcribed Spacer). Se realizaron pruebas con varias especies del subgénero Eucalyptus. Los grupos que se desean obtener ya son conocidos en la literatura. Se hicieron varias pruebas con diferentes tipos de alineamientos. k-means mostró un porcentaje de éxito de entre 60 y 70 %, mientras que NN se desempeñó mejor al obtener un porcentaje de éxito de entre 77 y 80 % [8].

Karabulut e Ibrikci presentaron un método para descubrir Sitios de Enlace de Transcripción (Transcription Binding Sites, TBS). Este método está basado en un algoritmo C-Means difuso. La forma de representar los datos es mediante una matriz de peso de posición, sus elementos están conformados por el logaritmo de la frecuencia con la que aparece un nucleótido en la posición i, sobre la frecuencia con que aparece el mismo nucleótido en el *background*. Aquí el *background* está formado por las regiones intergenómicas. El algoritmo logró predecir los motivos conocidos en las secuencias intergenómicas GAL4, CBF1 y GCN4. Sin embargo, utiliza una longitud fija para los motivos, es decir, no es capaz de decidir automáticamente cuál debe ser su tamaño, y es necesario decidir el número de grupos que se deben formar. [7].

En la siguiente figura se presenta un resumen de las características de los métodos mencionados anteriormente. En la primera columna se indica el nombre del algoritmo. En la segunda columna se muestra el método de búsqueda utilizado. La tercera columna indica la forma en la que se representaron los datos. Por último, en la cuarta columna se indican los autores y el año.

En la tabla 3.2 se encuentran algunas ventajas y desventajas de los enfoques existentes.

Algoritmo	**Principio de Operación**	**Representación**	**Autores**
WordUP	Enumeración	Cadena	Pesole et al. 1992
MEME	Expectation maximization	PSM	Bailey and Elkan 1995
AlignACE	Gibbs sampling	PSM	Roth et al. 1998
Oligo-Analysis	Enumeración	Cadena	van Helden et al. 1998
Dyad-Analysis	Enumeración	Cadena	van Helden et al. 2000
Bioprospector	Gibbs sampling	PSM	Liu et al. 2001
Weeder	Enumeración	Cadena	Pavesi et al. 2001
MotifSampler	Gibbs sampling	PSM	Thijs et al. 2001
MITRA	Árbol prefijo/Grafos	Cadena	Eskin and Pevzner 2002
MDScan	Algoritmo Voraz	PSM	Liu et al. 2002
MOPAC	Enumeración	Cadena	Ganesh et al. 2003
FMGA	Algoritmo Genético	PSM	Liu et al. 2004
MUSA	Biclustering	PSM	Mendes et al. 2006
GAME	Algoritmo Genético	PSM	Wei and Jensen 2006
Svmotif	SVM	PSM	Kon et al. 2007
Miscluser	Agrupamiento Jerárquico	PSM	Wang y Lee 2008
MCEMDAD	Em-MonteCarlo	PSM	Chenpeng 2008
DMOPSH	Híbrido	PSM	Jong y Seungjin 2009
Metodo Propuesto	Agrupamiento Jerárquico	PSM y cadenas	Dulce 2010

Tabla 3.1: Trabajos relacionados

Basados en palabra	**Probabilistas**	**Aprendizaje computacional**
Motivos cortos	Motivos largos	Trabajan bien con diferentes tamaños de motivos.
Enumeración exhaustiva	Métodos probabilistas	SVM, SOM, Clustering, GA,
Representados como cadenas	Representado como matrices de pesos	Ambas Representaciones
Pueden producir resultados espurios	Pueden quedar atrapados en mínimos locales	No producen tantos resultados espurios
Más eficientes en Eukariotes	Más eficientes en Prokariotes	Pueden ser utilizados para los dos tipos de organismos.

Tabla 3.2: Características de los diversos enfoques

Capítulo 4

Método propuesto

Como se ha visto los métodos existentes para la identificación de secuencias reguladoras tienen varias desventajas, como son el estar limitados a un tamaño fijo, el uso de una sola representación para los datos, o trabajan con un número pequeño de secuencias. El método que se propone explorará secuencias de distintos tamaños, se aprovecharán las ventajas de una representación como cadenas y una representación basada en matrices, además, no se verá limitado por el número de secuencias de entrada permitidas, ya que se podrán analizar las secuencias cercanas correspondientes a todos los genes del genoma.

4.1. Metodología

El método propuesto está compuesto de tres pasos generales (ver figura 4.1).

Pre-procesamiento. En este paso se obtienen las regiones de interés del genoma. Estas son las regiones en donde se sabe pueden encontrarse las secuencias reguladoras.

Agrupamiento. Se propone un algoritmo de Agrupamiento Jerárquico Divisivo. Este algoritmo da como resultado grupos con los patrones encontrados en el conjunto de secuencias dado.

Evaluación. Los patrones obtenido son evaluados para seleccionar aquellos que tiene probabilidades de ser secuencias reguladoras.

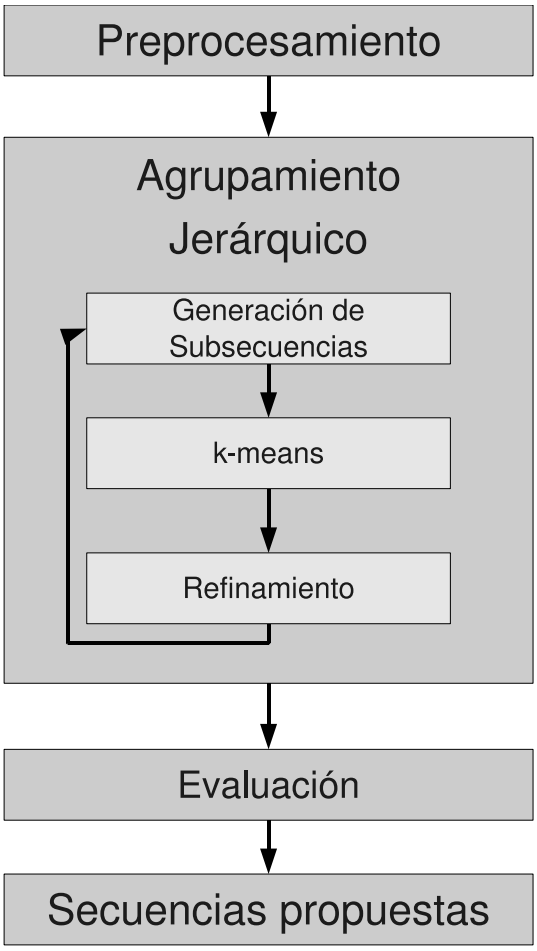

Figura 4.1: Solución Propuesta

4.2. Preprocesamiento

Se desarrolló una herramienta para extraer las regiones intergénicas. Esta herramienta recibe un archivo en formato fasta. Este archivo contiene el genoma completo del organismo del que se desean extraer las regiones intergénicas. El archivo fasta contiene un listado de los nombres de los genes, así como la posición de inicio y terminación de estos. A partir de estos datos, se extraen 220 bases hacia arriba de cada gen. Y se guardan estas regiones, así como el gen al que están asociadas, y su posición global en el genoma.

Otra herramienta para el preprocesamiento de estos datos se encarga de segmentar las regiones intergénicas, es decir, toma subcadenas de estas. Las secuencias reguladoras pueden variar de tamaño dependiendo del organismo al que pertenecen. Por esta razón, es deseable poder iniciar con diferentes longitudes de secuencias, ya que, para elementos que se sabe que sus secuencias reguladoras son largas, será conveniente iniciar la búsqueda con secuencias largas, para que no exista el riesgo de que alguna secuencia reguladora quede cortada, mientras que para secuencias reguladoras que, se sabe son pequeñas, se puede iniciar con secuencias de una menor longitud. Esta herramienta creará subsecuencias de una longitud l de una forma escalonada, es decir, supóngase que se obtuvo la siguiente región, CAGTCGATCGATCGA, y se desea obtener subsecuencias con una longitud de 10 caracteres, entonces se generan las 5 siguientes subsecuencias:

```
CAGTCGATCGATCGA
CAGTCGATCG
 AGTCGATCGA
  GTCGATCGAT
   TCGATCGATC
    CGATCGATCG
     GATCGATCGA
```

Se almacena cada una de estas subsecuencias junto con la posición que tienen en el genoma, y el nombre del gen al que se encuentran asociadas.

4.3. Búsqueda de Secuencias Reguladoras

4.3.1. Representación de Datos

En este trabajo se utilizaron dos representaciones diferentes. Una representación basada en cadenas, y otra representación basada en matrices. La primera representación se utiliza en los algoritmos de agrupamiento propuestos. Los centroides de los grupos permiten el uso de símbolos IUPAC. La representación mediante matrices es principalmente

utilizada en la evaluación de los grupos. Se utilizan matrices de pesos de posición para representar a los grupos formados.

4.3.2. Medidas de similitud

Los algoritmos de agrupamiento dependen mucho de la función de similitud utilizada. Ya que es una gran cantidad de datos los que se van a agrupar es muy importante que la función de similitud pueda comparar eficientemente las secuencias para que las secuencias que se agrupen no queden agrupadas con otras secuencias con las que no podrían tener alguna relación biológica. Se ha definido una medida de similitud para evaluar la similitud entre las secuencias. Esta está basada en el número de posiciones donde la coincidencia de las secuencias se mantiene continua.

Nuestra medida de similitud se calcula como el número de coincidencias, multiplicado por el número de veces en que la coincidencia es continua. Supóngase que se tienen las siguientes secuencias:

1) CGATGCATGCACTGCATCCG
2) CGATGACCAAGTACGATCCG
3) GGCTAAGCGATTCCAAGCGG

Las tres secuencias coinciden en 10 nucleótidos. Sin embargo, biológicamente, las secuencias uno y dos son más parecidas puesto que tienen una coincidencia mayor en lugares adyacentes. Se puede calcular la similitud entre 1 y 2 como: $(5\times5)+(5\times5)=50$, mientras que la similitud entre las secuencias 2 y 3 es $(1\times1) + (1\times1) + (1\times1) + (1\times1) + (1\times1) + (1\times1) + (1\times1) + (1\times1) + (1\times1) + (1\times1) = 10$. En la figura 4.2 se presenta otros ejemplos. Cuando mientras más elementos tienen en común las secuencias sin que haya huecos, mas crece el valor de la función. en las ultimas dos comparaciones se puede observar como, aunque entre la secuencia S1 y S3 haya mucho parecido pues coinciden

en la mitad de los elementos, su valor no supera a la similitud entre la secuencia S1 y S2, las cuales, aunque no tienen tantos elementos en la misma posición, los elementos donde coinciden estén juntos, lo que biológicamente es mas significativo.

