TESIS DEFENDIDA POR

Mauricio Antonio Chalita Williams

Y APROBADA POR EL SIGUIENTE COMITÉ

Dr. Carlos Alberto Brizuela Rodríguez

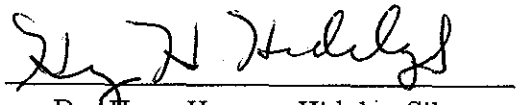
Director del Comité

Dr. Miguel Ángel del Río Portilla

Miembro del Comité

Dr. Andrey Chernykh

Miembro del Comité

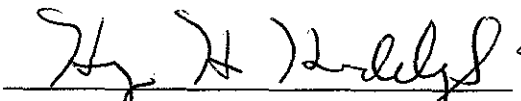


Dr. Hugo Homero Hidalgo Silva

Miembro del Comité

Dr. Israel Marck Martínez Pérez

Miembro del Comité



Dr. Hugo Homero Hidalgo Silva

Coordinador del programa de posgrado en Ciencias de la Computación

Dr. David Hilario Covarrubias Rosales

Director de Estudios de Posgrado

6 de Diciembre de 2011

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA

PROGRAMA DE POSGRADO EN CIENCIAS

EN CIENCIAS DE LA COMPUTACIÓN

ENCADENAMIENTO DE ALGORITMOS PARA MEJORAR MÉTODOS DE BÚSQUEDA DE MOTIVOS EN SECUENCIAS DE ADN

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

MAESTRO EN CIENCIAS

Presenta:

MAURICIO ANTONIO CHALITA WILLIAMS

Ensenada, Baja California, México, Diciembre de 2011

RESUMEN de la tesis de **MAURICIO ANTONIO CHALITA WILLIAMS**, presentada como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN . Ensenada, Baja California, Diciembre de 2011.

ENCADENAMIENTO DE ALGORITMOS PARA MEJORAR MÉTODOS DE BÚSQUEDA DE MOTIVOS EN SECUENCIAS DE ADN

Resumen aprobado por:

Dr. Carlos Alberto Brizuela Rodríguez

Director de Tesis

En esta tesis se describe una propuesta de solución para el problema de la búsqueda de motivos en secuencias de ADN. Los motivos son pequeños fragmentos de ADN los cuales están ubicados en la región promotora del gen. Estos fragmentos son sitios de acoplamiento de proteínas llamadas factores de transcripción las cuales permiten que se comience o se inhiba la transcripción de un gen. Debido a la complejidad de este problema, se han propuesto diferentes modelos y algoritmos para poder abordarlos. En este trabajo se propone utilizar un encadenamiento de algoritmos de búsqueda de motivos para atacar este problema y mejorar la exactitud de predicción de los algoritmos que lo componen. El encadenamiento RM+BP+MM consiste en preprocesar los casos de prueba con RepeatMasker, para remover secuencias irrelevantes, seguido de BioProspector, el cual es utilizado para limitar el espacio de búsqueda que MEME realiza para la búsqueda de motivos. El encadenamiento RM+BP+WR+MM preprocesa la información de la misma manera, mientras que para limitar el espacio de búsqueda de MEME, utiliza Weeder y BioProspector simultáneamente. Se utilizó el *benchmark* propuesto por Tompa *et al.* (2005), que consiste en casos de prueba reales de organismos eucariotas (humano, ratón, mosca y levadura). Resultados computacionales muestran que el encadenamiento de algoritmos propuesto puede mejorar la búsqueda de motivos. Ambos encadenamientos propuestos (RM+BP+MM y RM+BP+WR+MM), mejoraron los resultados de los algoritmos que los componen. RM+BP+MM obtuvo la sensibilidad, rendimiento y especificidad más alta en comparación con Weeder, MEME y BioProspector, mientras que RM+BP+WR+MM únicamente superó en sensibilidad y rendimiento a los algoritmos anteriores.

Palabras Clave: Búsqueda de motivos, regiones promotoras, sitios de pegado de factores de transcripción.

ABSTRACT of the thesis presented by **MAURICIO ANTONIO CHALITA WILLIAMS**, in partial fulfillment of the requirements of the MASTER IN SCIENCES degree in COMPUTER SCIENCE . Ensenada, Baja California, December 2011.

A PIPELINE FOR THE IMPROVEMENT OF MOTIF FINDING METHODS IN DNA SEQUENCES

In this work we present two pipelines for the motif finding problem. Motifs are small fragments of DNA located in the gene promoter region. These fragments are binding sites for proteins known as transcription factors which are involved in the gene regulation process. Due to the complexity of this problem, many models and algorithms to solve them have been proposed. This paper proposes the use of pipeline algorithms to attack this problem and improve the accuracy of prediction of the component algorithms. The RM+BP+MM pipeline preprocesses the data sets with RepeatMasker to remove irrelevant sequences, followed by BioProspector, which is used to limit the search space that MEME explorea for the motif search. The RM+BP+WR+MM pipeline preprocesses the data sets in the same way, while BioProspector and Weeder are used simultaneously in order to reduce the search space for MEME. Tompa's *et al.*(2005) benchmark was used, it consists of several real instances of eukaryotic organisms (human, mouse, fruit fly and yeast). In this work, it is shown that pipeline algorithms can improve the search for motifs. Both proposed pipelines (RM+BP+MM y RM+BP+WR+MM), improved the results of the component algorithms. RM+BP+MM obtained better sensitivity, specificity and performance against Weeder, MEME and BioProspector, while RM+BP+WR+MM only obtained better sensitivity and performance against the previous algorithms.

Keywords: Motif finding, regulatory regions, transcription factor binding sites.

A mis padres

Agradecimientos

A mis padres Juan Carlos y Verónica, porque a ellos les debo todos mis éxitos.

A mis hermanos Juan Carlos, Alejandro y Verónica que siempre me han apoyado en todas mis decisiones.

Al Dr. Carlos A. Brizuela Rodríguez por su amistad, consejo y guía durante el desarrollo de este trabajo.

A todo el personal y compañeros del departamento de Ciencias de la Computación por su amistad y ayuda durante toda mi estancia en Ensenada.

Y al CONACYT por su apoyo económico.

Contenido

Contenido (continuación)

Contenido (continuación)

Contenido (continuación)

Lista de Figuras

Lista de Figuras (continuación)

Lista de Figuras (continuación)

Lista de Figuras (continuación)

Lista de Tablas

Lista de Tablas (continuación)

Capítulo I

Introducción

I.1 Antecedentes

Durante mucho tiempo, uno de los problemas básicos en la biología fue entender la herencia. Es decir, cómo se transmiten en los organismos ciertos patrones de generación en generación. Mendel propuso un primer modelo abstracto esencialmente matemático de la herencia en el cual la unidad básica era el gen. Watson y Crick descubrieron la estructura de doble hélice para el ADN, abriendo así la posibilidad de estudiar la mecánica de la herencia (Espinoza, 2004).

El ADN junto con el ARN y las proteínas son las moléculas de la célula más importantes para nosotros. El ADN es la base fundamental de la herencia en los organismos vivos. Está constituido por pequeñas moléculas llamadas nucleótidos. Estos nucleótidos son cuatro en número y pueden ser distinguidos por una de las cuatro bases nitrogenadas que la componen: adenina (A), citosina (C), guanina (G) y timina (T). Dicho de otra forma, el ADN es una cadena sobre un alfabeto de cuatro letras: A,C,G,T.

Un organismo no usa el producto de todo gen en su genoma al mismo tiempo o en todos sus tejidos. El uso y la activación de un gen se puede dar de distintas maneras, una de estas formas de regulación es el control de transcripción génica por una clase especial de proteínas llamadas factores de transcripción. Estas proteínas se adhieren al ADN en unos sitios de pegado, regiones que contienen sólo unas pocas bases de largo, que poseen secuencias específicas que sirven de señal para el factor (Vanet *et al.*, 2000).

Los sitios de pegado de los factores de transcripción son difíciles de identificar en secuencias de ADN. Se sabe que esos sitios se encuentran en zonas cercanas a las regiones promotoras de los genes, típicamente dentro de unos cientos o miles de bases de donde comienza el sitio de transcripción (Galas *et al.*, 1985). Estas regiones promotoras son secuencias conservadas en el ADN y juegan un papel fundamental en el proceso de la transcripción, sin embargo, estos sitios son difíciles de encontrar porque aparecen mutados, es decir, no todos los sitios de pegado que son reconocidos por un mismo factor de transcripción están compuestos por las mismas letras sino que pueden haber variaciones de sitio a sitio.

Un desafío relevante en bioinformática es diseñar algoritmos eficientes para la búsqueda e identificación de sitios de pegado. Esto, por las posibles aplicaciones en la localización de sitios reguladores e identificadores de objetivos de drogas que tendría conocer la biblioteca de motivos de un organismo (Galas *et al.*, 1985).

Debido a que el descubrimiento de motivos es un problema complejo, se han hecho distintos planteamientos con el objetivo de obtener versiones más simplificadas del mismo para así poder plantear propuestas de solución. Sagot (1998) plantea una formulación combinatoria precisa de este problema, llamada problema del motivo implantado, la cual es de interés particular para este trabajo, ya que es un modelo comúnmente utilizado por algoritmos de búsqueda de motivos (Price *et al.*, 2003).

Diversos algoritmos han sido desarrollados para la búsqueda de motivos utilizando distintas estrategias como son enumeración (Blanchette *et al.*, 2002; Brazma *et al.*, 1998; Sinha y Tompa, 2000), búsqueda local (Hertz y Stormo, 1999; Lawrence y Reilly, 1990; Bailey y Elkan, 1995), árboles de sufijos (Sagot, 1998), proyecciones aleatorias (Buhler y Tompa, 2002), entre otras. Algunos de ellos, construidos para abordar el problema bajo el modelo del motivo implantado, han obtenido un alto desempeño en

casos artificiales como ocurre con el algoritmo Pattern Branching (Price *et al.*, 2003), sin embargo, esto no asegura un alto desempeño en los casos reales. Por otra parte, algoritmos que no están basados en este modelo, como lo son Gibbs Sampler (Lawrence *et al.*, 1993) y MEME (Bailey y Elkan, 1995), los cuales están basados en búsqueda local sobre un modelo probabilístico, son muy utilizados en la práctica aunque para los casos implantados tienen un bajo desempeño, esto debido a que están fuertemente influenciados por la distribución precisa de las mutaciones del motivo, ya que comienzan su búsqueda adivinando una ocurrencia inicial del motivo y posteriormente tratan de encontrar otras ocurrencias seleccionando aquellas que son similares a la inicial. Múltiples modificaciones a estos dos últimos se han hecho en los últimos años, con el propósito de mejorar los resultados con cierto tipo de secuencias en específico (Bailey *et al.*, 2010).

Durante los últimos años, se han desarrollado enfoques utilizando casos reales, suponiendo que entre los nucleótidos en el ADN existe cierta dependencia entre ellos y sus posiciones, esto es, utilizando información de factores de transcripción de la misma familia, de la cual, se conoce de antemano, sus motivos (Fatemeh *et al.*, 2009; Tomovic *et al.*, 2009).

I.2 Definición del Problema

El problema de búsqueda de motivos consiste en identificar pequeños sitios conservados en el ADN sin conocer, a priori, la longitud ni los nucleótidos que los conforman. Para secuencias de ADN, la búsqueda de motivos es comúnmente aplicada a conjuntos de secuencias (de un mismo genoma o de genomas de distintas especies) que han sido identificadas por poseer un motivo en común, lo cual convierte el problema biológico original,

en uno combinatorio donde herramientas computacionales pueden ser utilizadas para resolverlo.

La dificultad de este problema recae en que las instancias del motivo presentan mutaciones (cambio de nucleótido), inserciones o ausencia de algunos nucleótidos y usualmente no ocurren exactamente igual, por lo que, mientras que ocurrencias aproximadas de un patrón dado pueden ser encontradas eficientemente, buscar todos los posibles 4^l patrones se vuelve más costoso conforme crece la longitud l del motivo.

El problema de búsqueda de motivos se puede definir de la siguiente manera: Dado un conjunto de N secuencias cada una de longitud T_i donde $i = 1, ..., N$ y dada la longitud del motivo l y el número máximo de mutaciones permitidas d, encontrar todas las ocurrencias del $motivo - (l, d)$ que se encuentran implantadas en las N secuencias.

I.3 Motivación

La regulación genética se relaciona directamente con el control del desarrollo de los organismos, producción de hormonas, la respuesta celular ante el estrés como las enfermedades o condiciones físicas adversas. El mal funcionamiento de la regulación genética tiene efectos adversos en el hombre y en otros organismos (tumores, cáncer, alergias, etc.). Por lo que poder entender la regulación genética sería de gran beneficio para el hombre (Vanet *et al.*, 2000).

Debido a que no se conoce completamente el fenómeno que rodea a los sitios de pegado en ADN (motivos), el diseño de algoritmos que puedan predecir dichos motivos es un desafío mayor (Tompa *et al.*, 2005).

La predicción de motivos es un problema amplio, tanto de exactitud en la predicción, como en la complejidad computacional. Esto ha llevado a proponer distintos modelos

del problema con el objetivo de encontrar uno que más se acerque a la realidad.

Tompa *et al.* (2005) proponen la combinación de varias heurísticas para mejorar la exactitud de predicción, todo esto bajo la hipótesis de que las herramientas evaluadas, generan resultados favorables para problemas específicos. Por lo que generar una herramienta que combine varios algoritmos podría ayudar a mejorar la calidad de predicción en este problema. También señala que la forma de evaluar dichas heurísticas es un problema todavía sin solución, por lo que aún se necesitan metodologías de evaluación confiables. Además mencionan que los algoritmos para la búsqueda de motivos tienen buenos resultados cuando son aplicados a organismos procariotas, pero cuando se busca resolver este problema en eucariotas, la exactitud de los algoritmos empeora, debido a que los sitios de pegado son degenerados. Obtener buena exactitud en eucariotas es un paso necesario para la comprensión del funcionamiento regulador de genes y para el desarrollo de drogas y vacunas para el hombre (Vanet *et al.*, 2000).

En resumen, un gran número de métodos que abordan el problema desde distintos enfoques han sido propuestos (Blanchette *et al.*, 2002; Brazma *et al.*, 1998; Sinha y Tompa, 2000; Hertz y Stormo, 1999; Lawrence y Reilly, 1990; Bailey y Elkan, 1995; Sagot, 1998; Buhler y Tompa, 2002; Price *et al.*, 2003; Lawrence *et al.*, 1993; Bailey y Elkan, 1995). Sin embargo, con todos estos métodos hay varias preguntas importantes que están aún por responderse, una de ellas es: ¿existe una combinación o encadenamiento de estos métodos que permitan mejorar la exactitud de cada método del encadenamiento cuando se aplican por separado? El trabajo de esta tesis propone una respuesta a esta pregunta.

I.4 Objetivos

I.4.1 Objetivo General

Diseñar un encadenamiento de algoritmos para búsqueda de motivos que mejore la capacidad de predicción de sus metodologías componentes y que sea competitivo con los resultados del estado del arte.

I.4.2 Objetivos Específicos

Para lograr el objetivo general, se definieron una serie de puntos que nos ayudaron a guiar el trabajo. Estos fueron planteados como preguntas, de tal manera que se pretende sean contestadas al finalizar el trabajo de investigación:

- ¿Cómo funcionan los métodos actuales?
- ¿Cómo se mide el desempeño de los métodos actuales?
- ¿Cuál es el desempeño de éstos?
- ¿Cómo generar una metodología que combine varios algoritmos para mejorar los resultados?
- ¿Cuál es la mejoría que se logra en comparación con las metodologías actuales?
- ¿Cómo afecta la estructura de los casos de prueba a los algoritmos?
- ¿Cuál es la ventaja de usar un algoritmo combinado?
- ¿Es significativa la mejoría del algoritmo combinado?
- ¿Cuántos falsos positivos generan los algoritmos?

I.5 Metodología de Investigación

La metodología seguida para alcanzar los objetivos establecidos es como sigue:

1. Observación: revisión bibliográfica que permitió de una manera concreta definir objetivos y limitantes.

2. Hipótesis: se hicieron una serie de suposiciones sobre las cuales se diseñaron los encadenamientos de algoritmos.

3. Experimentación: se construyeron algoritmos combinados utilizando casos de prueba reales.

4. Análisis de Resultados: se analizó la mejoría de las combinaciones para detectar posibles puntos de mejoría o buscar combinaciones adicionales.

5. Se iteró sobre los puntos 2 a 4 hasta que se obtuvo una mejoría significativa contra los algoritmos existentes.

I.6 Organización de la Tesis

Este trabajo consta de siete capítulos y está organizado de la siguiente manera:

En el Capítulo II "Marco Teórico" se explica el problema biológico de los sitios de pegado y factores de transcripción, así como los algoritmos que actualmente existen para resolver este problema.

En el Capítulo III "Evaluaciones de los algoritmos de búsqueda de motivos" se explica cómo evaluar la exactitud de los algoritmos, así como los diferentes tipos de casos de prueba que se pueden utilizar.

En el Capítulo IV "Algoritmos" se explican los algoritmos que se utilizaron en los algoritmos encadenados.

En el Capítulo V "Encadenamiento de Algoritmos" se explican las combinaciones de algoritmos propuestas.

En el Capítulo VI "Experimentos y Resultados" se muestran los resultados de los algoritmos existentes y de los algoritmos propuestos, así como un análisis en cuanto a mejoría entre algoritmos.

En el Capítulo VII "Conclusiones y trabajo futuro" se discuten los resultados obtenidos y se comentan los problemas y propuestas que se podrían considerar como trabajo futuro.

Capítulo II

Marco Teórico

II.1 El Aspecto Biológico

II.1.1 Regulación Génica

La expresión génica es el proceso por medio del cual todos los organismos transforman la información codificada en los ácidos nucleicos en proteínas necesarias para su desarrollo y funcionamiento. En todos los organismos, el contenido del ADN de practicamente todas sus células es idéntico. Esto quiere decir que contienen toda la información necesaria para la síntesis de todas las proteínas. Pero no todos los genes se expresan al mismo tiempo ni en todas las células, esto es debido a la regulación génica. La transcripción es el primer proceso de la expresión génica, mediante el cual se transfiere la información contenida en la secuencia del ADN a secuencias de ARN, donde la timina, es reemplazada por el uracilo (U), de esta manera, la transcripción del ADN también podría llamarse síntesis del ARN (Espinoza, 2004).

II.1.2 Factor de transcripción

Un factor de transcripción es una proteína que participa en la regulación de la transcripción del ADN, pero que no forma parte de la ARN polimerasa. La ARN-polimerasa es un conjunto de proteínas con carácter enzimático capaz de polimerizar los ribonucleótidos para sintetizar ARN a partir de una secuencia de ADN que sirve como patrón o molde. Los factores de transcripción pueden actuar reconociendo y uniéndose a se-

cuencias concretas de ADN también llamadas sitios de pegado o motivos (ver Figura 1), uniéndose a otros factores o uniéndose directamente a la ARN polimerasa (Kodadek, 1998).

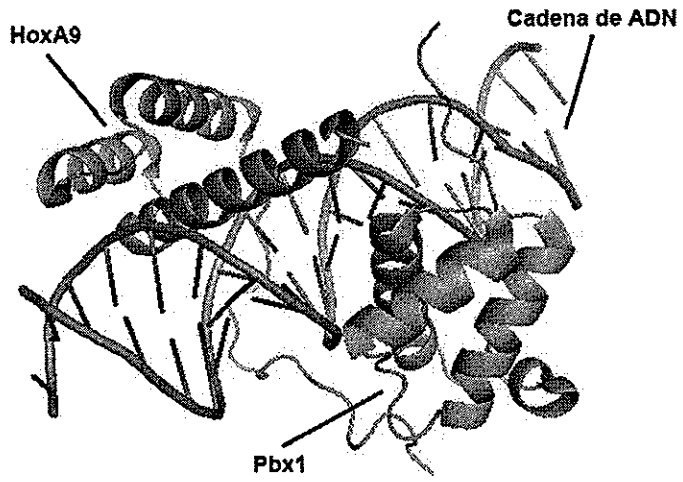


Figura 1. Diagrama ilustrativo del complejo de transcripción 1PUF, que consta de los factores de transcripción HoxA9 y Pbx1, pegado a su secuencia de ADN.

II.1.3 Regiones promotoras

Una región promotora es un segmento de ADN donde las proteínas de unión al ADN, tales como los factores de transcripción, se ligan preferentemente. Estas regiones promotoras, que corresponden a tramos normalmente cortos del ADN, se encuentran posicionadas adecuadamente en el genoma, usualmente a una distancia corta "corriente arriba" del gen que regulan. Dentro de una región promotora, se encuentran los promotores, que son regiones capaces de iniciar la transcripción (inicio de la síntesis del ARN).

Las regiones reguladas comparten características en común a pesar de su obvia divergencia en la secuencia. Muchas de estas características no son directamente evi-

dentes desde la secuencia de nucleótidos sino resultan de las restricciones impuestas por requerimientos funcionales.

La funcionalidad biológica de las regiones promotoras no es generalmente una propiedad uniformemente distribuida sobre la región regulatoria. Unidades funcionales usualmente son definidas por una combinación de estiramientos específicos que pueden delimitar y poseer una propiedad funcional intrínseca (i.e., el pegado de una proteína o una estructura curva de ADN). Muchos tipos funcionales similares de estiramientos de ADN son conocidos y serán referidos como *elementos*. Estos elementos no son ni restringidos a una región promotora ni individualmente suficientes para la función promotora de los denominados promotores ni de los potenciadores. La función de las regiones promotoras completas está compuesta de las funciones de los elementos individuales ya sea en una manera aditiva (elementos independientes) o por efectos de sinergia (módulos). Para más detalles ver Lengauer (2002).

II.1.4 Sitios de pegado de factores de transcripción (TFBS)

Los sitios de pegado para proteínas específicas son lo más importante entre los elementos reguladores. Estos consisten entre alrededor de 10 a 30 nucleótidos, no todos con igualdad de importancia para el pegado de la proteína. Como consecuencia, sitios de pegado de proteínas pueden variar en secuencia, incluso si se trata de la misma proteína. Existen nucleótidos contactados por la proteína de acuerdo a una secuencia en específico, los cuales son usualmente la parte más conservada de varios sitios de pegado del mismo factor de transcripción. Diferentes nucleótidos están envueltos en los contactos de la espina dorsal del ADN y existen espaciadores internos que no entran en contacto con la proteína.

En general, los sitios de pegado muestran suficiente conservación en la secuencia

para permitir la detección de sitios candidatos por una variedad de heurísticas basadas en la similitud de secuencia. Sitios de pegado potenciales pueden ser encontrados en casi todo el genoma y no están restringidos a las regiones promotoras. Se conoce un gran número de sitios de pegado que se encuentran fuera de la región promotora, conocidos como potenciadores y silenciadores, los cuales pueden estimular o debilitar la regulación del gen, dichos sitios de pegado deben localizarse dentro del mismo lazo de cromatina del gen que regulan (Kodadek, 1998).

Otros rasgos en la secuencia que son difíciles de detectar por los métodos computacionales incluyen la señal relativamente débil de posicionamiento de los nucleótidos (Ioshikhes *et al.*, 1996), estiramientos del ADN con estructuras tridimensionales (como el ADN curvo (Sloan, 1998)), señales de metilación, entre otros elementos estructurales.

II.1.5 Elementos estructurales

Las estructuras secundarias del ADN son los elementos estructurales más útiles con respecto al análisis computacional, estas son reconocidas por el ARN y las proteínas, pero también juegan roles importantes en el ADN. Estructuras secundarias potenciales pueden ser fácilmente determinadas y evaluadas a través de la entalpía negativa que debe estar asociada con la formación de la estructura de tipo horquilla (cadena sencilla) o cruciforme (cadena doble).

Las estructuras secundarias tampoco son necesariamente conservadas en secuencias primarias de nucleótidos, pero están sujetas a una fuerte correlación posicional dentro de la estructura. Aspectos tridimensionales de las secuencias de ADN son sin duda muy importantes para la funcionalidad de dichas regiones. Sin embargo, a pesar de los intentos para calcular dichas estructuras en tiempo razonable, dicho cálculo está muy lejos de ser viable para la resolución de problemas reales (Kodadek, 1998).

II.2 El Problema Computacional

Identificar elementos regulatorios, en especial los sitios de pegado de factores de transcripción es una tarea difícil. El descubrimiento de patrones en las secuencias de ADN es uno de los problemas más desafiantes en la biología molecular y en las ciencias de la computación. El problema se define como sigue: dado un conjunto de secuencias, encontrar un patrón desconocido que ocurre frecuentemente. Si un patrón de W letras de largo aparece exactamente en cada secuencia, una simple enumeración de todos los patrones de W letras que aparecen en las secuencias dan la solución. Pero cuando se trabaja con las secuencias de ADN, no es tan simple porque los patrones incluyen mutaciones, inserciones o borrado de nucleótidos.

Un motivo en ADN es definido como un patrón de secuencias de ácidos nucleídos que tiene algún significado biológico tales como los sitios de pegado para una proteína regulatoria, como lo es un factor de transcripción. Normalmente el patrón dentro de un sitio de pegado es corto (de 5 a 20 pares de bases de largo) y es conocido por recurrir en diferentes genes o varias veces en el mismo gen. Las secuencias pueden tener cero, una o múltiples copias de un motivo (denominados también instancias del motivo o sitios de pegado). En adición a las formas comunes de motivos, hay dos tipos especiales de motivos: motivos palindrómicos y motivos espaciados. Un motivo palindrómico es una subsecuencia que es exactamente igual a su complemento reverso, *e.g.*, CACGTG. Un motivo espaciado consiste de dos sitios pequeños conservados, separados por un espacio. El espacio ocurre en la mitad del motivo porque el factor de transcripción se pega como un dímero. Esto significa que el factor de transcripción está hecho de dos subunidades que tienen dos puntos de contacto separados dentro de la secuencia de ADN. Las partes donde los factores de transcripción se pegan al ADN son conservados

pero relativamente pequeños (3 a 5 pares de bases). Estos dos puntos de contacto son separados por un espacio que no es conservado. Dicho espacio puede ser de un largo ligeramente variable.

Dado un conjunto de secuencias de ADN (una región promotora), el problema de la búsqueda de motivos es la de detectar motivos sobrerrepresentados como también motivos conservados sobre secuencias ortólogas que son buenos candidatos de sitios de pegado de factores de transcripción.

Podemos identificar tres casos del problema de descubrimiento de motivos. En el primer caso, el motivo aparece en cada una de las secuencias de entrada con algunas mutaciones. En el segundo caso, el motivo aparece sólo en algunas secuencias presentando algunas mutaciones, y, en el tercer caso, más de una ocurrencia puede aparecer en una sola secuencia, presentando diferentes mutaciones (Sagot, 1998).

II.2.1 Definición Formal del Problema

El problema de búsqueda de motivos se define de la siguiente manera:

Entrada: Un conjunto de N secuencias de ADN provenientes de regiones reguladoras de genes relacionados, conteniendo cada una de ellas cero, una o varias ocurrencias de un motivo (W, e) donde W indica la longitud de éste y e el número máximo de mutaciones que puede tener cada ocurrencia.

Salida: Un conjunto S de posiciones de inicio $\{s_1, s_2, ..., s_n\}$ las cuales indican la primera base de cada una de las n ocurrencias del motivo.

II.2.2 Modelos

Existen diferentes modelos usados para resolver el problema de búsqueda de motivos; el combinatorio, probabilístico y huellas filogenéticas.

Modelo Combinatorio

El modelo combinatorio se plantea de la siguiente forma: Sea p una secuencia desconocida de nucleótidos (el motivo consenso) de longitud W. Se supone que p ocurre una vez en cada una de las N secuencias de fondo cuyas longitudes es l, pero cada ocurrencia de p tiene exactamente e mutaciones en posiciones independientes y aleatorias. Dadas las N secuencias, se desea recuperar las ocurrencias del motivo y el consenso p. Una cadena consenso o *consensus* representa una cadena $c = c_1c_2...c_l$ donde W es la longitud del motivo buscado y c_i la base que más veces aparece en la posición i de todas las ocurrencias identificadas del motivo.

