ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"APLICACION DE ALGORITMOS EVOLUTIVOS A LA BUSQUEDA DE MOTIVOS BIOLOGICOS EN BASES DE REGIONES PROMOTRAS DE ADN"

INFORME DE PROYECTO DE GRADUACIÓN

Previa a la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS MULTIMEDIA

Presentado por:

Carlos Josué Jordán Martínez

Guayaquil – Ecuador

2012

A G R A D E C I M I E N T O

A Jehová, *Amoroso Creador del Universo e Inteligente Diseñador de toda forma de vida presente,*

A Carlos Jordán, p*or su constante apoyo y guía en la realización de este trabajo*

Al Dr. Daniel Ochoa, *por la ayuda brindada en la revisión de este trabajo y sus significativas sugerencias.*

D E D I C A T O R I A

A Jehová, por permitirme conocer mediante este trabajo tan sólo un resquicio del diseño hermoso y perfectamente ordenado de las células y procesos que nos mantienen vivos. (Salmo 139:14)

Con mucho cariño, a mis padres, Carlos Jordán Villamar y Margarita Martínez de Jordán, por sus esfuerzos incansables en ayudarme a alcanzar la excelencia en toda labor que realice. Todo su arduo trabajo nunca habrá sido en vano.

A todos mis hermanos, que con su apoyo desinteresado y sincera amistad me han brindado las herramientas necesarias para seguir esforzándome por convertirme en una persona útil para la sociedad, y en especial, para Aquel que nos creó con un propósito.

T R I B U N A L D E S U S T E N T A C I Ó N

PRESIDENTE

Ing. Sara Ríos Orellana, M. Sc.

DIRECTOR DEL PROYECTO DE GRADUACIÓN

Ing. Carlos Jordán Villamar, M. Sc.

MIEMBRO PRINCIPAL

Dr. Daniel Ochoa Ch., Ph.D.

D E C L A R A C I Ó N E X P R E S A

“La responsabilidad del contenido de este Proyecto de Graduación me corresponde exclusivamente; y el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Carlos Josué Jordán Martínez

R E S U M E N

El presente trabajo explora dos métodos de búsqueda de motivos biológicos sobre bases de ADN. Los motivos biológicos son patrones de nucleótidos que se ubican en la zona de regulación de los genes, y que controlan la primera etapa del proceso de sintetización de las proteínas, fase conocida como transcripción genética. Las proteínas son moléculas esenciales para la vida, pues constituyen no solo parte estructural de las células, sino que participan también de los procesos de comunicación intracelular y entre células; basta mencionar unas pocas: colágeno, insulina, globulinas y una infinidad de hormonas. El identificar estos patrones permitiría a la industria farmacéutica y agrícola fabricar compuestos químicos orientados a la cura natural de múltiples enfermedades y plagas con base en el estímulo o supresión de las proteínas involucradas en la anomalía biológica. Este problema constituye para la ciencia y la tecnología un verdadero desafío, pues no se conoce a priori cual es el patrón que se busca, donde está ubicado en la zona de regulación y que longitud tiene; más aun, el patrón buscado muta de una instancia a otra. La solución de este problema se reduce a efectuar una búsqueda sobre espacios extraordinariamente grandes, lo que hace que este problema combinatorio sea considerado del tipo NP-Hard: su solución exacta requeriría tiempos de ejecución que están mas allá de lo razonable aún utilizando máquinas muy potentes, como supercomputadores, por ejemplo.

En este trabajo se presentan dos métodos de búsqueda de motivos con base en la computación evolutiva: el MBMAG (método de búsqueda de motivos basado en algoritmos genéticos) y el MBMEDA (método de búsqueda de motivos con base en algoritmos por estimación de distribuciones). Ambos métodos se probaron utilizando 6 bases reales de ADN: conjuntos de secuencias de nucleótidos donde de manera experimental se ha determinado exactamente la posición del patrón a buscar en cada secuencia. Para medir y comparar el rendimiento de estos métodos de búsqueda con los de otros métodos existentes en la literatura se utilizaron 2 métricas: ***Precisión*** y ***Exhaustividad*****,** tomadas del campo de la recuperación de información. Estas métricas miden cuan exacta y cuan completa es la búsqueda sobre los datos.

Los resultados obtenidos al aplicar estos dos métodos a las bases de datos reales anteriormente indicadas, dieron como mejores resultados los valores de 0.9 y 0.8 para la precisión y exhaustividad, respectivamente; lo que significa que de cada 10 patrones encontrados por los métodos evolutivos 9 fueron motivos reales que estaban presentes en los datos, y que de cada 10 motivos en los datos se encontraron 8. Esto muestra que los métodos evolutivos objeto de este trabajo resuelven de manera satisfactoria el problema de la búsqueda de motivos biológicos: con un desempeño similar y en algunos casos superior a los obtenidos por otros métodos estadísticos o combinatorios. Además, los tiempos de ejecución obtenidos son

razonablemente buenos al compararlos con los de otros métodos. Todos esto permite concluir que los métodos evolutivos aquí utilizados constituyen una alternativa factible para la solución aproximada de este problema.

Índice General

Índice de Figuras

Índice de Tablas

Introducción

El problema de la búsqueda de motivos biológicos trata de identificar un patrón que consiste de una secuencia de nucleótidos que regula la expresión genética al interior de las células. La expresión genética es un proceso por medio del cual la célula sintetiza proteínas a partir de la información contenida en los genes del genoma del organismo. La síntesis de las proteínas está controlada por un conjunto de otras proteínas conocidas como factores de transcripción (FT); estos factores se fijan en secuencias específicas de nucleótidos localizadas en las regiones promotoras de los genes cuya expresión genética regulan. Este acoplamiento promueve la transcripción de la información del gen en una molécula de ARN mensajera, la cual transporta esta información genética al ribosoma donde se forma una nueva proteína. Si fuese posible conocer aquellos sitios de fijación de los factores de trascripción que regulan la producción de una determinada proteína, entonces se podrían desarrollar sustancias que supriman o estimulen la producción de las proteínas responsables del funcionamiento correcto de un organismo, con lo cual diferentes enfermedades y problemas relacionados con plagas en la agricultura podrían ser resueltos, por ejemplo.

Sin embargo, identificar estos sitios de fijación es un problema difícil de resolver, puesto que se desconocen tres de sus características

importantes, a saber: su longitud, su ubicación exacta en la región promotora del gen regulado por dicho factor de transcripción, y la secuencia misma de nucleótidos que lo constituyen. Para complicar las cosas aún más, el patrón compuesto por los sitios de fijación presentan mutaciones en los nucleótidos de las que están compuestas las secuencias. Esto sugiere que identificar un motivo es equivalente a buscar a ciegas un patrón desconocido. Sin embargo, la clave para resolver este problema consiste en utilizar un conjunto de zonas promotoras de genes corregulados por el mismo factor de transcripción, pues ahora en esta base de ADN el patrón de nucleótidos donde se fija dicha proteína aparecerá repetido muchas veces, y es esta redundancia lo que hace posible identificar al patrón buscado.

Sin duda, existen varias soluciones a este problema en el campo de la biología molecular, por ejemplo aquella con base en el experimento de gel electroforésis [1]. Sin embargo, estas soluciones experimentales demandan en general mucho tiempo y son muy costosos, lo cual ha sido visto por la novel disciplina de la Bioinformática como una oportunidad para desarrollar soluciones computacionales a dicho problema. Dado que el patrón que se busca aparece en la base de ADN repetidas veces, el problema podría modelarse matemáticamente como uno de optimización combinatorial, donde se busca aquel patrón que se repite con mayor frecuencia en la base de ADN. El espacio de soluciones posibles para este problema es extraordinariamente grande; si se dan valores típicos a los datos, este

espacio sería del orden de 10^{45} soluciones, donde una búsqueda exhaustiva del motivo sería en la práctica una tarea fútil; de hecho, se desconoce que exista para este problema una solución exacta que tenga tiempos de ejecución polinómicos, por eso este un problema NP-hard. Razón que justifica de sobra la aplicación de métodos meta-heurísticos a la búsqueda de motivos biológicos; estos no garantizan obtener una solución correcta, pero cuando menos son de varios órdenes de magnitud más rápidos que los métodos exactos, y, además, procuran encontrar una solución aproximada a la exacta. Los métodos evolutivos, que pertenecen a esta clase de soluciones, han sido y son utilizados extensamente en la solución de problemas de optimización en casi todas las áreas de la ciencia y de la ingeniería.

Recientemente se han desarrollado algunos algoritmos de búsqueda de motivos con base en métodos estadísticos, filogenéticos y de inteligencia computacional, siendo los primeros aquellos que más han sido utilizados en la práctica. En este trabajo se describe el diseño y la implementación de métodos para realizar la búsqueda de motivos que aplican procedimientos de computación evolutiva, específicamente: los algoritmos genéticos y los algoritmos por estimación de distribuciones. Ciertamente existen ya soluciones a este problema con base en algoritmos genéticos, como GAME [11] y GALF [12] y en algoritmos por estimación de distribuciones, como EDAMD [24], sin embargo el propósito de este trabajo fue desarrollar

métodos para la identificación de motivos biológicos por medio de ensamblar elementos tomados de soluciones ya existentes y otros que son de factura propia, para: la representación de las poblaciones, de los individuos, las funciones de adaptación, operadores de variación, métricas, etc. Esto con el fin de integrar soluciones propias que resuelvan el problema de manera satisfactoria y que sean competitivas con otras soluciones ya existentes. Por otro lado, se espera que este proyecto sirva como un ejercicio realista que contribuya al desarrollo de la bioinformática en el Ecuador; de hecho, este es el primer trabajo de graduación en esta disciplina en la ESPOL, y probablemente también en el Ecuador.

Todos los métodos de búsqueda de motivos que se desarrollaron en este trabajo fueron probados con bases de ADN, tanto sintéticas, es decir: generadas de manera artificial insertando aleatoriamente en diferentes posiciones de la base el patrón que debe ser identificado, como también reales: construidas con las regiones promotoras de genes de uno o más organismos, donde el patrón de nucleótidos al que se fija el factor de transcripción ha sido identificado experimentalmente. Las bases reales utilizadas fueron tomadas de trabajos similares de otros autores, y actualmente son consideradas como estándares para probar nuevos métodos de búsqueda. Las bases sintéticas sirven para probar inicialmente los métodos evolutivos en desarrollo, a fin de corregir errores antes de realizar la búsqueda sobre bases reales de ADN.

Para medir el desempeño de los métodos evolutivos, se utilizaron las métricas *precisión* y *exhaustividad*, que fueron tomadas del área de recuperación de información. Estas métricas permiten determinar si un método de búsqueda encuentra todas las secuencias del motivo, y si todas las secuencias encontradas son correctas. Con base en los resultados obtenidos para estas métricas, fue posible comparar los métodos de búsqueda aquí desarrollados con otros publicados.

Concretamente, los métodos implementados en este trabajo son dos: MBMAG, un método que busca motivos utilizando en algoritmos genéticos, y MBMEDA que los busca aplicando algoritmos por estimación de distribuciones. Ambos métodos comparten entre sí varios componentes, como por ejemplo: los modos de representación de los individuos y de la población, la función de evaluación utilizada para estimar cuán buenas como soluciones son los individuos de una población. La diferencia principal entre estos métodos está en la forma como se genera una nueva población a partir de la presente; en el caso de los algoritmos genéticos se lo hace mediante los operadores de variación: cruce, mutación, etc., mientras que en los algoritmos por estimación de distribuciones se lo hace por medio de un modelo probabilístico construido a partir de las mejores soluciones de la población presente. El método que utiliza algoritmos genéticos depende de una combinación apropiada de las tasas de cruce y mutación para que la población evolucione según lo esperado, en tanto que el algoritmo por

estimación de distribuciones no depende de dichos parámetros, construye un modelo probabilístico de la distribución del contenido de información a partir de un conjunto de individuos seleccionados de la población para luego muestrear a partir del modelo los individuos de la siguiente generación.

En base a diferentes experimentos realizados sobre bases sintéticas y reales de ADN, vimos la necesidad de añadir operadores adicionales al proceso evolutivo que sirvieran de ayuda para evitar la convergencia de los métodos en óptimos locales o en falsos positivos.

Los resultados obtenidos al aplicar estos métodos evolutivos a la búsqueda de motivos biológicos demuestran que son altamente precisos y exhaustivos tanto con las bases sintéticas como con las bases reales de ADN. En especial, el método basado en los algoritmos por estimación de distribuciones tuvo una precisión promedio de 0.88 y exhaustividad promedio de 0.79. Estos resultados superan en más del 30% aquellos que se obtienen utilizando métodos estadísticos, y difieren de otros métodos evolutivos en tan sólo un 2%. Por lo tanto, los métodos evolutivos aquí desarrollados se presentan como soluciones competitivas para identificar nuevos motivos sobre bases reales de ADN.

Este documento está organizado de la siguiente manera: 6 capítulos y un apéndice. El capítulo 1 presenta una introducción a los conceptos básicos de la biología molecular que son necesarios para entender el problema de

búsqueda de motivos. Se pone especial énfasis en la explicación de lo que constituye un organismo, las moléculas fundamentales de la vida, el dogma central de la biología molecular, y una explicación del problema de la búsqueda de motivos desde el punto de vista biológico.

El capítulo 2 introduce los conceptos necesarios para entender algunos modelos matemáticos del problema de la búsqueda de motivos, así como también se presentan algunos conceptos requeridos para comprender los varios métodos de solución de este problema. La mayoría de las soluciones utilizadas para realizar la búsqueda de motivos aplican en mayor o menor grado estos conceptos, modelos y métodos matemáticos. También en este capítulo se presenta una clasificación ad hoc de dichos métodos.

En el capítulo 3 se introducen los conceptos de la computación evolutiva necesarios para entender las soluciones propuestas en este trabajo al problema de la búsqueda de motivos: representación de la población y de los individuos, función de evaluación, adaptación o fitness, mecanismos de selección, operadores de variación para los algoritmos genéticos y estimadores de modelos probabilísticos para los algoritmos basados en estimación de distribuciones. Además, en esta unidad se muestran los resultados de aplicar ambos métodos evolutivos a la solución de dos problemas de optimización sencillos; esto con el propósito de mostrar que las implementaciones de dichos métodos funcionan satisfactoriamente.

En el capítulo 4 se muestra la aplicación de estos métodos evolutivos a la búsqueda de motivos, planteada como un problema de optimización; se utilizarán los conceptos de los capítulos 2 y 3. Dependiendo de la manera como se integren los varios elementos de un método evolutivo, se obtienen varias soluciones al problema de la búsqueda de motivos; específicamente, en este capítulo se desarrollan tres métodos: dos con base en los algoritmos genéticos y uno basado en estimación de distribuciones.

En el capítulo 5 se muestra una introducción a las bases de regiones reguladoras de genes, también llamadas bases de ADN. Estas bases son muy importantes en el proceso de probar los métodos que se desarrollen para resolver el problema. En este capítulo se introducen las dos métricas antes mencionadas: precisión y exhaustividad, utilizadas extensamente para medir el desempeño de la mayoría de los métodos de análisis de la regulación genética.

En el capítulo 6 se muestran los resultados obtenidos al aplicar los métodos desarrollados en el capítulo 5 a las bases de ADN definidas en el capítulo 6. Los resultados consisten en los valores de las métricas definidas en el capítulo 5, los gráficos de convergencia de los métodos evolutivos y los logos de secuencia para los mejores resultados de cada método; estos logos son representaciones gráficas de los motivos encontrados. Finalmente, se comparan mediante tablas los resultados obtenidos por los métodos

desarrollados en este trabajo con aquellos obtenidos por otras soluciones tales como: los métodos MEME y BioProspector.

Finalmente, el apéndice describe el prototipo funcional de un buscador de motivos diseñado e implementado en este trabajo. Esta descripción contiene la presentación de los formularios que se utilizan para especificar la búsqueda, así como un diagrama de clases del diseño arquitectónico del prototipo.

CAPITULO 1

El problema de la búsqueda de motivos

Este capítulo es una introducción al entorno biológico del problema de la búsqueda de motivos reguladores en bases de ADN, cuya solución mediante métodos evolutivos es el objetivo de este trabajo. Los conceptos de biología molecular que aquí se introducen servirán para comprender mejor la importancia que tiene dicho problema.

El capítulo empieza presentando una descripción de la célula y clasificando los organismos vivos en función del tipo de célula de que están compuestos. Luego se presentan las moléculas fundamentales para la vida, las cuales permiten describir de manera concisa los mecanismos de replicación celular y de síntesis de las proteínas que las células necesitan para su correcto funcionamiento. Luego, con base en estos conceptos fundamentales se realizará el planteamiento biológico del problema de la

Identificación de los motivos reguladores de proteínas; así como también el reto que representa la solución de este problema.

1.1 La célula

Todos los organismos están compuestos en su nivel más simple por células. La célula es un conjunto de átomos y moléculas que constituyen la unidad más pequeña de vida. El tamaño de una célula depende del organismo al que pertenezca y la función que cumpla. El tamaño de una célula oscila entre los 0,2 a 300 μm (micrómetros) con una masa típica de 1 nanogramo. Las células son consideradas un sistema bioquímico vivo debido a que realizan las siguientes funciones que mantienen la vida:

Nutrición: las células toman sustancias del medio para transformarlas en compuestos necesarios para su subsistencia. Producto de esto, liberan energía y eliminan productos de desecho.

Crecimiento y multiplicación: las células dirigen de forma autónoma la síntesis celular, un proceso que consiste en regular sus ciclos de vida y mecanismos de multiplicación..

Diferenciación: muchas células sufren cambios de forma o función, lo que permite que un grupo de células cumplan funciones específicas en un organismo. Esta diferenciación ocurre generalmente en el momento de la concepción de un nuevo individuo.

Señalización: las células responden a estímulos químicos y biológicos tanto de medios externos como internos. Existen también procesos de comunicación intracelulares por medio de proteínas como las hormonas y neurotransmisores que permiten la interacción intracelular mediante procesos complejos de transducción de señales.

La célula presenta los siguientes componentes principales:

Material genético.- es el conjunto de información sobre el funcionamiento de la célula referentes a los procesos de nutrición, comunicación, reproducción y síntesis proteica. Esta información se encuentra codificada en forma de moléculas de ácido desoxirribonucleico (ADN) y moléculas de ácido ribonucleico (ARN).

Citoplasma.- es una sustancia gelatinosa dentro de la célula que mantiene el contenido de la célula dentro de las paredes exteriores de la célula

Organelas.- Las organelas son moléculas que cumplen funciones específicas dentro de la célula, similar a los órganos presentes en animales y humanos. Ejemplos de organelas son las mitocondrias y cloroplastos, los cuales cumplen el papel de plantas generadoras de energía en células de organismos vegetales y animales. Otro ejemplo de organelas son los Ribosomas, que sirven de fábrica de polipéptidos conocidos como proteínas.

Membrana celular.- La membrana celular sirve como un muro semi permeable que separa el interior de la célula del ambiente externo. Esta membrana también es conocida como pared celular. Está compuesta por una doble capa de lípidos y posee una variedad de proteínas que actúan como puertas de entrada y salida de elementos al interior de la célula. Junto con estas puertas biológicas, están presentes proteínas receptoras de señales que permiten establecer comunicación entre varias células y estimulan el trabajo en conjunto; estas proteínas son conocidas como hormonas.

Citoesqueleto.- El citoesqueleto consiste en un conjunto de filamentos y microtúbulos que mantiene una estructura interna que permite a la célula conservar su forma estable. El citoesqueleto es fundamental en procesos de división celular al evitar que la célula replicante se destruya al replicarse.

Las células cuentan con organelas externas que permiten su movilización. Este es el caso de las flagelas que consisten en un látigo a un costado de la célula, o pequeños filamentos al exterior de la membrana celular como son las fimbrias. Estas brindan a las células cierto grado de movilidad.

Las células necesitan de 4 tipos de moléculas para mantener su correcto funcionamiento. Estas moléculas se clasifican en función del propósito que cumplen en la célula. Las moléculas que cumplen un papel

funcional en los procesos de la célula son conocidas como aminoácidos y proteínas. Las moléculas que codifican y transportan la información de los procesos y funcionalidad de la célula son las moléculas de ADN y ARN. En particular, las moléculas de ADN son la estructura bajo la cual se codifica el material genético que sirve como planos del funcionamiento de cada una de las células.

1.1 Clasificación de los organismos

Todos los seres vivos se clasifican en organismos. Un organismo es un conjunto de células que forman una estructura material que se relaciona con el ambiente mediante un intercambio de materia y energía de tal modo que presenta la capacidad de desempeñar funciones básicas de la vida como la nutrición, la relación con otros organismos y la reproducción hasta concluir el ciclo vital con la muerte.

Los organismos se clasifican en base al tipo de célula por la que estén compuestos. A finales del 1970 se definieron 3 dominios bajo los cuales se basa la clasificación de los organismos: Bacteria, Archaea y Eukaryota.

Las bacterias son microorganismos unicelulares con tamaños entre 0,5 y 5 micrómetros. Se clasifican en función de su estructura externa: bacterias con forma esférica son conocidas como cocos, bacterias con forma de barras son conocidos como bacilos, y en forma de hélice son conocidas como espirilos. Las bacterias son procariotas, esto significa que no tienen

una membrana que contenga su material genético, también conocido como núcleo. Tampoco presentan organelas internas. Las bacterias son microorganismos móviles ya que cuentan con un flagelo como mecanismo de desplazamiento.

Las bacterias son los organismos más abundantes del planeta, su densidad en nuestro planeta es asombrosa. Se estima que por cada gramo de tierra hay 40 millones de bacterias; sólo en un mililitro de agua dulce existen no menos de un millón de bacterias. En conjunto, un total de $5x10^{30}$ bacterias pueblan la Tierra. Las bacterias son muy útiles en procesos de reciclaje de sustancias, y cumplen un papel fundamental en la fijación del nitrógeno de la atmósfera en la tierra. Si bien es cierto que las bacterias son responsables de muchas enfermedades infecciosas (cólera, sífilis, lepra, difteria, etc.), existen otras bacterias que cumplen un papel importante en la descomposición del alimento a lo largo del sistema digestivo.

Las Arqueas son organismos procariotas con características similares a las bacterias y al igual que estas, carecen de núcleo que contenga su material genético y de orgánulos. Sin embargo, la diferencia principal entre las bacterias y archaea estriba en las forma estructural que presentan, siendo características en el caso de las Archaeas las células planas y cuadradas. Junto con diferencias en su estructura externa, las archaeas poseen genes y enzimas implicadas en la transcripción y traducción genética que las acercan más a las células eucariotas que a las bacterias.

Las Arqueas fueron consideradas como parte del reino de las Bacterias hasta mediados de 1970 cuando el Dr Carl Woese de la universidad de Illinois descubrió diferencias en la estructura interna de organismos procariotas, lo cual dio lugar a la conformación de la división de organismos procariotas en bacterias y archaeas.

Las archaeas están presentes en diversos ambientes en la Tierra: desde aguas termales y lagos salados, pasando por océanos y humedales hasta en los intestinos de muchos rumiantes. A diferencia de otros organismos, utilizan una amplia gama de recursos como nutrientes desde azúcares hasta amoníaco, iones de metales e incluso hidrógeno. Las archaeas cumplen papeles importantes en los ciclos del carbono y nitrógeno. Las arqueas no presentan un papel parasitario sobre otros organismos, por el contrario son mutualistas con otros organismos vivos para alcanzar un fin en común. Hasta ahora, el conocimiento de las archaeas es limitado, por lo que no es posible estimar el número total de arqueas existentes.

El dominio de las Eukarya, o eucariotas está compuesto por organismos unicelulares y pluricelulares, en donde cada célula posee un núcleo celular que contiene la información genética y un conjunto de organelas que se encargan de realizar funciones específicas al interior de cada célula. Dentro del núcleo celular, el material genético se divide en varios bloques lineales llamados cromosomas. El núcleo celular tiene un

tamaño aproximado de 6 micrómetros, un 10% del volumen total de una célula eucariota.

El citoesqueleto es una estructura dinámica que mantiene la forma de la célula y cumple un papel regulador del tráfico intracelular. El citoesqueleto de los organismos eucariontes está compuesto de filamentos intermedios y microtúbulos, a diferencias del citoesqueleto de los organismos procariontes, el cual está compuesto por proteínas como FtsZ y MreB. Las células presentes en el dominio de las Eukarya se reproducen mediante procesos de mitosis y meiosis; los organismos procariontes sólo presentan una reproducción de tipo asexual como es el proceso de la mitosis.

Los eucariontes son organismos conformados por células eucariotas. Estos organismos más complejos tienen células eucariotas especializadas en ciertas funciones formando órganos, los mismos que cumplen funciones específicas dentro del organismo; ejemplo de los órganos son los pulmones, el hígado, el corazón. Las plantas, animales, y seres humanos son clasificados como organismos eucariontes.

1.3 Las moléculas fundamentales de la vida

Existen 4 tipos de moléculas fundamentales para mantener la vida y el correcto funcionamiento de todo tipo de célula de cualquier organismo. Estas moléculas son conocidas como las moléculas fundamentales de la vida, y son las moléculas de ADN y ARN, los aminoácidos y las proteínas.

La molécula de ADN

El ácido desoxirribonucleico es una macromolécula que contiene toda la información genética utilizada en el desarrollo y funcionamiento de todos los organismos vivos conocidos. En la naturaleza, el ADN no se encuentra como una molécula individual, sino como un par de moléculas formando una estructura helicoidal de cadenas de nucleótidos. Esta estructura estable del ADN presenta un ancho de 2 a 2,6 nm. , donde cada nucleótido tiene una longitud promedio de 1 nanómetro.

Una molécula de ADN consiste en la unión de dos hebras de nucleótidos. Un nucleótido está formado por un ácido fosfórico o fosfato, una desoxirribosa que es un monosacárido de cinco carbonos y una base nitrogenada. La base nitrogenada es una molécula que se deriva de los compuestos heterocíclicos aromáticos purina y primidina. En base al compuesto del cual se deriven, las bases nitrogenadas se dividen en:

Bases nitrogenadas purínicas: las bases nitrogenadas purínicas son la adenina (A) y la guanina (G). Ambas bases forman parte de las cadenas de ADN y ARN.

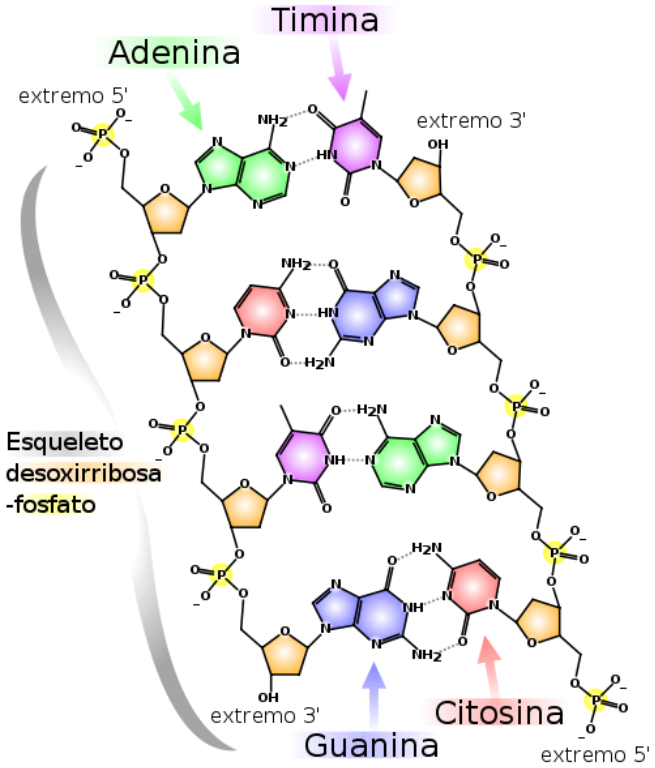
Bases nitrogenadas pirimidínicas: las bases nitrogenadas pirimidínicas son la timina (T), la citosina (C) y el uracilo (U). La timina y citosina están presentes en el ADN. En el ARN, solo aparecen la citosina y el uracilo.

Esta división de las bases conlleva la propiedad de complementariedad específica entre bases purínicas y pirimidínicas. Es por ello que la base A sólo se aparea de forma estable con la base C y la base G sólo con la base T. Esta propiedad es muy importante, pues en base a la información proveniente de una hebra de ADN es posible conocer la hebra complementaria.

Los nucleótidos se unen en una misma hebra mediante la molécula de fosfato, formando una cadena que sirve como columna vertebral de la hebra de nucleótidos. Una molécula de ADN está conformada por 2 hebras de nucleótidos unidas entre sí mediante enlaces de hidrógeno entre las bases de los nucleótidos; esto lleva a formar pares de bases, las cuales constituyen una medida de la longitud del ADN en un organismo. El genoma humano tiene alrededor de 3.000 millones de pares de bases.

Las dos hebras que conforman la molécula de ADN presentan 2 propiedades importantes en el comportamiento de la molécula de ADN: son anti-paralelas y complementarias. La propiedad de anti-paralelismo está relacionada con el direccionamiento de cada hebra en la molécula; el inicio y final de una hebra está acotado por un átomo de carbono conocido como 5’ y 3’. Las hebras de nucleótidos en la molécula de ADN se encuentran en direcciones contrarias, por eso se dice que son anti-paralelas. La propiedad de complementariedad está relacionada con la forma en que se unen las bases purínicas y pirimidínicas. Estas propiedades son importantes en el proceso de replicación exacta del ADN en el proceso de división celular. La siguiente figura muestra una sección de una molécula de ADN.

Figura 1.- Estructura de una molécula de ADN



Los enlaces entre nucleótidos de las dos hebras consisten en una fuerza de atracción electromagnética entre dos átomos de hidrógeno. Estos enlaces son relativamente débiles, se pueden romper las hebras y formarse nuevos enlaces de forma relativamente sencilla, lo cual permite una fácil replicación del ADN. Las interacciones entre los nucleótidos producto de los enlaces de hidrógeno son responsables de la forma de doble hélice que toma la unión de las hebras en la molécula de ADN.

El ADN se organiza dentro de cada célula en moléculas de cromatina ensambladas en cromosomas. Los cromosomas son almacenados en el núcleo celular en las eucariotas. En las células procariotas, el ADN se encuentra almacenado en una organela conocida como nucleoide.

La molécula de ADN presenta una densidad lineal de almacenamiento de información asombrosa. Un giro completo en la molécula de ADN tiene una longitud de 3.4 nm o 10 bp (pares de bases). Esto significa que la distancia entre un par de nucleótidos de una hebra es de 0.34 nm. Por lo tanto, si fuese posible almacenar un bit por nm, dado que la molécula de ADN consiste en un par de hebras de nucleótidos, la densidad de almacenamiento lineal en 10 bp de ADN sería de 6.8 bits/nm o $6 * 10^8$ bits/cm, lo que es aproximadamente 75 GB por cm de longitud en una molécula de ADN.

La molécula de ARN

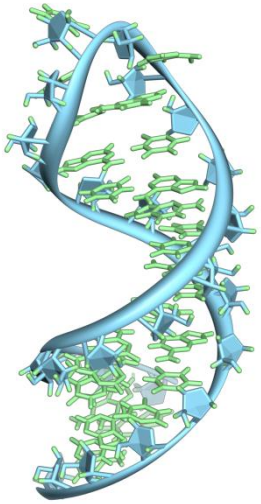
El ácido ribonucleico es un polímero de nucleótidos que cumple importantes funciones en la síntesis de proteínas en células procariotas y eucariotas. Algunos organismos, como los virus, poseen únicamente moléculas de ARN que contienen todo su material genético.

La molécula de ARN se presenta como una larga hebra de nucleótidos. Los nucleótidos están compuestos por una molécula de monosacárido de cinco carbonos llamada ribosa, un grupo fosfato y 2 pares

de bases nitrogenadas: 1 par de bases púricas (Adenina y Guanina) y 1 par de bases pirimidínicas (Citosina y Uracilo en sustitución de la Timina en la molécula de ADN).

La estructura básica de una molécula de ARN es la sola hebra de nucleótidos, como lo indica la siguiente figura:

Figura 2.- Estructura de una molécula de ARN



Sin embargo, existen ciertos tipos de ARN que presentan un apareamiento de bases (Adenina con Uracilo y Citosina con Guanina). Los enlaces de hidrógeno presentes entre los nucleótidos provocan el doblamiento de la molécula de ARN tomando estructuras denominadas secundarias y terciarias.

Las moléculas de ARN se clasifican en función del papel que jueguen en la síntesis proteica. Los ARN codificantes son moléculas de ARN que tienen codificado en su interior los nucleótidos que conforman la proteína. El

ARNm (ARN mensajero) es un ARN codificante generado a partir del proceso de transcripción genética.

El otro tipo de moléculas de ARN se conocen como no codificantes; las moléculas de ARN no codificante más conocidas son el ARNt (ARN de transferencia) y el ARNr (ARN ribosómico). Los ARNt son polímeros de unos 80 nucleótidos de longitud que sirven de puente entre las tripletas de nucleótidos en el ARN mensajero y la cadena de polipéptidos, añadiéndole el aminoácido en base a un código contenido en el ARNm al interior del ribosoma. El ARN ribosómico es un polímero que se une a la proteína para formar los ribosomas, representando las 2/3 partes de los mismos. Este tipo de ARN se encarga de crear los enlaces peptídicos entre los aminoácidos del polipéptido en formación durante el proceso de síntesis de proteínas. Este tipo de ARN es muy abundante en el citoplasma, llegando hasta representar el 80% de las moléculas de ARN al interior de la célula.

El material genético

El material genético al interior de una célula es un conjunto de instrucciones de las funciones que cumple la célula dentro de un organismo. Este material genético se encuentra codificado al interior de la célula en base a un alfabeto genético compuesto de 5 nucleótidos; estos nucleótidos se agrupan en moléculas de ADN. En las células eucariotas, este material genético se encuentra dentro de una organela conocida como núcleo celular,

el cual ocupa el 10% del volumen celular. En las procariotas, este material genético se encuentra repartido al interior del citoplasma.

La unidad más pequeña de material genético conocido toma el nombre de gen. Un gen contiene una porción de información genética que contiene la información necesaria para la síntesis de una molécula con una función específica; generalmente un gen codifica una proteína, pero existen casos en donde también codifican moléculas de ARN como el ARN mensajero y ARN ribosomático.

Los genes se agrupan en estructuras mayores conocidas como los cromosomas [2]. Los cromosomas tienen una forma de bastoncillos de 0,2 a 20 µm de longitud. En su estructura cuentan con la cromatina, una proteína que sirve como pegamento de los genes y que permite mantener una estructura estable dentro del núcleo celular en el caso de las células eucariotas. Esta estructura estable permite la replicación completa en una nueva célula, la cual tendrá el mismo número de cromosomas que la célula original.

Finalmente, el genoma de un organismo consiste en un conjunto definido de cromosomas contenido por todas las células que conforman el organismo. Por ejemplo, el genoma humano está compuesto de 46 cromosomas, dentro de los cuales, un par de cromosomas determinan el sexo del individuo. Estos cromosomas son conocidos en base a la forma que

presentan, parecida a las letras del alfabeto X y Y. Si se observa la presencia de dos cromosomas en forma de X, el individuo será mujer, si están presentea los cromosomas X y Y, será varón. Dado que sólo los hombres presentan el cromosoma Y, son éstos lo que determinan el sexo de la criatura.

Los aminoácidos.

Los aminoácidos son moléculas orgánicas compuestas por un carbono alfa unido a un grupo carboxilo (-COOH), un grupo amino ($-NH_2$), un átomo de hidrógeno y una cadena lateral denominada R que determina la identidad de los diferentes aminoácidos. La unión de varios aminoácidos da lugar a cadenas llamadas polipéptidos, los cuales se conocen como proteínas cuando la cadena polipéptida supera los 50 aminoácidos.

Existen muchas formas de clasificar a los aminoácidos. Una de las más comunes se basa en las propiedades de la cadena lateral R, como la siguiente:

- Neutros polares: Serina (Ser, S), Treonina (Thr, T), Asparagina (Asn, N), Glutamina (Gln, Q) y Tirosina (Tyr, Y).

- Neutros no polares: Glicina (Gly, G), Alanina (Ala, A), Valina (Val, V), Leucina (Leu, L), Isoleucina (Ile, I), Cisteína (Cys, C), Metionina (Met, M), Prolina (Pro, P), Fenilalanina (Phe, F) y Triptófano (Trp, W).

- Con carga negativa, o ácidos: Ácido aspártico (Asp, D) y Ácido glutámico (Glu, E).

- Con carga positiva, o básicos: Lisina (Lys, K), Arginina (Arg, R) e Histidina (His, H).

- Aromáticos: Fenilalanina (Phe, F), Tirosina (Tyr, Y) y Triptófano (Trp, W)

Otra forma de clasificación tiene que ver con la forma en que se obtienen estos aminoácidos para el organismo humano. Los aminoácidos que necesitan ser ingeridos por el ser humano ya que no son producidos al interior de las células son conocidos como aminoácidos esenciales para la vida; los aminoácidos que son generados a partir de la transcripción genética son clasificados como aminoácidos no esenciales.

Los aminoácidos esenciales son:

- Valina (Val)
- Leucina (Leu)
- Treonina (Thr)
- Lisina (Lys)
- Triptófano (Trp)
- Histidina (His)
- Fenilalanina (Phe)
- Isoleucina (Ile)
- Arginina (Arg)
- Metionina (Met)

Los aminoácidos no esenciales son:

- Alanina (Ala)
- Prolina (Pro)
- Glicina (Gly)
- Serina (Ser)
- Cisteína (Cys)
- Asparagina (Asn)
- Glutamina (Gln)
- Tirosina (Tyr)
- Ácido aspártico (Asp)
- Ácidoglutámico (Glu)

La siguiente tabla muestra 20 aminoácidos conocidos que se codifican en base a una tripleta de nucleótidos. Cada aminoácido está descrito por su abreviatura. En general, la nomenclatura básica de un aminoácido está compuesta por el nombre, un símbolo del alfabeto latino y una abreviatura de 3 letras. Por ejemplo, la Fenilalanina tiene la abreviatura **Phe** y como símbolo la letra **F**.

Tabla 1.- Código genético de los aminoácidos no esenciales

	U		C		A		G	
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys
	UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys
	UUA	Leu	UCA	Ser	UAA	Stop	UGA	Stop
	UUG	Leu	UCG	Ser	UAG	Stop	UGG	Trp
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg
	CUC	Leu	CCC	Pro	CAC	His	CGC	Arg
	CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg
	CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser
	AUC	Ile	ACC	Thr	AAC	Asn	AGC	Ser
	AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg
	AUG	Met	ACG	Thr	AAG	Lys	AGG	Arg
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly
	GUC	Val	GCC	Ala	GAC	Asp	GGC	Gly
	GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly
	GUG	Val	GCG	Ala	GAG	Glu	GGG	Gly

La combinación de 3 nucleótidos que codifican un aminoácido es conocido como un codón. Este codón se encuentra en el ARNm producto de la transcripción genética. Como se aprecia en la tabla, existen 64 codones para codificar 20 aminoácidos, lo cual lleva a que varios codones codifiquen un mismo aminoácido.