Algoritmo 4 Algoritmo Similitud

1: $i \leftarrow 1$
2: $adyacencia \leftarrow 0$
3: $similitud \leftarrow 0$
4: **mientras** $i < longitud$ **hacer**
5: **si** $S_1(i) == S_2(i)$ **entonces**
6: $adyacencia \leftarrow adyacencia + 1$
7: **si no**
8: $similitud \leftarrow similitud + (adyacencia^2)$
9: $adyacencia \leftarrow 0$
10: **fin si**
11: $i \leftarrow i + 1$
12: **fin mientras**

4.3.3. Algoritmo de Agrupamiento Jerárquico

Sea *S* el conjunto de subsecuencias de longitud *l*, se desean encontrar los patrones existentes en dichas secuencias que tengan una mayor probabilidad de ser secuencias reguladoras. El algoritmo de agrupamiento jerárquico propuesto permite ir reduciendo el espacio de búsqueda, al mismo tiempo que identifica los patrones existentes.

El algoritmo consiste de una serie de pasos iterativos que se repiten hasta alcanzar grupos con patrones que tengan una alta probabilidad de ser secuencias reguladoras. Estos pasos son los siguientes:

1. El procedimiento se inicia aplicando el algoritmo k-means para secuencias, el cual se describirá en la siguiente sección. Este algoritmo particiona el conjunto inicial de

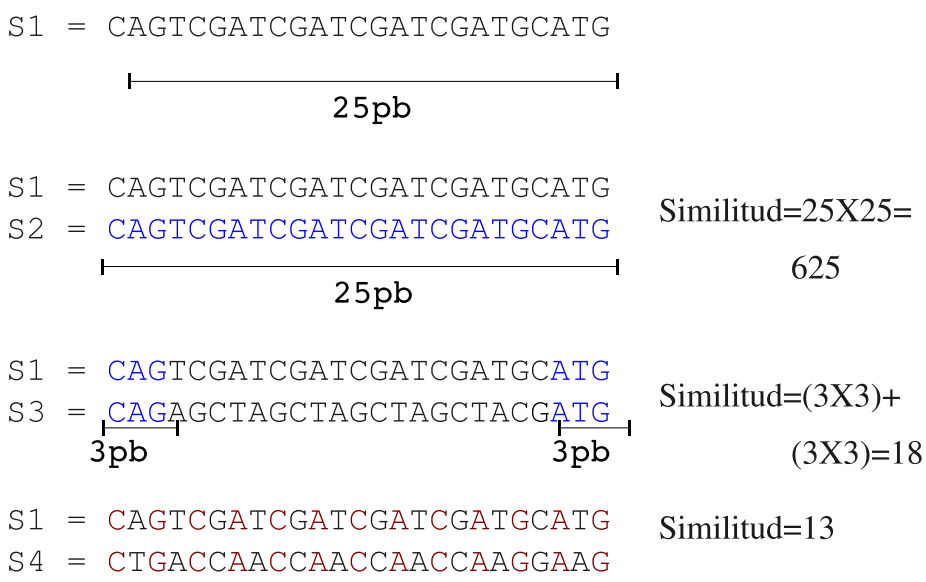

Figura 4.2: Similitud

datos, en un número de k grupos. Este algoritmo descarta todas aquellas secuencias que no consigan sobrepasar un umbral de similitud con las demás secuencias.

2. Una vez que los grupos han sido creados, sigue un análisis de éstos. Con este fin, se define para cada grupo: una matriz de frecuencias de posición.

$$PFMF = [f_{n,c}]. \tag{4.1}$$

Donde $f_{n,c}$, es la frecuencia con que aparece el nucleótido *n*, en la columna *c*.

También se define una matriz de pesos

$$PWM = \frac{i}{p}\sum_{r=1}^{p} e(k_r) = [f(i,j)]_{4xK} \tag{4.2}$$

Esta matriz representa la media de cada nucleótido en una posición determinada.

3. En cada nivel de la jerarquía se hace un refinamiento de los grupos. Con este propósito, se calculan las siguientes medidas.

$$KLD(X,Y) = \frac{1}{2}\left(\sum_{a\epsilon A} X_a \log\left(\frac{X_a}{Y_a}\right) + \sum_{a\epsilon A} Y_a \log\left(\frac{Y_a}{X_a}\right)\right) \tag{4.3}$$

La divergencia Kullback-Lieber 4.3 o entropia cruzada, es utilizada para medir el parecido que existe entre grupos. Se fusionan los grupos que produzcan la menor entropia cruzada. Si, al realizar esta fusión, la entropia disminuye, se mantiene la fusión. Si no, se dejan los grupos originales.

$$IC = \sum_{j=1}^{l}\sum_{i=1}^{4} f(i,b)log(f(i,b)/p(i)) \tag{4.4}$$

El contenido de información ayuda a determinar qué tan bien conservado se encuentra un grupo. Mientras más alto sea el valor de la entropía, mejor conservado se encuentra el grupo. Se elige al grupo que tenga el menor IC. Si el contenido de información es menor a un umbral *e* y el número de elementos es mayor a nMin, que es el número mínimo de elementos que debe tener un grupo, entonces el grupo se divide en dos. Si al dividirlo, el contenido de información de alguno de los dos grupos formados aumenta, se mantiene la división, y se elimina el otro grupo. De otra manera se mantienen los grupos originales.

El procedimiento para la división de los grupos es el siguiente:

a) Seleccionar grupos con menor IC

b) Calcular la similitud media de los elementos del grupo

c) Los elementos que tengan similitud mayor o igual a la similitud media se añaden al grupoA

d) Los elementos que tengan similitud menor a la similitud media se añaden al grupoB

e) Se recalculan los centroides para estos dos nuevos grupos

f) Se reasignan los elementos de acuerdo a los nuevos centroides

g) Si el IC aumenta en alguno de lo grupos, se mantiene la división, si no se descarta

4. Una vez terminado este refinamiento de grupos se procede al siguiente nivel en la jerarquía. Para esto, se crean subsecuencias de una longitud menor que las secuencias originales. Estas subsecuencias no contendrán traslapes entre si. Por ejemplo, si se tiene la secuencia CGATGCTAGCATGCTACGTC, y el siguiente nivel en la jerarquía tendrá un tamaño 4, entonces se generaran las siguientes subsecuencias: CGAT, GCTA, GCAT, GCTA, CGTC.

5. Se repiten los pasos anteriores hasta que las subsecuencias llegan a a un tamaño mínimo determinado.

Algoritmo 5 Algoritmo de Agrupamiento Jerárquico Divisivo

```
1: mientras tamaño de las subsecuencias > tamaño_Mínimo hacer
2:     agrupamiento()
3:     refinamiento()
4:     generarSubsecuencias()
5: fin mientras
```

4.3.4. Algoritmo k-means para secuencias

En cada nivel del algoritmo de Agrupamiento Jerárquico Divisivo se crearan los grupos, con el algoritmo de k-means. Este algoritmo consiste de los siguientes pasos:

1. Se selecciona una secuencia aleatoriamente, esta secuencia será el centroide del un grupo C_i

2. Se calcula la similitud entre el centroide y cada una de las secuencias restantes. Se añaden al grupo las secuencias que tengan una distancia menor a un umbral μ

3. Se repiten el paso 1 y 2 hasta que se hayan asignado todas las secuencias

4. Se actualizan los grupos hasta que no haya cambio en los centroides.

Para mejorar la diversidad entre los grupos, se eligen centroides con una distancia alta entre ellos. Se utiliza la distancia de Hamming para medir la separación entre secuencias, y solo se eligen como centroides a las secuencias que tengan una distancia mayor a cierto umbral entre ellas.

Algoritmo 6 Modificación Algoritmo K-means para Secuencias

```
1: Elegir una secuencia sAleatoria aleatóriamente
2: Crear grupo G
3: G.centroide = sAleatoria
4: Para todas las secuencias restantes
5: si (distancia (S, C.centroide)> umbralSimilitud) entonces
6:     C.agrega(S)
7: fin si
8: actualizacion()
9: refinamiento()
```

El procedimiento para actualizar los grupos es el siguiente:

1. Se recalcula el centroide de los grupos. Este está formado por la cadena constituida por nucleótidos que tengan la frecuencia más alta en cada una de las posiciones. En caso de que dos o más nucleótidos aparezcan con la misma frecuencia en la misma posición, se utilizará una letra perteneciente al código IUPAC. También se calcula con consenso, que es la cadena formada por nucleótidos que tengan la frecuencia más alta en cada una de las posiciones.

$$centroide_i = MAX(F[b, i]) \tag{4.5}$$

donde, F[b,i], es la frecuencia del nucleótido b en la posición i.

2. El siguiente paso consiste en reasignar las secuencias a los grupos. Para cada una de las secuencias se calcula su similitud contra todos los centroides y la distancia

con los consensos. Se añade al grupo con el que tenga la mayor similitud solo si su distancia es menor a cierto umbral.

3. Se repiten estos dos pasos hasta que ya no haya cambio en los centroides ó hasta que se alcance cierto número de iteraciones, esto último se hace, ya que, debido al gran número de secuencias a asignar, les puede llevar a los centroides un gran número de iteraciones para converger.

   Un vez que se ha terminado con la actualización se hace un pequeño refinamiento. Para cada uno de lo grupos formados se calcula la similitud media y la desviación estándar. Se eliminan los elementos de los grupos cuya similitud sea menor a la media menos la desviación estándar.

Algoritmo 7 actualización

```
1: mientras Exista cambio en los centroides hacer
2:    Recalcular centroide para cada grupo C
3:    Para todas las secuencias restantes
4:    si (similitud (S, C.centroide)> umbralSimilitud) entonces
5:       si (distancia (S, C.centroide)< umbralDistancia) entonces
6:          C.agrega(S)
7:       fin si
8:    fin si
9: fin mientras
```

4.3.5. Evaluación

Una vez que las secuencias han sido agrupadas se deben evaluar estos grupos para determinar cuáles de ellos tienen más probabilidades de contener secuencias reguladoras. Para hacer esto se utiliza la función de evaluación propuesta por Wang y Lee en [20]

$$Score = \frac{-ln(|N_c|)}{k}\left[E(M) + \frac{1}{N_c}\sum_{S\epsilon C} lnp_0(S)\right] \tag{4.6}$$

Algoritmo 8 refinamiento

1: $simMedia \leftarrow 0$
2: $desviacionEstandar \leftarrow 0$
3: **para** Todos los grupos g_i **hacer**
4: $simMedia \leftarrow \frac{\sum_{j=0}^{n} sim(e_j)}{n}$
5: $desviacionEstandar \leftarrow \sqrt{\frac{\sum_{j=0}^{n} (sim(e_j))^2}{n}}$
6: **si** $sim(e_j) < simMedia - desviacionEstandar$ **entonces**
7: Se elimina e_j del grupo g_i
8: **fin si**
9: **fin para**

Donde N_c es el número de elementos que contiene el grupo, E(M) es la entropía de Shannon del grupo; k, es la longitud de las secuencias, y p_0 es la probabilidad de la secuencia S.