Existen varias funciones de puntaje para evaluar los patrones candidatos p y su correspondiente alineamiento local $O = \{o_1, o_2, ..., o_n\}$:

- Puntaje de distancia de Hamming total (Eskin y Pevzner, 2002):

  $s(p) = \sum d_H(p, o_i)$ para cada $o_i \in O$, o_i es el fragmento de la secuencia i en el alineamiento local O.

- Evaluación general de distancia matricial (Hansen *et al.*, 2006):

  $s(p) = \sum_{j=1}^{W} \sum D(p_j, o_{i,j})$ para cada $o_i \in O$.

  donde W es la longitud del patrón, p_j es la letra del patrón en la posición j, $o_{i,j}$ es la letra en la posición j en la secuencia i, y $D(a, b)$, una entrada en la matriz de distancia D, es la distancia de Hamming entre las letras a y b.

El objetivo del modelo combinatorio es el de minimizar el número total de diferencias en todas las secuencias que expresan la variabilidad de un motivo común. Aunque el modelo combinatorio no necesariamente requiere alineamientos, se puede decir que este enfoque busca encontrar el mejor alineamiento local de un número grande de secuencias.

Varios métodos basados en el modelo combinatorio garantizan optimización global y por lo general son útiles para encontrar motivos cortos y conservados en genes eucariotas donde los mismos son más cortos y conservados que en los procariotas, aunque por lo general, la mayoría de los sitios de pegado en eucariotas son muy degenerados. Es por esto que, los algoritmos combinatorios son incapaces de detectar un gran porcentaje de sitios de pegado en eucariotas (Tompa *et al.*, 2005).

Los métodos combinatorios pueden ser muy rápidos cuando son implementados con estructuras de datos optimizadas, como árboles de sufijos (Sagot, 1998).

Modelo Probabilístico

El objetivo del modelo probabilístico es el de encontrar la mejor ocurrencia del motivo en cada secuencia y calcular un puntaje total para el motivo como la probabilidad de encontrar aleatoriamente otro motivo con un puntaje más alto (Hertz y Stormo, 1999).

Los enfoques probabilísticos involucran la representación del modelo del motivo por una matriz de pesos (Bucher, 1990). Dichas matrices son usualmente visualizadas como un pictograma donde cada posición es representada como una pila de letras, en las cuales su altura es proporcional al contenido de información en dicha posición (Schneider y Stephens, 1990). La matriz de pesos, modela la variación específica de cada posición incluyendo las preferencias cuantitativas entre bases en cada posición. Un motivo está representado por una matriz H de pesos de $4 \times n$, con los renglones indexados por $k \in \{A, C, G, T\}$ y columnas indexadas por $j \in \{1, 2, ..., l\}$, siendo l la longitud de

las secuencias. Cada elemento $H(k, j)$ representa la probabilidad de que el j-ésimo carácter, en cierto sitio de entrenamiento $b \in B'$(B'es un subconjunto de motivos sobre los que es entrenada la matriz), sea la base k.

Un puntaje probabilístico es necesario para determinar la significancia del motivo (Liu *et al.*, 1999), las más utilizadas son:

- ***z-score*** es el número de desviaciones estándar en las cuales el puntaje observado del motivo en las secuencias de entrada excede el puntaje esperado suponiendo que las secuencias de entrada han sido generadas aleatoriamente.

- ***Valor E (E value)*** es el número esperado de modelos encontrados aleatoriamente con un puntaje mayor o igual al del modelo del motivo.

- ***Valor P (P value)*** es la probabilidad de encontrar por casualidad un modelo con puntaje mayor o igual al del modelo del motivo.

Los métodos probabilísticos tienen la ventaja de requerir pocos parámetros pero confían demasiado en los modelos de las regiones promotoras, los cuales pueden ser muy sensibles a pequeños cambios en la información de entrada. Muchos de los algoritmos desarrollados con un enfoque probabilístico son diseñados para encontrar motivos largos o más generales que los requeridos por los sitios de pegado de factores de transcripción, por lo que son más apropiados para la búsqueda de motivos en procariotas. Estos algoritmos no garantizan la solución óptima global, ya que emplean una forma de búsqueda local, como Gibbs sampling, esperanza máxima (EM) o algoritmos voraces que convergen a una solución óptima local.

Modelo Basado en Huellas filogenéticas

Los modelos basados en huellas filogenéticas utilizan secuencias ortólogas. Esto elimina la dificultad de seleccionar regiones correguladas. La premisa simple detrás de las huellas filogenéticas es que la presión selectiva causa que los elementos funcionales evolucionen más despacio que las secuencias no funcionales. Esto significa que usualmente sitios conservados entre un conjunto de regiones promotoras ortólogas son excelentes candidatos para elementos regulatorios funcionales o motivos.

El problema se define como sigue:

Entrada: Un árbol filogenético T, un conjunto de secuencias ortólogas como hojas de T, el largo W del motivo y un umbral ϵ.

Problema: Encontrar un conjunto S de $W - meros$, un $W - mero$ por cada hoja, donde la parsimonia de S en T, sea a lo más ϵ.

La parsimonia consiste en que ante dos hipótesis evolutivas es más probable de ser cierta aquella que implique menos cambios evolutivos, ya que la naturaleza tiende siempre a la simplicidad.

Muchos algoritmos han sido desarrollados basados en huellas filogenéticas. Más recientemente, algoritmos que integran información de la secuencia de ADN de genes corregulados y huellas filogenéticas han mejorado significativamente el descubrimiento de motivos en secuencias genómicas (Das y Dai, 2007).

II.3 Algoritmos para el descubrimiento de motivos

Un gran número de algoritmos para encontrar motivos en ADN han sido desarrollados. Muchos de estos algoritmos están hechos para encontrar motivos considerando la región promotora de genes corregulados de un solo genoma. Se supone que la co-

expresión de genes se presenta debido a la coregulación transcripcional. Debido a que los genes corregulados son conocidos por compartir algunas similaridades en su mecanismo de regulación, posiblemente a nivel transcripcional, sus regiones promotoras pueden contener algunos motivos en común que son sitios de pegado para factores de transcripción. Un acercamiento sensible para detectar estos elementos regulatorios es el de buscar estadísticamente motivos sobrerrepresentados en la región promotora de un conjunto de genes co-expresados. Un motivo estadísticamente sobrerrepresentado es aquel que ocurre más frecuentemente de lo que usualmente se espera. Por lo que estos algoritmos buscan motivos sobrerrepresentados en esta colección de secuencias promotoras. Muchos de estos algoritmos han mostrado tener resultados satisfactorios en la levadura y otros organismos pequeños, pero su desempeño empeora significativamente en organismos más complejos.

Para poder compensar esta dificultad, algoritmos recientes están tomando la ventaja de la comparación de genomas de diferentes especies bajo huellas filogenéticas (Tagle *et al.*, 1988).

Durante los últimos años, se han propuesto enfoques interesantes utilizando casos reales, suponiendo que entre los nucleótidos en el ADN y sus posiciones existe cierta dependencia, esto es, utilizando información de proteínas de la misma familia, de la cual, se conoce de antemano sus motivos (Fatemeh *et al.* (2009), Tomovic *et al.* (2009)).

Basados en el tipo de información de la secuencia de ADN empleada por el algoritmo para deducir los motivos, los algoritmos se clasifican en tres clases: aquellos que usan las secuencias promotoras de genes corregulados de un solo genoma, aquellos que usan secuencias promotoras ortólogas de un solo gen de múltiples especies (i.e., huellas filogenéticas) y aquellos que usan una combinación de ambos. La Tabla XXI (en el Apéndice B) muestra una lista de los algoritmos disponibles a la fecha.

II.3.1 Algoritmos diseñados para secuencias promotoras de genes corregulados

La mayoría de los algoritmos diseñados para encontrar motivos usan un conjunto de secuencias promotoras de genes corregulados para identificar estadísticamente motivos sobrerrepresentados.

Algoritmos combinatorios

Van Helden *et al.* (1998) desarrollaron un algoritmo de búsqueda de motivos llamado Oligo-Analysis basado en un enfoque combinatorio. Aunque conceptualmente es simple, su algoritmo fue eficiente para extraer motivos en *Saccharomyces cerevisiae* (levadura). La metodología usada en el algoritmo incluye la constitución de familias promotoras y el cálculo de las frecuencias esperadas de los nucleótidos.

Tompa (1999) propone un método combinatorio exacto que encuentra motivos cortos en las secuencias de ADN. Su algoritmo fue particularmente aplicado al problema de sitios de pegado de los ribosomas. Tompa tomó en cuenta tanto el número absoluto de ocurrencias y la distribución de fondo y creó una tabla que, para cada $l - mero\ s$ (una secuencia de largo l), guarda el número N_s de secuencias que contienen una ocurrencia de s, donde una ocurrencia permite un número pequeño y fijo d de nucleótidos de sustitución en s. Entonces una medida razonable de s como motivo está basada en qué tan probable es tener N_s ocurrencias cuando las secuencias son seleccionadas aleatoriamente de acuerdo a la distribución de fondo.

Usando un enfoque similar, Sinha y Tompa (2000) desarrollaron el algoritmo YMF (Yeast Motif Finder). Ellos derivaron el modelo de un estudio de sitios de pegado conocidos en la levadura. La entrada del algoritmo son el conjunto de secuencias corri-

ente arriba, el número de caracteres sin espaciadores en los motivos a ser enumerados, y la matriz de transición de una cadena de Markov de orden m construida con todo el complemento de secuencias corriente arriba en la levadura. Ellos condujeron un experimento de validación donde YMF fue usado para identificar sitios de pegado candidatos en 23 regulones (conjunto de genes controlados por un regulador en común) de la levadura. Para 18 de estos regulones, YMF reportó satisfactoriamente los sitios de pegado conocidos para el factor de transcripción principal en los regulones.

Brazma *et al.* (1998) usaron un enfoque combinatorio para desarrollar un algoritmo que busca ocurrencias de patrones regulares de tipo regulatorios. Ellos aplicaron el algoritmo para descubrir patrones en un conjunto de 6000 secuencias tomadas de genes de levadura y patrones en las regiones corriente arriba de genes corregulados en la levadura. Entre los patrones mejor calificados, la mayoría tuvieron coincidencias con motivos conocidos en la levadura.

Sagot (1998) introdujo un enfoque combinatorio el cual está basado en la representación de un conjunto de secuencias con un árbol de sufijos. Marsan y Sagot (2000) extendieron este método para buscar combinaciones de motivos. La representación de secuencias corriente arriba como árboles de sufijos dieron un gran número de posibles combinaciones, pero, la implementación fue muy eficiente. Los algoritmos de búsqueda Weeder y MITRA (Mismatch Tree Algorithm), desarrollados por Pavesi *et al.* (2001) y Eskin y Pevzner (2002) respectivamente, son también basados en árboles de sufijos y variantes. Los algoritmos WINNOWER (Pevzner y Sze, 2000) y cWINNOWER (Liang, 2003) usan enfoques combinatorios de teoría de grafos para la búsqueda de motivos.

Algoritmos probabilísticos

Uno de los primeros algoritmos para encontrar una representación matricial de un sitio de pegado fue un modelo voraz probabilístico propuesto por Hertz *et al.* (1990). El algoritmo encuentra el sitio con la mayor información de contenido (IC). Ellos usaron este algoritmo para identificar motivos comunes que estaban presentes una vez en cada secuencia. Este algoritmo ha sido sustancialmente mejorado con los años y en su última implementación (Consensus) Hertz y Stormo (1999) proveen un método para estimar la significancia estadística de un puntaje de IC.

Down y Hubbard (2005) desarrollaron un algoritmo llamado NestedMICA. Utiliza una técnica Monte Carlo llamada *Nested Sampling*, con modelos de secuencia de fondo multi-clase para representar diferentes partes irrelevantes de las secuencias que no contienen motivos de interés.

La gran mayoría de los modelos probabilísticos aplican técnicas tales como EM y Gibbs sampling o extensiones de estas.

Métodos basados en EM

EM para búsqueda de motivos fue introducido por Lawrence y Reilly (1990) y fue una extensión del algoritmo voraz propuesto por Hertz *et al.* (1990). Fue desarrollado para motivos en proteínas, pero también puede ser aplicado para búsqueda de motivos en ADN. Ningún alineamiento de los sitios es requerido y el modelo básico supone que cada secuencia debe de contener al menos un sitio en común. La incertidumbre de la localización de los sitios es controlada al emplear el principio de información perdida para desarrollar el algoritmo EM. En MEME de Bailey y Elkan (1995) se extiende el algoritmo EM para identificar motivos en secuencias no alineadas. Bailey *et al.* (2010) agregaron a MEME la opción de utilizar información adicional en la búsqueda

de motivos en caso de que se conozca la preferencia de pegado de algún factor de transcripción en específico.

Métodos basados en Gibbs Sampling

Entre los métodos probabilísticos, el de Gibbs sampling ha sido usado extensivamente para algoritmos de búsqueda de motivos, desarrollado por Lawrence *et al.* (1993). Ellos no aplicaron este algoritmo al ADN, pero si a secuencias de proteínas. Como una de las suposiciones iniciales del algoritmo fue que debía existir al menos una instancia por motivo en cada secuencia, el método también es llamado *sitesampler*. Gibbs sampler es un enfoque de cadenas de Markov tipo Monte Carlo (MCMC). Cadenas de Markov porque los resultados de cada paso dependen solo de los resultados del paso anterior, y Monte Carlo porque la forma de seleccionar el siguiente paso no es determinístico si no basado en muestreo aleatorio.

Extensiones al método de Gibbs Sampling

Basado en la estrategia de Gibbs Sampling, Roth *et al.* (1998) desarrollaron el algoritmo AlignACE (*Aligns Nucleic Acid Conserved Elements*). Este algoritmo regresa una serie de motivos (como matrices de peso) que son sobrerrepresentados en el conjunto de entrada de las secuencias de ADN. Las diferencias con el algoritmo de Gibbs son las siguientes: Hace una búsqueda en ambas hebras de ADN simultáneamente, la búsqueda simultánea de diferentes motivos es reemplazada por un enfoque en el cual motivos simples son encontrados e iterativamente enmascarados. AlignACE usa el MAP ($maximum - a - priori - log - likelihood$) para calificar los diferentes motivos.

Thijs *et al.* (2001) desarrollaron el algoritmo MotifSampler usando una modificación de Gibbs Sampling. Los cambios fueron el uso de una distribución probabilística para

estimar el número de copias del motivo en una secuencia y la incorporación de un modelo de alto orden de cadenas de Markov.

Usando una estrategia Gibbs, Liu *et al.* (2001) desarrollaron el algoritmo Bio-Prospector que usa las regiones promotoras de genes corregulados. Las diferencias con el Gibbs Sampling original es que usa un modelo de Markov de orden cero a orden tres cuyos parámetros son dados por el usuario o estimados desde una secuencia especificada, la significancia de cada motivo es juzgada basándose en un puntaje de distribución estimado por un método de Monte Carlo, y permite el modelado de motivos espaciados y motivos palindrómicos. Los autores encontraron satisfactoriamente motivos para la proteína de pegado RAP1 en la levadura, la caja TATA en *Bacillus subtilis* y el sitio de pegado de la proteína CRP en *Escherichia coli.*

Métodos de aprendizaje de máquina

Liu *et al.* (2004) desarrollaron un algoritmo FMGA basado en algoritmos genéticos (GAs) para encontrar motivos potenciales en regiones localizadas a partir de -2000 pares de bases corriente arriba hasta +1000 pares de bases corriente abajo del sitio de inicio de transcripción. La mutación en el GA es realizada al usar una matriz de peso de posiciones para reservar completamente las posiciones conservadas. El cruzamiento es implementado con un diseño especial para castigar espacios para producir el patrón hijo óptimo. Este algoritmo usa un método de re-arreglo basado en matrices de peso de posiciones para evitar la presencia de un mínimo local estable, lo cual puede hacer difícil para otros operadores generar un patrón óptimo. Los autores probaron su algoritmo para identificar motivos de ADN en *E.coli* y reportaron mejores resultados al compararlo con MEME y Gibbs sampler.

Liu *et al.* (2006) desarrollaron un algoritmo utilizando una estructura de redes neu-

ronales para la búsqueda de motivos en ADN y en secuencias de proteínas. La red contiene varias capas, las cuales cada una realiza una clasificación en un nivel diferente. Los autores mantuvieron una complejidad computacional baja al usar la estructura con capas, de tal forma que cada clasificación de patrones es realizada con respecto a un subespacio pequeño de todo el espacio de entrada. De resultados en simulación, los autores reportaron que su algoritmo fue mejor que MEME y Gibbs Sampler en ciertos aspectos y que su algoritmo funciona bien para secuencias largas de ADN.

Otros métodos

Kingsford *et al.* (2006) usó programación matemática para el descubrimiento de motivos en ADN. Utilizaron programación lineal entera que explota la naturaleza discreta de la distancia impuesta entre pares de subcadenas. Como encontrar una solución a la programación lineal entera es computacionalmente difícil, los autores usaron una relajación lineal al agregar un conjunto exponencial de limitantes y usaron un algoritmo de separación eficiente que puede encontrar las restricciones violadas y en consecuencia tener una solución en tiempo polinomial.

Kaplan *et al.* (2005) usó un modelo basado en la estructura. Ellos combinaron información de la secuencia de ADN y la información estructural del factor de transcripción para inferir un contexto específico de los aminoácidos y determinar las preferencias de reconocimiento entre un nucleótido y un aminoácido.

Mientras que muchos algoritmos de búsqueda de motivos han sido probados satisfactoriamente con la levadura y otros organismos, la mayoría tiene resultados malos en organismos más complejos (Tompa *et al.*, 2005). Hon y Jain (2006) desarrollaron un algoritmo determinístico con aplicación al genoma humano. Este método depende de una técnica de indexado que optimiza el proceso de búsqueda. El procedimiento de

búsqueda funciona con una función de puntaje muy simple que combina una preferencia de conservación entre las secuencias de entrada con una preferencia a las secuencias poco-representadas relativas al genoma.

II.3.2 Algoritmos diseñados para huellas filogenéticas

La mayor ventaja de las huellas filogenéticas sobre los métodos de genes corregulados es que este último requiere de un método confiable para identificar genes corregulados. Mientras que, usando huellas filogenéticas es posible identificar motivos específicos incluso dentro de un solo gen, mientras se encuentren lo suficientemente conservados a través de todas las secuencias ortólogas consideradas. La rápida acumulación de secuencias genómicas de una gran diversidad de organismos hace que sea posible utilizar esta técnica para la búsqueda de motivos. El método estándar usado en huellas filogenéticas es el de construir un alineamiento múltiple global de la región promotora ortóloga e identificar la región conservada en el alineamiento usando herramientas como CLUSTAL W (Thompson *et al.*, 1994). Sin embargo, se ha observado (Cliften *et al.*, 2001; Blanchette y Tompa, 2002; Tompa, 2001) que este método no siempre funciona. La razón es que si las especies están cercanamente relacionadas, el alineamiento es obvio pero insignificante, ya que los elementos funcionales no son lo suficientemente conservados como la región no funcional a su alrededor. Por otro lado, si las especies están lejanamente relacionadas, es difícil o imposible encontrar un alineamiento significativo. Para intentar eliminar este inconveniente, muchos de los algoritmos ya existentes como MEME, Consensus, Gibbs sampler han sido usados para huellas filogenéticas.

Cliften *et al.* (2001) usaron AlignACE para la búsqueda de motivos al comparar la secuencia de ADN de diferentes especies de *Saccharomyces cerevisiae* y reportaron algunos resultados exitosos donde las herramientas de alineamiento múltiple habían fal-

lado. McCue *et al.* (2001) usaron Gibbs sampler para la búsqueda de motivos en genomas proteobacteriales. Blanchette y Tompa (2002) mostraron que el uso de métodos de huellas filogenéticas puede ser problemático, ya que dichos algoritmos no toman en cuenta la relación filogenética entre las secuencias, debido a que dichos métodos suponen que las secuencias de entrada son independientes.

Cliften *et al.* (2003) usaron el método de huellas filogenéticas para encontrar motivos en genomas de *Saccharomyces cerevisiae.* Ellos buscaron huellas filogenéticas entre las secuencias genómicas de seis especies de *Saccharomyces* utilizando CLUSTAL W. Usando este simple alineamiento ellos pudieron encontrar muchas secuencias conservadas de motivos estadísticamente significativos.

Berezikov *et al.* (2004) publicaron el algoritmo CONREAL basado en huellas filogenéticas. Dicho algoritmo usa motivos potenciales representados por una matriz de pesos posicional para establecer anclas entre las secuencias ortólogas y para guiar el alineamiento de la secuencia promotora.

Wang y Stormo (2005) desarrollaron el algoritmo PHYLONET que sistemáticamente identifica filogenéticamente motivos conservados al analizar todas las secuencias promotoras de genomas relacionados y define una red de sitios reguladores para el organismo. Este algoritmo construye perfiles filogenéticos para cada promotor y entonces utiliza un algoritmo similar a BLAST para buscar eficientemente a través de todo el espacio de perfiles de todos los promotores en el genoma para identificar motivos conservados y los promotores que los contienen.

Carmack *et al.* (2007) desarrollaron un algoritmo de escaneo, PhyloScan, el cual combina evidencia de sitios que coinciden encontrados en información ortóloga de varias especies relacionadas con evidencia de sitios múltiples dentro de una región intrínseca. Las secuencias ortólogas pueden ser alineadas múltiplemente, no alineadas o una com-

binación de ambas. Si es alineada, PhyloScan cuenta estadísticamente la dependencia filogenética de la información contribuyente de las especies al alineamiento y, si no es alineada, la evidencia de los sitios es combinada suponiendo independencia filogenética de las especies.

II.3.3 Algoritmos diseñados para secuencias promotoras de genes corregulados y huellas filogenéticas

Estos algoritmos integran dos aspectos importantes de la significancia de los motivos, la sobrerrepresentación y la conservación entre especies, todo dentro de un puntaje probabilístico. Gelfand *et al.* (2000) usa promotores de genes corregulados y promotores ortólogos para la búsqueda de motivos sobrerrepresentados en *Archaea*. Ellos utilizaron el algoritmo de Smith-Waterman (Smith y Waterman, 1981) para la identificación de señales, construcción de perfiles de reconocimiento, identificación de señales candidatas y búsquedas de similitud proteínica.

Un algoritmo desarrollado por Kellis *et al.* (2003) extrae motivos en dos pasos desde la información mixta de los dos tipos de secuencias. Primero el algoritmo encuentra un conjunto de motivos altamente conservados, y en el segundo paso los motivos sobrerrepresentados son extraídos.

Prakash *et al.* (2004) desarrollaron el algoritmo OrthoMEME basado en un enfoque EM que usa secuencias de genes corregulados y ortólogos. Este algoritmo busca el espacio de los motivos y los alineamientos de los motivos simultáneamente. Cada ocurrencia de motivo se supone tiene una copia ortóloga en las otras especies, los cuales pueden ser localizadas en cualquier parte del promotor correspondiente. OrthoMEME está diseñado para usar secuencias ortólogas de dos especies.

Basado en el algoritmo Consensus (Hertz *et al.*, 1990), Wang y Stormo (2003) desarrollan el algoritmo PhyloCon que toma en consideración tanto la conservación entre los genes ortólogos como la corregulación de los genes entre especies. Este algoritmo primero alinea las regiones conservadas de las secuencias ortólogas usando alineamientos múltiples o perfiles, y después compara los perfiles que representan las secuencias no ortólogas.

Sinha *et al.* (2004) desarrollaron el algoritmo PhyMe basado en un método probabilístico que maneja información de promotores de genes corregulados y secuencias ortólogas. Una característica importante de este algoritmo es que permite a los motivos ocurrir en regiones conservadas o no conservadas dentro de los promotores ortólogos.

Moses *et al.* (2004) desarrollaron EMnEm, un algoritmo que utiliza EM y un modelo filogenético para encontrar motivos de información que envuelven genes corregulados y secuencias ortólogas. Este algoritmo supone que las secuencias de entrada están completamente alineadas, pero, tal suposición no es aplicable para especies que están relativamente a una distancia evolutiva grande, tal como el humano y el ratón.

Siddharthan *et al.* (2005) desarrollaron el algoritmo denominado PhyloGibbs que combina las estrategias de búsqueda de las huellas filogenéticas y de Gibbs *sampling*, dentro de un solo marco Bayesiano. Dicho algoritmo busca sobre todos los arreglos en los cuales un número arbitrario de sitios de pegado para un número arbitrario factores de transcripción pueden ser asignados a los alineamientos múltiples de secuencias. Estas configuraciones de sitios de pegado son evaluadas por un modelo Bayesiano que trata a las secuencias alineadas como un modelo evolutivo de los sitios de pegado y la información intergénica de fondo del ADN.

II.4 Encadenamiento de algoritmos para la búsqueda de motivos

Un encadenamiento de algoritmos está compuesto por una serie de algoritmos en serie o en paralelo y tienen como objetivo un incremento en rendimiento y exactitud de resultados de sus componentes. Para la búsqueda de motivos, se han propuesto algunos encadenamientos para organismos y casos muy específicos.

Doi *et al.* (2008) proponen un encadenamiento de algoritmos para encontrar motivos en regiones promotoras del arroz, el cual consta de cuatro algoritmos diferentes, el primero que identifica los genes corregulados que el usuario introduce, después se construyen regiones cis-regulatorias al alinear motivos ya conocidos en la base de datos AGRIS, también se utiliza el algoritmo MEME para buscar motivos desconocidos, y por último, se evalúan los motivos con el algoritmo BLASTN utilizando la base de datos RiceTFDB.

Weinberg *et al.* (2007) proponen un encadenamiento para la búsqueda de motivos en ARN en *lacto bacilos* y en genes de *Burkholderiales*, donde reportan haber encontrado 22 motivos, en los cuales, 2 han sido confirmados como verdaderos.

Hu *et al.* (2006) proponen el algoritmo EMD, donde se ejecutan 5 algoritmos en paralelo (AlignACE, BioProspector, MDScan, MEME y MotifSampler). Fue probado en secuencias del *E.coli*, donde se reporta una mejoría en sensibilidad de 22.4% sobre el mejor algoritmo componente, aunque la cantidad de falsos positivos se triplica en el mejor de los casos.

Capítulo III

Evaluaciones de los algoritmos de búsqueda de motivos

III.1 Antecedentes

En la actualidad, un gran número de algoritmos para la búsqueda de motivos se encuentran disponibles, por lo que es deseable tener alguna guía sobre cómo escoger las mejores herramientas para realizar dicha búsqueda. Un obstáculo para obtener esta guía es la gran dificultad que existe para conducir pruebas de rendimiento y de comparaciones entre las herramientas de búsqueda disponibles. Uno de los pasos importantes en el análisis comparativo de buscadores de motivos es el de seleccionar un conjunto útil de casos de prueba. También la inclusión de muchas secuencias incorrectas (por ejemplo, que no contengan promotores) puede prevenir cualquier análisis significativo.

Secuencias aleatorias pueden ser generadas con facilidad, pero usualmente su uso es limitado, ya que no representan características importantes del ADN. Esto incluye características que casi no están representadas (por ejemplo, islas CpG), características asimétricas, o elementos repetitivos de ADN. La selección de secuencias de control apropiadas puede ser un reto mayor, pero también son cruciales para la validez de la evaluación de cualquier método.

Como se menciona en el trabajo de Tompa *et al.* (2005), la dificultad de la evaluación de los algoritmos viene de diferentes fuentes. Las herramientas han sido desarrolladas sobre una variedad de modelos y con diferente complejidad, por lo que, herramientas

individuales pueden tener mejores resultados en un tipo de información pero tener peores en otro tipo. También nuestro entendimiento biológico aún incompleto sobre el mecanismo regulador no siempre ayuda a realizar una evaluación adecuada sobre los algoritmos de búsqueda.