Las proteínas

Las proteínas son cadenas lineales de aminoácidos [6]. El número de aminoácidos que componen un polipéptido varía de una proteína a otra; por ejemplo, la insulina está conformada por 51 aminoácidos, mientras la timina tiene aproximadamente 28000 aminoácidos. Las proteínas constituyen cerca del 20% de la estructura celular, siendo después del agua su mayor componente. Se conocen más de 15.000 proteínas, con funciones tan diversas e importantes como:

1. **Enzimas.-** son catalizadores en procesos bioquímicos, como la conversión de glucosa en fosfato realizada por la enzima hexokinasa.
2. **Reguladores.-** controlan la temperatura al interior de la célula, el volumen celular, la conexión de la misma con nuevas moléculas y la generación de gradientes iónicos, necesarios para el funcionamiento de células nerviosas y musculares.
3. **Estructurales.-** sirven como bloques estructurales de componentes más complejos. Por ejemplo, el Colágeno, considerado como una molécula proteica, es el componente fundamental de la piel y los huesos formando las fibras colágenas.

Las proteínas tienen una estructura tridimensional compuesta por 4 estratos o niveles:

1. En primer nivel o estructura primaria, una proteína consiste de una secuencia lineal de aminoácidos.
2. En el segundo nivel o estructura secundaria, la proteína se dobla formando pliegues que son de dos tipos: alfa-hélices o beta-hojas.
3. Debido a las propiedades físico químicas de los aminoácidos se producen fuerzas entre los aminoácidos de una misma proteína conocidas como fuerzas electroestáticas que dan lugar a la estructura tridimensional de la proteína, estructura que presenta una energía potencial mínima. Esta es conocida como la estructura terciaria.
4. La unión de diversas estructuras proteínicas, formando proteínas más grandes y complejas como en el caso de la hemoglobina, da lugar a la estructura cuaternaria.

La estructura de las proteínas depende principalmente de las uniones de hidrógeno presentes entre las bases de los aminoácidos. Teniendo en cuenta que las proteínas poseen un tamaño 10.000 veces menor a una célula, todos los modelos generados por computadora de cualquier proteína son sólo suposiciones. El problema de estimar la estructura de una proteína a partir de una cadena de aminoácidos que la conforman es justamente un problema sin resolver en el campo de la bioinformática.

Las proteínas se clasifican según su composición química en: simples, si su hidrólisis sólo produce aminoácidos, como es el caso de la insulina y el colágeno; proteínas conjugadas o heteroproteínas, aquellas proteínas en

donde su hidrólisis produce aminoácidos y otras sustancias no proteicas. Las proteínas también se pueden clasificar según la forma estructural estimada. Este es el caso de las proteínas fibrosas, las cuales son cadenas de polipéptidos en forma de hileras que presentan la propiedad de ser insolubles en agua, como la queratina y el colágeno. También existen las proteínas globulares, que presentan una forma esférica apretada y compacta que son solubles en agua. La mayoría de enzimas, anticuerpos y hormonas son proteínas globulares. Finalmente, existen las proteínas mixtas que poseen una parte fibrilar en el centro de la proteína y una parte globular en los extremos.

Las proteínas son fundamentales en mantener con vida a toda clase de organismos en la naturaleza. Por ello es importante entender el proceso mediante el cual las proteínas son creadas al interior de las células; este procedimiento es conocido como la biosíntesis proteica. La biosíntesis proteica es el proceso biológico mediante el cual las células producen las proteínas en el ribosoma a partir de la información genética codificada en el ADN. Una explicación universalmente aceptada de cómo ocurre la biosíntesis proteica se encuentra en el dogma central de la biología molecular.

1.4 El dogma central de la biología molecular

El dogma central de la biología molecular es una hipótesis referente a la dirección en que fluye la información genética [1]. Este orden se describe en la siguiente figura:

Figura 3.- Hipótesis del flujo de información genética

El dogma general de biología molecular establece que la información genética en los genes que dan lugar a la producción de las proteínas no esenciales se transporta desde el ADN a las Proteínas a través de moléculas de ARN mediante los procesos de transcripción y traducción celular. El dogma general de la biología molecular explica a su vez el proceso de replicación del ADN, mediante el cual se realiza una copia exacta del ADN de una célula en los procesos de reproducción celular.

En la actualidad ha quedado demostrado que este flujo de información genética no siempre ocurre en la dirección definida en el dogma. En algunos organismos, como los virus, la información fluye de una molécula de ARN a tomar posesión de una molécula de ADN de otro organismo; este proceso se conoce como "retro transcripción". Además, se sabe que existen secuencias

de ADN que se transcriben a ARN y son funcionales como tales, sin llegar a traducirse nunca a proteína, como es el caso de moléculas de ARN no codificantes.

El dogma central de la biología molecular fue definido por el biólogo molecular británico Francis Crick en 1958; su planteamiento inicial tuvo lugar en un artículo de investigación en la revista Nature publicado en 1970[1].

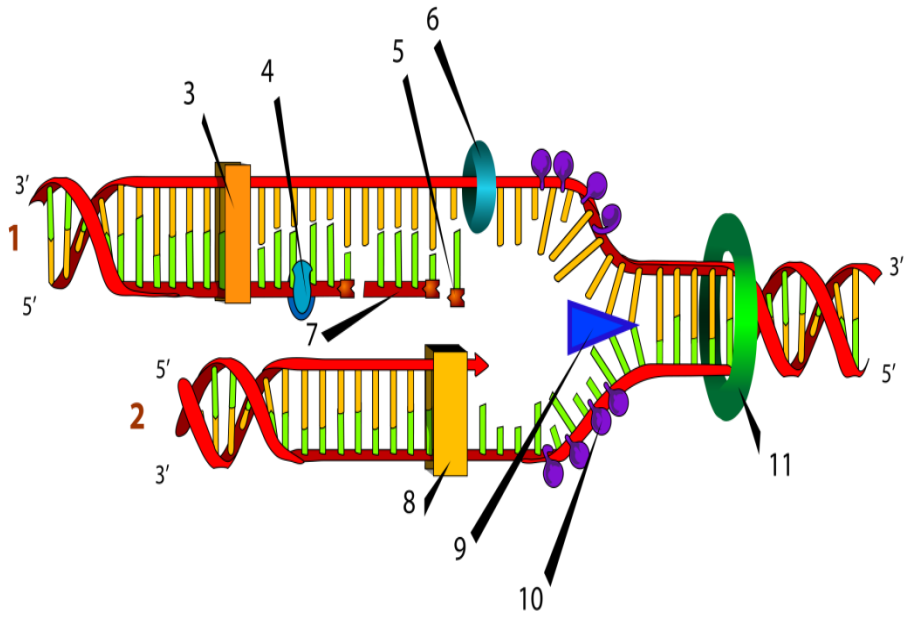
Los procesos biológicos más importantes que forman parte del dogma general de la biología molecular son 3: la replicación de ADN [3], la transcripción genética [5] y la traducción de la información genética en una proteína [8]. A continuación se explican cada uno de estos procesos.

Replicación de ADN

La replicación de ADN es un proceso de duplicación de una molécula original de ADN en dos moléculas de ADN idénticas a la molécula original. La replicación de una molécula de ADN parte del principio que toda molécula de ADN es semi conservativa, es decir, que dos hebras complementarias de una molécula ADN sirven de molde para la síntesis de una cadena complementaria de la cadena molde; de esta manera, cada nueva molécula nueva de ADN contiene una de las cadenas de la molécula de ADN original.

El proceso de replicación empieza en puntos de la molécula de ADN conocidos como orígenes de replicación. En estos puntos las proteínas iniciadoras reconocen secuencias de nucleótidos específicas en esos puntos y facilitan la fijación de otras proteínas que permiten la separación de las dos hebras de ADN formándose una horquilla de replicación. Un gran número de enzimas y proteínas intervienen en el mecanismo de replicación de moléculas de ADN. La siguiente figura muestra los componentes necesarios para el proceso de replicación.

Figura 4.- Proceso de replicación de una molécula de ADN



1	Hebra retrasada
2	Hebra adelantada
3 y 8	ADN polimerasa
4	ADN ligasa
5	Cebador
6	ADN primasa
7	Fragmento de Okazaki
9	Helicasa
10	Proteínas SSB
11	Topoisomerasa

A partir del punto de origen de replicación en la molécula de ADN, la proteína Helicasa inicia el rompimiento de los enlaces de hidrógeno entre ambas hebras de la molécula de ADN. Al mismo tiempo, la enzima Topoisomerasa mantiene unidos los enlaces de hidrógeno entre los nucleótidos en la molécula de ADN que no estén contemplados en la el proceso de replicación. Las proteínas SSB mantienen a las hebras de nucleótidos separadas, evitando la formación de los enlaces de hidrógeno entre las hebras sujetas a la replicación.

Las hebras separadas por la proteína Helicasa reciben los nombres de hebra adelantada y hebra atrasada. Una hebra adelantada es una hebra que va replicándose en la dirección 3´ → 5´. El proceso de replicación sobre la hebra adelantada es más sencillo ya que la construcción de la hebra complementaria sobre la hebra adelantada es realizada de forma continua, como indica la figura anterior.

Una hebra atrasada es la hebra que se replica en el orden 5´→3´ y necesita de fragmentos de RNA conocidos como Cebador los cuales sirven de puntos de unión de fracciones de ADN replicados. Estas fracciones de ADN replicados se conocen como fracciones de Okazaki. Esta hebra es conocida como hebra atrasada porque requiere un mayor número de enzimas adicionales con respecto a la hebra adelantada para realizar los procesos de replicación; requiere además un mayor tiempo en completar el proceso de replicación que la hebra adelantada.

Una vez realizado el proceso de división de las hebras adelantada y atrasada, se añaden en cada hebra una molécula de ADN polimerasa, la cual empieza a construir la hebra complementaria en cada hebra molde tomando nucleótidos del núcleo y añadiéndolos mediante enlaces de hidrógeno a los nucleótidos presentes en las hebras adelantada y atrasada. En este paso se empiezan a construir las dos moléculas de ADN clones de la molécula original.

La hebra complementaria construida sobre la hebra adelantada tiene la orientación 3´→5´. Debido a esto, la hebra complementaria a la hebra adelantada es construida de forma continua. La construcción de la hebra complementaria en la hebra atrasada tiene una orientación de construcción 5´→3´; provocando que los enlaces entre los nucleótidos entre la hebra creada por el ADN polimerasa y la hebra retrasada no sean continuos, por lo que deja cadenas de nucleótidos conocidas como los fragmentos de Okazaki. Los fragmentos de Okazaki se unen en moléculas de ARN conocidos como Cebadores. Los cebadores son moléculas de ADN que se encuentran sobre la hebra retrasada y determinan los puntos de uniones de los fragmentos de Okazaki. La enzima ADN ligasa se encarga de unir los fragmentos de Okazaki, eliminando los cebadores y formando la unión de cadenas de nucleótidos en la hebra complementaria a la hebra retrasada.

El proceso de replicación del ADN termina cuando unas moléculas de ADN polimerasa se encuentran con el punto de terminación. En ese momento, las moléculas de ADN polimerasa se separan de las hebras adelantada y atrasada, formando estas una réplica exacta de la molécula de ADN original.

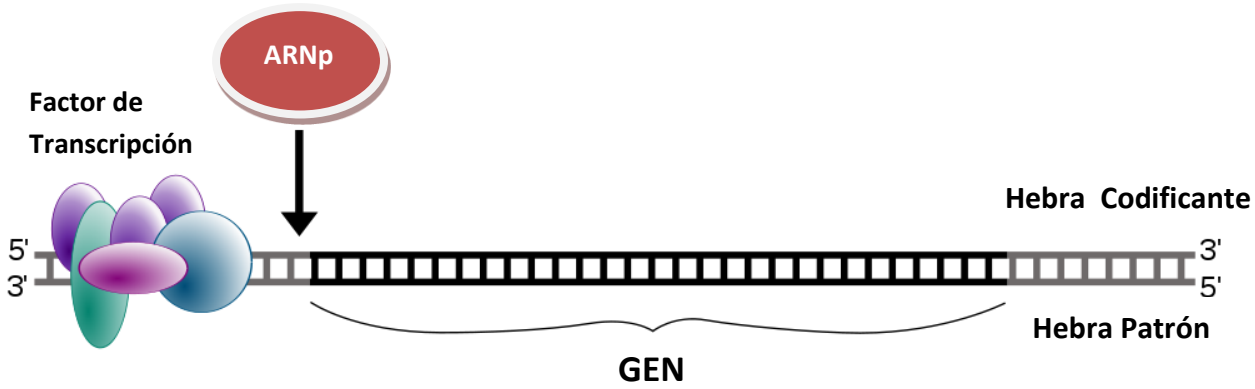
Transcripción Genética

La transcripción genética consiste en el proceso de transcribir la información genética codificada en el ADN en una molécula de ARN mensajero. El proceso de transcripción genética está compuesto de 3 pasos principales:

Iniciación: La iniciación de la transcripción genética comienza en una cadena de nucleótidos que se encuentra al inicio del gen conocida como región promotora. En esta región se fija una proteína conocida como factor de transcripción que atrae a una molécula de ARN polimerasa a posarse sobre la región en el gen que contiene la información necesaria para formar la proteína. El ARN polimerasa o ARNp rompe las uniones de hidrógeno entre los nucleótidos contenidos en el gen, provocando una abertura de las molécula helicoidal de ADN similar al efecto de abrir un cierre; ambas hebras de ADN no son separadas, solo las uniones de hidrógeno sufren ruptura. La hebra de ADN con direccionamiento 3´ → 5´ es considerado la hebra plantilla

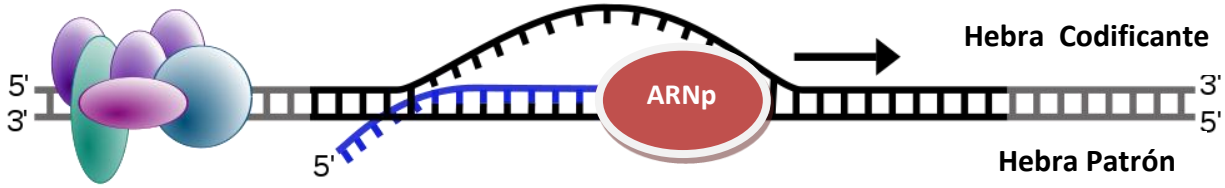
para construir la molécula de ARNm. La siguiente figura muestra el inicio del proceso de transcripción.

Figura 5.- Inicio del proceso de transcripción celular



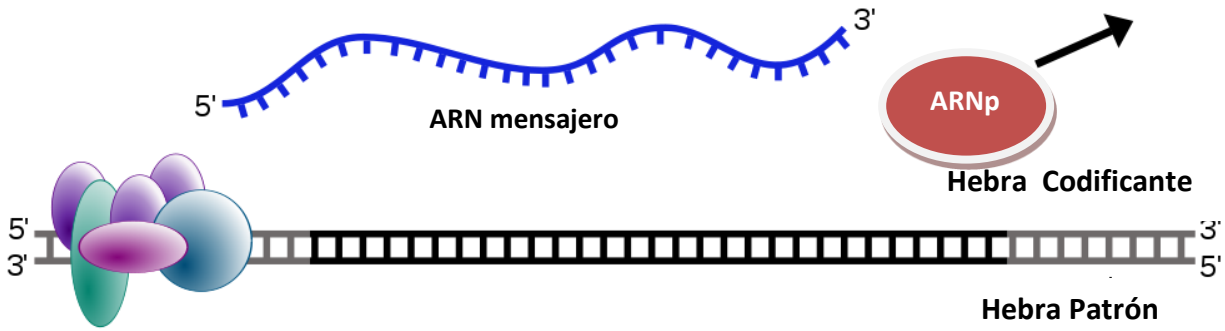
Elongación: La elongación consiste en la construcción del ARNm a partir de la hebra de ADN plantilla. Una molécula de ARNp se posa al inicio de la región de codificación proteica del gen y empieza a construir los nucleótidos complementarios de la hebra plantilla. Los nucleótidos construidos por el ARNp constituyen una transcripción de la información en la hebra plantilla en una molécula de ARN. En esta molécula de ARN los nucleótidos están unidos por un esqueleto formado de fosfatos con orientación 5´ → 3´; la base Timina es cambiada por la base Uracilo en la molécula de ARN mensajero. La siguiente figura muestra el proceso de formación del ARNm.

Figura 6.- Rompimiento de enlaces de hidrógeno por el ARN polimerasa



Terminación: la terminación del proceso de transcripción consiste en la unión de un nucleótido A al final de la molécula de ARN. En este momento el ARNp se desprende de la hebra plantilla, liberando la nueva molécula de ARNm. Estrictamente hablando, la molécula de ARN formada es una molécula ARN pre mensajera. A continuación ocurre el splicing, el cual es el proceso mediante el cual se transforma la molécula de ARN pre mensajera en el ARNm, al separar la información genética en el ARN pre mensajero en intrones y exones. Los intrones son nucleótidos de los que no se conoce un propósito en el proceso de síntesis proteica. Los exones, por otro lado, contienen la información necesaria para formar las proteínas en el ribosoma. Los exones son agrupados formando la molécula de ARNm, y los intrones son desechados. La siguiente figura muestra la molécula de ARNm terminada.

Figura 7.- Finalización del proceso de transcripción celular



Traducción Proteica

El proceso de traducción proteica consiste en decodificar la información en el ARN mensajero, en el ribosoma, para producir una cadena específica de aminoácidos, la cual a partir de las interacciones entre péptidos toma la forma de una proteína.

La traducción comienza en las células eucariotas a través del retículo endoplasmático, una organela al interior de la célula que se encarga de agrupar los exones en el ARNm en tripletas de bases llamadas codones. Estos codones codifican uno o varios aminoácidos, como se mostró en la tabla referente al código genético.

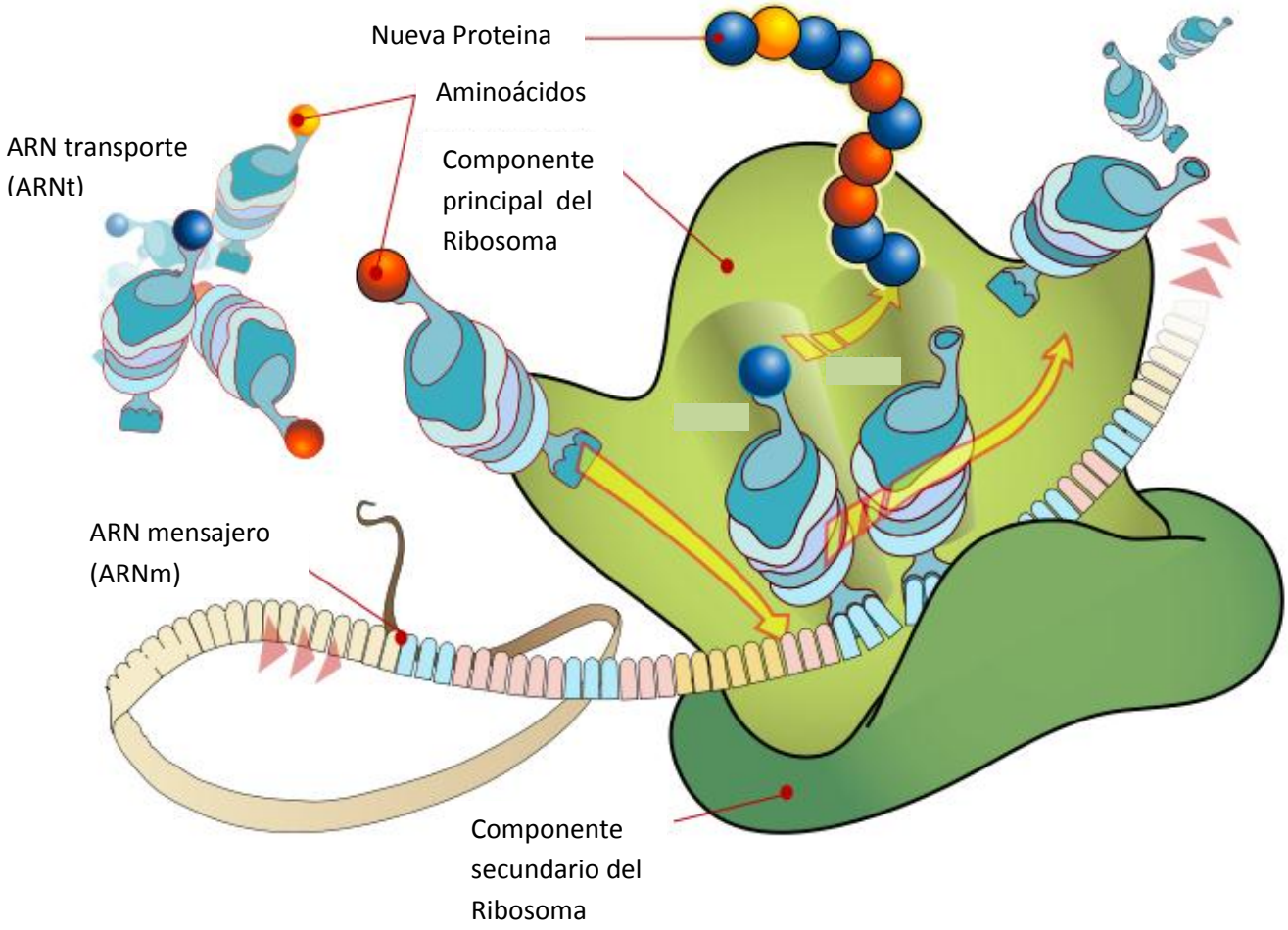
Una vez conformados los codones al interior del ARNm, esta molécula pasa al interior del ribosoma, donde se une a su anti codón correspondiente en una molécula de ARN de transferencia o ARNt. La molécula de ARNt presenta dos componentes fundamentales en la traducción proteica:

1. Una secuencia de 3 nucleótidos complementarios a los codones en el ARNm. Esta tripleta de nucleótidos es conocido como anti codón
2. Un aminoácido atado mediante un enlace a la molécula de ARNt. Este enlace se da en función del anti codón presente en el ARNt.

Dentro del ribosoma, los ARNt buscan el codón correspondiente en la molécula del ARNm. Una vez que se unen el codón y el anti codón el ARNt libera el aminoácido contenido, el cual pasa a formar parte de la cadena de polipéptidos; la cual pasará a formar la proteína codificada por el gen.

La figura 8 muestra el proceso mediante el cual se produce una nueva proteína a partir de la lectura de tripletas de nucleótidos en la molécula de ARN mensajero.

Figura 8.- Proceso de producción de proteínas a partir de la comunicación entre el ARN mensajero y el ARN de transporte

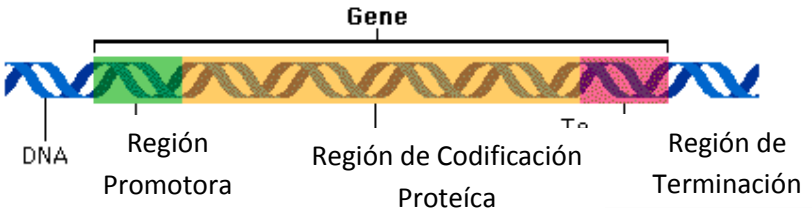


La lectura de codones en el ribosoma empieza en el codón de inicio (AUG); a partir de entonces inicia el ensamblaje de proteínas en base a la unión de un codón del ARN mensajero con el anticodón correspondiente en uno de los extremos de una molécula de ARNt. Esta unión libera al aminoácido de la molécula de ARNt el cual se une a la cadena de polipéptidos que conformará la nueva proteína. Este proceso se detiene cuando se llegue al codón de terminación, el cual está conformado por los nucleótidos UAG.

1.5 Planteamiento de problema de búsqueda de motivos en regiones promotoras

Como se mencionó anteriormente, un gen consiste en una secuencia de nucleótidos que contiene la información necesaria para la síntesis de una o varias proteínas. Un gen está compuesto por una región promotora, una región de codificación de una proteína y una región de terminación. La siguiente figura muestra los componentes de un gen.

Figura 9.- Partes que componen un gen



La región promotora está ubicada al inicio del gen, con una longitud mayor o igual a 100 bps. Dentro de esta región se encuentran secuencias de nucleótidos conocidas como secuencias reguladoras que cumplen funciones de control sobre el proceso de transcripción de los nucleótidos de la región de codificación proteica. La región de codificación proteica contiene parte de la información necesaria para la producción de una o varias proteínas. Esta información está codificada en nucleótidos. La región de terminación es una cadena de nucleótidos que determina el tramo final de un gen dentro de la molécula de ADN.

En la región promotora se fija una proteína llamada factor de transcripción en determinadas secuencias reguladoras. La unión del factor de transcripción con las secuencias reguladoras constituye un interruptor biológico que inhibe o estimula el proceso de síntesis proteica. Estos sitios de fijación tienen una longitud promedio de 30 pares de bases (o bps). Dado que el código genético que forma una proteína está distribuido en varios genes, el conjunto de sitios de fijación conforman una base de ADN, dentro de la cual se encuentran todas las secuencias de nucleótidos que sirven de sitios de regulación para el factor de transcripción. Este conjunto de sitios de fijación constituyen el motivo de la proteína; generalmente, el motivo se denomina en función del factor de transcripción asociado al mismo.

En base a los conceptos definidos anteriormente, el problema de búsqueda de motivos puede ser planteado de la siguiente manera:

Dada una base de ADN, encontrar el conjunto de secuencias reguladoras donde se fije un factor de transcripción en común que regula la producción de una proteína.

Para entender los desafíos que plantea este problema, tómese por ejemplo el factor de transcripción CRP de la bacteria Escherichia Coli, el cual regula la producción de una proteína para procesar la lactosa en esta bacteria. La base de regiones promotoras, o base de ADN contiene 18 regiones reguladoras provenientes de los genes corregulados por el mismo

factor de transcripción. La siguiente figura ilustra cómo están distribuidas las secuencias reguladoras en las regiones promotoras involucradas:

Figura 10.- Base de ADN de la bacteria Escherichia Coli

taatgtttgtgctggtTTTTGTGGCATCGGGCGAGAATagcgcgtggtgtgaaagactgtTTTTTTGATCGTTTTCACAAAAatggaagtccacagtcttgacag

gacaaaaacgcgtaacAAAAGTGTCTATAATCACGGCAgaaaagtccacattgaTTATTTGCACGGCGTCACACTTtgctatgccatagcatttttatccataag

acaaatcccaataacttaattattgggatttgttatatataactttataaattcctaaaattacacaaagttaatAACTGTGAGCATGGTCATATTTttatcaat

cacaaagcgaaagctatgctaaaacagtcaggatgctacagtaatacattgatgtactgcatGTATGCAAAGGACGTCACATTAccgtgcagtacagttgatagc

acggtgctacacttgtatgtagcgcatctttctttacggtcaatcagcaAGGTGTTAAATTGATCACGTTTtagaccatttttcgtcgtgaaactaaaaaaacc

agtgaaTTATTTGAACCAGATCGCATTAcagtgatgcaaacttgtaagtagatttccttAATTGTGATGTGTATCGAAGTGtgttgcggagtagatgttagaata

gcgcataaaaaacggctaaattcttgtgtaaacgattccacTAATTTATTCCATGTCACACTTttcgcatctttgttatgctatggttatttcataccataagcc

gctccggcggggttttttgttatctgcaattcagtacaAAACGTGATCAACCCCTCAATTttccctttgctgaaaaattttccattgtctcccctgtaaagctgt

aacgcaatTAATGTGAGTTAGCTCACTCATtaggcaccccaggctttacactttatgcttccggctcgtatgttgtgtggAATTGTGAGCGGATAACAATTTcac

acattaccgccaaTTCTGTAACAGAGATCACACAAagcgacggtggggcgtaggggcaaggaggatggaaagaggttgccgtataaagaaactagagtccgtttta

ggaggaggcgggaggatgagaacacggcTTCTGTGAACTAAACCGAGGTCatgtaaggaaTTTCGTGATGTTGCTTGCAAAAatcgtggcgattttatgtgcgca

gatcagcgtcgttttaggtgagttgttaataaagatttggAATTGTGACACAGTGCAAATTCagacacataaaaaaacgtcatcgcttgcattagaaaggtttct

gctgacaaaaaagattaaacatacctttatacaagacttttttttcatATGCCTGACGGAGTTCACACTTgtaagttttcaactacgttgtagactttacatcgcc

ttttttaaacattaaaattcttacgtaatttataatctttaaaaaaagcatttaatattgctccccgaacGATTGTGATTCGATTCACATTTaaacaatttcaga

cccatgagagtgaaatTGTTGTGATGTGGTTAACCCAAttagaattcgggattgacatgtcttaccaaaaggtagaacttatacgccatctcatccgatgcaagc

ctggcttaactatgcggcatcagagcagattgtactgagagtgcaccatatgCGGTGTGAAATACCGCACAGATgcgtaaggagaaaataccgcatcaggcgctc

CTGTGACGGAAGATCACTTCgcagaataaataaatcctggtgtccctgttgataccgggaagccctgggccaacttttggcgaAAATGAGACGTTGATCGGCACG

gatttttatactttaacttgttgatatttaaaggtatttaattgtaataacgatactctggaaagtattgaaagttaATTTGTGAGTGGTCGCACATATcctgtt

Las secuencias donde se fija la proteína CRP se encuentran resaltadas. Es interesante notar que en una región promotora puede haber más de un sitio de fijación para el mismo factor de transcripción. Nótese que entre 2 sitios de regulación, la diferencia entre los nucleótidos que la componen puede ser significativa. Tomemos por ejemplo los 2 sitios de regulación en la primera fila:

TTTTGTGGCATCGGGCGAGAAT

TTTTTTGATCGTTTTCACAAAAT

Ambas secuencias de nucleótidos tienen una longitud de 22 bps. A pesar que ambas secuencias cumplen la misma función, ambas tienen 12 nucleótidos diferentes en posiciones correspondientes. Y este caso se repite en los 24 sitios de regulación presentes en este ejemplo. No hay 2 secuencias que sean iguales y que estén ubicadas en las mismas posiciones en las regiones promotoras. Y dado que, el genoma de los organismos eucariontes contiene millones de genes, el trabajo de encontrar los sitios de regulación que permiten la creación de una proteína constituye un problema similar a buscar una aguja en un pajar, con la diferencia que en este problema no se conoce cómo es la aguja.

A pesar de ser un problema muy complejo, encontrar el motivo de una proteína tendría enormes beneficios para curar muchas dolencias de salud de la actualidad. La diabetes, por ejemplo, es el resultado de la falta de insulina, una proteína producida en el páncreas de los seres humanos. En la actualidad, no se conoce la razón por la que el organismo deja de producirla; de encontrarse los sitios de regulación para la producción de la insulina, sería posible estimular la producción propia del organismo de esta valiosa hormona.

Existen procedimientos en el campo de la biología que resuelven este problema, como el uso del gel electrophoresis [13] y DNA footprinting [12]. Sin embargo, estos métodos son muy laboriosos y consumen mucho tiempo y recursos. Por otra parte, métodos computacionales han demostrado una mayor eficiencia en encontrar motivos. Estos métodos serán explicados de forma más amplia en los capítulos posteriores.

CAPITULO 2

Planteamiento matemático del problema y soluciones existentes

Este capítulo tiene por propósito presentar los conceptos matemáticos necesarios para entender los métodos computacionales de búsqueda de motivos; la mayoría de los métodos existentes utilizan estos conceptos en mayor o menor grado. En particular, en esta sección se describen dos modelos matemáticos del problema, que se utilizaran para desarrollar los métodos evolutivos que son el objeto de este trabajo.

Además, en este capítulo el lector encontrará una clasificación ad hoc de algunos de los métodos existentes para resolver el problema de la búsqueda de motivos biológicos; junto con esta clasificación se presenta está una descripción breve de los métodos más populares, así como de sus debilidades y fortalezas.

2.1 Conceptos necesarios para construir un modelo matemático del problema de búsqueda de motivos

A fin de construir modelos del problema de la búsqueda de motivos, es necesario primero definir los siguientes conceptos que se utilizarán a lo largo y ancho de este trabajo.

Alfabeto.- conjunto finito de símbolos de interés. Un buen alfabeto permite representar correctamente las instancias del problema. Para la descripción de los modelos del problema utilizaremos un alfabeto de cuatro símbolos, donde cada uno representa un nucleótido. El alfabeto a utilizar es llamado el alfabeto genético compuesto por 4 símbolos {A, C, G, T}.

Palabra.- Una palabra es una secuencia de símbolos de un alfabeto. La longitud de una palabra se mide por el número de símbolos que tiene. Por ejemplo, las regiones promotoras son palabras de longitud ***n*** donde *n* toma valores típicos entre cientos y miles de bps. Un motivo está compuesto por un número finito de palabras de longitud *l*, donde *l* toma valores en el intervalo de 8 a 30 bps.

Texto.- Un texto representa un conjunto de palabras de interés. Para fines de este trabajo un texto consistirá típicamente de un conjunto de **t** palabras que representan las zonas de regulación de los genes corregulados por un mismo factor de transcripción. La base de ADN sobre la cual se

realiza la búsqueda de motivos es representada mediante un texto, similar al que se ilustra en la siguiente figura

CGGGGCTGGGTCGTCACATTCCCCTTTCGATA

TTTGAGGGTGCCCAATAACCAAAGCGGACAAA

GGGATGCCGTTTGACGACCTAAATCAACGGCC

AAGGCCAGGAGCGCCTTTGCTGGTTCTACCTG

AATTTTCTAAAAAGATTATAATGTCGGTCCTCAA

CTGCTGTACAACTGAGATCATGCTGCTTCAACA

Para el ejemplo de la ilustración, la base de ADN representada por el texto T está compuesto por 6 regiones promotoras con tamaño $l = 32$ bps.

Los conceptos presentados permiten hacer una definición más acorde con el ámbito sobre el cual se va a resolver el problema. Esto conlleva un modelamiento del problema de búsqueda de motivos, el cual estaría planteado de la siguiente manera:

Dado un texto T que representa la base de ADN, encontrar un patrón M que más se repita.

La noción básica detrás del patrón es la de un conjunto de palabras que conserven los mismos símbolos en cada ocurrencia. Sin embargo, en la práctica, las palabras que conforman un patrón o motivo no conservan variaciones bien definidas; mientras algunos símbolos en ciertas palabras se mantienen, otros presentan mutaciones.

Este es uno de los principales desafíos en la búsqueda de motivos: no se conoce a priori la posición de las palabras que conforman el motivo, tampoco se conoce la distribución de los símbolos en las palabras. En síntesis, no se conoce lo que se busca ni dónde se encuentra dentro de la base de ADN.

Este problema complejo puede ser resuelto tomando en cuenta el factor de redundancia del motivo. Si bien es cierto, el patrón de las palabras que conforman al motivo no es visible a simple vista, debe existir un factor común que permita a las palabras constituirse en un motivo. La mayoría de métodos de búsqueda de motivos utilizan diversas estrategias para descubrir estos patrones.

El profesor *Pavel A. Pevzer*, en su libro *Introduction to Bioinformatics,* realiza una analogía de solución al problema con el problema del Escarabajo Dorado, una novela escrita por Edgar Allan Poe. El problema del Escarabajo Dorado consiste en un texto cifrado, el cual contiene un mensaje escrito en un lenguaje conocido, como el inglés. La clave para descifrarlo está en tomar

en consideración un conjunto de símbolos que más se repiten; estos representan una palabra usada frecuentemente, lo que lleva a la postre a encontrar el tesoro escondido. Así como en la novela del Escarabajo Dorado, la redundancia estaba en la repetición de tripletas de símbolos, existen patrones que se repiten en las palabras que conforman el motivo; el reto está en encontrarlos.

Un concepto fundamental para medir el grado de similitud entre dos palabras es el concepto de distancia entre ellas. La distancia entre dos palabras *a* y *b* se define por el número de símbolos en posiciones similares que son diferentes. Si *a* y *b* no tienen símbolos diferentes, a y b son iguales y su distancia es 0. Mientras más cercano a 0 sea la distancia entre ellas, mayor similitud existirá entre a y b; mientras mayor sea la distancia entre a y b tendrán menos símbolos en común.

Existen muchas funciones para medir la distancia entre dos cadenas, en este trabajo nos centraremos en 2 de las más importantes: la distancia de Levenshtein y la distancia de Hamming.

Distancia de Levenshtein

La distancia de Levensthein entre dos palabras de igual longitud *a* y *b* se define como el mínimo número de ediciones necesarias para transformar a *a* en *b*; las operaciones de edición permitidas son inserción, eliminación y sustitución.

Tómese como ejemplo las palabras a = casa y b = calle. Para transformar *a* en *b* se necesitan 3 pasos:

1. casa -> cala (sustitución de ‘s’ por ‘l’)
2. cala -> calla (inserción de ´l´ entre ´l´ y ´a´)
3. calla -> calle (sustitución de ´a´ por ´e´)

Este método para medir la similitud entre 2 cadenas fue definido por Vladimir Levenshtein en 1965. En la actualidad es utilizado en los correctores de palabras de los procesadores de texto, estos comparan las similitud entre palabras escritas y las que estén almacenadas en un diccionario, sugiriendo correcciones en base a una mayor o menor similitud entre ellas. Existe una extensión de la distancia de Levenshtein para medir dos palabras de diferente longitud, la misma que es muy poco utilizada debido a que requiere un gran procesamiento, lo cual la convierte en un método poco práctico.

Distancia de Hamming

La distancia de Hamming entre dos palabras de igual longitud se define por el número de pares de símbolos correspondientes que sean distintos. Por ejemplo, la distancia entre las palabras

ACC**G**TGAT ACC**T**TGAT

es 1, pues sólo los símbolos en la cuarta posición son diferentes; esto indica que ambas palabras son muy similares, como de hecho lo son. Podemos hacer una extensión a este concepto cuando las palabras son de distinta longitud; la distancia entre palabras de longitudes diferentes, *u* y *v*, siendo l la longitud de u y m la longitud de m y l > m. La distancia entre *u* y *v* se define como el mínimo de la distancia de Hamming entre la palabra *v* y las palabras vistas a través de una ventana que recorre secuencialmente la palabra *u*.

