Algoritmo Genético Multi-objetivo para el descubrimiento de secuencias reguladoras.

por:

José Luis Hernández Domínguez

Tesis sometida como requisito parcial para obtener el grado de **Maestro en Ciencias en el Área de Ciencias Computacionales** en el Instituto Nacional de Astrofísica, Óptica y Electrónica

Supervisado por:

Dr. Aurelio López López

Dr. Jesús Antonio González Bernal

AGRADECIMIENTOS

CONACyT por el apoyo económico que proporciono durante los dos años de la maestría.

Dra. María Patricia Sánchez Alonso y Dr. Candelario Vázquez Cruz por el apoyo y guía en la parte biológica durante esta investigación. Ambos doctores en el departamento de microbiología de la BUAP.

A mis asesores, el Dr. Jesús A. González Bernal y Dr. Aurelio López López, por su guía y asesoría.

A los doctores y sinodales Miguel Arias Estrada, Jesús Ariel Carrasco Ochoa y Gustavo Rodríguez Gómez por las observaciones finales de la tesis.

Contenido

RESUMEN

En algunas áreas como la biología y la medicina, es de suma importancia entender el comportamiento y funcionamiento de los seres vivos a un nivel genético. Esto se debe, principalmente, a que los seres vivos son regulados por segmentos de código llamados genes. Para que un gen se active debe recibir una directriz de segmentos del ADN y ARN conocidos como secuencias reguladoras. Para encontrar una secuencia reguladora es fundamental buscar motivos dentro del código genético. Un motivo es una secuencia de nucleótidos que se repite en determinadas regiones del ADN, comúnmente cerca de los genes. Una de las técnicas que se han utilizado para realizar la búsqueda de secuencias reguladoras son los Algoritmos Genéticos. Estas búsquedas estocásticas, basadas en la evolución de las especies, se basan en el perfeccionamiento de las características de los individuos mediante el entrecruzamiento (reproducción) de los mismos, con una continua evaluación. Dada la complejidad de los motivos y las diferentes características que tienen, un acercamiento con un solo objetivo es insuficiente dado que cada motivo puede tener diferentes cualidades y no todas las cualidades se repiten en todos los motivos. Debido a esto, se plantea el uso de varios objetivos (optimización multi-objetivo). Esto se hace para encontrar la mayor cantidad de motivos y de esta forma obtener las secuencias reguladoras. Los resultados obtenidos muestran que el método es competente para la tarea, ya que ha encontrado regiones altamente conservadas y, también, se han encontrado secuencias dentro de las cuales se han observado sub-secuencias obtenidas por otros métodos. Otra ventaja del método propuesto es que fue capaz de encontrar secuencias de gran tamaño con una longitud de hasta 18bp (*"base pair"* o pares de bases). Esto es importante debido que, entre más larga es la cadena, menos probable es que sea un patrón aleatorio sin significado biológico.

ABSTRACT

In some areas such as biology and medicine, it is important to understand de behavior and functionality of living organism in a genetic level. The main reason is that every organism is controlled by short strings of genetic code called genes. To active gen it must receive a directive of the DNA and RNA segments known as regulatory sequence. To find a regulatory sequence is fundamental to search motives inside of the genetic code. A motive is a short sequence made of nucleotides that repeat in certain region of the DNA, commonly near a gen. One technique that has been used to search regulatory sequence is Genetic Algorithm. These stochastic searches, based on evolution of the species, are based on the improvement of individual characteristic through breeding of itself. Since the motive and different characteristic complexity an approach of a mono-objective is not enough since motives have many qualities and not every quality is in all motives. Because of this, the use of a multi-objective algorithm is proposed. This is in order to find most of the motives and, in this way, be able regulatory sequences. The obtained results show that this method is well suited to do the task. This is because it found high conserved regions and, also, there are sequences inside those founded by other methods. Another advantage is that our algorithm was capable on finding big sequence with a length of 18bp (base pair). This is important because is less probable to find a random patron without biological meaning with big sequences.

Capítulo I
INTRODUCCIÓN

El descubrimiento de secuencias reguladoras es de gran importancia para las ciencias médicas, así como la farmacéutica, biológica y agrónoma, entre otras. Esto se debe a que la mayoría de las funciones que regulan los aspectos vitales de los seres vivos son controladas por las secuencias reguladoras. Por ejemplo, en agronomía se utilizan los genes para mejorar los cultivos. Al modificar sus genes se pueden alterar las propiedades del organismo, haciéndolo más resistente a las plagas o humedad, que generen más alimento o que tengan una vida de almacén más larga. Tomemos el caso de la gamma-tocoferol metiltransferasa (GTM). La GTM es una sustancia relacionada directamente con la producción de vitamina E. La vitamina E sólo la producen las plantas y, si se encuentra la secuencia reguladora que activa la producción de GMT, se puede crear una planta con un alto contenido de vitamina E.

Para encontrar una secuencia reguladora es necesario entender cómo las secuencias repetidas o motivos entran en juego. Como dice D'haeseleer [4], uno puede tomar una base de datos de un genoma de algún organismo suficientemente bien estudiado y empezar a buscar los motivos más frecuentes, como lo hace un explorador que busca oro. Sin embargo, nada más lejos de la realidad es lo descrito; la búsqueda de motivos es pesada, complicada y escabrosa. No todos los motivos encontrados y repetidos son relevantes o contienen alguna información. Tampoco un motivo desperdigado por ahí, puede ser ignorado por no encontrarse en otros sitios. D'haeseleer [4] muestra un panorama de los diferentes tipos de acercamientos que existen en el descubrimiento de motivos.

Los diferentes métodos para la búsqueda de motivos se pueden dividir en tres grandes grupos, los enumerativos, los de optimización determinística y los de optimización probabilística. A continuación se describe brevemente cada método.

- Enumerativos. Los enumerativos buscan exhaustivamente todos los posibles motivos en el espacio de búsqueda, esto para un modelo descriptivo específico.

Un ejemplo de estos métodos son los basados en diccionarios, los cuales buscan todas las ocurrencias de n secuencias y calculan cuál es la que tiene una mayor presencia.

- Optimización determinística. Aquí entran los basados en maximización de expectación (*Expectation Maximization*) y que pueden optimizar simultáneamente matrices de posiciones de peso para describir determinados motivos. Uno de los más famosos es *MEME* [1] que busca en los motivos repetidos dentro de una base de datos de genes el mejor resultado y luego itera hasta converger, evitando los máximos locales.
- Optimización probabilística. Utiliza una búsqueda estocástica de entrada, luego se itera varias veces y cada secuencia se evalúa con el modelo inicial. En cada iteración, el algoritmo decide de forma probabilística si agrega o elimina un sitio. Al terminar la iteración se ajustan las probabilidades y se calcula de nuevo el modelo. Un ejemplo representativo de este tipo de algoritmos sería *Gibbs Sampler* [25].

Ningún método ha logrado solucionar satisfactoriamente este problema. Esto se debe a que la búsqueda de motivos, en especial los encargados de regular los genes, tiene diferentes características que cambian de especie en especie e, inclusive, del entre diferente organismos muestreados de la misma especie.

El objetivo de este trabajo es proponer un método para encontrar posibles secuencias reguladoras considerando un enfoque donde se evalúe un motivo visto desde diferentes puntos de observación. Para realizar dicha búsqueda se planea utilizar los Algoritmos Genéticos. Se decidió utilizar a los Algoritmos Genéticos por el potencial que tienen los Algoritmos Evolutivos (siendo los Genéticos uno de ellos) para la resolución de problemas multi-objetivos. La razón por los que los Algoritmos Genéticos son buenos resolviendo problemas multi-objetivos, como explica Coello et al. [3], es por la capacidad de manejar poblaciones, cuyos individuos son soluciones del problema, y obtener varios resultados en una misma ejecución. Al obtener varios resultados en una misma ejecución se puede tener varios miembros del frente de Pareto. Los Algoritmos Genéticos pertenecen al grupo de búsqueda estocástica

basados en la evolución de las especies, dado que crean una población y la van haciendo evolucionar cruzando los individuos y mutándolos hasta obtener la solución deseada o hasta determinado número de iteraciones.

Para lograr que los individuos evolucionen y se dirijan hacia la solución deseada es necesario evaluarlos y mantener los individuos más adaptados de la población. En la evolución darwiniana a ese proceso se le llama selección natural. En la naturaleza, los individuos más aptos sobreviven y se reproducen mientras que los otros individuos simplemente se extinguen. Dado que los Algoritmos Genéticos se desarrollan en un entorno virtual, dentro de una computadora, no existen las reglas naturales que ponen a prueba a los individuos. Por eso se crea la función de aptitud, que es la encargada de evaluar a los individuos y categorizarlos con la misma finalidad que la naturaleza evalúa a sus creaturas, para sobrevivir y reproducirse o morir.

Existen dos enfoques de evaluación de individuos, el enfoque mono-objetivo y el enfoque multi-objetivo. Ambos enfoques son validos para determinados problemas. El enfoque mono-objetivo es usado cuando se tiene un problema en el cual se tiene que maximizar o minimizar una variable, mientras que en el enfoque multi-objetivo se busca maximizar o minimizar más de una variable. Estas variables pueden tener el mismo valor o, inclusive, ser contradictorias entre ellas. La importancia de utilizar un enfoque multi-objetivo es porque puede existir más de una posible solución válida o que resuelve de forma satisfactoria dicho problema y, los enfoques multi-objetivos, tienden a evaluar y presentar este tipo de soluciones. En esta tesis se utilizó la evaluación multi-objetivo porque, en la búsqueda de motivos para encontrar secuencias reguladoras, existen varios factores que son importantes a la hora de evaluar una cadena o un conjunto de cadenas.

Esta tesis está organizada de la siguiente forma, en el capítulo dos se da a conocer las bases teóricas en las que se basa esta tesis. En el capítulo tres se muestra el trabajo relacionado al tema que se aborda en esta tesis. En el capítulo cuatro se describe a detalle los procedimientos y los pasos que se siguieron para llegar a la

solución. En el capítulo cinco se abordan los resultados obtenidos y, finalmente, las conclusiones y discusiones están en el capítulo seis.

Capítulo II
MARCO TEÓRICO

Esta sección presenta información de utilidad para poder entender el contenido posterior de la tesis. Se espera que el lector tenga un conocimiento previo sobre computación y un conocimiento básico de biología, en especial, genética.

2.1 Algoritmos genéticos.

Los Algoritmos Genéticos son algoritmos de búsqueda estocástica, basados en la teoría la evolución Darwinista de la adaptación y supervivencia de los individuos más aptos. Goldberg[9] describe a los Algoritmos Genéticos como algoritmos tomados de la naturaleza y sus constantes avances, y dándoles un toque de la genialidad humana.

Los Algoritmos Genéticos constan de los elementos básicos de todo individuo y su camino a la trascendencia, como son el nacimiento, la reproducción (cruza), mutación y selección de individuos, lo que lleva a los individuos más aptos a la siguiente generación y extingue a los menos aptos. Estos elementos se describen a continuación.

- Individuo: El individuo es el núcleo de los Algoritmos Genéticos. Cada individuo es una posible solución, que va a ir evolucionando conforme avancen las generaciones. Para lograr esto, el problema es codificado para su manejo posterior. Normalmente es codificado como una cadena de números (generalmente unos y ceros) o letras. Por ejemplo, tenemos una maquina simple que opera abriendo o cerrando válvulas y se debe encontrar la combinación de válvulas que mejor optimice el gasto de energía contra la velocidad de trabajo. La maquina cuenta con tres grupos de válvulas, cada grupo cuenta con tres válvulas que deben cerrarse y abrirse para obtener una configuración. Los grupos de válvulas son flujo de combustible, velocidad de la banda transportadora y tiempo de secado. La cadena que codificaría el problema sería {[0][0][0][0][0][0][0][0][0]}. La codificación está conformado por dos partes, genes y alelos.