La mayoría de los autores prueban sus algoritmos contra otros algoritmos disponibles utilizando secuencias biológicas con información sintética como motivos implantados. Pevzner y Sze (2000) compararon su algoritmo SP-STAR con los algoritmos Consensus (Hertz y Stormo, 1999) y MEME (Bailey y Elkan, 1995), y reportando que su algoritmo tuvo mejores resultados en motivos pequeños. Sinha y Tompa (2003) compararon la exactitud de tres algoritmos: YMF (Sinha y Tompa, 2000), MEME (Bailey y Elkan, 1995) y AlignACE (Roth *et al.*, 1998). El criterio de evaluación que utilizaron fue: el número de posiciones, sobre todas las secuencias donde las ocurrencias de los motivos conocidos se traslapaban, dividido por el número total de posiciones en las cuales el motivo conocido ocurre. La comparación fue utilizando información sintética con motivos implantados, así como en conjuntos de información real de genes corregulados de *S. cerevisiae*.

Tompa *et al.* (2005) evaluaron el rendimiento de 13 algoritmos de búsqueda de motivos. El propósito de su evaluación fue: proveer una guía sobre la exactitud de las herramientas que están disponibles actualmente, y el de proveer un *benchmark* de conjuntos de secuencias para evaluar futuras herramientas. Basado en el hecho de que poco es conocido sobre la mayoría de los factores de transcripción y sus sitios de pegado, incluso en organismos muy bien estudiados, los autores incluyeron aquellas herramientas computacionales diseñadas para encontrar elementos reguladores nuevos, donde nada se supone a priori acerca del factor de transcripción y de su sitio de pegado. Para estas herramientas, un usuario provee una colección de regiones reguladoras de genes donde se

cree que son correguladas, y la herramienta identifica motivos que son estadísticamente sobrerrepresentados en dichas regiones reguladoras. Utilizaron diferentes estadísticas para evaluar el rendimiento de cada herramienta en cada conjunto de prueba a nivel nucleótido así como a nivel de posición utilizando la información de los sitios de pegado conocidos y del conjunto de sitios de pegado predichos por la herramienta. Los autores advirtieron que esta evaluación no debe ser tomada como una evaluación precisa por una variedad de razones. La más importante es que no se entiende completamente el mecanismo regulador. Por lo que aún no se tiene un estándar para comparar la exactitud de las herramientas.

Hu *et al.* (2005) también generaron un *benchmark* para comparaciones de predicciones utilizando secuencias del *E.coli* obtenidas de la base de datos RegulonDB (Gama-Castro *et al.*, 2010). Los autores señalan cómo su trabajo difiere con respecto al realizado por Tompa *et al.* (2005), ya que en dicho estudio, a los algoritmos se les permitió un ajuste de parámetros y que únicamente se reportaron los mejores resultados, mientras que Hu *et al.* (2005) únicamente permitieron ajustes mínimos durante sus comparaciones. También sugieren que la evaluación de la exactitud basado solo en las predicciones con mejor calificación tiene el riesgo de penalizar algunos algoritmos prácticamente efectivos, ya que en muchos casos los motivos predichos con la más alta calificación no son siempre los motivos con la mayor exactitud.

III.2 Criterios para la evaluación de algoritmos

Algunos criterios han sido propuestos para la evaluación de algoritmos de búsqueda de motivos, dichos criterios son descritos en Tompa *et al.* (2005) detalladamente:

Para un algoritmo T y para un caso de prueba D, tenemos un conjunto de sitios de

pegado conocidos y un conjunto de sitios de pegado predichos. La calificación de T en D puede ser evaluada tanto a nivel de nucleótidos como a nivel de sitio. Específicamente, en nivel de nucleótidos se define verdaderos positivos (nTP), falsos negativos (nFN) y otros como sigue:

- nTP es el número de posiciones de nucleótidos predichos que coinciden con los verdaderos.
- nFN es el número de posiciones de nucleótidos en sitios conocidos que no son sitios predichos.
- nFP es el número de posiciones de nucleótidos que no están en sitios conocidos pero sí en sitios predichos.
- nTN es el número de posiciones de nucleótidos que no están en sitios conocidos pero tampoco son predichos como tales.

Se dice que un sitio predicho traslapa con un sitio conocido si su traslape es al menos una cuarta parte del largo del sitio conocido. A nivel del sitio definimos:

- sTP es el número de sitios conocidos que traslapan al menos en 40% de la longitud del motivo con sitios predichos.
- sFN es el número de sitios conocidos que no traslapan con los sitios predichos.
- sFP es el número de sitios predichos que traslapan en menos del 40% con sitios conocidos.

Tanto a nivel de nucleótido ($x = n$) o a nivel de sitio ($x = s$), se define:

- Sensibilidad: $xSn = xTP/(xTP + xFN)$

- Valor predictivo positivo: $xPPV = xTP/(xTP + xFP)$

La sensibilidad nos da la fracción de sitios conocidos (o nucleótidos) que son predichos correctamente, y el valor predictivo positivo nos da la fracción de los sitios predichos (o nucleótidos) que coinciden con los conocidos.

A nivel nucleótido también se puede definir:

- Especificidad: $nSP = nTN/(nTN + nFP)$

Se define el coeficiente de rendimiento a nivel de nucleótido como:

- $nPc = nTP/(nTP + nFN + nFP)$

El coeficiente de correlación a nivel de nucleótido:

- $nCC = \frac{nTP \cdot nTN - nFN \cdot nFP}{\sqrt{(nTP+nFN)(nTN+nFP)(nTP+nFP)(nTN+nFN)}}$

Y por último, el rendimiento promedio del sitio (a nivel de sitio):

- $sASP = (sSn + sPPV)/2$

El coeficiente de correlación nCC mide la correlación entre el conjunto de sitios conocidos y predichos. El valor de nCC está en el intervalo de -1 (indicando perfecta anti-correlación) y $+1$ (indicando perfecta correlación). Por lo que si los motivos predichos coinciden exactamente con los sitios conocidos, nCC será $+1$. Si cada posición de los nucleótidos es aleatoria e independiente, entonces el valor esperado de nCC será de 0, indicando ninguna correlación.

Para los casos de prueba negativos, es decir, aquellos que no contienen sitios de pegado, $TP + FN = 0$.

III.3 Casos de prueba en la literatura

Los casos de prueba por el tipo de organismo se clasifican en procariotas (bacterias) y eucariotas (animales, plantas, hongos, protozoos y algas). Por el tipo de secuencia, los casos de prueba se clasifican en casos reales y ficticios.

Los casos reales consisten en secuencias promotoras de procariotas o eucariotas. Dichos casos pueden ser pre-procesados, truncando su longitud y eliminando motivos espaciados y poco conservados.

Los casos ficticios consisten en secuencias no promotoras, generadas computacionalmente o tomadas de genes reales con motivos implantados. Dichos motivos pueden ser reales o ficticios.

Tompa *et al.* (2005) crearon conjuntos de secuencias de prueba conteniendo sitios de pegado conocidos para evaluar las herramientas. Para los sitios de pegado utilizaron la base de datos TRANSFAC (Wingender *et al.*, 1996) para escoger factores de transcripción reales, sus sitios de pegado reales, y las posiciones y orientación de los mismos. Cada factor de transcripción tiene un conjunto de secuencias. Cada conjunto consiste de tres tipos diferentes de secuencias, con los sitios de pegado implantados en sus posiciones y orientaciones conocidas. Los tres tipos de secuencia son: las secuencias promotoras reales, promotores del mismo genoma seleccionados aleatoriamente, y secuencias generadas por cadenas de Markov de orden 3. En total generaron 164 casos de prueba, los cuales consisten de sitios de pegado del hombre, mosca, ratón y levadura. En la Tabla I se muestra la longitud máxima de las secuencias (Long. Max.), la longitud mínima (Long. Min.), el número máximo de secuencias por caso de prueba (Max. Sec.), el número mínimo de secuencias (Min. Sec), y el número de casos de prueba totales por organismo.

Hu *et al.* (2005) generaron un *benchmark* utilizando secuencias del *E.Coli* obtenidas de la base de datos RegulonDB (Gama-Castro *et al.*, 2010). En total generaron 61 casos de prueba únicos. Para cada uno de estos casos de prueba, se generaron otros 8 casos de prueba de diferente longitud. De esta forma, es fácil evaluar la sensibilidad de cada algoritmo con secuencias de diferente tamaño al utilizar el mismo caso de prueba.

Geir *et al.* (2007) generaron 50 casos de prueba reales de eucariotas utilizando la base de datos TRANSFAC (Wingender *et al.*, 1996). Después pre-procesaron dichos casos de prueba al truncarlos a un máximo de 2000 nucleótidos por secuencia y eliminaron los motivos que contenían bases degeneradas, dos o más ocurrencias dentro de la región de 2000 nucleótidos y aquellos motivos que contenían espacios. También utilizando los sitios de pegado de dichos casos de prueba, generaron cadenas ficticias utilizando modelos ocultos de Markov e insertaron los sitios de pegado en la misma posición en la que ocurren originalmente. De tal forma que crearon un benchmark con un total de 150 casos de prueba (50 reales, 50 reales pre-procesados y 50 ficticios).

Tabla I: Casos de prueba de Tompa *et al.* (2005).

Organismo	Long. Min.	Long. Max.	Min. Sec.	Max. Sec.	No de Casos
Mosca	1500	2500	1	4	22
Humano	500	3000	2	35	78
Ratón	500	1500	2	13	36
Levadura	500	1000	3	16	28

Capítulo IV

MEME, BioProspector, Weeder y RepeatMasker: Una breve descripción

IV.1 Maximización de la Esperanza

Lawrence y Reilly (1990) introdujeron el método de maximización de la esperanza como una forma de resolver el problema de aprendizaje de motivos. Su algoritmo toma un conjunto de secuencias y una longitud de motivo W, y regresa un modelo probabilístico del motivo en común. La idea detrás de este método es que cada secuencia dentro del conjunto comparte un motivo. Se supone que la posición del motivo en común es desconocida. Debido a que las secuencias de entrada no se encuentran alineadas, se debe estimar la probabilidad de que un motivo compartido se encuentra en la posición j dentro de la secuencia i. Estas probabilidades estimadas, z_{ij}, son usadas para reestimar la probabilidad de la letra l en la columna c en el motivo p_{lc}, para cada letra l en el alfabeto y $1 \leq c \leq W$. El algoritmo EM alternadamente reestima z y p hasta que p cambie muy poco de iteración a iteración. (La notación z es usada para referir a la matriz de probabilidades z_{ij}. De la misma forma p refiere a la matriz de probabilidades de letras p_{lc}).

El algoritmo EM se muestra a continuación, donde inicia de un estimado de los parámetros del modelo p, dados por el usuario o generados aleatoriamente.

EM(secuencias, W){

1. Se elige una probabilidad inicial p

2. Hacer{
3. Reestimar z desde p
4. Reestimar p desde z
5. }Hasta que el cambio en p sea menor que ϵ
6. Retorno
7. Fin
8. }

El algoritmo EM simultáneamente descubre un modelo del motivo (la secuencia de variables independientes discretas y aleatorias con parámetros p) y estima la probabilidad de cada posible punto de inicio de motivos en las secuencias (z). Por definición, la verosimilitud del modelo dado el conjunto de datos de entrenamiento es la probabilidad del conjunto de datos dado el modelo. El algoritmo EM encuentra valores de los parámetros del modelo que maximizan la verosimilitud esperada del conjunto de datos dado el modelo p, y la pérdida de información z. El logaritmo de la verosimilitud está dado por:

$$log(verosimilitud) = N\sum_{j=1}^{W}\sum_{l\in\alpha} f_{lj}log(p_{lj}) + N(L-W)\sum_{l\in\alpha} f_{l0}log(p_{l0}) + Nlog(\frac{1}{L-W+1}) \quad (1)$$

donde N es el número de secuencias en el conjunto de prueba, L es la longitud de las secuencias, W es la longitud del motivo en común, α es el alfabeto, p_{lj} es la probabilidad (desconocida) de la letra l en la posición j del motivo, p_{l0} es la probabilidad (desconocida) de la letra l en todas las posiciones que no contienen motivos, f_{lj} es la frecuencia observada de la letra l en la posición j del motivo, y f_{l0} es la frecuencia observada de l en todas las posiciones que no contienen motivos dentro de las secuencias.

IV.2 MEME

El algoritmo MEME (Bailey y Elkan, 1995) es una versión modificada del algoritmo EM, tiene como objetivo descubrir la posición de ocurrencia de un motivo en una secuencia de ADN y la descripción de dicho motivo. MEME posee tres modelos:

- OOPS: Supone que hay exactamente una ocurrencia del motivo por secuencia.
- ZOOPS: Supone que hay a los más una ocurrencia del motivo por secuencia.
- ANR: No supone nada acerca de la cantidad de ocurrencias por secuencia del motivo.

MEME es una combinación de las siguientes herramientas:

- Maximización de la Esperanza (EM).
- Un método heurístico para escoger la solución inicial para el algoritmo EM.
- Un estimador de Máxima Verosimilitud.

Dentro de MEME, se encuentra un ciclo que contiene un algoritmo EM que es ejecutado repetidamente con diferentes posiciones de inicio. Los puntos de inicio son derivados de las subsecuencias las cuales ocurren dentro de las secuencias de entrada. EM es ejecutado únicamente una vez, sin converger, para cada punto de inicio para ahorrar tiempo. Cada corrida de EM produce un modelo probabilístico de un posible motivo compartido. El punto de inicio con la mayor verosimilitud es elegido y EM se ejecuta hasta su convergencia a partir de este punto inicial. El modelo del motivo compartido es desplegado. Finalmente, todas las apariciones del motivo compartido dentro de las secuencias de entrada son borradas. El siguiente ciclo repite nuevamente todo el proceso para descubrir nuevos motivos compartidos.

EM sufre del problema de quedarse en óptimos locales (Redner y Walker, 1984). Una manera de resolver este problema es volver a correr EM repetidamente de diferentes puntos de partida y escoger al final el modelo que tenga una mayor puntuación. Otra alternativa más rápida es encontrar un buen punto de partida para el método y correr EM para que converja desde ahí. MEME usa la idea anterior, es decir, primero encuentra un buen punto de partida antes de iniciar EM, hecho vía un algoritmo de programación dinámica, el cual estima un buen punto de partida simultáneamente a partir de muchos posibles puntos de partida.

A continuación se muestra el algoritmo descrito:

MEME(S, W, N – *Sitios*, *NumIter*){

1. Para i=1 hasta *NumIter*{
2. Para cada subsecuencia $s \in S$ {
3. Correr **EM** una vez con el punto de inicio derivado de esta subsecuencia.
4. Escoger el motivo compartido con la verosimilitud mayor.
5. Correr **EM** hasta su convergencia con el punto de inicio generado.
6. Desplegar el modelo del motivo compartido.
7. Eliminar las apariciones del motivo compartido de las secuencias de entrada.
8. }
9. }
10. }

Donde S es el conjunto de secuencias de entrada, W la longitud del motivo, $N - Sitios$ el número de sitios de pegado por motivo, y $Numiter$ el número de motivos compartidos a encontrar.

La salida de MEME incluye una matriz de probabilidad logarítmica, *spec*. Dicha matriz tiene L renglones y W columnas y es calculada como $spec_{ij} = log(p_{ij}/p_{0j})$ para $i \in \alpha$ y $j = 1, ..., W$. El puntaje de contenido de información de una subsecuencia es calculado al sumar las letras correspondientes en la matriz. Este puntaje da una medida de la probabilidad de que la subsecuencia sea una instancia del motivo o una instancia de la distribución de fondo.

MEME permite al usuario especificar el tipo de modelo a usar. El modelo $uno - por$ supone que en cada secuencia de entrada hay exactamente una ocurrencia del motivo. Esta suposición determina la forma en cómo se reestiman las probabilidades. El procedimiento de reestimación asegura que la compensación de probabilidad para cada secuencia sume 1.0. Esto significa que si una secuencia dada tiene más de una ocurrencia del motivo, su contribución a la reestimación de la frecuencia de letras es igual que la de una secuencia con una sola aparición.

Para el modelo $n - por$, MEME modifica el algoritmo EM. En lugar de normalizar la reestimación de probabilidades para que sume 1.0 para cada secuencia, las probabilidades son normalizadas para sumar a un valor dado por el usuario. Esta normalización es realizada sobre todas las secuencias simultáneamente. El intento es que $N - Sitios$ sea el número esperado de ocurrencias del motivo en las secuencias de entrada. Si $N - Sitios$ es igual al número de secuencias en el conjunto de entrada, es posible que el modelo $n - por$ obtenga los mismos resultados que el modelo $uno - por$. Dicho modelo beneficia a las secuencias con múltiples apariciones del motivo. Cuando $N - Sitios$ es menor que el número de secuencias de entrada, MEME asigna una probabilidad baja a

todas las posiciones en una secuencia que no contenga el motivo. En contraste, el modelo *uno – por* debe asignar probabilidades que sumen 1.0 en cada secuencia de entrada. MEME tiene una complejidad computacional de $O((NM)^2W)$ (Bailey y Elkan, 2005) donde N es el número de secuencias en el caso de prueba, M es la longitud promedio de las secuencias, y W la longitud del motivo.

IV.3 BioProspector

BioProspector (Liu *et al.*, 2001) es un algoritmo que examina las regiones corriente arriba de los genes buscando secuencias regulatorias (motivos). BioProspector utiliza modelos de fondo de Markov de orden cero a orden tres cuyos parámetros son dados por el usuario o son estimados desde una secuencia especificada. La significancia de cada motivo encontrado es juzgada basado en una calificación de distribución de motivos estimado por un método Monte Carlo. Además, BioProspector modifica el modelo de motivos utilizado anteriormente por *Gibbs sampler* (Lawrence *et al.*, 1993) para permitir el modelado de motivos espaciados y motivos con patrones palindrómicos.

IV.3.1 Modelo básico

BioProspector busca motivos en un conjunto de secuencias de ADN. Toma los siguientes parámetros de entrada:

- N secuencias de ADN en donde se encuentran los motivos a buscar.
- El largo de los bloques de motivos w_1 y w_2 y su intervalo de espaciado $[g_L, g_M]$.
- Un indicador de si cada secuencia tiene al menos una copia del motivo.
- Un indicador de si el motivo puede ocurrir en ambas hebras del ADN.

- Un indicador de si el motivo tiene un patrón palindrómico, en cuyo caso w_1 debe ser igual a w_2.

Al finalizar, BioProspector tiene como salida la siguiente información:

- La calificación del motivo y el número de segmentos alineados.
- Una expresión regular consensus del motivo, como también la matriz de probabilidad del motivo.
- El número de segmentos de cada secuencia de entrada que contribuyen al motivo, la posición inicial y la secuencia de cada segmento.

En cada corrida de BioProspector, un proceso llamado *threshold sampler* es realizado cierto número de veces. *Threshold sampler* adopta una estrategia de tipo *Gibbs sampling*, la cual inicializa la matriz de probabilidad del motivo Θ con un alineamiento aleatorio de las secuencias de entrada y mejora la matriz iterativamente y estocásticamente por un método de actualización predictivo (Liu, 1994). La fórmula de actualización predictiva utilizada, está basada en las siguientes modificaciones del modelo estadístico.

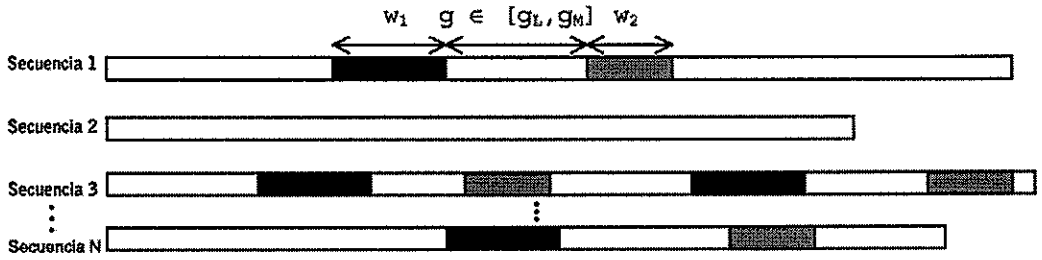


Figura 2. El modelo utilizado por BioProspector. Supone que tenemos N secuencias de ADN, cada una contiene de 0 a n copias de la secuencia del motivo. El motivo tiene dos bloques de pegado de largo w_1 y w_2, respectivamente, los cuales son separados por un espacio de largo variable de g_L a g_M.

Calificando segmentos con un fondo de dependencia tipo Markov

En *Gibbs sampler* (Lawrence *et al.*, 1993), cada posible segmento con un largo W dentro de una secuencia $s \in S$ seleccionada aleatoriamente es considerada. Un criterio de calificación $A_x = Q_x/P_x$ es calculado y una nueva posición de alineamiento a_s es probada con una probabilidad proporcional a A_x. Aquí Q_x y P_x son la probabilidad de generar un segmento x de la matriz actual Θ y del modelo de fondo independiente β, respectivamente. Pero en el ADN, la presencia de un nucleótido en particular usualmente tiene influencia en sus posiciones vecinas, así que una mejor forma de evaluar P_x es basándose en un modelo de fondo de Markov. Por ejemplo, la probabilidad de generar el segmento ATGTA de un modelo de fondo de Markov de tercer orden β se calcula como:

$P^3_{ATGTA} = p(A) \times p(T/\text{letra anterior es } A) \times p(G/\text{letras anteriores son } AT) \times p(T/\text{letras anteriores son } ATG) \times p(A/\text{letras anteriores son } ATGT)$

Encontrando motivos de dos bloques y palindrómicos

Para motivos de dos bloques, BioProspector utiliza dos matrices de probabilidad Θ_1 y Θ_2 para capturar los dos bloques. Las matrices son inicializadas al escoger aleatoriamente las posiciones de alineamiento (a_{s1}, a_{s2}) sobre la misma hebra de cada secuencia con un espaciado fijo $g_0 = (g_L + g_M)/2$. Dos segmentos x_1 de largo w_1 y x_2 con un largo w_2 con un intervalo de espacio son calificados como $A_{x1,x2} = (Q_{x1}/P_{x1})$ * (Q_{x2}/P_{x2}), en el cual Q_{x1} es la probabilidad de generar x_1 con Θ_1 y Q_{x2} es la probabilidad de generar x_2 con Θ_2. Se muestra x_1 desde su distribución marginal, la cual es proporcional a $A_{x1,*} = \Sigma_{x2} A_{x1,x2'}$, donde la sumatoria es sobre todos los segmentos de largo w_2 dentro de $[g_L, g_M]$ corriente abajo de x_1. Entonces el segmento x_2 es escogido con probabilidad

$A_{x1,x2}/A_{x1,*}$ condicionada en x_1. Cuando los dos bloques del motivo son palindrómicos, solo se necesita una matriz de probabilidad Θ. Cada secuencia alineada contribuye a dos segmentos en la misma matriz, una de cada hebra del ADN.

Usando una calificación de distribución de motivos para medir la calidad de un motivo

La información de $Kullback - Leibler$, también conocida como entropía relativa, ha sido utilizada para medir el contenido de información de un motivo (Schneider *et al.*, 1986). Pero, cuando un *motivo*1 tiene 150 segmentos alineados y *motivo*2 solo tiene 3, el *motivo*2 fácilmente puede tener mejor entropía relativa, aunque el primero representa un motivo más conservado y significativo. Para resolver este dilema, se introduce el siguiente criterio:

$$\textbf{\textit{Calificación}}=\#seg \times exp\left\{\left[\Sigma_{todas\ las\ posiciones\ i}\Sigma_{todos\ los\ nucleotidos\ j} q_{i,j} \times log(q_{i,j}/p_j)\right]/w\right\}$$

en donde $\#seg$ es el número de segmentos alineados en el motivo, $q_{i,j}$ es la probabilidad de observar el nucleótido j en la posición i de la matriz del motivo Θ, y p_j es la probabilidad de observar el nucleótido j en la distribución de la secuencia de fondo β. Para ver que tan significativo es el puntaje de un motivo, se utilizan simulaciones Monte Carlo para estimar una distribución nula de este puntaje. Para ser más precisos, el programa genera M conjuntos de secuencias *iid* bajo el modelo de probabilidad de las secuencias de entrada, donde cada conjunto generado es idéntico al archivo de entrada F en número de secuencias y longitud. Para cada conjunto de secuencias generadas, un número de corridas de *threshold sampler* es realizado, y el puntaje más alto de un motivo es almacenado. Una distribución normal es ajustada a los puntajes almacenados

M. Con esta distribución de puntajes, BioProspector corre la secuencia original con el *threshold sampler*, y reporta los z motivos (el valor predeterminado es 5). BioProspector tiene una complejidad computacional de $O(NK)$ (Porteus *et al.*, 2008), donde N es el número de secuencias y K el número de iteraciones.

IV.4 Weeder

El algoritmo Weeder (Pavesi *et al.*, 2001), es un algoritmo combinatorio basado en árboles de sufijos.

IV.4.1 Árboles de sufijos

Un árbol de sufijos es una estructura de datos que expone la estructura interna de una cadena. Un árbol de sufijos T para una cadena de n caracteres $S = s_1...s_n$ es un árbol dirigido enraizado con exactamente n hojas numeradas de 1 a n. Cada nodo interno, aparte de la raíz tiene por lo menos dos hijos. Cada arista está etiquetada con una subcadena de S. Dos aristas que salen del mismo nodo no pueden tener etiquetas empezando con el mismo carácter. Para cualquier hoja i, la concatenación de las etiquetas de las aristas en el camino de la raíz a la hoja i deletrea exactamente el sufijo de S comenzando por la posición i, esto es, deletrea $s_i...s_n$.

Dado un conjunto de secuencias y el árbol de sufijos anotado, cada palabra aparece en al menos una secuencia del conjunto que es deletreado por un camino único comenzando desde la raíz del árbol. Por lo que buscar un patrón M en el conjunto de secuencias es trivial. Comenzando de la raíz, comparamos los símbolos de M junto con el camino único en el árbol hasta que M haya terminado o ya no se encuentren coincidencias.

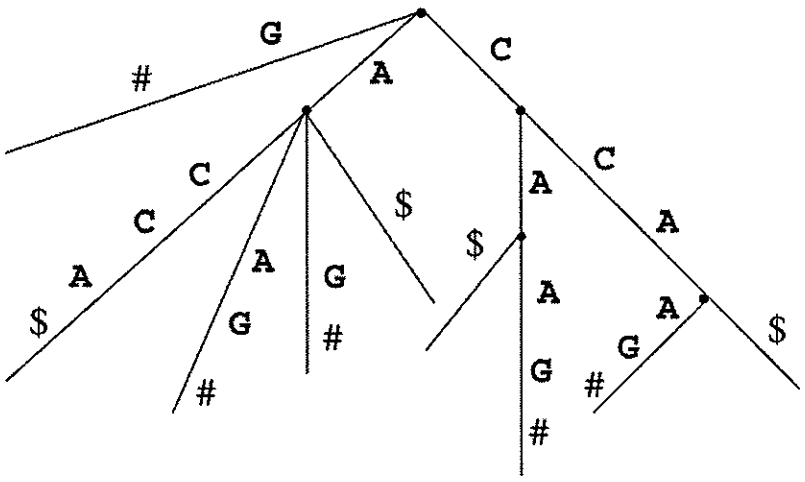


Figura 3. Árbol de sufijos para las secuencias ACCA y CCAAG. Los símbolos $ y # son usados como marcadores de fin de las cadenas ACCA y CCAAG, respectivamente.

También se puede buscar un patrón M con a lo más e mutaciones en forma similar. En este caso, buscamos las coincidencias en M junto con diferentes caminos en el árbol al mismo tiempo, almacenando el número de mutaciones encontradas en cada camino. Cuando el número de mutaciones en un camino es mayor a e, desechamos ese camino. Si completamos M, los caminos sobrevivientes representan todas las ocurrencias de M en las secuencias con a lo más e mutaciones.