Richard Haming introdujo este concepto de distancia en 1950 para medir los errores en la transmisión de códigos mediante medios electrónicos. Si la distancia entre las palabras código enviadas y recibidas es menor o igual a 1, eso significa que la comunicación entre el receptor y el transmisor es confiable.

Por ejemplo, sea *u* = ACCATTCG**ACCGTGAT**TGAT y l=20 y *v*=ACCGTGAT y m=8.

La ventana es de igual tamaño que *v*. Cuando la ventana de longitud 8 recorre secuencialmente en 13 (20-8+1) secuencias de tamaño 8, siendo la primera secuencia ACCATTCG, la segunda CCATTCGA y así sucesivamente. La siguiente tabla muestra la distancia de Hamming entre *v* y las sub cadenas resultado del desplazamiento secuencial de la ventana W sobre *u*:

Tabla 2- La distancia de Hamming entre 2 palabras de distinto tamaño

u	ACCATTCG**ACCGTGAT**TGAT
v	ACCGTGAT
Ventana W_u	$d_H(W_u, v)$
ACCATTCG	5
CCATTCGA	6
CATTCGAC	7
ATTCGACC	7
TTCGACCG	6
TCGACCGT	6
CGACCCGT	7
GACCGTGA	7
ACCGTGAT	0
CCGTGATT	6
CGTGATTG	7
GTGATTGA	7
TGATTGAT	4

Si la palabra *v* está contenida en *u*, la distancia de Hamming d_H (*u, v*) = 0.

Otra extensión al concepto de la distancia de Hamming es la definición de la distancia de una palabra *u* de longitud *l* y un texto T que consiste de *t*

palabras de longitud *n*. Esta distancia se define como la suma de las distancias entre u y cada palabra de texto T. Si en cada una de las palabras en T está contenido el patrón u, entonces la distancia de u a T, d (*u*, T) = 0.

Si T fuese la base de ADN y *u* el motivo que se busca, entonces la distancia entre T *u* es 0, lo que significa que para resolver el problema de la búsqueda de motivos es necesario buscar un patrón cuya distancia al texto que representa la base de ADN sea mínima.

Representaciones de los motivos

Otro concepto importante para entender las soluciones existentes al problema resulta la forma de estructurar el motivo en función de las palabras que lo conforman. Las palabras que conforman el patrón pueden ser representadas de forma explícita, como un texto T_p donde cada fila representa la palabra del patrón correspondiente a su región promotora en la base de ADN. Sin embargo, resulta más práctico representar al patrón como un vector de enteros [15], denotado por la variable *v*; cada elemento en v almacena la posición en donde inicia una palabra del motivo en una región reguladora en la base de ADN.

Por ejemplo, si la base de ADN estuviera compuesta de la siguiente manera:

CGGGGCTGGGTCGTCACAATCCCCTTTCGATA

TTTGAGGGTGCCCAATAACCAAAGCGGACAAA

GGGATGCCGTTTGACGACCTAAATCAACGGCC

AAGGCCAGGAGCGCGTTTGCTGGTTCTACCTG

AATTTTCTAAAAAGATTATAATGTCGGTCCTC

CTGCTGTACAACTGAGATCATGCTGCTTCAAC

Donde las palabras del motivo se denotan de color rojo, el correspondiente vector de posiciones iniciales *V* sería:

$$V = \begin{bmatrix} 14 \\ 3 \\ 5 \\ 14 \\ 0 \\ 1 \end{bmatrix}$$

El vector *v* define una matriz de palabras T_p de dimensión *t* x *l*, donde *t* es el número de filas presentes en la base de ADN y *l* la longitud de las palabras del patrón. A partir de esta matriz es posible hacer un análisis más detallado de la frecuencia de los símbolos de las palabras que conforman el motivo definidas por *V*. Este análisis permite descubrir un patrón en el conjunto de palabras representadas en T_p y para ello, se define una matriz de perfil a partir de T_p. La matriz de perfil M_f es una matriz de 4 x l donde cada fila representa la frecuencia con que se repite un símbolo k en cada columna de T_p, donde k pertenece al alfabeto del problema.

Tomando el ejemplo anterior, v define un texto de palabras T_p con su correspondiente matriz de perfil M_f:

$$T_p = \begin{bmatrix} C & A & C & A & A & T \\ G & A & G & G & G & T \\ G & C & C & G & T & T \\ G & T & T & T & G & C \\ A & A & T & T & T & T \\ T & G & C & T & G & T \end{bmatrix}$$

$$M_f = \begin{bmatrix} A\mid & 1 & 3 & 0 & 1 & 1 & 0 \\ C\mid & 1 & 1 & 3 & 0 & 0 & 1 \\ G\mid & 2 & 1 & 1 & 2 & 3 & 0 \\ T\mid & 1 & 1 & 2 & 3 & 2 & 5 \end{bmatrix}$$

La matriz de perfil aporta información referente al símbolo dominante en cada columna de T_p. La matriz de perfil M_f permite definir una palabra de consenso al motivo, denotada por P_C. La palabra de consenso es una palabra representativa del conjunto de palabras que conforman al motivo y se construye a partir de la matriz de perfil, cada letra de P_c es elegida del alfabeto y corresponde al símbolo de mayor frecuencia en cada columna de M_f.

Por ejemplo, para una matriz M_f definida de la siguiente manera:

$$M_f = \begin{bmatrix} A\mid & 1 & 3 & 0 & 1 & 1 & 0 \\ C\mid & 1 & 1 & 3 & 0 & 0 & 1 \\ G\mid & 2 & 1 & 1 & 2 & 3 & 0 \\ T\mid & 1 & 1 & 2 & 3 & 2 & 5 \end{bmatrix}$$

La palabra de consenso sería $\boldsymbol{P_c}$= G A C T G T

El score de la palabra de consenso mide la fortaleza de un patrón en base al vector de posiciones iniciales *v.* El score del consenso se calcula como la sumatoria del valor de frecuencia del símbolo dominante en cada columna. Dado el caso hipotético que todas las palabras del motivo fueran

iguales, es decir, tuvieran los mismos símbolos en las posiciones correspondientes, el score del consenso será igual a:

Score (***Pc***) = #longitud del motivo x #número de apariciones

Este sería la cota superior del valor que podría tomar el score de la palabra de consenso. Sin embargo, suponer que las palabras del motivo son iguales no es un caso práctico. Pero un conjunto de palabras, cuyo consenso tenga un valor de score muy cercano al máximo denota una presencia fuerte de un patrón que probablemente sea el motivo buscado.

2.2 Modelos matemáticos del problema de búsqueda de motivos

En base a los conceptos mostrados anteriormente, es posible definir 2 modelos al problema de búsqueda de motivos. Tómese por ejemplo la extensión a la distancia de Hamming de una palabra patrón p a una base de ADN representada por un texto T tal que la d_H (p, T) = 0. Esto implica que el problema de búsqueda de motivos se reduce a encontrar una palabra p de longitud *l* tal que la distancia entre p y T sea lo más cercana a 0. Por ende, el problema de búsqueda de motivos se ha reducido a un problema de optimización de una palabra con distancia mínima al texto T. En este caso, la palabra p constituye el motivo buscado.

Otro modelo del mismo problema depende de la representación elegida para el motivo; en vez de considerar al motivo como una palabra p de

longitud *l*, puede considerarse a la palabra de motivo como la palabra consenso de t palabras de longitud *l*, donde t es el número de filas del texto T. En este caso, el motivo buscado será la palabra de consenso p con Score máximo. Esto implica que el problema de búsqueda de motivos es un problema de optimización, donde se requiere hallar la palabra de consenso con score máximo. Para este modelo el motivo es la palabra de consenso p.

Ambos modelos plantean el problema de búsqueda de motivos como un problema de optimización del valor de una palabra de longitud *l*. Estos modelos matemáticos constituyen la base bajo la cual se construyen las soluciones existentes al problema de búsqueda de motivos.

2.3 Soluciones existentes al problema de búsqueda de motivos

Clasificación ad hoc de soluciones existentes al problema de búsqueda de motivos

A pesar de no existir un criterio unificado para clasificar los métodos de solución al problema de la búsqueda de motivos, Modan K Das [16] clasifica las soluciones en dos grupos: métodos basados en el análisis de cadenas de nucleótidos de la base de ADN y métodos basados en modelos probabilísticos de las secuencias de las regiones reguladoras.

Para propósitos de este trabajo, hemos clasificado los métodos de solución en 4 categorías:

- **Métodos de Búsqueda Exhaustiva**
- **Métodos Heurísticos y Meta Heurísticos**
- **Métodos Estadísticos**
- **Métodos Filogenéticos**

Métodos de búsqueda exhaustiva

En los métodos de búsqueda exhaustiva se explora cada alternativa en el espacio de búsqueda a fin de hallar la solución al problema. Estos algoritmos encuentran siempre la solución correcta al problema; sin embargo sus tiempos de ejecución son exponenciales, volviéndose poco prácticos, sin embargo, sirven como una introducción a métodos más complejos.

A continuación se presentarán 2 métodos de búsqueda Exhaustiva. El primero se basa en el concepto de distancia: se quiere hallar el patrón de longitud l de distancia mínima a la base ADN, mientras que en el segundo método se busca el patrón de consenso que tiene máximo valor de score.

El primer método descrito en [24] supone la siguiente condición inicial: el motivo buscado es único; la palabra motivo se repite una sola vez en cada secuencia del texto.

Este método calcula la distancia mínima de un patrón u de longitud *l* a la base de ADN, representada por un texto T. El conjunto de palabras de

longitud *l* constituye el espacio de búsqueda, denotado por U; este espacio de soluciones está formado por permutaciones con repetición *l* de 4. La estrategia consiste en calcular la distancia de cada palabra en U a T; aquella palabra que tenga distancia mínima será una solución del problema.

El número de elementos de U es 4^l, donde l es el número de símbolos del patrón que se busca. Por lo tanto, el tiempo de ejecución de este método estaría acotado superiormente por una función exponencial T que pertenece a O (4^l). Esto significa que un ligero aumento de símbolos en el patrón provoca un aumento considerable de los tiempos de ejecución, Por ejemplo, si l=20, que es un valor típico para los motivos, entonces el tiempo de búsqueda sería el necesario para realizar 4^{20} operaciones, o escrito de otra manera serían 10^{12} operaciones. Si se utilizara un procesador de 2.5 Ghz que realiza 2.5 *10^9 operaciones por segundo, el tiempo que le tomaría realizar 10^{12} operaciones serían 400 segundos. Si bien es cierto que los tiempos no son prohibitivos para encontrar todos los patrones posibles, el patrón del motivo presenta mutaciones, lo cual es una dificultad que no contempla este método

El método segundo, descrito en [25] busca una palabra de consenso de valor de score máximo. Cada elemento del espacio de búsqueda es un vector de posiciones iniciales, denotado por la variable *v*. El espacio de solución U está compuesto por permutaciones t de n-l+1, donde l es número de símbolos del motivo, n el número de símbolos de las palabras de la base

de ADN, y t el número de palabras en T, la base de ADN. Así, el tamaño de U es $(n-l+1)^t$.

La estrategia aquí consiste en elegir un vector de posiciones iniciales v. A partir de v se calcula la matriz de perfil correspondiente, y con ella la palabra de consenso y el score de correspondiente a v. Esta operación se realiza para cada vector v posible sobre la base de ADN hasta encontrar el patrón que produzca el score máximo.

El espacio de búsqueda definido por este método tiene una cardinalidad de $(n-l+1)^t$, siendo n el tamaño de la región promotora, l el tamaño de las instancias del motivo y t el número de regiones promotoras en la base de ADN. El espacio de búsqueda generado es mayor que el espacio del primer método. Para demostrar esta afirmación, considere el ejemplo anterior. Para un motivo de longitud l=20, el tamaño del espacio de búsqueda para el método 1 es 4^{20}; para el método 2 su espacio de búsqueda es de 10^{45}. Si se utilizara el mismo procesador del ejemplo anterior, encontrar todas las combinaciones posibles tomaría $4 * 10^{40}$ segundos Este ejemplo demuestra que la práctica, los métodos basados en la búsqueda exhaustiva son ineficientes.

Ambos métodos de búsqueda exhaustiva encuentran siempre el motivo en la base de ADN cuando el patrón está compuesto por palabras cuyos símbolos son iguales en posiciones correspondientes. En la práctica,

este caso no se ha encontrado, por lo que los métodos exhaustivos no son los métodos más eficientes para resolver este problema por el tiempo que les toma encontrar la solución al problema. Ambos métodos pueden mejorar significativamente sus tiempos de ejecución utilizando la estrategia de evaluar y podar, conocida como Branch & Bound.

Branch & Bound

El método Branch & Bound o Ramificación y Poda [23] es un método de búsqueda en profundidad que utiliza un árbol ordinario como estructura para organizar los elementos del espacio de soluciones. Esta estructura evita la necesidad de evaluar cada solución del espacio de soluciones permitiendo podar vértices cuyas hojas no tengan la solución correcta del problema, reduciendo significativamente el espacio de búsqueda y con ello disminuyendo el tiempo necesario requerido para encontrar la solución correcta.

Branch & Bound mejora de forma significativa los tiempos de ejecución de los métodos exhaustivos mencionados anteriormente. A continuación mostraremos su aplicación para mejorar la estrategia de búsqueda sobre el método 2 que tiene un espacio de búsqueda mayor.

Para ello se construye un árbol de búsqueda de profundidad *t*, el cual tiene un factor de ramificación uniforme de $n - l + 1$. Los vértices interiores representan una matriz de $k \times l$, siendo k el nivel de profundidad del vértice;

las hojas representan una matriz de *t* x *l* formado por las palabras correspondientes a las posiciones de un vector de posiciones iniciales.

Para evaluar cada vértice se define una función f, que es la suma del score del vértice sumado al score de la estimación de las hojas debajo del vértice. Esta estimación es optimista, es decir, busca la mejor solución presente en las hojas cuya raíz sea el vértice evaluado.

En el método dos, f sería igual a:

$$f(v) = \text{Score}(M_{k,l}, ADN) + \text{Score}(M_{t-k,l}, ADN)$$

Donde *v* es el vértice interior, *k* el nivel del vértice, *l* la longitud del motivo, *t* número de palabras de la base de ADN, $M_{k,l}$ representa un vector de *k* palabras de longitud *l*, y $M_{t-k,l}$ representa un vector de *t-k* palabras de longitud *l* .

Al inicio del método se define un valor base B=0. Este valor representa el valor de f asignado por default a la raíz del árbol de búsqueda. Dado que B&B es un método de búsqueda en profundidad, siempre evalúa los vértices de izquierda a derecha. La decisión de profundizar la búsqueda o podar el árbol debajo del vértice depende del valor de f del vértice comparado con el valor base presente hasta el momento. Si f(v) < B, quiere decir que ninguna de las hojas aportan con una mejor solución que la existente, por lo que se elimina el vértice junto con sus hojas. Si f (v) > B, debajo del vértice *v* existe una hoja debajo de v que representa una solución

mejor que la existente; esto lleva a que la búsqueda se profundice dentro del subárbol cuya raíz es el vértice *v*.

A pesar de que el uso de Branch & Bound para el método 2, para el peor de los casos sigue estando acotado por $O(n-l+1)^t$, el tiempo promedio de ejecución es polinómico por lo que es considerado más eficiente con respecto al método 1.

Sin embargo, el problema detrás de la estructura de la búsqueda que ofrece B&B trae una desventaja importante: el tiempo que requiera en construir el árbol de búsqueda. Suponga por ejemplo una base de ADN que tenga dimensión 100 x 18, en la cual se busca un patrón de longitud $l = 19$, su espacio de búsqueda tendría 80^{18} individuos; construir un árbol de este tamaño tomaría mucho tiempo. Por ello, en la práctica el estudio de estos métodos solamente sirve como introducción al desarrollo de métodos de búsqueda más eficientes.

Métodos Heurísticos

Los métodos heurísticos son métodos cuya estrategia de búsqueda evalúa un subespacio de soluciones en base a algún conocimiento a priori sobre el problema. La heurística puede definirse de forma arbitraria, puede depender del conocimiento de un experto en el área del problema, o en base a resultados en experimentos similares realizados anteriormente.

La heurística brinda a los métodos heurísticos un menor tiempo de ejecución con respecto a los métodos de búsqueda exhaustiva; sin embargo, los métodos heurísticos no encuentran siempre la solución correcta al problema, pues la heurística en que se basan podría ser incorrecta. Además, los métodos heurísticos toman decisiones basadas en el estado actual de la búsqueda, sin tomar en cuenta decisiones previas; esto quiere decir que, si el subespacio elegido es erróneo el resultado del método será con toda seguridad incorrecto.

No hemos encontrado métodos de búsqueda heurística para resolver el problema de búsqueda de motivos. Sin embargo, estos sirven como una introducción a los métodos metaheurísticos, bajo los cuales sí existen métodos de solución al problema de búsqueda de motivos.

Métodos Metaheurísticos

Los métodos metaheurísticos [27] son algoritmos de búsqueda estocástica. La búsqueda estocástica consiste en buscar las soluciones al problema de forma aleatoria sobre un espacio de búsqueda. El espacio de búsqueda contiene todas las soluciones existentes al problema.

Los problemas que resuelven mejor los métodos metaheurísticos presentan las siguientes características: no se conoce una heurística que guíe la búsqueda sobre el espacio de soluciones y no se conoce a priori la solución óptima al problema. Los métodos metaheurísticos requieren de una

función que evalúe cada solución en el espacio de búsqueda y le asigne un valor que represente cuán buena es para resolver el problema.

Una clasificación ad hoc de los métodos metaheurísticos está basada en el método de exploración escogido sobre el espacio de búsqueda. Esta clasificación presenta dos vertientes:

Métodos basadas en trayectorias.- son métodos que parten de una solución inicial e intentan reemplazarlo por otra solución de su vecindario con mejor calidad. Ejemplos de este tipo de algoritmos son: la búsqueda local, búsqueda tabú, recocido simulado.

Métodos basadas en poblaciones.- son métodos que encuentran la solución óptima en base a modificaciones sobre una población de soluciones. Ejemplos de estos algoritmos son: La computación evolutiva [10], Búsqueda Dispersa, Particle Swamp Optimization.

La búsqueda de motivos presenta las características de los problemas donde los métodos metaheurísticos presentan buenos resultados; no se conoce el patrón del motivo a priori, tampoco la posición de las secuencias del motivo pero sí existen funciones que permitan evaluar la presencia de un patrón dado un conjunto de secuencias, lo cual cumple el papel de función de evaluación.

La búsqueda de motivos utilizando métodos basados en trayectorias resulta poco práctico dado el tamaño del espacio de soluciones del problema; realizar una evaluación y búsqueda sobre las soluciones vecinas de una solución en el espacio da lugar a una convergencia prematura en óptimos locales, esto sin tomar en cuenta el tiempo que este tipo de métodos requerirían hasta encontrar la solución correcta del problema conlleva que no se hayan desarrollado hasta el momento métodos de búsqueda de motivos basados en trayectorias.

En este trabajo se desarrollan 2 soluciones al problema de búsqueda de motivos utilizando algoritmos metaheurísticos con una estrategia de búsqueda basada en poblaciones, como son los algoritmos evolutivos. En el siguiente capítulo se explicarán con mayor detalle.

Métodos estadísticos

Los métodos estadísticos de búsqueda de motivos se basan en la creación de un modelo probabilístico que representa la distribución de los nucleótidos en una solución candidata a ser el motivo correcto buscado. Este modelo representa a un individuo en base a una matriz de frecuencias, cuyos valores son modificados en base al cálculo de probabilidades condicionadas por la distribución de los nucleótidos en la base de ADN.

Los métodos estadísticos son en la actualidad los métodos mayormente utilizados en la identificación de motivos, debido tanto a los tiempos de ejecución como a los buenos resultados obtenidos con respecto a otros métodos de búsqueda existentes. Los métodos estadísticos más conocidos son el método de múltiple expectativa maximización (MEME) y el método de Gibbs Sampler. Para entender el método MEME es necesario hacer una breve introducción al método EM.

Método de Expectativa Maximización (EM)

Para entender el algoritmo EM aplicado a la regulación genética, es necesario definir previamente ciertos conceptos que son utilizados en este tipo de métodos de búsqueda de motivos.

Poff.- Es una matriz que representa las probabilidades de que las palabras del motivo comiencen en una posición (i, j) de la base de ADN. La matriz Poff tiene dimensión $l - n + 1 \times t$, donde t es el número de filas de la base de ADN y $n - l + 1$ las posiciones iniciales que pueden tener las palabras de longitud l en la base de ADN.

Freq.- es una matriz que representa un modelo del motivo buscado. Esta matriz tiene dimensión $\zeta \times j$, siendo ζ el número de símbolos del alfabeto y j el tamaño del motivo. En el caso del análisis de la regulación genética, $\zeta=4$. Las celdas de la matriz Freq contienen los nucleótidos que conforman el motivo estimado del problema.

Log (likelihood).- el log likelihood es una función que estima los parámetros de un modelo probabilístico en base a un conjunto de datos.

El método de Expectativa Maximización (EM) [17] es un algoritmo de aprendizaje no supervisado cuyo objetivo es maximizar la probabilidad de que una matriz modelo Freq sea el motivo buscado dado una base de ADN. EM fue desarrollado a finales de los 90 por Lawrence y Reilly como un método más sencillo de resolver el problema de la búsqueda de motivos en regiones promotoras. Para hallar el motivo, EM encuentra un modelo del mismo, representado por Freq. El modelo del motivo es encontrado por EM en base a estimaciones sucesivas de Poff y Freq. EM estima los valores de Poff utilizando Bayes a partir de los valores en la matriz Freq. Luego, los valores de la matriz Freq son estimados en base a los valores esperados en la matriz Poff. Este proceso continúa hasta que el valor de la función log likelihood del modelo Freq sea máximo, por lo que no sufra cambios significativos. La función Log (likelihood) que evalúa cada modelo del motivo tiene la siguiente fórmula:

$$\log(likelihood) = N \sum_{j=1}^{J} \sum_{l \in W} freq_{lj} \log\left(freq_{lj}\right) + N(W - J) \sum_{l \epsilon W} fout_l \log(fout_l)$$

Donde *N* representa el número de regiones promotoras en la base de ADN, *W* es la longitud de las secuencias en la base de ADN, *J* es la longitud

del motivo y $fout_l$ representa la frecuencia de la letra *l* fuera de las secuencias pertenecientes al motivo. El siguiente pseudocódigo muestra el funcionamiento de EM:

```
EM(base de ADN,J){
 Freq -> elegir de forma aleatoria
 do{
   estimar Poff a partir de Freq utilizando Bayes
   estimar Freq a partir de Poff
 } until(log likelihood(Freq) no varie)

 return Freq;
 }
```

El algoritmo EM presenta algunas desventajas .EM trabaja únicamente bajo el supuesto que sólo existe 1 instancia del motivo por región promotora en la base de ADN. Esto supone que en bases de ADN en cuya región promotora no se encuentre una instancia del motivo, el modelo del motivo se verá forzado a tomar información en forma de ruido lo cual hará que el modelo no represente al motivo de una manera confiable. Para elegir el primer modelo, EM puede basarse en una heurística como generarla de forma aleatoria. Esto puede provocar problemas, pues en la convergencia EM puede no llegar a un modelo apropiado del motivo, lo que requiere de sucesivas ejecuciones para alcanzar un buen resultado. Estos dos problemas son resueltos mediante la implementación de una mejora de EM, conocida como MEME.

MEME

MEME [19] fue desarrollado por Timothy Bailey a finales del 2009 en la universidad de Queensland, Australia. El algoritmo MEME, cuyas siglas significa Múltiple EM for Motif Elicitation es una versión modificada de EM que añade mejoras en las debilidades presentes en EM. El siguiente pseudocódigo muestra el funcionamiento de MEME

```
Dm //número de motivos presentes en la base de ADN
B // base de ADN
MEME(B,Dm){

   for 1 to Dm{
      for (cada posición inicial en B){
       Freq -> VPI;
       EM(B,Freq); //EM se ejecuta sólo 1 vez

      }
      Escoger Freq con mejor Likelihood
      EM(Freq con mejor Likelihood) //ejecutar hasta convergencia
      Eliminar Freq;
   }

}
```

MEME no genera de forma aleatoria los modelos del motivo Freq, los modelos son tomados de la base de ADN escogiendo cada vector de posiciones iniciales posible con palabras de longitud J en B y calculando la matriz Freq correspondiente. Luego, por cuestiones de tiempo ejecuta EM en base a Freq sólo 1 vez y calcula la función log likelihood del nuevo modelo. El algoritmo de EM que utiliza MEME es una modificación ligera del método mostrado anteriormente, teniendo como entradas la base de ADN y el primer modelo basado en un vector de posiciones iniciales VPI. Una vez terminado

este proceso, MEME elige al modelo con log likelihood máximo y ejecuta EM en base a $Freq_{max}$ hasta que converja. Una vez terminado este proceso, MEME “Elimina” el modelo de motivo encontrado para buscar otros motivos que puedan estar en la misma base de ADN.

MEME incorpora varias ventajas en el proceso de búsqueda de motivos en la base de ADN. MEME no trabaja sobre el modelo 1-S o una secuencia del motivo por región reguladora, pues busca sobre toda la base los modelos de motivo que presenten un likelihood máximo. Esto conlleva a que todas las secuencias de tamaño J son tomadas en cuenta y no sólo 1 por región reguladora. Además, la eliminación del modelo del motivo encontrado después de la convergencia de EM permite buscar si fuera el caso que otro motivo se encontrase en la misma base de ADN, lo cual es una posibilidad que otros métodos no contemplan. MEME presenta tiempos de ejecución en el orden de $O(n^2)$, donde n es el tamaño de las regiones promotoras en la base de ADN. Para regiones promotoras de tamaño superior a 100 bps el tiempo de ejecución de MEME resulta incómodo. Sin embargo, MEME es uno de los métodos más conocidos y utilizados en la búsqueda de motivos

Gibbs

Gibbs Sampler [7] es un algoritmo que estima un modelo probabilístico a partir de la probabilidad condicionada de dos o más variables aleatorias. Este algoritmo es utilizado en problemas en los que no se conoce de forma explícita la distribución conjunta de las variables aleatorias, pero sí existe la posibilidad de calcular la probabilidad condicional entre las variables. El algoritmo de Gibbs Sampler brinda un mecanismo para muestrear a partir de las distribuciones condicionales; el algoritmo Gibbs Sampler es un ejemplo de una clase de algoritmos conocidos como Markov Chain Monte Carlo.

Gibbs Sampler toma su nombre en honor al físico norteamericano J.W. Gibbs en base a sus investigaciones realizadas en el campo de física estadística (statistical physics); en particular, este científico estaba interesado en el desarrollo de métodos de muestreo a partir de una distribución desconocida, contando únicamente con los datos experimentales. Una definición general del algoritmo Gibbs Sampler fue hecha por los hermanos Stuart y Donald German en 1984[].

Lawrence et al. adaptaron el algoritmo de Gibbs Sampler a la búsqueda de motivos en una base de ADN. Este algoritmo presenta el siguiente funcionamiento:

Dada una base de ADN cuyas regiones promotoras son identificadas por las variables $s_1,s_2,\ldots s_n$, donde n es el número de regiones promotoras

que tiene la base de ADN; siendo w el tamaño de las instancias del motivo, el algoritmo de Gibbs Sampler realiza las siguientes operaciones de forma iterativa hasta alcanzar la convergencia en un conjunto de palabras de longitud w que constituyen el motivo buscado:

1. Escoge una secuencia s_i al azar
2. Construye un modelo de frecuencias que representa el motivo por la matriz $M_{4 \times w}$ a partir de las n-1 secuencias restantes.
3. Para cada posición j en $s_{i,}$ se escoge una de los l-w+1 palabras de tamaño w que comienzan en la j. Para cada una de estas palabras, se calcula el likelihood entre la cadena y el modelo M
4. Se escoge una cadena de la secuencia s_i que tenga el mayor likelihood con el modelo M.

La aplicación de Gibbs Sampler no trabaja bajo el esquema 1-S planteado anteriormente en el algoritmo EM, ya que Gibbs Sampler no descarta secuencias s_i de la base de ADN seleccionadas con anterioridad; esto permite que varias instancias del motivo en la misma secuencia s_i puedan ser seleccionadas. Sin embargo, Gibbs Sampler no asegura devolver el mismo resultado en todos los casos; esto depende de la secuencia s_i elegida al azar al inicio del experimento. Por ello, es recomendable ejecutar este algoritmo en varias ocasiones hasta encontrar un resultado en común. Esto afecta también el tiempo de ejecución de este algoritmo, el cual para regiones promotoras muy grandes toma tiempos de orden exponencial.

Métodos filogenéticos

La filogenética molecular es el estudio de las similitudes y diferencias entre secuencias de ADN de diferentes especies para determinar patrones evolutivos, con el objetivo de construir un árbol evolutivo para todas las especies de organismos vivos. Un árbol evolutivo o filogenético es un diagrama en forma de árbol que muestra las relaciones evolutivas entre especies biológicas, basándose en las diferencias y similitudes físicas y genéticas.

Las soluciones que hemos considerado anteriormente están orientadas a resolver el problema buscando los motivos en genes de una región promotora en común. El modelo filogenético en cambio, se fundamenta en la premisa de que existe un antepasado biológico común entre varias especies.

Los métodos filogenéticos buscan los motivos comunes en genes ortólogos basándose en un árbol filogenético. Los genes ortólogos son secuencias homólogas que se encuentra separadas por la especiación, es decir, genes que se encuentran en diferentes especies y que son altamente similares debido a que se han originado de un ancestro común. Teniendo un conjunto de regiones promotoras con genes muy similares se aumenta la redundancia del motivo lo que facilita su búsqueda.

El problema con este enfoque está en la forma de elegir qué especies están relacionadas genéticamente entre sí. La biología evolutiva no ha presentado hasta la fecha un árbol filogenético común a todas las especies, la elección de las especies relacionadas depende del árbol evolutivo que los investigadores tomen como correcto, criterio que puede variar y conduce a la larga a problemas en encontrar resultados correctos.

La mecánica detrás de los métodos filogenéticos consiste en un alineamiento global múltiple de las regiones promotoras de genes ortólogos, para luego identificar las regiones con un mayor grado de conservación con herramientas como CLUSTAL W [49]. Estas regiones conservadas constituyen los motivos buscados.

Este esquema aparentemente sencillo presenta algunos inconvenientes con respecto a la elección de las especies relacionadas. Si las especies están muy relacionadas, los genes presentarán muchas similitudes, lo cual no aportará ninguna información que facilite su búsqueda. Por el contrario, si la relación entre especies resulta muy lejana, alinear secuencias con ningún patrón en común resulta tedioso e inútil. Cuando la alineación múltiple falla, muchos métodos filogenéticos utilizan métodos como MEME y Gibbs para mejorar los resultados.

Wang y Stormo desarrollaron un algoritmo llamado PHYLONET [47] que presenta un alto grado de efectividad en determinar la presencia de motivos

en secuencias promotoras. PHYLONET realiza un alineamiento local de secuencias promotoras en genes ortólogos buscando patrones que permitan crear perfiles filogenéticos. Un perfil filogenético es un texto de cadenas alineadas de las regiones promotoras. Una vez definidos los perfiles, estos se comparan con otros perfiles de secuencias que representan motivos ya identificados. Una vez identificado un patrón, se realiza una búsqueda local utilizando un algoritmo como BLAST [48] para encontrar las posiciones dentro de los genes ortólogos donde se encuentran las secuencias del motivo.

Este método de búsqueda de motivos ha demostrado ser efectiva; tómese como ejemplo la búsqueda de motivos en la bacteria *Saccharomyces cerevisiae*, en la cual se encontraron el 90 % de los motivos existentes en 3315 regiones promotoras. Los autores afirman poder aplicar esta solución a la búsqueda de motivos en genomas mucho más grandes como el del ser humano.

CAPITULO 3

Introducción a la computación Evolutiva

El propósito de este capítulo es presentar los conceptos que subyacen a esta rama de la inteligencia computacional conocida como la computación evolutiva. Se presentarán aquí tres de sus paradigmas más importantes y conocidos: las estrategias evolutivas, los algoritmos genéticos y los algoritmos por estimación de distribuciones. En cada caso se explicarán los conceptos que permiten entender su funcionamiento, los operadores comúnmente utilizados y reseñará un poco de la historia tras su desarrollo. Finalmente, en este capítulo se mostrarán los resultados de la aplicación de los algoritmos genéticos y de los algoritmos por estimación de distribuciones a la solución de dos problemas de optimización; se escogieron estos métodos porque son aquellos que se utilizarán para resolver el problema de la búsqueda de motivos biológicos.

3.1 Introducción a la computación evolutiva

La Inteligencia Computacional

La inteligencia computacional es una rama de la inteligencia artificial centrada en el estudio y diseño de sistemas que demuestren un comportamiento inteligente en la resolución de problemas. Definir lo que se considera un comportamiento inteligente es una tarea compleja; sin embargo, en líneas generales, un comportamiento inteligente debe mostrar estas características:

- Flexibilidad a los cambios de entornos y fines
- Aprendizaje a partir de la experiencia
- Toma de decisiones en base a las limitaciones del problema

Los algoritmos basados en inteligencia computacional se clasifican en 3 vertientes: la computación evolutiva, las redes neuronales y los sistemas de lógica difusa.

La Computación Evolutiva

La computación evolutiva [20] es un término que describe un conjunto de algoritmos que encuentran la solución a un problema a partir del proceso de adaptación de una población de soluciones utilizando mecanismos de selección y exploración sobre el espacio de búsqueda.

Dicho de otra manera, los algoritmos evolutivos son métodos de búsqueda en paralelo sobre un espacio de soluciones del problema. Esta mecánica de búsqueda reduce el tiempo de ejecución frente a otros métodos de búsqueda, pues evalúa de forma simultánea grandes grupos de soluciones. Adicionalmente, los algoritmos evolutivos presentan métodos que intentan evitar la convergencia prematura del algoritmo en soluciones óptimas a nivel local; estos métodos no están presentes en otros algoritmos de búsqueda.

La computación evolutiva fue desarrollada a principios de 1960 por Nils Aall Barricelli cuyo propósito fue simular la teoría de la evolución de las especies biológicas mediante el desarrollo de vida artificial, cuyo comportamiento estaría regido por un algoritmo evolutivo. No fue sino hasta 1970 que los algoritmos evolutivos tomaron importancia en la solución de problemas en el campo de la ingeniería; el científico alemán Ingo Rechenberg inventó las estrategias evolutivas como una herramienta para diseñar alas de aviones más aerodinámicas. A partir de entonces, la

computación evolutiva ha sido aplicada con éxito en el diseño de soluciones en diferentes campos como la optimización de procesos industriales, bioinformática, ciencias sociales, optimización de transacciones financieras, etc.

Para entender el funcionamiento de un algoritmo evolutivo es necesario definir sus componentes principales. Los componentes principales de un algoritmo evolutivo son:

a. Una población de soluciones
b. Una función de fitness
c. Operadores de variación

Una población es un subconjunto del espacio de soluciones compuesta por individuos; cada individuo representa una solución del problema. El tamaño de la población permanece constante dentro de todo proceso evolutivo, lo que varía son los individuos dentro de la población. Los nuevos individuos de la población componen una nueva generación. Por ello, una generación es una instancia de la población en un punto específico en el transcurso del proceso evolutivo.

Un individuo es una estructura que agrupa las características esenciales de una solución al problema en un espacio de soluciones. Estas características almacenadas en unidades más simples son conocidas como genes. Los genes a su vez se encuentran ordenados en una estructura,

como un arreglo, conocida como cromosoma. Estos términos son tomados de la biología molecular pues cumplen funciones análogas en el almacenamiento de la información “genética” que distingue a un individuo de otro. Suponga, como ejemplo, el problema de hallar la persona con menos peso y altura dentro de una población. Para este problema, una solución se representa como un individuo con un arreglo de 2 genes, donde un gen representa el peso y otro gen representa la altura de una persona. El arreglo de 2 genes constituye el cromosoma contenido por cada individuo del espacio de soluciones.

En el proceso evolutivo, las poblaciones evolucionan en función del valor de fitness de sus individuos. El valor de fitness es un número real asignado por una función de evaluación a cada individuo de la población. Este valor representa cuán bueno es un individuo como solución al problema. El fitness determina la calidad de una población, por lo que, si el valor de fitness del mejor individuo aumenta o si mejora el valor de fitness promedio quiere decir que la población está evolucionando en la dirección deseada.

Elegir una función apropiada de fitness es una tarea complicada. En problemas de optimización donde existe una función objetivo definida, se toma ésta como la función de fitness; sin embargo, muchos problemas no cuentan con una función objetivo definida, para estos se necesita adaptar funciones ya existentes que determinen la aptitud de las soluciones del problema.

Los operadores de variación añaden nuevos individuos a una población. Estos operadores se basan en la explotación y exploración sobre el espacio de búsqueda para brindar una diversidad a la población que permita encontrar mejores soluciones al problema. Las operaciones de variación se ejecutan sobre uno o varios individuos de la población generando uno o más individuos para la nueva generación. Los individuos envueltos en los métodos de explotación y exploración se los conoce como individuos padres; los individuos generados como resultado de las operaciones de variación se los denomina individuos hijos.

Los métodos de explotación, también conocidos como operadores de recombinación combinan las características genéticas deseables de dos padres en uno o varios hijos, reforzando la presencia de individuos con características similares en la población. En líneas generales, esto provoca que la población se homogenice, pues la mayoría de hijos tendrán las características de los padres y por ende estarán relacionados entre sí. La recombinación tiende a converger las poblaciones alrededor de un selecto grupo de individuos cuyas características son refinadas con la intención de encontrar la solución correcta del problema.

Los métodos de exploración, también conocidos como operadores de mutación, exploran el espacio de soluciones en búsqueda de individuos con características diferentes a los individuos presentes en la población. La mutación consiste en modificar el contenido de los genes en un individuo a

fin de generar un nuevo genotipo. Este genotipo presenta características diferentes a los demás individuos de la población, brindando una diversidad necesaria para evitar una rápida convergencia del algoritmo en soluciones óptimas locales.

Es importante notar que las operaciones de variación y evaluación se realizan sobre todos los individuos de una población. Esto influye directamente en el tiempo de ejecución de los algoritmos evolutivos, pues mientras otros métodos realizan una búsqueda secuencial sobre el espacio de búsqueda, los algoritmos evolutivos realizan operaciones simultáneas sobre varios individuos, lo cual brinda dos ventajas: evita la posibilidad de caer en mínimos locales, pues tienen mecanismos para explorar otras soluciones del espacio de búsqueda, y evalúa varios individuos a la vez, por lo que demora menos tiempo en encontrar la solución correcta al problema.