- Genes: los genes son la unidad mínima con significado dentro de la codificación de un individuo. Tomando como ejemplo la máquina de válvulas mencionado anteriormente el primer grupo de tres ceros sería el gen que controla el flujo de combustible, el segundo grupo de tres sería la velocidad de la banda transportadora y, por último, los tres restantes serían el gen que controla el tiempo de secado.
- Alelos: Es la unidad mínima de la codificación del problema. Usando el ejemplo de la maquina anterior, un cero elegido al azar sería un alelo.

- Generaciones: Son el número de ciclos vitales (de vida y muerte de los individuos) que conforman la población de individuos. Las generaciones son utilizadas como un criterio de paro, de tal forma que el algoritmo se detenga después de cierto número de iteraciones (generaciones).
- Evaluación de los individuos: Para saber si un individuo es apto o no, se evalúa según las características u objetivos buscados. La evaluación de los individuos es de suma importancia, dado que una mala evaluación, puede llevar a un resultado erróneo o poco deseado. La diferencia entre los Algoritmo Genéticos mono-objetivos y los multi-objetivos es la evaluación de los individuos o, como se llama en el argot del oficio, función de aptitud. Todas las operaciones genéticas son similares.
- Inicio o creación de la población: La población es generada de forma aleatoria y, después, los individuos recién generados son evaluados para su entrada en los procesos vitales.
- Operadores genéticos: Se dividen en cuatro partes aunque, según el acercamiento que se plantee, pueden estar ausentes uno o más elementos.
  - Selección: Busca encontrar a los individuos que han de reproducirse. Existen diferentes métodos, dos de los más conocidos son el de la ruleta y el de torneo.
    - Ruleta: Para determinar la probabilidad de ser escogido como elemento de reproducción y se obtiene la frecuencia relativa de cada individuo. A continuación se escogen individuos al azar como punto de inicio, se va recorriendo los individuos ordenados según su frecuencia y se van sumando sus

frecuencias hasta pasar un límite preestablecido. El individuo que rompió con el límite es el escogido y se repite el proceso para encontrar a su pareja.

- Torneo: Se escogen dos parejas al azar de la población, se selecciona el mejor individuo de cada pareja y se cruzan. Se repite el proceso hasta alcanzar el número de individuos iniciales.

- Cruza: Los individuos se reproducen intercambiando el material genético que tienen para producir una nueva generación. Los métodos más comunes son: cruza en un punto (un elemento del vector es intercambiado); y el intercambio en un punto (todos los elementos del vector se intercambian hasta el punto seleccionado).
- Mutación: Un individuo es seleccionado al azar para que su material genético sea modificado. Generalmente se modifica sólo un
- Elitismo: Significa seleccionar el individuo más apto y pasarlo a la siguiente generación sin modificación alguna, con la finalidad de preservar la mejor solución hasta el momento.

Existen diferentes tipos de Algoritmos Genéticos como el estándar, el Genitor y el CHC, entre otros. Aquí nos enfocamos únicamente en dos de ellos, siendo el primero un punto de referencia y el segundo el usado en nuestra propuesta de solución. Larry [14] muestra las ventajas del CHC frente a los algoritmos genéticos antes mencionados.

- Algoritmo Genético estándar: Es el más sencillo de todos los Algoritmos Genéticos que incluye todos los procesos vitales antes mencionados. Este algoritmo es el más rápido de todos los Algoritmos Genéticos (a excepción del Algoritmo Genético simple, el cual no incluye el elitismo), pero no cuenta con ninguna forma de manejo de la población y, por lo tanto, no tiene manera de evitar tener individuos idénticos ni evitar máximos locales.
- Algoritmo Genético CHC. La abreviatura CHC viene de *"Cross generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation" (*Selección

elitista generacional, recombinación heterogénea y mutación cataclísmica). Este algoritmo se destaca del estándar en su capacidad de manejar los individuos. El algoritmo CHC está dentro de los algoritmos considerados "*steady-state*" (estado de equilibrio) que busca no tener ninguna solución repetida, lo que significa, que no deben existir dos individuos iguales dentro de la población. Este algoritmo implementa todos los procesos vitales además de los siguientes:

- Selección elitista generacional: A diferencia del estándar, el CHC elige los mejores individuos entre los padres y los hijos, hasta llenar de nuevo la población. El estándar remplaza los padres por los hijos, a excepción del individuo élite (éste puede ser un individuo de varias generaciones atrás o un individuo recién creado).
- Recombinación heterogénea: Busca cruzar individuos cuyo material genético difiera entre ellos hasta cierto umbral definido previamente, con la finalidad de evitar el "incesto" (dos individuos con material genético muy similar).
- Mutación cataclísmica: CHC sigue las soluciones de cerca y, si después de determinado número de generaciones, no existe una mejora dentro de la población (debido, posiblemente, a un estancamiento en un máximo local), la población es mutada drásticamente, conservando intacta la mejor solución hasta el momento.

2.2 Descubrimiento de motivos.

La búsqueda de motivos es una parte importante para el descubrimiento de secuencias reguladoras o promotores. Un motivo es una secuencia que se repite con cierta frecuencia en regiones determinadas del ADN. Algunas de las regiones antes mencionadas son las regiones promotoras y, los motivos dentro de estas regiones, son los sitios de transcripción.

Para diferenciar las regiones promotoras de las regiones no promotoras, se han considerado diferentes características, como las islas CpG, las cajas TATA, cajas CAAT,

algunos sitios de transcripción específicos, matrices de pentamer y oligonucleótidos. También se han usado varias tecnologías para el reconocimiento de patrones como redes neuronales, análisis discriminante tanto lineal como cuadrático y análisis de componentes independientes, entre otros, tal como menciona Xundong et al. [27].

Liu et al. [13] señala los sitios de transcripción más conocidos, tales como las cajas TATA, BRE, Inr y DPE. Las cajas TATA, BRE, Inr y DPE, entre otras, son secuencias muy bien conocidas que tienen pocas variantes y son relativamente fáciles de localizar. Sin embargo, menciona que los sitios de transcripción antes mencionados no aparecen en todas las regiones promotoras. Por ejemplo, en el organismo llamado *Drosophila*, se estima que cerca del 29% de las secuencias promotoras tiene únicamente cajas TATA pero no DPE; 26% contiene sólo DPE sin cajas TATA; 14% contiene ambos y 31% ninguno. Los sitios de transcripción tampoco son constantes para todos los organismos ya que, gracias a la evolución, los organismos tienden a especializarse para adaptarse a su entorno, modificando tanto las dimensiones como las características que conforman los sitios de transcripción. Esta variabilidad sugiere un problema al momento de la búsqueda, ya que el tiempo de la búsqueda se expande exponencialmente.

Computacionalmente hablando, la dificultad aumenta por el hecho de que la posición de los sitios de transcripción puede variar significativamente en las regiones río arriba de los diferentes genes homólogos. La expresión río arriba se refiere a desplazarse a la izquierda del inicio del gen (figura 7 en la sección 4.1). Dongsheng et al. [6] comenta que la búsqueda de todas las posibles combinaciones de la posición de inicio es impráctica y requiere un tiempo computacional exponencial. Para evadir la búsqueda exhaustiva, se han desarrollado varias herramientas computacionales, tales como *AlignACE, BioProspector* y *Gibbs Sampler* [22, 13 y 25].

2.3 Secuencia reguladora.

Se ha estado mencionando “secuencia reguladora” desde el inicio del documento, pero en sí: ¿qué es una secuencia reguladora y qué proceso de la célula

maneja? Como ya se ha dicho, la secuencia reguladora es un segmento del ADN que ayuda a la célula con los procesos vitales de ésta. Sin embargo, no es tan simple como suena. En la figura 1 se ilustra el proceso de la regulación a grandes rasgos, usando el ejemplo del ARN polimerasa. El ARN polimerasa es el encargado de extraer la información del ADN y luego llevar esa información a determinados sectores de la célula (por ejemplo, la mitocondria), para que pueda llevar a cabo una función específica (digamos la generación de energía produciendo adenosín trifosfato o ATP). Para que se logre llevar a cabo esto, los pasos a seguir son:

1) Búsqueda de los elementos principales. Lo primero que hace la célula para llevar a cabo la transcripción es buscar los elementos necesarios (modelo descrito en la figura 7 de la sección 4.1). Estos elementos son el gen (2.4 de la figura 1) que está encargado del proceso y la secuencia reguladora (2.2) con sus correspondientes elementos de inicio (2.1) y fin (2.3). Los Pasos 1 y 2 en la figura 1 muestran gráficamente este proceso.
2) Identificación de los elementos. Una vez encontrados los elementos antes descritos, la célula empieza el proceso de la transcripción dividiendo la cadena de ADN en dos partes (paso 3).
3) Adherencia del ARN polimerasa. Una vez separada la cadena de ADN, el ARN polimerasa (4.1) se adhiere a un segmento de la cadena dividida y empieza la transcripción de la información (paso 4).
4) Unión de la cadena de ADN. Al terminar el proceso de la transcripción, la cadena de ADN vuelve a unirse como originalmente se encontraba.

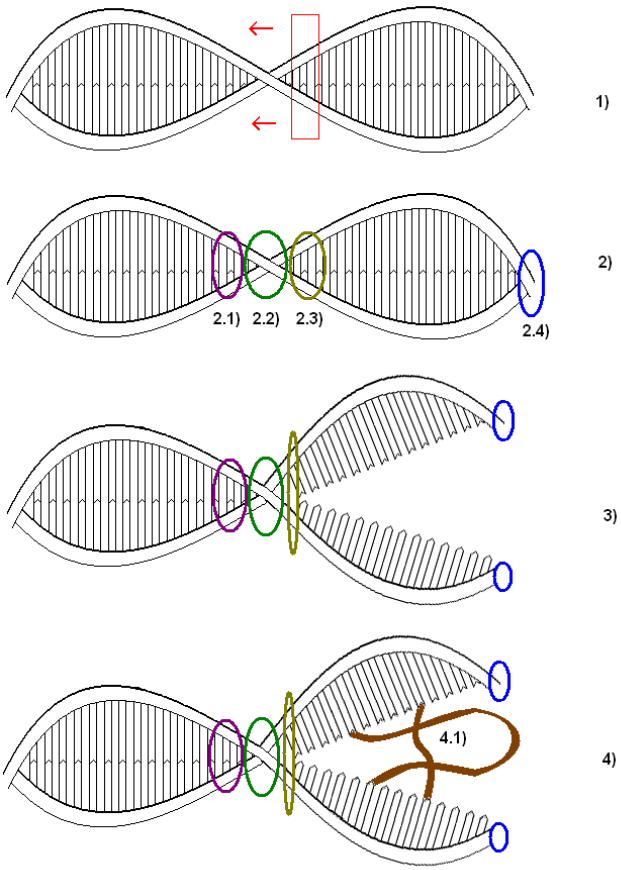


Figura 1. Proceso de la transcripción. 1) Primero se identifican los elementos de la célula. Estos elementos son 2.1) secuencia inicial, 2.2) secuencia reguladora, 2.3) secuencia final y 2.4) inicio del gen. 3) La cadena de ADN se despliega y en 4) el ARN polimerasa [4.1)] se une al ADN para extraer su información.