IV.4.2 El algoritmo Weeder

El punto inicial del algoritmo es la búsqueda de ocurrencias aproximadas de un patrón en un árbol de sufijos. Dado un conjunto de k secuencias sobre el alfabeto $\sum$=$\{A, C, G, T\}$, queremos encontrar todos los patrones (M, e), esto es, patrones de largo $|M|$ que ocurren con a lo más e mutaciones en al menos q secuencias del conjunto. Supongamos que se han encontrado en el árbol los caminos correspondientes a las ocurrencias de un patrón $M = M_1...M_m$ en las secuencias, esto es, todos los caminos que deletrean

palabras con una distancia e de M, con $m < |M|$. También se ha asociado en cada camino la distancia (número de mutaciones) de M con la subcadena correspondiente. Si M es válido, esto es, que ocurre en a lo menos q secuencias, se intenta expandir en un símbolo. Por cada carácter $b \in \{A, C, G, T\}$, comparamos b contra el siguiente símbolo en cada camino. Si un camino termina justo antes de un nodo T del árbol, comparamos b con el primer símbolo en cada arista que sale de T. Cuando no hay coincidencia, incrementamos el error previo junto al camino en uno. De lo contrario el error permanece sin cambios. Si el nuevo error es mayor que e, se desecha el camino. Una vez que todos los caminos han sido revisados, los sobrevivientes representan las ocurrencias aproximadas de $M' = M_1...M_m b$. Si M' ocurre en al menos q secuencias y es más corto que M, se expande, de lo contrario se continua con p y con el siguiente carácter en Σ. El algoritmo comienza con un patrón vacío de la raíz del árbol, y recursivamente se expande. Esto es, compara el primer símbolo en cada arista que sale de la raíz con "A". Si "A" ocurre en al menos q secuencias, el patrón se expande a "AA". Si "AA" es válido, nos movemos a "AAA" y así sucesivamente. Si no es válido, se procede con "C", buscando ocurrencias de "AC".

Para cada patrón válido, tenemos que seguir al menos N caminos diferentes en el árbol, donde N es el largo total de las k secuencias. Para cada palabra de largo $|M|$ deletreada por un camino en el árbol (como el árbol tiene N hojas, son a lo más N), hay a lo mas $\sum_{i=1}^{e} \binom{m}{i} (|\Sigma| - 1)^i \leq |\Sigma|^e |M|^e$ patrones diferentes a una distancia e (Sagot, 1998).

Para aplicar el algoritmo en patrones largos se podría reducir el número de patrones que son buscados, por ejemplo aquellos que ocurren exactamente en las secuencias, al revisar que al menos un camino tiene error cero. Pero en lugar de reducir el conjunto de patrones que son buscados, se restringe el número de caminos que son revisados

para cada patrón. Esto es, reducimos el número de ocurrencias válidas. Por ejemplo, queremos encontrar patrones de largo 16 que ocurren con a lo más 4 errores. La búsqueda de cada patrón comienza con 4^4 caminos. Entre esos caminos, 3^4 van a deletrear palabras con distancia 4 del patrón, por lo que es poco probable que contengan una ocurrencia válida. La idea es cortar todos esos caminos. También se cortan los caminos con error dos y tres, considerando caminos con a lo más, una mutación.

En Weeder, el umbral del error es determinado dinámicamente según el largo del patrón. Se inicializa el error ϵ. Dado un patrón M, un camino es válido si la distancia de M de la palabra deletreada por el camino no es mayor que $\lceil \epsilon |M| \rceil$, donde $|M|$ es el largo del patrón. Si en el ejemplo, inicializamos ϵ=0.25, eliminamos todos los caminos de largo 4 con un error mayor a uno. Cuando se expande M en un símbolo, el umbral del error está dado por $\lceil \epsilon(|M|+1) \rceil$. El resultado es que, para cada patrón $M = M_1...M_m$, las ocurrencias válidas son palabras $s_{i+1}...s_{i+m}$ ocurriendo en las secuencias para las cuales: $\forall j \in \{1,...,m\}\, d(M_1..M_{1+j}, s_{i+1}..s_{i+j}) \leq \lceil \epsilon j \rceil$ donde $d(M_1..M_{1+j}, s_{i+1}..s_{i+j})$ es el número de mutaciones entre $M_1...M_{1+j}$ y $s_{i+1}...s_{i+j}$

El núcleo del algoritmo Weeder es el procedimiento **expand**:

1.Procedimiento **expand**(*pattern* M, Loc_p, *char* a)

2. $M' = Ma$;

3. $errmax = \lceil \epsilon |M'| \rceil$;

4. $OccBits = [0, 0, ..., 0]$;

5. $Loc_{M'} = \emptyset$

6. for all$(Pos, e) \in loc_M$

7. for all $Pos' \in Next(Pos)$

8. if $Last_{Pos'} = a$

9. $Loc_{M'} = Loc_{M'} \cup (Pos', e)$;

10. $OccBits = OccBits$ OR $Occ(Pos')$;
11. else if $e + 1 \leq errmax$
12. $Loc'_M = Loc'_M \cup (Pos', e + 1)$;
13. $OccBits = OccBits$ OR $Occ(Pos')$;
14. end if
15. end if
16. end for
17. end for
18. if (al menos q bits están en $OccBits$)
19. for all $a \in \{A, C, G, T\}$
20. expand(M', Loc'_M, a);
21. end for
22. end if
23. Regresar;

En el procedimiento **expand**:

- La letra a es añadida al final del patrón M.
- Loc_M es un conjunto de punteros (Pos, e) a los puntos finales de las ocurrencias del patrón M en el árbol, con su error correspondiente e.
- $OccBits$ es una cadena de $k - bits$ representando las ocurrencias de M en las k cadenas.
- $Next(Pos)$ regresa un conjunto de punteros a las posiciones en el árbol alcanzadas al mover una letra abajo del camino apuntado por Pos.

- $Occ(pos)$ regresa el bit de la cadena del primer nodo siguiendo el camino apuntado por Pos.

- $Last_{pos'}$ es la última letra del camino terminando en la posición apuntada por Pos'.

Weeder tiene una complejidad computacional de $O(|\Sigma|^e|M|^e kN)$ (Pavesi *et al.*, 2001), donde $\Sigma = \{A, C, G, T\}$, N es la longitud total de las k secuencias del caso de prueba, M es la longitud de los motivos y e el número de mutaciones permitidas.

IV.5 RepeatMasker

RepeatMasker (Smit *et al.*, 2010) es un programa que busca en las secuencias de ADN repeticiones entremezcladas y secuencias de baja complejidad. La salida del programa es una anotación detallada de las repeticiones presentes en la secuencia de entrada, así como una versión modificada de la secuencia en la cual todas las repeticiones anotadas han sido enmascaradas (los nucleótidos son remplazados por N). En promedio, casi el 50% de todo el genoma humano sería enmascarado por el programa.

RepeatMasker encuentra y enmascara elementos repetitivos al alinear cada una de las secuencias de entrada con cada una de las secuencias consensus de repeticiones que se encuentran en la base de datos RepBase (Jurka *et al.*, 2005). El alineamiento es realizado por una implementación eficiente del algoritmo de Smith-Waterman-Gotoh (Smith y Waterman (1981); Gotoh (1982)), mientras que RepeatMasker se encarga de todo el procesamiento y enmascaramiento de los alineamientos. El alineamiento utilizado por RepeatMasker usando el algoritmo Smith-Waterman-Gotoh tiene una complejidad de $O(MN)$, el acceso a la base de datos RepBase tiene una complejidad de $O(n)$ para n secuencias.

IV.5.1 Repeticiones en ADN

En un genoma los nucleótidos no se encuentran en igual cantidad; A y T se encuentran en un 60% (30% cada uno) y C y G en un 40% (20% cada uno). Pero un genoma no solo varía en la cantidad de nucleótidos que contiene, también presenta una distribución no uniforme de los nucleótidos a lo largo de la secuencia genómica; incluso existen algunos segmentos (con una longitud de 300 kb aproximadamente) con una alta concentración de un par de nucleótidos (A,T) o (C,G). Las repeticiones de secuencias en un genoma, principalmente en los organismos eucariotas contienen un alto número de secuencias repetidas de longitud, composición y frecuencia variables. Estas repeticiones se pueden encontrar en los genomas en dos formas: en tandem, dos o más copias consecutivas de un patrón (Benson, 1999) o dispersas aleatoriamente a lo largo del genoma.

Repeticiones en Tandem

Una repetición en tandem está compuesta de repeticiones contiguas de una subsecuencia. Se pueden clasificar de la siguiente manera:

- Satélites con una longitud total superior a 100 nucleótidos.
- Minisatélites con una longitud total de 9-100 nucleótidos.
- Microsatélites con una longitud total de 1-8 nucleótidos.

Repeticiones entremezcladas

Las repeticiones entremezcladas se clasifican de acuerdo con su longitud en cortas denominadas *SINE (short interspersed repeat)* y en largas denominadas *LINE (long interspersed)* (Wen-Hsiung, 1997), estas repeticiones pueden estar formadas por la combinación de nucleótidos en diferente orden y cantidad:

SINE:, son fragmentos de ADN cortos repetidos millones de veces y dispersos por todo el genoma. En el genoma humano los *SINE* tienen una longitud de 100 a 300 pb y se repiten aproximadamente 1.5 millones de veces (13% del genoma humano), las más conocidas son las secuencias *ALU*.

LINE:, son fragmentos de ADN de gran tamaño repetidos miles de veces y dispersos por todo el genoma. En el genoma humano los *LINES* tienen una longitud de 6000 a 8000 pb y se repiten 850 mil veces (21% del genoma humano).

IV.5.2 Base de datos de repeticiones

Las repeticiones entremezcladas buscadas por RepeatMasker están basadas en la base de datos de repeticiones Repbase (Jurka *et al.*, 2005) registrada por el Instituto de Investigación de Información Genética (GIRI) en Mountain View, CA. Dicha base de datos contiene anotaciones de repeticiones de diversas especies de eucariotas.

Repbase es una base de datos que contiene transposones. Es implementada en mySQL. La secuenciación a gran escala de genomas eucariotas ha resultado en un incremento de transposones descubiertos. La filosofía detrás de Repbase ha sido el de incorporar una cantidad significativa de depuración manual en la base de datos. Repbase es primariamente usada para la anotación y detección de ADN genómico. RepeatMasker utiliza una versión personalizada de las bibliotecas de Repbase en formato FASTA.

Las secuencias de Repbase son secuencias consensus de grandes familias y subfamilias de repeticiones. Pequeñas familias son representadas con secuencias de ejemplo. Repbase describe muchas familias de repeticiones que no están reportadas en otras bases de datos.

Capítulo V

Propuesta de Encadenamiento de algoritmos

V.1 Reduciendo el espacio de búsqueda

Un motivo es un patrón en común que tienen los segmentos de ADN donde se pegan los Factores de Transcripción, dichos segmentos se llaman sitios de pegado. A veces buscar un patrón en común en segmentos de ADN puede generar gran cantidad de falsos positivos (patrones en común donde no se pegan los Factores de Transcripción). Una solución es encontrar alguna característica en común para poder diferenciar ciertos patrones irrelevantes de los sitios de pegado reales. Xiaohui *et al.* (2010) muestran que en eucariotas, tales como el humano y el ratón, es normal encontrar múltiples sitios de pegado de diferentes motivos traslapados. Por lo que definen un módulo de motivos como un grupo de varios motivos, cuyos sitios de pegado pueden ocurrir en secuencias cortas de ADN traslapadas o con una separación pequeña entre ellos (1000 bases). Encontrar una forma de buscar diferentes motivos traslapados podría ser una forma de encontrar sitios de pegado.

Jianjun *et al.* (2005) proponen usar algoritmos estocásticos, tales como Gibbs Sampler (Lawrence *et al.*, 1993), AlignACE (Roth *et al.*, 1998) y BioProspector (Liu *et al.*, 2001), ya que al correr dichos algoritmos un número definido de veces, se obtienen diferentes predicciones, debido a las condiciones de ejecución, tales como el largo del motivo y la semilla inicial. Estas predicciones, a pesar de ser diferentes, tienen la tendencia

de agruparse, y si se obtiene un consenso de dicho grupo, es posible incrementar la exactitud de las predicciones.

Lo que Jianjun *et al.* (2005) y Xiaohui *et al.* (2010) proponen tiene sentido. Si aplicamos un algoritmo estocástico, podremos detectar regiones con una alta concentración de motivos, lo cual ayudará a detectar módulos de motivos y de esta manera tener una exactitud mayor en la búsqueda de sitios de pegado. Después se procederá a buscar los sitios de pegado dentro de dichos módulos.

El primer paso es elegir el algoritmo estocástico a utilizar, para esto ejecutamos los algoritmos propuestos por Jianjun *et al.* (2005) utilizando los casos de prueba real de Tompa *et al.* (2005). Se corrió cada algoritmo con los mismos casos de prueba, 5 veces cada uno para una longitud diferente (10,12,14,16,18,20) del motivo a encontrar, y entonces, se analizó qué algoritmo tenía más traslape con los sitios de pegado reales conocidos, y cuál generó grupos más definidos de motivos. Un grupo definido de motivos es un grupo que contiene por lo menos cinco o más sitios de pegado traslapados. AlignAce falló en generar un grupos definidos de motivos, Gibbs Sampler y BioProspector generaron grupos muy definidos de motivos, pero fue BioProspector quien tuvo más traslape con los sitios de pegado reales. Debido a lo anterior, se decide utilizar BioProspector para nuestra propuesta.

Por lo anterior, se ha generado un procedimiento que reduce el espacio de búsqueda dentro de las regiones promotoras (búsqueda de sitios de pegado dentro de módulos de motivos).

V.1.1 El Encadenamiento RM+BP+MM

El objetivo del primer encadenamiento, es simplemente detectar aquellas regiones en las secuencias, que se creen son módulos de motivos y eliminar el resto de la secuencia.

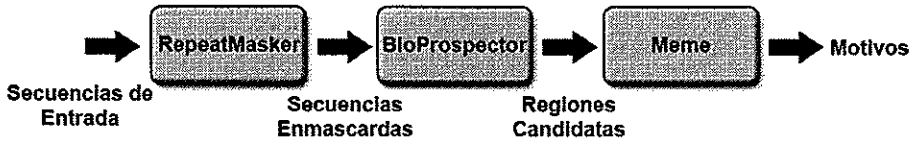


Figura 4. El encadenamiento RM+BP+MM.

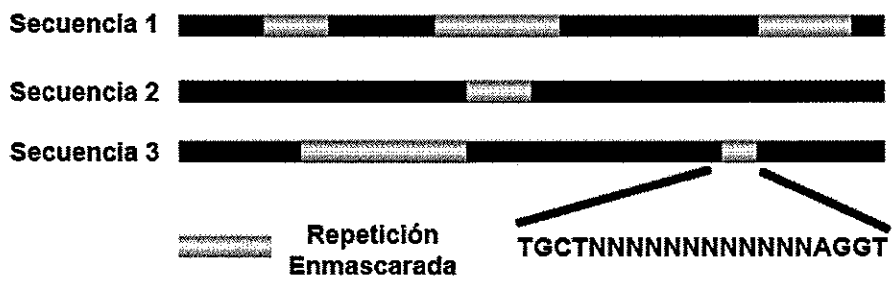


Figura 5. Tres regiones promotoras con secuencias enmascaradas por RepeatMasker.

Al hacer esto, se reduce el espacio de búsqueda, con el objetivo de incrementar la exactitud. Este encadenamiento, supone que los factores de transcripción se pegan en módulos, es decir, se encuentran dentro de una región corta de ADN, y su secuencia de reconocimiento es contigua o tiene traslape con la de algún otro factor de transcripción.

El primer encadenamiento de algoritmos propuesto RM+BP+MM (ver Figura 4) consiste en aplicar RepeatMasker (Smit *et al.*, 2010) a todos los casos de prueba, para hacer esto, se debe especificar el organismo del cual provienen las regiones promotoras. Como se explicó en el capítulo anterior (ver Sección IV.4), este software va a enmascarar aquellas regiones (ver Figura 5), las cuales de acuerdo a la literatura, se creen irrelevantes para el proceso de regulación.

El siguiente paso es ejecutar BioProspector (ver Sección IV.2) en los casos de prueba que ya se encuentran enmascarados. Este se ejecuta 10 veces para diferentes longitudes de motivos (10, 12, 14,17), reportando únicamente los primeros 10 motivos, obteniendo 400 predicciones de motivos en total. Esto se hace, con el objetivo de encontrar aquellas regiones que tengan una alta concentración de motivos (al menos 5 motivos traslapados),

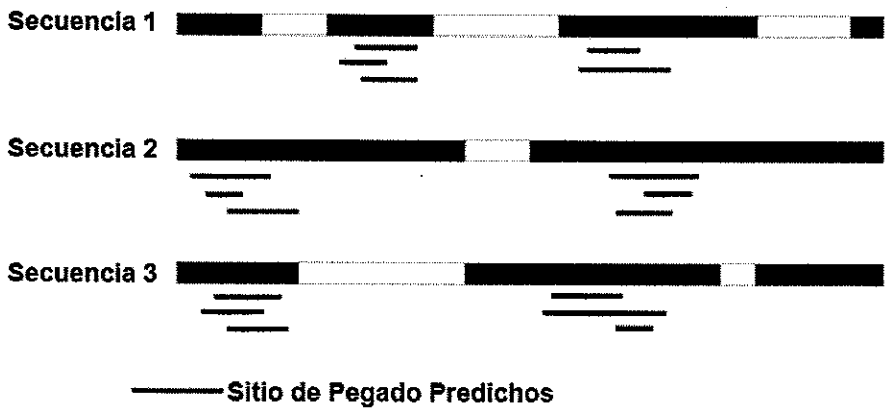


Figura 6. Los sitios de pegado predichos son alineados con las regiones promotoras.

para posteriormente, eliminar el resto de la secuencia. Al hacer esto, suponemos que los casos de prueba a analizar, contienen módulos de motivos, es decir, sitios de pegado traslapados o contiguos dentro de una misma región.

Después, se alinean las predicciones según su posición dentro de la región promotora (ver Figura 6), y se extraen aquellas regiones que traslapan con los sitios de pegado predichos (ver Figura 7), a las cuales denominaremos regiones candidatas. La razón por la que se extraen las regiones candidatas, en lugar de simplemente enmascarar las zonas sin traslape, es porque de esta forma podremos buscar sitios de pegado dentro de una sola región promotora. No todas las regiones promotoras van a generar el mismo número de regiones candidatas.

El último paso es ejecutar MEME (ver Sección IV.1) usando las regiones candidatas (ver Figura 8). Se decidió usar MEME porque el mismo no supone que cada secuencia de entrada debe tener un sitio de pegado, usando una distribución ANR (cualquier número de repeticiones por secuencia). Lo anterior debido a que no todas las regiones candidatas comparten un sitio de pegado en común. Por otro lado no se post-procesa la salida generada por MEME.

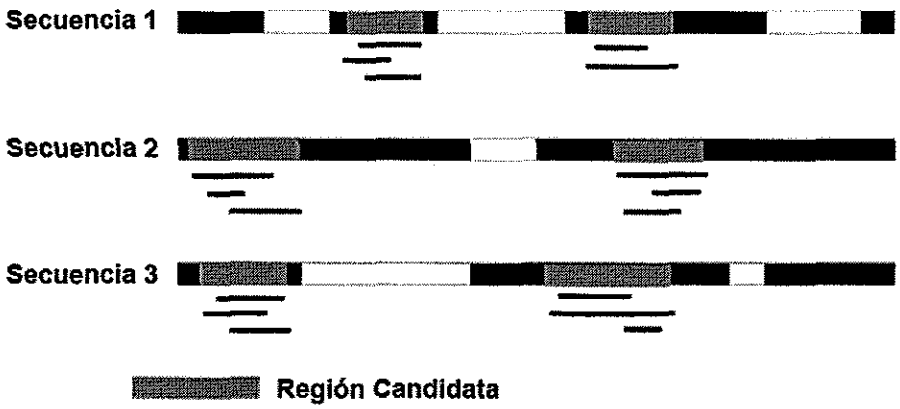


Figura 7. Usando el area de traslape con los sitios de pegado predichos, se generan regiones candidatas, que servirán como secuencias de entrada para el algoritmo MEME.

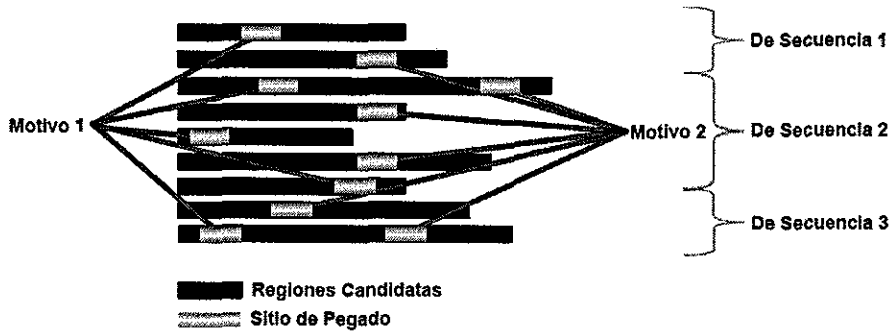


Figura 8. Sitios de pegado candidatos dentro de las regiones candidatas.

En resumen, los pasos del encadenamiento son:

1. Se obtienen los casos de prueba.

2. Se aplica RepeatMasker a cada conjunto de secuencias de cada caso de prueba.

3. Se ejecuta BioProspector 10 veces para un largo de motivo de 10, 12, 14 y 17 nucleótidos, y se toman los primeros 10 motivos para cada longitud.

4. Se toman todos los sitios de pegado reportados en el paso anterior, y se ordenan según su posición y secuencia de origen.

5. Se forman grupos ordenados para cada secuencia de sitios de pegado cuyas posi-

ciones de inicio y final tengan algún traslape entre sí. Para cada uno de estos grupos, se toma la posición inicial del primer sitio de pegado, y la posición final del último sitio. Con dichas posiciones, se forma una región única para cada grupo, la cual denominamos región candidata.

6. Con las regiones candidatas extraídas del paso anterior, se ejecuta MEME con una distribución ANR.

7. MEME reporta los sitios de pegado candidatos.

En cuanto a la complejidad computacional, se tiene que: acceder a la base de datos Repbase tiene un costo de $O(n)$, donde n es el número de repeticiones en la base de datos, debido a que se encuentra en formato FASTA y no contiene índice. El alineamiento utilizado por RepeatMasker usando el algoritmo Smith-Waterman tiene una complejidad de $O(MN)$ (Chan, 2004). La búsqueda de módulos con BioProspector tiene una complejidad de $O(NM)$ para cada iteración (Porteus *et al.*, 2008). La selección de motivos con MEME tiene una complejidad de $O((NM)^2W)$ (Bailey y Elkan, 2005). Por lo que la complejidad computacional de RM+BP+MM es de $O((NM)^2W)$ donde N es el número de secuencias en el caso de prueba, M es la longitud promedio de las secuencias, y W la longitud del motivo.

V.1.2 El encadenamiento RM+BP+WR+MM

Al igual que el encadenamiento anterior, el objetivo del segundo encadenamiento, es detectar aquellas regiones en las secuencias, que se creen son módulos de motivos y eliminar todos los bloques de secuencias que no se encuentran dentro de estos módulos para reducir así el tamaño del espacio de búsqueda.

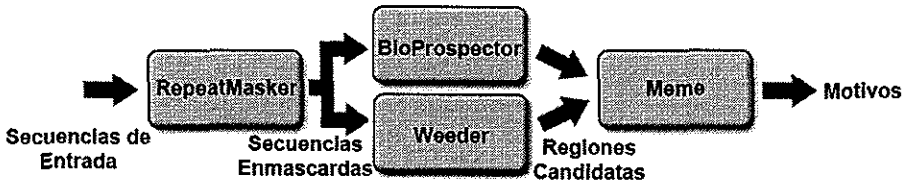


Figura 9. El encadenamiento RM+BP+WR+MM.

El encadenamiento RM+BP+WR+MM (ver Figura 9) es similar al encadenamiento anterior (RM+BP+MM), la única diferencia es la inclusión del algoritmo Weeder (Pavesi *et al.*, 2001) (ver Sección IV.3). Dicho algoritmo, según la evaluación realizada por Tompa *et al.* (2005), es uno de los que mejor sensibilidad obtiene para los casos de prueba. Según los experimentos que se realizaron (Capítulo 6, Sección VI.1), efectivamente se obtiene una sensibilidad mayor en comparación con MEME y BioProspector, pero con una baja especificidad, esto es debido a que se buscan patrones cortos (de largo 6 y 8), además de largo 10. Así que antes de la inclusión del algoritmo Weeder, simplemente limitaremos la búsqueda de motivos de largo 10. Con esto, se espera tener una especificidad mayor, pero reduciendo la sensibilidad.

Los pasos del encadenamiento RM+BP+WR+MM son los siguientes:

1. Se obtienen los casos de prueba.

2. Se aplica RepeatMasker a los casos de prueba.

3. Se ejecutan los siguientes algoritmos:

   - BioProspector 10 veces para un largo de 10, 12, 14 y 17, y se toman los primeros 10 motivos.

   - Weeder una vez para un largo de 10 y 12, y se toman los primeros 10 motivos.

4. Se toman todos los sitios de pegado reportados en el paso anterior, y se ordenan según su posición y secuencia de origen.

5. Se forman grupos ordenados para cada secuencia de sitios de pegado cuyas posiciones de inicio y final tengan algún traslape entre sí. Para cada uno de estos grupos, se toma la posición inicial del primer sitio de pegado, y la posición final del último sitio. Con dichas posiciones, se forma una región única para cada grupo, la cual denominamos región candidata.

6. Con las regiones candidatas extraídas del paso anterior, se ejecuta MEME con una distribución ANR.

7. MEME reporta los sitios de pegado candidatos.

En cuanto a la complejidad computacional de RM+BP+WR+MM es de $O(|\Sigma|^e W^e k M)$ donde $\Sigma = \{A, C, G, T\}$, M es la longitud promedio de las k secuencias del caso de prueba, W es la longitud de los motivos y e el número de mutaciones permitidas.

Capítulo VI

Experimentos y Resultados

En este capítulo se presentan los resultados tanto de los algoritmos existentes, como de los encadenamientos propuestos. Se utilizaran los casos de prueba propuestos por Tompa *et al.* (2005), los cuales se encuentran detallados en el Apéndice C. En las siguientes secciones, se mostraran resultados de cada caso de prueba del humano, al igual que resultados promedio sobre todos los casos de prueba (mosca, humano, ratón y levadura). En la Sección VI.4.2 se podrá ver a más detalle la exactitud de cada algoritmo para cada uno de los organismos.

VI.1 Comparando Algoritmos

Evaluamos los algoritmos MEME (Bailey y Elkan, 1995), BioProspector (Liu *et al.*, 2001) y Weeder (Pavesi *et al.*, 2001) utilizando los casos de prueba propuestos por Tompa *et al.* (2005). Se calcula para cada caso de prueba la sensibilidad, especificidad y rendimiento, los resultados para los casos de prueba del humano se muestran en las Figuras 10, 11 y 12, respectivamente. Como se muestra en la Figura 13, en promedio para todos los organismos, Weeder obtuvo la mejor sensibilidad (0.24), le siguió BioProspector (0.19) y por último MEME (0.15). En cuanto a especificidad, MEME obtuvo el puntaje más alto (0.038), le siguió BioProspector (0.033) y por último Weeder (0.022). Hay que notar que Weeder obtuvo mejor sensibilidad con muy poca especificidad, mientras que MEME obtuvo baja sensibilidad con una especificidad más alta, es decir, encuentra menos sitios de pegado, pero con menor cantidad de falsos positivos.