El siguiente pseudocódigo describe a breves rasgos el procedimiento detrás de la mayoría de algoritmos evolutivos:

```
P -> población inicial
Q -> mejores individuos de la población

P <- Constuir Población Inicial
Best <- valor constante
Repeat
   Evaluar_Fitness(P)
   Q <-Seleccionar_Mejores(P)
   P <- Operadores de Variación(Q)
   Best <- Mejor(Best,P)
until (Best es la mejor solucion o se acabe el tiempo)
return Best
```

La mayoría de algoritmos evolutivos construyen una población inicial de forma aleatoria, generando los individuos de la población en base a una distribución uniforme. Luego se evalúa cada individuo de la población en base a la función de fitness, y en base a ella se seleccionan los mejores individuos de la población. Las operaciones de variación se realizan sobre los mejores individuos en Q y se añaden a la nueva población P. Estas operaciones se repiten hasta encontrar la solución correcta al problema o si después de un cierto número de iteraciones el algoritmo no encuentra una mejor solución. Al final de la ejecución del algoritmo evolutivo, el mejor individuo de la población es presentado como la mejor solución encontrada al problema.

La computación evolutiva presenta 3 ramas principales de algoritmos: las estrategias evolutivas, los algoritmos genéticos y los algoritmos por estimación de distribuciones.

3.2 Las estrategias evolutivas

Las estrategias evolutivas [21] son algoritmos evolutivos que sólo utilizan la mutación como operador de variación para generar los individuos de la nueva población. La selección de los mejores individuos se realiza utilizando la selección por truncamiento. Otra característica de un algoritmo basado en estrategias evolutivas es la representación de los individuos como

un vector de números reales. Las estrategias evolutivas fueron desarrolladas por Ingo Rechenberg y Hans Paul Schwefel en la universidad técnica de Berlín a mediados de los 60.

La operación de mutación consiste en alterar de forma aleatoria el contenido de uno o más genes dentro de un individuo. Ese operador se ejecuta sólo sobre un individuo y el resultado es un nuevo individuo, pues al cambiar su configuración genética, éste se constituye en un nuevo individuo. Dada la representación numérica de los genes en un individuo, la operación de mutación elige al azar una posición en el cromosoma de un individuo y cambia su contenido del gen seleccionado por un valor obtenido de una distribución normal con media 0 y variancia 1.

La selección por truncamiento ordena todos los individuos en función de su valor de fitness y sólo escoge los individuos con mejor fitness. Si bien este método asegura que los mejores individuos servirán de padres para la siguiente generación, también ocurre que resta diversidad a la población provocando que el algoritmo converja rápidamente en soluciones óptimas a nivel local.

Existen 2 versiones de estrategias evolutivas, conocidas como EE-(μ, λ) y EE-(μ+λ). La estrategia evolutiva EE-(μ, λ) consiste en una población de λ individuos, de los cuales se eligen los μ mejores mediante la selección por truncamiento, y mediante la operación de mutación se produce una nueva

población de λ individuos descartando por completo los individuos de la población anterior. Generalmente, el valor de μ es un múltiplo de λ.

La estrategia evolutiva EE-(μ+λ) consiste en elegir los μ mejores individuos de una población de tamaño λ. Los mejores individuos de la población (μ) pasan a formar parte de la siguiente generación, y generan a su vez λ hijos. La nueva población estaría compuesta de los μ padres más los λ hijos generados, de allí la numeración (μ+λ). Para la siguiente generación, la selección por truncamiento elegiría μ padres de la nueva población μ+λ y descartaría los individuos con menor valor de fitness y así sucesivamente hasta terminar el proceso evolutivo.

3.3 Los algoritmos genéticos

Los algoritmos genéticos [30] son métodos adaptativos de búsqueda sobre un espacio de soluciones. Se caracterizan por utilizar las operaciones de mutación y recombinación como agentes principales en la variación de los individuos de la población.

Los algoritmos genéticos fueron desarrollados a finales de los 70 por el científico americano John Holland. En su obra clásica, *Adaptation in Natural and Artificial Systems* [11] Holland aplicó los conceptos de adaptación biológica para resolver problemas en campos tan diversos como la economía, psicología, teoría de juegos e inteligencia artificial. Los AG son ampliamente utilizados como métodos de optimización; sin embargo, su

diseño inicial fue orientado a la implementación de sistemas adaptativos robustos.

La diferencia principal de los algoritmos genéticos con respecto a las estrategias evolutivas radica en los operadores de variación utilizados. Mientras que los algoritmos genético utilizan la recombinación como operación principal de variación, las estrategias evolutivas dependen casi exclusivamente del operador de mutación para modificar los individuos de una población. Los algoritmos genéticos requieren al igual que todo método de la computación evolutiva de una estructura que represente a los individuos junto con una función de fitness que evalúe cuán buena es un individuo como solución correcta del problema. Adicionalmente a estos factores, los algoritmos evolutivos utilizan los siguientes componentes para su correcto funcionamiento:

Operadores de selección.- los operadores de selección cumplen un papel importante en el proceso evolutivo, se encargan de seleccionar los mejores individuos de una población para ejecutar sobre ellos operadores de variación que darán paso a nuevos individuos de la población. La selección de los individuos padres es siempre aleatoria, pero se guía mayormente por el valor de fitness de los individuos. Existen una gran variedad de funciones de selección, pero las más comunes son las siguientes:

Selección por truncamiento: La selección por truncamiento ordena todos los individuos en función de su valor de fitness y elige sólo los

individuos que tienen el mejor valor de fitness por generación. Esta selección asegura que los individuos de la siguiente generación serán hijos de los mejores individuos de la generación anterior. Esto, sin embargo, puede llevar en algunos casos a una rápida convergencia del algoritmo genético en soluciones óptimas localmente.

Selección proporcional al valor de fitness: También conocida como la selección por la rueda de ruleta, consiste en asignar a todos los individuos de la población una probabilidad de selección. Luego se genera un valor al azar entre [0,1] y en base a este valor se elige al individuo con mayor probabilidad de selección. La probabilidad de selección representada por p_i está dada por la fórmula:

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$$

Donde f_i representa el valor de fitness del individuo, f_j representa la suma del valor de fitness de todos los individuos de la población. Mientras mayor sea el valor de fitness del individuo, mayor será la probabilidad de resultar seleccionado como padre para la siguiente población.

Esta selección puede ser vista como una rueda de ruleta, en la cual los individuos con mejor fitness abarcan una mayor área en la rueda de ruleta que los individuos menos aptos. Una vez que la rueda comienza a dar vueltas los individuos con mejor fitness tendrán mejores oportunidades de

ser seleccionados; sin embargo, siempre puede darse el caso que un individuo menos idóneo salga favorecido. Esto brinda a la siguiente generación una diversidad necesaria para evitar caer en óptimos locales.

La selección proporcional al fitness presenta problemas cuando existen individuos con valores de fitness muy similares. En esos casos este criterio de selección no permite discriminar cuál de ellos resulta en mejor padre para la siguiente generación, por lo que la selección por torneo es una buena alternativa como operador de selección.

Selección por torneo: La selección por torneo realiza una competencia entre individuos de la población en función de su valor de fitness. La selección por torneo consiste en seleccionar 2 individuos al azar y comparar sus valores de fitness. El individuo con mayor valor de fitness es elegido como padre para la siguiente población y el individuo con menor valor de fitness es descartado. Este método de selección permite almacenar el número de victorias que ha tenido cada individuo. Esto sirve como criterio adicional de selección, pues no solo se compara el valor de fitness, sino que además se toma en cuenta el número de veces que ha demostrado ser una solución superior al resto de individuos. Eso es muy útil en escenarios donde existen muchos individuos con valores de fitness muy cercanos entre sí.

Funciones de variación.- Las funciones de variación son operadores que exploran el espacio de soluciones del problema en base a los individuos

presentes en la población en cierta instancia del proceso evolutivo. Esta exploración tiene como propósito encontrar individuos con un mejor fitness que los individuos presentes en la población, lo que conlleva a que la nueva generación estará más próxima a la solución correcta del problema y con ello, a la convergencia del algoritmo genético.

Los operadores de variación se dividen en función de su operación sobre el espacio de búsqueda. Los operadores de exploración son conocidos como operadores de mutación; los operadores de explotación son conocidos como operadores de cruce o recombinación. A continuación se muestran los métodos de cruce y mutación más conocidos:

Operadores de Mutación

Los operadores de mutación modifican el contenido de los genes de un individuo de forma aleatoria. A partir de esta operación se obtiene un nuevo individuo, el cual puede estar en un área diferente en el espacio de soluciones al área sobre la cual se está llevando a cabo la búsqueda; por esta razón, esta operación es conocida como una exploración del espacio de búsqueda pues explora áreas que de otra forma pudieron haber pasado desapercibidas. Los operadores de mutación más conocidos son:

Mutación uniforme: La mutación uniforme consiste en seleccionar los genes de un individuo en base a probabilidad p =0.5. El contenido del gen seleccionado en la mutación toma un valor aleatorio del conjunto de símbolos correspondientes a su representación.

Mutación Gaussiana: La mutación gaussiana consiste en elegir un gen del individuo al azar y modificar el contenido del gen en función de una distribución normal con $\mu=0$ y $\sigma^2=1$.

Mutación por cambio de orden: este método de exploración consiste en elegir de forma aleatoria un par de genes en un individuo e intercambiar el contenido de los genes seleccionados.

Mutación de un punto: la mutación en un punto consiste en elegir de forma aleatoria una posición en el individuo y cambiar el contenido únicamente de ese gen. Esta mutación depende en gran medida de la representación utilizada para los genes. Si cada gen almacena un valor booleano, esta operación de mutación es conocida como bit-flip. En el caso que los genes tomen valores de un alfabeto definido al inicio del experimento, el gen seleccionado para mutar su contenido tomará un valor aleatorio del alfabeto.

Operadores de Recombinación

Los operadores de cruce o recombinación tienen como propósito generar nuevos individuos a partir de la combinación de las mejores características de dos individuos que presenten buenos valores de fitness. La idea detrás de este método se basa en la combinación de características deseables de dos individuos que conlleva la creación de un individuo que tenga lo mejor de ambos padres, y por ende, sea un individuo más cercano a la solución del problema. La operación de recombinación es también conocida como una operación de explotación de un área de búsqueda, pues concentra los esfuerzos de búsqueda en el espacio cercano a los individuos seleccionados, buscando explotar las mejores características de los mejores individuos para la siguiente generación. La operación de cruce trabaja sobre 2 individuos y genera a su vez dos nuevos individuos para la siguiente población. Los operadores de cruce más utilizados son:

Cruce de un punto: El cruce en un punto es una operación que selecciona al azar una misma posición en ambos individuos. La selección se hace sobre el intervalo [0, *l*-1] donde *l* representa la longitud del vector de genes perteneciente a cada individuo. Una vez seleccionada la posición inicial p, se intercambia el contenido genético entre p y *l* - 1. Nótese si p = 0 o p = *l* -1, la operación de cruce no se realiza pues ambos individuos permanecen iguales.

Cruce en dos puntos: el cruce en dos puntos sigue un procedimiento similar al cruce en 1 punto, con la variante en que ahora existen dos posiciones p_1 y p_2 donde p_1 pertenece al intervalo [0, *l*-1] y p_2 pertenece al intervalo [p_1, *l*-1]. Una vez definidas las posiciones al azar, se intercambia el contenido de los genes entre p_1 y p_2.

Cruce uniforme: El cruce uniforme, a diferencia de los cruces en uno y dos puntos, intercambia el contenido genético entre 2 individuos en base a una probabilidad p =0.5. Si un valor aleatorio p_i toma valores en el intervalo [0,1] es mayor o igual a p, el contenido de genes en la posición correspondiente es intercambiado, de lo contrario, permanecen en las mismas posiciones

Los operadores de variación mostrados anteriormente pueden sufrir modificaciones en base a la representación utilizada para los individuos del problema; queda a criterio del investigador la elección de qué operadores son los más apropiados en el diseño de una solución basada en un algoritmo genético. De darse el caso, es posible que tenga que inventarse nuevos operadores de variación, siempre y cuando sigan el principio de explorar o explotar nuevas soluciones sobre el espacio de búsqueda.

A continuación, se muestra un pseudocódigo de un algoritmo genético básico para ayudar a entender el funcionamiento del mismo.

```
P // poblacion
Q // poblacion de hijos
popsize //tamaño P
childsize//tamaño Q
P <- {}
For popsize do
   P <- P U {individuo generado al azar}
Repeat
   for Pi e P do
      FuncionFitness(Pi)
   Q <- Elitismo(P) //de estar presente el elitismo
   for childsize do
      Padre Pa <- Seleccion(P)
      Padre Pb <- Seleccion(P)
      Hijos Ha,Hb <- Cruce(Pa,Pb)
      Q <- Q U {Mutar(Ha),Mutar(Hb)}
   P <- Q
   Best <- Mejor (P)
Until (condición terminación)
Return Best
```

El proceso evolutivo empieza generando al azar los individuos de la población P. A partir de esta tarea entra a un proceso iterativo en el cual:

1. Se evalúa cada individuo en P en base a la función de fitness.
2. Se seleccionan los padres de la siguiente generación en base al operador de Selección elegido.
3. Se realiza la operación de cruce entre los individuos padres seleccionados
4. Se realiza la operación de mutación sobre los individuos hijos producidos en el proceso de cruce anterior.

5. Se añaden los individuos hijos a la siguiente generación.

El proceso evolutivo se ejecuta hasta que se ejecute la condición de terminación. Esta condición de terminación puede ser cuando no se encuentre el mejor individuo de las poblaciones, es decir, no presente cambios en su valor de fitness, o después de un cierto número de ejecuciones. Finalmente, el mejor individuo de la población es presentado como la solución al problema encontrado por el algoritmo genético.

Este esquema de funcionamiento de un algoritmo genético sirve sólo como un modelo, ya que existen diversas variaciones dependiendo de la naturaleza del problema y requerimientos en el diseño de la solución.

Algunos algoritmos genéticos añaden elitismo antes de ejecutar los operadores de variación para evitar perder los mejores individuos en el proceso de selección y variación. Otros algoritmos genéticos trabajan con tasas de cruce y mutación. Estas tasas brindan al proceso evolutivo la posibilidad de no ejecutar operaciones de mutación o cruce sobre los individuos seleccionados de la población. Esto refuerza la idea detrás de un proceso de búsqueda aleatoria sobre el espacio de búsqueda, una de las principales fortalezas de los algoritmos genéticos. Otros algoritmos genéticos optan por no generar aleatoriamente los individuos en la primera población, prefieren tomar los mejores individuos almacenados en experimentos anteriores.

3.4 Los algoritmos por estimación de distribuciones (EDA)

Los algoritmos de estimación de distribuciones o EDA [18] son métodos de búsqueda en un espacio de soluciones basados en la estimación de un modelo probabilístico a partir de los mejores individuos de la población. Un modelo probabilístico es una representación explícita del comportamiento de una o varias variables aleatorias. Esta representación se expresa mediante una función de distribución de probabilidad y un conjunto de parámetros, como la media y varianza, y permite inferir resultados futuros de las variables relacionadas a partir del modelo.

Existen dos diferencias principales entre los métodos evolutivos clásicos y los basados en la estimación de distribuciones. La primera diferencia tiene que ver con la manera en que generan los individuos para la siguiente generación; mientras los AE utilizan operadores de variación como el cruce y mutación de los individuos padres, los EDAs obtienen una nueva generación mediante el muestreo de una distribución de probabilidad; esta distribución es estimada a partir de los individuos seleccionados de la generación anterior.

Como se mencionó anteriormente, el comportamiento de los algoritmos evolutivos habituales depende de varios parámetros como las tasas de cruce y mutación, tamaño de la población, el número de generaciones, etc. Si no se tiene experiencia en el uso del algoritmo evolutivo, determinar los valores adecuados para los parámetros anteriores

se convierte en sí mismo en un problema de optimización. Los EDA requieren un menor número de parámetros pues no utilizan operadores de variación; esto conlleva a un ahorro en tiempo y recursos al momento de resolver el problema.

Otra diferencia importante estriba en la forma de expresar la relación entre las variables (en el campo de la computación evolutiva, una variable es representada por un gen) de los individuos. Mientras que la mayoría de algoritmos evolutivos no toman en cuenta las relaciones existentes entre los individuos, los EDA permiten en ciertos casos expresar de forma explícita la interrelación entre las variables mediante la distribución de probabilidad conjunta asociada a los individuos seleccionados de cada generación.

Para entender el funcionamiento de un algoritmo por estimación de distribución básico resulta útil el siguiente pseudocódigo:

EDA
Do **<-** Generar M individuos al azar
Repetir para l**=**1**,**2**,.....** hasta que se cumpla el criterio de parada
  D^{Se}_{l-1} **<-** Seleccionar N**<=**M individuos D_{l-1} acorde con el método de selección
  p_l**(**x**)=** p**(**x**|** D^{Se}_{l-1}**)** **<-** Estimar la distribución de probabilidad de los individuos seleccionados
  D_l **<-** Muestrear M individuos **(**la nueva población**)** a partir de p_l**(**x**)**

D_0 representa la población inicial de tamaño M, D_{l-1}^{Se} al conjunto de individuos seleccionados de D_{l-1}, D_l representa la nueva generación muestreada a partir de $p_l(x)$ y $p_l(x)$ representa la estimación de la distribución de probabilidad de los individuos seleccionados.

Las operaciones de generación y selección de individuos son tomadas de los algoritmos genéticos. Las operaciones de estimación y muestreo son métodos únicos de los EDAs, y son claves en la dirección que toma el proceso evolutivo.

La estimación de una distribución de probabilidad está basada en un modelo probabilístico que describa el comportamiento de las variables en base a los datos existentes. Esta estimación se realiza en base a distribuciones ya existentes, como la distribución dicotómica, distribución de Poisson, distribución Normal, etc. Cada distribución tiene parámetros que deben estimarse; por ejemplo, la distribución normal cuenta con dos parámetros que definen el modelo: la media y la varianza. El estimador que se utilice para calcular los parámetros queda a criterio del investigador.

La elección del modelo probabilístico apropiado para estimar el comportamiento de los datos es arbitraria. Para fines de este trabajo hemos notado que la distribución normal define un modelo probabilístico que se presenta con mayor frecuencia en la naturaleza. Existen algunos EDAs que trabajan con la distribución normal univariada o multivariada en función del número de variables implicadas y la complejidad del fenómeno natural que se está modelando.

Una vez elegida la función de distribución apropiada, es necesario tomar una muestra del modelo probabilístico estimado a partir de los mejores individuos de la población. Ese muestreo conlleva la creación de nuevos individuos para la siguiente generación. La operación de muestreo depende tanto del modelo estadístico como del número de variables del problema. Por ejemplo, para obtener una muestra M de un modelo normal multivariado, es necesario aplicar la siguiente fórmula:

$$M = \mu + Chol(\Sigma) + Z$$

Donde μ representa la media, Chol (Σ) es la descomposición de Cholesky de Σ, que representa la matriz de covarianza y Z un vector de la mismas dimensiones de M, cuyos valores son generados aleatoriamente utilizando la transformada de Box-Müller. Además de muestrear el modelo, en algunos casos se necesita de una función de mapeo, que transforme los elementos en el espacio del modelo probabilístico en individuos con su respectiva representación en enteros, bits o caracteres, según los requerimientos del problema.

La condición de terminación en los EDAs depende al igual que en los algoritmos evolutivos de la convergencia en una solución correcta al problema. Criterios como el tiempo de ejecución o el número de repeticiones del mejor individuo de la población son factores adicionales que dependen

del investigador para ser tomados en cuenta según las características del problema en cuestión.

Los EDAs se clasifican de acuerdo al tipo de dependencia entre las variables presentes en el modelo probabilístico [33]; esta dependencia puede ser de 3 tipos: univariada, bivariada o multivariada.

Los EDAs univariados asumen que todas las variables son independientes y calculan la distribución de probabilidad conjunta de los individuos seleccionados como el producto de las distribuciones univariantes independientes. Esta clase de EDAs, como PBIL [9] y UMDA [4] son los más sencillos que existen, son muy útiles para resolver problemas con variables continuas [11].

Los EDAs multivariados calculan la distribución de probabilidad conjunta del modelo mediante estadísticas de orden mayor a uno. A medida que el número de dependencias entre la variables es mayor que en las categorías anteriores, la complejidad de la estructura probabilística requiere un proceso de aprendizaje más complejo. La mayoría de algoritmos por estimación de probabilidades de varias variables utilizan las redes bayesianas como modelo probabilístico base para estimar el comportamiento de las variables.

3.5 Aplicación de los algoritmos genéticos y algoritmos por estimación de distribuciones a problemas de optimización clásicos

A continuación se presentan 2 problemas típicos en donde los métodos basados en la computación evolutiva brindan mejores resultados que otros métodos de solución. Para ambos problemas se diseñaron 2 soluciones, la primera basada en los algoritmos genéticos y la segunda basada en los algoritmos por estimación de distribuciones, tomando un modelo normal univariado como modelo probabilístico para generar nuevos individuos en la población.

Problema #1.- Cálculo del punto central de una Esfera de n dimensiones

Dada una esfera de N dimensiones con radio R, el objetivo es encontrar un punto Z dentro de la esfera tal que f(Z) = R^2. f(Z) está dada por la siguiente ecuación

$$f(Z) = R^2 - (x_1^2 + x_2^2 + x_3^2 + \cdots + x_n^2)$$

Esta ecuación está basada en la ecuación cartesiana de una esfera de N dimensiones. Mientras Z esté más cerca del origen, f (Z) se acercará a R^2 pues la diferencia entre R^2 y la suma de los componentes de Z será mayor. Por lo tanto, este problema puede ser visto como uno de maximización de la función f. Para efectos prácticos, se toma al eje coordenado en el centro de

la esfera, de tal manera que se busca el punto Z en donde todos sus componentes tienen el valor de

$$Z = \begin{bmatrix} 0_1 \\ \vdots \\ 0_n \end{bmatrix}$$

En la resolución de este problema utilizando AG y EDA, se tomó una esfera de dimensión N = 5 y R=20.

En ambos métodos evolutivos, cada individuo de la población representa un punto dentro de la hipersfera. Este punto es representado mediante un vector de enteros de tamaño n = 5. Cada gen de un individuo representa la coordenada del punto sobre una de las dimensiones del espacio donde se encuentra la hiperesfera. Por ejemplo, el siguiente individuo representa un punto ubicado dentro de esfera de radio R.

$$I = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 14 \\ 18 \end{bmatrix}$$

Nótese que los valores que los genes pueden tomar están acotados por el valor que tenga el radio de la esfera.

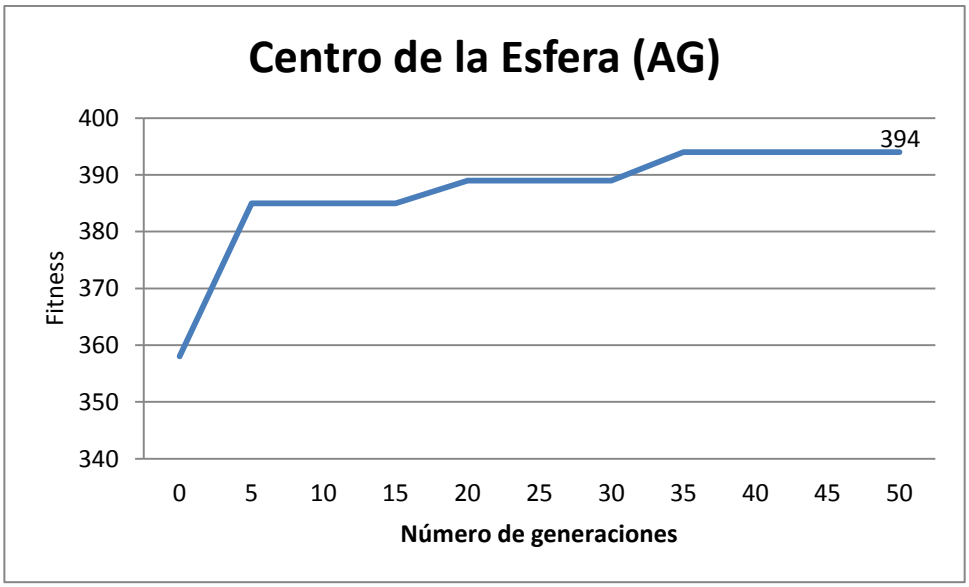
La primera solución a este problema está basada en un algoritmo genético. La siguiente tabla muestra las características del método basado en AG; P_c representa la tasa de cruce y P_m la tasa de mutación.

Tabla 4.- Parámetros del método basado en AG que encuentra el máximo de la función con simetría esférica

Población	100
P_c	0.8
P_m	0.005
Elite	0.1
Selección	Prop fitness
Función de Fitness	Ecuación de la esfera
Tipo de Cruce	1 Punto
Tipo de Mutación	Uniforme
Radio	20

El siguiente gráfico muestra el comportamiento del mejor individuo en cada generación hasta alcanzar la Generación 50 en donde el método termina su ejecución.

Figura 11.- Resultados en la evolución del fitness del mejor individuo a través de generaciones sucesivas



El método basado en AG tiene una tasa de cruce de 0.80, esto quiere decir que realiza la operación de cruce con mayor frecuencia que la operación de mutación. Este método también utiliza el elitismo, tomando el 10% de los mejores individuos de cada población como individuos élite. Para una población de 100 individuos el método converge en la generación 35 a un punto Z con fitness f (Z)=394. Para este valor de fitness, el punto Z correspondiente tiene las coordenadas $(1,0,0,2,0)$, el cual está cerca del centro de la Esfera.

El otro método desarrollado está basado en los algoritmos por estimación de distribuciones. La siguiente tabla muestra los únicos parámetros necesarios para su ejecución.

Tabla 5.- Parámetros del método basado en ED que encuentra el máximo de la función con simetría esférica

Población	30
Modelo	Normal Univariado
Media inicial (μ)	1
Varianza inicial (σ^2)	0
Radio	20

A diferencia del método basado en AG, este método no requiere de muchos parámetros, sólo necesita conocer el tamaño de la población, el radio R y el modelo probabilístico utilizado. La distribución de las variables es estimada en base a un modelo normal univariado estándar con media y varianza iniciales de 1 y 0 respectivamente. Estos valores se modifican en

función de las estimaciones posteriores de los modelos probabilísticos a partir de los individuos seleccionados de la población.

Para entender mejor el proceso en que se generan los nuevos individuos en los algoritmos por estimación de distribuciones, suponga que los individuos I_1,I_2 y I_3 son seleccionados como los mejores de la población. Estos individuos están constituidos de la siguiente manera:

$$I_1 = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 14 \\ 18 \end{bmatrix} \qquad I_2 = \begin{bmatrix} 5 \\ 12 \\ 1 \\ 4 \\ 3 \end{bmatrix} \qquad I_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

La media del modelo probabilístico estimado a partir de los 3 individuos tendrá una media igual a 4.22 y una varianza igual a 32.63. A partir de este modelo es posible generar nuevos individuos en base a la siguiente fórmula:

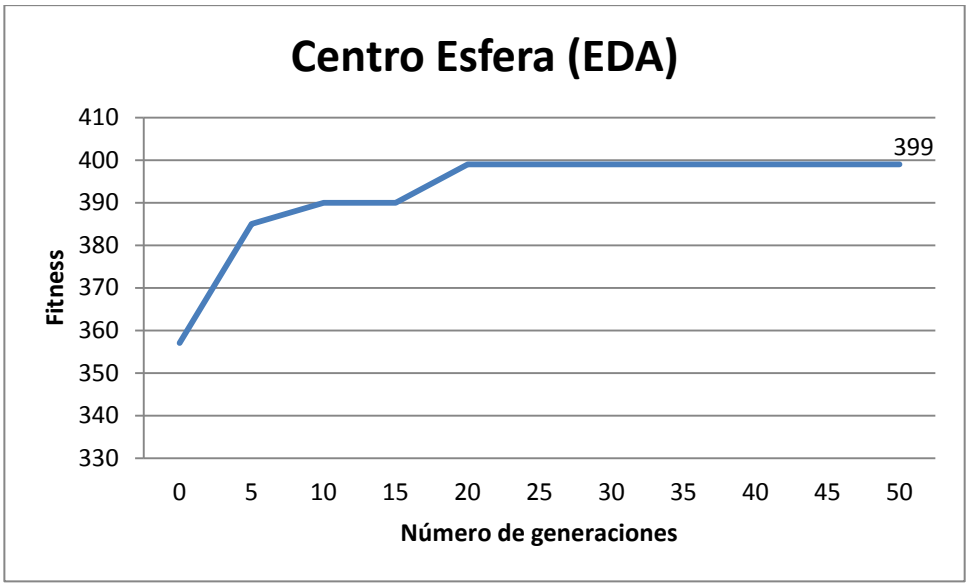
$$I_i = \mu \quad + Z * \sigma^2$$

Donde i representa el componente en la coordenada i del nuevo individuo generado y Z es una variable aleatoria que toma valores de la distribución Box-Muller. En base a esta fórmula, los nuevos individuos generados tienen los siguientes valores.

$$I_{1'} = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad I_{2'} = \begin{bmatrix} 5 \\ 3 \\ 1 \\ 2 \\ 3 \end{bmatrix} \qquad I_{3'} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 2 \\ 0 \end{bmatrix}$$

Estos nuevos individuos representan puntos más cercanos al centro de la esfera, mostrando así que los nuevos individuos de la población están más cerca a la solución del problema q los individuos anteriores. Los valores generados por la fórmula son valores reales, de los cuales el componente entero es tomado para los genes de los nuevos individuos. La siguiente tabla muestra la evolución de los valores de fitness del mejor individuo de cada generación.

Figura 12.- Resultados en la evolución del fitness del mejor individuo a través de generaciones sucesivas



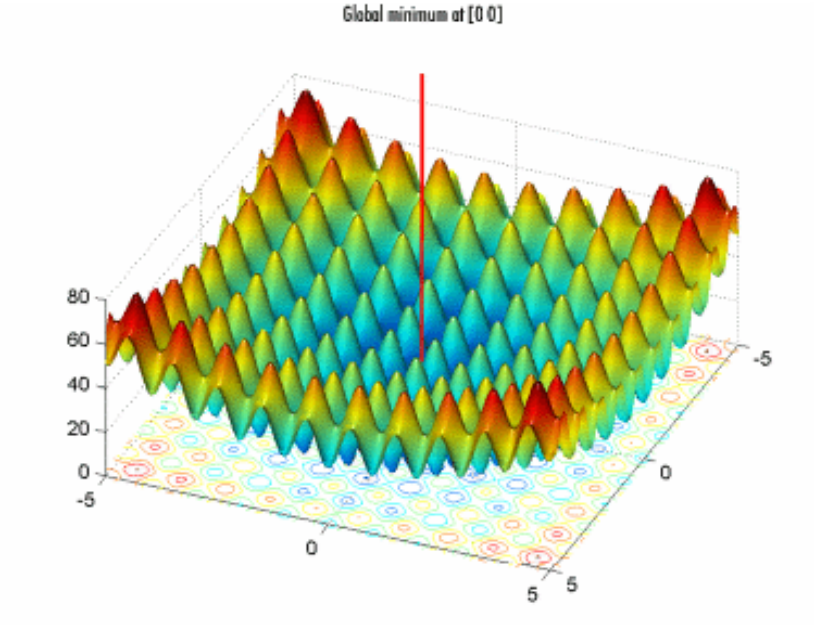
El centro de la esfera encontrada fue el punto Z= $(1,0,0,0,0)$ con valor de fitness f (Z)= 399. El método basado en EDA converge desde la generación 25.

En los resultados de resolver este problema se puede ver que el método basado en la ED encuentra una solución más precisa y en menos generaciones que el método basado en los algoritmos genéticos.

Problema #2.- Búsqueda del mínimo global de la función de Rastrigin

La función de Rastrigin es una función multimodal muy utilizada para probar el desempeño de cualquier algoritmo de optimización. Esta función multimodal es apropiada para probar si un método puede encontrar el mínimo en una función, pues, como lo indica la siguiente gráfica, presenta muchos picos y valles. El mínimo global de la función de Rastrigin se encuentra en el punto (0,0)

Figura 13.- Gráfica de la función de Rastrigin



La función de Rastrigin presenta la siguiente ecuación:

$$Ras(X) = 20 + x_1^2 + x_2^2 - 10(cos2\pi x_1 + cos2\pi x_2)$$

Las variables x_1 y x_2 toman valores únicamente dentro del intervalo [-5.12, 5.12]. Encontrar el mínimo de esta función es un problema de optimización de Ras(X). Por ende, se toma la ecuación de la función de Rastrigin como función objetivo, en donde se busca el punto X tal que R(X)=0.

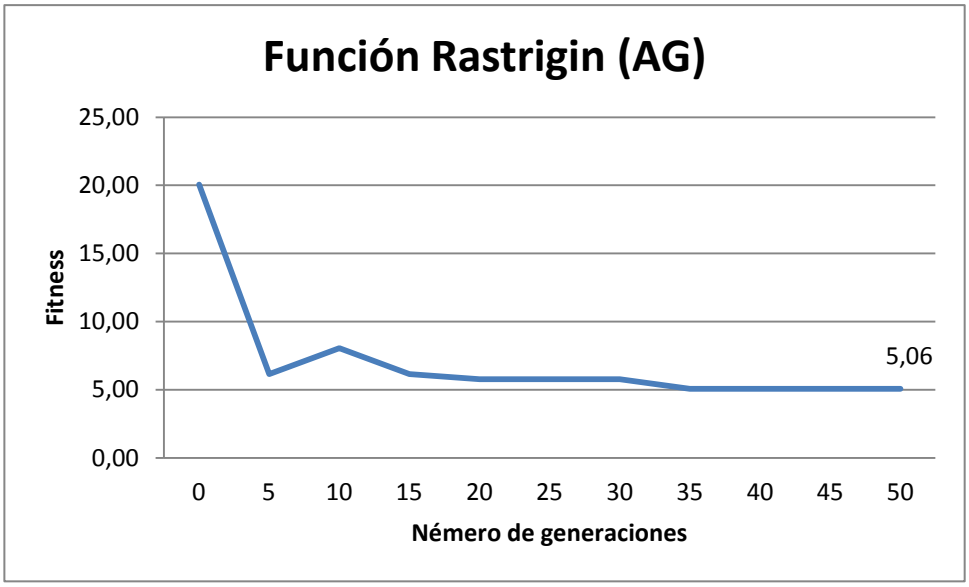
El primer método, basado en AG fue utilizado para encontrar el mínimo de la función de Rastrigin. La siguiente tabla muestra los parámetros utilizados para ejecutar este método:

Tabla 6.- Parámetros del método basado en AG para encontrar el mínimo de la función de Rastrigin

Población	100
P_c	0.8
P_m	0.005
Elite	0.1
Selección	Prop. fitness
Función de Fitness	Función de Rastrigin
Tipo de Cruce	1punto
Tipo de Mutación	Uniforme

Los resultados pueden verse en el siguiente gráfico.

Figura 14.- Evolución del fitness del mejor individuos a través de generaciones sucesivas bajo el método basado en EDA



El punto mínimo encontrado fue el punto X= $(-0.93,0)$. Para este punto, su valor de fitness f(X)= 5.06. Si bien es cierto X está muy cerca al mínimo global, su valor de fitness se encuentra lejos del valor esperado. Este método convergió desde la generación 35.

Por otro lado, el algoritmo por estimación de distribución utilizado para resolver este problema utiliza los siguientes parámetros que se encuentran en la siguiente tabla.

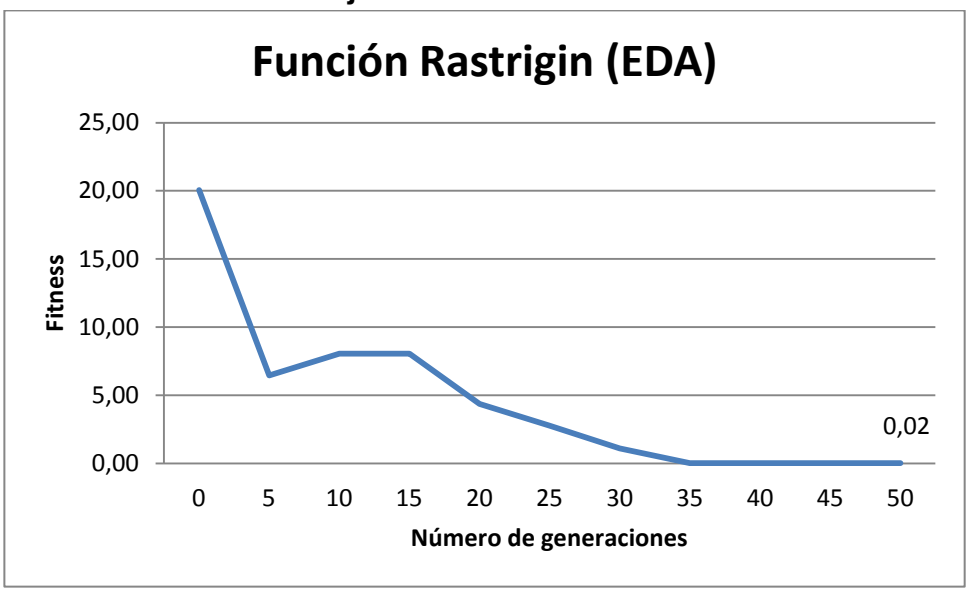
Tabla 7.- Parámetros del método basado en EDA para encontrar el mínimo de la función de Rastrigin

Población	30
Modelo	Normal Univariado
Media inicial (μ)	1

Varianza inicial (σ^2)	0

Los resultados después de 50 iteraciones están graficados en el siguiente cuadro:

Figura 15.-Evolución del fitness del mejor individuos a través de generaciones sucesivas bajo el método basado en EDA



El mínimo encontrado en la función de Rastrigin fue en el punto X con coordenadas $(0.01,0)$. El valor de fitness de este punto es f(X)= 0.02. El método basado en la ED converge desde la generación 35.

En ambos problemas, los mejores resultados se obtuvieron al trabajar con el método basado en la estimación de distribuciones utilizando como modelo probabilístico la distribución normal univariada. Estos resultados

fueron obtenidos en un menor tiempo que el método basado en los algoritmos genéticos.

Del análisis anterior se concluye que, dentro de la computación evolutiva, los algoritmos basados en la estimación de la distribución presentan mejores resultados y demoran menos tiempo en encontrarlos que los métodos basados en los algoritmos genéticos. Ya en la implementación de las soluciones al problema de búsqueda de motivos utilizando métodos evolutivos, esperamos observar un comportamiento similar.