Esta puntuación mide la conservación del grupo con respecto al *background*, que en este caso es el genoma completo.

Capítulo 5

Experimentos y Resultados

En este capítulo se describen los experimentos así como el conjunto de datos utilizado para evaluar el método propuesto. Y por último se presentan los resultados obtenidos.

5.1. Descripción de datos

El método propuesto se probó en la identificación de secuencias reguladoras para dos organismos, E. coli, que ha sido ampliamente estudiado, por lo que ya se tienen identificados varios elementos reguladores, a menudo es usado como *benchmark* para la evaluación de métodos de descubrimiento de secuencias reguladoras, y el organismo Bacillus Subtilis, que, aunque también ha sido estudiado, no es tan utilizado para experimentación.

El organismo E.coli es una bacteria, su ADN consta de aproximadamente 5 millones de nucleótidos. Se utilizó la base de datos RegulonDB. En esta base de datos se encuentran registradas las secuencias reguladoras descubiertas hasta ahora, las interacciones que existen entre ellas, algunas predicciones realizadas por otros métodos desarrollados, y las secuencias promotoras donde se encuentran las secuencias reguladoras.

El organismo Bacillus Subtilis consta de 4214630 pares de bases y posee 4,234 genes. La secuencia del genoma de este organismo fue obtenida de la página del National Center for Biotechnology Information, NCBI, www.ncbi.nlm.nih.gov/. Además se utilizó la base de datos DBTBs [4], para obtener algunas secuencias reguladoras que ya han sido

identificadas. Es importante señalar que está base de datos se ha formado utilizando 947 referencias, utilizando métodos tanto biológicos como computacionales para identificar los elementos reguladores.

5.2. Parámetros

Se llevaron a cabo varios experimentos haciendo variaciones de los siguientes parámetros.

Número máximo de iteraciones para el algoritmo k-means. El algoritmo k-means itera hasta que su centroide no experimenta ningún cambio entre una iteración y otra. Sin embargo, el número de secuencias que se agruparan es muy alto, esto provocará que la convergencia a un centroide sea lenta. Por esta razón se pide un número máximo de iteraciones. Así, el algoritmo se detendrá cuando alcance el número máximo de iteraciones, o bien, los centroides converjan. Elegir el número máximo de iteraciones, afecta el desempeño del algoritmo. si las iteraciones son muy pocas los grupos no quedarán bien conservados.

Umbral de distancia Este umbral se refiere a la distancia máxima que puede existir entre una secuencia y el centroide de un grupo para que la secuencia pueda ser añadida a dicho grupo. Si el número es muy pequeño se garantiza mayor parecido entre las secuencias, pero, secuencias que si deben pertenecer al grupo, podrían quedar fuera. Con un número muy grande, se corre el riesgo de tener grupos con elementos sin tanto parecido.

Valor de la entropía cruzada necesario para fusionar grupos. Este valor indica que tanto deben parecerse los grupos para que pueda realizarse la fusión.

Longitud de las secuencias iniciales. En la sección anterior se mencionó que la longitud inicial de las secuencias esta relacionada con la longitud máxima de las secuencias conocidas. La longitud inicial siempre debe ser mayor a la longitud conocida para

evitar secuencias incompletas. Con secuencias más largas, la cantidad de comparaciones entre secuencias disminuye, pero también disminuye la probabilidad de encontrar coincidencias entre secuencias, con secuencias pequeñas aumenta esta probabilidad, pero las comparaciones también aumentan.

Longitud mínima de las secuencias finales. De la misma manera que la longitud inicial de las secuencias depende del conocimiento que se tenga de las secuencias reguladoras, la longitud mínima también está en función de este conocimiento. Si no se posee ningún conocimiento acerca de la composición de las secuencias reguladoras del elemento buscado es recomendable empezar con secuencias de longitud relativamente larga, y terminar con secuencias cortas. Si no se proporciona una longitud mínima el algoritmo se detiene cuando la longitud de las secuencias es de 5pb.

5.3. Experimentos

5.3.1. Experimentos con Secuencias Conocidas

Para probar la efectividad del método se midió su desempeño en la identificación de secuencias ya conocidas. El método recibe un conjunto de secuencias intergénicas de una longitud de 220 pb. Con estas secuencias se crean las subsecuencias sobre las que se realizará la búsqueda. Este método es sensible al tamaño inicial de las secuencias, por lo que el método se probará con diferentes longitudes para analizar cómo se comporta el método con cada una de ellas.

Las tablas contienen el puntaje obtenido (puntaje), el contenido de información (IC), la entropía (entropía), el número de elementos de cada grupo (NE), el número de elementos reguladores encontrado en ese grupo (NF), el número de elementos reguladores obtenidos entre el número de elementos en el grupos (PE), y por último el número de elementos encontrados sobre el número de elementos reguladores conocidos (PC). El objetivo de los experimentos realizados es comprobar si el método propuesto es capaz de encontrar secuencias reguladoras. Con este fin se analizan diversos grupos de secuencias

reguladoras que ya han sido previamente identificadas.

5.3.2. CRP

El conjunto de elementos CRP ha sido ampliamente usado en la evaluación de los métodos de descubrimiento de secuencias reguladoras. Se tiene conocimiento de 161 secuencias de este grupo, y se encuentran ubicadas en las regiones cercanas de 18 genes. La secuencia consenso para representar este grupo está formada por dos núcleos donde existe la mayor conservación: TGTGA - NNNNNN - TCACA Sin embargo hay muchas variaciones entre estas secuencias. Aunque la longitud de la secuencia consenso es de 16 bp, las secuencias conocidas tienen una longitud de 24 bp. En los resultados de algunos métodos es frecuente que solo se identifique alguno de estos nucleos [9].

Se usará esta conjunto de prueba para ilustrar la forma en que el método propuesto va disminuyendo el largo de las secuencias. En la figura 5.1 se encuentra la representación gráfica de algunos de los grupos obtenidos por el método propuesto. El primer grupo es el padre del segundo, el segundo del tercero, y el tercero del cuarto. En cada nivel existían otros grupos, pero se muestran los que mantienen una mayor conservación de los elementos. Como se puede observar, el tamaño de las secuencias va disminuyendo. No está indicado, pero el número de elementos que tiene el grupo también disminuye debido a que los elementos de grupo padre se vuelven parte de diferentes grupos. Sin embargo, se puede observar como, conforme se baja de nivel, se hace más claro el patrón. Esto indica que el método es efectivo al determinar en donde seccionar las secuencias, manteniendo la región donde la similitud entre ellas es mayor.

5.3.3. MYOD, CREB, MEF2

En [20] valoran su método con secuencias tomadas de distintos organismos. Se utilizaron algunas de ellas para evaluar el método.

MYOD Son 21 secuencias, distribuidas en 18 regiones intergénicas. Las bases de datos usadas corresponden a las regiones intergénicas totales. Su longitud es variable de

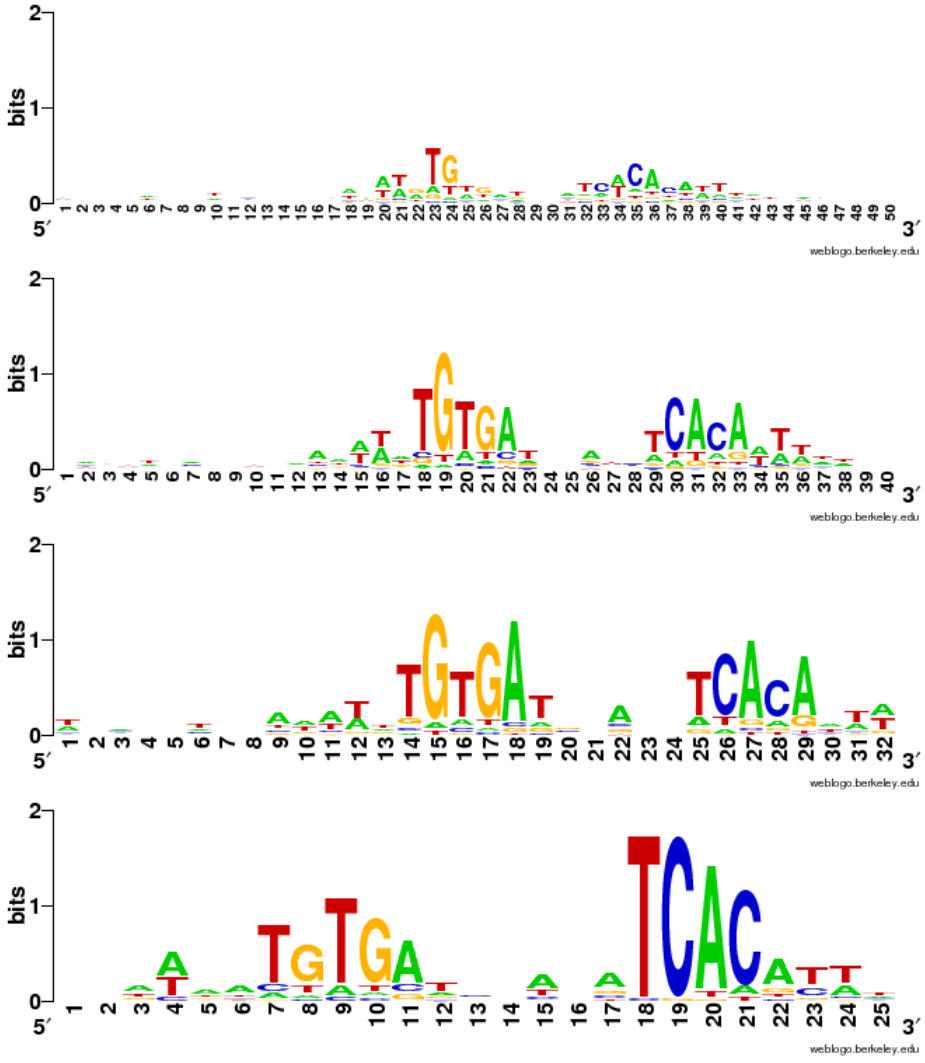

Figura 5.1: Jerarquía Obtenida

TF	MEME		AlignACE		Sombrero		Mis		M.P.	
	P	R	P	R	P	R	P	R	P	R
CREB	0.88	0.59	0.66	0.52	0.43	0.83	0.8	0.73	0.5	0.11
CRP	0.76	0.81	0.98	0.83	0.87	0.73	0.99	0.64	0.8	0.79
MYOD	0.38	0.23	0.31	0.34	0.32	0.5	0.43	0.57	0.35	0.29
MEF2	0.8	0.92	0.87	0.86	0.22	0.35	0.99	0.81	0.85	0.37
prom	0.71	0.64	0.71	0.64	0.46	0.6	0.8	0.69	0.63	0.39

Tabla 5.1: Comparación de algoritmo. Porcentaje de acierto

aproximadamente 500pb cada una.