Por lo anterior, MEME tuvo el mejor rendimiento (0.032), seguido de BioProspector (0.030) y por último Weeder (0.020). Una de las razones por las que Weeder tuvo un rendimiento bajo, es debido a su búsqueda de sitios de pegado de longitud corta, 6 y 8 nucleótidos de largo, con esto se incrementa la cantidad de posibles patrones conservados, ya que entre más corto el patrón, se pueden encontrar más ocurrencias debidas al azar dentro de las regiones promotoras. BioProspector mostró en general una buena sensibilidad y especificidad, lo cual se ve reflejado en un rendimiento muy cercano al de MEME.

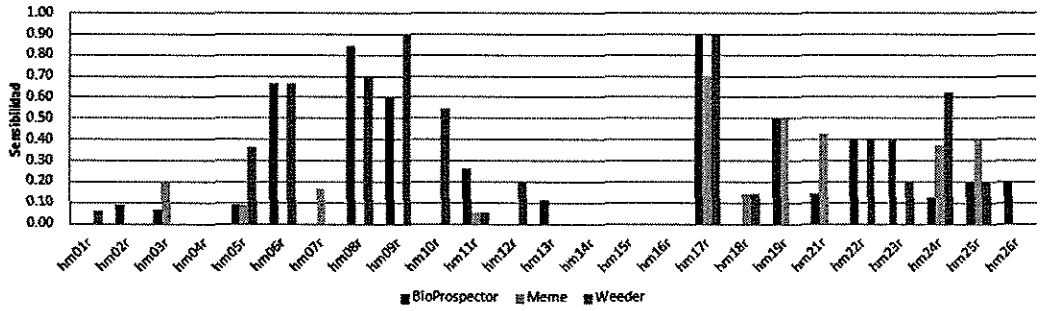


Figura 10. Sensibilidad de Weeder, MEME y BioProspector sobre 26 casos de prueba del humano.

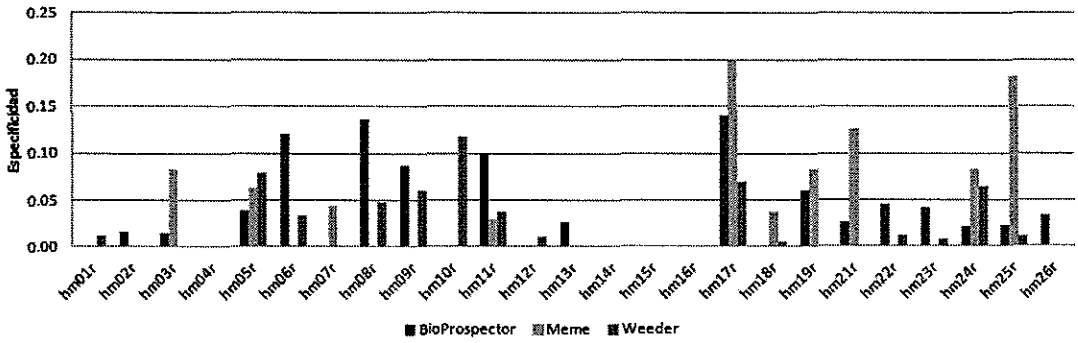


Figura 11. Especificidad de Weeder, MEME y BioProspector sobre 26 casos de prueba del humano.

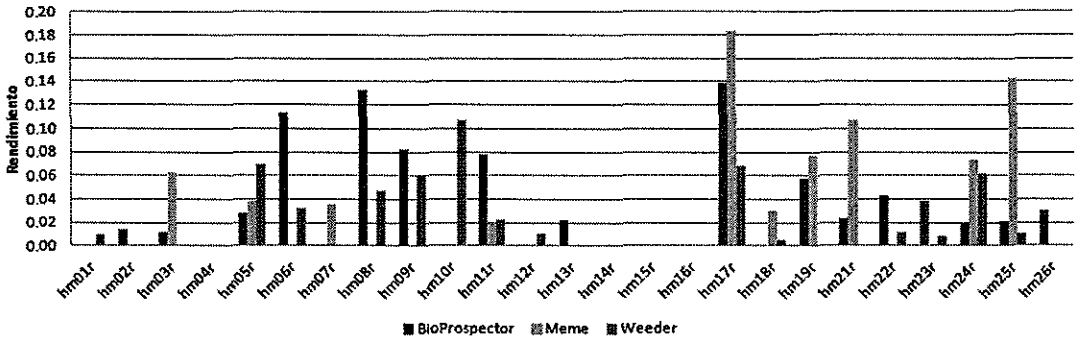


Figura 12. Rendimiento de Weeder, MEME y BioProspector sobre 26 casos de prueba del humano.

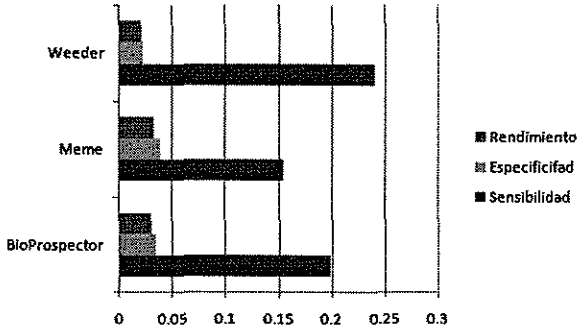


Figura 13. Sensibilidad, Especificidad y Rendimiento de Weeder, MEME y BioProspector, promedio sobre todos casos de prueba de Tompa *et al.* (2005).

VI.2 Preprocesando con RepeatMasker

Preprocesar los datos de entrada con RepeatMasker (Smit *et al.*, 2010) incrementa la exactitud en los algoritmos vistos en la sección anterior, a continuación se muestra a detalle dicha mejoría.

VI.2.1 BioProspector

Aplicar RepeatMasker a BioProspector mejora la exactitud de la predicción, en la Figura 14 se observa como la sensibilidad mejora en algunos de los casos de prueba, y en otros, mantiene la sensibilidad anterior, es decir, no hay pérdida de información.

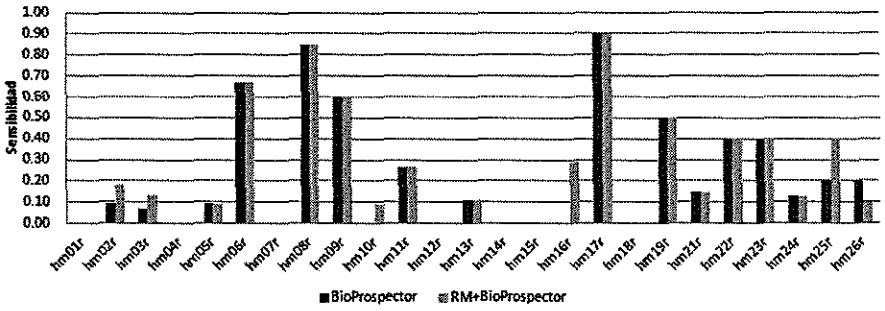


Figura 14. Sensibilidad de BioProspector y RepeatMasker + BioProspector sobre 26 casos de prueba del humano.

La especificidad varía con la información preprocesada y la completa. En la Figura 15 se muestran 4 casos de prueba (hm06, hm19, hm22 y hm26), en los cuales, con la información sin preprocesar se obtiene mejor especificidad, en el resto de los casos de los casos de prueba, se obtiene una especificidad similar o mayor.

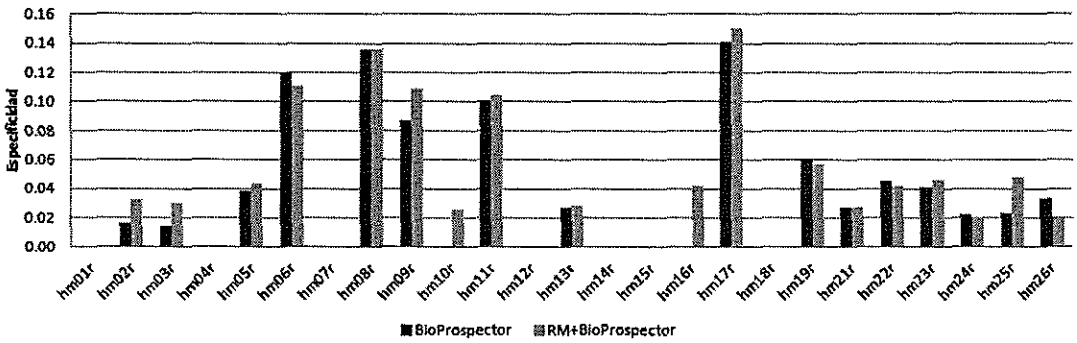


Figura 15. Especificidad de BioProspector y RepeatMasker + BioProspector sobre 26 casos de prueba del humano.

En lo referente al rendimiento, como se observa en la Figura 16 se obtiene una mejoría en la mayoría de los casos de prueba, exceptuando 4 casos (hm06, hm19, hm22 y hm26), en los cuales, el rendimiento baja. Esto se debe a que los motivos propuestos por RM+BioProspector contienen más sitios de pegado en comparación con BioProspector.

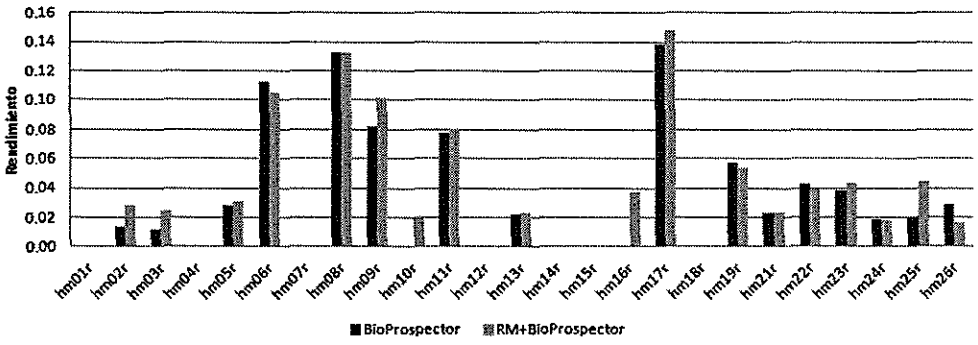


Figura 16. Rendimiento de BioProspector y RepeatMasker + BioProspector sobre 26 casos de prueba del humano

VI.2.2 MEME

Con MEME, RepeatMasker también favorece la exactitud, para todos los casos de prueba se mantuvo o se incrementó la sensibilidad (ver Figura 17). En cuanto a la especificidad, se incrementó en la mayoría de los casos de prueba, exceptuando el caso de prueba hm11, donde se decrementó (ver Figura 18).

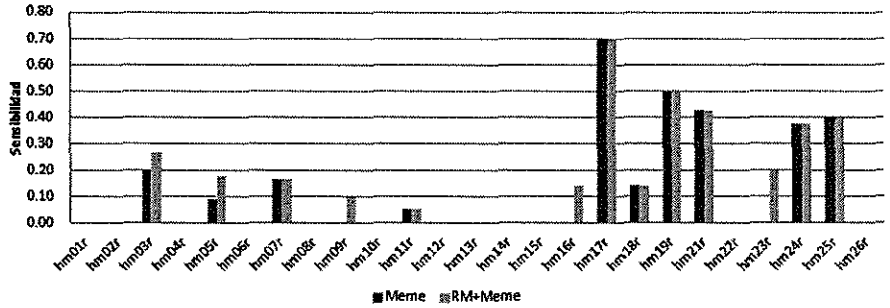


Figura 17. Sensibilidad de MEME y RepeatMasker + MEME sobre 26 casos de prueba del humano.

Para el rendimiento, se mantuvo o incrementó, exceptuando el caso hm11, donde hubo un ligero decremento (ver Figura 19).

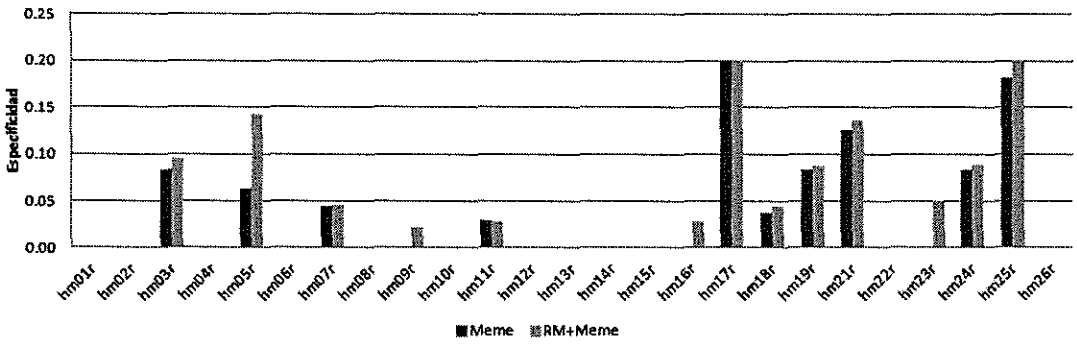


Figura 18. Especificidad de MEME y RepeatMasker + MEME sobre 26 casos de prueba del humano.

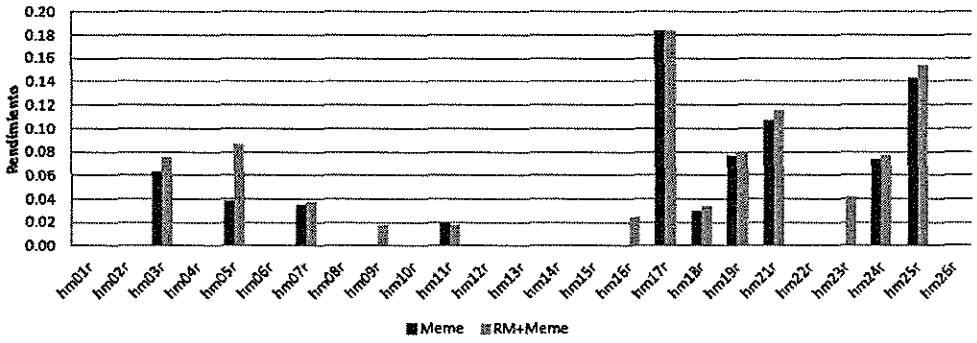


Figura 19. Rendimiento de MEME y RepeatMasker + MEME sobre 26 casos de prueba del humano.

VI.2.3 Weeder

Utilizando RepeatMasker con Weeder también mejoró la exactitud, en cuanto a la sensibilidad (ver Figura 20) se mantuvo en la mayoría de los casos, en 5 hubo una mejoría significativa (hm02, hm03, hm11, hm16 y hm21). En la especificidad (ver Figura 21) hubo incremento significativo en 10 de los casos (hm02, hm03, hm06, hm08, hm09, hm10, hm11, hm16, hm17 y hm21), en el resto, se mantuvo, no hubo decrementos significativos.

El rendimiento (ver Figura 22) se mantuvo en la mayoría de los casos, en 10 de los casos de prueba hubo mejoría significativa (hm02, hm03, hm06, hm08, hm09, hm10,

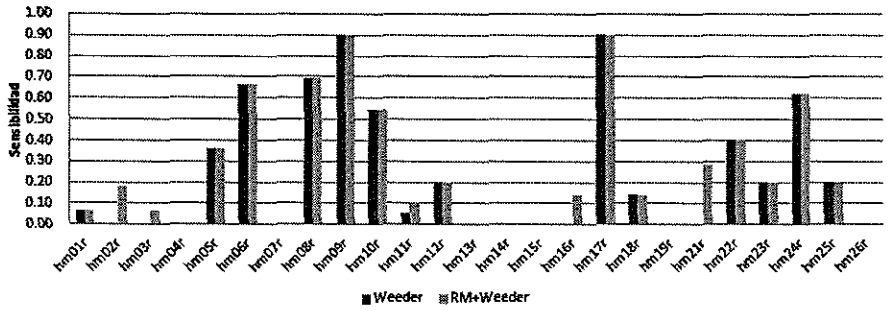


Figura 20. Sensibilidad de Weeder y RepeatMasker + Weeder sobre 26 casos de prueba del humano.

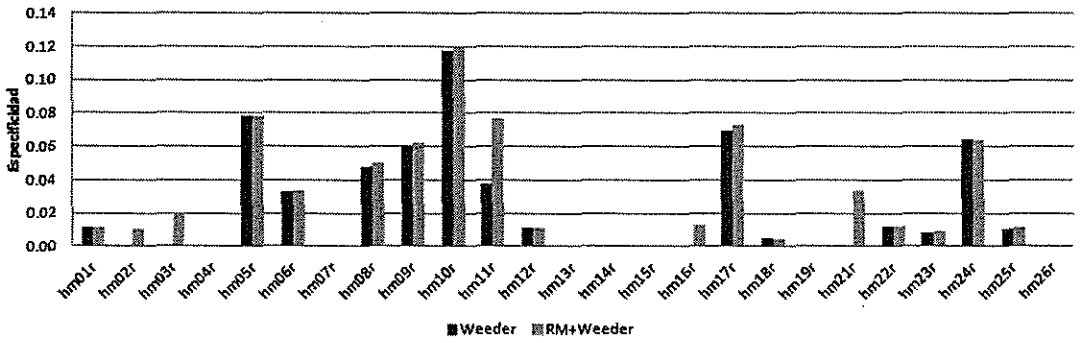


Figura 21. Especificidad de Weeder y RepeatMasker + Weeder sobre 26 casos de prueba del humano.

hm11, hm16, hm17 y hm21), no hubo decrementos significativos.

VI.2.4 Análisis

En promedio, hubo un incremento significativo al utilizar RepeatMasker. Esto es debido a que las secuencias que RepeatMasker elimina, no son relevantes para la regulación. Dichas secuencias son repetitivas dentro de un genoma, y encontrar un motivo dentro de dichas regiones es inservible. Es por eso que, al eliminarlas, los algoritmos pueden enfocarse en a aquellas regiones que se creen relevantes y de esta manera, reducir el número de falsos positivos.

Como se puede observar en la Figura 23, la sensibilidad se incrementó en todos los

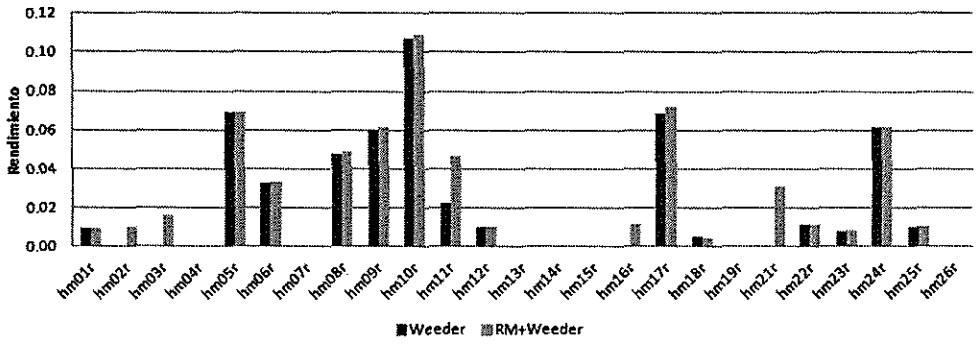


Figura 22. Rendimiento de Weeder y RepeatMasker + Weeder sobre 26 casos de prueba del humano.

algoritmos, siendo Weeder con RepeatMasker quien obtuvo el puntaje más alto. Pero en la especificidad (ver Figura 24), Weeder obtuvo la calificación más baja, incluso al utilizar RepeatMasker con Weeder falla en obtener un puntaje más alto que MEME y BioProspector en sus versiones sin preprocesado. Esto es debido a la gran cantidad de sitios de pegado candidatos que Weeder da como salida. Fue MEME quien obtuvo el puntaje más alto.

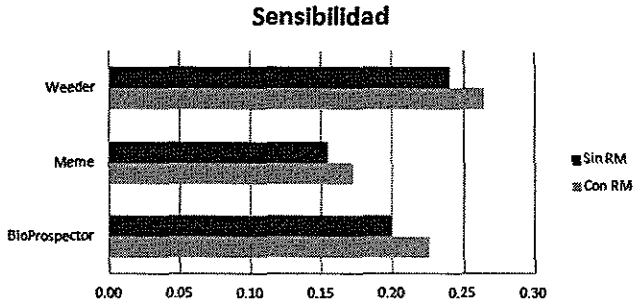


Figura 23. Sensibilidad promedio de las combinaciones anteriores.

MEME también obtiene el rendimiento más alto (ver Figura 25), seguido de Bio-Prospector. Weeder, a pesar de que tiene una sensibilidad mayor, obtiene un rendimiento muy bajo, debido a su baja especificidad.

En general, MEME obtiene pocos sitios de pegado, pero con una alta especificidad,

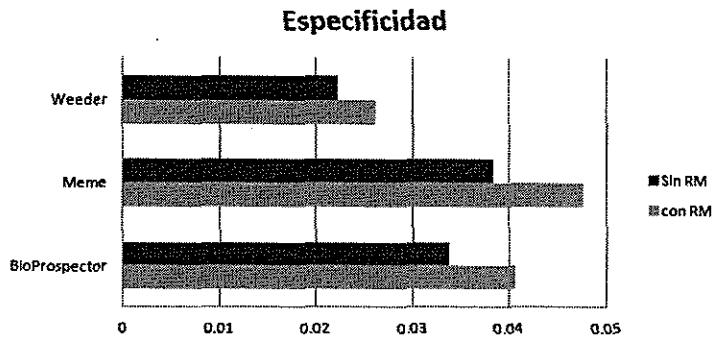


Figura 24. Especificidad promedio de las combinaciones anteriores.

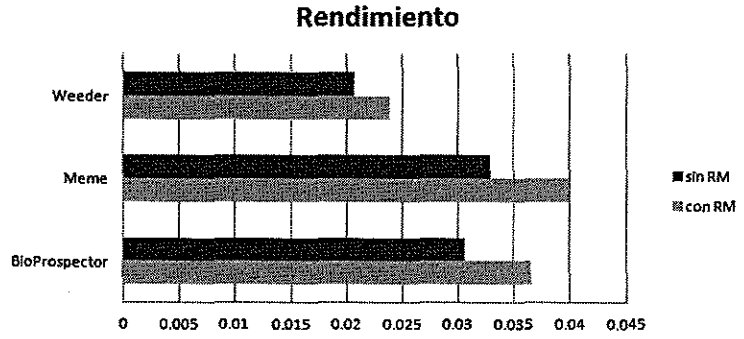


Figura 25. Rendimiento promedio de las combinaciones anteriores.

debido a que dicho algoritmo produce pocos sitios de pegado candidatos, justamente lo opuesto de Weeder, donde encuentra muchos sitios de pegado, pero con un alto índice de falsos positivos. BioProspector muestra una consistencia tanto en sensibilidad como en especificidad.

VI.3 Encadenamiento de Algoritmos

A continuación se muestran los resultados de los dos encadenamientos propuestos, ambos son comparados con las versiones preprocesadas con RepeatMasker de MEME, Weeder y BioProspector.

VI.3.1 RepeatMasker + BioProspector + MEME

Para la combinación RepeatMasker + BioProspector + MEME, se obtuvo una mejoría en la sensibilidad en el 57% de los casos de prueba (ver Figura 26), Weeder con Repeat-Masker en 21% de los casos, y tanto MEME y BioProspector con 11% cada uno. En cuanto a la especificidad (ver Figura 27) y rendimiento (ver Figura 28), la combinación obtiene el número mayor de casos de prueba 37% y 34%, respectivamente con puntaje más alto.

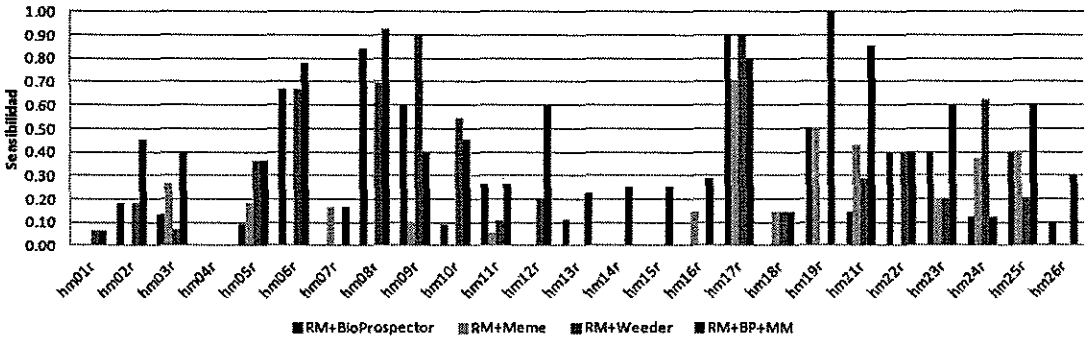


Figura 26. Sensibilidad de la combinación RepeatMasker + BioProspector + MEME sobre 26 casos de prueba del humano.

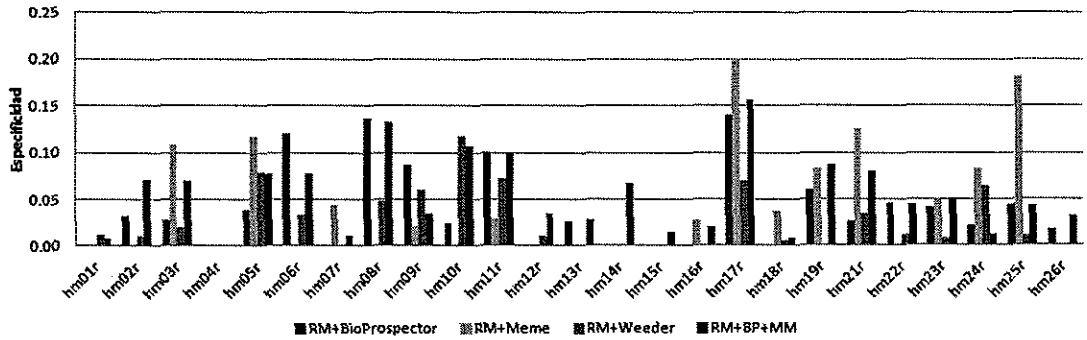


Figura 27. Especificidad de la combinación RepeatMasker + BioProspector + MEME sobre 26 casos de prueba del humano.

En promedio (ver Figura 29), se obtuvo la sensibilidad más alta (0.334) seguida por RM+Weeder con (0.252). También se obtuvo la especificidad más alta (0.0421),

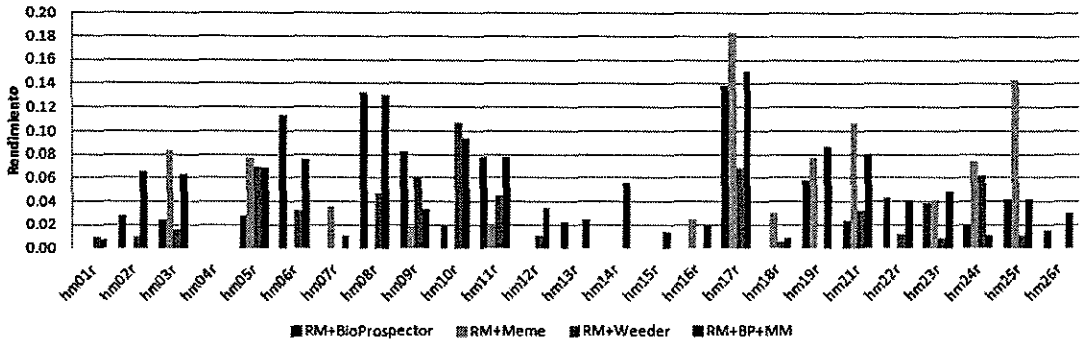


Figura 28. Rendimiento de la combinación RepeatMasker + BioProspector + MEME sobre 26 casos de prueba del humano.

seguido por RM+MEME (0.0419). En cuanto al rendimiento, se obtiene el puntaje más alto (0.03929), seguido de RM+MEME (0.0357), RM+Weeder obtiene el rendimiento más bajo (0.0222).