Capítulo 4

Aplicación de métodos evolutivos al problema de búsqueda de motivos

Este capítulo tiene como propósito mostrar cómo se aplicaron los conceptos y métodos de la computación evolutiva a la solución del problema de la búsqueda de motivos biológicos. De los varios métodos implementados, se dará especial énfasis a aquellos dos que mejores resultados dieron al momento de ser probados: el MBMAG que utiliza Algoritmos genéticos y el MBMEDA que aplica los algoritmos por estimación de distribuciones.

Este capítulo se encuentra organizado en dos secciones. En la primera sección se describe de manera detallada el diseño de los elementos que fueron necesarios para construir las soluciones evolutivas que son el objeto de este trabajo; aquí se muestra cómo toman forma concreta los conceptos del capítulo de introducción a la computación evolutiva, al particularizarlos al

problema de búsqueda de motivos biológicos. Estos elementos o componentes están agrupados según la función que cumplen en el proceso evolutivo, siendo los grupos principales los siguientes: la representación de los individuos, la función de fitness, los operadores que se introdujeron para mejorar el rendimiento de los métodos. En cada grupo se presentan las opciones que fueron probadas en los diferentes métodos que se implementaron.

La segunda sección a su vez se divide en otras dos: en la primera se presenta el diseño de dos métodos evolutivos que utilizan algoritmos genéticos para resolver el problema, y que son llamados Método 1 y Método 2 o MBMAG; en la segunda parte se presentan otros dos, llamados Método 3 y método 4 o MBMEDA, que en cambio aplican algoritmos por estimación de distribuciones. Cada método se describe mostrando la forma en que los componentes descritos en la primera parte de este capítulo se ensamblan para realizar la búsqueda. Finalmente, se presenta en pseudocódigo el procedimiento de cada método.

4.1 Componentes de los métodos de búsqueda de motivos basados en la computación evolutiva

4.1.1 Estructuras de datos de representación de individuos

Como se vio en el capítulo anterior, los individuos de una población son una estructura bajo la cual se agrupa información importante que

representa una solución al problema. Estas unidades de información, los genes, se pueden representar por variables de tipo booleana, enteras, reales o caracteres.

En el problema de búsqueda de motivos, un individuo constituye una representación del conjunto de palabras que conforman el motivo buscado. Existen dos formas de expresar esta representación:

Vector de caracteres: Los individuos representados como palabras de longitud *l* pertenecen a un espacio de soluciones S de cardinalidad 4^l. Los genes toman valores al azar del conjunto definido por el alfabeto genético {A, C, G, T}.

Vector de números enteros: Los individuos representados por un vector de números enteros pertenecen a un espacio de soluciones Q compuesto por permutaciones con repetición *t* de $n - l + 1$, donde *n* es la longitud de las secuencias en la base de ADN, *l* la longitud de las palabras del motivo y *t* el número de regiones promotoras en la base de ADN. Cada individuo es representado por un vector de posiciones iniciales (VPI) donde cada celda valores aleatorios entre $[0, n - l + 1]$. El valor en cada celda almacena la posición donde inicia una palabra del motivo en una región promotora en la base de ADN. El tamaño del vector de posiciones iniciales está limitado por el número de regiones promotoras en la base de ADN, es decir, por la variable *t*.

La siguiente tabla muestra un individuo representado por un vector de posiciones iniciales (VPI) de tamaño t = 10; cada gen guarda la posición inicial de una palabra del motivo en las secuencias correspondientes en la base de ADN, mientras que *S* es un vector de las palabras pertenecientes al patrón que representa el individuo. En la práctica, S puede ser construido como una matriz de t x l, donde t es la dimensión del VPI y l la longitud de las palabras del motivo.

Tabla 8.- Construcción del vector de palabras S a partir de un vector de posiciones iniciales VPI

VPI	Base de Secuencias	S
4	acgtCGATTGCctaag	CGATTGC
2	taTGATCGAtgacgca	TGATCGA
3	cgaCAATTGAgcttac	CAATTGA
1	gCGCTCGAcaagctgt	CGCTCGA
4	cgttTGTCACAgtctaa	TGTCACA
5	tcagcCACACCCagct	CACACCC
6	ccagagCGTCTGAttg	CGTCTGA
6	gacttcaCGACTGAttg	CGACTGA
10	gctgcccatCGATTGA	CGATTGA
8	ccagtacCGATTGCa	CGATTGC

A pesar que el espacio de soluciones utilizando la representación de un individuo como vector de posiciones iniciales es mayor a la representación como un vector de caracteres, la representación como un vector de posiciones iniciales permite una modificación más sencilla de un

individuo, lo cual permite realizar los cálculos de su valor de fitness y funciones de variación de forma más eficiente.

Junto con la representación de un individuo como vector de posiciones iniciales, es posible conocer la distribución de los nucleótidos dentro de las palabras del motivo definidas por este individuo. Para ello, se utiliza una representación conocida como la matriz de pesos posicionales.

Matriz de pesos posicionales (MPP)

Una matriz de pesos posicionales o MPP es en pocas palabras una matriz normalizada de las frecuencias de los nucleótidos en un individuo. La matriz de frecuencias, como se vio en el capítulo 2.1, se obtiene calculando la frecuencia en cada columna del vector de palabras S que conforman el motivo. La matriz de pesos posicionales multiplica la matriz de perfiles por $^{1}/_{t}$, donde t es la dimensión del vector de posiciones iniciales de un individuo. La matriz de pesos posicionales MPP tiene una dimensión de 4 x l, donde l representa el tamaño de una palabra del motivo.

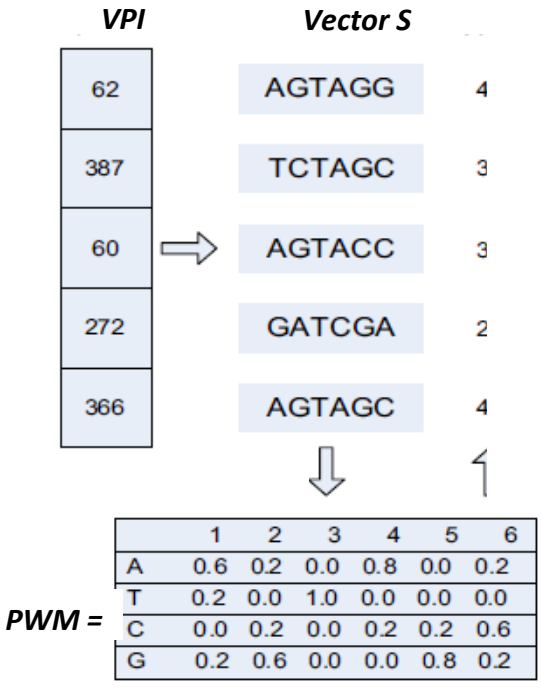
Los valores dentro de la matriz de pesos posicionales (MPP (i, j)), representan la frecuencia del símbolo i en la columna j; i toma valores numéricos basados en la enumeración arbitraria de los símbolos del alfabeto genético. El criterio de enumeración depende de un orden arbitrario de las filas en la matriz MPP. Para efectos de este trabajo, el orden de las filas se determina por la ubicación en el alfabeto latino de las letras que representan

los nucleótidos. Por ende, una matriz de pesos posicionales tendrá el siguiente cuadro:

$$PWM = \begin{bmatrix} A & | & 0.2 & \dots \\ C & | & 0.3 & \dots \\ G & | & 0.3 & \dots \\ T & | & 0.2 & \dots \end{bmatrix}$$

La siguiente figura muestra el proceso a partir del cual puede obtenerse la matriz MPP basada en la representación de un individuo como vector de posiciones iniciales (VPI).

Figura 16.- construcción de la matriz de pesos posicionales a partir del vector de posiciones iniciales



Cada elemento del VPI determina una palabra en la secuencia correspondiente en la base de ADN. Luego de tener todas las palabras almacenadas en S, se calculan los valores en MPP como la frecuencia de cada símbolo por columna dividido para la dimensión del VPI. Por ejemplo, MPP (A, 1) = 3/5 = 0.6. Esto indica que la variable A aparece con mayor frecuencia en la primera columna que el resto de símbolos del alfabeto. Una vez calculados los valores en la matriz de pesos posicionales, se puede calcular el contenido de información del individuo.

4.1.2 Funciones de Evaluación de los individuos

Distancia de Hamming

El concepto de distancia entre 2 secuencias sirve como herramienta para evaluar el grado de similitud entre las mismas. Mientras menor sea la distancia entre ellas, tendrán un mayor número de símbolos comunes en posiciones correspondientes. Si las secuencias son iguales, la distancia entre ellas es 0.

La distancia de Hamming es un algoritmo que calcula la distancia entre dos secuencias de igual longitud; esté método fue presentado con más detalles en el capítulo 2.1. Una extensión de este algoritmo es utilizado para calcular la distancia entre una secuencia y la base de ADN en el primer método de búsqueda exhaustiva.

Aunque los métodos exhaustivos están descartados como soluciones eficientes al problema de búsqueda de motivos, la distancia de Hamming puede ser utilizada como función de fitness para una solución utilizando algoritmos genéticos, especialmente para la representación de individuos como palabras de un alfabeto genético.

Contenido de información (Entropía relativa)

El contenido de información (IC) [43] es un indicador de la similitud entre la distribución de los nucleótidos de una base de ADN y los nucleótidos presentes en las palabras del motivo. Mientras menor sea la similitud entre un conjunto de palabras y la base de ADN, mayor será la probabilidad que estas palabras conformen el patrón buscado.

Generalmente se usan dos indicadores que determinan el grado de divergencia entre dos funciones de distribución: la prueba X^2 (chi cuadrado) y el ratio log likelihood. La mayoría de trabajos en este campo utilizan el ratio log likelihood, que tiene la siguiente función:

$$\sum_{j=1}^{L}\sum_{i=1}^{A} n_{i,j}\, ln\frac{p_i}{f_{i,j}}$$

El ratio log likelihood Donde *L* representa el tamaño de las palabras del motivo, bajo el supuesto que todas las palabras del motivo tienen el mismo tamaño. *A* es el número de símbolos del alfabeto, en este caso 4 ya que es un alfabeto genético. La probabilidad p_i es la probabilidad del

nucleótido *i* en la base de ADN; $n_{i,j}$ es la ocurrencia del nucleótido *i* en la posición *j* y $f_{i,j}$ es la frecuencia del nucleótido *i* en la posición *j*, la cual se calcula mediante la fórmula $f_{i,j} = n_{i,j}/N$, siendo *N* el número de palabras que componen el motivo.

Talk-Ming Chan et al. [40] utilizan una versión normalizada de este ratio, la cual tiene la siguiente fórmula:

$$IC = \sum_{i=1}^{L} \sum_{b} f_b(i) log \frac{f_b(i)}{p_b}$$

Donde b representan los valores que pueden tomar las variables que representan los nucleótidos p_b es la probabilidad del símbolo *b* en la base de ADN y f_b es la frecuencia normalizada de un símbolo *b* en cierta posición *i*. Este indicador de contenido de información se lo conoce como la divergencia Kullback-Leibler o entropía relativa.

Para entender mejor el concepto detrás del contenido de información de un individuo, tomemos como ejemplo la búsqueda de un patrón en la siguiente base de ADN. El patrón a buscar se encuentra resaltado con letras mayúsculas:

aagtcat**AACAG**tgagtgatgtga

tagtcattgag**AACAG**cgatctga

ga**AACAG**atcgatgagcgagctg

cactcgtcgagcgat**AACAG**ctgc

Para fines prácticos, el patrón es el mismo en cada una de las filas y esta identificado por la variable I_1.

$$I_1 = \begin{bmatrix} AACAG \\ AACAG \\ AACAG \\ AACAG \end{bmatrix}$$

El contenido de información de I_1 es 2.02, siendo este el mayor valor de contenido de información que puede alcanzar un individuo. Si se escoge al azar un conjunto de secuencias de la base anterior para formar un individuo, denotado por I_2,

$$I_2 = \begin{bmatrix} TGAGT \\ TCATT \\ GAGCG \\ TCGTC \end{bmatrix}$$

Este debería de tener un valor de fitness inferior a las secuencias que conforman el patrón. Calculando su contenido de información, este es 1.07, lo cual confirma que las secuencias que conforman el patrón a buscar tendrá un mayor contenido de información mientras este patrón sea lo más diferente posible a la distribución de los nucleótidos en el resto de la base de ADN.

4.1.2 Operadores de Selección

Selección por torneo

La selección por torneo es un método que realiza una competencia entre *m* individuos elegidos al azar de la población, y sólo escoge al mejor individuo en base a su valor de fitness. El número *m* de individuos presentes en el torneo es arbitrario; muchos trabajos que utilizan a la selección por torneo como método de selección trabajan con un valor de $m = 2$.

Para los métodos evolutivos desarrollados en este trabajo, la selección por torneo utiliza un valor de $m = 2$ donde cada individuo contendiente es seleccionado de forma aleatoria de la población. Adicionalmente se añade una tasa de selección p_{tour} que toma valores entre [0,1] Para valores altos de p_{tour}, la probabilidad de elegir el individuo con mayor fitness es elevada; sin embargo, a veces puede ser seleccionado el individuo perdedor para así brindar mayor diversidad a la siguiente generación. Para los métodos evolutivos desarrollados en este trabajo se utiliza una tasa de selección $p_{tour} = 0.8$.

Selección proporcional al fitness

El método de selección proporcional al fitness, también conocido como rueda de la ruleta, es un método que asigna a los individuos con mayor fitness una mayor probabilidad de ser elegidos como padres de la siguiente generación. Como se explicó en el capítulo 3, la selección proporcional al fitness suma los valores de fitness de todos los individuos de una población y

asigna una probabilidad de selección a cada individuo conforme al porcentaje que representen de la suma del fitness de la población. De esta forma, los individuos con mayor fitness tendrán siempre una mayor probabilidad de ser escogidos, sin descartar por completo individuos con bajo valor de fitness. La probabilidad de selección representada por p_i está dada por la fórmula:

$$p_i = \frac{f_i}{\Sigma_{j=1}^{N} f_j}$$

Donde f_i representa el valor de fitness del individuo, f_j representa la suma del valor de fitness de todos los individuos de la población. Mientras mayor sea el valor de fitness del individuo, mayor será la probabilidad de resultar seleccionado como padre para la siguiente población.

4.1.3 Operadores genéticos de variación de la población

Operaciones de cruce

Los operadores de cruce utilizados en los métodos desarrollados en este trabajo son la recombinación en 1 y entre 2 puntos.

Cruce en un punto.- La recombinación en 1 punto elige al azar una posición *i* entre [0, *l*-1] en dos individuos y cruza el contenido de los genes a partir de *i*.

Cruce en 2 puntos.- El cruce entre 2 puntos es similar al cruce en un punto, con la diferencia en que se escogen 2 posiciones *i* y *j* entre [0, *l*-1] en dos individuos y se intercambia el contenido en los genes dentro del intervalo en esas posiciones.

Ambos operadores de cruce utilizan una tasa de cruce que controla la frecuencia con que esta operación ocurre a través del proceso evolutivo. La tasa de cruce ($p_{c)}$ toma valores entre [0,1]. Un valor cercano a 1 en la tasa de cruce implica una mayor probabilidad en la ejecución del cruce como operador de variación de la población. El valor de la tasa de cruce utilizada en este trabajo es igual a 0.9. La elección de este valor es de carácter arbitrario y depende tanto del criterio del investigador como del cálculo del valor óptimo para el problema.

Operadores de Mutación

Los operadores de mutación utilizados en este trabajo son la mutación uniforme y la mutación en un punto.

Uniforme.- La mutación uniforme altera los genes de un individuo siguiendo una distribución normal univariada. Dado un valor ρ = 0.5, se genera un valor aleatorio x entre 0 y 1 por cada gen del individuo; si x > 0.5, el gen toma un valor al azar del alfabeto genético.

Un punto.- La mutación en un punto cambia el contenido de un gen elegido al azar en el individuo. Para la representación de un individuo como un VPI, el valor que el gen puede tomar esta dentro del intervalo [0, n-l+1], siendo n el tamaño de la secuencia en la base de ADN y l la longitud de las palabras del motivo.

La operación de mutación se ejecuta en base a una tasa de mutación denotada por p_m. En los métodos desarrollados en este trabajo, se utilizó una tasa de mutación $p_m = 0.1$.

4.2 Operadores de variación de la población de los métodos basados en la estimación de distribuciones

Para construir una solución al problema de búsqueda de motivos en base a los algoritmos por estimación de distribuciones, es necesario construir un modelo probabilístico y una función de muestreo correspondiente al modelo probabilístico elegido. Muchos fenómenos naturales presentan un comportamiento que puede ser modelado utilizando una distribución gaussiana; por ello, en este trabajo utilizamos 2 modelos probabilísticos basados en la distribución gaussiana de una o varias variables, la cual supone una relación de dependencia o independencia entre las variables del problema.

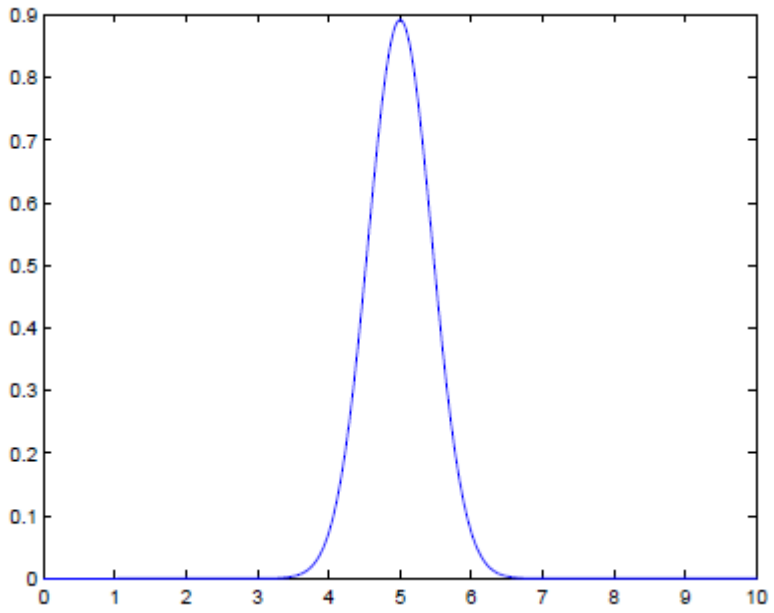
Modelos Probabilísticos

Modelo Normal Univariado.- Una variable aleatoria x se comporta según una distribución normal con media $\mu \in R$ y varianza $\sigma^2 \in R$, y su función de densidad de probabilidad está definida por:

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(x-\mu)^2)$$

La siguiente figura ilustra la distribución normal de una variable que toma valores entre 0 y 10.

Figura 18.- Gráfico de una distribución normal de una variable con media 5 y varianza 9.16



La media es el valor promedio bajo el cual se agrupan los valores posibles que puede tomar la variable *x*. La varianza es una medida de la dispersión de valores alrededor del valor medio. Para la ilustración anterior, μ =

5 y σ^2 =9.16. En este ejemplo, la variable x representa la distribución de un símbolo del alfabeto en la base de ADN.

Operador de muestreo de una distribución normal

Para muestrear información a partir de un modelo normal univariado se presenta la siguiente fórmula:

$$I_b = \mu_b + Z * \sigma_b^2$$

Donde I_b es el componente del nucleótido *b* de la matriz de pesos posicionales MPP de un individuo I, $\mu_b$representa la media del nucleótido *b* y σ_b^2 la desviación estándar del nucleótido *b* en base a la distribución de *b* en las secuencias de un individuo. Z es un valor aleatorio de una distribución normal obtenida mediante la transformada de Box-Müller. Bajo este modelo probabilístico, la construcción de un nuevo individuo comienza mediante la generación de sus componentes en la matriz de pesos posicionales MPP en base a un modelo normal univariado.

Una vez realizado el muestreo sobre el modelo probabilístico, es necesaria una función de mapeo la cual genera un nuevo individuo a partir de los datos obtenidos en el muestreo del modelo probabilístico elegido. La operación de muestreo obtiene valores en el rango de [0,1] del modelo, estos valores permiten construir una matriz de pesos posicionales o MPP de dimensiones b x t, donde *b* representa el número de símbolos del alfabeto genético, donde b =4 y t el número de palabras que conforman el motivo. Al

multiplicar MPP por el valor escalar t se obtiene la matriz de perfiles que determina la frecuencia de los nucleótidos en cada columna del vector de palabras S. El siguiente pseudocódigo muestra el proceso de cálculo del vector de posiciones iniciales a partir de la matriz de pesos posicionales.

```
Mapeo (MPP){
        for i < = l
                for x < = 3
                        for y < = MPP(x,i)
                                k <- aleatorio (0,t-1)
                                S(k,i) <- b;
        for z <= t
                VPI[Z] <- Distancia_Hamming (S(z), ADN(z))
Return VPI
}
```

El vector de palabras S es construido a partir de los valores de frecuencia de la matriz de pesos posicionales. Cada celda en MPP, representada por MPP(x,i) determina la frecuencia del nucleótido x en la columna i. La construcción de S se realiza por columnas; dada la frecuencia de cada símbolo del alfabeto, se distribuyen de forma aleatoria en una columna hasta que todas las posiciones quedan llenas. Para entender mejor este proceso, suponga que la matriz MPP a partir del muestreo de un modelo gaussiano Univariado es la siguiente:

A:	0.3	0.1	0.6	0.7	0.4	0.0	0.7
C:	0.6	0.4	0.2	0.1	0.2	0.2	0.1
G:	0.1	0.5	0.0	0.1	0.2	0.8	0.0
T:	0.0	0.0	0.2	0.1	0.2	0.0	0.2

Si el vector de posiciones iniciales de cada individuo tiene dimensión 10, hallar las frecuencias es cuestión de aplicar la operación PFM = MPP x

$^{1}/_{10}$. MPP es la matriz de pesos posicionales donde cada celda representa la frecuencia. PFM sería:

A:	3	1	6	7	4	0	7
C:	6	4	2	1	2	2	1
G:	1	5	0	1	2	8	0
T:	0	0	2	1	2	0	2

En la primera columna hay 3 nucleótidos A. 6 nucleótidos C, 1 nucleótido G y ningún T. La forma para llenar los espacios es aleatoria; una forma de completar todos los espacios de la primera columna podría ser como sigue:

S=
A….
G….
C….
C….
C….
C….
C….
C….
A….
A….

Luego de llenar las *l* columnas de S, se calcula la distancia de Hamming de cada palabra en S a la base de ADN para encontrar la posición de la palabra en la secuencia correspondiente de la base de ADN de menor distancia a la palabra en S. Una vez completado el vector de posiciones iniciales, se

calcula su contenido de información en base a las palabras determinadas por el VPI.

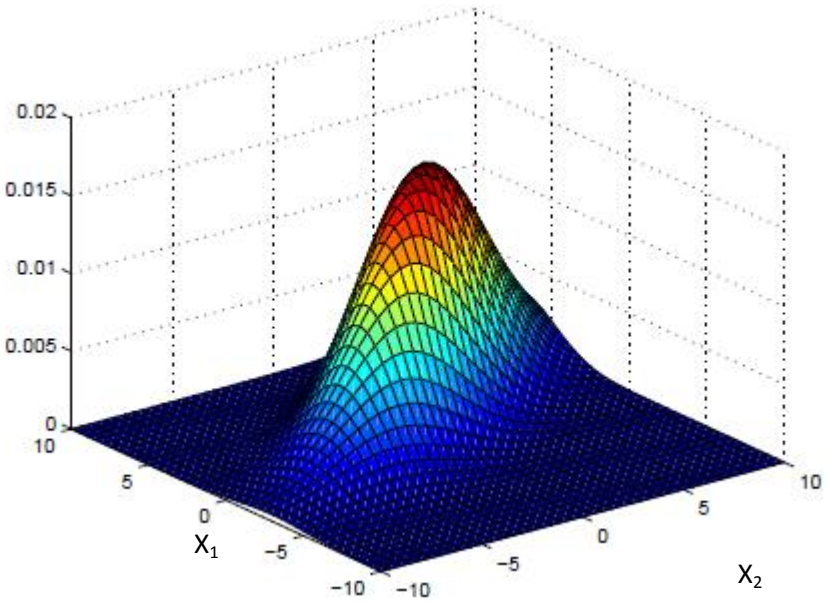
Modelo Normal Multivariado.- Un vector aleatorio $X = [X1……X_n]^T$ tiene una distribución normal multivariada con media $\mu \varepsilon R^n$ y una matriz de covarianza $\sum \varepsilon R^{n \times n}$ si presenta una función de densidad de probabilidad definida por :

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

Donde la variable n es la dimensión del vector X. Al igual que el modelo normal de una variable, la media representa el valor central de los valores en X. La matriz de covarianza Σ, también conocida como matriz de dispersión, es una matriz cuyas celdas representan la covarianza entre un par de variables x_1 y x_2. La covarianza es un estimador de la correlación entre 2 o más variables. El modelo normal multivariado es utilizado para describir la interrelación entre 2 o más variables que presenten una relación de dependencia.

El siguiente gráfico representa una distribución normal multivariada para una variable x de dos dimensiones cuyos componentes toman valores entre [-10, 10].

Figura 18.- Gráfico de una distribución normal multivariada



Operador de muestreo de una distribución normal multivariada

La operación de estimación de los parámetros del modelo es una tarea sencilla, pues basta con utilizar los estimadores básicos para cada parámetro. Sin embargo el muestreo de individuos a la siguiente población partir del modelo probabilístico está definido por la siguiente fórmula:

$$I = \mu + Chol(\Sigma) * Z$$

Donde I es el nuevo individuo, Z es un vector de números aleatorios con una distribución normal estándar generada usando la transformada Box-Müller y Chol (Σ) es una operación de factorización que se ejecuta sobre la matriz de varianza y covarianza Σ para hallar su correspondiente matriz triangular inferior. Para aplicar la factorización de Cholesky sobre una

matriz, ésta debe ser definida positiva. En el método 2, el estimador utilizado para hallar la matriz de covarianza $\sum$ genera una matriz de covarianza semipositiva. Esta restricción impuesta por el estimador de $\sum$, evita que en ocasiones se pueda realizar la descomposición de Cholesky para $\sum$ y por ende, no sea posible realizar el muestreo de nuevos individuos a partir del modelo probabilístico estimado.

4.3 Operadores Adicionales al proceso evolutivo

A veces resulta necesario añadir funciones al proceso evolutivo que eviten una rápida convergencia en óptimos locales y que mejoren el valor de fitness de ciertos individuos de la población. Las siguientes funciones son utilizadas en los diversos métodos evolutivos implementados de búsqueda de motivos.

Elitismo

El elitismo es un método que añade un conjunto de individuos llamados élites a la siguiente generación de la población. Los individuos élites son individuos con los mejores valores de fitness de toda la población, por lo que siempre serán útiles para mantener la búsqueda cerca de la solución correcta al problema. El número de individuos élites se calcula de forma porcentual a la población, mediante una tasa de elitismo. Generalmente, la tasa de elitismo no supera el 10%; ya que, si se escogen muchos individuos como élites, el algoritmo converge de forma prematura en

soluciones óptimas a nivel local, resultando difícil encontrar la solución global del problema.

Operador de filtrado local

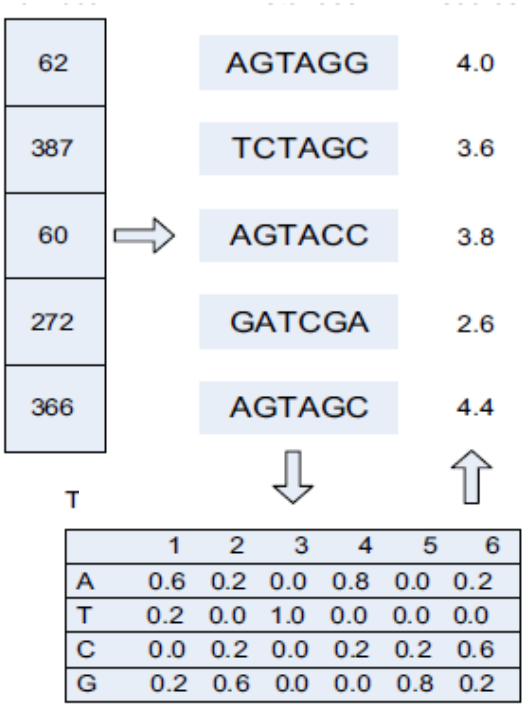
La operación de filtrado local [37] modifica el valor de los genes en un individuo en base a la similitud presente entre la palabra definida por la posición almacenada en el gen y la palabra de consenso del individuo. La palabra de consenso es una palabra de longitud *l* construida en base a la concatenación de los símbolos que aparecen con más frecuencia en cada columna de la matriz MPP. Esta palabra es una representación del conjunto de palabras que conforman el motivo. La similitud entre cadenas de nucleótidos describe cuánto en común tienen entre sí. Esto parte de la idea que dentro del motivo buscado, las cadenas de nucleótidos deben ser similares, lo cual no necesariamente quiere decir que tengan los mismos nucleótidos en posiciones similares.

La similitud de una palabra *p* en un individuo está definida por la siguiente fórmula:

$$Sim(p) = \sum_{i=1}^{L} PWM(b_i, i)$$

Donde L es el tamaño de *p*, $PWM(b_i, i)$ es el valor de la celda en la matriz de pesos posicionales correspondiente a la posición *i*. b_i es una variable que toma los valores en el rango del alfabeto genético. La siguiente figura muestra el valor de similitud de cada palabra correspondiente a su posición inicial en la base de ADN. El valor de similitud se encuentra en la columna derecha.

Figura 17.- Medición de la similitud entre las palabras de un individuo



	1	2	3	4	5	6
A	0.6	0.2	0.0	0.8	0.0	0.2
T	0.2	0.0	1.0	0.0	0.0	0.0
C	0.0	0.2	0.0	0.2	0.2	0.6
G	0.2	0.6	0.0	0.0	0.8	0.2

Palabra de Consenso: A G T A G C

La columna de la derecha muestra los valores de similitud entre las cadenas de nucleótidos correspondientes a cada posición inicial del VPI.

Nótese que 4 de 5 cadenas tienen valores similares, cercanos a 4. Pero la cadena cuya posición inicia en 272 tiene un valor de similitud muy por debajo del resto. Esta palabra es una posible candidata a ser filtrada.

Este algoritmo es útil cuando hay uno o dos genes de un individuo que provoquen un valor de fitness bajo; a pesar que, las demás posiciones almacenadas en los genes son muy cercanas a las palabras del motivo en el problemas, las posiciones erróneas dañan el resultado global. A fin de corregir este problema, el filtrado local intenta encontrar en la localidad de las posiciones incorrectas una palabra tal que el fitness del individuo mejore.

El algoritmo de filtrado local funciona de la siguiente manera: El primer paso consiste en calcular el valor de similitud para todas las palabras de un individuo. Como segundo paso, se elige el gen cuya palabra correspondiente tenga el menor valor de similitud, y se recorre de forma iterativa la secuencia correspondiente a esa posición en la base de ADN hasta encontrar otra palabra que presente una mayor similitud con las otras palabras del individuo. De encontrarse una palabra con mayor similitud, la posición inicial de la nueva palabra es almacenada en el gen que contenía la anterior posición inicial y el primer paso se vuelve a ejecutar hasta encontrar el gen cuya palabra tenga el menor valor de similitud. El filtrado local concluye si el valor de similitud de una palabra en su secuencia correspondiente no mejora.

Una vez finalizado el filtrado local, la nueva distribución de los nucleótidos en nuevas palabras del individuo deben ser expresadas volviendo a calcular los valores de la matriz de pesos posicionales. Esta operación de filtrado local es de tipo voraz, por lo que debe tenerse cuidado en la frecuencia con que se realice la operación de filtrado en el proceso evolutivo, pues podría llevar a una convergencia temprana en soluciones óptimas a nivel local.

Operador de desplazamiento

El operador Desplazamiento [46] desfasa las posiciones iniciales del vector de posiciones iniciales del mejor individuo de la población para encontrar de ser posible un individuo con mejor fitness. Esta operación es útil cuando el algoritmo de búsqueda se estanca en un individuo después de un determinado número de ejecuciones. En ciertas ocasiones, la solución óptima se encuentra desfasada en unas pocas posiciones del mejor individuo encontrado en la población; el operador Desplazamiento permite explorar el espacio cercano de posiciones para encontrar un mejor individuo.

El pseudocódigo siguiente describe cómo funciona el operador Desplazamiento. Este operador trabaja sobre tres individuos: el original I y dos individuos cuyas posiciones iniciales en el vector de posiciones iniciales han sido desplazadas a la derecha e izquierda respectivamente; estos nuevos individuos son expresados por las variables I_D e I_Z. Se calcula el

valor de fitness de los nuevos individuos y se selecciona el mejor de los tres como el mejor individuo de la población. La variable *s* representa el número máximo de desplazamientos que pueden ejecutarse hasta llegar al tope de las palabras en la base de ADN.

Código del operador Desplazamiento

```
Desplazamiento (I, s) {
 para (int k=1;k<=s;k++){
        I_D = I + k;
        I_Z = I - k;
        if(fitness(I_D) > fitness(I) Y fitness(I_D) >= fitness(I_Z))
                Retornar I_k+;
        if(fitness(I_Z) > fitness(I) Y fitness(I_Z) >= fitness(I_D))
                Retornar I_k-;
                }
        Retornar I;

}
```

4.3 Métodos evolutivos de búsqueda de motivos

En la segunda parte de este capítulo se presentan 4 algoritmos evolutivos de búsqueda de motivos que utilizan los conceptos mostrados en la primera parte como bloques para construir una solución al problema. En este trabajo se desarrollaron 2 métodos de búsqueda de motivos en base a los algoritmos genéticos denominados método 1 y método 2 (MBMAG), y 2 dos métodos de búsqueda de motivos basados en los algoritmos por estimación de distribuciones denominados método 3 y método 4 (MBMEDA).

4.3.1 Métodos de búsqueda de motivos basados en Algoritmos Genéticos

En esta sección se presentan dos aplicaciones de los algoritmos genéticos al problema de búsqueda de motivos. Estas aplicaciones utilizan componentes descritos al inicio de este capítulo, como son el modelo de representación de los individuos, la función de fitness, los operadores de selección y reproducción y otros operadores que mejoran su rendimiento.

Método 1

Los individuos en el método 1 son representados como un vector de caracteres, donde cada gen toma valores del alfabeto genético. La función de fitness utilizada en este método es una extensión de la distancia de Hamming entre un individuo y la base de ADN. El tamaño del espacio de búsqueda en base a esta representación es 4^{l}. El tamaño de la población es fijo durante todo el proceso evolutivo; para el método 1 se eligió una población de 500 individuos con una población de hijos generados de 250 individuos.

Como funciones de variación se utilizaron el cruce en dos puntos y la mutación uniforme, con tasas de cruce y mutación de 0.10. La selección de los mejores padres se realizó utilizando la selección proporcional al fitness. Los mejores individuos de cada población eran seleccionados para la nueva generación mediante la función de elitismo con una tasa de 0.10.

Este método fue una primera aproximación a una solución a la búsqueda de motivos utilizando algoritmos genéticos. El siguiente pseudocódigo muestra el funcionamiento del método 1 basado en algoritmo genético:

```
popsize = 500
childsize= 250
   P <- {}
   para popsize hacer
      P <- P U {individuo generado al azar}
   Best <- 0
   repetir
      para cada individuo Pi e P hacer
         Distancia_Hamming(Pi)
      si(Best = valor frontera o Fitness(Pi) > Fitness(Best)
         Best <- Pi
      Q <- Elitismo(P)
      para childsize hacer
         Padre Pa <- Selection_RW(P)
         Padre Pb <- Selection_RW(P)
         Hijos Ha,Hb <- Cruce_2Puntos(Pa,Pb)
            Q <- Q U {Mutar_Uniforme(Ha),Mutar_Uniforme(Hb)}
      P <- Q
   hasta (Best es la solucion ideal o paso del tiempo)
   retornar Best
```

Los resultados de éste y de los otros métodos implementados en este trabajo se analizarán en el capítulo 6. El método 1 presenta un excelente desempeño realizando la búsqueda sobre bases sintéticas de ADN, pero obteniendo resultados no deseables en la búsqueda sobre bases reales de ADN. Por ello se aplicó un nuevo método con mejoras sustanciales tanto en la representación de individuos y función de fitness como en operadores adicionales que prevengan una convergencia no deseada.

Método 2 (MBMAG)

El método 2 incorpora amplias mejoras en la representación y en la función de fitness utilizada con respecto al método 1. Cada individuo es representado como un vector de posiciones iniciales tomadas al azar de la base de ADN. Los genes toman valores dentro del rango [0, n-l+1]. La función de evaluación para calcular el valor de fitness de cada individuo utilizada es el contenido de información. Para evaluar el contenido de información de un individuo es necesario calcular los valores de la matriz de pesos posicionales a partir de las palabras definidas por el vector de posiciones iniciales correspondiente

La función de selección utilizada es la selección por torneo. Este operador presenta ventajas con respecto a la selección proporcional al fitness; mientras la selección proporcional al fitness asigna la misma probabilidad a individuos con valores de fitness muy similares, la selección por torneo elige los padres en base a los mejores individuos con respecto al resto de individuos de la población, lo cual brinda diversidad a la población evitando una convergencia prematura.

Los operadores de variación utilizados son: como función de cruce el cruce en 1 punto y como función de mutación la mutación en un punto, con tasas de cruce y mutación de 0.1 y 0.9 respectivamente. La estimación de estos

parámetros resulta en sí mismo un problema de optimización complejo; por cuestiones de tiempo se realizaron diferentes ejecuciones del método 2 variando las tasas de cruce y mutación hasta encontrar la combinación bajo la cual el método presentara los mejores resultados. El método 2 no utilizó elitismo, pero añadió las funciones de filtrado local y Desplazamiento como operadores de regulación de la convergencia. Ambos operadores de regulación se ejecutan cada 10 generaciones

El siguiente pseudocódigo muestra el funcionamiento del método 2:

```
popsize = 500
childsize= 250
Best <- 0
repetir
   P <- {}
   para popsize hacer
      P <- P U {individuo generado al azar}
   Best <- 0
   repetir
      para cada individuo Pi e P hacer
         Information_Content(Pi)
      si(Best = valor frontera o Fitness(Pi) > Fitness(Best))
         Best <- Pi
      para childsize hacer
         Padre Pa <- Selection_Tournament(P)
         Padre Pb <- Selection_Tournament(P)
         Hijos Ha,Hb <- Cruce_1Puntos(Pa,Pb)
         Q <- Q U {Mutar_1Pto(Ha),Mutar_1Pto(Hb)}
      si (%10)
         Filtrado Local(Q)
         Desplazamiento(Mejor(Q))
      P <- Q
   hasta (Best es la solucion ideal o paso del tiempo)
   retornar Best
hasta(numero de ejecuciones)
```

Una limitación de todo método evolutivo es que no siempre encuentran la misma solución dado que la generación de la población inicial es aleatoria. En base a esta limitación, es necesario la ejecución del proceso evolutivo un número determinado de veces para elegir la mejor solución posible. Para ello en el método 2 se define una variable que determina el número de ejecuciones del proceso evolutivo antes de mostrar el resultado final.