2.4 Matriz de pesos y posiciones.

Tal como explica D'haeseleer [4], las matrices de pesos y posiciones se utilizan en las búsquedas de motivos y, en nuestro caso, los sitios de transcripción. Para obtener la matriz de pesos, se analizan las cadenas que se quieren utilizar como base para comparar otras cadenas, la obtención se puede observar en la figura 2. Una de las ventajas de las matrices mencionadas con respecto al alineamiento de secuencias múltiples es la velocidad de comparación de secuencias ya que, mientras el alineamiento de secuencia es un problema NP-hard [16], el costo de la evaluación por matrices de peso en el peor de los casos es de $O(n^2)$. Otra ventaja es el significado biológico que se le puede atribuir a las matrices de peso, ya que, no todos los alineamientos tienen significado biológico (y es aún más difícil tratar de hacer un alineamiento con ese significado). Las desventajas de las matrices de peso son la

necesidad de construirlas y definirlas previas a la búsqueda, la cantidad limitada de motivos a buscar que, a diferencia del alineamiento de secuencias múltiples, está limitada por el número de matrices definidas y, por último, si no están bien construidas dan resultados poco fehacientes.

	1	2	3	4	5	6	7	8	9
Secuencia 1	A	A	G	A	C	T	T	A	T
Secuencia 2	A	A	G	T	A	T	T	T	T
Secuencia 3	A	A	C	T	T	T	A	T	G
Matriz de pesos									
A	3	3	0	1	1	0	1	1	0
G	0	0	2	0	0	0	0	0	1
C	0	0	1	0	1	0	0	0	0
T	0	0	0	2	1	3	2	2	2

Figura 2. Ejemplo de una matriz de pesos. Para obtener el peso se calcula la frecuencia de cada elemento de todas las secuencias a incorporar y se acumula el resultado formando así la matriz de pesos.

Para poder determinar si una cadena es adecuada para ser considerada como una secuencia válida según las expectativas, ésta debe superar un límite (threshold) en relación directa con la puntuación obtenida con la matriz de pesos. Utilizando la matriz de pesos de la figura 2, en la figura 3 se muestra como obtener dicha puntuación. Las puntuaciones que se pueden obtener de las matrices están influenciadas por la longitud de éstas y, como explican Pan y Phan [21], se atribuye comúnmente a que las columnas de las matrices definidas son independientes entre sí. Pan y Phan [21] mencionan las complicaciones de encontrar el límite para aprobar las cadenas encontradas. Ellos proponen un método para definir el límite basado en la significancia estadística y promedios. La significancia estadística es que tan probable es que una hipótesis no sea producto del azar. Este acercamiento es interesante, ya que no se ve afectado por la longitud de las secuencias o cadenas y esto es de gran importancia para la nuestra propuesta de solución.

Secuencia a analizar	A	A	G	T	T	T	A	T	G
Matriz de pesos									
A	3	3	0	1	1	0	1	1	0
G	0	0	2	0	0	0	0	0	1
C	0	0	1	0	1	0	0	0	0
T	0	0	0	2	1	3	2	2	2
Resultado	3	3	2	2	1	3	1	2	1
Total= 18									

Figura 3. Para obtener el resultado se utiliza el elemento i-ésimo de la secuencia a analizar y se compara con la matriz de pesos en la i-ésima columna. Se suman todas las ocurrencias y se obtiene el total.

2.5 Minería de datos.

Una de las formas para identificar los motivos dentro de las secuencias es utilizar el alineamiento de secuencias múltiples para obtener las regiones más conservadas. Las regiones con un alto nivel de conservación son las que mayor probabilidad tienen de ser sitios de transcripción. Sin embargo, el hacer alineamiento de secuencias múltiples para muchas secuencias conlleva un alto gasto de recursos y tiempo. El alineamiento es un problema NP-hard [16], por lo que encontrar el alineamiento óptimo para un número grande de secuencias es totalmente impráctico.

Varios métodos que han mostrado una buena eficiencia para el descubrimiento de motivos utilizan el alineamiento de secuencias múltiples (figura 4) para obtener las regiones altamente conservadas restringiendo la búsqueda a un número limitado de secuencias a comparar. Liu et al. [13] combina un algoritmo genético con el alineamiento de secuencias múltiples para encontrar los motivos conservados. Stein et al. [23] con su algoritmo llamado *St-Ga* utiliza el mismo principio. Sin embargo, debido al alto número de genes, utilizar el alineamiento de secuencias múltiples es ineficiente. Ulas y Mehmet [26] presentan un acercamiento para atacar el problema de utilizar el alineamiento de secuencias múltiples al manejar minería de datos, en particular, un acercamiento Top-Down con el algoritmo propuesto por Ester y Zhang [7]. El acercamiento propuesto por Balogu y Mehmet [26] presenta la ventaja de la velocidad, siendo más rápido que *MEME* y *Gibbs Sampler,* tendiendo a ser una excelente alternativa de trabajo; ese rendimiento se muestra en la figura 5.

Figura 4. Ejemplo de alineamiento de secuencias múltiples.

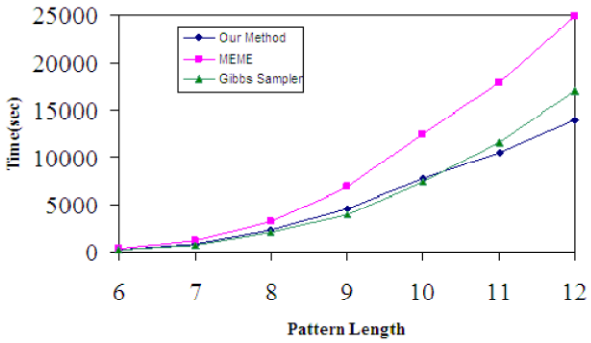


Figura 5. En esta gráfica [26] se muestra la diferencia de tiempo entre el método de Baloglu y Mehmet [26] y dos de los métodos más conocidos para encontrar motivos.

2.6 Secuencias con código inexacto.

Dada la naturaleza evolutiva de los sistemas biológicos, es poco probable que un grupo de secuencias queden inalterables entre los individuos de diferentes especies. Para resolver este problema se maneja la búsqueda de motivos con variaciones (códigos inexactos). En esta dirección se desarrolló el estándar de *IUPAC* para codificación con ambigüedad. Balogu y Mehmet [26] junto con Liu et al. [13] utilizan el estándar para lidiar con la búsqueda de códigos inexactos. El uso del estándar *IUPAC* permite cierto grado de certidumbre sobre la integridad del cambio de elemento, como puede notarse en la tabla 1.

Tabla 1.Representación de la codificación IUPAC para cada nucleótido o combinación de ellos.

IUPAC nucleotide code	Base
A	Adenine
C	Cytosine
G	Guanine
T (or U)	Thymine (or Uracil)
R	A or G
Y	C or T
S	G or C
W	A or T
K	G or T
M	A or C
B	C or G or T
D	A or G or T
H	A or C or T
V	A or C or G
N	any base
. or -	gap

2.7 Optimización mutli-objetivos y función de aptitud.

Los problemas de optimización multi-objetivo (MOP) se diferencian de los problemas de optimización de un sólo objetivo en que, los primeros buscan maximizar o minimizar un conjunto de variables para tratar de llegar a una solución lo más cercana a lo óptimo, mientras que la optimización de un sólo individuo, se enfoca en minimizar o maximizar una única variable. Otra diferencia entre ambos tipos de optimización se basa en los conflictos entre variables que existen en la MOP, ya que normalmente los diferentes objetivos a maximizar, tienden a ser contradictorios. Lo anterior provoca que haya más de una solución óptima al problema de optimización (en caso de que existiera alguna solución óptima). Sin embargo, esto no sucede en la optimización de un solo objetivo, en donde existe una sola variable que no entra en conflicto con ninguna otra, dando como resultado que se pueda alcanzar una solución óptima (cuando existe). La definición del problema de MOP general dada por Coello et al. [3] está descrita en la definición 1.

Definición 1: *Un MOP general es definido como una función* $F(x)=\{f_1(x),\ldots,f_k(x)\}$ *que es minimizada o maximizada sujeto a una* $g_i(x)\leq 0$, $i=\{1,\ldots,m\}$, *y a una* $h_j(x)=0$, $j=\{1,\ldots,p\}$ $x \in \Omega$. *Una solución de un MOP minimiza o maximiza los componentes de un vector F(x) donde x es una variable de un vector de decisiones de n dimensiones de un universo Ω. Ω contiene todas las posibles x que pueden ser usadas para satisfacer la evaluación de F(x) y, tanto* $g_i(x)\leq 0$ *como* $h_j(x)=0$, *representan restricciones que deben cubrirse mientras se maximiza o minimiza F(x).*

Esto significa que un MOP consiste en un grupo de *k* objetivos de la función, un total de *n* variables de decisión y *m + p* restricciones en los objetivos de la función. Cabe aclarar que los *k* objetivos pueden ser lineales o no lineales y continuos o discretos. También el vector de variables de decisión x_i puede ser continuo o discreto.

La optimización multi-objetivo busca encontrar aquellas soluciones en las que ya no se pueda maximizar algún aspecto (objetivo), sin que se afecte otro objetivo. Estas soluciones se encuentran en el frente de Pareto. El frente de Pareto (figura 6) es la frontera donde se encuentran todas las soluciones que cumplen con la condición antes mencionada y está definida por la definición 2 establecida por Coello et al. [3]. Las definiciones 3 y 4 también son establecidas por Coello et al. [3].

Definición 2 (Frente de Pareto): *Para un MOP dado, F(x), y un set óptimo de Pareto, P*, el frente de Pareto es definido como:* $PF^* := \{u = F(x) | x \in P^*\}$

Definición 3 (Set óptimo de Pareto): *Para un MOP dado, F(x), el set óptimo de Pareto, P*, se define como:* $P^* := \{x \in \Omega \mid \neg\exists x' \in \Omega\, F(x') \preccurlyeq F(x)\}$

Definición 4 (Óptimo de Pareto): *Una solución x ϵ Ω se dice que es un óptimo de Pareto con respecto a (w.r.t.) Ω si y sólo si no hay una x' Ε Ω por el cual v= F(x') = (f_1(x') …, f_k(x')) domina a u= F(x) = (f_1(x) …, f_k(x)). La frase óptimo de Pareto es tomada tal como significa con respecto a todo el espacio de variables de decisión a menos que se especifique otra cosa.*

La función de aptitud es la que guía al Algoritmo Genético (AG) para encontrar las soluciones más prometedoras, al calificar la población generada por éste. A diferencia de un acercamiento mono-objetivo, la optimización mediante múltiples objetivos tiene la dificultad de que pueden existir más de una solución que resuelva el problema de la mejor forma posible. Dado este problema, se emplea la optimización por Pareto para determinar la dominancia entre los individuos dentro de la población de los Algoritmos Genéticos. Un individuo "A" domina a un individuo "B" sí y sólo si A≥B y debe haber al menos un elemento en A que sea mayor absoluto que B. El algoritmo para la obtención del frente de Pareto que se usa en este trabajo de investigación es el propuesto por Deb et al. [5], ya que ha mostrado su eficiencia y ha sido utilizado en otros trabajos, por ejemplo, en el de Mehemt [17].

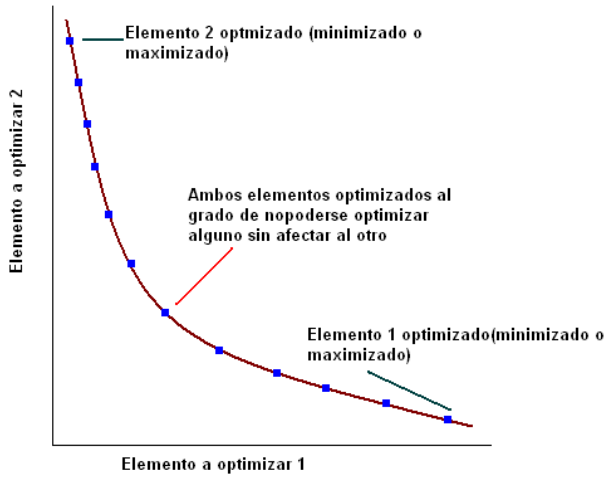


Figura 6. Ejemplo del frente de Pareto usando dos funciones cualquiera a optimizar.