CREB Son 19 secuencias, su longitud varia de 9pb a 31pb. Están distribuidas en 17 regiones intergénicas. Las bases de datos usadas corresponden a las regiones intergénicas totales. Su longitud es variable de aproximadamente 500pb cada una.

MEF2 Son 17 secuencias, su longitud varia de 8pb a 16pb. Están distribuidas en 16 regiones intergénicas. Las bases de datos usadas corresponden a las regiones intergénicas totales. Su longitud es variable de aproximadamente 500pb cada una.

5.3.4. FurR

Se experimentó con la secuencia conocida Fur, el consenso de esta secuencia es GATAATGATAATCATTATC, se conocen 27 instancias de ésta. De estas 27, 3 tienen una longitud de 42, 2 de 48 y las restantes de 50. Estas secuencias conocidas se alinearon con el programa ClustalW [10]. En la figura 5.2 se encuentra su representación gráfica obtenida con el programa WebLogo [1]. Para verificar si el método propuesto era capaz de identificar estas secuencias se ejecutaron un total de 12 pruebas. Se ejecutaron 4 pruebas, para 3 diferentes longitudes iniciales, 80bp, 50bp, 40bp. En la tabla 5.2, se presentan los mejores grupos de las 14 ejecuciones.

En las ejecuciones utilizando una longitud inicial de 40pb se obtuvieron 6 grupos

P	id	nivel	Puntaje	IC	entropía	Tamaño	NE	NF	PE	PC
20	0	1	23.61	0.34	1.15	80	17	6	0.35	0.2500
18	0	1	22.57	0.31	1.14	80	15	5	0.33	0.2083
18	3	0	22.44	0.32	1.15	80	15	7	0.47	0.2917
17	0	0	21.98	0.34	1.13	80	14	5	0.36	0.2083
16	0	2	21.96	0.38	1.1	80	14	6	0.43	0.2500
15	0	1	21.89	0.32	1.14	80	14	5	0.36	0.2083
14	3	0	21.31	0.31	1.13	80	13	5	0.38	0.2083
13	0	1	21.3	0.4	1.11	80	13	7	0.54	0.2917
12	6	0	20.69	0.44	1.08	80	12	6	0.5	0.2500
11	0	1	20.68	0.32	1.12	80	12	5	0.42	0.2083
10	0	0	20.68	0.36	1.08	80	12	6	0.5	0.2500
9	6	0	20.64	0.37	1.09	80	12	5	0.42	0.2083
8	5	0	20.63	0.33	1.12	80	12	6	0.5	0.2500
7	1	0	14.55	0.73	0.88	64	6	3	0.5	0.1250
6	5	0	12.96	0.73	0.7	64	5	5	1	0.2083
5	6	0	11.07	0.73	0.68	64	4	2	0.5	0.0833
4	5	0	8.9	0.89	0.62	64	3	1	0.33	0.0417
3	3	0	5.74	1.42	0.4	64	2	1	0.5	0.0417
2	0	1	5.69	1.14	0.36	64	2	2	1	0.0833
1	3	0	5.62	1.39	0.42	51	2	1	0.5	0.0417

Tabla 5.2: Mejores Grupos para FUR

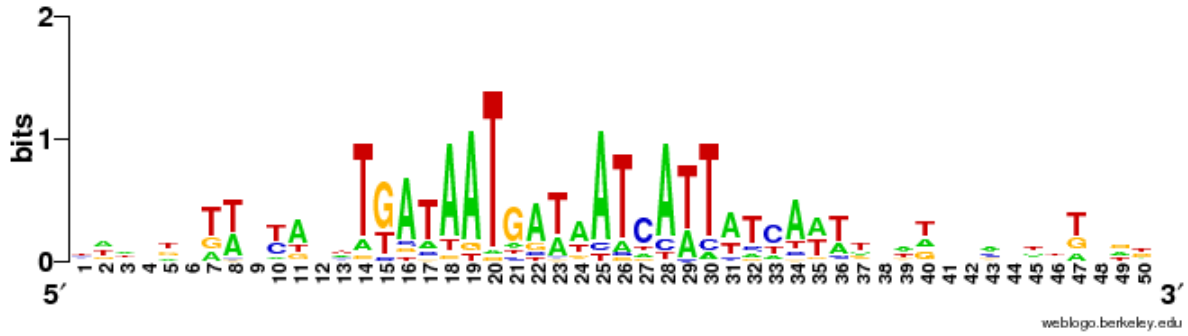

Figura 5.2: SequenceLogo FuR

iniciales en las 3 ejecuciones. El número de elementos de estos grupos va de 6 a 15. Para una longitud inicial de 50pb se obtuvieron en dos ejecuciones 5 grupos y en la tercera 4 grupos iniciales. El número de elementos va de 5 a 20. Por último, con un inicio de 80pb, se obtuvieron 5, 4 y 6 grupos iniciales. Su número de elementos estuvo entre 5 y 17. El grupo con el puntaje más alto contiene 17 secuencias, de estas 17, 6 corresponden a las 27 secuencias conocidas, esto significa el 25 % del total de secuencias buscadas. En la figura 5.3 se presenta el Logo de este grupo. Abajo se encuentra la comparación entre el consenso conocido y el consenso obtenido para este grupo.

```
-----------GATAATGATAATCATTATC--------------------------------
GAGGATAAACCCCGAATTGAGAATCATTCTCAAAAAAAAACATGACATAGAAAAGAACGAGAAG
```

Otro grupo interesante es el grupo en la posición 6. Este grupo contiene 5 elementos, y los 5 contienen a las secuencias conocidas. Estas secuencias representan el 20 % del total de secuencias conocidas. El Logo de este grupo se encuentra en la figura 5.4, se puede observar el mayor parecido con la secuencias consenso entre las posiciones 21 y 37.

```
-----------------GATAATGATAATCATTATC--------------------------
AGGTAAAATGGTATATTCTTAATTGATAATGATTCTCAATTACAACCTTGACATAGAAATAAAC
```

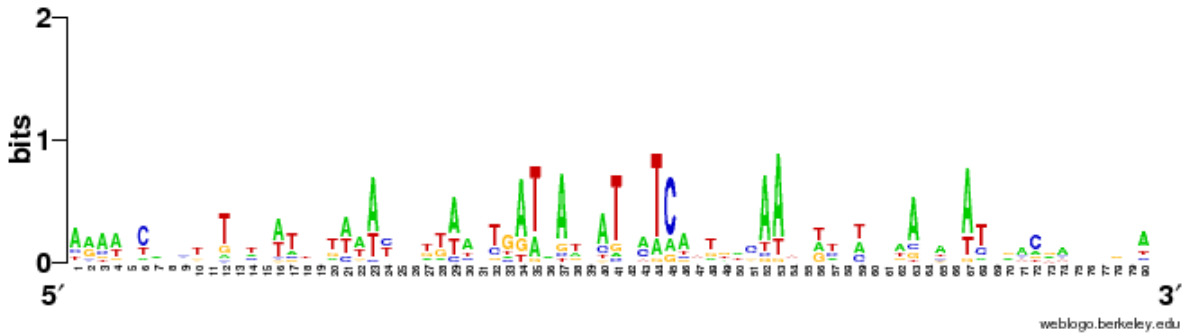

Figura 5.3: SequenceLogo Grupo3

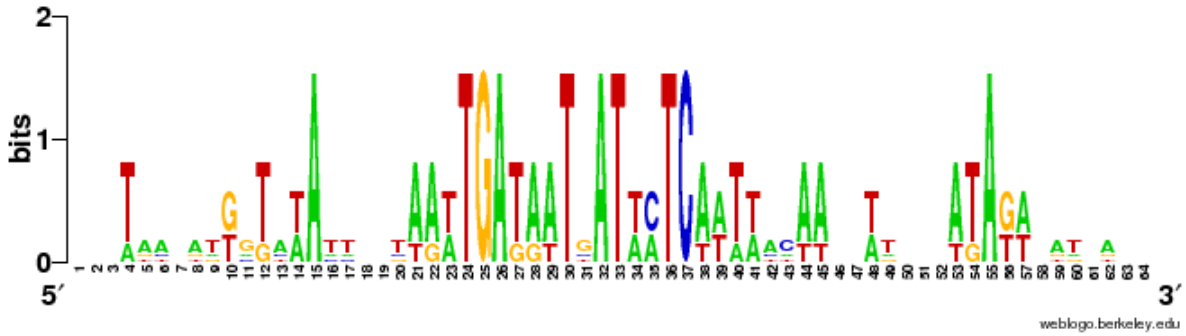

Figura 5.4: SequenceLogo Grupo0

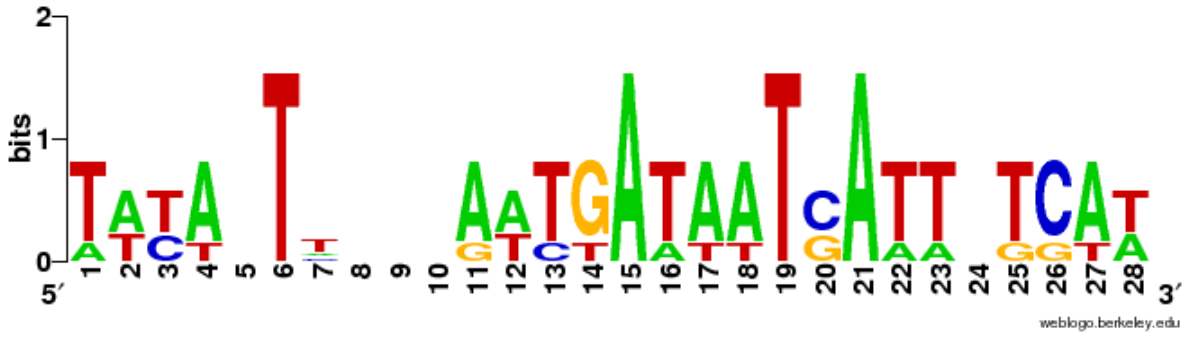

Figura 5.5: SequenceLogo FuR Grupo5

5.3.5. SigW

Para el siguiente conjunto de experimentos se utilizaron las secuencias correspondientes a los genes corregulados por la secuencia conocida como SigW. La secuencia consenso de este elementos es TGAAACN(16)CGTA. Su representación gráfica se encuentra en la figura 5.6. Se conocen 34 secuencias. La longitud de estas secuencias va de 38pb a 53bp. Para la identificación de estas secuencias se ejecutaron varías series de experimentos. Las subsecuencias iniciales fueron de longitudes de 100pb, 80pb, y 50pb. En la tabla **??**, se presentan los 5 mejores elementos de las diferentes longitudes obtenidas.