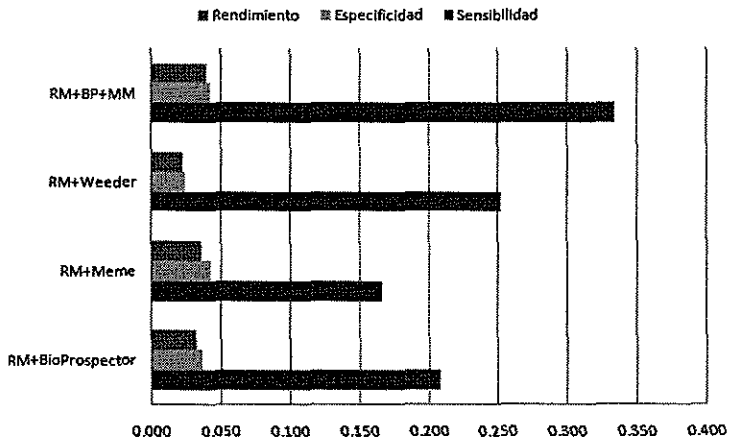


Figura 29. Promedios de Sensibilidad, Especificidad y Rendimiento de la combinación RepeatMasker + BioProspector + MEME.

VI.3.2 RepeatMasker + BioProspector + Weeder + MEME

Para la combinación RepeatMasker + BioProspector + Weeder + MEME, se obtuvo una mejor sensibilidad en el 46% de los casos de prueba (ver Figura 30). Para la

especificidad (ver Figura 31) y rendimiento (ver Figura 32) no obtiene los puntajes más altos, pero es capaz de obtener una especificidad consistente, quedando todos los casos de prueba en al menos, el segundo lugar.

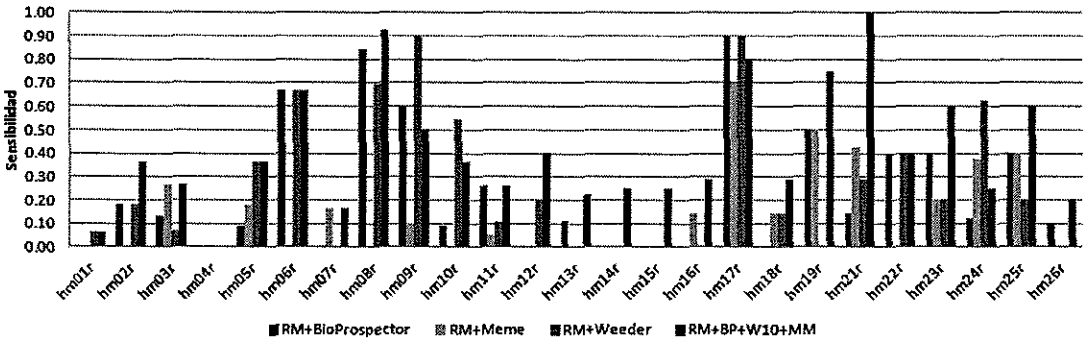


Figura 30. Sensibilidad de la combinación RepeatMasker + BioProspector + Weeder + MEME sobre 26 casos de prueba del humano.

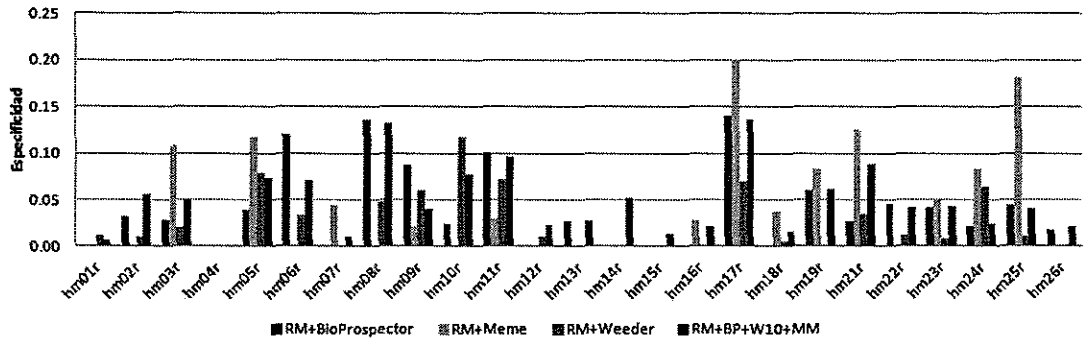


Figura 31. Especificidad de la combinación RepeatMasker + BioProspector + Weeder + MEME sobre 26 casos de prueba del humano.

En promedio (ver Figura 33), obtiene el puntaje más alto en sensibilidad (0.326), mientras que en cuanto a la especificidad, quedó en segundo lugar (0.0401) después de MEME (0.0419). Se obtiene el rendimiento más alto (0.0368), seguido de MEME (0.0357).

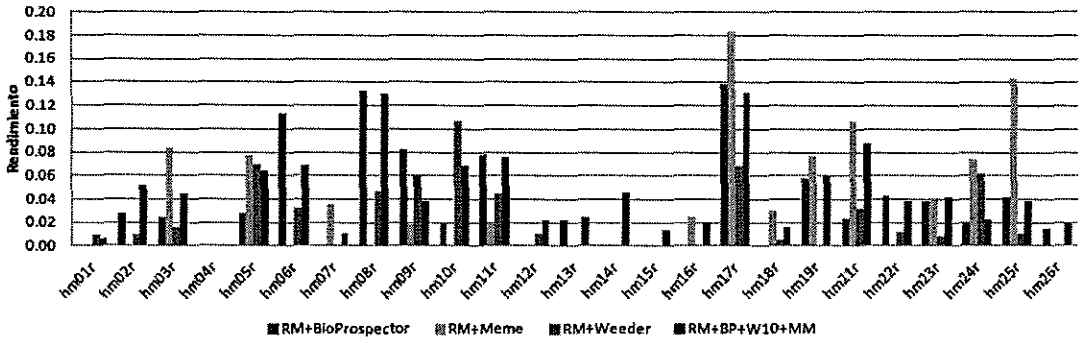


Figura 32. Rendimiento de la combinación RepeatMasker + BioProspector + Weeder + MEME sobre 26 casos de prueba del humano.

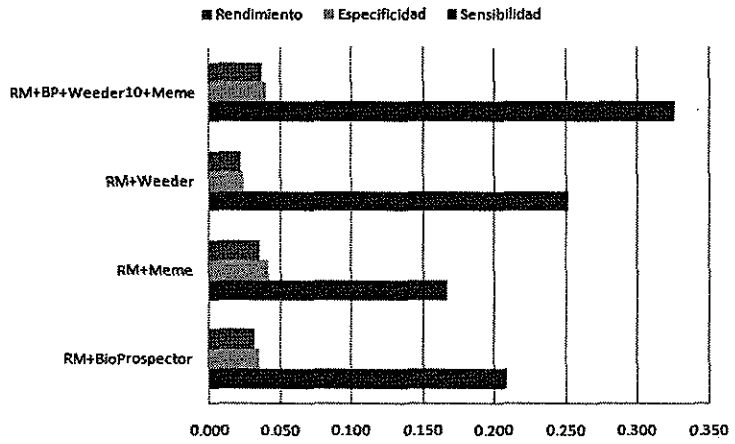


Figura 33. Totales de Sensibilidad, Especificidad y Rendimiento de la combinación RepeatMasker + BioProspector + Weeder + MEME.

VI.3.3 Comparación de las combinaciones

Comparando ambas combinaciones, se puede observar que la sensibilidad (ver Figura 34) por cada caso de prueba en ambas combinaciones es similar, siendo RM+BP+MM el que obtiene mayor puntaje en el 54% de los casos, mientras que RM+BP+WD+MM en el 46% restante. En la especificidad (ver Figura 35), se puede ver una gran diferencia, ya que RM+BP+MM obtiene un puntaje más alto en el 72% de los casos de prueba. Algo similar sucede con el rendimiento (ver Figura 36), RM+BP+MM obtiene mejor

puntaje en el 68% de los casos de prueba.

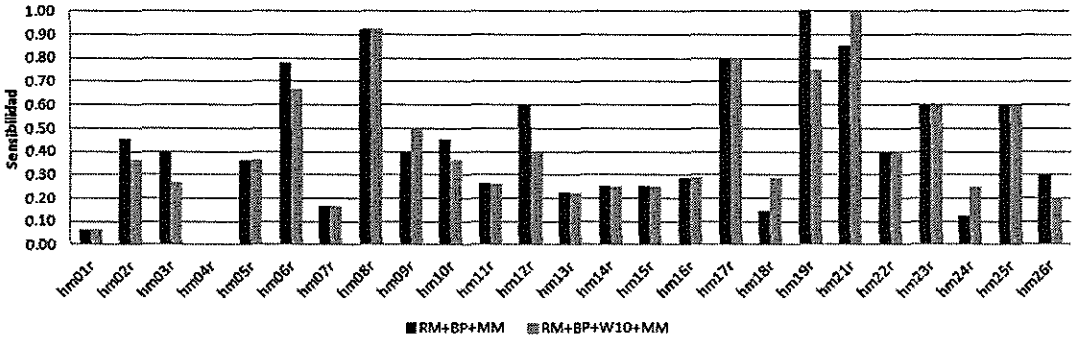


Figura 34. Sensibilidad de ambas combinaciones sobre 26 casos de prueba del humano.

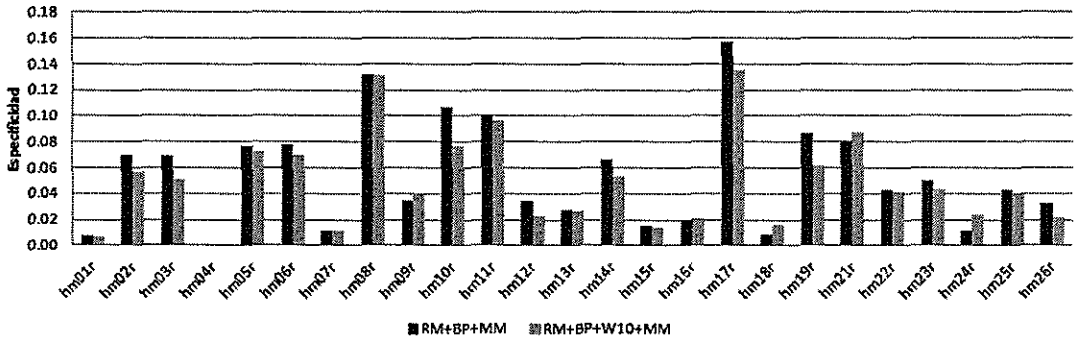


Figura 35. Especificidad de ambas combinaciones sobre 26 casos de prueba del humano.

En promedio (ver Figura 37), RM+BP+MM muestra una mejor sensibilidad (0.334), mientras que RM+BP+WD+MM le con (0.326). En cuanto a la especificidad y rendimiento, RM+BP+MM nuevamente obtiene un mejor puntaje (0.0421 y 0.0392, respectivamente). Se puede observar que los puntajes de ambos encadenamientos son cercanos, aunque al agregar el algoritmo de Weeder a la segunda combinación, se incrementa el número de posibles sitios de pegado candidatos, lo cual disminuye la especificidad y este a su vez, el rendimiento.

En conclusión, la combinación RM+BP+MM, es la que obtiene mejores resultados

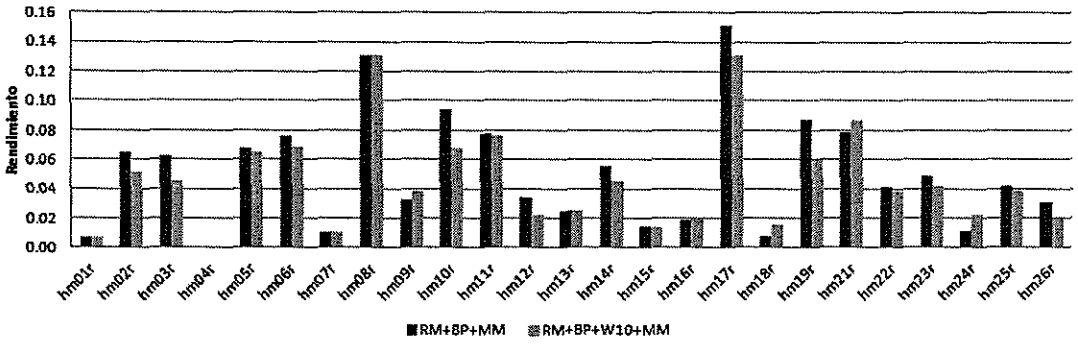


Figura 36. Rendimiento de ambas combinaciones sobre 26 casos de prueba del humano.

tanto en sensibilidad, especificidad y rendimiento entre todos los algoritmos utilizados.

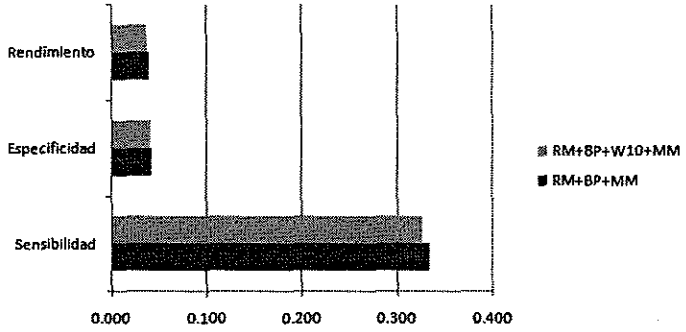


Figura 37. Totales de ambas combinaciones.

La Figura 38 muestra el porcentaje de mejoría en promedio con respecto al resto de los algoritmos.

VI.4 Análisis de Resultados

En la sección anterior, se observa que la combinación RM+BP+MM es la que obtiene mejores resultados, pero aun existen algunos casos de prueba en los cuales se obtienen puntajes bajos. Esto es debido a la naturaleza del mismo caso de prueba, así que a

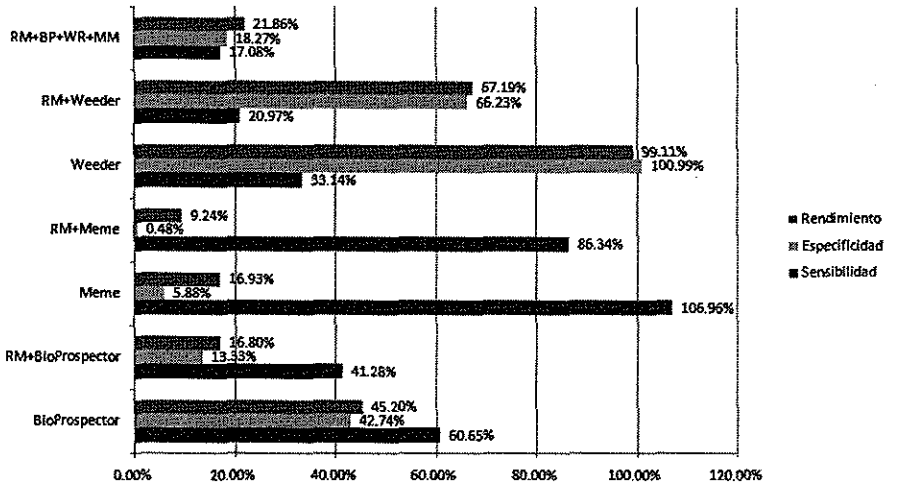


Figura 38. Porcentajes de mejoría de RM+BP+MM con respecto a los otros algoritmos, considerando todos los casos de prueba de Tompa *et al.* (2005).

continuación, analizaremos los resultados de dicha combinación junto con la estructura de los casos de prueba.

VI.4.1 Falsos Positivos

El número de falsos positivos es de gran importancia, y dicho número depende de la cantidad de motivos que el usuario le pida al algoritmo. Entre menos motivos se le pida al algoritmo, menos falsos positivos tendremos, pero al mismo tiempo tendremos menos sitios de pegado verdaderos.

En promedio, por cada sitio de pegado verdadero, BioProspector nos dará 21.23 falsos positivos, RM+BioProspector 21.41 falsos positivos, MEME 23.17 falsos positivos, RM+MEME 22.33 falsos positivos, Weeder 59.72 falsos positivos y por último, RM+Weeder nos dará 56.42 falsos positivos. En cuanto a las combinaciones RM+BP+MM y RM+BP+WR+MM, se generan 19.92 y 23.87 falsos positivos por cada sitio de pegado verdadero, respectivamente.

Una forma de eliminar falsos positivos es usar información sobre motivos ya conocidos en bases de datos, como Jaspar (Sandelin *et al.*, 2004) y Transfac (Wingender *et al.*, 1996). Sin embargo, esto tiene sus limitaciones por la incertidumbre misma de estas bases de datos, ya que no todo lo que en el algoritmo se evalúe como falso positivo realmente lo será.

VI.4.2 Análisis por Organismo

En las figuras 39, 40 y 41 se muestra la sensibilidad, especificidad y rendimiento promedio de cada organismo de los casos de prueba de Tompa *et al.* (2005).

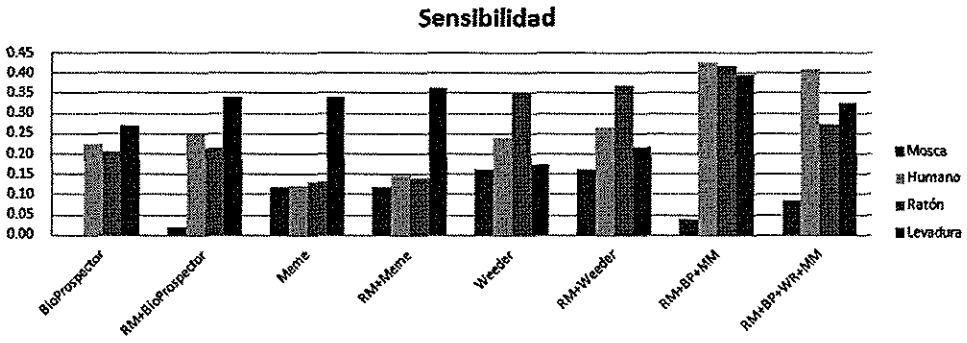


Figura 39. Sensibilidad promedio por organismo de Weeder, MEME y BioProspector.

Se puede observar, que para cada organismo y cada algoritmo, la sensibilidad, especificidad y rendimiento se comportan de diferente manera. Para la Mosca, el algoritmo RM+Weeder obtuvo la mejor sensibilidad (0.161), pero RM+MEME obtuvo mejor especificidad y rendimiento (0.042 y 0.035, respectivamente). En el caso del Humano, la combinación RM+BP+MM obtuvo la mejor sensibilidad, especificidad y rendimiento (0.427, 0.054 y 0.050, respectivamente).

RM+BP+MM también obtuvo la mejor sensibilidad en el caso del ratón (0.417), RM+MEME obtuvo la mejor especificidad y rendimiento (0.069 y 0.060). Por último,

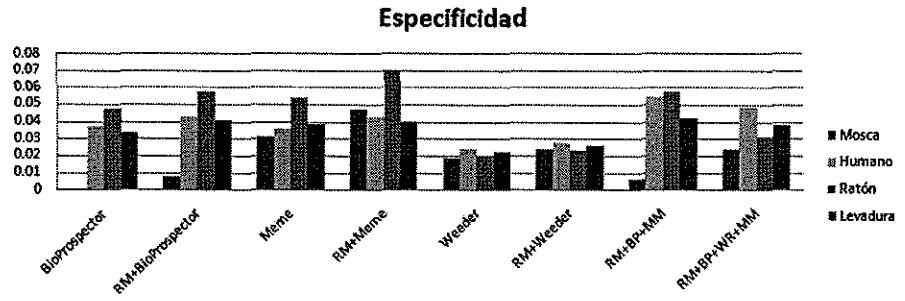


Figura 40. Especificidad por organismo de Weeder, MEME y BioProspector.

en caso de la Levadura, RM+BP+MM obtuvo la mejor sensibilidad, especificidad y rendimiento (0.395, 0.042 y 0.040, respectivamente).

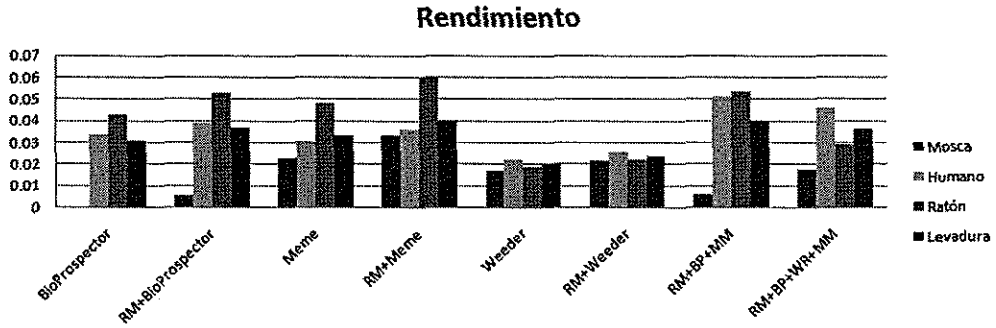


Figura 41. Rendimiento por organismo de Weeder, MEME y BioProspector.

VI.4.3 Análisis por número se secuencias

Uno de los factores que deciden si el algoritmo tiene buena exactitud o no, es el número de regiones promotoras por caso de prueba. La Figura 42 muestra como la sensibilidad con la combinación RM+BP+MM se incrementa a mayor número de regiones promotoras por caso de prueba.

La especificidad y el rendimiento (ver Figura 43) se ven afectados según el número de regiones promotoras por cada caso de prueba, a mayor número de regiones promotoras,

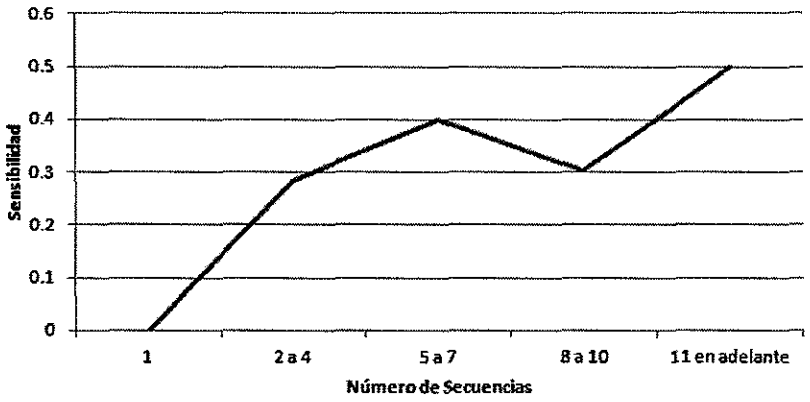


Figura 42. Sensibilidad de RM+BP+MM por número de secuencias en los casos de prueba.

mayor especificidad y rendimiento. Se puede observar un decremento en el intervalo de 8 a 10 secuencias tanto para sensibilidad, especificidad y rendimiento, esto es debido al bajo número de casos de prueba que representan dicho intervalo.

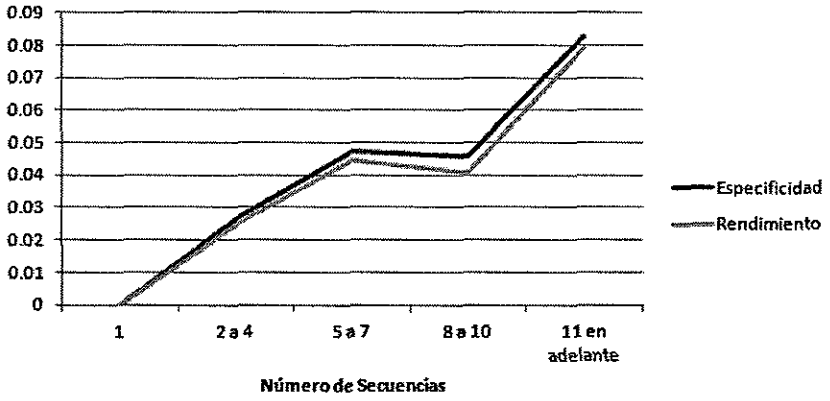


Figura 43. Especificidad y Rendimiento de RM+BP+MM por número de secuencias en los casos de prueba.

VI.4.4 Análisis por nivel de conservación de los sitios de pegado

Otro factor clave que decide la exactitud de los algoritmos, es el nivel de conservación de los sitios de pegado, es decir, que no sea muy diferentes el uno del otro. Para este análisis, el porcentaje de conservación fue calculado con el software Geneious (Drummond *et al.*, 2011). Se introducen los sitios de pegado reales al software Geneious, se alinean con el algoritmo ClustalW (Thompson *et al.*, 1994), y se genera el porcentaje de nucleótidos que son idénticos en el alineamiento. Dicho porcentaje es únicamente usado como referencia para ver el comportamiento de la exactitud del algoritmo RM+BP+MM, puede variar dependiendo del algoritmo de alineamiento que se utilice.

Los sitios de pegado obtuvieron un porcentaje de conservación que variaba del 35% al 84%. Se puede observar en la Figura 44 como la sensibilidad aumenta a mayor conservación se los sitios de pegado. En cuanto a la especificidad y rendimiento (Figura 45), se observa un incremento cuando el porcentaje de conservación es superior al 55%.

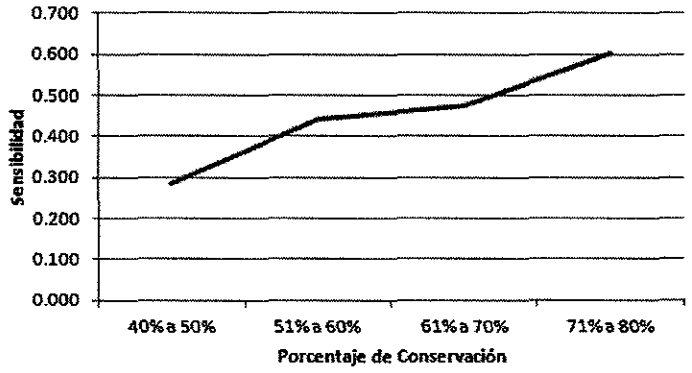


Figura 44. Sensibilidad de RM+BP+MM por conservación de los sitios de pegado.

Por lo anterior, se puede concluir como es de esperarse que a mayor número de regiones promotoras y a mayor porcentaje de conservación de los sitios de pegado,

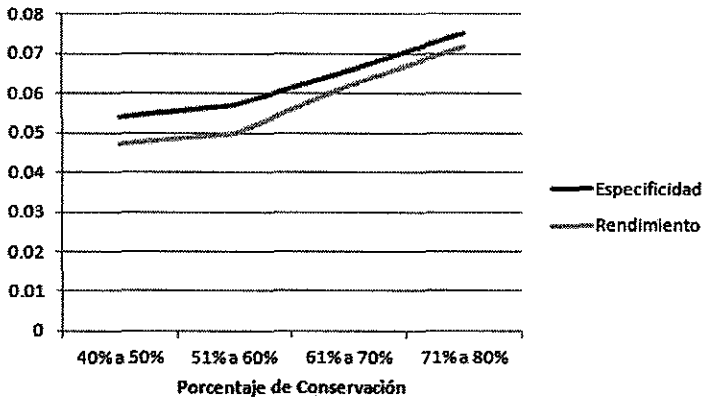


Figura 45. Especificidad y Rendimiento de RM+BP+MM por conservación de los sitios de pegado.

mayor exactitud en los algoritmos.

En el Apéndice D se muestran todos los datos de sensibilidad (Tabla XXII), especificidad (Tabla XXIII) y rendimiento (Tabla XXIV) utilizados en este capítulo.

VI.4.5 Análisis por número de secuencias y conservación de los sitios de pegado

Un análisis de resultados usando simultáneamente porcentaje de conservación y el número de secuencias fue realizado, pero debido a la falta de uniformidad en cuanto al número de casos de prueba por intervalo, no fue posible llegar a una conclusión. Se puede observar que para los casos de prueba de Tompa *et al.* (2005), en cuanto a la sensibilidad (Figura 46), aquellos que tienen de 1 a 4 secuencias y con un porcentaje de conservación de 51% a 60% fueron los más altos. También se puede observar que la sensibilidad de aquellos casos de prueba con un porcentaje de conservación de 71% a 80% va en incremento con el número de secuencias. De igual manera, en cuanto a la especificidad (Figura 47), los que obtuvieron la puntuación más alta fueron aquellos

con un porcentaje de conservación de 61% a 70% con más de 9 secuencias.