Capítulo III
TRABAJO PREVIO

En esta sección se presenta un resumen de algunos de los trabajos que se han realizado. En la tabla 2 se puede observar la diferencias entre nuestra solución y lo que se ha trabajado al momento de escribir la tesis. La tabla 3 compara los algoritmos genéticos que se han usado y el nuestro, con respecto a lo sensible que es la función de aptitud referente a la longitud de la cadena encontrada.

Tabla 2. Posicionamiento de la propuesta dentro del estado del arte.

Método	Año	Estocástico	Multi-objetivos	Tamaño definido
MEME	1994	NO	NO	SI
Gibbs	2003	SI	NO	SI
MOGAMOD (AG)	2007	SI	SI	NO
ST-GA	2003	SI	NO	NO
AG (otros)	-	SI	NO	SI
IPSO-GA	2005	SI	NO	SI
Propuesto	-	SI	SI	NO

Tabla 3. Diferencia entre la propuesta y los demás algoritmos genéticos

Método	Aptitud variable con la longitud
MOGAMOD (AG)	SI
MDGA	SI
TD-GA	SI
ST-GA	SI
FMGA	SI
IPSO-GA	SI
Propuesto	Parcialmente NO

Bailey y Elkan [1] utilizan maximización de expectativa (*Expectation Maximization*) para trabajar el problema del descubrimiento de motivos con su algoritmo llamado *MEME*. Ellos buscan encontrar múltiples motivos al complementar un modelo de combinaciones finitas de dos componentes con las cadenas, eliminando de manera probabilística las ocurrencias de los motivos que han encontrado y repiten el proceso para encontrar motivos contiguos. *MEME* es uno de los algoritmos más reconocidos en el campo del descubrimiento de motivos, sin embargo, tiene la reputación de ser muy lento pero con buenos resultados.

Thomson et al. [25] proponen el algoritmo llamado *Gibbs Sampler* para el descubrimiento de motivos que utiliza el procedimiento de Monte Carlo para cadenas de Markov. Una de las principales características es el uso de modelos de motivos en forma de modelos de productos multinomiales que capturan los patrones comunes. Un producto multinomial es la unión de las distribuciones de dos o más distribuciones multinomiales independientes. Una distribución mutlinomial es una generalización de la distribución binomial en la cual, el interés de estudio, no es la ocurrencia de un solo caso o su opuesto, sino de tres o más suceso. La ventaja principal de este método es la velocidad con la que encuentra los patrones, pero su principal desventaja es la calidad de los patrones que encuentra, en cuanto al tamaño y soporte de las secuencias encontradas.

Zhou et al. [28] proponen la combinación de la optimización por enjambre de partículas y los Algoritmos Genéticos. Ellos presentan un acercamiento para seleccionar la información más relevante para la búsqueda de motivos en las secuencias largas en las regiones rio arriba. El método propuesto es prometedor ya que presentan resultados alentadores, dado que es de los primeros enfoques en fusionar dos algoritmos y la descripción de los resultados obtenidos. Lamentablemente, en sus artículos no presentan el tiempo de ejecución del programa y la comparación con otros algoritmos.

Dongsheng et al. [6] utiliza *MDGA*, un algoritmo genético para la búsqueda de motivos en secuencias de ADN en genes homólogos. Una de las ventajas descritas de este trabajo es que, a diferencia de varios programas de búsqueda de motivos, la complejidad no aumenta drásticamente con la longitud del motivo a buscar. Las soluciones que presenta son mejores que las alcanzadas por *Gibbs Sampler* [25].

Stein et al. [23] emplean, para la búsqueda de motivos altamente conservados en regiones rio arriba, su Algoritmo Genético Estructurado. Para hacer la búsqueda de los motivos utiliza el alineamiento de secuencias múltiple para después encontrar las regiones más conservadas. La desventaja es el tiempo consumido por el alineamiento y el número de secuencias a buscar, ya que el alineamiento aumenta su complejidad al aumentar el número de elementos en la búsqueda.

Baloglu y Mehmet [26] utilizan una mezcla de Algoritmos Genéticos con un acercamiento Top-Down de minería de datos para eliminar la necesidad de utilizar el alineamiento de secuencias múltiples. Este acercamiento mostró una mayor velocidad que *Gibbs Sampler* [25], por lo que es una opción interesante para trabajar con grandes volúmenes de datos.

Otro algoritmo genético utilizado para el descubrimiento de motivos es el *FMGA*, propuesto por Liu et al. [13]. Este método utiliza el conjunto de operadores descritos por Notredame y Higgins en *SAGA* [2]. Una de las ventajas que tiene con respecto a otros trabajos es el uso de códigos con ambigüedad establecidos por el IUPAC. Los resultados obtenidos en este trabajo son mejores que los obtenidos por *MEME* y *Gibbs Sampler,* con la ventaja de ser más rápido que *MEME* pero más lento que *Gibbs Sampler*.

Parte del modelo multi-objetivo propuesto para esta solución es planteado por Mehmet [17]. En su planteamiento presenta la necesidad de manejar múltiples objetivos debido a las diferentes cualidades que presentan los motivos. El enfoque que utiliza Mehmet [17] está basado en los Algoritmos Genéticos y busca maximizar

similitud, longitud del motivo y soporte. La principal ventaja que tiene este acercamiento es la calidad de los motivos encontrados.

La diferencia principal con el trabajo antes mencionado es el enfoque de búsqueda de motivos, mientras Mehmet [17] busca un motivo con varias repeticiones nosotros buscamos dos motivos con repeticiones (uno inicial y otro final) y la secuencia reguladora que se encuentra dentro de esos dos motivos. Esto hace más compleja la búsqueda en comparación con el trabajo de Mehmet [17]. Al momento de la búsqueda del trabajo previo, sólo Mehmet [17] y la solución propuesta en esta tesis, manejan un enfoque multi-objetivo para la búsqueda de motivos.

Capítulo IV
ESTRUCTURA DE LA SOLUCIÓN PROPUESTA Y PROCEDIMIENTOS

En esta sección se describe cómo está estructurada la solución, los elementos que conforman el programa y los procedimientos y pasos que se efectuaron para tratar la información y así obtener la salida esperada.

4.1 Estructura de la solución propuesta.

Para obtener las posibles secuencias reguladoras se planteó, el modelo descrito en la figura 7. Una secuencia reguladora está conformada (la mayoría de las veces, aunque pueden existir excepciones) por un motivo inicial de tamaño Ω, el cual se encuentra antes de la secuencia reguladora (con tamaño β), y la terminación del motivo inicial indica donde inicia la secuencia. La secuencia final es el motivo (de longitud λ) encargado de marcar el final de la secuencia reguladora y marcar el inicio de la transcripción (proceso en el cual se activan enzimas y proteínas para el funcionamiento de la célula). La secuencia reguladora, junto con sus motivos inicial y final, se encuentran a una distancia *D* del gen, pero dentro de los primeros 220bp (pares de base).

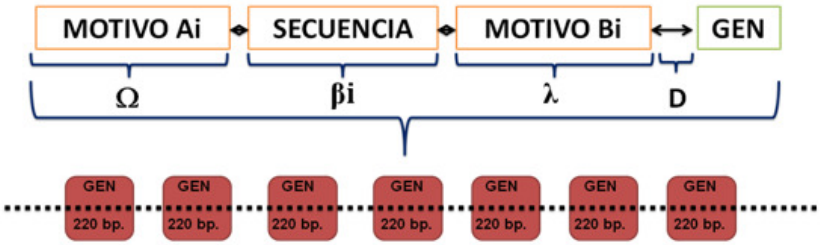


Figura 7. Diagrama de la estructura planteada para solucionar el problema de encontrar las secuencias reguladoras.

4.2 Diagrama de la solución propuesta.

En esta sub sección se presenta el panorama general de la solución planteada (diagrama 1), así como una descripción de cada elemento que lo conforma.

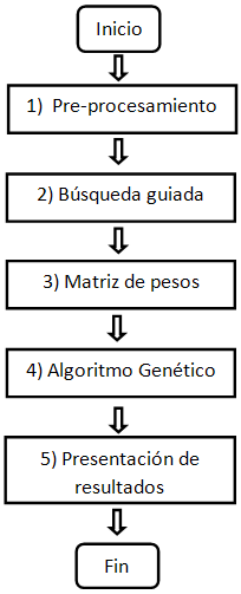


Diagrama 1. Diagrama de el procedimiento de la solución.

1. Pre-procesamiento: En este paso se eliminan los datos innecesarios (figura 8A), se identifican los datos de los genes y genes complemento con su correspondiente conversión (figura 8B). Esta conversión sólo se aplica a los genes complementos ya que, estos van en dirección contraria a la de los genes normales. Esta diferencia se debe a la estructura de doble hélice del ADN. La conversión es sencilla, y consiste en invertir el orden de la cadena (el primer elemento pasa a ser el último y viceversa). Después se sustituyen los nucleótidos: la A por la G, la G por la A, la T poro la C y la C por la T. Lo anterior corresponde a los pasos del 1 al 4 en el pseudocódigo dado al final de esta sub-sección.

```
LOCUS       NC_000964             4215606 bp    DNA     circular BCT 12-AUG-2009
DEFINITION  Bacillus subtilis subsp. subtilis str. 168, complete genome.
ACCESSION   NC_000964
VERSION     NC_000964.3  GI:255767013                    A)
DBLINK      Project:76
KEYWORDS    complete genome.
```

```
FEATURES             Location/Qualifiers
     source          1..4215606
                     /organism="Bacillus subtilis subsp. subtilis str. 168"
                     /mol_type="genomic DNA"
                     /strain="168"                    B)
                     /db_xref="taxon:224308"
     gene            410..1750
                     /gene="dnaA"
                     /locus_tag="BSU00010"
                     /db_xref="GeneID:939978"
```

Figura 8. Pre-procesamiento de los datos. En A) Se elimina todo lo que no es usado dentro del programa y después, en B) Se identifica el nombre del gen, posición inicial, terminación, locus y si es o no es complemento.

2. Búsqueda guiada: Esto se hace buscando motivos que estén dentro de los parámetros y las características establecidas en un consenso previamente definido (Anexo 1) y considerando el estándar IUPAC para los casos con ambigüedad. Esta búsqueda es secuencial y se hace comparando cada elemento del consenso con cada cadena de obtenida del procesamiento 1). El consenso es un conjunto elementos conocidos para los genes bacterianos que nos proporcionó el especialista (Dr. Candelario Vázquez Cruz, BUAP). El resultado de la búsqueda son cadenas con alta probabilidad de ser secuencias reguladoras (tabla 4). A esta búsqueda le corresponden las líneas 5 y 6 del pseudocódigo.

Tabla 4. Segmento de resultados de la búsqueda con conocimiento.