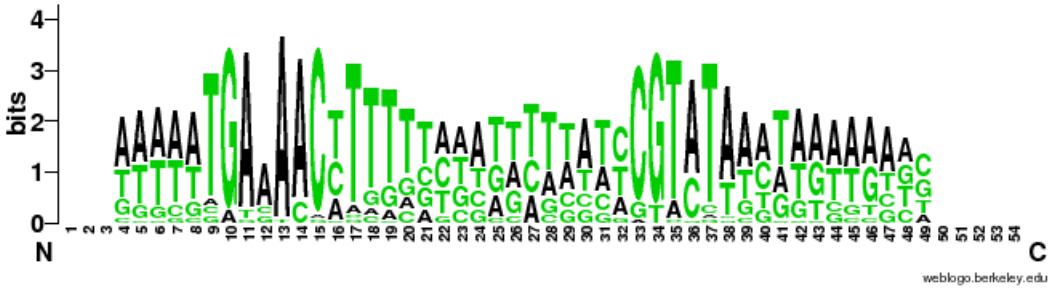

Figura 5.6: Sequence Logo sigW conocido

En las ejecuciones utilizando una longitud inicial de 50pb se obtuvieron entre 8 y 7 grupos iniciales. Estos grupos contienen en promedio 32 secuencias. Para una longitud inicial de 80pb se obtuvieron 7, 8 y 10 grupos iniciales en las tres ejecuciones. El número de elementos está entre 20 y 30. En el caso del inicio con subsecuencias de 100pb se obtuvieron entre 4 y 6 grupos. En número de elementos de estos grupos varió entre 5 y 21. Se observó que cuando se inició con secuencias de una longitud de 100pb, o mayor, la cantidad de elementos encontrados es baja, al igual que la conservación entre grupos. Esta cantidad de secuencias aumenta al generar grupos iniciales de secuencias con longitudes de 80pn, y se mantiene en los subgrupos generados de estos, con secuencias de entre 50pb y 70pb. La cantidad de secuencias encontradas disminuye al utilizar secuencias iniciales de 50pb. En las figuras 5.7, 5.8, 5.9 se presentan los Sequence Logo de los mejores grupos obtenidos con diferentes longitudes.

Alineación del consenso conocido y el grupo con mayor puntaje para las secuencias

P	id	nivel	Tamaño	Puntaje	IC	entropía	NE	NF	PE	PC
20	2	0	100	∞	0.33	1.16	16	6	0.38	0.18
19	2	0	100	∞	0.36	1.14	14	5	0.36	0.15
18	0	0	100	∞	0.3	1.17	17	5	0.29	0.15
17	0	0	100	∞	0.53	1.02	7	2	0.29	0.06
16	5	0	100	∞	0.36	1.14	14	4	0.29	0.12
15	11	0	80	27.9	0.28	1.19	29	15	0.52	0.44
14	5	0	80	27.67	0.27	1.2	28	13	0.46	0.38
13	14	0	80	27.27	0.27	1.19	27	15	0.56	0.44
12	4	0	80	27.02	0.25	1.2	26	11	0.42	0.32
11	2	0	80	26.74	0.24	1.22	25	10	0.4	0.29
10	3	1	64	25.82	0.34	1.15	24	10	0.42	0.29
9	2	1	64	25.63	0.32	1.16	24	12	0.5	0.35
8	0	1	64	25.57	0.33	1.15	24	8	0.33	0.24
7	0	1	64	24.58	0.32	1.14	21	11	0.52	0.32
6	0	1	64	24.38	0.3	1.18	20	10	0.5	0.29
5	1	2	51	24.4	0.31	1.16	22	9	0.41	0.26
4	3	2	51	23.74	0.42	1.1	21	2	0.1	0.06
3	0	2	51	22.86	0.41	1.1	19	0	0	0
2	1	2	51	22.62	0.45	1.07	18	1	0.06	0.03
1	1	2	51	22.59	0.44	1.07	18	1	0.06	0.03

Tabla 5.3: Mejores Grupos para SigW

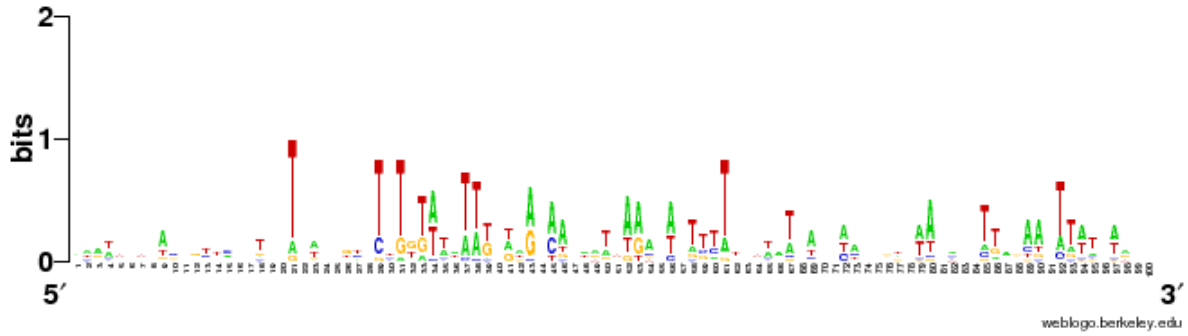

Figura 5.7: Sequence Logo mejor grupo SigW longitud 100

de longitud 80

```
----------------------------TGAAACN---------------CGTA-------------------------
AAGAAATTAAAAATTTTTTAAAAAAAAATGAAACCTTATTCTAATATAAACAGAATATATAAAAGAAAAAAAAAAAAGG
```

Alineación del consenso conocido y el grupo con mayor puntaje para las secuencias de longitud 60

```
------------TGAAACN---------------CGTA--------------------------
TTTAAAAAAAAATGAAACCTTATTCTAATATATCCGGAAAATATAAAAAAAAAAAAAAAAAGC
```

Alineación del consenso conocido y el grupo con mayor puntaje para las secuencias de longitud 50

```
-----TGAAACN---------------CGTA--------------------
AAAAAAGAAACCTTTTAATAAGTATATCATAAAATGTAAAAAAAACAAAAT
```

Alineación del consenso conocido y el grupo con el mayor número de elementos encontrados

```
-----------------------TGAAACN---------------CGTA---------------
TATAAAAATTTTTTACAACAAAATGAAACCTTTAAATAACTAAACCGTAATATTAAAAAAGAAA
```

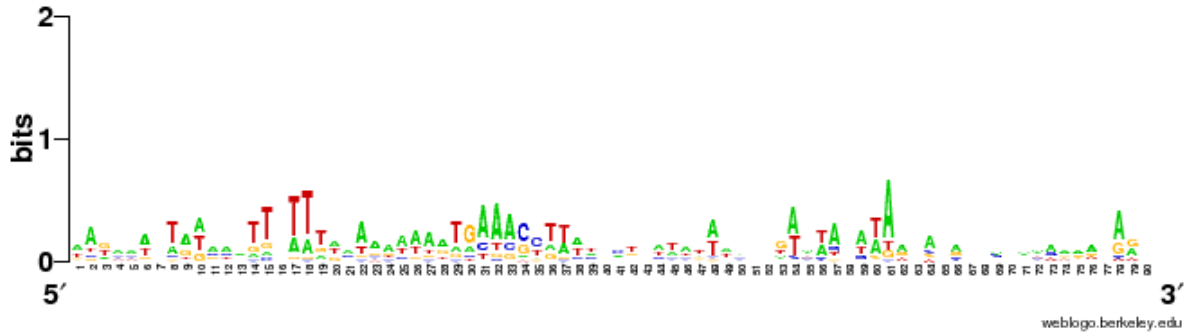

Figura 5.8: Sequence Logo mejor grupo SigW longitud 80

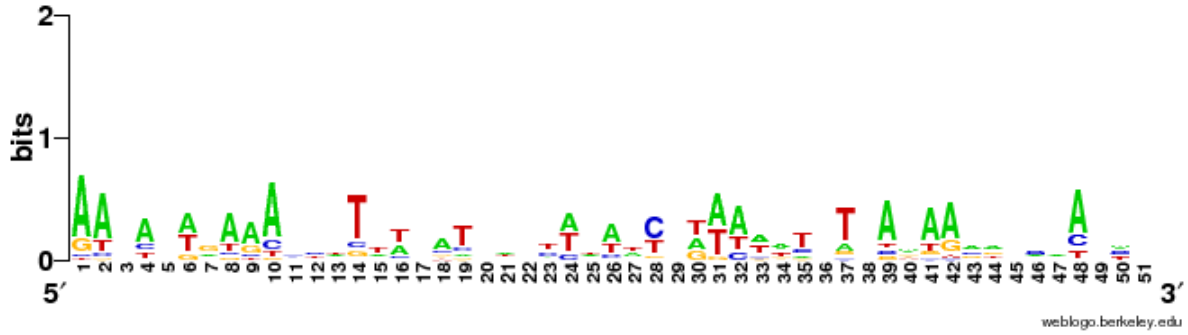

Figura 5.9: Sequence Logo mejor grupo SigW longitud 60

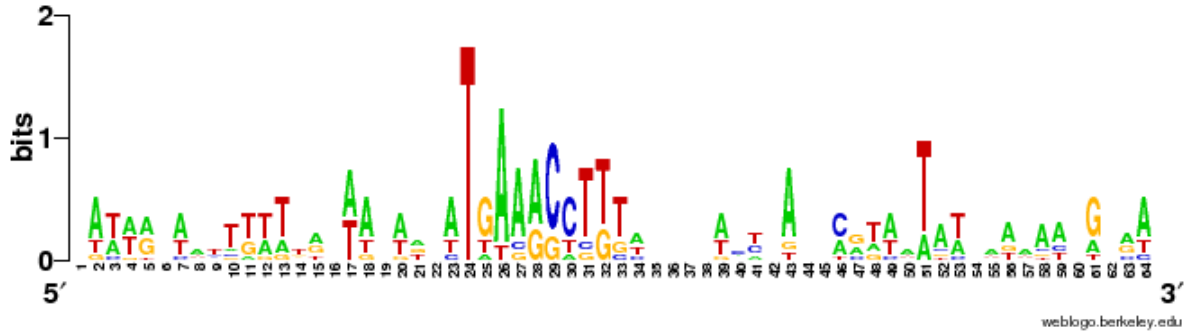

Figura 5.10: Sequence Logo Grupo con mayor número de elementos encontrados

5.3.6. SigD

La siguiente secuencia a utilizar es sigD. Su consenso es TAAA(-35)-N15-GCCGATAT(-10). La figura 5.11 corresponde a su Sequence Logo. En la base de datos con la que se está trabajando existen 30 secuencias identificadas. Sus longitudes están entre 47pb y

id	nivel	Tamaño	Puntaje	IC	entropía	NE	NF	PE	PC
3	0	80	25.52	0.26	1.18	22	7	0.32	0.23
4	0	80	25.95	0.25	1.2	23	6	0.26	0.2
3	0	80	25.55	0.28	1.18	22	6	0.27	0.2
5	0	80	25.52	0.27	1.17	22	6	0.27	0.2
1	0	80	25.18	0.26	1.18	21	6	0.29	0.2
0	0	80	25.99	0.26	1.19	23	5	0.22	0.17
0	0	80	25.58	0.26	1.19	22	5	0.23	0.17
5	0	80	25.53	0.27	1.17	22	5	0.23	0.17
4	0	80	25.15	0.25	1.18	21	5	0.24	0.17
1	0	80	25.88	0.27	1.17	23	4	0.17	0.13
2	1	64	22.4	0.34	1.12	16	3	0.19	0.1
2	1	64	21.77	0.36	1.11	15	3	0.2	0.1
0	1	64	19.32	0.5	1.04	11	3	0.27	0.1
4	1	64	18.44	0.44	1	10	3	0.3	0.1
0	1	64	18.42	0.48	1.01	10	3	0.3	0.1
0	1	64	18.37	0.43	1.03	10	3	0.3	0.1