En nuestro caso, el número de ejecuciones es 10. Este procedimiento es utilizado también por otros métodos evolutivos [49]

La siguiente tabla muestran las características de ambos métodos basados en el algoritmo genético

Tabla 9.- Tablas de parámetros de los métodos de búsqueda de motivos basados en los algoritmos genéticos

Método 1 basado en AG

Representación:		Vector Caracteres
Tamaño Población:		500
Número de Hijos:		250
Función Fitness:		Distancia Hamming
Operador Selección:		Proporcional Fitness
Operador Reproducción:	**Mutación**	Uniforme (0.10)
	Cruce	2 Puntos (0.10)
Métodos Adicionales:		Elitismo (0.10)

Método 2 basado en AG (MBMAG)

Representación:		Vector Posición Inicial
Tamaño Población:		500
Número de Hijos:		250
Función Fitness:		Contenido de Información
Operador Selección:		Torneo
Operador Reproducción:	**Mutación**	Un punto(0.1)
	Cruce	1 Punto (0.9)
Métodos Adicionales: :		Desplazamiento
		Filtrado Local

4.3.2. Métodos de búsqueda de motivos basados en los algoritmos por estimación de distribuciones

Los métodos basados en el algoritmo por estimación de distribuciones toman algunos elementos presentados en los métodos basados en algoritmos genéticos en la construcción de sus soluciones. Por ejemplo, dados los resultados al utilizar la representación de un individuo como vector de posiciones iniciales y el contenido de información como función de fitness, ambos elementos son utilizados en ambos métodos basados en la ED. La operación de selección utilizada es la selección por torneo con m = 2. Los operadores adicionales de Filtrado local y Desplazamiento son utilizados para mejorar los individuos en escenarios como estancamiento en la mejor solución y en la mejora de la similitud entre las secuencias de un individuo.

La principal diferencia entre los métodos basados en la ED y los métodos genéticos es en la elección del modelo probabilístico del conjunto

de individuos padres, la función de muestreo relacionada con la densidad de probabilidad del modelo y de ser necesaria, una función de mapeo de la información obtenida del modelo probabilístico a la construcción de un nuevo individuo.

Método 3

El método 3 supone una relación de dependencia entre los genes de un individuo. Esta relación es expresada de forma explícita bajo el modelo Gaussiano multivariado. Este modelo es estimado a partir de la selección de los individuos padres de la población. El número de variables del modelo depende del número de palabras del motivo estimado en la base de ADN. Para este trabajo suponemos la presencia de una palabra del motivo en cada secuencia de ADN, donde la variable t representa el número de palabras del motivo.

El modelo Gaussiano multivariado depende de la estimación de la media y matriz de covarianza para poder muestrear valores a partir de su densidad de probabilidad. El estimador estándar de la media está definido por la función:

$$\mu = \frac{1}{T}\sum_{i=1}^{T} x_i$$

Donde T representa el número de genes de cada individuo. El estimador estándar de la matriz de covarianza se define por la siguiente función:

$$\frac{1}{T}\sum_{i=1}^{T}(x_i-\mu)\hat{(}x_i-\mu)'$$

El siguiente pseudocódigo muestra el procedimiento seguido para resolver el problema de búsqueda de motivos utilizando el método 1 basado en un Algoritmo por Estimación de distribuciones

```
popsize = 500 //tamaño la población
childsize = 250 //número de hijos generados
repeat
   P <- {}
   Pad <-{}
   para popsize times para
      P <- P U {individuo generado al azar}
   Best <- 0
   repeat
         para cada individuo Pi e P hacer
            Information_Content(Pi)
            si(Best = valor frontera o Fitness(Pi) > Fitness(Best)
               Best <- Pi
         Q <- {}
         Padres Pad <- Seleccionar_Tournament(P)
         M <- Modelo_Gaussiano_Multivariado(Pad)
         para childsize hacer
            H <- Muestrear_MGM(M)
            Q <- Q U {H}
         si (%10)
            Filtrado Local(Q)
            Desplazamiento(Mejor(Q))
         P <- Q
   hasta (Best es la solución ideal o paso del tiempo)
   retornar Best
hasta(constante de terminacion)
```

El muestreo se intentó realizar utilizando la función Muestrear MGM(M), donde M representa el modelo Normal Multivariado de dimensión t. La operación de muestreo está basada en la ecuación:

$$I = \mu + Chol(\Sigma) * Z$$

Donde I es el nuevo individuo y Z es un vector de números aleatorios con una distribución normal estándar generada usando la transformada Box-Müller. Para aplicar Cholesky la matriz debe tener la propiedad de ser positiva definida, y el estimador básico de ∑ obtiene generalmente una matriz semi positiva definida.

Este método presentó problemas al momento de muestrear un individuo del modelo, pues el estimador estándar de la matriz de covarianzas Σ no siempre estima una matriz definida semipositiva, requisito imprescindible para poder realizar la transformada de Cholesky.

Dadas estas limitaciones, fue necesario desarrollar un segundo método que evitara los problemas del método 3.

Método 4 (MBMEDA)

El método 4, también conocido por las siglas MBMEDA (método de búsqueda de motivos basado en un algoritmo por estimación de

distribuciones) intenta resolver el problema en el muestreo de nuevos individuos cambiando el modelo probabilístico estimado a partir de los mejores individuos de la población. En vez de contar con un modelo normal multivariado, el MBMEDA trabaja con 4 modelos normales de una variable; cada modelo representa la distribución de un símbolo del alfabeto genético a través de los individuos seleccionados de la población. Bajo este esquema, los nucleótidos en las palabras que forman un motivo son independientes.

Los parámetros que definen una distribución normal univariada son la media y la varianza. En este método se utilizaron el estimador estándar de estos parámetros. La media es calculada usando la siguiente fórmula:

$$\mu_b = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Donde N representa el número de padres seleccionados. La varianza es calculada bajo la siguiente fórmula:

$$\sigma^2{}_b = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2$$

Una vez estimado el modelo en base a sus parámetros, el muestreo de un nuevo individuo es realizado mediante la siguiente fórmula:

$$I_b = \mu_b + Z * \sigma_b^2$$

Donde I_b representa los valores muestreados a partir del modelo normal univariado correspondientes a una fila de una matriz de pesos posicionales (MPP).

El siguiente pseudocódigo muestra el procedimiento seguido para resolver el problema de búsqueda de motivos utilizando el MBMEDA basado en Algoritmos por Estimación de distribuciones:

```
popsize = 500 //tamaño la población
childsize = 250 //número de hijos generados
repetir
   P <- {}
   Pad <-{}
   para popsize times hacer
      P <- P U {individuo generado al azar}
   Filtrado_Local(P)
   Best <- 0
   repetir
         para cada individuo Pi e P hacer
            Information_Content(Pi)
            si(Best = valor frontera o Fitness(Pi) > Fitness(Best)
               Best <- Pi
         Q <- {}
         Padres Pad <- Seleccionar_Tournament(P)
         M[4] <- Modelo_Normal_Univariado(Pad)
         para childsize hacer
            H <- Muestrear_MNU(M)
            I <- Mapeo(H)
            Q <- Q U {I}
         si (%10)
            Filtrado_Local(Q)
            Desplazamiento(Mejor(Q))
         P <- Q
   hasta (Best es la solución ideal o paso del tiempo)
   retornar Best
hasta(constante de terminación)
```

La siguiente tabla muestra los parámetros y características del método MBMEDA.

Tabla 10.- Tabla de parámetros del método MBMEDA

Representación:		Vector Posición Inicial
Tamaño Población:		500
Número de Hijos:		250
Función Fitness:		Contenido de Información
Operador Selección:		Torneo
Operador Reproducción:	**Modelo Prob.**	Normal Univariado
	Muestreo	Func. Asociada al Modelo
Métodos Adicionales:		Desplazamiento
		Transformación
		Filtrado Local

Los métodos basados en algoritmos genéticos y el método EDAMD fueron probados sobre 2 tipos de bases de ADN: sintéticas, que son generadas bajo ciertos parámetros que se presentarán con más detalle en el capítulo 5 y bases de ADN reales, los cuales contienen motivos de organismos biológicos cuyas posiciones iniciales han sido previamente identificadas de forma experimental.

Estos métodos fueron desarrollados en el lenguaje C++ utilizando un IDE proporcionado por Microsoft Visual Studio 2008. Fueron desarrollados y corrieron sobre el sistema operativo Windows 7 de 64 bits con 6 GB de

memoria RAM. En el apéndice de este trabajo se encuentra mayor información acerca del prototipo de buscador de motivos implementado.

CAPITULO 5

Bases de Datos de ADN y métricas de Desempeño

Este capítulo tiene por propósito describir las características de los datos con que fueron probaron los métodos de búsqueda de motivos biológicos del capítulo 4, es decir: las bases de ADN. Las bases de ADN se clasifican según su procedencia; se conocen como bases sintéticas de ADN aquellas que se construyen artificialmente, tanto las instancias del motivo como las secuencias de nucleótidos que representan las zonas promotoras; la generación de estas bases requiere de ciertos parámetros que se definen posteriormente en este mismo capítulo. Las bases reales de ADN son conjuntos de secuencias de nucleótidos obtenidas de zonas promotoras de genes corregulados por el mismo factor de transcripción; su denominación se basa en el factor de transcripción regulador, que se fija en uno o más sitios (TFBS) de la zona promotora según se ha encontrado de manera experimental.

Posteriormente se presentarán dos métricas para medir el desempeño de los métodos de búsqueda de motivos con estas bases de ADN; estas métricas son la *precisión* y la *exhaustividad*, que fueron tomadas del campo de *recuperación de información*.

5.1 Bases sintéticas de ADN

Las bases sintéticas son bases de secuencias de ADN con una distribución aleatoria de símbolos tomados del alfabeto genético dentro del cual se inserta un patrón de palabras de longitud l el cual se busca identificar. Las bases sintéticas son utilizadas como prueba primaria de algunos algoritmos en el campo de la bioinformática, pues su construcción es sencilla y no requieren gran consumo de recursos en el proceso de hallar los motivos. Sin embargo, los resultados presentes en encontrar el motivo en bases sintéticas pueden resultar engañosos pues no implican que los resultados sean similares en la búsqueda sobre bases de ADN reales.

Construir una base de ADN sintética consiste en insertar un patrón de palabras de longitud l en una matriz de nucleótidos M de tamaño t x n, donde n representa la longitud de las secuencias en M y t el número de filas presentes en M. Este patrón insertado puede ser tan simple como t palabras con símbolos iguales en posiciones correspondientes, cada palabra en una secuencia de posiciones entre [0, n-l+1] tal como indica la siguiente figura.

Figura 11.- Base sintética de ADN donde se ha insertado el patrón a identificar

aagtcat**AACAG**tgagtgatgtga

tagtcattgag**AACAG**cgatctga

ga**AACAG**atcgatgagcgagctg

cactcgtcgagcgat**AACAG**ctgc

Este tipo de bases sintéticas es el más sencillo que existe. Encontrar el patrón de palabras es muy sencillo para algoritmos que utilizan fuerza bruta, pues las palabras del patrón no presentan ninguna alteración en sus símbolos. Es por eso, que estas bases sintéticas no sirven para probar la fortaleza de un algoritmo al enfrentarse contra bases de ADN reales.

Una opción para aumentar el grado de dificultad de una base sintética consiste en mutar símbolos en cada palabra a mostrar de forma aleatoria. Esta manera de construir los motivos sintéticos resulta ser diferente a la manera en que se distribuyen nucleótidos en los motivos reales, puesto que las mutaciones no ocurren al azar dentro de las palabras que conforman al motivo. Por ende, es necesario un nuevo enfoque en la creación de bases sintéticas con una mayor similitud a las bases reales de ADN conllevando a la creación de bases sintéticas basadas en escenarios biológicos.

Bases sintéticas basadas en escenarios biológicos

Un trabajo realizado por Zhi Wei et al [46] define 4 parámetros básicos para construir bases sintéticas de ADN más cercadns a bases reales. Estos parámetros son:

Número de Secuencias.- también conocido como el número de filas en M, esta variable puede tomar sólo 2 valores: 20 o 100.

Tamaño de las palabras del motivo.- definida por l, la longitud de cada palabra del motivo. Los valores que l puede tomar son dos: 8bp y 16bp.

Conservación del Motivo.- la conservación del motivo determina la integridad de un nucleótido a través de posiciones correspondientes en las palabras que conforman el patrón. Una conservación alta significa que cada nucleótido de una palabra de motivo en la primera secuencia de M tiene una probabilidad de 0.91 de conservar el mismo símbolo a través de las demás palabras. Una conservación baja asigna a cada nucleótido de la primera palabra del motivo una probabilidad de conservación de 0.7. Las siguientes bases muestran la diferencia entre una baja y alta conservación en las palabras del motivo:

Base con alta conservación	Base con baja conservación
aagtcat**AACAG**tgagtgatgtga	tagtcattgag**AGCAG**cgatctgaa
aatagtcattgag**AACAG**cgatct	ga**CATAT**atcgatgagcgagctgg
ga**AACTG**atcgatgagcgagctg	cactcgtcgagcgat**TAGAA**ctgca
cactcgtcgagcgat**TACAG**ctgc	cactcgtcgagcgat**TAGAT**ctgca

Nótese que las palabras en la base con alta conservación son muy similares a la palabra en la primera secuencia; por el contrario, en la base con baja conservación las palabras del motivo presentan mutaciones considerables, lo cual se espera en bases con palabras con poca conservación entre sí.

Ruido: el ruido en una base sintética está relacionado con el número de palabras del motivo presentes en cada secuencia de la base sintética. En la naturaleza, la mayoría de regiones reguladoras de los genes poseen sólo un binding site por factor de transcripción relacionado con la proteína producida. Sin embargo, existen ocasiones donde en una región reguladora no hay ninguna palabra del motivo, o hay más de una palabra en la misma secuencia. El factor de ruido de la base sintética añade esta característica para aumentar la dificultad en encontrar el motivo. La variable que representa el ruido puede tomar dos valores: 1 o 0, donde 1 representa la presencia de ruido y 0 una base sintética sin ruido.

Una base sintética de ADN sin ruido implica que en cada fila de M se inserta una sola palabra del motivo en una posición aleatoria entre [0, n-l +1]. Una base sintética con presencia de ruido tiene al menos una palabra del motivo en el 90% de las secuencias; la frecuencia con que aparecen más de una palabra en una misma secuencia está determinada por la distribución geométrica $p(t)=0.1*(0.9)^{(t-1)}$, donde t es el número de secuencias en la base sintética. El siguiente ejemplo muestra dos bases sintéticas, una con presencia de ruido y otra sin la presencia de ruido.

Base sintética sin ruido	**Base sintética con ruido**
aagtcat**AACAG**tgagtgatgtga	tagtcattgagAGCAGcgatc
tagtcattgag**AGCAG**cgatctga	gaCATATatccTATAAgca
ga**CATAT**atcgatgagcgagctg	cactcgtcgagcgatTAGAAc
cactcgtcgagcgat**TAGAA**ctgc	aagtcatAACAGtgagtgatga
aagtcat**AACAG**tgagtgatgtga	tagtcattgagaaccccgatctga

La siguiente tabla muestra todas las combinaciones posibles de los parámetros para construir una base sintética similar a las bases de ADN biológicas.

Tabla 1.- Tabla de bases sintéticas de ADN generadas

Ruido	Conservación	Tamaño palabras (*l*)	No. de secuencias(*t*)
Sin Ruido	Sin conservación	8	100
		16	100
	Con conservación	8	20
		16	20
		8	100
		16	100
Con Ruido	Sin conservación	8	100
		16	100
	Con conservación	8	20
		16	20
		8	100
		16	100

El siguiente código muestra cómo son generadas las bases sintéticas en c++.

```
const int n //longitud de cada secuencia en la base de ADN
/*
l.- tamaño de las palabras del motivo
t.- número de regiones reguladoras
*/
BaseSintetica(l,t,conservacion,ruido){

 char Motif[l];
 char BaseADN[t][n];
  //Generar el Motif
 Motif -> GenerarMotivo(l);
  //Generar la Base
 for  i e t do
    for  j e n do
       BaseADN[i][j] -> Generar_Aleatoriamente()
       pos-> aleatorio(0,n-l+1);
       if(!conservacion)
          Motif -> Mutar_Motif()
          BaseADN[i][j] -> Insertar_Motivo(Motif,pos)
       else
          BaseADN[i][j] -> Insertar_Motivo(Motif,pos)
```

```
 if(ruido)
   BaseADN -> Add_Ruido();

 return BaseADN
}
```

5.2 Bases Reales de ADN

Una base real de ADN representa un conjunto de regiones promotoras de genes en la cual se fija una proteína conocida como el factor de transcripción; las cadenas de nucleótidos donde se fijan los factores de transcripción son las palabras que forman el motivo buscado. Las bases reales se encuentran codificadas en archivos de texto en un formato común en bioinformática conocido como FASTA.

FASTA es un formato de representación mediante archivos de texto de secuencias de nucleótidos y péptidos. Cada aminoácido o nucleótido es representado utilizando símbolos correspondientes del alfabeto genético o aminoácido respectivamente. Este formato es considerado un estándar en el campo de la bioinformática debido a lo simple que resulta codificar la información en este formato, lo cual permite un procesamiento de los datos presentes en el fichero mucho más sencilla.

Una secuencia bajo formato FASTA comienza con un línea de cabecera que sirve para describir los datos contenidos en la siguiente secuencia de nucleótidos. Esta línea de descripción se distingue de los datos

de secuencia por el símbolo '>'. A continuación de este símbolo se añade una descripción de la secuencia, como el organismo del que proviene.

Una región promotora de la bacteria Escherichia Coli en formato FASTA tiene la siguiente forma:

>AB004306 |489-500:GAGGGTATAAAC|

tgattgtggctcaccctccatcactcccagggcccctggcccagcagccgcagctcccaaccacaatatcc
tttggggtttggcctacggagctggggcggatgacccccaaatagccctggcagattcccccctagacccgc
ccgcaccatggtcaggcatgcccctcctcatcgctggcacagcccaGAGGGTATAAAC

El código AB004306 representa un identificador único por cada región promotora en el archivo de texto. A continuación, se muestra la posición en la región promotora donde inicia y termina una palabra del motivo; además, se añade la cadena que se encuentra en ese intervalo de posiciones: GAGGGTATAAAC. Siguiendo con el ejemplo anterior, una base de regiones promotoras de la misma bacteria en formato FASTA tendría la siguiente forma:

Figura 19.- Base de regiones promotoras de la bacteria Escherichia Coli en formato FASTA

```
>1
taatgtttgtgctggtTTTTGTGGCATCGGGCGAGAATagcgcgtggtgtgaaagactgtTTTTTTGATCGTTTTCACAAAAatggaagtccacagtcttgacag
>2
gacaaaaacgcgtaacAAAAGTGTCTATAATCACGGCAgaaaagtccacattgaTTATTTGCACGGCGTCACACTTtgctatgccatagcatttttatccataag
>3
acaaatcccaataacttaattattgggatttgttatatataactttataaattcctaaaattacacaaagttaatAACTGTGAGCATGGTCATATTTTtatcaat
>4
cacaaagcgaaagctatgctaaaacagtcaggatgctacagtaatacattgatgtactgcatGTATGCAAAGGACGTCACATTAccgtgcagtacagttgatagc
>5
acggtgctacacttgtatgtagcgcatctttctttacggtcaatcagcaAGGTGTTAAATTGATCACGTTTtagaccatttttcgtcgtgaaactaaaaaaacc
>6
agtgaaTTATTTGAACCAGATCGCATTAcagtgatgcaaacttgtaagtagatttccttAATTGTGATGTGTATCGAAGTGtgttgcggagtagatgttagaata
>7
gcgcataaaaaacggctaaattcttgtgtaaacgattccacTAATTTATTCCATGTCACACTTttcgcatctttgttatgctatggttatttcataccataagcc
>8
gctccggcggggttttttgttatctgcaattcagtacaAAACGTGATCAACCCCTCAATTttccctttgctgaaaaattttccattgtctcccctgtaaagctgt
>9
aacgcaatTAATGTGAGTTAGCTCACTCATtaggcaccccaggctttacactttatgcttccggctcgtatgttgtgtggAATTGTGAGCGGATAACAATTTcac
>10
acattaccgccaaTTCTGTAACAGAGATCACACAAagcgacggtggggcgtaggggcaaggaggatggaaagaggttgccgtataaagaaactagagtccgttta
>11
ggaggaggcgggaggatgagaacacggcTTCTGTGAACTAAACCGAGGTCatgtaaggaaTTTCGTGATGTTGCTTGCAAAAatcgtggcgattttatgtgcgca
>12
gatcagcgtcgttttaggtgagttgttaataaagatttggAATTGTGACACAGTGCAAATTCagacacataaaaaaacgtcatcgcttgcattagaaaggtttct
>13
gctgacaaaaaagattaaacataccttatacaagacttttttttcatATGCCTGACGGAGTTCACACTTgtaagttttcaactacgttgtagactttacatcgcc
>14
ttttttaaacattaaaattcttacgtaatttataatctttaaaaaaagcatttaatattgctccccgaacGATTGTGATTCGATTCACATTTaaacaatttcaga
>15
cccatgagagtgaaatTGTTGTGATGTGGTTAACCCAAttagaattcgggattgacatgtcttaccaaaaggtagaacttatacgccatctcatccgatgcaagc
>16
ctggcttaactatgcggcatcagagcagattgtactgagagtgcaccatatgCGGTGTGAAATACCGCACAGATgcgtaaggagaaaataccgcatcaggcgctc
>17
CTGTGACGGAAGATCACTTCgcagaataaataaatcctggtgtccctgttgataccgggaagccctgggccaacttttggcgaAAATGAGACGTTGATCGGCACG
>18
gatttttatactttaacttgttgatatttaaaggtatttaattgtaataacgatactctggaaagtattgaaagttaATTTGTGAGTGGTCGCACATATCctgtt
```

Nótese que los motivos ya definidos se distinguen del resto de secuencias mediante letras mayúsculas.

Los archivos en formato FASTA pueden abrirse como archivos de texto, los que desean trabajar a un nivel más amplio sobre su contenido pueden utilizar programas como GeneStudio SeqVerter.

Las bases reales de ADN utilizadas en este trabajo fueron obtenidas vía Web mediante pruebas realizadas sobre las mismas bases reales de ADN utilizadas en otros trabajos [37][40][46]. Las bases reales de ADN se describen por el factor de transcripción que se fija en las secuencias del motivo buscadas. Las bases más ampliamente utilizadas para probar el desempeño en un entorno real de métodos de búsqueda de motivos son 8:

CRP, ERE, E2F, MYOD, MEF2, CREB, SRF y TBP. En este trabajo se utilizaron sólo las primeras 5 bases, debido a que en ellas el tamaño de las palabras del motivo es constante. En las otras bases, el motivo cambia de tamaño en diferentes regiones reguladoras, lo cual constituye una limitante a la búsqueda por parte de los métodos evolutivos desarrollados en este trabajo. Las bases reales de ADN utilizadas se encuentran en la siguiente tabla:

Tabla 12.- Bases de regiones promotoras de organismos biológicos utilizadas en los métodos evolutivos

Base	# Secuencias (T)	Tamaño Secuencias(bp)	l	N_t
CRP	18	105	22	23
ERE	25	200	13	25
E2F	25	200	11	27
MYOD	17	200	6	17
ME2F	17	199	7	21

La variable *l* representa el tamaño de las palabras del motivo en cada base de ADN; Nt representa el número total de palabras que conforman el motivo. Cada base de ADN se identifica en función del factor de transcripción asociado al motivo del organismo. Cada factor de transcripción cumple un papel específico en la producción de una determinada proteína. A continuación se muestra la funcionalidad de cada factor de transcripción:

CRP.-La proteína reguladora CRP (cAMP receptor protein) cumple funciones de regulación en la transcripción genética de algunas bacterias

como la *Escherichia Coli.* Los genes regulados por esta proteína están envueltos en la producción de la galactosa, un azúcar envuelto en el metabolismo de la energía en los seres vivos.

ERE.-Esta proteína también es conocida como receptor de estrógeno, cumple un papel importante en la prevención de la obesidad en ratones y seres humanos. Además es importante para evitar el envejecimiento prematuro en pequeños mamíferos como los ratones. En el caso del ser humano, una presencia anormal de esta proteína en las glándulas mamarias conlleva la aparición del cáncer de mama.

E2F.- E2F es un término que agrupa un conjunto de factores de transcripción presentes en eucariotas complejas. Dentro de E2F se encuentran tres proteínas que inician el proceso de transcripción celular, que son el E2F 1, 2 Y E2F3a. También se agrupan detrás de este término supresores como E2F3b, E2F4-8. Cumplen un papel fundamental en la síntesis de moléculas del ADN en las células de algunos mamíferos.

MYOD.- Esta proteína humana de diferenciación miogenética (Myogenetic regulartor factor) pertenece a la familia de proteínas conocidas como factores reguladores miogenéticos. Cumple un papel fundamental en la regulación y diferenciación del tejido muscular.

MEF2.- Esta proteína cumple un papel importante en la regulación de la diferenciación celular que ocurre en las etapas embrionarias de los

organismos vivos desde las bacterias de la levadura hasta en los seres humanos.

Una vez definidas las bases sobre las cuales se evalúan los algoritmos de búsqueda de motivos, es necesario definir métricas que permitan realizar una clasificación de métodos de búsqueda de motivos en base a la calidad de los resultados obtenidos.

5.3 Métricas de medición del desempeño de los métodos evolutivos de búsqueda de motivos

Las métricas de desempeño miden el grado de éxito de un algoritmo de búsqueda en función de la relevancia del conjunto de información obtenida a partir de los requerimientos de búsqueda. Dos métricas de desempeño muy comúnmente utilizadas son la Precisión y Exhaustividad, también conocidas en el idioma inglés como Precision y Recall.

Ambas métricas provienen de un campo del conocimiento conocido como Recuperación de Información o Information Retrieval (IR) [41]. La Recuperación de Información agrupa un conjunto de métodos enfocados en obtener información a partir un universo de datos que no presentan ningún tipo de estructura.

Las métricas de Precisión y Exhaustividad miden la calidad de los resultados de la búsqueda a partir de un conjunto de datos. La Precisión mide la habilidad de un motor de búsqueda en encontrar sólo información

relevante a los parámetros de búsqueda. La Exhaustividad mide la habilidad de un motor de búsqueda en presentar el mayor número posible de información relevante existente.

Estas métricas son muy utilizadas para medir la calidad de resultados en buscadores Web. Para entender los conceptos detrás de estas métricas, tómese como ejemplo un buscador web. A partir de una cadena de palabras, el buscador devuelve un conjunto de información en forma de documentos. A partir del conjunto de documentos encontrados, la persona que realiza la búsqueda requiere conocer 2 cosas: 1) cuántos de los documentos encontrados satisfacen los criterios de búsqueda, y 2) si todos los documentos relevantes se encuentran dentro del conjunto de información devuelta.

Estos dos requerimientos son determinados mediante la Precisión y Exhaustividad. La Precisión mide cuánto de la información obtenida es relevante; la Exhaustividad determina cuánta de la información relevante se ha obtenido con respecto al total de información devuelta. Utilizando una notación matemática, la Precisión y la Exhaustividad se definen de la siguiente forma:

$$\boldsymbol{Precision} = \frac{|S_{rel} \cap S_{ret}|}{|S_{ret}|}$$

Donde S_{ret} denota el conjunto de documentos obtenidos y S_{rel} el conjunto de documentos relevantes con respecto al texto de búsqueda. La Exhaustividad tendría la siguiente fórmula:

$$\boldsymbol{Exhaustividad} = \frac{|S_{rel} \cap S_{ret}|}{|S_{rel}|}$$

Los conjuntos S_{rel} y S_{ret} pueden contener documentos, imágenes, cadenas, cualquier clase de representación de la información con referencia al desempeño de un algoritmo que se desee medir. Los valores que toman las métricas de Precisión y Exhaustividad se encuentran en el rango de [0,1].

Un valor de precisión de 0 implica que ningún elemento del conjunto de datos obtenido de la búsqueda es relevante con respecto al criterio de búsqueda. Por otro lado, si la precisión de un método de búsqueda es 1, implica que todos los elementos del conjunto obtenido son relevantes. Una exhaustividad de valor 0 significa que ninguno de los elementos relevantes se encuentra en el conjunto de información recuperado. Por otro lado, si la exhaustividad es igual a 1, todos los documentos relevantes que existen se encuentran en el conjunto recuperado de la búsqueda. Esto no implica, sin embargo, que todos los elementos del conjunto recuperado sean relevantes.

Medir la Precisión es sencillo, basta determinar la cardinalidad de la intersección del conjunto de información relevante con el conjunto correspondiente a la información encontrada. La Exhaustividad, por otro lado, requiere conocer la cardinalidad del conjunto que contenga toda la

información relevante existente. Esto, dependiendo del problema que se pretenda resolver, puede resultar tan complejo como resolver el problema en sí.

Las fórmulas de Precisión y Exhaustividad permiten describir una relación inversa entre ambas métricas, relación que se cumple sólo en ciertas circunstancias. Por ejemplo, aumentando el tamaño de S_{ret} aumenta la probabilidad de que en el conjunto de información recuperada haya un mayor número de elementos relevantes, con lo que aumenta la exhaustividad de la búsqueda. Esto, sin embargo, provoca la disminución de la Precisión del método de búsqueda, pues la diferencia entre la información recuperada y la información relevante aumenta.

Hay otra forma en la cual ambas métricas pueden mejorar sus resultados sin que esto conlleve una relación inversa entre ellas: aumentar el tamaño del conjunto de información relevante o S_{rel}. Esto implica una mejora en la calidad de los resultados, lo cual puede ser consecuencia de aplicar herramientas de la inteligencia computacional en métodos tradicionales de búsqueda, brindando una mejoría cualitativa en los resultados obtenidos y permitiendo distinguir en base a los resultados la eficiencia entre estrategias de búsqueda.

Las métricas de Precisión y Exhaustividad son utilizados como métodos de evaluación del desempeño de los métodos de búsqueda de

motivos. En este trabajo utilizamos la definición de Precisión y Exhaustividad utilizada en [37], la cual se expresa de la siguiente manera:

$$Precision = \frac{N_c}{N_p}$$

$$Exhaustividad = \frac{N_c}{N_t}$$

Donde N_c es el número de palabras del motivo encontradas por el algoritmo, N_p es el total de palabras que conforman el motivo, y N_t es el total de palabras de motivo encontradas de forma experimental. En este trabajo se supone que sólo existe 1 palabra del motivo en cada secuencia de ADN, por ende N_p será siempre igual al número de secuencias en la base de ADN. El valor de N_t es encontrado de forma experimental y es una constante definida por la base real de ADN.

CAPITULO 6

Resultados de los métodos evolutivos en la búsqueda de Motivos

En este capítulo se presentan los resultados de la aplicación de los métodos desarrollados en este trabajo tanto a las bases de ADN sintéticas como a las reales. Los resultados se muestran mediante tablas que presentan las métricas precisión y exhaustividad obtenidas para cada método sobre 12 bases sintéticas y 5 bases reales de ADN.

Además se presentan gráficas para mostrar la convergencia de los métodos aquí desarrollados (dos que utilizan algoritmos genéticos y uno que aplica estimación de distribuciones) tanto en el caso de búsqueda sobre bases de ADN sintéticas como reales.

También se realiza una comparación del desempeño de estos métodos entre sí, y con los de otros que han sido encontrados en la

literatura. Esto permitirá conocer las fortalezas y debilidades de los métodos desarrollados en este trabajo, así como también proponer actividades futuras para mejorarlos.

Finalmente, se incluye una representación gráfica en forma de *logos de secuencia* para los motivos encontrados en tres bases de ADN reales, y se los compara con los *logos* correspondientes para los motivos verdaderos, que han sido obtenidos experimentalmente y se los conoce.

6.1 Resultados de la aplicación de métodos de búsqueda de motivos basados en AG para el problema de la búsqueda de motivos

6.1.1 Método 1 basado en AG

El método 1 representa a un individuo como una palabra de nucleótidos y pertenece al espacio de búsqueda 4^l, donde l es el tamaño de la palabra motivo. Esta solución se probó sobre bases sintéticas y bases reales. La siguiente tabla muestra los resultados basados en las métricas de Precisión y Exhaustividad sobre 12 bases sintéticas.

Tabla 13.- Resultados de la búsqueda del motivo en bases sintéticas del método 1 basado en AG

Numero de Secuencias	Tamaño motivo	Conservación	Ruido			
			Sin Ruido		Con Ruido	
			Precision (%)	Exh. (%)	Precision (%)	Exh. (%)
100	16	1	0.99	0.99	0.95	0.85
20	16	1	0.94	0.94	0.90	0.81
100	8	1	0.99	0.99	0.95	0.86
20	8	1	0.95	0.95	0.91	0.82
100	16	0	0.89	0.89	0.84	0.763
20	16	0	0.59	0.59	0.55	0.49

Como se aprecia en la tabla anterior, los resultados fueron excelentes en la búsqueda de un motivo artificial sobre bases sintéticas de ADN. Dentro de la tabla se muestran resultados diferentes sobre bases sintéticas con ruido y sin ruido, siendo mejores los resultados sobre bases sintéticas sin ruido debido a que en todas las secuencias está presente sólo una palabra del motivo.

Dentro de las bases sintéticas existentes, los mejores resultados fueron sobre las bases con un mayor número de secuencias y cuyo motivo presentaba conservación a través de las secuencias. Esto se debe a que un número mayor de secuencias implica una redundancia mayor en las palabras del motivo y por eso aumenta la probabilidad de encontrar el motivo buscado.

Los buenos resultados presentes en las bases sintéticas llevaron a probar la búsqueda de motivos sobre bases reales de regiones promotoras de ADN. Para ello, utilizamos las bases reales de ADN que se encuentran en esta tabla, mostrada en el capítulo 5:

Tabla 14.- Bases reales de ADN utilizados para probar los métodos evolutivos

Base	# Secuencias	longitud Secuencia(bp)	Tamaño motivo	N_t
CRP	18	105	22	23
ERE	25	200	13	25
E2F	25	200	11	27
MYOD	17	200	6	17
ME2F	17	199	7	21

Los resultados de la búsqueda sobre estas bases reales fueron pobres, como se observa en la siguiente tabla:

Tabla 15.- Resultados de la búsqueda de motivos del método 1 basado en AG sobre las bases reales de ADN

Base	Precision	Exhaustividad
CRP	0.30	0.20
ERE	0.15	0.05
E2F	0	0
MYOD	0.1	0.01
ME2F	0.44	0.44

Los resultados obtenidos en la búsqueda de motivos en las bases reales de ADN llevaron a realizar un análisis de las palabras del motivo en las bases reales. Este análisis permitió llegar a las siguientes observaciones:

1. Los patrones presentes en las palabras del motivo en las bases reales no siguen criterios de conservación presentes en las bases sintéticas. Esto añade complejidad al problema de búsqueda de motivos, lo cual hace que el enfoque presente en el método 1 sea insuficiente para resolver el problema.
2. La distancia de Hamming entre una palabra del conjunto de búsqueda 4^l no es una función de evaluación apropiada para bases reales, pues las palabras del motivo son muy diferentes a través de las secuencias en la base de ADN; por ende, no es prudente utilizar una palabra de longitud l como criterio fundamental para buscar las palabras del motivo

Por estas razones, se realizó una investigación para encontrar formas más apropiadas en la representación de individuos, en la elección de una función de evaluación más precisa y en mecanismos que permitan evitar una rápida convergencia en soluciones locales. Como resultado, se desarrolló el método 2 basado en algoritmos genéticos.

6.1.2 Método 2 basado en AG (MBMAG)

El método 2 está basado en la representación de un individuo como un vector de posiciones iniciales; su espacio de búsqueda tiene tamaño $(n-l+1)^t$, donde n es el tamaño de una secuencia en la base de ADN, l el tamaño de las palabras del motivo y t el número de secuencias de regiones promotoras en la base de ADN. La función de fitness utilizada es el contenido de información de las palabras del motivo definidas por el vector de posiciones iniciales. Además, se utilizaron funciones como el Filtrado local y Desplazamiento, que evitaban la convergencia prematura del método en soluciones óptimas a nivel local.

Los resultados obtenidos en las bases sintéticas pueden verse en la siguiente tabla:

Tabla 16.- Resultados de la búsqueda de motivos sobre bases sintéticas del método 2 basado en AG (MBMAG)

Numero de Secuencias	Tamaño motivo	Conservación	Ruido			
			Sin Ruido		Con Ruido	
			Precision (%)	Exh. (%)	Precision (%)	Exh. (%)
100	16	1	1	1	0.99	0.97
20	16	1	0.99	0.99	0.98	0.97
100	8	1	0.99	0.99	0.97	0.93
20	8	1	0.98	0.98	0.91	0.89
100	16	0	0.95	0.95	0.88	0.80
20	16	0	0.94	0.94	0.89	0.85

Los resultados obtenidos fueron mejores que los del método 1, sobre todo en bases sintéticas con presencia de ruido. Esto brinda la confianza en obtener mejores resultados en la búsqueda de motivos sobre bases reales de ADN. Estos resultados se pueden observar en la siguiente tabla

Tabla 17.- Resultados de la búsqueda de motivos sobre bases reales del método 2 basado en AG (MBMAG)

Base	Precision	Exhaustividad
CRP	0.88	0.69
ERE	0.76	0.76
E2F	0.76	0.70
MYOD	0.94	0.76
ME2F	0.94	0.94

Los mejores resultados del método 2 basado en AG son en las bases MYOD y ME2F; ambas bases tienen el mismo número de regiones promotoras junto con la base CRP. A pesar que el tamaño de las palabras del motivo es diferente entre la base CRP y las bases MYOD y ME2F el resultado en la precisión es muy similar. La precisión del MBMAG disminuye en bases con un mayor número de regiones promotoras como ERE y E2F. Esto permite concluir que el MBMAG es más preciso que el método 1 basado en AG.