#	Gen	Inicio	Fin	Secuencia encontrada	Posición
1	yabE	48629	49942	acatgctttcttctc	48520
2	yabG	51680	52552	accacatggactgccgc	51619
3	veg	52763	53023	ttgacaacgtcttattaa	52648
4	ctc	58783	59397	aaatccttatcgtt	58645
5	ybcH	210224	210514	acatcatcccctagtgcc	210218
6	ybzH	211429	211731	ttgacatatcgaaata	211258
7	pssA	247744	248277	gaatgctgctttttt	247667
8	nagP	254907	256802	gaatgatatgaatat	254786
9	trnS-Asn	528704	528778	ttgacactgcaaatcaa	528544

3. Matriz de pesos: A partir de los resultados de la búsqueda guiada, se crea una matriz de pesos (capítulo 2 sección 2), para ser utilizada en la futura evaluación de las cadenas encontradas y determinar así la probabilidad de ser una secuencia reguladora (figura 9). Este es el paso número 7 del pseudocódigo.

```
A:  0.547619    0.190476  0.535714
G:  0.190476    0.0119048 0.25
C:  0.0833333   0.571429  0.0952381
T:  0.178571    0.22619   0.119048
```

Figura 9. Ejemplo de una matriz de pesos generada con los resultados de la búsqueda con conocimiento

4. Búsqueda de motivos: Utilizando un Algoritmo Genético multi-objetivo, parte central de este trabajo, se realizará una búsqueda de motivos (paso 8 del pseudocódigo).
5. Presentación de resultados: Se presentan los individuos determinados por el algoritmo como más aptos, como el resultado final del proceso. La línea 9 del pseudocódigo corresponde a este paso.

El pseudocódigo del programa a utilizar, se muestra a continuación.

```
Main()
Begin:
Vector <base_datos> BD;
Vector <base_conocimiento> BC;
Vector <resultados> RES_BC,RES;
Vector <matriz_pesos> MP;
1. Cargar_base_datos(BD);
2. Eliminar_basura(BD);
3. Obtener_genes(BD);
4. Obtener_genes_complemento(BD);
5. Cargar_base_de_conocimiento(BC);
6. RES_BC=Busqueda_guiada(BD,BC);
7. MP=sacar_matriz_pesos(RES_BC);
8. RES=algoritmo_genetico(BD,MP);
9. Presentar_resultados(RES);
End Main();
```

4.3 Diagrama del Algoritmo Genético.

En esta sub sección se desglosa el algoritmo genético que se usó para desarrollar esta solución. En el diagrama 2 se puede observar este desglose en sus componentes básicos. Más adelante, en esta misma sub sección se presenta el pseudocódigo utilizado.

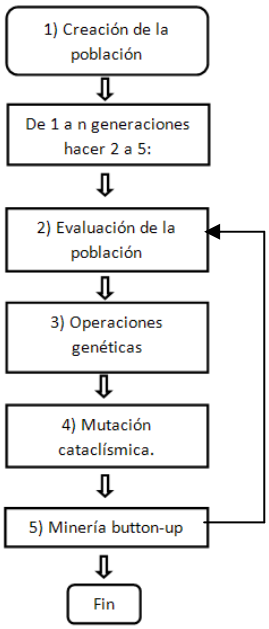


Diagrama 2. Diagrama del procedimiento del Algoritmo Genético.

1. Se crea la población inicial. La población es generada aleatoriamente. Paso 7 y 8 del pseudocódigo del Algoritmo Genético que está más adelante.
2. Evaluación de la población: Cada individuo es evaluado y ordenado según el frente de Pareto. Paso 9 del pseudocódigo.
3. Operaciones genéticas: Se aplican los operadores genéticos de cruza, mutación, selección y elitismo. Pasos del 11.1 al 11.4.
4. Mutación cataclísmica: Si los individuos no mejoran en determinado número de generaciones, la población es mutada drásticamente, a excepción de los individuos élites. Paso 11.3.
5. Minería *button-up*: En cada generación se hace una búsqueda con la finalidad de incrementar el número de cadenas, lo que aumenta el soporte en cada individuo dentro de la población. Paso 11.4 del pseudocódigo.

El pseudocódigo del algoritmo genético encargado del descubrimiento de conocimiento se describe en los siguientes renglones.

```
Vector  <resultados> algoritmo_genetico(Vector <base_datos> BD, Vector <matriz_pesos> MP)
1.  Begin:
2.  Int generaciones, i;
3.  Vector <Float> Fitness;
4.  Vector <poblacion> padres, hijos;
5.  población elite;
6.  Fitness=reserva_espacio(N_poblacion*2);
7.  padres=crea_poblacion(N_poblacion);
8.  padres=bottom_up(padres, BD);
9.  Fitness=evalua_población(padres, MP);
10. padres=rearreglo_poblacion(padres, Fitness);
11. For i=0 to N_generaciones begin:
    11.1.       elite=elitismo(Fitness, padres);
    11.2.       hijos=cruza(padres);
    11.3.       hijos=mutación(hijos);
    11.4.       hijos=bottom_up(hijos, BD);
    11.5.       Fitness=evalua_poblacion(hijos, MP);
    11.6.       padres=sustitución(padres,hijos);
    11.7.       if i%N := 0 then
        11.7.1.  padres=rearreglo_poblacion(padres, Fitness);
    11.8.       end if;
12. end for;
13. end;
```

4.4 Función de aptitud multi-objetivo.

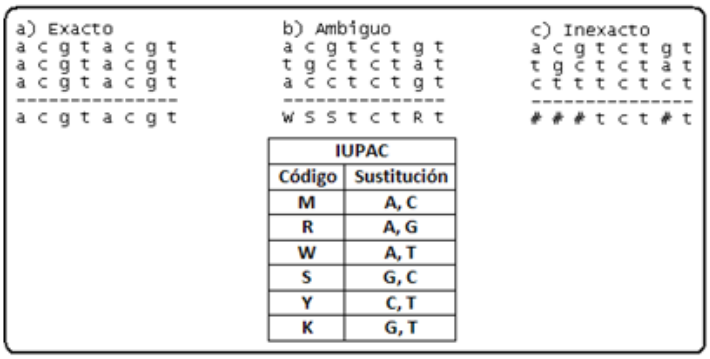
Para encontrar cadenas con diferentes cualidades pero que todas éstas sean satisfechas, es necesario tener un acercamiento que englobe todos los objetivos. Por lo tanto, la mejor forma de atacar el problema es con la teoría de múltiples objetivos u multi-objetivo. Los objetivos con los que trabajamos en esta solución son:

- Tamaño de la secuencia: El tamaño de la secuencia juega un papel primordial para determinar la importancia, tanto del motivo como de la secuencia encontrada. Sin embargo, no todas las secuencias siguen esta regla y, también, las secuencias pequeñas tienen importancia y significado biológico. Debido a esto, se debe crear una función de evaluación que mantenga cierta relación con el tamaño de la secuencia pero sin menospreciar secuencias pequeñas. El tamaño de la secuencia es evaluada en conjunto con la similitud.
- Soporte: El soporte, como en minería de datos, indica qué tanto una secuencia se repite dentro de la base de datos. Entre más se repita la secuencia mayor será su soporte. Un individuo que tiene 4 grupos de cadenas tiene un soporte de 4.
- Similitud. La similitud significa qué tan semejantes son dos secuencias entre sí. Existen tres aspectos a evaluar, cada aspecto tiene su peso en la evaluación para este atributo.
    - Cadena idéntica: Es cuando dos o más cadenas son exactamente iguales. Este es el aspecto de mayor peso (Figura 10a). El peso que se propuso para este aspecto es de 1.
    - Código con ambigüedad: Cuando las combinaciones de nucleótidos de tres o más columnas de diferentes secuencias corresponden a variantes en la codificación *IUPAC* para códigos ambiguos (Figura 10b). Éstos son los segundos con mayor peso dentro de la función de aptitud. El peso para este aspecto es de 0.2.
    - Código inexacto: A diferencia de los códigos con ambigüedad, los códigos inexactos son columnas que no tienen una representación dentro de la codificación IUPAC a dos caracteres utilizada en esta

solución (Figura 10c). El peso para las columnas con código inexacto es de -1.

Para calcular la similitud se utilizó la ecuación , donde *Ci* es el peso dado a la columna según las reglas mencionadas anteriormente y *n* es el número de columnas. Aunque la ecuación sea un simple promedio, los pesos otorgados castigan severamente a las secuencias pequeñas que tengan código inexacto y/o ambiguo para evitar que sean secuencias repetidas sin significado biológico (entre más pequeña es una secuencia más fácil es encontrar otra secuencia idéntica o similar). Mientras que para las cadenas grandes los pesos no las castigan tan severamente. Esto nos permite darle valor a las secuencias grandes pero sin menospreciar secuencias pequeñas bien conservadas que pueden ser importantes.

- Matriz de pesos: Las cadenas que se han considerado secuencias reguladoras se evalúan con esta matriz para determinar la semejanza que tienen con las secuencias encontradas con el consenso. En el capítulo 2 en la sección “Matriz de pesos y posiciones” se muestra como se calcula susodicha matriz.



IUPAC	
Código	Sustitución
M	A, C
R	A, G
W	A, T
S	G, C
Y	C, T
K	G, T

Figura 10. Ejemplo de la similitud entre secuencias: a) secuencias exactas, b) secuencias ambiguas y c) secuencias inexactas.

Una vez determinado los valores de cada objetivo se pasan a evaluarlos para encontrar el frente de Pareto (capítulo 2 en la sección “optimización multi-objetivos y función de aptitud”). Para determinar el frente de Pareto se debe establecer la dominancia de los individuos.

4.5 Estructura de los individuos.

La estructura de los individuos es la representación de la solución que el Algoritmo Genético opera para obtener el o los resultados deseados. Es de suma

importancia diseñar lo mejor posible al individuo. Un buen diseño puede hacer más eficiente la evaluación de los individuos, facilitar las operaciones genéticas y, en mayor grado, hace posible tener resultados coherentes y de buena calidad. En la figura 11 se muestra la base propuesta para los individuos que se usaron para los experimentos.

# Cadenas	Gen	Posición inicial	Tamaño Ω	Motivo Ai	Tamaño βi	Secuencia reguladora	Tamaño λ	Motivo Bi

Figura 11. Estructura de los individuos usados en el Algoritmo Genético usado en esta tesis.

El número de cadenas es la cantidad de genes que se comparan, el mínimo de cadenas a usar es 2 y no existe un máximo de cadenas. El gen es, simplemente, el gen que se está analizando y no puede ser el mismo dentro de un mismo individuo. La posición inicial es la posición relativa en los 220bp que se toman a partir del inicio del gen. Los tamaños de motivos Ω, βi y λ y los motivos Ai, Bi y secuencia reguladora están descritos al inicio de este capítulo. La posición inicial no puede ser menor a 0bp y la suma de los tamaños Ω, βi y λ mas la posición inicial no puede sobrepasar los 220bp. Se usó en todos los casos una numeración decimal y el alfabeto tradicional.

4.6 Operadores genéticos.

Los operadores genéticos son los encargados de evolucionar a los individuos al ejercer una fuerza de cambio en ellos, como la mutación, o una interacción entre los individuos, como la cruza. Se utilizó la cruza en un punto con una probabilidad del 100%. La probabilidad de la mutación es de 5% y, dado que son decimales los valores a mutar, la mutación se hizo con la suma y resta (la probabilidad de seleccionar una de las dos es de 50%) con un valor de 1 teniendo en consideración de no sobrepasar los límites de 0bp y 220bp y la selección del elemento a mutar se hizo de forma aleatoria. La selección fue por torneo y se uso elitismo para conservar al o los individuos más aptos. Para el Algoritmo Genético CHC la mutación cataclísmica se hizo cada 100 generaciones. La descripción de los operadores genéticos que se utilizaron para el desarrollo de esta tesis se discutieron en el capítulo dos en la sección “Algoritmo Genéticos”.