Tabla 5.4: Mejores Grupos para SigD

58pb. Las longitud de las secuencias iniciales en los experimentos fue de 100pb, 80pb, y 50pb.

Los mejores resultados se observaron cuando se utilizaron secuencias iniciales de 80pb. En la tabla 5.4 se encuentran estos resultados. En la figura 5.12 se encuentra el Sequence Logo para el primer grupo de la tabla. Debido a que la secuencia consenso está formada por dos núcleos, TAAA, GCCGATAT, se formaron grupos con secuencias de menor tamaño a la secuencia consenso, que contienen alguna de estas dos subsecuencias. En las figuras 5.13, 5.14 se encuentran los sequenceLogo de los grupos en donde se identificaron los fragmentos TAAA y GCCGATAT respectivamente.

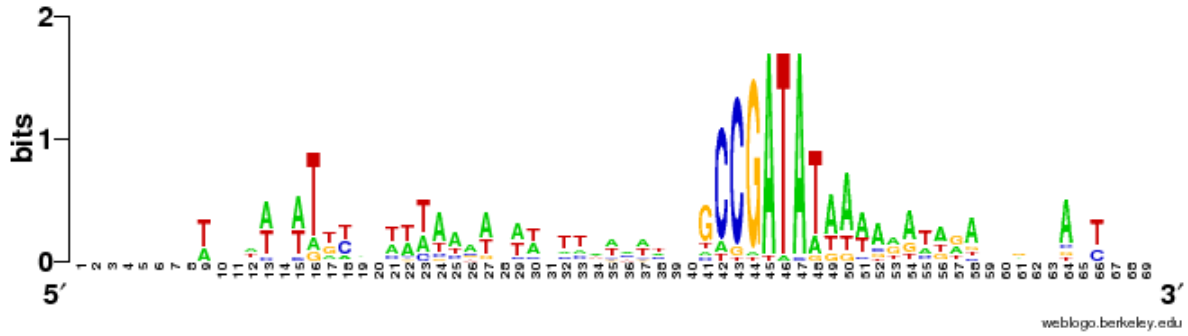

Figura 5.11: Sequence Logo sigD

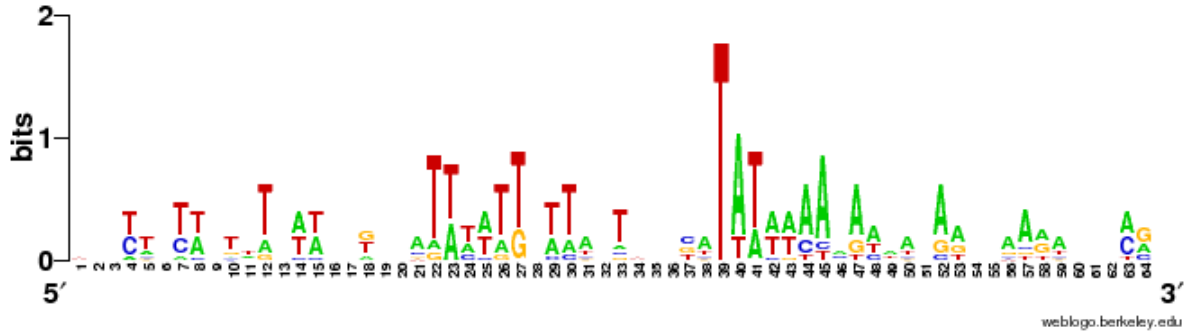

Figura 5.12: Sequence Logo sigD Grupo con mayor número de ocurrencias

Figura 5.13: Sequence Logo sigD Grupo con conservación del fragmento TAAA

5.3.7. Spo0A

Otro elemento con el que se probó fue Spo0A, este elemento pose 24 secuencias conocidas. Las longitudes de estas son muy variables, ya que van desde 16pb la más pequeña, hasta 59pb la más larga.

P	id	nivel	Tamaño	Puntaje	IC	entropía	NE	NF	PE	PC
1	0	0	80	21.41	0.32	1.16	13	5	0.38	0.21
2	2	0	80	20.77	0.34	1.13	12	4	0.33	0.17
3	2	0	80	19.25	0.36	1.1	10	3	0.3	0.13
4	0	0	80	19.23	0.34	1.11	10	5	0.5	0.21
5	5	0	80	18.38	0.51	1.03	9	3	0.33	0.13
6	5	0	80	17.37	0.46	0.98	8	2	0.25	0.08
7	2	0	80	17.35	0.59	0.98	8	3	0.38	0.13
8	2	0	80	17.34	0.58	0.98	8	3	0.38	0.13
9	1	1	64	14.66	0.66	0.93	6	2	0.33	0.08
10	6	1	64	9.03	0.97	0.56	3	2	0.67	0.08
11	0	1	64	8.9	1.24	0.53	3	0	0	0
12	0	1	64	5.73	1.44	0.39	2	0	0	0
13	0	1	64	5.71	1.41	0.41	2	2	1	0.08
14	0	0	40	20.25	0.41	1.1	14	2	0.14	0.08
15	1	0	40	20.19	0.31	1.17	14	2	0.14	0.08
16	3	0	40	20.15	0.33	1.15	14	0	0	0
17	0	0	40	20.11	0.36	1.14	14	1	0.07	0.04
18	1	0	40	20.11	0.34	1.15	14	1	0.07	0.04

Tabla 5.5: Mejores Grupos para Spo0A

Figura 5.14: Sequence Logo sigD Grupo con mayor conservación del fragmento CCGA-TA

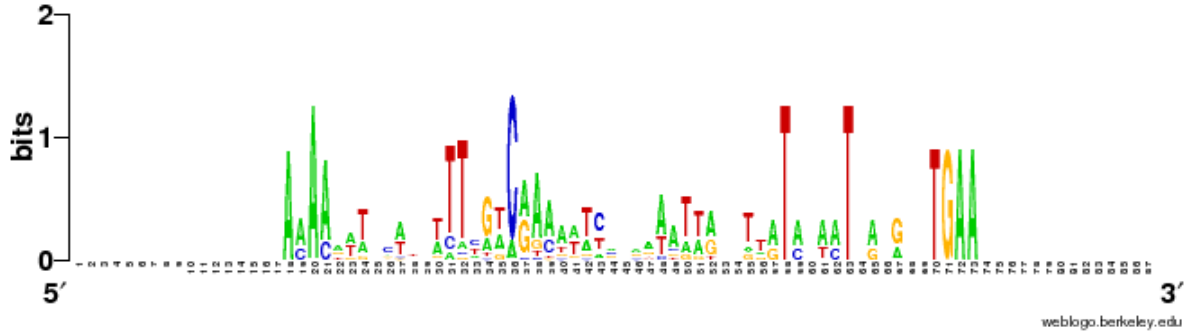

Figura 5.15: Sequence Logo spo0A

Se realizaron experimentos con longitudes de 20pb, 40pb, y 80pb. Cuando se inició con longitudes de 20pb, se formaron en promedio 2 grupos iniciales. El número de elementos estuvo entre 10 y 13. Y con 40pb y 80pb se formaron 3 grupos en promedio. Con un número de elementos de entre 4 y 12, para 80pb iniciales, y entre 10 y 12 elementos para los grupos de 40pb iniciales.

5.3.8. Genoma Completo

El método se probó con el genoma completo del organismo Bacillus Subtilis. Se utilizaron longitudes iniciales de 80pb, 70pb, 50pb, y 25pb. Los grupos con los puntajes más altos para las diferentes longitudes obtenidas se encuentran en el apéndice A. En los experimentos se formaban inicialmente entre 600 y 1000 grupos. Debido al gran número

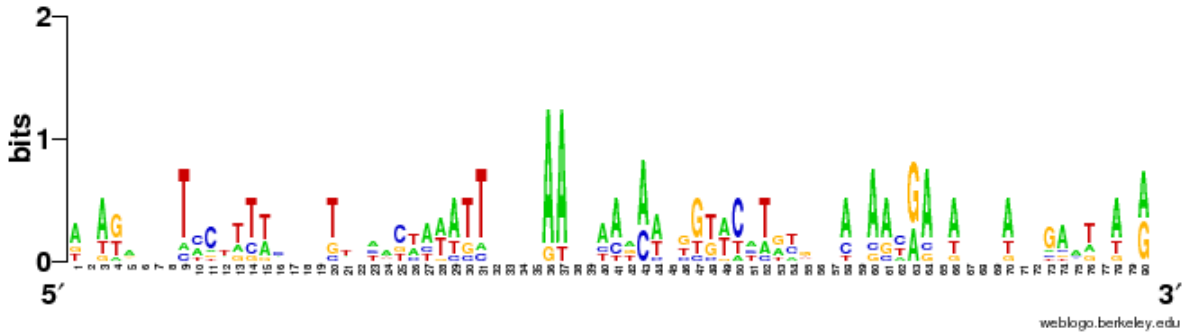

Figura 5.16: Sequence Logo spo0A Grupo con mayor puntaje para 80pb

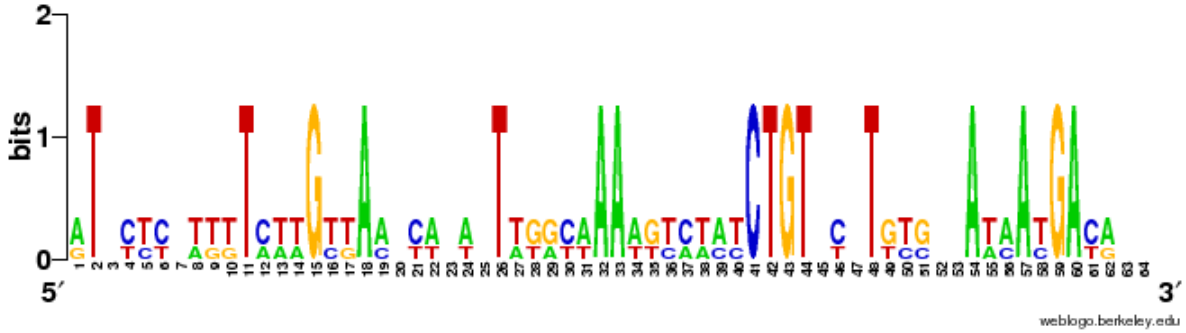

Figura 5.17: Sequence Logo spo0AGrupo con mayor puntaje para 60pb

de secuencias el método consigue obtener pocos grupos con secuencias reguladoras, y no son necesariamente los grupos con los puntajes más altos. En los niveles inferiores se encontraron grupos que contenían secuencias pertenecientes a secuencias reguladoras que ocurren muy comúnmente en el genoma, como son las cajas CAAT, CCAAT, y TATA.