6.2 Resultados de la aplicación del método MBMEDA para el problema de la búsqueda de motivos

Los métodos basados en ED son métodos muy parecidos a los métodos basados en AG, pues utilizan una representación de individuos y funciones de evaluación y selección muy similares; sin embargo, al momento de generar una nueva población evitan el uso de operadores genéticos. Los métodos basados en ED estiman un modelo probabilístico a partir de los individuos de la población seleccionados y muestrean nuevos individuos de la siguiente generación a partir del modelo. En base a este modelo de búsqueda se desarrollaron 2 métodos en los cuales el espacio de soluciones es igual al espacio de búsqueda del método 2 basado en AG, el cual tiene un tamaño de $(n-l+1)^t$.

El método funcional basado en ED (MBMEDA) supone la independencia entre las variables de las palabras del motivo. Esto conlleva que en vez de tener un modelo normal de varias variables, se trabaje con 4 modelos Gaussianos de una variable. Cada modelo representa la distribución de un símbolo del alfabeto genético (A, C, G, T) sobre el conjunto de individuos seleccionados de la población. El muestreo de nuevos individuos a partir de un modelo normal univariado está basado en los estimadores estándares de la media y la varianza; la función de muestreo de un nuevo individuo está dada por:

$$I_b = \mu_b + Z * \sigma_b^2$$

Donde, como se explica con mayor detalle en la sección 4.4 I_b representa el componente de la matriz de pesos posicionales (MPP) del individuo en el nucleótido b, μ_b representa la media de la distribución normal basada en el nucleótido b y, σ_b^2 la varianza de la distribución normal basada en el mismo nucleótido.

El método MBMEDA obtuvo los siguientes resultados sobre las bases sintéticas:

Tabla 18.- Resultados de la búsqueda de motivos sobre bases sintéticas del método MBMEDA

Numero de Secuencias	Tamaño motivo	Conservacion	Ruido			
			Sin Ruido		Con Ruido	
			Pr. (%)	Ex. (%)	Pr. (%)	Ex. (%)
100	16	1	1	1	0.99	0.97
20	16	1	0.99	0.99	0.98	0.95
100	8	1	1	1	0.99	0.93
20	8	1	0.99	0.99	0.98	0.92
100	16	0	0.97	0.97	0.84	0.79
20	16	0	0.95	0.95	0.93	0.86

Los resultados fueron excelentes, mejores incluso que los obtenidos en el MBMAG. Por ello, se probó la búsqueda de motivos en bases reales de ADN; los resultados se muestran a continuación:

Tabla 19.- Resultados de la búsqueda de motivos sobre bases reales del método MBMEDA

Base	l	T	N_t	Precisión	Exhaustividad
CRP	22	18	23	0.83	0.65
ERE	9	25	25	0.8	0.8
E2F	11	25	27	0.80	0.74
MYOD	6	17	21	1	0.80
ME2F	7	17	17	1.00	1.00

Como se aprecia en esta tabla, el MBMEDA obtiene excelentes resultados en la búsqueda de motivos en bases reales de ADN. En las bases con motivos de pequeño tamaño, el MBMEDA tiene un promedio de precisión de 0.93 y exhaustividad de 0.86. Estos resultados permiten concluir que el método basado en ED encuentra de forma mas precisa y completa los motivos en las bases reales que el MBMAG. A continuación se muestran tablas donde se comparan los resultados de desempeño de los métodos evolutivos desarrollados junto con otros métodos computacionales de búsqueda de motivos.

6.3 Demostración de la convergencia de los métodos de búsqueda de motivos basados en ED y AG.

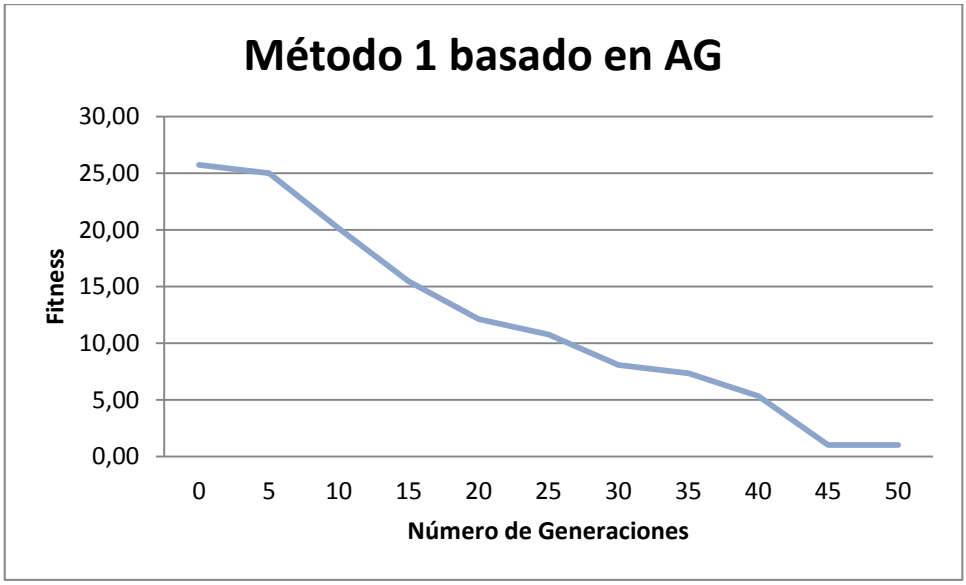
A fin de comprobar la convergencia de los métodos evolutivos implementados se realizó un experimento sobre una base sintética y una base real de ADN. El experimento consiste ejecutar una sola vez cada método desarrollado para cada base de ADN y graficar la evolución de la

función de fitness a través de las generaciones sucesivas dentro de un solo ciclo evolutivo.

6.3.1 Convergencia de los métodos evolutivos sobre bases sintéticas

La base sintética utilizada en este experimento se la conoce como 100-16-1-0. Esto quiere decir es una base de ADN de 100 secuencias con un motivo con un patrón de tamaño 16, donde todas las palabras se conservan y no existe presencia de ruido. La siguiente figura muestra la convergencia del método 1 basado en AG

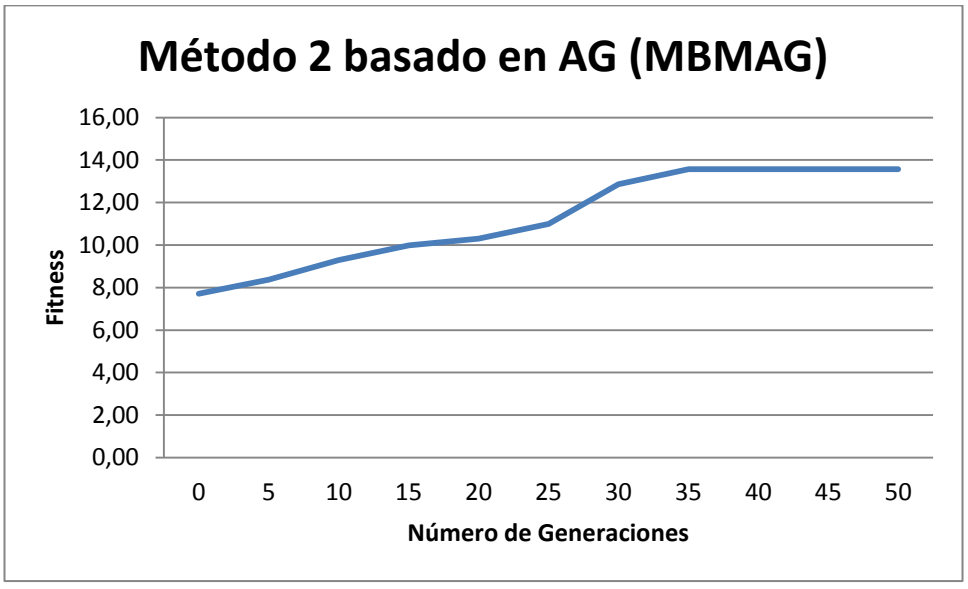
Figura 20.- Evolución del fitness del mejor individuos en el método 1 basado en AG sobre la base sintética 100-16-1-0



El método 1 converge a partir de la generación 45 a un valor de fitness de 1,01. Tómese en cuenta que la función de fitness utilizada en este método es la distancia de Hamming, por lo que el objetivo del método es encontrar la palabra de motivo con distancia mínima a la base de ADN.

La siguiente figura muestra la convergencia del método 2 basado en AG.

Figura 21.- Evolución del fitness del mejor individuo en el método 2 basado en AG (MBMAG) sobre la base sintética 100-16-1-0

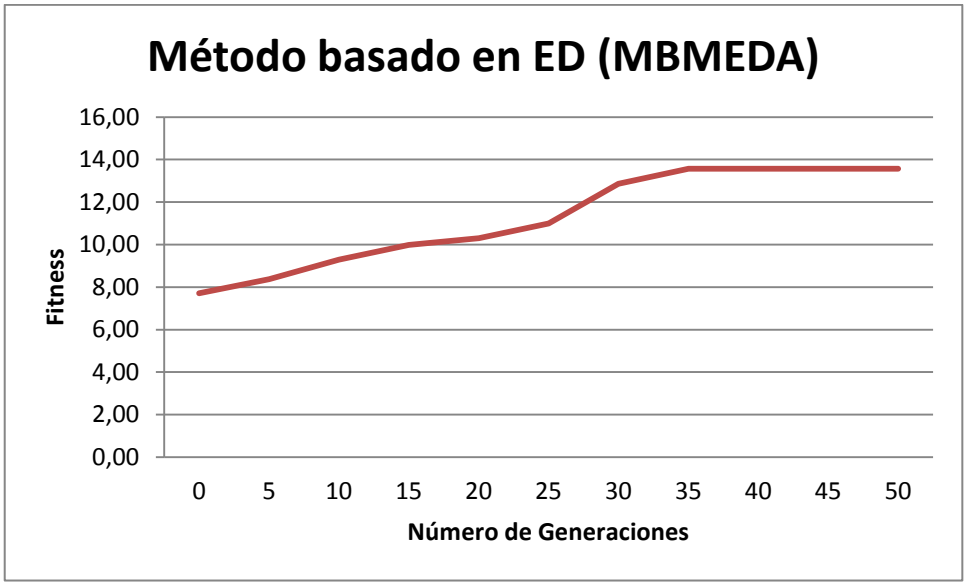


El método 2 tiene a la función de fitness como función de evaluación de los individuos de la población. En este caso, se busca al individuo cuyo contenido de información sea máximo. Es por ello que desde la generación

35 el método 2 basado en AG converge a un valor máximo de fitness de 13.57.

La siguiente figura muestra la convergencia del método basado en ED sobre la base sintética:

Figura 22.- Evolución del fitness del mejor individuo en el método 2 basado en ED (MBMEDA) sobre la base sintética 100-16-1-0



El método basado en ED converge a un valor de fitness de 13.57 desde la generación número 35. Su velocidad de convergencia es igual a la del método 2 basado en los algoritmos genéticos.

En este experimento realizado sobre una base sintética, resulta importante destacar que los métodos que utilizan el IC como función de fitness convergen más rápido, y como se verá más adelante, presentan

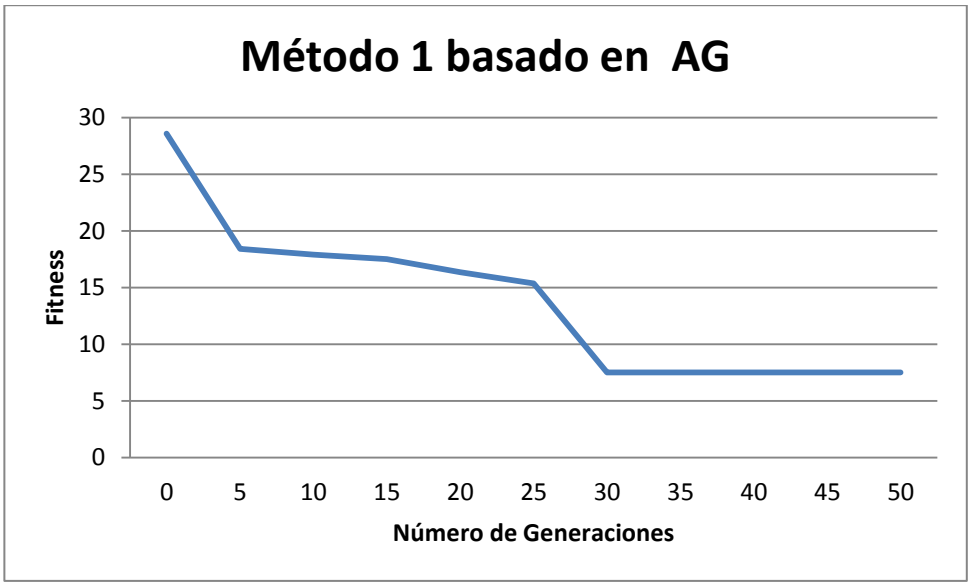
mejores resultados que el método que utiliza la distancia de Hamming para elegir el mejor individuo.

6.3.2 Convergencia de los métodos evolutivos sobre bases reales

La base real utilizada en este experimento es la base CRP. Las regiones promotoras correspondientes a esta base pertenecen a la bacteria Escherichia Coli. La base tiene 18 regiones promotoras de 105 bp de longitud. El motivo esta compuesto por 23 palabras de tamaño l= 22.

La siguiente tabla muestra la convergencia del método 1 basado en AG:

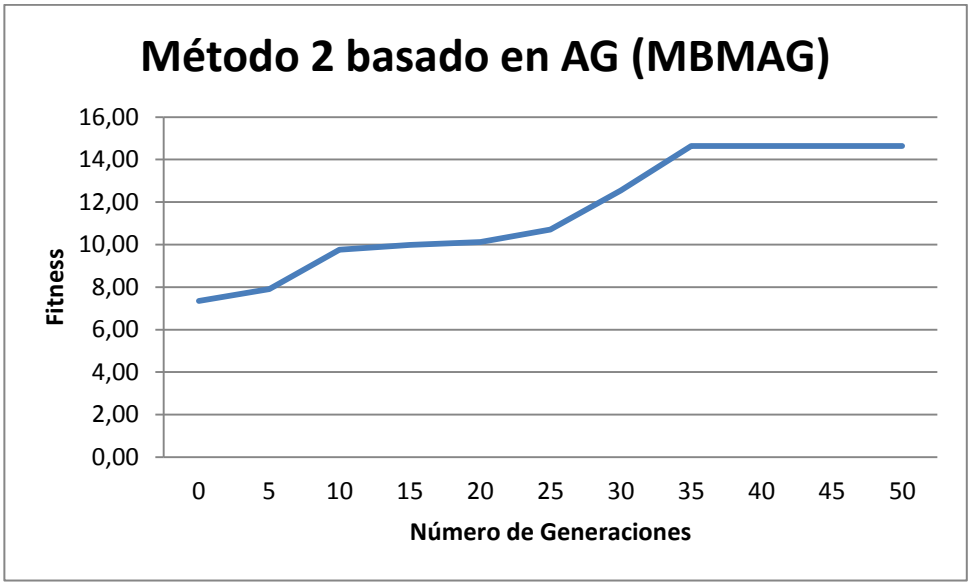
Figura 23.- Evolución del fitness del mejor individuo en el método 1 basado en AG sobre la base CRP



El método 1 converge al valor de 7,51 a partir de la generación 40. Como se mostrará más adelante, los valores medidos por la precisión y exhaustividad de este método dejan mucho que desear.

La siguiente figura muestra la convergencia del método 2 basado en AG

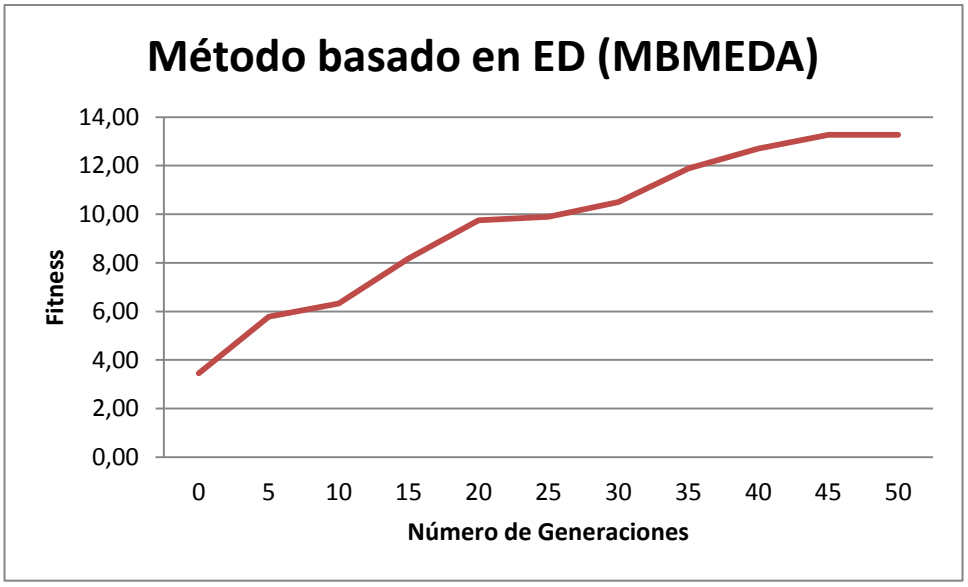
Figura 24.- Evolución del fitness del mejor individuo en el método 2 basado en AG (MBMAG) sobre la base CRP



El método 2 converge a un valor de fitness de 14,64 desde la generación 35. Aunque demora más tiempo en converger que el método 1, el resultado en precisión y exhaustividad es superior al del método 1 basado en AG.

La siguiente figura muestra la convergencia del Método 2 basado en ED sobre la base real CRP.

Figura 25.- Evolución del fitness del mejor individuo en el método basado en ED (MBMEDA) sobre la base CRP



El método 2 basado en ED convergió al valor de fitness de 13.68 desde la generación 45. La velocidad de convergencia es menor a los métodos basados en AG, y para el caso particular la base CRP, los resultados son inferiores al método 2 basado en AG.

El método 1 basado en AG converge antes que los demás métodos a su mejor solución, sin embargo, presentan pobres resultados en precisión y exhaustividad; es por ello que la velocidad de convergencia no es un criterio confiable con respecto a la calidad de los resultados.

El método 2 basado en AG tiene una menor velocidad de convergencia con respecto al método 1, sin embargo, presenta los mejores

resultados en precisión y exhaustividad en la base CRP que los demás métodos desarrollados. El método 1 basado en AG demora un promedio de 1 hora en ejecutarse, eso lo hace el método más rápido de búsqueda de motivos entre los métodos evolutivos desarrollados. El método 2 basado en AG tiene un tiempo promedio de ejecución entre 3 y 4 horas. El método basado en ED es el que más tiempo demora en ejecutarse, un promedio de 9 a 15 horas. Esto se debe a las operaciones adicionales necesarias en este método.

Un resultado como consecuencia de estos experimentos fue comprobar que los métodos basados en AG y en ED siempre convergen a una solución independientemente de si ésta es la solución correcta al problema.

6.4 Comparación de los resultados de los métodos evolutivos desarrollados con otros métodos de búsqueda de motivos

Una comparación en los resultados basados en las métricas de desempeño entre los métodos MBMAG y MBMEDA arroja los siguientes resultados:

Tabla 20.- Comparación de resultados de la búsqueda de motivos entre los métodos evolutivos desarrollados

				MBMEDA		MBMAG	
Base	**l**	**T**	**N_t**	**Pr.**	**Ex.**	**Pr.**	**Ex.**
CRP	22	18	23	0.83	0.65	**0.88**	**0.69**
E2F	11	25	27	**0.80**	**0.74**	0.76	0.70
ERE	9	25	25	**0.8**	**0.8**	0.76	0.76
ME2F	7	17	17	**1.00**	**1.00**	0.94	0.94
MYOD	6	17	21	**1.00**	**0.80**	0.94	0.76

La tabla demuestra que en 4 de las 5 base reales de ADN el método basado en ED presenta mejores resultados que el método 2 basado en AG. Esto permite concluir que los métodos basados en la estimación de distribuciones presentan mejores resultados en encontrar las instancias de un motivo que los métodos basados en algoritmos genéticos. Esto no quiere decir que los resultados de los algoritmos genéticos sean malos, sólo que el método basado en ED resulta ligeramente superior.

Existen otros métodos de búsqueda de motivos desarrollados en base a los AG y por ED. Estos se conocen por sus siglas GAME [46]

(*Genetic Algorithm for Motif Elicitacion*), GALF [37] (*Genetic Algorithm with local filtering*) y EDAMD [40] (*Estimation of distribution algorithm for motif discovery*). La siguiente tabla muestra los resultados del método 2 basado en AG con respecto a los métodos GAME y GALF

Tabla 21.- Comparación de los resultados en la búsqueda de motivos en bases reales de 3 métodos evolutivos basados en los algoritmos genéticos

				MBMAG		GAME		GALF	
Base	**l**	**T**	**N_t**	**Pr.**	**Ex.**	**Pr.**	**Ex.**	**Pr.**	**Ex.**
CRP	22	18	23	0.88	0.69	0.94	0.70	**0.94**	**0.74**
ERE	9	25	25	0.76	0.76	0.73	0.76	**0.84**	**0.84**
E2F	11	25	27	0.76	0.70	**0.96**	**0.85**	0.80	0.74
MYOD	6	17	21	**0.94**	**0.94**	0.48	0.48	0.88	0.71
ME2F	9	17	17	0.94	0.94	0.88	0.88	**1.00**	**1.00**

Los métodos GAME y GALF incluyen un procesamiento posterior de los mejores individuos una vez terminado el proceso evolutivo; esta operación es conocida como *post-processing* . Esta operación se encarga de buscar palabras del motivo adicionales que hayan sido pasadas por alto en el proceso evolutivo. Esto influye de forma positiva en los resultados obtenidos por estos métodos, de allí la razón por la que el método basado en AG desarrollado en este trabajo presente resultados inferiores a la de estos dos métodos, sin embargo, se encuentran en promedios cercanos a los de estos 2 métodos de búsqueda de motivos basados en AG

La siguiente tabla muestra una comparación de resultados entre el método desarrollado basado en ED y EDAMD:

Tabla 21.- Comparación de los resultados en la búsqueda de motivos en bases reales de 2 métodos evolutivos basados en los algoritmos por estimación de distribuciones

				EDA		EDAMD	
Base	**l**	T	**N_t**	**Pr.**	**Ex.**	**Pr.**	**Ex.**
CRP	22	18	23	0.83	0.65	**0.94**	**0.74**
ERE	9	25	25	**0.80**	**0.80**	0.76	0.76
E2F	11	25	27	**0.80**	0.74	0.71	**0.80**
MYOD	6	17	21	**1.00**	**0.80**	0.86	0.9
ME2F	9	17	17	1.00	1.00	1.00	1.00

El método MBMAG presenta una mayor precisión que el método EDAMD; sin embargo, la exhaustividad del método EDAMD es superior. Esto se debe a que el método EDAMD estima un modelo multivariado, lo que le permite expresar explícitamente la interrelación entre las variables del problema, mejorando las probabilidades de encontrar un mayor número de palabras correctas del motivo. EDAMD incluye asimismo una operación de procesamiento posterior al proceso evolutivo. A pesar de estas ventajas, los resultados del método MBMEDA presenta resultados similares o superiores en ciertas bases de ADN.

6.4.1 Comparación de los resultados de los métodos evolutivos con métodos no evolutivos

La siguiente tabla muestra una comparación entre los resultados de los métodos desarrollados en este trabajo y 2 métodos de búsqueda de motivos ampliamente utilizados, MEME [17] y BioProspector [7]. Los

resultados se miden en base a los valores de las métricas de Precisión y Exhaustividad definidas anteriormente. La variable *l* representa el tamaño de las palabras del motivo y la variable T representa el número de secuencias en la base de ADN.

Tabla 22.- Comparación de los resultados en la búsqueda de motivos en bases reales los métodos MBMAG y MBMEDA con métodos probabilísticos de búsqueda de motivos

			MBMEDA		**MBMAG**		**MEME**		**BioProspector**	
Base	**l**	**T**	**Pr.**	**Ex.**	**Pr.**	**Ex.**	**Pr.**	**Ex.**	**Pr.**	**Ex.**
CRP	22	18	0.83	0.65	0.88	**0.69**	0.92	0.52	**1.00**	0.35
E2F	11	25	**0.80**	**0.74**	0.76	0.70	0.80	0.70	0.52	0.41
ERE	9	25	0.80	**0.80**	0.76	0.76	**0.88**	0.60	0.46	0.56
ME2F	9	17	**1.00**	**1.00**	0.94	0.94	0.93	0.82	0.71	0.71
MYOD	6	17	**1**	**0.80**	0.94	0.76	0.00	0.00	0.00	0.00

Como se muestra en la tabla anterior, en 4 de las 5 bases, los resultados del método MBMEDA presentan mejores resultados en Precisión y Exhaustividad que los métodos MEME y BioProspector. En la base CRP, Bioprospector tiene una Precisión de 1, sin embargo, el valor en base a la métrica de exhaustividad es inferior al resto de métodos. Esto se debe a que Bioprospector supone que sólo existen 13 instancias del motivo en la base CRP. Esto conlleva a que aunque encuentre 13 palabras del motivo correctas, su exhaustividad resulta ser insuficiente.

Dados estos resultados, es posible afirmar que los métodos basados en la computación evolutiva presentan mejores resultados tanto en precisión

y exhaustividad que otros métodos de búsqueda existentes. Dentro de la rama de la Computación Evolutiva, el método basado en ED presentó los mejores resultados.

6.5 Representación gráfica de los motivos encontrados utilizando logos de secuencias

Un logo de secuencias [43] es una representación gráfica de la conservación presente en un conjunto de secuencias de nucleótidos o de aminoácidos. Esta representación está basada en el alineamiento múltiple de las secuencias involucradas.

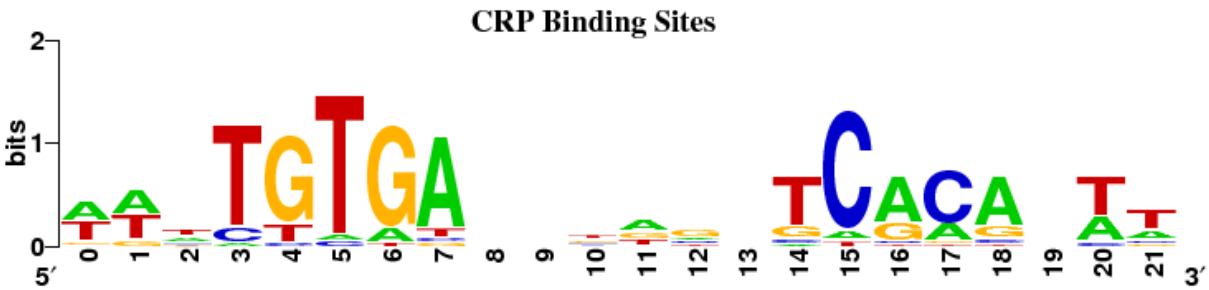
El logo de secuencias (*sequence logo*) ordena los nucleótidos de las palabras en pilas, y cada pila contiene los nucleótidos presentes en la columna correspondiente en la alineación de las palabras. La altura de la pila está dada por la frecuencia con que se repiten los símbolos presentes en cada columna; mientras mayor sea la altura de un símbolo en una pila, mayor es la frecuencia de ese nucleótido en la columna correspondiente.

El logo de secuencias es utilizado por algunos métodos de búsqueda de motivos [45] como una alternativa a la representación basada en la palabra consenso, debido a que un logo de secuencias aporta información más precisa y es más fácil comparar los resultados obtenidos con los resultados experimentales.

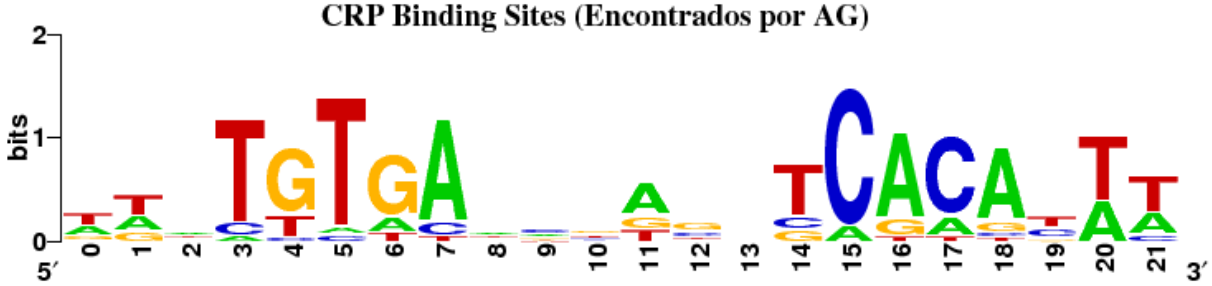
Los logos de secuencias pueden ser generados mediante una herramienta desarrollada por la universidad de Berkeley conocida como WebLogo. Esta herramienta web genera el logo de secuencias a partir de un conjunto de palabras de nucleótidos provistas.

En nuestro trabajo utilizamos el logo de secuencias como un criterio adicional para comparar de forma visual los motivos obtenidos en base a la ejecución de los métodos evolutivos desarrollados. Por cuestiones de tiempo, sólo presentamos los resultados en forma de logos de secuencia de 3 bases de ADN: las bases CRP MYOD y ME2F. Para cada base se obtuvo el logo de secuencias a partir de las palabras correctas del motivo, luego se obtuvo el logo de secuencias de las palabras del motivo encontradas por los métodos MBMEDA y MBMAG. A continuación se presentan las 3 figuras correspondientes para cada base de ADN.

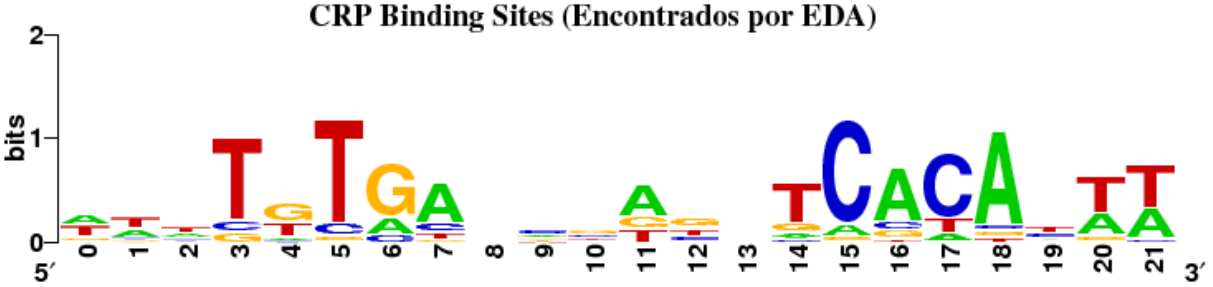
Figura 26.- Figuras de los logos de secuencia de las palabras del motivo encontradas de forma experimental y los logos de secuencia de los resultados obtenidos por los métodos evolutivos MBMAG y MBMEDA sobre la bases CRP, MYOD y ME2F



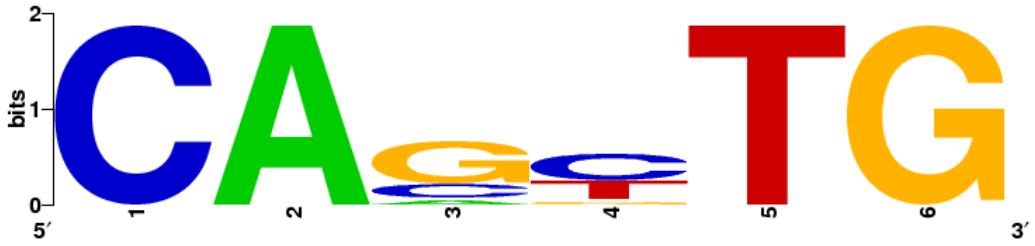
CRP Binding Sites (Encontrados por AG)



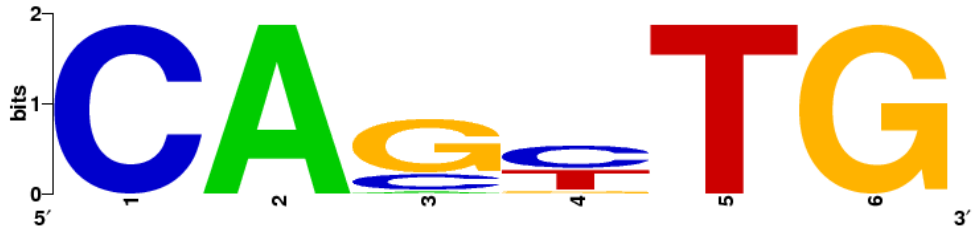
CRP Binding Sites (Encontrados por EDA)



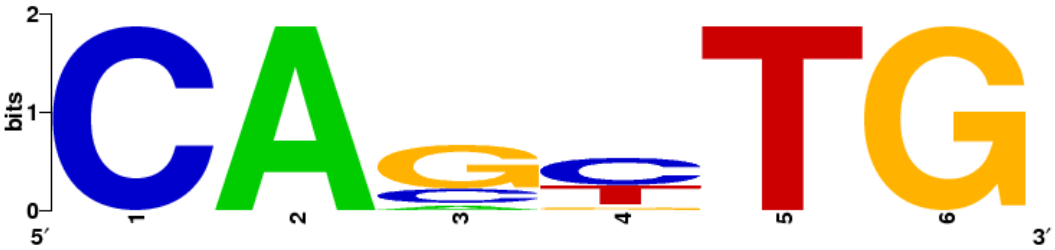
MYOD Binding Sites



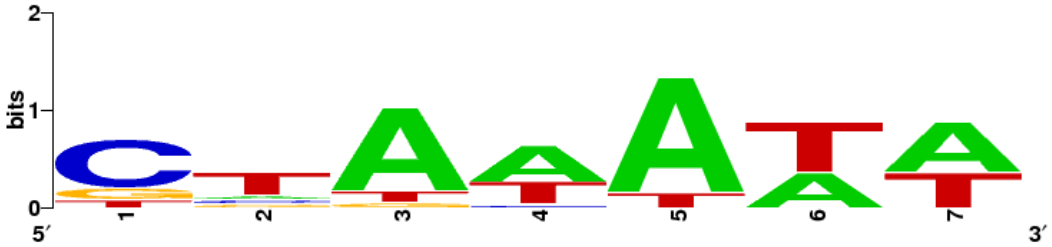
MYOD Binding Sites (Encontrado por AG)



MYOD Binding Sites (Encontrado por EDA)

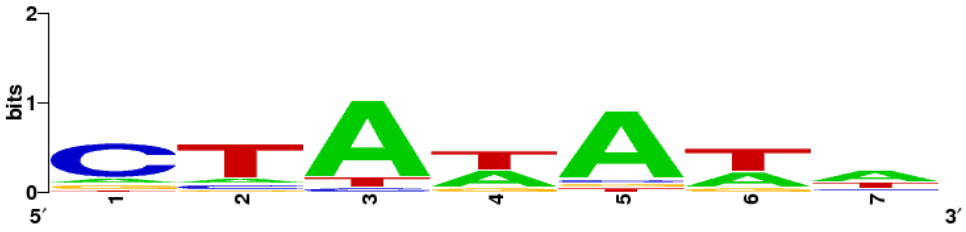


ME2F Binding sites



ME2F Binding sites (Encontrado por AG)

ME2F Binding sites (Encontrado por EDA)



En la base CRP se puede observar que el logo de secuencia de verdaderas palabras del motivo es más similar al logo de secuencias de las palabras encontradas por el método 2 basado en AG que al logo de secuencias de las palabras encontradas por el método basado en ED.

En la base MYOD los 3 logos de secuencia son muy similares, a pesar que la precisión y exhaustividad del método basado en ED es superior al método basado en AG.

En la base ME2F se puede observar que el logo de secuencia de las palabras encontradas por el método basado en ED es más similar al logo de secuencia de las palabras reales del motivo que el logo de secuencia obtenido por el método basado en AG.

Esta comparación mediante logos de secuencia de los resultados de la búsqueda del motivo en bases reales de ADN permite concluir que los motivos encontrados por los métodos evolutivos desarrollados guardan bastante similitud con los motivos reales en las bases de ADN

Conclusiones y Recomendaciones

Conclusiones

Sin duda, la identificación de los motivos que regulan la síntesis de las proteínas constituye un problema importante en el campo de la biología molecular debido a sus potenciales aplicaciones en el control de la expresión genética. Además, este problema es un verdadero desafío, puesto que del motivo o patrón que se busca se desconocen a priori detalles importantes: su tamaño, ubicación, constitución; y, más aun, que algunos de sus nucleótidos pueden mutar de una instancia a otra, y por lo tanto no siempre es el mismo.

La clave para identificar un motivo está en disponer de un conjunto de regiones promotoras de genes corregulados por el mismo factor de transcripción, donde dicho patrón entonces se repite muchas veces. Con base en este concepto fue posible modelar el problema como uno de optimización combinatoria, donde se busca aquel patrón de la base de ADN que se repite con mayor frecuencia. A partir de este modelo se

implementaron dos métodos evolutivos para resolver el problema: MBMAG que utiliza algoritmos genéticos, y MBMEDA que aplica algoritmos por estimación de distribuciones. Ambos métodos realizan una búsqueda global con base en el concepto de poblaciones, y evalúan la aptitud de sus individuos como soluciones correctas del problema mediante el concepto contenido de información, que mide la diferencia entre la distribución de nucleótidos en el patrón y la que estos tienen en toda la base de ADN, correspondiendo el motivo que se busca al patrón con la mayor diferencia o contenido de información.

Al aplicar estos métodos a la búsqueda de motivos sobre bases de ADN sintéticas y reales se obtuvieron los siguientes resultados:

1. Sobre bases sintéticas de ADN, los métodos evolutivos MBMAG y MBMEDA lograron una precisión y exhaustividad promedios de 0.94 y 0.90, respectivamente. Estos excelentes resultados demuestran que la búsqueda de motivos sobre estas bases se realiza casi con perfección, encontrándose casi siempre la mayoría de las palabras o secuencias de nucleótidos correctas que constituyen el patrón.
2. Sobre bases reales de ADN, estos mismos métodos lograron una precisión y exhaustividad promedios de 0.87 y 0.76, respectivamente. Estos resultados son superiores en más de 20 puntos porcentuales a los resultados obtenidos por los

mejores métodos estadísticos de búsqueda de motivos. Por esta razón los métodos evolutivos generalmente encuentran un mayor número de patrones correctos que los hallados por métodos estadísticos.

3. Al comparar entre sí los resultados obtenidos por los dos métodos evolutivos desarrollados en este trabajo, podemos afirmar que aquel con base en la estimación de distribuciones es más preciso y exhaustivo en la identificación de los patrones del motivo que el método que utiliza los algoritmos genéticos. Esto porque el MBMEDA obtiene valores promedios de precisión y exhaustividad de 0.88 y 0.79 sobre las bases de ADN reales utilizadas, mientras que el MBMAG logra valores de 0.85 y 0.73, respectivamente, para las mismas métricas y casos.
4. Cuando se comparan los resultados de los métodos aquí desarrollados con aquellos obtenidos por otros métodos evolutivos que aparecen en la literatura (GAME, GALF, EDAMD), se observa que la diferencia en precisión y exhaustividad promedios es de sólo 2 puntos porcentuales, siendo máxima la exhaustividad del método EDAMD con un valor de 0.84. Por otro lado, el método de búsqueda GALF

presenta una precisión de 0.89, que es superior a la de los demás métodos evolutivos.