4.7 Cálculo de función de aptitud.

Los rangos de valores que definen que tan buenos o malos son los resultados para cada parte del algoritmo son:

- Soporte: El soporte va de 2 hasta donde sea posible con los recursos computacionales, sin embargo, gracias al acercamiento *bottom-up*, el cual busca calidad antes de cantidad se puede esperar un soporte en promedio de 5. Esto puede mejorar si se aumenta el número de generaciones (cada generación se hace una búsqueda para intentar incrementar el soporte) o disminuir la calidad de la búsqueda.
- Matriz de pesos: Para definir el valor máximo que puede alcanzar se deben considerar dos opciones. Una es la longitud de la cadena a evaluar, ya que ésta modifica el valor promedio que se obtiene de la matriz de pesos. La segunda consideración son los valores de la matriz de pesos, dado que éstos cambian según los parámetros de la búsqueda guiada. Para una mejor referencia sobre estos resultados, la tabla 5 muestra el promedio máximo que se puede obtener para cada longitud.

Tabla 5. Valor máximo obtenible de la matriz de pesos para cada longitud.

Tamaño de la secuencia	Valor máximo
4	0.58
5	0.6
6	0.53605433
7	0.47857125
8	0.45677
9	0.4488235
10	0.43965086
11	0.42861338
12	0.42903767
13	0.4239717
14	0.41736982
15	0.4141205
16	0.40715192
17	0.40117886
18	0.3984336
19	0.39326831
20	0.39366429

- Promedio del tamaño de las secuencias reguladoras propuestas: Simplemente es la media del tamaño de las cadenas propuestas como secuencias reguladoras de cada individuo, el promedio mínimo es de 4 y el máximo es de 20.
- Similitud: La similitud se refiere a qué tan identicas dos o más cadenas son entre sí. El valor teórico máximo y mínimo mantiene una relación directa con los valores de penalización y premiación sobre los valores de la codificación IUPAC (En la sección anterior se describieron los elementos de la similitud). Los valores para esta prueba para cada elemento son:
    - Exacta: 1
    - Ambigua: 0.2
    - Inexacta: -1

    Dados los valores anteriores, el promedio máximo teórico es de 1 y el mínimo es de -1. Sin embargo, el valor máximo es muy difícil de alcanzar cuanto mayor es el soporte del individuo y la longitud de la cadena analizada.

En los siguientes párrafos se describe una muestra de los individuos que se obtuvieron para ejemplificar lo que serían buenos y malos individuos. También se hace un pequeño análisis de cada individuo y un panorama de cómo están constituidos, junto con sus valores y descripciones.

Individuo 1.

El individuo 1 es considerado un buen individuo. Se destaca por una longitud en la cadena inicial de buen tamaño con una muy buena similitud, considerando que el soporte es de buen tamaño. La similitud de la secuencia final también es buena, pero, la evaluación de la matriz de peso no es de las mejores. La figura 12 y 13 corresponden al grupo de cadenas inicial y final, respectivamente, que se detallan en el modelo propuesto en la figura 7 del capítulo 4 en la sección “Estructura de la solución” y, en la figura 11, en la sección del mismo capítulo “Estructura de los individuos”.

Soporte= 6

Matriz de pesos= 0.298418

Promedio del tamaño de las secuencias reguladoras propuestas=16.5596

Secuencias reguladoras propuestas:

AAAATAATATCGACAGTTT
AAACAGTTTATGTCCCT
TCACATAAGCTTGGCCTCA
TCATGTTTTCCCCTCA
ATGAAACTGTTTTGC
ATGAAACTGTTTTG

A	T	A	G	A	T	T	A	T	A	T	T	A	A	G
A	A	A	T	T	G	T	T	T	C	T	T	A	A	T
C	A	T	G	T	C	T	T	A	A	T	A	G	A	T
C	T	T	T	C	G	T	T	T	C	T	T	G	A	T
C	T	T	T	T	T	T	T	A	C	T	T	A	T	T
C	T	T	T	T	T	T	T	A	C	T	T	A	T	T
M	W	W	K	#	#	T	W	W	M	T	W	R	W	K
0.2	0.2	0.2	0.2	-1	-1	1	0.2	0.2	0.2	1	0.2	0.2	0.2	0.2

Figura 12. Cadena inicial del individuo 1. Se pueden observar dos regiones muy bien conservadas de T (Tiaminas), regiones conservadas de dos elementos (ambiguas) y sólo dos regiones inexactas.

Similitud=(0.2+0.2+0.2+0.2-1-1+10.2+0.2+0.2+1+0.2+0.2+0.2+0.2)/15=5.5/15=0.146667

Nota: Las letras diferentes a A, G, C, T son elementos de la codificación IUPAC a dos elementos y, el guión, indica que el elemento no se encuentra dentro de la codificación IUPAC a dos elementos.

T	T	T	T	A	C	A
T	T	T	A	A	A	T
T	T	T	T	A	C	A
T	T	T	C	T	A	T
G	T	T	T	A	A	T
C	G	T	T	T	A	A
#	K	T	#	W	M	W
-1	0.2	1	-1	0.2	0.2	0.2

Figura 13. Cadena final del individuo 1. Al igual que en la figura 11, en esta figura se puede observar una región muy conservada de T con una mayoría de regiones ambiguas y dos regiones inexactas.

Similitud= -0.02857

Individuo 2.

El individuo 2 es otro ejemplo de un buen individuo teniendo un buen soporte y sus cadenas inicial y final también tienen una buena similitud entre ellas. La evaluación

de la matriz de pesos no es muy buena, sin embargo, el tamaño promedio de las secuencias reguladoras propuestas es de muy buen tamaño. Las figuras 14 y 15 corresponden a las cadenas inicial y final de este individuo.

Soporte: 6
Matriz de pesos: 0.284412
Tamaño promedio de las secuencias reguladoras propuestas: 19.9069
Secuencias reguladoras propuestas:

TCACAATAGAGGAATTGTCG
ATAGTATGGATGAATGGCTTC
TCCTCATTGTGTCTAGTTAAA
TCGTAATAGATGCAAC
ATAGTATGGATGAATGGCTTC
ATAGTATGGATGAATGGCTTC

T	G	G	C	A
T	T	T	C	G
T	T	T	A	G
G	T	G	C	A
T	T	T	C	G
T	T	T	C	G
K	K	K	M	R
0.2	0.2	0.2	0.2	0.2

Figura 14. Esta cadena inicial tiene todas sus columnas como códigos con ambigüedad, es decir, no tiene ni regiones exactas pero tampoco inexactas.

Similitud= 0.2

A	G	G	G	A	G	A	C	A	A	T	G	A	T	T
G	G	C	A	A	T	T	T	T	T	T	G	A	A	G
A	T	T	G	G	A	A	T	T	T	T	A	A	A	G
A	T	A	A	A	A	T	T	T	G	A	T	A	T	T
G	G	C	A	A	T	T	T	T	T	T	G	A	A	G
G	G	C	A	A	T	T	T	T	T	T	G	A	A	G
R	K	#	R	R	#	W	Y	W	#	W	#	A	W	K
0.2	0.2	-1	0.2	0.2	-1	0.2	0.2	0.2	-1	0.2	-1	1	0.2	0.2

Figura 15. La cadena final que corresponde al individuo dos. Ninguna secuencia exacta con cuatro cadenas inexactas, aún dominando las secuencias ambiguas.

Similitud=-0.066666

Individuo 3.

El individuo 3 es el ejemplo de un mal individuo, la similitud de ambas cadenas es negativa, lo que indica una similitud a dos caracteres muy pobre y más si

consideramos que tiene muy poco soporte (entre mayor sea el soporte, más difícil es encontrar una buena similitud entre cadenas). Si la similitud es negativa aumenta la probabilidad de que los motivos encontrados para identificar las secuencias reguladoras no tengan significado biológico, por lo tanto, disminuye la probabilidad de encontrar secuencias reguladoras verdaderas. Sin embargo, el valor obtenido por la evaluación con la matriz de pesos es bueno, lo que no invalida por completo el resultado obtenido. Las cadenas inicial y final se muestran en la figura 16 y 17

Soporte: 4

Matriz de pesos: 0.364551

Tamaño promedio de las secuencias reguladoras propuestas: 7.74597

Secuencias reguladoras propuestas:

ATAT
AAAT
AATAAAAAGAAAAAG
GCATGCCTTCCATCC

G	G	G	T	G	T	T	A	T
T	A	T	T	T	A	T	C	C
G	G	G	T	G	C	G	G	A
T	G	A	G	G	C	G	A	A
K	R	#	K	K	#	K	#	#
0.2	0.2	-1	0.2	0.2	-1	0.2	-1	-1

Figura 16. Cadena inicial del tercer individuo. Se puede notar una gran cantidad de secuencias inexactas, ninguna secuencia exacta.

Similitud= -0.3333

G	A	T	T	A	G	T	T	A	A	T	A
T	C	C	G	C	C	A	T	G	A	T	G
C	C	T	C	A	A	T	A	G	C	C	T
A	T	T	T	A	T	T	T	A	C	G	A
#	#	Y	#	M	#	W	W	R	M	#	#
-1	-1	0.2	-1	0.2	-1	0.2	0.2	0.2	-1	-1	-1

Figura 17. Cadena final del tercer individuo. Igual que su contraparte inicial, un gran número de secuencias inexactas con ninguna secuencia exacta.

Similitud= -0.4

Capítulo V
PRUEBAS Y RESULTADOS

En esta sección se presentan los resultados obtenido después de la fase de experimentación. Esta divida en 4 partes. La primera parte es interpretación y es donde se describe cómo se debe entender los resultados. La segunda sección es la de banco de pruebas y da a conocer las bases de datos usados y los argumentos usados. La tercera sección es la de análisis de resultados y gráficas y es la sección donde se da a conocer los resultados y se hace un análisis de ellos.

5.1 Interpretación

Para darles un significado tanto computacional como biológico se contó con la guía de los doctores Candelario Vázquez Cruz y María Patricia Sánchez Alonso, ambos doctores en microbiología de la BUAP. Para poder ser graficadas las cadenas encontradas fueron agrupadas en segmentos de 5 bp, empezando desde el inicio de la cadena y se graficó su frecuencia por grupo. En la figura 18 se puede observar un ejemplo. El inicio de la cadena o ventana indica a qué grupo pertenece, se contabiliza el número de cadenas que existe en cada grupo y se grafica. Esta agrupación se debe a que el sistema no da una solución exacta y que las cadenas encontradas pueden estar desplazadas levemente (pero aun manteniendo nichos de elementos conservados entre ellas).

Ventana 1		A	C	T	G	G	T								
Ventana 2						G	T	T	C	C	C	T	A	A	
Ventana 3								T	C	C	C				
Ventana 4											C	T	A	A	A
Ventana 5			C	T	G	G	T								
Secuencia	A	A	C	T	G	G	T	T	C	C	C	T	A	A	A
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Grupo 1					Grupo 2					Grupo 3				

Figura 18. Ejemplo de agrupamiento, el inicio de cada ventana determina el grupo al que pertenece.