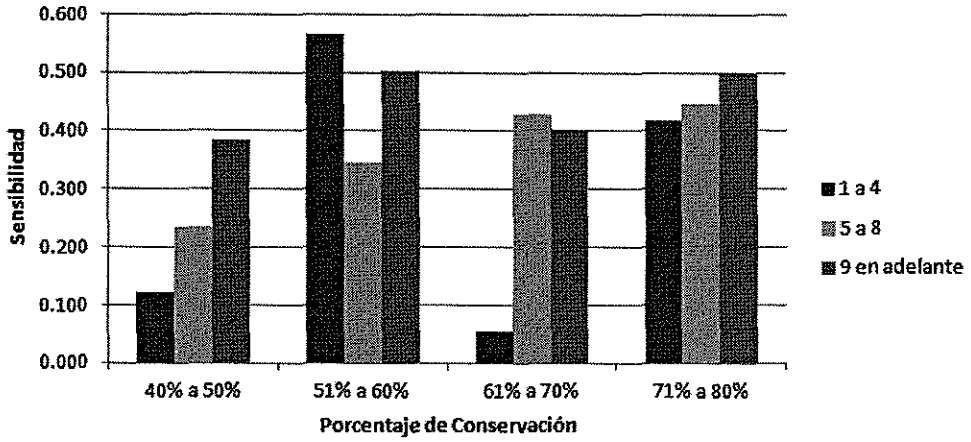


Figura 46. Sensibilidad de RM+BP+MM por número de secuencias y porcentaje de conservación en los casos de prueba.

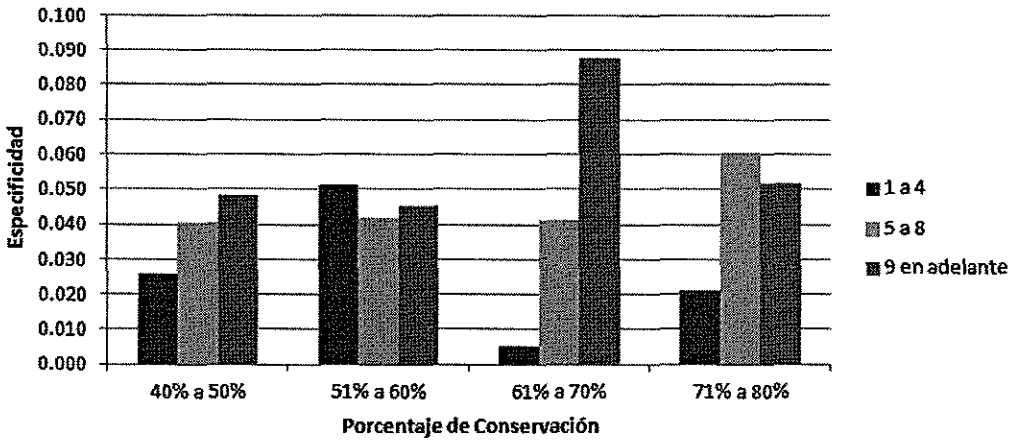


Figura 47. Especificidad de RM+BP+MM por número de secuencias y porcentaje de conservación en los casos de prueba.

VI.4.6 Prueba de Wilcoxon Signed Rank Test

Para evaluar si las mejorías por organismo (Sección VI.4.2) son estadísticamente significativas a un 5% de nivel de confianza, se aplicó la prueba de Wilcoxon Signed Rank

Tabla II. Prueba de Wilcoxon Rank Sum Test para la sensibilidad sobre casos de prueba del humano.

	RM+BP	RM+MM	RM+WR	RM+BP+MM
RM+BP	-	0.4807	1	**0.0001**
RM+MM		-	0.2379	**0.0000**
RM+WR			-	**0.0118**
RM+BP+MM				-

Tabla III. Prueba de Wilcoxon Rank Sum Test para la especificidad sobre casos de prueba del humano.

	RM+BP	RM+MM	RM+WR	RM+BP+MM
RM+BP	-	0.8238	0.2632	**0.0347**
RM+MM		-	0.8238	0.4049
RM+WR			-	**0.0066**
RM+BP+MM				-

Test.

Se puede observar en la Tabla II que la diferencia en la sensibilidad para el humano para todos los algoritmos es estadísticamente significativa. En la Tabla III se muestra cómo para el algoritmo RM+MEME, la mejoría de la exactitud no es estadísticamente significativa, lo cual es consistente con los resultados previos (Sección VI.4.2), donde se muestra como a pesar de que RM+BP+MM es quien obtiene la mejor especificidad, es MEME quien le sigue.

Para el ratón, se puede observar en la Tabla IV que la diferencia en la sensibilidad entre el RM+BP+MM y el resto de los algoritmos es estadísticamente significativa. En la Tabla V se muestra como para el algoritmo RM+BP+MM, la mejoría en la exactitud

Tabla IV. Prueba de Wilcoxon Rank Sum Test para la sensibilidad sobre casos de prueba del ratón.

	RM+BP	RM+MM	RM+WR	RM+BP+MM
RM+BP	-	0.5078	1	**0.0117**
RM+MM		-	0.1797	**0.0010**
RM+WR			-	**0.0054**
RM+BP+MM				-

Tabla V. Prueba de Wilcoxon Rank Sum Test para la especificidad sobre casos de prueba del ratón.

	RM+BP	RM+MM	RM+WR	RM+BP+MM
RM+BP	-	0.7138	0.2312	**0.0231**
RM+MM		-	0.8148	0.3049
RM+WR			-	**0.0026**
RM+BP+MM				-

con respecto a RM+MM no es estadísticamente significativa, debido a que es RM+MM quien obtiene la mejor sensibilidad según la evaluación de la Sección VI.4.2.

En cuanto a la levadura, se puede observar en la Tabla VI que la diferencia en la sensibilidad entre RM+BP+MM con el resto de los algoritmos es estadísticamente significativa. En la Tabla VII se muestra que para el algoritmo MEME, la mejoría en la exactitud no es estadísticamente significativa, ya que a pesar de que RM+BP+MM tiene una especificidad mayor, RM+MEME le sigue muy de cerca.

Para la mosca no se realizó una evaluación de Wilcoxon, debido a que el algoritmo RM+BP+MM no obtuvo mejoría según la evaluación de la Sección VI.4.2.

Se puede observar por lo anterior que RM+BP+MM no tiene una especificidad sig-

Tabla VI. Prueba de Wilcoxon Rank Sum Test para la sensibilidad sobre casos de prueba de la levadura.

	RM+BP	RM+MM	RM+WR	RM+BP+MM
RM+BP	-	0.4038	0.7842	**0.0025**
RM+MM		-	0.1452	**0.0006**
RM+WR			-	**0.0042**
RM+BP+MM				-

Tabla VII. Prueba de Wilcoxon Rank Sum Test para la especificidad sobre casos de prueba de la levadura.

	RM+BP	RM+MM	RM+WR	RM+BP+MM
RM+BP	-	0.4187	0.6754	**0.0134**
RM+MM		-	0.4568	0.2145
RM+WR			-	**0.0056**
RM+BP+MM				-

nificantiva diferente de RM+MEME, debido a que ambos algoritmos han mantenido una especificidad cercana. Esto es debido a la naturaleza del algoritmo combinado (RM+BP+MM), ya que es MEME también quien elige los motivos de las regiones candidatas. Pero a pesar de esto último, el algoritmo RM+BP+MM obtiene una mejoría significativa en la sensibilidad a comparación de RM+MEME, dicho de otra forma, se obtienen más sitios de pegado reales con la misma especificidad. Para el resto de los algoritmos, RM+BP+MM obtiene una diferencia significativa tanto en sensibilidad como en especificidad (Humano, Ratón y Mosca).

La evaluación propuesta por Tompa *et al.* (2005) difiere de la analizada en esta tesis en dos aspectos importantes: solo permitieron un motivo por cada caso de prueba y se permitió la consulta de biólogos y bibliografía para ayudar a seleccionar el motivo a reportar. En su evaluación, MEME tuvo una sensibilidad de 0.067 y una especificidad de 0.107, mientras que Weeder obtuvo una sensibilidad de 0.086 y una especificidad de 0.299, BioProspector no fue evaluado en Tompa *et al.* (2005). Aquellos casos de prueba que no reportaron ningún motivo, no fueron considerados al calcular la sensibilidad y especificidad, es por esto que ellos afirman que Weeder tuvo una alta sensibilidad y especificidad debido a que en 17 casos, no reportaron ningún motivo. También sugieren que limitar los algoritmos a reportar únicamente un motivo puede dar una alta especificidad, pero esto sólo hace que los algoritmos supongan que los casos de prueba únicamente tienen un sólo motivo, lo cual es falso.

Capítulo VII

Conclusiones y Trabajo a Futuro

En este trabajo se abordó el problema de búsqueda de motivos utilizando una estrategia de encadenamiento de algoritmos. Como un primer intento de entender el problema, se hizo una revisión bibliográfica sobre los algoritmos que intentan resolver la búsqueda de motivos. Basándonos en las ideas de Jianjun *et al.* (2005) y de Xiaohui *et al.* (2010), se implementó el encadenamiento BioProspector con MEME (BP+MM). Con el objetivo de mejorar la exactitud del encadenamiento, se preprocesó la información de los casos de prueba utilizando RepeatMasker para eliminar así fragmentos irrelevantes, dando lugar al encadenamiento RM+BP+MM. Se comparó la exactitud de dicho encadenamiento con los algoritmos componentes, así como con la información preprocesada.

Con el objetivo de mejorar la exactitud de la búsqueda de motivos, se implementó un segundo encadenamiento, RM+BP+WD+MM, al cual se le agregó Weeder y funciona de forma similar que RM+BP+MM.

Se utilizó el *benchmark* propuesto por Tompa *et al.* (2005), que consiste en casos de prueba reales de organismos eucariotas (humano, ratón, mosca y levadura), el cual se comprobó que contuviera únicamente secuencias promotoras de genes (Apéndice C).

Por último, se hizo un análisis de los resultados, así como de la composición de los casos de prueba, para ver si había casos específicos en los cuales los encadenamientos funcionaban mejor.

VII.1 Conclusiones

- Preprocesar los casos de prueba antes de ejecutar un algoritmo es importante. En todos los casos, preprocesar las secuencias mantuvo o mejoró la exactitud de los algoritmos MEME, BioProspector y Weeder.

- El número de falsos positivos en todos los algoritmos utilizados en este trabajo es elevado. Postprocesar la información de los algoritmos puede reducir dicho número. En promedio, por cada sitio de pegado verdadero, BioProspector nos dará 21.23 falsos positivos, RM+BioProspector 21.41 falsos positivos, MEME 23.17 falsos positivos, RM+MEME 22.33 falsos positivos, Weeder 59.72 falsos positivos y por último, RM+Weeder nos dará 56.42 falsos positivos. RM+BP+MM y RM+BP+WR+MM, generan 19.92 y 23.87 falsos positivos por cada sitio de pegado verdadero, respectivamente.

- La especificidad es el criterio más importante cuando se evalúan los algoritmos de búsqueda de motivos, debido a que actualmente resulta muy cara la comprobación experimental de los sitios de pegado, especialmente cuando se tiene un número alto de falsos positivos.

- El encadenamiento de algoritmos puede mejorar la búsqueda de motivos, aunque el tiempo de ejecución siempre será mayor al de sus algoritmos componentes. Ambos encadenamientos propuestos en este trabajo, mejoraron los resultados de los algoritmos que los componen. RM+BP+MM obtuvo la sensibilidad, rendimiento y especificidad más alta en comparación con Weeder, MEME y BioProspector, mientras que RM+BP+WR+MM únicamente superó en sensibilidad y rendimiento a los algoritmos anteriores.

- RM+BP+MM obtuvo un porcentaje de mejoría de sensibilidad con respecto a los algoritmos MEME (106.96%), RM+MEME (86.34%), BioProspector (60.65%), RM+BP (41.28%), Weeder (33.14%) y RM+Weeder (20.97%). También hubo mejoría en especificidad, MEME (5.88%), RM+MEME (0.48%), BioProspector (42.74%), RM+BP (13.33%), Weeder (100.99%) y RM+Weeder (66.23%). Aquellos que tuvieron un porcentaje bajo de mejoría en sensibilidad, obtuvieron un incremento mayor en especificidad, y viceversa.

- La conservación de los motivos, así cómo el número de secuencias en los casos de prueba son fundamentales para la exactitud de los algoritmos, donde el caso de prueba ideal, tendría una conservación alta en sus sitios de pegado, junto con un número de secuencias también alto.

- La búsqueda de motivos degenerados es todavía una tarea complicada para los algoritmos actuales, ya que la mayoría busca sitios de pegado con una conservación superior al del 50%.

- No todos los sitios de pegado tienen un consenso de caracteres en común. Algunos factores de transcripción no buscan un patrón de nucleótidos en común, sino mas bien, un conjunto combinado de nucleótidos que proporcionen una estructura tridimensional requerida para su pegado, tal como lo muestran Napoli *et al.* (2006) con el factor de transcripción CAP de la mosca, donde busca un patrón de energía que le permita doblar al ADN 46° para su pegado.

- Intentar resolver el problema de búsqueda de motivos con un solo algoritmo y una sola función objetivo para todos los organismos y factores de transcripción, es una tarea difícil en bioinformática.

- Definir criterios de evaluación más estrictos para los algoritmos de búsqueda de motivos es algo necesario.

VII.2 Trabajo a futuro

Para el paradigma actual de búsqueda de motivos, existe un número de condiciones que se deben cumplir para que un algoritmo tenga buena exactitud, las cuales son: en cuanto a las secuencias, su longitud, su distribución de nucleótidos, distribución de subcadenas y número de secuencias; en cuanto a los sitios de pegado, porcentaje de conservación, número de sitios de pegado, distribución en las secuencias y longitud; en cuanto al algoritmo, función objetivo, modelo (probabilístico, combinatorio, filogenético), longitud del motivo a buscar, mutaciones. Definir los valores de algunas de estas variables en los algoritmos, ayudaría a mejorar la exactitud. Para ayudar a solucionar este problema, se propone:

- Generar casos de prueba descriptivos y diversos por organismo (diversidad de número de secuencias por caso y variedad de sitios de pegado en cuanto a porcentaje de conservación), utilizando bases de datos con información actualizada.

- Para cada algoritmo y función objetivo, definir las características de los casos de prueba necesarias para una alta exactitud. Debido a que inicialmente solo se conocen las características técnicas del caso de prueba (número de secuencias, longitud de secuencias y distribución de nucleótidos y de subcadenas), tener una función objetivo dependiendo de las características del caso de prueba podría ayudar a su mejoría. Independientemente del organismo que se analice, si dos casos de prueba contienen las mismas características, es posible tener el mismo porcentaje de exactitud.

Para solucionar la búsqueda de motivos de sitios de pegado muy degenerados o mejorar la especificidad de los motivos conservados, se tiene que utilizar información adicional, tal como patrones de energía y estructuras tridimensionales del ADN.

VII.3 Productos de Investigación

- Herramienta para la búsqueda de motivos en secuencias de ADN para casos reales, la cual contiene los algoritmos MEME, Weeder y BioProspector entrelazados con programación en PERL, además de RepeatMasker y su base de datos RepBase para el preprocesamiento opcional de secuencias de ADN.

- Plataforma para el fácil encadenamiento de algoritmos para la búsqueda de motivos, programada en PERL y SHELL de Linux, en la cual es posible agregar diferentes programas de búsqueda de motivos según se requiera.

Referencias

Bailey, T. y Elkan, C. (1995). Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, **21**: 51–80.

Bailey, T. y Elkan, C. (2005). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *UCSD Technical Report CS94-351*.

Bailey, T., Bodén, M., Whitington, T., y Machanick, P. (2010). The value of position-specific priors in motif discovery using meme. *BMC Bioinformatics*, **12**: 1–12.

Benson, D., Karsch, M., Lipman, D., Ostell, J., y Wheeler, D. (2008). Genbank. *Nucleic Acids Res..*, **110**: 462–467.

Benson, G. (1999). Tandem repeats finder: a program to analyze dna sequences. *Nucleic Acids Research*, **27**: 573–580.

Berezikov, E., Guryev, V., Plasterk, R., y Cuppen, E. (2004). Conreal: Conserved regulatory elements anchored alignment algorithm for identification of transcription factor binding sites by phylogenetic footprinting. *Genome Res*, **14**: 170–178.

Bergers, G., Graninger, P., Braselmann, S., y Wrighton, C. (1995). Transcriptional activation of the fra-1 gene by ap-1 is mediated by regulatory sequences in the first intron. *Mol Cell Biol*, **15**: 3748–3758.

Blanchette, M. y Tompa, M. (2002). Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Res*, **12**: 739–748.

Blanchette, M., Schwikowski, B., y Tompa, M. (2002). Algorithms for philogenetic footprinting. *J. Comp. Biol.*, **9**: 211–223.

Brazma, A., Jonassen, I., Vilo, J., y Ukkonen, E. (1998). Predicting gene regulatory elements *in silico* on a genomic scale. *Genome Res.*, **15**: 1202–1215.

Bucher, P. (1990). Weight matrix description for four eukaryotic rna polymerase ii promoter element derived from 502 unrelated promoter sequences. *J Mol Biol*, **212**: 563–578.

Buhler, J. y Tompa, M. (2002). Finding motifs using random projections. *Journal of Computational Biology*, **9**(2): 225–242.

Carmack, C., McCue, L., Newberg, L., y Lawrence, C. (2007). Phyloscan: identification of transcription factor binding sites using cross-species evidence. *Algorithms for Molecular Biology*, **2**: 14–19.

Chan, A. (2004). An analysis of pairwise sequence alignment algorithm complexities. *Algorithms for Molecular Biology*, **111**: 451–457.

Cliften, P., Hillier, L., Fulton, L., Graves, T., Miner, T., Gish, W., Waterston, R., y Johnston, M. (2001). Surveying saccharomyces genomes to identify functional elements by comparative dna sequence analysis. *Genome Res*, **11**: 1175–1186.

Cliften, P., Sudarsanam, P., Desikan, A., Fulton, L., Fulton, B., Majors, J., Waterston, R., Cohen, B., y Johnston, M. (2003). Finding functional features in saccharomyces genomes by phylogenetic footprinting. *Science*, **301**: 71–76.

Colgan, J. y Manley, J. L. (1995). Cooperation between core promoter elements influences transcriptional activity in vivo. *Proc Natl Acad Sci*, **92**: 1955–1959.

Conaway, J. W. y Conaway, R. C. (1995). Initiation of eukaryotic messenger rna synthesis. *J Biol Chem*, **31**: 1255–1259.

Cramer, P., Pesce, C. G., Baralle, F. E., y R., K. A. (1997). Functional association between promoter structure and transcript alternative splicing. *Proc Natl Acad Sci*, **94**: 11456–11460.

Das, M. y Dai, H.-K. (2007). A survey of dna motif finding algorithms. *BMC Bioinformatics*, **8**(Suppl 7): 21–28.

Doi, K., Hosaka, A., Nagata, T., Satoh, K., y Suzuki, K. (2008). Development of a novel data mining tool to find cis-elements in rice gene promoter regions. *BMC Plant Biology*, **90**: 1156–1170.

Down, T. y Hubbard, T. (2005). Nestedmica: sensitive inference of over-represented motifs in nucleic acid sequence. *Nucleic Acids Res*, **33**: 1445–1453.

Drummond, A., Ashton, B., Buxton, S., Cheung, M., Cooper, A., Duran, C., Field, M., Heled, J., Kearse, M., Markowitz, S., Moir, R., Stones-Havas, S., Sturrock, S., y Thierer, T. (2011). Geneious v5.4, available from http://www.geneious.com/.

Eskin, E. y Pevzner, P. (2002). Finding composite regulatory patterns in dna sequences. *Bioinformatics*, **18**(Suppl 1): S354–S363.

Espinoza, M. (2004). Analisis de algoritmos de busqueda de motivos en secuencias de adn.

Fatemeh, Z.-M., Hayedeh, A., Mehdei, S., Abbas, N.-D., y Bahram, G. (2009). New scoring schema for finding motifs in dna sequences. *BMC Bioinformatics*, **147**: 195–197.

Galas, D., Eggert, M., y Waterman, M. (1985). Rigorous pattern-recognition methods for dna sequences: analysis of promoter sequences from escherichia coli. *J Mol Biol*, **186**: 117–128.

Gama-Castro, S., Salgado, H., Peralta-Gil, M., Santos-Zavaleta, A., Muniz-Rascado, L., Solano-Lira, H., Jimenez-Jacinto, V., Weiss, V., Garcia-Sotelo, J. S., Lopez-Fuentes, A., Porron-Sotelo, L., Alquicira-Hernandez, S., Medina-Rivera, A., Martinez-Flores, I., Alquicira-Hernandez, K., Martinez-Adame, R., Bonavides-Martinez, C., Miranda-Rios, J., Huerta, A. M., Mendoza-Vargas, A., Collado-Torres, L., Taboada, B., Vega-Alvarado, L., Olvera, M., Olvera, L., Grande, R., Morett, E., y Collado-Vides, J. (2010). Regulondb (version 7.0): transcriptional regulation of escherichia coli k-12 integrated within genetic sensory response units (gensor units). *Nucleic Acids Res...*

Geir, K. S., Osman, A., Vegard, W., y Finn, D. (2007). Improved benchmarks for computational motif discovery. *BMC Bioinformatics*, **8**: 144–151.

Gelfand, M., Koonin, E., y Mironov, A. (2000). Prediction of transcription regulatory sites in archaea by a comparative genome approach. *Nucleic Acids Res*, **28**: 695–705.

Gilinger, G. y Alwine, J. C. (1993). Transcriptional activation by simian virus-40 large t-antigen - requirements for simple promoter structures containing either tata or initiator elements with variable upstream factor binding sites. *J Virol.*

Gotoh (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**: 705–708.

Hansen, P., Mladenovic, N., y Urosevic, D. (2006). Variable neighborhood search and local branching. *Computers and Operations Research*, **33**: 3034–3045.

Hertz, G. y Stormo, G. (1999). Identifying dna and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**: 563–577.

Hertz, G., Hartzell, G., y Stormo, G. (1990). Identification of consensus patterns in unaligned dna sequences known to be functionally related. *Comput Appl Biosci*, **6**: 81–92.

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

Hu, J., Li, B., y Kihara, D. (2005). Limitations and potentials of current motif discovery algorithms. *Nucleic Acids Res*, **33**: 4899–4913.

Hu, J., Yang, Y., y Kihara, D. (2006). Emd: an ensemble algorithm for discovering regulatory motifs in dna sequences. *BMC Bioinformatics*, **7**: 342.

Ioshikhes, I., Bolshoy, A., Derenshteyn, K., Borodovsky, M., y Trifonov, E. N. (1996). Nucleosome dna sequence pattern revealed by multiple alignment of experimentally mapped sequences. *J. Mol. Biol.*, **262**: 129–139.

Jianjun, H., Li, B., y Kihara, D. (2005). Limitations and potentials of current motif discovery algorithms. **110**: 462–467.

Jurka, J., Kapitonov, V., Pavlicek, A., Klonowski, P., Kohany, O., y Walichiewicz, J. (2005). Repbase update, a database of eukaryotic repetitive elements. *Cytogentic and Genome Research.*, **110**: 462–467.

Kaplan, T., Friedman, N., y Margalit, H. (2005). Ab initio prediction of transcription factor targets using structural knowledge. *PLoS Comput Biol*, **1**(1): e1.

Kellis, M., Patterson, N., Endrizzi, M., Birren, B., y Lander, E. (2003). Sequencing and comparison of yeast species to identify genes and regulatory element. *Nature*, **423**: 241–254.

Kingsford, C., Zaslavsky, E., y Singh, M. (2006). A compact mathematical programming formulation for dna motif finding. *Lecture Notes in Computer Science*, **4009**: 233–245.

Kodadek, T. (1998). Mechanistic parallels between dna replication, recombination and transcription. *Trends Biochem. Sci.*, **23**: 79–83.

Kuo, M. y Allis, C. (1999). In vivo cross-linking and immunoprecipitation for studying dynamic protein: Dna associations in a chromatin environment. *BMC Bioinformatics*, **19**: 425–433.

Lawrence, C. y Reilly, A. (1990). An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, **7**: 41–51.

Lawrence, C., Altschul, S., Boguski, M., Liu, J., Neuwald, A., y Wootton, J. (1993). Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, **262**: 208–214.

Lengauer, T. (2002). *From genomes to drugs*. Wiley-VCH.

Liang, S. (2003). cwinnower algorithm for finding fuzzy dna motifs. *IEEE Computer Society Bioinformatics Conference*, páginas 260–265.

Liu, D., Xiong, X., DasGupta, B., y Zhang, H. (2006). Motif discoveries in unaligned molecular sequences using self-organizing neural network. *IEEE Transactions on Neural Networks*, **17**: 919–928.

Liu, F., Tsai, J., Chen, R., Chen, S., y Shih, S. (2004). Fmga: finding motifs by genetic algorithm. *Fourth IEEE Symposium on Bioinformatics and Bioengineering*, página 459.

Liu, J. (1994). The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *J Amer Statist Assoc*, **89**: 958–966.

Liu, J., Neuwald, A., y Lawrence, C. (1999). Motif statistics. *J Amer Statist Assoc*, **90**: 1156–1170.

Liu, X., Brutlag, D., y Liu, J. (2001). Bioprospector: discovering conserved dna motifs in upstream regulatory regions of co-expressed genes. *Proceedings of the Sixth Pacific Symposium on Biocomputing*, páginas 127–138.

Liu, X., Brutlag, D., y Liu, J. (2002). An algorithm for finding protein-dna binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nat Biotechnol*, **20**: 835–839.

Marsan, L. y Sagot, M. (2000). Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *J Comput Biol*, **7**: 345–362.

McCue, L., Thompson, W., Carmack, C., Ryan, M., Liu, J., Derbyshire, V., y Lawrence, C. (2001). Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. *Nucleic Acids Res*, **29**: 774–782.

Moses, A., Chiang, D., y Eisen, M. (2004). Phylogenetic motif detection by expectation-maximization on evolutionary mixtures. *Proceedings of the Ninth Pacific Symposium on Biocomputing*, páginas 324–335.

Napoli, A., Lawson, C., Ebright, R., y Berman, H. (2006). Indirect readout of dna sequence at the primary-kink site in the cap-dna complex: recognition of pyrimidine-purine and purine-purine steps. *J. Mol. Biol.*, **357**: 173–183.

Novina, C. D. y Roy, A. L. (1996). Core promoters and transcriptional control. *Trends in genetics*, **12**: 351–355.

Pavesi, G., Mauri, G., y Pesole, G. (2001). An algorithm for finding signals of unknown length in dna sequences. *Bioinformatics*, **17**(Suppl 1): S207–S214.

Pevzner, P. y Sze, S. (2000). Combinatorial approaches to finding subtle signals in dna sequences. *Proceedings of the Eighth International Conference on Intelligent Systems on Molecular Biology, San Diego, CA*, páginas 269–278.

Porteus, I., Asuncion, A., Newman, D., y Smyth, P. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation. *KDD*.

Prakash, A., Blanchette, M., Sinha, S., y Tompa, M. (2004). Motif discovery in heterogeneous sequence data. *Proceedings of the Ninth Pacific Symposium on Biocomputing*, páginas 348–359.

Price, A., Ramabhadram, S., y Pevzner, P. (2003). Finding subtle motifs by branching from sample strings. *Bioinformatics*, **1**(1): 1–7.

Redner, R. y Walker, H. (1984). Mixture densities, maximum likelihood, em algorithm. *SIAM Review*, **26**: 195–239.

Roth, F., Hughes, J., Estep, P., y Church, G. (1998). Finding dna regulatory motifs within unaligned noncoding sequences clustered by whole-genome mrna quantitation. *Nature Biotechnology*, **16**: 939–945.