5.4. Discusión

Se ha probado el algoritmo propuesto en diferentes secuencias reguladoras conocidas. Durante estos experimentos se ha visto que el algoritmo es sensible al tamaño inicial de las secuencias, mientras mayor sea el tamaño de los k-mers menor es el número de secuencias generadas, por lo que la cantidad de comparaciones se reduce, pero, si este número es demasiado grande, es posible que no se consiga que las secuencias se alineen en los sitios

deseados. Cuando los k-mers iniciales son muy pequeños aumenta la probabilidad de que queden en diferentes grupos. A pesar de estos, como se ilustró con el conjunto de CRP, el algoritmo es capaz de hallar las secciones de las secuencias donde el parecido entre ellas es mayor. Así, a diferencia de otros métodos que necesitan tener secuencias de longitudes muy cercanas a las reales, el método es capaz de dar una aproximación al tamaño real. Esto es de gran ayuda para el caso de organismos donde hay poco conocimiento, ya que puede aportar información a los biólogos sobre el posible tamaño de las secuencias. En la tabla 5.1 se presentan el porcentaje de aciertos de algunos métodos, incluyendo el que aquí se propone. El valor de precisión de el método propuesto se obtiene del promedio del porcentaje de elementos reguladores conocidos obtenidos por los grupos con mayor puntaje de todos los experimentos. Cabe señalar que para obtener la precisión de los demás métodos se considera que se ha hallado una secuencia si la secuencia obtenida traslapa en 4 posiciones a la secuencias esperada, mientras que para el método propuesto se verificó que la secuencia estuviera completamente contenida. Esto está relacionado con la longitud que deben tener las secuencias reguladoras. Si en los demás métodos se da una tamaño menor al real, no encontraran la secuencia completa. Pero, esta diferencia afecta la precisión calculada.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

Se diseñó y se probó un método para identificar secuencias reguladoras. El método propuesto alcanzó porcentajes similares a los método existentes, con la ventaja de que no es necesario saber con exactitud la longitud de las secuencias buscadas. También se propuso una medida de similitud que capturara algunos aspectos biológicos de las secuencias reguladoras. El agrupamiento permite reducir el espacio de búsqueda, lo cual es muy importante cuando se trabaja con bases de datos grandes. Se aprovecharon las ventajas del agrupamiento jerárquico para que, conforme se va descendiendo por los niveles, se vaya disminuyendo la longitud de las secuencias, con esto, se pueden explorar diferentes longitudes de secuencias, para así, elegir el tamaño más conveniente. Al no utilizarse conocimiento del dominio, el método no está ligado al tipo de organismo, lo que le da flexibilidad al método para poder utilizarse con cualquier organismo. El tiempo de ejecución de este métodos es relativamente corto, para las conjuntos con los que se experimentó le tomaba entre 1 y 3 minutos realizar el agrupamiento, esto dependiendo del tamaño inicial de las secuencias, y el número de genes en los que se busca, mientras mayor sea el tamaño de las secuencias, menor el tiempo, y mientras menor sea el número de genes también será menor el tiempo.

6.2. Trabajo Futuro

Uno de los factores principales para que el método propuesto tenga un buen desempeño es la función de evaluación de los grupos. Por lo tanto, para una identificación más exacta se puede trabajar en el diseño de una medida de evaluación de grupos que tome en cuenta características que no se ven reflejadas en la función utilizada aquí, y que aporten más información sobre la relación entre los elementos de los grupos.

La elección de centroides para el método propuesto es aleatoria. Sin embargo no necesariamente es la más adecuada. Si se tiene cierto conocimiento del organismo a analizar, podría ser utilizado para elegir los centroides. Aunque esto podría limitar un poco el método ya que se volvería dependiente de la información del dominio.

También podrían implementarse técnicas de paralelización para disminuir el tiempo de ejecución. Así podrían irse subdiviendo varios grupos al mismo tiempo.

El método le proporciona al usuario una buena forma de analizar las secuencias, puede ir observando los patrones de los grupos que se van formando, y puede ayudarle a tomar una decisión sobre el tamaño de las secuencias a buscar.

Bibliografía

[1] Crooks G. E., Hon G., Chandonia J.-M. M., y Brenner S. E. (2004). Weblogo: a sequence logo generator. *Genome research*, 14(6):1188–1190.

[2] Das M. y Dai H. K. (2007). A survey of dna motif finding algorithms. *BMC Bioinformatics*, 8(Suppl 7):S21.

[3] Hon L. S. y Jain A.Ñ. (2006). A deterministic motif finding algorithm with application to the human genome. *Bioinformatics*, 22(9):1047–1054.

[4] Ishii T., Yoshida K.-i., Terai G., Fujita Y., y Nakai K. (2001). Dbtbs: a database of bacillus subtilis promoters and transcription factors. *Nucl. Acids Res.*, 29(1):278–280.

[5] Jain A. K., Murty M.Ñ., y Flynn P. J. (1999). Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323.

[6] Jensen S. T., Shen L., y Liu J. S. (2005). Combining phylogenetic motif discovery and motif clustering to predict co-regulated genes. *Bioinformatics*, 21(20):3832–3839.

[7] Karabulut M. y Ibrikci T. (2008). Fuzzy c-means based dna motif discovery. En *ICIC '08: Proceedings of the 4th international conference on Intelligent Computing*, pp. 189–195, Berlin, Heidelberg. Springer-Verlag.

[8] Kelarev A., Kang B., y Steane D. (2006). Clustering algorithms for its sequence data with alignment metrics. pp. 1027–1031.

[9] Kon M., Fan Y., Holloway D., y DeLisi C. (2007). Svmotif: A machine learning motif algorithm. *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*, pp. 573–580.

[10] Larkin M. A., Blackshields G., Brown N. P., Chenna R., McGettigan P. A., McWilliam H., Valentin F., Wallace I. M., Wilm A., Lopez R., Thompson J. D., Gibson T. J., y Higgins D. G. (2007). Clustal w and clustal x version 2.0. *Bioinformatics*, 23(21):2947–2948.

[11] Lones M. A. y Tyrrell A. M. (2007). Regulatory motif discovery using a population clustering evolutionary algorithm. *IEEE/ACM Trans Comput Biol Bioinform*, 4(3):403–414.

[12] MacIsaac K. D. y Fraenkel E. (2006). Practical strategies for discovering regulatory dna sequence motifs. *PLoS Comput Biol*, 2(4-e36):201–210.

[13] Middendorf M., Kundaje A., Shah M., Freund Y., Wiggins C. H., y Leslie C. (2005). Motif discovery through predictive modeling of gene regulation. *Research in Computational Molecular Biology*, pp. 538–552.

[14] Pavesi G., Mereghetti P., Mauri G., y Pesole G. (2004). Weeder web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucl. Acids Res.*, 32(suppl_2):W199–203.

[15] Pavesi G., Zambelli F., y Pesole G. (2007). Weederh: an algorithm for finding conserved regulatory motifs and regions in homologous sequences. *BMC Bioinformatics*, 8(1):233–241.

[16] Singh C. P. P., Khan F., Mishra B. N.Ñ., y Chauhan D. S. S. (2008). Performance evaluation of dna motif discovery programs. *Bioinformation*, 3(5):205–212.

[17] Stavrovskaya, E., Makeev, V., Mironov, y A. (2006). Clustertree-rs: A binary tree algorithm identifying coregulated genes by clustering regulatory signals. *Molecular Biology*, 40(3):465–473.

[18] van Helden J., André B., y Collado-Vides J. (1998). Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281(5):827–842.

[19] van Nimwegen E., Zavolan M., Rajewsky N., y Siggia E. D. (2002). Probabilistic clustering of sequences: inferring new bacterial regulons by comparative genomics. *Proceedings of the National Academy of Sciences of the United States of America*, 99(11):7323–7328.

[20] Wang D. y Lee N. K. (2008). Computational discovery of motifs using hierarchical clustering techniques. *Data Mining, IEEE International Conference on*, 0:1073–1078.

[21] Wei Z. y Jensen S. T. (2006). Game: detecting cis-regulatory elements using a genetic algorithm. *Bioinformatics*, 22(13):1577–1584.