5.2 Bancos de pruebas.

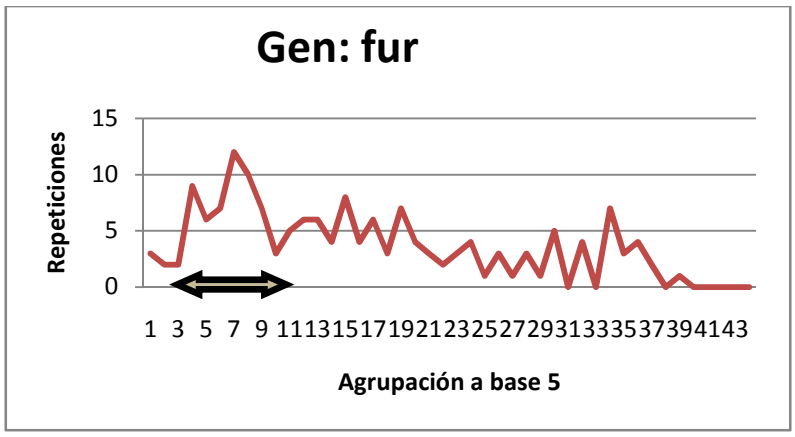
Se trabajó con la base de datos de *Bacillus subtilis* obtenida de NCBI [24] de la cual se extraen 4,423 cadenas (correspondiente al número de genes) con una longitud de 220 pares de bases (bp) cada una. Se hicieron 430 experimentos con el Algoritmo Genético CHC, descrito en el capítulo 2 en la sección "Algoritmos Genéticos", con 50 individuos y 5000 generaciones en cada experimento. Se hicieron cuatro grupos de experimentos, cada uno organizado según el número de genes involucrados. Algunos genes tienen relación entre ellos al ser de la misma familia. Otros genes, como el dnaA, son genes que se encuentran en muchos organismos y son vitales para el funcionamiento de la célula u organismo. El primer grupo de 100 experimentos se tomaron 59 genes (dnaA, fur, el grupo spo, scuA y gerE) y sus características son:

- dnaA: Este gen es uno de los más importantes, sin éste el organismo simplemente no podría vivir. Es el gen encargado de la inicialización de replicación que promueve el des-enrollamiento durante la replicación del ADN en los procariontes.
- El grupo spo: Es un grupo de genes que se activan en determinados momentos y que, mientras algunos son vitales para el organismo, otros son circunstanciales.
- scuA: es el encargado de sintetizar la enzima *cytochrome c oxidase,* la cual está dentro del proceso de respiración de la célula.
- fur: este gen es el encargado del transporte de hierro.
- gerE: Proceso de transcripción de la célula.

El segundo grupo de 100 experimentos fue excluyendo al grupo spo, mas 3 genes seleccionados al azar. El tercer grupo de 100 experimentos fue exclusivamente del grupo spo. El cuarto grupo fue de 70 experimentos y se seleccionaron 300 genes de forma aleatoria. Los últimos 60 experimentos fueron considerando los 4423 genes. Todo esto se realizó en una computadora Core 2 duo a 2.2 GHz, con 2Gb en RAM y Windows xp sp3. El tiempo de ejecución del método propuesto fue de 1250 segundos aproximadamente.

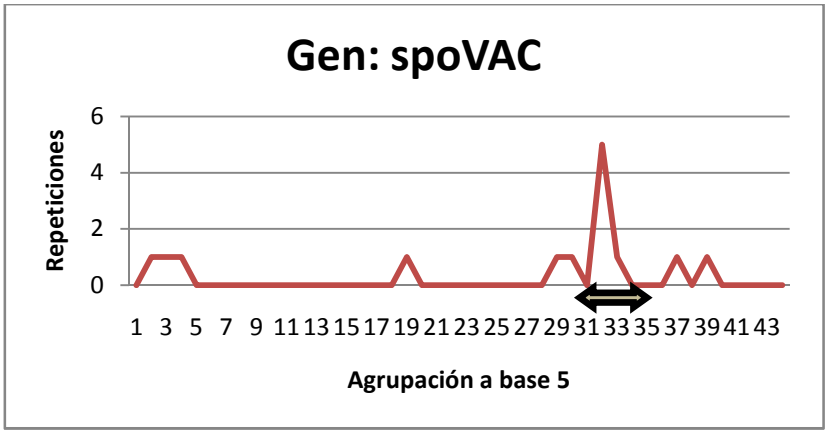
5.3 Análisis de resultados y gráficas.

En la gráfica 1 se puede notar una región muy conservada de los grupos que van del 3 al 9. Esa región es donde se podría esperar que se encuentre la secuencia reguladora del gen fur. Existen otros pequeños grupos pero no son dominantes.



Gráfica 1. Distribución de frecuencias de las cadenas propuestas para el gen fur.

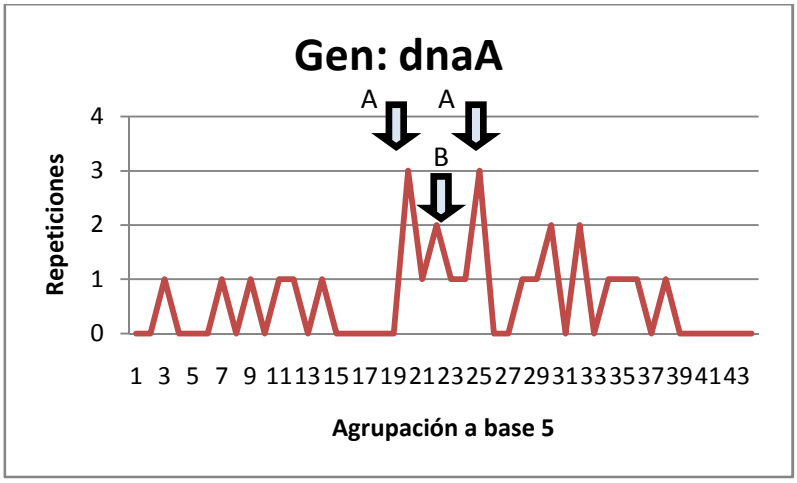
La gráfica 2 muestra una zona muy bien conservada y muy cercana al inicio del gen spoVAC. Esta característica hace que las regiones del 31 al 35 tengan una altísima probabilidad de ser secuencia reguladora, además, se encuentran muy cerca del sitio promedio donde se encuentra la transcripción.



Gráfica 2. El gen spoVAC muestra una gran región conservada en su gráfica de frecuencia.

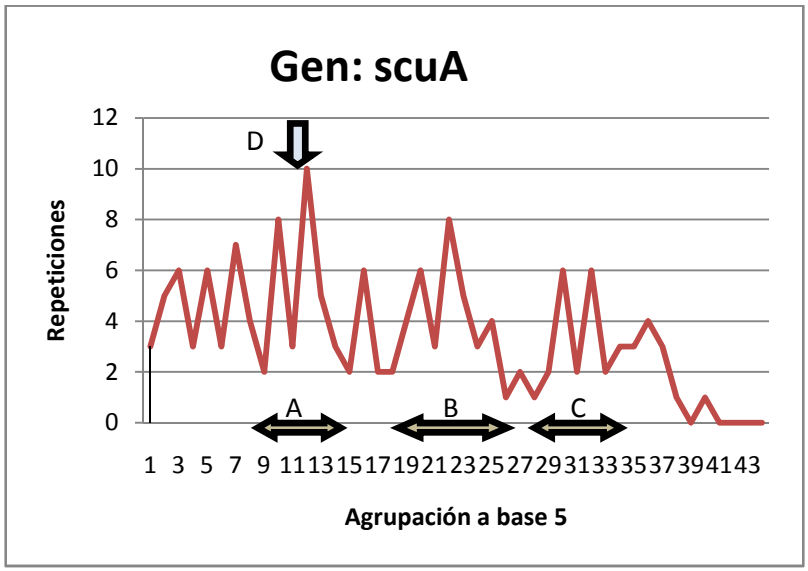
El resultado del gen dnaA es interesante (gráfica 3). Se tiene dos zonas altamente conservadas muy cercanas (A), pero divididas por una zona medianamente conservada (B). Lo que podría dar este comportamiento es lo que se podría llamar una huella genética. Dado que con el tiempo los seres vivos evolucionan y, por lo tanto,

cambian su material genético, uno de los picos (probablemente el del lado izquierdo ya que es el más alejado al sitio donde, generalmente, se hace la transcripción) es sólo una reminiscencia del pasado próximo del individuo.



Gráfica 3. Un caso curioso, el gen dnaA muestra dos sitios altamente conservados.

Otro gen que presenta una huella genética, al igual que el gen dnaA, es el scuA. Sin embargo, en éste se puede apreciar con mayor claridad en la gráfica 4 un descenso en el número de repeticiones (A, B y C). A diferencia de dnaA, en este caso es posible observar una región dominante muy bien conservada (D). Este es un ejemplo de gen cuyo posible inicio de transcripción no esté cercano al inicio del gen (El inicio del gen es del lado derecho, como se muestra en la figura 7 en el capítulo 4 en la sección "Estructura de la solución").



Gráfica 4. Otro caso curioso, aun más que el gen dnaA. Se puede apreciar con mayor claridad la huella genética y, también, es un caso donde es posible que la región reguladora esté alejada del inicio del gen.

Como se puede apreciar en las gráficas, el método propuesto es útil también para el descubrimiento de conocimiento. Aunque su propósito inicial es el encontrar las posibles secuencias reguladoras los resultados muestran que puede ser útil en el descubrimiento Serendipiti (encontrar algo sin buscarlo).

5.4 Comparativas con otros trabajos

Parte de las cadenas encontradas con nuestro método fueron comparadas con las encontradas por Mota [18], que también trabajó con la base de datos del *Bacillus subtilis*, siendo la misma de nuestros experimentos. Varias cadenas encontradas por Mota fueron halladas como sub-cadenas que entregó el Algoritmo Genético implementado. Esto es bueno ya que las cadenas que aporta tienen un mayor soporte para ser consideradas como secuencias reguladoras o, en otro caso no considerado en esta tesis, cadenas con valor biológico. La razón por la que Mota encuentra cadenas pequeñas es por el motor de búsqueda que funciona mediante comparación de grafos, siendo un método efectivo pero al aumentar el tamaño de las secuencias, el tiempo computacional que requiere aumenta. La ventaja principal del algoritmo propuesto en comparación con el de Mota [18] es el tamaño de las cadenas propuestas y complejidad dados los múltiples objetivos que evalúan las cadenas. En la tabla 6 puede apreciarse un pequeño ejemplo de estas ocurrencias.

Tabla 6. Secuencias encontradas por la solución planteada y las encontradas por Mota.

Secuencias encontradas por Mota	Secuencias encontradas por el algoritmo propuesto
AAAAA	AAGTTT**AAAAA**TCCTGATT
GAAAA	**GAAAA**
TTTTT	**TTTTTT**ATACAGATTGCTT
GGAAA	A**GGAAA**ATAATGGCATATC
TGATG	TTGTGAACG**TGATG**AGAA
GTGAA	CATTGTCC**GTGAA**G
GAATT	CTAAAAATAAT**GAATT**
GAAAC	AAA**GAAAC**AATT
TGAAA	**TGAAA**TAA
GTGAA	CATTGTCC**GTGAA**G
GAAAC	AAA**GAAAC**AATT
GAATA	TGTT**GAATA**AGAATAGTTT
GGATT	ATTTTA**GGATT**TTGTCAAC
GGATA	CTAT**GGATA**AGTCAAG
TGAAT	TGT**TGAAT**AAGAATAGTTTT

Como se puede apreciar en la tabla 6. El método propuesto encontró varias secuencias que el algoritmo de Mota [18] y, como la teoría indica, es importante que dichas cadenas sean de mayor tamaño. El tiempo de ejecución del algoritmo propuesto por Mota [18] aumentaría considerablemente con el tamaño de las secuencias utilizadas. Al tal grado que sería impráctico en situaciones que requieran de un mayor tamaño de secuencias. El tiempo de ejecución es algo importante a considerar, pero, en cuanto a los Algoritmos Genéticos se refieren ese tiempo puede variar según las necesidades. Esto se debe por el hecho que, en los Algoritmos Genéticos, el tiempo de ejecución depende de dos factores importantes; el número de generaciones y la población que maneje. Se puede hacer que un Algoritmo Genético sea absurdamente rápido o lento, pero eso al final sólo afectará la calidad del resultado. En nuestro caso, el tiempo de ejecución fue de 1250 segundos (aproximadamente 20 minutos). Con eso se aseguró una buena calidad en los resultados.