5. Todas estas comparaciones permiten concluir que en general los métodos desarrollados en este trabajo tienen una precisión y exhaustividad que son similares a las de otros métodos evolutivos existentes, y superiores a las de los métodos estadísticos.

Ambos métodos de búsqueda fueron probados utilizando un computador personal con las siguientes características: procesador Intel CORE I3 de 2.93GHz, y memoria RAM DDR3 de 6GB; el sistema operativo usado fue el Windows 7, y Visual el entorno de desarrollo, ambos productos de Microsoft; el diseño fue orientado a objetos y los algoritmos se codificaron en C++.

Los tiempos de ejecución del método de búsqueda que utiliza algoritmos genéticos fluctuaron entre 2 y 5 horas, mientras que para el método por estimación de distribuciones estos variaron entre 5 y 15 horas. Estos tiempos de ejecución fueron mayores que los reportados en la literatura para otros métodos de búsqueda de motivos, pero esta comparación no conduce a conclusiones puesto los entornos de desarrollo y prueba fueron diferentes.

En todo caso, los excelentes resultados obtenidos justifican el tiempo y recursos utilizados.

Recomendaciones

Los resultados obtenidos en este trabajo sugieren oportunidades futuras de investigación y desarrollo en el campo de la bioinformática, orientadas mejorar las herramientas existentes para identificar motivos biológicos. A continuación se presentan algunas recomendaciones para ampliar y profundizar el trabajo realizado en este proyecto de graduación.

1. El modelo utilizados para desarrollar los métodos evolutivos de búsqueda descritos en este trabajo, suponen la ocurrencia máxima de un patrón por cada zona promotora en la base de ADN. Esto disminuye de manera innecesaria el valor de exhaustividad que se puede obtener con los métodos evolutivos aquí desarrollados, por lo que se recomienda modificar el modelo para admitir la repetición del motivo en la misma zona promotora una o mas veces.
2. El modelo probabilístico utilizado por el método de búsqueda por estimación de distribuciones considera que las variables que definen a un individuo de la población son independientes; supuesto que no necesariamente es verdadero. Es posible que los resultados obtenidos mejoren si se toma en cuenta la interacción

posible entre dichas variables, debiendo entonces construirse nuevos estimadores para la media y la matriz de covarianza del modelo, puesto que los tradicionales que fueron probados no dieron buenos resultados.

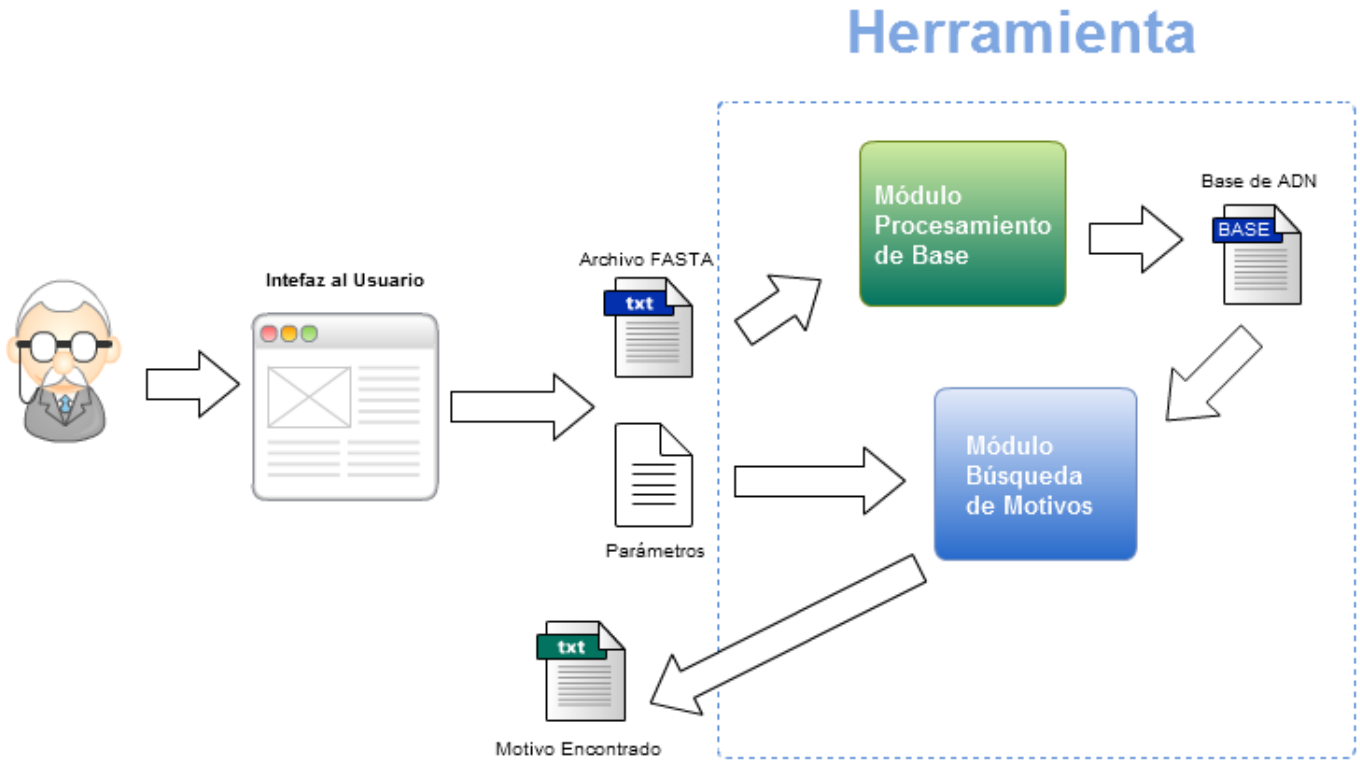
3. En cuanto al proceso de generar bases de ADN sintéticas, lo que en principio facilitaría el desarrollo de nuevos métodos de búsqueda de motivos, se recomienda investigar nuevos procedimientos que permitan construir bases con una distribución de nucleótidos de mayor similitud a las que ocurren en la naturaleza.
4. Con el propósito de disminuir los tiempos para desarrollar nuevos métodos y también el de ejecución de estos algoritmos, se recomienda utilizar el paradigma de programación en paralelo, lo que permitiría explotar convenientemente los varios núcleos que tienen los nuevos procesadores, distribuyendo las operaciones entre los varios núcleos y acelerando la entrega de los resultados. Con el mismo fin, se podría probar también usar los recursos de los procesadores gráficos (GPU) que traen los computadores, y que generalmente son unidades de procesamiento muy potentes.

Anexos

Implementación del prototipo de buscador de motivos

El prototipo de buscador de motivos es un prototipo funcional que busca las palabras que componen un motivo sobre una base de ADN en base a los 2 algoritmos evolutivos desarrollados en este trabajo. Este prototipo de buscador recibe un archivo de texto con las regiones promotoras del ADN de un organismo y un archivo que contiene los parámetros deseados por el usuario para un algoritmo evolutivo determinado. En base a estos 2 archivos, este prototipo devuelve al usuario un archivo de texto con el mejor individuo encontrado en todo el proceso evolutivo, junto con las posiciones iniciales de las palabras del motivo en la base de ADN y las secuencias de nucleótidos correspondientes a las posiciones iniciales encontradas. La siguiente figura muestra el funcionamiento de esta herramienta y sus 3 principales componentes:

Figura 29.- Esquema representativo del funcionamiento del prototipo funcional de un buscador de motivos basado en los métodos evolutivos desarrollados

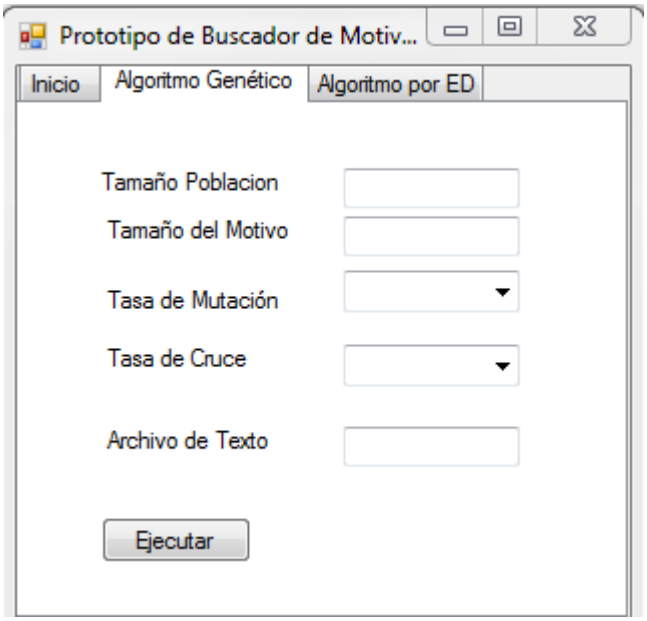


El prototipo desarrollado en este trabajo tiene 3 componentes principales: la interfaz al usuario (IU), el módulo de procesamiento de base (MPB) y el módulo de búsqueda de motivos (MBM). Estos 3 componentes fueron desarrollados de forma independiente, de tal manera que modificaciones en el código de un componente no afecten la funcionalidad de los otros componentes.

El módulo Interfaz al usuario (IU) ofrece una interfaz gráfica que permite al usuario ingresar la información requerida por la herramienta para efectuar la búsqueda de las instancias del motivo. La IU presenta dos

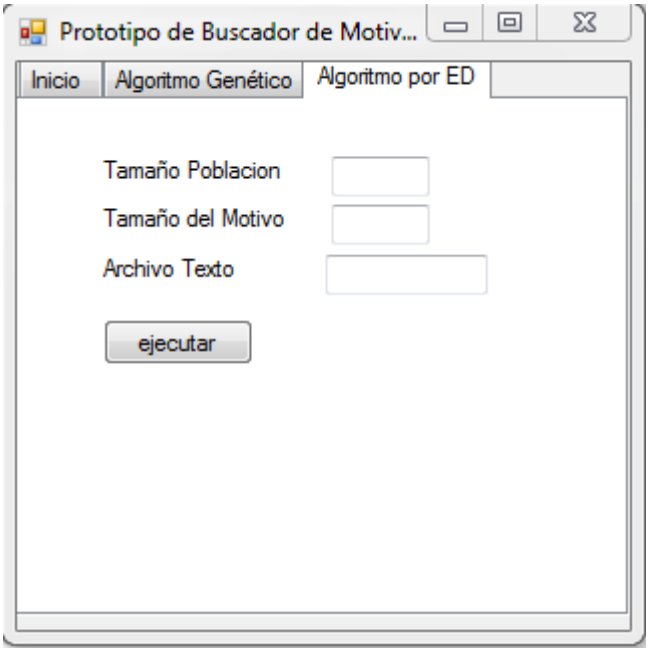
formularios divididos por pestañas, cada pestaña contiene un formulario para la ejecución de un algoritmo evolutivo. Los siguientes gráficos muestran los formularios para la ejecución del MBM en base a un algoritmo genético y un algoritmo por estimación de distribución.

Figura 30.- Formulario de parámetros necesarios para ejecutar la búsqueda de motivos utilizando el método MBMAG



Para el caso de la búsqueda del motivo utilizando un algoritmo genético, el buscador requiere que el usuario ingrese el tamaño de la población, el tamaño de las palaras del motivo, las tasas de mutación y cruce y el archivo de texto que contenga la base de ADN en formato FASTA.

Figura 31.- Formulario de parámetros necesarios para ejecutar la búsqueda de motivos utilizando el método MBMEDA



Si el usuario escoge realizar la búsqueda de motivos utilizando un algoritmo por estimación de distribuciones, el buscador sólo necesita conocer el tamaño de la población, el tamaño de las palabras del motivo y el archivo de texto de la base de ADN. Dado que es un prototipo, el archivo de texto que represente la base de ADN debe estar en una carpeta determinada de donde el programa realizará una copia de la misma.

Una vez recibida la información necesaria para ejecutar el algoritmo evolutivo, la interfaz al usuario entrega un archivo que contiene la base de ADN en formato FASTA al módulo procesamiento de Base y otro archivo que contiene los parámetros descritos por el usuario al módulo de búsqueda de motivos.

El módulo Procesamiento de Base recibe un archivo en Formato FASTA de la Interfaz y entrega al MBM un archivo de texto que contiene la base de ADN. El archivo de texto entregado por el usuario mediante la IU se encuentra en formato FASTA, el cual, como se mencionó con más detalles en el capítulo 5, añade información sobre las regiones promotoras e identifica las instancias del motivo mediante transcribir los nucleótidos en letras mayúsculas. Como ejemplo, para la Escherichia Coli, el factor de transcripción CRP se fija en una base de ADN cuyo formato FASTA tiene la siguiente presentación:

Figura 32.- Figura de la presentación de una base de regiones promotoras de ADN en formato FASTA

El MBM realiza una copia del archivo de texto de la IU en un archivo de texto base.txt y reescribe todos los nucleótidos del archivo en

minúsculas; además elimina los identificadores de las regiones promotoras, dejando un archivo de texto con un formato como muestra la siguiente figura:

Figura 33.- Figura de la presentación de una base de ADN lista para el procesamiento sobre el módulo de búsqueda de motivos

El MPB crea un archivo de texto llamado inter.txt que guarda las posiciones iniciales de las instancias del motivo en la base de ADN. Tanto el archivo de base de ADN como el archivo de ubicación de las posiciones iniciales son entregados al MBM.

El módulo Búsqueda de Motivos contiene las funciones necesarias para encontrar las instancias del motivo; es el módulo fundamental en el análisis de la base de ADN para encontrar las posiciones iniciales donde inicia el motivo buscado.

El MBM recibe de la IU un archivo parámetros.txt el cual contiene las especificaciones del usuario en cuánto a qué algoritmo evolutivo desea utilizar, el tamaño de la población de individuos, el tamaño estimado de las instancias del motivo, y en el caso del método basado en AG, las tasas de

cruce y mutación que el usuario estime convenientes. El MBM recibe del MPB dos archivos, base.txt y inter.txt; el archivo base contiene la base de ADN en un formato sobre el cual el MBM puede buscar las instancias del motivo. El archivo inter.txt sirve para comprobar si las posiciones encontradas por el algoritmo evolutivo son similares o iguales a las verdaderas palabras que componen el motivo. A partir del archivo inter.txt es posible calcular las métricas de desempeño como la Precisión y la Exhaustividad.

Nuestra herramienta toma ciertas suposiciones comunes en otras herramientas evolutivas desarrolladas. Una diferencia de +/- 3 posiciones entre las posiciones correctas y las encontradas por el algoritmo evolutivo se toma como una posición correctamente encontrada. Además, el algoritmo evolutivo se ejecuta un número n de veces antes de terminar el experimento; para nuestro trabajo, n =10. Esta consideración es necesaria ya que las poblaciones son generadas al inicio de forma aleatoria, debido a esto es recomendable ejecutar el algoritmo un número predeterminado de veces y elegir el mejor individuo de todas las iteraciones como el individuo que resuelve el problema de búsqueda de motivos.

Finalmente, el MBM entrega al usuario un archivo que contiene:

1. El mejor individuo en cada iteración

2. Las palabras del motivo correspondientes al VPI del mejor individuo de todas las iteraciones.

El mejor individuo en cada iteración es guardado siguiendo el siguiente formato

```
63 57 78 65 52 9 44 41 11 16 63 43 50 73 19 55 7 78 14.6456
60 56 75 62 49 6 41 38 8  13 60 40 47 70 16 52 0 77
```

Las posiciones iniciales correspondientes a las palabras del motivo son almacenadas en la primera fila, terminando con el valor de fitness correspondiente al individuo cuyo VPI tiene esos valores. La segunda fila tiene las posiciones correctas provenientes del archivo de texto entregado por el usuario. El archivo final a entregar al usuario tiene la siguiente presentación:

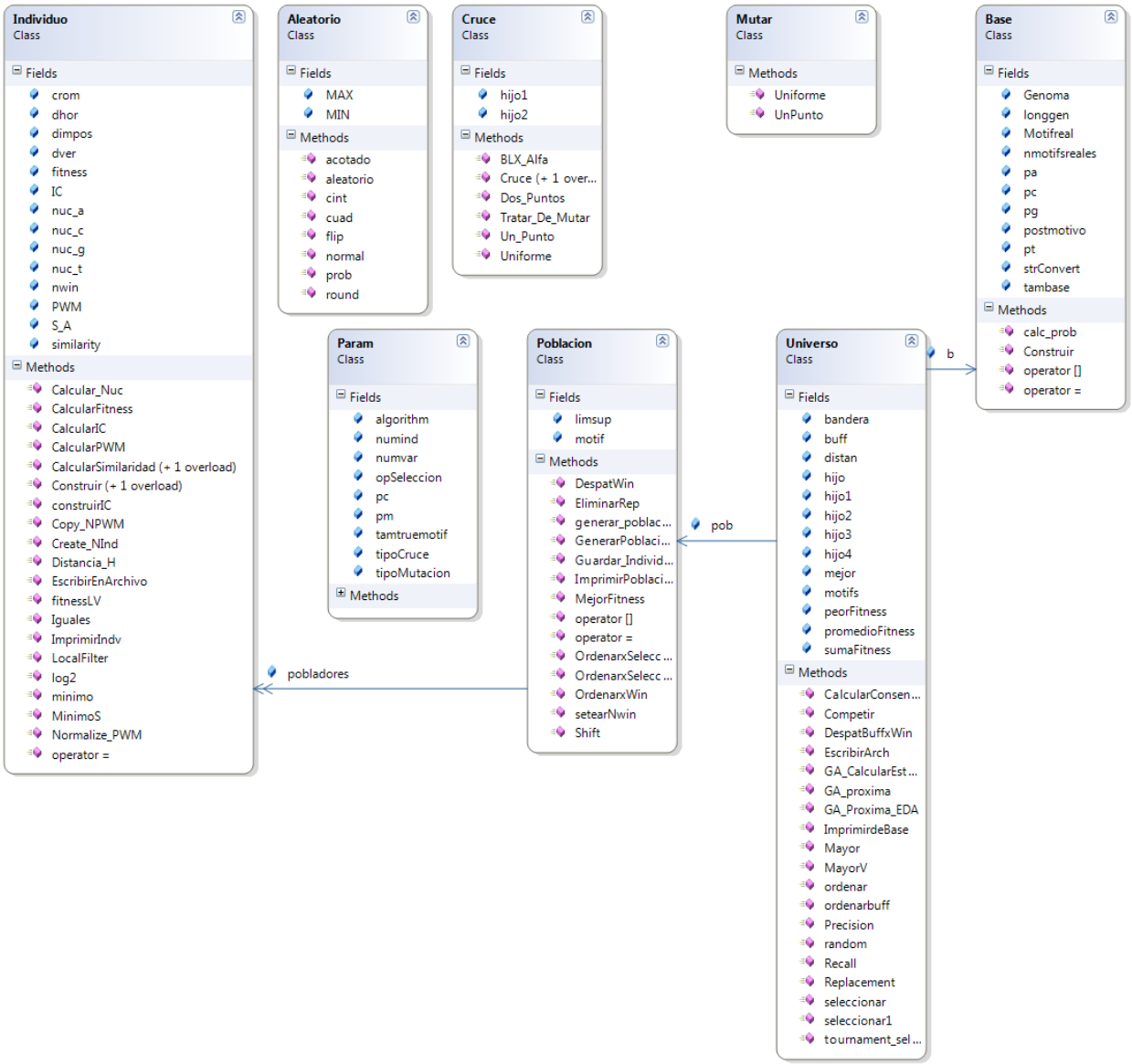
Figura 34.- Archivo de Resultados de la búsqueda de motivos sobre bases de ADN utilizando métodos evolutivos

```
63 57 78 65 52 9 44 41 11 16 63 43 50 73 19 55 7 78 14.6456
60 56 75 62 49 6 41 38 8  13 60 40 47 70 16 52 0 77

47 34 78 17 52 9 53 70 75 16 51 43 12 73 8 55 7 72 10.8574
16 16 75 62 49 6 41 38 8  13 28 40 47 70 16 52 0 77

63 57 78 63 52 62 24 68 11 16 51 43 50 73 19 55 75 78 11.5244
16 16 75 62 49 6 41 38  8 13 28 40 47 70 16 52 0 77

63 57 78 65 52 9 26 41 11 16 51 43 50 73 19 55 86 80 13.4611
16 16 75 62 49 6 41 38 8  13 28 40 47 70 16 52 0 77

63 57 78 47 52 62 64 15 11 16 63 43 41 18 19 55 38 80 10.8773
16 16 75 62 49 6  41 38 8  13 28 40 47 70 16 52 0 77

63 57 78 65 52 9 44 12 11 16 63 36 75 50 19 55 29 80 11.7129
16 16 75 62 49 6 41 38 8  13 28 40 47 70 16 52 0 77

61 55 17 63 76 7 42 74 9 14 35 69 64 87 75 6 27 76 12.1791
16 16 75 62 49 6 41 38 8 13 28 40 47 70 16 52 0 77

63 49 87 47 52 62 53 24 75 16 63 43 41 73 19 55 84 48 11.5273
16 16 75 62 49 6  41 38 8  13 28 40 47 70 16 52 0 77

47 57 78 65 17 42 24 68 11 16 63 53 50 73 19 55 1 80 12.0778
16 16 75 62 49 6  41 38 8  13 28 40 47 70 16 52 0 77

61 55 61 63 50 7 76 33 75 14 61 69 48 28 75 6 84 34 10.8231
16 16 75 62 49 6 41 38 8  13 28 40 47 70 16 52 0 77


El mejor motivo encontrado es:
63 57 78 65 52 9 44 41 11 16 63 43 50 73 19 55 7 78 14.6456
ttttgatcgttttcacaa
atttgcacggcgtcacac
ctgtgagcatggtcatat
atgcaaaggacgtcacat
gtgttaaattgatcacgt
atttgaaccagatcgcat
gtgtaaacgattccacta
acgtgatcaacccctcaa
atgtgagttagctcactc
ctgtaacagagatcacac
atgtaaggaatttcgtga
ttgtgacacagtgcaaat
gcctgacggagttcacac
ttgtgattcgattcacat
ttgtgatgtggttaaccc
gtgtgaaataccgcacag
atgagacgttgatcggca
ttgtgagtggtcgcacat
```

El prototipo de buscador fue codificado utilizando los lenguajes de C++ y C# usando Microsoft Visual Studio 2010.Los módulos MBM y MPB fueron codificados utilizando C++ y la IU fue codificada en C#. La programación utilizada para MPB fue siguiendo un esquema estructurado;

MBM utilizó una programación orientada a objetos. A continuación se muestra el diagrama de clases de MBM:

Figura 35.- Diagrama de las clases en el módulo de búsqueda de motivos



Las clases Mutar y Cruce contienen todos los métodos para realizar el cruce y la mutación en el método basado en AG. La clase Param contiene los parámetros provenientes de IU para elegir el tipo de algoritmo evolutivo sobre el cual se trabaja, el número de individuos, el tamaño de las palabras del motivo, etc. La clase Universo contiene una Población *pob* de individuos, junto con otros campos de control.

La clase Cromosoma agrupa a los genes utilizando un vector de números enteros. Un Cromosoma es la representación en un Individuo del vector de posiciones iniciales en la base de ADN. La clase Individuo contiene información sobre un individuo como su valor de fitness, la matriz de pesos posicionales (MPP), las secuencias correspondientes al cromosoma (S_A) junto con funciones como calcular el fitness y realizar el filtrado local. La clase Población contiene un vector de Individuos junto con funciones que se ejecutan sobre todos los individuos, como los operadores selección de los mejores individuos, la operación de Desplazamiento, el generar una nueva población o eliminar individuos que se repitan dentro de la población.

En la clase Universo se ejecutan 2 funciones fundamentales en el MBM: **GA_Proxima** y **GA_Proxima_EDA**. GA_Proxima toma los mejores individuos de la población y genera una nueva población de individuos basándose en el esquema de un algoritmo genético. Esta función utiliza un operador de cruce y mutación en 1 punto basándose en el segundo método desarrollado que se encuentra en el capítulo 4. La selección se realiza

mediante un torneo entre los individuos en la población. Estos operadores no están sujetos a elección en el prototipo desarrollado.

GA_Proxima_EDA realiza funciones similares a GA_Proxima pero siguiendo el esquema de un algoritmo por ED. Esta función muestrea los nuevos individuos en base a 4 modelos normales univariados, siguiendo los parámetros descritos en el método 2 basado en ED mostrado en el capítulo 4.

La función main del módulo de búsqueda de motivos se muestra a continuación

```
void main(){
Base b;
b.Construir(); //Construye la base a partir de base.txt proveniente de MPB
Individuo inv;
Param par;
double dif;
int nrep=1; //número de veces q se repite el mejor individuo en una población
int ngen=1; //número de generaciones en el proceso evolutivo
par.SetValues(b);
for(int t=0;t<10;t++){
        srand (time(NULL));
        ngen=nrep=1;
        w.random(0); //Genera la primera población de forma aleatoria
        for(int i=0;i<= par.numind-1;i++)
           w.pob[i].LocalFilter(w.b); //Primer Filtrado Local
                do {
                   cout<<"Numero de generacion:"<<ngen<<"\n";
                   w.ordenar();
                   if(par.method=="AG")
                      w.GA_proxima(); //Utiliza AG
                   else
                      w.GA_Proxima_EDA(); //Utiliza EDA
                   w.ordenar();
                   w.pob[0].ImprimirIndv();
```

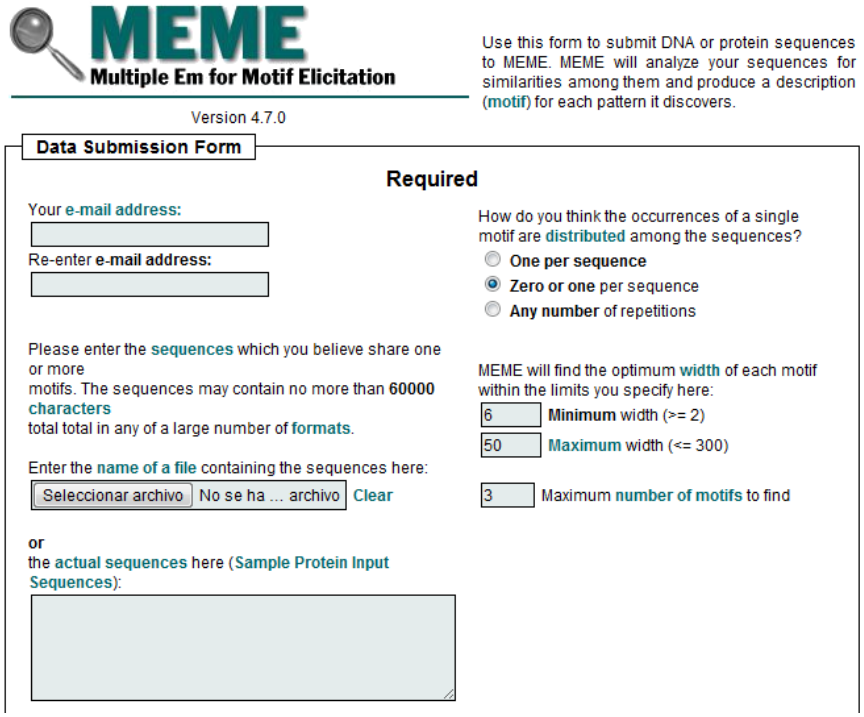
```
                if(ngen==1){
                    inv=w.pob[0];
                }
                else{
                    if(inv.fitness==w.pob[0].fitness&&
inv.Iguales(w.pob[0].crom.genes))
                        nrep++;
                    else{
                        inv=w.pob[0];
                        nrep=1;
                    }
                    if(nrep%10==0 && nrep!=0) { //estancamiento
                    //Ejecuta Desplazamiento sobre mejor
                        cout<<"Comienzo del Desplazamientoing"<<"\n";
                        w.pob.Desplazamiento(w.b);
                    }
                    if(ngen%10==0 && ngen!=0){ // cada 10 generaciones
                        //Ejecuta Filtrado Local
                            cout<<"Comienzo del local filtering"<<"\n";
                            for(int i=0;i<=par.numind-1;i++)
                                w.pob[i].LocalFilter(w.b);
                    }

                }
                ngen++;
            }while(ngen<=numgeneraciones && nrep<=50);
            GuardarRes();
    }
  }
```

El prototipo desarrollado deja abierta la posibilidad de desarrollar algunas funcionalidades que permitirían presentar una herramienta funcional para cualquier investigador en el campo de la biología molecular. Una funcionalidad que podría ser deseable es la oportunidad de escoger en el formulario para el algoritmo genético el tipo de operador de cruce, mutación y selección que el usuario desee. Otra funcionalidad interesante para el

usuario sería la posibilidad de buscar más de dos motivos en una misma región promotora, junto con un rango posible del tamaño de las palabras que conforman el motivo.

El prototipo de buscador desarrollado en este trabajo guarda muchas similitudes con un formulario vía web que ofrece un centro de cómputo en San Diego, EEUU basado en el método MEME para encontrar las palabras de uno o varios motivos dentro de una misma base de ADN.

Figura 36.- Formulario del método de búsquedas de motivo basado en MEME



Referencias Bibliográficas

[1] Fundación Wikimedia, "Central dogma of molecular biology." http://en.wikipedia.org/wiki/Central_dogma_of_molecular_biology, Web, 22 Nov. 2011.

[2] Fundación Wikimedia, "Chromosome." http://en.wikipedia.org/wiki/Chromosome, Web, 22 Nov. 2011.

[3] Fundación Wikimedia "DNA replication." http://en.wikipedia.org/wiki/DNA_replication, Web, 22 Nov. 2011.

[4] Fundación Wikimedia, "Nucleotide." http://en.wikipedia.org/wiki/Nucleotide, Web. 22 Nov. 2011.

[5] Fundación Wikimedia, "Transcription (genetics).". http://en.wikipedia.org/wiki/Transcription_(genetics), Web, 22 Nov. 2011

[6] Fundación Wikimedia, "Protein." http://en.wikipedia.org/wiki/Protein, Web, 22 Nov. 2011.

[7] Liu, X. L. "Bioprospector: Discovering Conserved DNa Motifs in Upstream Regulatory Regions of Co-expressed Genes." *Pacific Symposium on Biocomputing* 6 (2001): 127-38

[8] Fundación Wikimedia, "Translation (biology)". http://en.wikipedia.org/wiki/Translation_genetics, Web, 22 Nov. 2011

[9] Fundación Wikimedia, "Gel electrophoresis" http://en.wikipedia.org/wiki/Gel_electrophoresis, Web, 22 Nov. 2011

[10] Fundación Wikimedia, "Evolutionary computation" http://en.wikipedia.org/wiki/Evolutionary_computation, Web, 22 Nov. 2011.

[11] Fundación Wikimedia, "Genetic algorithm." http://en.wikipedia.org/wiki/Genetic_algorithm, Web, 22 Nov. 2011.

[12] Galas, David J., and Albert Schmitz. "DNAase Footprinting a Simple Method for the Detection of Protein-DNA Binding Specificity." *Nucleic Acids Research* 5.9 (1978): 3157-170.

[13] Garner, Mark M., and Arnold Revzin. "A Gel Electrophoresis Method for Quantifying the Binding of Proteins to Specific DNA Regions: Application to Components of the Escherichia Coli Lactose Operon Regulatory System." *Nucleic Acids Research* 9.13 (1981): 3047-060.

[14] Fundación Wikimedia, "Cholesky decomposicion." http://en.wikipedia.org/wiki/Cholesky_decomposition, Web, 22 Nov. 2011.

[15] Stormo GD. “DNA binding sites: representation and discovery”. *Bioinformatics.* 2000 Jan;16(1):16-23. Review. PubMed PMID: 10812473.

[16] Das MK, Dai HK. “A survey of DNA motif finding algorithms”. *BMC Bioinformatics*. 2007 Nov 1;8 Suppl 7:S21. Review. PubMed PMID: 18047721; PubMed Central PMCID: PMC2099490.

[17] Bailey, Timothy L., and Charles Elkan. "Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization." *Machine Learning* 21.1-2 (1995): 51-80.

[18] Bengoetxea, Endika, Pedro Larrañaga, Isabelle Bloch, and Aymerci Perchant. "Estimation of Distribution Algorithms: a New Evolutionary Computation Approach for Graph Matching Problems." *Energy Minimization Methods in Computer Vision and Pattern Recognition* 2134 (2001): 454-69.

[19] Dellaert, Frank. *The Expectation Maximization Algorithm*. Tech. no. GIT-GVU-02-20. Georgia: Georgia Institute of Technology, 2002.

[20] Eiben, Agoston E., and J. E. Smith. "What Is an Evolutionary Algorithm." *Introduction to Evolutionary Computing*. New York: Springer, 2003. Cap 2.

[21] Beyer Hans-Georg , Schwefel Hans-Paul. *Evolution Strategies: A Comprehensive Introduction*. 1st ed. Amsterdam: Springer, 2002. 1-16.

[22] Janet, Carolina, and Pasto Humpiri. "Estrategias Evolutivas No Planejamento Energetico Da Operacao De Sistemas Hidrotermincos De Potencia." Thesis. Campiñas,Sao Paulo, 2005. Págs. 29-32

[23] Jones, Neil C., and Pavel Pevzner. "Exhaustive Search." *An Introduction to Bioinformatics Algorithms*. Cambridge, MA: MIT, 2004. 100-13.

[24] Jones, Neil C., and Pavel Pevzner. "Exhaustive Search." *An Introduction to Bioinformatics Algorithms*. Cambridge, MA: MIT, 2004. 91-97.

[25] Jones, Neil C., and Pavel Pevzner. "Exhaustive Search." *An Introduction to Bioinformatics Algorithms*. Cambridge, MA: MIT, 2004. 97-100.

[26] Jones, Neil C., and Pavel Pevzner. "Molecular Biology Primer." *An Introduction to Bioinformatics Algorithms*. Cambridge, MA: MIT, 2004. 57-68.

[27] Luke, Sean. *Essencials of Metaheuristics*. 1st ed. Washington: Lulu, 2009.

[28] Nedjah, Nadia, and Liuza De Marcedo Mourelle. *Multi-objective Machine Learning: Studies in Computational Intelligence*. By Ajith Abraham. 2006 ed. Vol. 13. [S.l.]: Springer, 2009. 920-23.

[29] Eiben, Agoston E., and J. E. Smith. "¿What Is an Evolutionary Algorithm?" *Introduction to Evolutionary Computing*. New York: Springer, 2003. 15-35

[30] Eiben, Agoston E., and J. E. Smith. "Genetic Algorithms" *Introduction to Evolutionary Computing*. New York: Springer, 2003. 37-69

[31] Poole, David L., Alan K. Mackworth, and Randy Goebel. *Computational Intelligence: a Logical Approach*. New York: Oxford UP, 1998.

[32] Schwefel, Hans Paul. "Introduction." *Natural Computing*. By Hans Georg Beyer. Amsterdam: Springer, 2002. 1-16.

[33] Schwefel, Hans Paul. "Natural Computing." *Evolution Strategies: A Comprehensive Introduction*. By Hans Georg Beyer. Amsterdam: Springer, 2002. 1-16.

[34] Shao, Linlin, Ajith Abraham, and Yuehi Chen. "Motif Discovery Using Evolutionary Algorithms." *Soft Computing and Pattern Recognition* (2009): 420-25.

[35] Wang, T. "Identifying the Conserved Network of Cis-regulatory Sites of a Eukaryotic Genome." *Proceedings of the National Academy of Sciences* 102.48 (2005): 17400-7405.

[36] Armañanzas, Rubén, Iñaki Inza, Roberto Santana, Yvan Saeys, Jose Flores, Jose Lozano, Yves Peer, Rosa Blanco, Víctor Robles, Concha Bielza, and Pedro Larrañaga. "A Review of Estimation of Distribution Algorithms in Bioinformatics." *BioData Mining* 1.1 (2008): 6.

[37] Chan, Talk Ming, Kwong Sak Leung, and Kin Hong Lee. "TFBS Identification Basen on Genetic Algorithm with Combined Representations and Adaptive Post-processing." *Bioinformatics* 24.3 (2008): 341-49.

[38] Chipperfield, A. J., and P. J. Fleming. "Applied Control Techniques Using Matlab." *IEE Colloquium* (2002). The Matlab genetic algorithm toolbox

[39] Hertz, Gerald Z., and Gary D. Stormo. "Identifying DNA and Protein Patterns with Statistically Significant Aligments of Multiple Sequences." *Bioinformatics* (1999): 563-77.

[40] Li, Gang, Tak Ming Chan, Kwong Sak Leung, and Kin Hong. "An Estimation of Distribution Algorithm for Motif Discovery." *Evolutionary Computation* (2008): 2411-418.

[41] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*. New York: Cambridge UP, 2008. 151-58.

[42] Mitchell, Melanie. "Genetic Algorithms: An Overview." *An Introduction to Genetic Algorithms*. New Delhi: Prentice Hall of India, 2002. 1-31.

[43] Schneider, T., G. Stormo, L. Gold, and A. Ehrenfeucht. "Information Content of Binding Sites on Nucleotide Sequences." *Journal of Molecular Biology* 188.3 (1986): 415-31.

[44] Schneider, Thomas D., and R.Michael Stephens. "Sequence Logos: a New Way to Display Consensus Sequences." *Nucleic Acids Research* 18.20 (1990): 6097-100.

[45] Ussery, David W., Trudy M. Wassenaar, and Stefano Borini. "Sequences as Biological Information: Cells Obey the Laws of

Chemistry and Physics." *Computing for Comparative Microbial Genomics Bioinformatics for Microbiologists*. London: Springer, 2009. 3-17

[46] Wei, Z. "GAME: Detecting Cis-regulatory Elements Using a Genetic Algorithm." *Bioinformatics* 22.13 (2006): 1577-584.

[47] Wang T, Stormo GD: Identifying the conserved network of cisregulatory sites of a eukaryotic genome. *PNAS* 2005,102:17400-17405.

[48] Wheeler, David; Bhagwat, Medha (2007). "Chapter 9 BLAST QuickStart". In Bergman, Nicholas H.. Comparative Genomics Volumes 1 and 2. Methods in Molecular Biology. 395-396. Totowa, NJ: Humana Press

[49] Whompson JD, Higgins DG, Gibson TJ: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Res* 1994, 22:4673