Apéndice A. Puntajes más altos de los experimentos con todo el genoma

Tabla 1: Grupos con mayores puntajes

id	Nivel	Longitud	Puntaje	IC	entropía	Elementos
0	0	80	76.5	0.94	0.74	10240
0	0	80	75.56	0.32	1.17	8704
0	0	80	75.41	0.33	1.16	8704
0	0	80	75.41	0.33	1.16	8704
0	0	80	75.26	0.67	0.93	9216
0	0	80	75.11	0.34	1.15	8192
0	0	80	75.02	0.35	1.15	8192
0	0	80	74.99	0.35	1.15	8192
0	0	80	74.93	0.3	1.18	8192
0	0	80	74.92	0.35	1.14	8192
0	1	64	41.65	0.22	1.23	146
0	1	64	39.63	0.22	1.23	121
0	1	64	39.24	0.23	1.21	112
13	1	64	38.67	0.25	1.2	109
26	1	64	38.53	0.23	1.22	107
7	1	64	38.36	0.21	1.23	87
13	1	64	38.14	0.25	1.2	106
17	1	64	38.03	0.22	1.23	97
0	1	64	38	0.21	1.23	92
0	1	64	37.83	0.24	1.21	94

id	Nivel	Longitud	Puntaje	IC	entropía	Elementos
0	2	51	35.7	0.47	1.06	61
0	2	51	35.04	0.26	1.2	72
1	2	51	34.63	0.28	1.19	71
0	2	51	34.5	0.26	1.2	74
0	2	51	34.46	0.31	1.16	78
0	2	51	34.23	0.3	1.16	70
0	2	51	33.85	0.56	0.92	48
0	2	51	33.81	0.29	1.17	72
0	2	51	33.5	0.26	1.19	63
0	2	51	33.47	0.29	1.17	64
0	0	50	72.9	0.28	1.19	9728
0	0	50	72.59	0.27	1.2	9216
0	0	50	72.41	0.27	1.2	8704
0	0	50	71.22	0.34	1.15	7680
8	0	50	27.65	0.21	1.24	32
3	0	50	27.62	0.2	1.25	32
8	0	50	27.55	0.23	1.23	32
1	0	50	27.47	0.22	1.23	32
4	0	50	27.42	0.25	1.22	32
0	0	50	27.29	0.22	1.22	31
0	0	48	68.89	0.41	1.1	6656
0	0	48	68.54	0.41	1.1	6144
0	0	48	67.29	0.44	1.08	5632
0	0	48	66.51	0.47	1.06	5120
0	0	48	65.35	0.6	0.97	4096
0	0	48	65.32	0.59	0.97	4608
0	0	48	64.76	0.6	0.97	3584
0	0	48	64.75	0.57	0.99	4096
0	0	48	64.64	0.53	1.02	4096
0	0	48	64.37	0.57	0.99	4096
0	0	40	70.51	0.31	1.17	9728
0	0	40	70.4	0.31	1.17	9728

id	Nivel	Longitud	Puntaje	IC	entropía	Elementos
0	0	40	70.36	0.29	1.19	9728
0	0	40	69.67	0.3	1.18	9216
0	0	40	69.02	0.34	1.15	8192
0	0	40	64.09	0.47	1.06	4608
0	0	40	63.13	0.54	1.02	4096
0	0	40	62.32	0.66	0.93	3584
0	0	40	61.81	0.66	0.93	3072
0	2	38	19.38	0.41	1.07	13
2	2	38	19.19	0.42	1.1	13
0	2	38	18.67	0.38	1.09	12
0	2	38	18.2	0.43	1.08	11
0	2	38	18.09	0.39	1.07	11
0	2	38	18.07	0.4	1.05	11
0	2	38	18	0.37	1.04	11
0	2	38	17.92	0.44	1.06	11
1	2	38	17.51	0.52	1.03	10
2	2	38	17.44	0.47	1.02	10
0	0	36	63.6	0.54	1.01	4608
0	0	36	63.57	0.51	1.03	4608
0	0	36	63.21	0.55	1.01	4608
0	0	36	62.35	0.55	1	4096
0	0	36	61.47	0.66	0.93	3584
0	0	36	61.45	0.61	0.96	3584
0	0	36	61.02	0.71	0.9	3072
0	0	36	60.97	0.65	0.93	3584
0	0	36	60.93	0.63	0.95	3584
0	0	36	60.35	0.71	0.9	3072
0	4	32	30.85	0.69	0.86	44
0	4	32	29.78	0.76	0.86	40
5	4	32	28.99	0.31	1.16	46
2	4	32	27.64	0.41	1.1	43
0	4	32	26.48	0.33	1.15	38

id	Nivel	Longitud	Puntaje	IC	entropía	Elementos
2	4	32	26.39	0.34	1.13	33
6	4	32	26.24	0.43	1.09	29
0	4	32	25.91	0.4	1.09	36
0	4	32	25.69	0.39	1.09	32
0	4	32	25.44	0.32	1.14	31
0	0	30	62.3	0.46	1.07	5632
0	0	30	61.65	0.47	1.06	5120
0	0	30	59.69	0.57	0.99	4096
0	0	30	59.63	0.58	0.98	4096
0	0	30	59.16	0.55	1	3584
0	0	30	59.04	0.64	0.94	3584
0	0	30	58.74	0.6	0.97	3584
0	0	30	56.47	0.73	0.88	2560
0	0	30	55.59	0.88	0.78	2560
0	3	30	18.22	0.44	1.08	13
0	0	28	58.34	0.54	1.01	4608
0	0	28	58.04	0.6	0.97	4096
0	0	28	57.7	0.65	0.94	4096
0	0	28	57.63	0.71	0.89	3584
0	0	28	56.09	0.81	0.82	3584
0	0	28	56.02	0.74	0.87	3072
0	0	28	55.89	0.77	0.85	3584
0	0	28	55.76	0.82	0.82	3584
0	0	28	55.66	0.95	0.72	2560
0	0	28	55.64	0.77	0.86	3072
1	5	25	28.33	0.88	0.72	41
2	5	25	28.3	0.88	0.72	40
0	5	25	27.01	0.66	0.87	37
1	5	25	25.76	0.43	1.09	42
0	5	25	24.57	0.4	1.08	36
2	5	25	24.43	0.77	0.72	25
0	5	25	23.99	0.95	0.61	23

id	Nivel	Longitud	Puntaje	IC	entropía	Elementos
1	5	25	23.49	0.73	0.88	24
1	5	25	23.26	0.47	1.05	24
0	0	21	52.02	0.77	0.85	3072
0	0	21	51.83	0.67	0.92	3072
0	0	21	51.2	0.77	0.85	2560
0	0	21	51.16	0.67	0.92	3072
0	0	21	50.99	0.76	0.86	3072
0	0	21	50.84	0.76	0.86	3072
0	0	21	50.31	0.91	0.76	2560
0	0	21	49.09	1.07	0.64	2048
0	0	21	48.06	1.13	0.6	1536
0	0	21	48	1.04	0.66	2048
0	0	20	50.74	0.64	0.94	3072
0	0	20	49.88	0.83	0.81	2560
0	0	20	49.65	0.7	0.9	3072
0	0	20	49.64	0.9	0.76	2560
0	0	20	49.56	0.64	0.94	3072
0	0	20	49.51	0.82	0.82	2560
0	0	20	49.33	0.77	0.85	2560
0	0	20	49.29	0.63	0.95	2560
0	0	20	49.18	0.78	0.85	2560
0	0	20	49.15	0.85	0.79	2560
0	5	19	12.57	0.56	1	8
1	5	19	12.53	0.55	0.95	8
0	5	19	12.21	0.64	0.81	8
1	5	19	12.01	0.72	0.87	7
0	5	19	11.96	0.6	0.97	7
0	5	19	11.94	0.66	0.93	7
0	5	19	11.88	0.75	0.81	7
0	5	19	11.85	0.7	0.8	7
1	5	19	11.79	0.74	0.86	7
1	5	19	11.78	0.73	0.88	7

id	Nivel	Longitud	Puntaje	IC	entropía	Elementos
0	0	16	46	0.71	0.89	4096
0	0	16	45.66	0.69	0.91	3584
0	0	16	45.6	0.69	0.91	4096
0	0	16	44.4	0.73	0.88	3584
0	0	16	43.77	0.81	0.83	3072
0	0	16	43.22	0.84	0.81	3072
0	0	16	43.09	1.12	0.61	2560
0	0	16	43.05	1.3	0.49	2048
0	0	16	42.94	1.13	0.61	2048
0	0	16	42.72	0.92	0.75	2560
1	6	15	11.01	0.74	0.87	8
0	6	15	10.55	0.79	0.84	7
0	6	15	10.49	0.63	0.95	7
1	6	15	10.42	0.82	0.82	7
1	6	15	10.34	0.82	0.82	7
0	6	15	10.33	0.62	0.88	7
0	6	15	10.33	0.66	0.93	7
0	6	15	10.3	0.66	0.93	7
0	6	15	10.11	0.67	0.87	7
0	6	15	9.71	0.61	0.96	6
0	0	14	44.26	0.47	1.06	5632
0	0	14	44.22	0.54	1.01	5632
0	0	14	43.26	0.53	1.02	4608
0	0	14	43.18	0.62	0.96	4608
0	0	14	43.14	0.48	1.05	5120
0	0	14	42.9	0.55	1	4096
0	0	14	42.23	0.73	0.88	4096
0	0	14	41.16	0.79	0.84	3072
0	0	14	40.55	0.76	0.86	3072
0	0	14	40.52	0.87	0.78	3072
0	0	12	34.78	0.78	0.85	2560
0	0	12	34.1	1.01	0.69	2048

id	Nivel	Longitud	Puntaje	IC	entropía	Elementos
0	0	12	33.98	0.73	0.88	2048
0	0	12	33.89	1.29	0.49	2048
0	0	12	33.73	1.46	0.37	1536
0	0	12	33.68	1.25	0.52	2048
0	0	12	33.42	0.93	0.74	2048
0	0	12	33.15	1.36	0.44	2048
0	0	12	32.68	1.49	0.36	1536
0	0	12	32.23	1.3	0.49	1536
0	0	10	29.2	0.76	0.86	3072
0	0	10	28.86	0.83	0.81	3584
0	0	10	28.83	0.57	0.99	3584
0	0	10	28.1	0.92	0.75	2560
0	0	10	27.79	0.85	0.8	2560
0	0	10	27.75	1.06	0.65	2048
0	0	10	27.71	0.91	0.76	2560
0	0	10	27.6	0.98	0.71	2048
0	0	10	27.48	1.08	0.64	2048
0	0	10	27.37	1.09	0.63	2048
0	0	8	18.71	0.67	0.92	5632
0	0	8	17.84	0.99	0.7	4096
0	0	8	17.84	0.58	0.99	4096
0	0	8	17.72	0.8	0.83	3072
0	0	8	17.66	0.52	1.03	4608
0	0	8	17.57	0.63	0.95	4608
0	0	8	17.38	1.16	0.58	3072
0	0	8	17.32	0.88	0.77	2560
0	0	8	16.99	0.82	0.82	3072
0	0	8	16.97	1.1	0.62	3072
0	0	8	16.83	1.07	0.64	3584
0	0	7	9.2	1.48	0.36	1536
0	0	7	9.14	1.03	0.67	2048
0	0	7	9.03	1.51	0.34	1536

id	Nivel	Longitud	Puntaje	IC	entropía	Elementos
0	0	7	9.01	1.24	0.53	2048
0	0	7	8.94	1.25	0.52	1536
0	0	7	8.91	1.48	0.36	1536
0	0	7	8.83	1.21	0.55	1536
0	0	7	8.83	1.57	0.3	1024
0	0	7	8.83	1.57	0.3	1024
0	0	7	8.74	1.61	0.27	1536
id	Nivel	Longitud	Puntaje	IC	entropía	Elementos