Capítulo VI
CONCLUSIONES

El uso de un algoritmo multi-objetivo para el descubrimiento de conocimiento es una realidad, especialmente dentro del área de la genética, ya que, los múltiples objetivos a optimizar proveen de una mayor flexibilidad y fortaleza al momento de la búsqueda. Aunado con el poder de los algoritmos genéticos y su flexibilidad para las búsquedas en grandes espacios, tenemos una herramienta poderosa y flexible para el trabajo.

En esta tesis se propuso un método para encontrar posibles secuencias reguladoras utilizando Algoritmos Genéticos utilizando una evaluación multi-objetivo para encontrar secuencias con diferentes características. El algoritmo propuesto es prometedor ya que muestra posibles regiones de transcripción, lo que se traduce en las secuencias reguladoras. Dado que no es posible encontrar con exactitud las secuencias reguladoras por la complejidad de la búsqueda, el poco entendimiento del ADN como un lenguaje controlador y programador del cuerpo y la limitante en el poder de cómputo hace que el poder sesgar hacia una región ayude a minimizar el tiempo y costo de la fase de experimentación. Dicha fase se usa para verificar que realmente sea una secuencia reguladora.

Otro aspecto a considerar es la relación que existe entre los genes que se analizan. Si se usa un grupo de genes que son similares es mayor la probabilidad que se encuentren regiones conservadas y que aumente la probabilidad de encontrar las secuencias reguladoras. Esto nos lleva a concluir que la comparación de todo el genoma sin orden alguno, sólo nos lleva a introducir ruido a los datos, lo cual merma la calidad de la búsqueda. Sin embargo, también hay que recalcar que la selección adecuada de genes es un proceso que conlleva un gran esfuerzo y donde es necesario un profundo conocimiento del individuo a analizar para su máximo aprovechamiento. Es ahí donde entra el seguir proponiendo y mejorando las técnicas y los algoritmos encargados de dicha tarea.

6.1 Trabajo a futuro

Una posibilidad es usar un modelo de búsqueda más corto que el propuesto en esta tesis, ya que eso simplifica la búsqueda y reduce el número de posibles combinaciones. El modelo de búsqueda propuesto se puede ver como tres segmentos de secuencias o cadenas (cadena inicial → secuencia reguladora → cadena final). Una mejora serpia usar un modelo simplificado de dos elementos; ya sea cadena inicial → secuencia reguladora o secuencia reguladora → cadena final. De acuerdo a cómo trabaja la mayoría de los procesos de regulación, el modelo que más éxito podría tener es el secuencia reguladora → cadena final, ya que la cadena inicial no siempre es clara o bien definida pero, la cadena final es donde se inicia la transcripción en la mayoría de los casos. Otro aspecto que podría ayudar a mejorar el desempeño es el introducir un algoritmo de agrupamiento que organice las secuencias encontradas con la finalidad de: ayudar a evaluar la población (si se utiliza dentro del algoritmo genético); para refinar la salida (si se coloca al final de éste), o una combinación de los dos para ambos propósitos y generar mejores resultados. Una propuesta de algoritmo de agrupación que trabajaría bien con los Algoritmo Genéticos sería optimización por colonia de hormigas (ACO) ya que, las optimizaciones por ACO son muy flexibles en cuanto a elementos a agrupar y son paralelizables (haciendo buena combinación con los Algoritmos Genéticos, ya que estos también son paralelizables).

REFERENCIAS

[1] Bailey, T. L. and Elkan, C.: “Fitting a Mixture Model by Expectation Maximization to Discover Motifs in Biopolymers”, Procs. Int’l Conf. Intelligent Systems for Molecular Biology, 2 pp. 28-36 (1994).

[2] Cédric, N. and Desmond, G. H.: "SAGA: sequence alignment by genetic algorithm", Nucleic Acids Research, 24 (8) pp. 1515–152 (1996).

[3] Coello Coello, C. A., Van Veldhuize, D. A. and Lamont, G. B.: “Evolutionary Algorithms for solving Multi-Objective Problems”, Kluwer Academic Publishers, New York, pp. 6762-6763 (2002).

[4] D’haeseleer, P.: “How does DNA sequence motif discovery work?” Nature Biotechnology 24, pp. 959-961 (2006).

[5] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.: "A fast and elitist multi-objective genetic algorithm: NSGA II", IEEE Trans. Evolutionary Computation 6, pp 182–197 (2002).

[6] Dongsheng, C., Yinglei, S. and Khaled, R.: “MDGA: motif discovery using a genetic algorithm”, Procs. GECCO’05, USA pp. 447–452 (2005).

[7] Ester, M. and Zhang, X. “A Top-Down Method for Mining Most Specific Frequent Patterns in Biological Sequence Data”, Procs. the Fourth SIAM International Conference on Data Mining (SDM’ 2004) pp. 90-101 (2004).

[8] Falcon, F. M. L., Jeffrey, J. P. T., Chen, R. M., Chen, S. N. and Shih, S. H.: “FMGA: finding motifs by genetic algorithm”, Procs. BIBE’04 Taiwan pp. 459–466 (2004)

[9] Goldberg, D. E.:”Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. (1989).

[10] Hongwei and Vojislav: “A simulated annealing algorithm for multiple sequence alignment with guaranteed accuracy”, Third International Conference on Natural Computation (ICNC 2007) IEEE pp 270-274 (2007).

[11] Hong-Wei, G. and Yan-Chun L.: “A Hidden Markov Model and Immune Particle Swarm Optimization-Based Algorithm for Multiple Sequence Alignment”, Springer-Verlag Berlin Heidelberg pp. 756 – 765 (2005).

[12] Ling, C., Lingjun, Z. and Chen, J.: “An Efficient Ant Colony Algorithm for Multiple Sequences Alignment”, Third International Conference on Natural Computation (ICNC 2007) IEEE pp 208-212 (2007).

[13] Liu, X., Brutlag, D. L. and Liu, J. S.: "bioprospector: discovering conserved dna motifs in upstream regulatory regions of co-expressed genes" Pacific symposium on Biocomputing 6, pp 127-138 (2001).

[14] Larry J. Eshelman.:”The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination”. Gregory J. E. Rawlins editor. Proceedings of the First Workshop on Foundations of Genetic Algorithms. Morgan Kaufmann. pp 265-283 (1991).

[15] Lones, M. A. and Tyrrell, A. M.: “Regulatory motif discovery using a population clustering evolutionary algorithm”, IEEE-ACM Transactions on Computational Biology and Bioinformatics, 4 (3) pp. 403-414 (2007).

[16] Maier, D.: “The complexity of some problems on subsequences and supersequences”, Journal of the ACM 25 pp. 322–336 (1978).

[17] Mehmet, K. “MOGAMOD: Multi-Objective Genetic Algorithm for Motif Discovery”, Expert Systems with Applications, 36(2) pp 1039-1947 (2009).

[18] Mota P.: “Descubrimiento de Conocimiento en el Genoma de la Bacteria *Bacillus subtilis*” Tesis Profesional, Benemérita Universidad de Puebla, (2005).

[19] Nguyen, H. D., Yoshihara, I., Yamamori, K. and Yasunaga, M.: “Aligning Multiple Protein Sequences by Parallel Hybrid Genetic Algorithm”, Genome Informatics, 13 pp. 123–132 (2002).

[20] Nuin, P., Wang, Z. and Tillier, E.: "The accuracy of several multiple sequence alignment programs for proteins”, BMC Bioinformatics, (7) pp 471-488 (2006).

[21] Pan, Y. and Phan, S. “Threshold for positional weight matrix”, Engineering Letters, 16 (4) pp. 498-504. (2008)

[22] Roth, F. P., Hughes, J. D., Estep, P. W. and Church, G. M.:"Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation". Nature Biotechnology 16, pp. 939-945 (1998).

[23] Stine, M., Dasgupta, D. and Mukatira, S.: “Motif Discovery in Upstream Sequences of Coordinately Expressed Genes”, The 2003 Congress on Evolutionary Computation pp. 1596-1603 (2003).

[24] The National Center for Biotechnology Information. WEBPAGE: http://www.ncbi.nlm.nih.gov/

[25] Thompson, W., Rouchka, E. C. and Lawrence, C.E. “Gibbs Recursive Sampler: Finding Transcription Factor Binding Sites,” Nucleic Acids Research, 31 (13) pp. 3580-3585 (2003).

[26] Ulas, B. B., Mehmet, K. "Top-Down Motif Discovery in Biological Sequence Datasets by Genetic Algorithm", International Conference on Hybrid Information Technology, 2 (6) pp. 103-107 (2006).

[27] Xundong, X., Shuanhu, W., Kin-Man, L. and Hong, Y. “An effective promoter detection method using the adaboost algorithm”, City University Hong Kong, pp 1-10 (2006)

[28] Zhou, W., Zhu, H., Liu, G., Huang, Y., Wang, Y., Han, D. and Zhou, C.: "A Novel Computational Based Method for Discovery of Sequence Motifs from Coexpressed Genes", International Journal of Information Technology, 11 (8) pp 75-83 (2005).

GLOSARIO

Estándar IUPAC: El estándar IUPAC es un estándar definido por la "*International Union of Pure and Applied Chemistry"* y la *"International Union of Biochemistry and Molecular Biology"* para especificar el conjunto de códigos usados para la ambigüedad en nucleótidos.

Locus: Significa locación. En biología se utiliza para marcar una posición de mucha importancia dentro de la cadena de ADN.

Matriz de pentamer: Es una matriz de pesos formada por cinco elementos.

Motivo: Un motivo es una secuencia (o segmento de secuencia) que se repite en varias partes del código genético.

Nucleótidos: Los nucleótidos son moléculas orgánicas que conforman el código genético. En el ADN son adenina, citosina, guanina y tiamina, en el ARN la tiamina es cambiada por el uracilo.

Oligonucleótidos: Es una secuencia corta de ADN o ARN, con cincuenta o menos pares de bases. Tienen distintas funciones: como cebadores en reacciones de amplificación, como sondas de hibridación y en bloqueos específicos de ARN mensajero.

Pares de base: Es la unión de dos nucleótidos que, dada la característica inconfundible de la doble hélice del ADN, son consideradas como un único carácter dentro de una secuencia de ADN decodificada.

Regiones promotoras: La región promotora es una porción del ADN situada al principio del gen y que, sin codificar ningún aminoácido, sirve para que las enzimas que realizan la transcripción reconozcan el principio del gen.

Sitios de transcripción: La transcripción del ADN es el primer proceso mediante el cual se transfiere la información contenida en la secuencia del ADN hacia la secuencia de proteína utilizando diversos ARN como intermediarios. El sitio de transcripción es el lugar donde el ARN empieza su tarea como intermediario.

ANEXOS

Anexo 1: consenso de genes bacterianos.

Consenso			
Clases	Inicio	Distancia	Fin
Clase 1	ttgaca	16-18 bases	tataat
Clase 2	cccttgaa	13-15 bases	cccgatnt
Clase 3	ctggna	6 bases	ttgca
Clase 4	ctaaa	15 bases	gccgataa
Clase 5	agganpupu	11-12 bases	gctgaatca
Clase 6	rggxttra	14 bases	gggtat
Clase 7	aaatc	15 bases	taxtgyttzta
Clase 8	taaa	15 bases	gccgatat
Clase 9	rwaggaxxt	14 bases	hgaat
Clase 10	tggcac	5 bases	ttgcannn
Clase 11	zhataxx	14 bases	catacaht
Clase 12	gcatr	15 bases	gghrarhtx
Clase 13	ghatr	18 bases	catxhta
Clase 14	ac	17 bases	catannnta

La sustitución de elementos diferentes a las cuatro bases se basa en la codificación IUPAC.