Sagot, M.-F. (1998). Spelling approximate repeated or common motifs using a suffix tree. En C. L. Lucchesi y A. V. Moura, editores, *LATIN*, Vol. 1380 de *Lecture Notes in Computer Science*, páginas 374–390. Springer. ISBN 3-540-64275-7.

Sandelin, A., Alkema, W., Engstrom, P., Wasserman, W., y Lenhard, B. (2004). Jaspar: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.*.

Sap, J., Muñoz, A., Schmitt, J., y Stunnenberg, H. (1989). Repression of transcription mediated at a thyroid hormone response element by the v-erb-a oncogene product. *Nature*, **340**: 242–244.

Schneider, T. y Stephens, R. (1990). Sequence logos: a new way to display consensus sequence. *Nucleic Acids Res*, **18**: 6097–6100.

Schneider, T., Stormo, G., y Gold, L. (1986). Information content of binding sites on nucleotide sequences. *J. Mol. Biol.*, **39**: 1–38.

Shida, K. (2006). Gibbsst: a gibbs sampling method for motif discovery with enhanced resistance to local optima. *BMC Bioinformatics*, **7**: 486.

Siddharthan, R., Siggia, E., y van Nimwegen, E. (2005). Phylogibbs: A gibbs sampling motif finder that incorporates phylogeny. *PLoS Comput Biol*, **1**: 534–556.

Sinha, S. y Tompa, M. (2000). A statistical method for finding transcription factor binding site. *Proceedings of the Eighth International Conference on Intelligent Systems on Molecular Biology, San Diego, CA*, páginas 344–354.

Sinha, S. y Tompa, M. (2003). Performance comparison of algorithms for finding transcription factor binding sites. *Third IEEE Symposium on Bioinformatics and Bioengineering*, páginas 214–220.

Sinha, S., Blanchette, M., y Tompa, M. (2004). Phyme: A probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics*, **5**: 170.

Sloan, L. S., S. A. (1998). Sequence determinants of the intrinsic bend in the cyclic amp response element. *Biochemistry*, **37**: 7113–7118.

Smit, A., Hubley, R., y Green, P. (2010). Repeatmasker open-3.0. *http://www.repeatmasker.org*.

Smith, T. F. y Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**: 195–197.

Tagle, D., Koop, B., Goodman, M., Slightom, J., Hess, D., y Jones, R. (1988). Embryonic epsilon and gamma globin genes of a prosimian primate (galago crassicaudatus): nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. *J Mol Biol*, **203**: 439–455.

Thijs, G., Marchal, K., y Moreau, Y. (2001). A gibbs sampling method to detect over-represented motifs in upstream regions of co-expressed genes. *RECOMB*, **5**: 305–312.

Thompson, J., Higgins, D., y Gibson, T. (1994). Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Res*, **22**: 4673.

Tomovic, A., Stadler, M., y Oakeley, E. (2009). Position dependencies in transcription factor binding sites. *BMC Bioinformatics*, **12**: 1–12.

Tompa, M. (1999). An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. *Proceedings of the Seventh International Conference on Intelligent Systems on Molecular Biology*, páginas 262–271.

Tompa, M. (2001). Identifying functional elements by comparative dna sequence analysis. *Genome Res*, **11**: 1143–1144.

Tompa, M., Li, N., L. Bailey, T., M. Church, G., De Moor, B., Eskin, E., V. Favorov, A., C. Frith, M., Fu, Y., Kent, W. J., J. Makeev, V., A. Mironov, A., Stafford Noble, W., Pavesi, G., Pesole, G., Régnier, M., Simonis, N., Sinha, S., Thijs, G., van Helden, J., Vandenbogaert, M., Weng, Z., Workman, C., Ye, C., y Zhu, Z. (2005). Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, **23**: 137–144.

Van Helden, J., Andre, B., y Collado-Vides, J. (1998). Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J Mol Biol*, **281**: 827–842.

Van Helden, J., Rios, A., y Collado-Vides, J. (2000). Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res*, **28**: 1808–1818.

Vanet, A., Marsan, L., Labigne, A., y Sagot, M. (2000). Inferring regulatory elements from a whole genome. an analysis of helicobacter pylori sigma80 family of promoter signals. *J Mol Biol*, **297**: 335–353.

Wang, T. y Stormo, G. (2003). Combining phylogenetic data with coregulated genes to identify regulatory motifs. *Bioinformatics*, **19**: 2369–2380.

Wang, T. y Stormo, G. (2005). Identifying the conserved network of cis-regulatory sites of a eukaryotic genome. *PNAS*, **102**: 17400–17405.

Weinberg, Z., Barrick, J., Yao, Z., Roth, a., y Kim, J. (2007). Identification of 22 candidate structured rnas in bacteria using the cmfinder comparative genomics pipeline. *Nucleic Acids Research*, **90**: 1156–1170.

Wen-Hsiung, L. (1997). Molecular evolution. *Sinauer Associates inc.*.

Wingender, E., Dietze, P., Karas, H., y Knuppel, R. (1996). Transfac: a database on transcription factors and their dna binding sites. *Nucleic Acids Res*, **24**: 238–241.

Xiaohui, C., Lin, H., Naifang, S., Haiyan, H., Minghua, D., y Xiaoman, L. (2010). Systematic identification of conserved motif modules in the human genome. *BMC Genomics*.

Yamauchi, M., Ogata, Y., Kim, R., Li, J., y Freedman, L. (1996). Ap-1 regulation of the rat bone sialoprotein gene transcription is mediated through a tpa response element within a glucocorticoid response uniit in the gene promoter. *Matrix Biol.*, **15**: 119–139.

Zawel, L. y Reinberg, D. (1995). Common themes in assembly and function of eukaryotic transcription complexes. *Annu. Rev. Biochem.*, **64**: 533–561.

Apéndice A

Principios organizacionales de las regiones promotoras

Los sitios de pegado sin promotores no muestran ningún patrón general con respecto a localización y orientación dentro de las secuencias promotoras. Incluso sitios de pegado importantes para un factor de transcripción en específico pueden ocurrir en casi cualquier lugar dentro de un promotor. Por ejemplo, el AP-1 (proteína activadora, un complejo de dos factores de transcripción), se pega muy lejos del inicio de la transcripción, como en el caso del gen de la sialoproteína en la rata, donde el AP-1 se pega casi 900 nucleótidos de distancia del inicio de la transcripción (Yamauchi *et al.*, 1996). También el AP-1 es necesario para el virus de la leucemia Moloney Murine, donde se pega muy cerca al sitio de inicio de la transcripción (Sap *et al.*, 1989). Otros sitios funcionales para el AP-1 se encuentran fuera de la región promotora, como es el caso para el gen de la proopiomelanocortina y del gen fra-1 (Bergers *et al.*, 1995). Ejemplos similares pueden encontrarse, en donde un mismo factor de transcripción puede pegarse en regiones diferentes, por lo que una región de pegado no puede definirse.

El contexto de los sitios de pegado es uno de los mayores determinantes en el control de la transcripción. Como consecuencia de los requerimientos del contexto, es que los sitios de pegado están usualmente agrupados y tales grupos funcionales han sido descritos en muchos casos. Subconjuntos de grupos de sitios de pegado que tienen una función independiente en específico del promotor son llamados módulos promotores. Estos módulos contienen muchos sitios de pegado que actúan juntos para realizar una

función común, como la de la expresión de un tejido en específico. Dentro de un modulo promotor, tanto el orden secuencial como la distancia son cruciales para la función, indicando que estos módulos pueden ser determinantes críticos de un promotor, mucho más que los sitios de pegado individuales. Los módulos promotores siempre están constituidos por más de un sitio de pegado. Como los promotores pueden contener varios módulos que utilizan conjuntos traslapados de sitios de pegado, el contexto conservado de un sitio de pegado en particular no puede ser determinado de la secuencia primaria. Los módulos correspondientes deben ser detectados separadamente antes de que la estructura modular funcional de un promotor o de cualquier otra región promotora en ADN pueda ser revelada por un análisis computacional. Uno de los módulos promotores mejor conocidos es el promotor mínimo (Zawel y Reinberg, 1995).

A.1 Propiedades modulares del promotor mínimo

El módulo del promotor mínimo puede ser definido por su capacidad de armar el complejo iniciador de la transcripción (ver Figura 48), y orientarlo hacia el sitio de inicio de la transcripción (TSS) del promotor, definiendo la localización exacta del TSS. Varias combinaciones de cuatro elementos que constituyen un promotor mínimo general pueden lograr esto. Este módulo incluye la caja TATA, la región iniciadora (TSS), un elemento activador corriente arriba (a la izquierda del 5′) y un elemento corriente abajo (a la derecha del 3′). La caja TATA es un elemento básico de la transcripción, la cual está localizada cerca de 20 a 30 nucleótidos del TSS, es el sitio de unión tanto de los factores de transcripción como de las histonas y está implicada en el proceso de transcripción por la ARN polimerasa. No todos los cuatro elementos son requeridos, o algunos elementos pueden tener tantas variaciones como para ser reconocidos por las herramientas

computacionales.

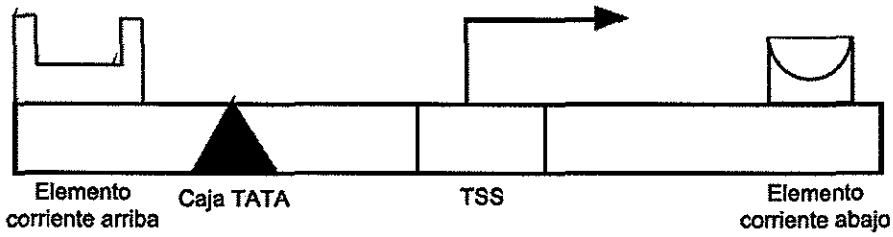


Figura 48. Estructura general del promotor mínimo de la polimerasa II. TSS = región iniciadora. Las formas sobre la barra representan sitios de pegado de proteína adicionales y la flecha representa el sitio de inicio de la transcripción. Tomado de Lengauer (2002).

A.1.1 Primer grupo: Caja TATA que contiene promotores sin un iniciador conocido

El posicionamiento satisfactorio del complejo de iniciación puede comenzar en la caja TATA conteniendo promotores del complejo TFIID, el cual contiene la proteína de pegado de la caja TATA así como otros factores de pegado. Junto con otro complejo de factores de transcripción (TFIIB), lleva al ensamble del complejo de iniciación (Conaway y Conaway, 1995). Si un sitio de pegado corriente arriba coopera con la caja TATA, ningún iniciador especial o secuencia corriente abajo puede ser requerida, lo cual permite el ensamble de un módulo mínimo promotor funcional con solo dos de los cuatro elementos.

A.1.2 Segundo grupo: Promotores sin caja TATA con un iniciador funcional

La caja TATA no es un elemento esencial de un promotor funcional. Una TSS combinada con un solo elemento corriente arriba también puede ser capaz de iniciar la transcripción (Gilinger y Alwine, 1993), aunque los iniciadores no han sido claramente definidos a nivel secuencial hasta ahora. Generalmente una región de 10 a 20 nucleótidos alrededor del TSS es suficiente para representar un iniciador.

A.1.3 Tercer grupo: Caja TATA con un iniciador funcional

La combinación de la caja TATA y un iniciador puede encontrarse en muchos promotores virales y se ha demostrado que un sitio de pegado adicional corriente arriba puede influenciar si la caja TATA o el elemento iniciador es quien determinará las propiedades del promotor (Colgan y Manley, 1995). Se sabe también que elementos corriente arriba pueden incrementar significativamente la eficiencia de la TSS con esta combinación.

A.1.4 Cuarto grupo: Promotores nulos con únicamente elementos corriente arriba y corriente abajo

Los promotores nulos que no contienen ni la caja TATA ni un iniciador, únicamente funcionan con elementos corriente arriba y corriente abajo (Novina y Roy, 1996).

Al menos estos cuatro tipos de promotores mínimos han sido identificados hasta ahora, si además de estas combinaciones se consideran los elementos corriente arriba y corriente abajo, se tendría un total de 7 posibles promotores mínimos.

Aparentemente el único común denominador de la iniciación de la transcripción en un promotor es la existencia de al menos un elemento del promotor mínimo dentro

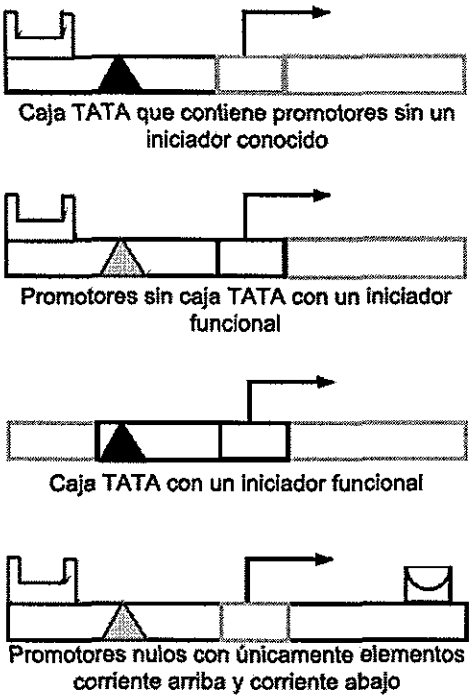


Figura 49. Las 4 variaciones de un promotor mínimo para la polimerasa II en eucariotas. Tomado de Lengauer (2002).

de cierta región. Esta deducción es incorrecta, pues tanto el espaciado como el orden secuencial de los elementos dentro del módulo del promotor mínimo son de mucha importancia a pesar de la presencia o ausencia de elementos individuales. Muchos promotores distintivos tienen requerimientos de elementos corriente arriba o corriente abajo y van a funcionar con su factor de transcripción específico. Mover el iniciador, la caja TATA y elementos corriente arriba puede tener efectos en las funciones promotoras y afectar la expresión génica (Cramer *et al.*, 1997).

La Figura 50 muestra una representación de un promotor pol II con un complejo de iniciación armado, la cual ilustra que sí importa dónde una proteína en específico se pega al ADN para permitir un ensamble apropiado del rompecabezas molecular del complejo de la iniciación.

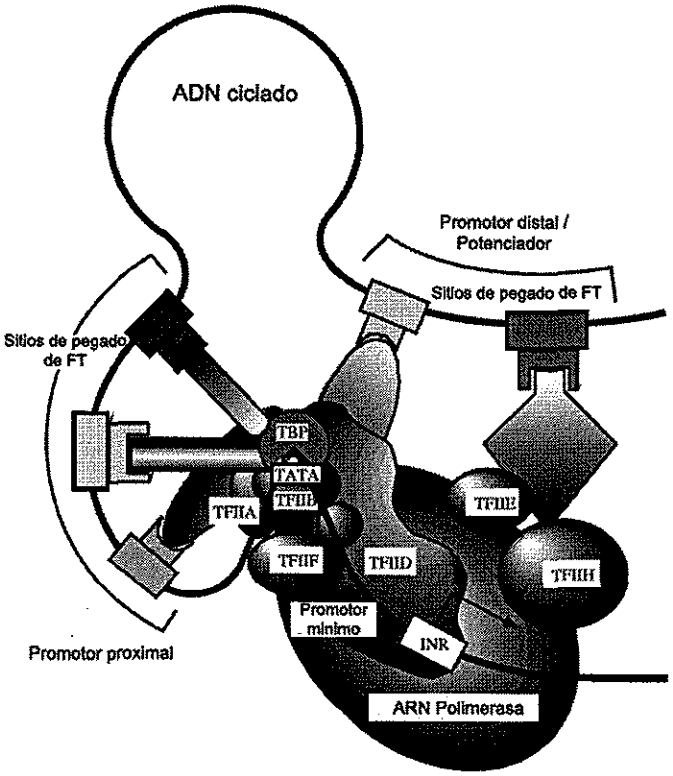


Figura 50. El complejo de la iniciación de la transcripción. Tomado de Lengauer (2002).

A.2 Tipos de regiones promotoras

A.2.1 Región de matriz adjunta

Un lazo de cromatina es una región del ADN cromosomal localizado entre dos puntos de contacto del ADN con la matriz nuclear marcada por la región de matriz adjunta. La matriz nuclear es un acoplamiento de proteínas alineándose con la superficie interna de la envoltura nuclear. La regulación transcripcional requiere la asociación del ADN con esta matriz nuclear, la cual mantiene una variedad de proteínas promotoras. Las regiones de matriz adjunta están compuestas de diferentes elementos los cuales incluyen sitios de pegado de factores de transcripción, potencial ADN cruciforme, segmentos ricos

en AT, por nombrar los más importantes (Lengauer, 2002).

A.2.2 Potenciadores y Silenciadores

Los potenciadores son regiones promotoras que pueden aumentar significativamente el nivel de transcripción de un promotor sin importar su orientación y distancia con respecto al promotor mientras se encuentren localizados dentro del mismo lazo de cromatina. Los silenciadores son básicamente idénticos a los promotores y con los mismos requerimientos, pero tienen un efecto negativo en las actividades del promotor.

A.2.3 Promotores

Los promotores son regiones en el ADN que son necesarias para iniciar la transcripción (inicio de la síntesis del ARN) y consiste en tres regiones básicas (ver Figura 49). La parte que determina los nucleótidos exactos para la iniciación de la transcripción es el llamado promotor mínimo y es una secuencia de ADN donde la ARN polimerasa y sus cofactores se ensamblan en el promotor.

La siguiente región corriente arriba del promotor mínimo es conocído como promotor proximal y usualmente contiene un número de sitios de pegado de factores de transcripción responsables del ensamble de un complejo de activación. Es generalmente aceptado que en eucariotas, los elementos promotores proximales se encuentren localizados a una distancia de entre 250 a 500 nucleótidos de distancia corriente arriba del sitio de inicio de la transcripción.

La tercer parte del promotor está localizada mucho más lejos y es llamado el promotor distal. Esta región usualmente regula la actividad del promotor mínimo y proximal y también contiene sitios de pegado de factor de transcripción. Aunque las regiones

promotoras distales y los potenciadores no muestran diferencia alguna, si una región promotora distal actúa independientemente a la posición y orientación, entonces es llamada potenciadora.

A.2.4 ¿Cómo se identifican los sitios de pegado en ADN experimentalmente?

Usualmente el punto de inicio es un fragmento de ADN amplificado por PCR que se cree contiene un sitio de pegado para una proteína en particular. Para determinar experimentalmente si una proteína en particular se pega, es necesario tener la proteína de pegado parcialmente purificada almacenada.

Una forma de identificar si el sitio de pegado está presente en el fragmento de ADN es utilizando un ensayo de cambio en la corrida electroforética. La técnica se basa en una corrida electroforética entre fragmentos de ADN marcados y la presencia de extractos celulares. El ensayo básico para el ADN consiste en un carril solo con el fragmento de ADN de interés marcado. Una segundo carril con el mismo ADN más proteínas que, en las condiciones dadas no interaccionan con el mismo, y el tercer carril consiste en el mismo fragmento de ADN más las proteínas con supuesta interacción con esa secuencia. Dependiendo del resultado del tercer carril se puede determinar si existió o no interacción. Un fragmento de ADN interaccionando con proteínas tiende a recorrer menos distancia a través del gel en el mismo tiempo que en los carriles 1 o 2, y por lo tanto se debería ver retrasado.

Cuando se tiene alguna proteína que se sabe se pega al ADN pero no se tiene la región de interés de ADN, es posible colectar fragmentos de ADN que contienen el sitio de pegado utilizando una técnica denominada inmunoprecipitación de cromatina ($ChIP$).

Esta técnica consiste en el uso de un anticuerpo que reconozca la proteína de interés no solamente en disolución sino también en la cromatina. La $ChIP$ consta básicamente de dos pasos, entrecruzamiento con formaldehído del ADN a las proteínas unidas a éste *in vivo* en células para que se fijen las interacciones proteína-proteína y las interacciones proteína-ADN seguido de la inmunoprecipitación de los complejos proteína-ADN con anticuerpos específicos a partir de extractos sonicados para fragmentar la cromatina. Las secuencias específicas de ADN inmunoprecipitadas son entonces amplificadas por PCR para determinar si han sido o no enriquecidas en las muestras correspondientes para cada anticuerpo (Kuo y Allis, 1999).

Apéndice B

Algoritmos de búsqueda de motivos

A continuación, una compilación de algoritmos de búsqueda de motivos:

Tabla VIII: Lista de algoritmos de búsqueda de motivos.

Algoritmo	Principio Operacional	Modelo	Características	Referencia
AlignACE	Gibbs *sampling*	Probabilístico	Mejores resultados en eucariotas simples. Es utilizado principalmente en la levadura.	Roth *et al.* (1998) *Nature Biotech.*
Bioprospector	Gibbs *sampling*	Probabilístico	Mejores resultados en procariotas y eucariotas simples. Los autores utilizaron casos de prueba de levadura y *E.coli*. Encuentra motivos palindrómicos.	Liu *et al.* (2001) *Pacific Biocomp.*

Continuación en la siguiente página.

Tabla VIII – continúa de la página anterior

Algoritmo	Principio Operacional	Modelo	Características	Referencia
Consensus	Matriz de pesos	Probabilístico	El uso de su función mejorada para calificar motivos es útil para identificar motivos débiles. Ha encontrado sitios de pegado en *E.coli.*	Hertz y Stormo (1999) *Comput. Appl. Biosci.*
Dyad-Analysis	Enumeración	Combinatorio	Eficiente al encontrar motivos espaciados en la levadura.	Van Helden *et al.* (2000) *Nucleic Acids Res.*
EM	Esperanza Máxima	Probabilístico	Encuentra motivos en proteínas. Cada secuencia debe de tener un motivo en común.	Lawrence y Reilly (1990) *Proteins*

Continuación en la siguiente página.

Tabla VIII – continúa de la página anterior

Algoritmo	Principio Operacional	Modelo	Características	Referencia
EMD	Ensamblado	Probabilístico	Combina predicciones de múltiples algoritmos. Eficiente para secuencias cortas. Fue probado utilizando casos de prueba del *E.coli*, tomados de RegulonDB.	Hu *et al.* (2006) *BMC Bioinformatics*
FMGA	Algoritmo genético	Bioinspirado	Mejores resultados con secuencias menores a 2000 nucleótidos. Encuentra motivos espaciados.	Liu *et al.* (2004) *IEEE Bioinformatics*
Footprinter	Programación dinámica	Combinatorio	Basado en huellas filogenéticas.	Blanchette y Tompa (2002) *Genome Res.*

Continuación en la siguiente página.

Tabla VIII – continúa de la página anterior

Algoritmo	Principio Operacional	Modelo	Características	Referencia
Gibbs sampler	Gibbs *sampling*	Probabilístico	Encuentra varios motivos por secuencia. El alineamiento de secuencias puede encontrar motivos en procariotas y eucariotas simples.	Lawrence *et al.* (1993) *Science*
GibbsST	Gibbs *sampling*	Probabilístico	Simula el temple de la termodinámica. Fue probado en casos sintéticos y en promotores de la levadura.	Shida (2006) *BMC Bioinformatics*
MaMF	Enumeración	Combinatorio	Utiliza índices dentro del genoma humano para una búsqueda rápida de motivos.	Hon y Jain (2006) *BMC Bioinformatics*

Continuación en la siguiente página.

Tabla VIII – continúa de la página anterior

Algoritmo	Principio Operacional	Modelo	Características	Referencia
MDScan	Algoritmo Voraz	Probabilístico	Encuentra motivos en procariotas. Utilizado con promotores del *E.coli*.	Liu *et al.* (2002) *Nature Biotech.*
MEME	Esperanza Máxima	Probabilístico	Elimina estadísticamente motivos encontrados, por lo que encuentra motivos diferentes. Puede encontrar cero, uno o más motivos por secuencia.	Bailey y Elkan (1995) *Machine Learning*
MITRA	Árbol de Prefijos / Grafo	Combinatorio	Buenos resultados en motivos compuestos. Ha sido probado en promotores de la levadura.	Eskin y Pevzner (2002) *BMC Bioinformatics*

Continuación en la siguiente página.

Tabla VIII – continúa de la página anterior

Algoritmo	Principio Operacional	Modelo	Características	Referencia
MotifSampler	Gibbs *sampling*	Probabilístico	El alineamiento de secuencias puede encontrar motivos en procariotas y eucariotas simples. Es utilizado con promotores bacterianos.	Thijs *et al.* (2001) *RECOMB*
Oligo-Analysis	Enumeración	Combinatorio	Gran eficiencia en encontrar motivos en la levadura.	Van Helden *et al.* (1998) *J. Mol. Biol.*
OrthoMEME	Esperanza Máxima	Probabilístico	Busca motivos en secuencias ortólogas de dos especies.	Prakash *et al.* (2004) *Pacific Biocomputing*
PhyloCon	Consensus	Probabilístico	Toma en consideración tanto genes ortólogos como corregulados. Basado en huellas filogenéticas.	Wang y Stormo (2003) *BMC Bioinformatics*

Continuación en la siguiente página.

Tabla VIII – continúa de la página anterior

Algoritmo	Principio Operacional	Modelo	Características	Referencia
PhyloGibbs	Gibbs *sampling*	Probabilístico	Basado en Gibbs para encontrar motivos en secuencias ortólogas.	Siddharthan *et al.* (2005) *PLoS Comput. Biol.*
PHYLONET	Alineamiento	Probabilístico	Encuentra motivos dependientes en secuencias ortólogas.	Wang y Stormo (2003) *BMC Bioinformatics*
PhyloScan	Escaneo	Combinatorio	Encuentra motivos filogenéticamente dependientes en secuencias ortólogas.	Carmack *et al.* (2007) *Alg. Mol. Biol.*
PhyME	Esperanza Máxima	Probabilístico	Basado en EM y en huellas filogenéticas.	Sinha *et al.* (2004) *BMC Bioinformatics*
Projection	Hashing	Probabilístico	Requiere un motivo en común en todas las secuencias.	Buhler y Tompa (2002) *J. Compuy. Biol.*

Continuación en la siguiente página.

Tabla VIII – continúa de la página anterior

Algoritmo	Principio Operacional	Modelo	Características	Referencia
SMILE	Árbol de sufijos	Combinatorio	Gran número de combinaciones de motivos. Implementación es eficiente.	Marsan y Sagot (2000) *J. Compuy. Biol.*
Weeder	Enumeración	Combinatorio	Excelentes resultados con un número grande de secuencias, pero cortas. Ha sido utilizado en casos ficticios y reales de eucariotas.	Pavesi *et al.* (2001) *BMC Bioinformatics*
WINNOWER	Grafos	Combinatorio	Encuentra motivos en secuencias cortas. Costo computacional alto.	Pevzner y Sze (2000) *P. Mol. Biol.*

Continuación en la siguiente página.

Tabla VIII – continúa de la página anterior

Algoritmo	Principio Operacional	Modelo	Características	Referencia
YMF	Enumeración	Combinatorio	Modelo derivado de sitios de pegado conocidos en la levadura. Ha encontrado sitios de pegado nuevos.	Sinha y Tompa (2000) *P. Mol. Biol.*

Apéndice C

Documentación de Casos de Prueba

En este apéndice, se documentan los casos de prueba generados por Tompa *et al.* (2005), donde se muestran las características de los casos de prueba reales, genéricos y ficticios. Se define como:

- Caso de prueba real: Aquel que contiene regiones promotoras reales y sitios de pegado reales.
- Caso de prueba genérico: Aquel que contiene secuencias no promotoras reales con sitios de pegado reales insertados en su ubicación original.
- Caso de prueba ficticio: Aquel que contiene secuencias generadas computacionalmente con sitios de pegado reales insertados en su ubicación original.

C.1 Documentación técnica

C.1.1 Mosca

Tabla IX: Casos de prueba reales de la Mosca.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
dm01r	4	1500	7
dm02r	1	2000	5

Continuación en la siguiente página.

Tabla IX – continúa de la página anterior

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
dm03r	3	2000	9
dm04r	4	2000	9
dm05r	3	2500	14
dm06r	1	3000	7

Tabla X: Casos de prueba genéricos de la Mosca.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
dm01g	4	1500	7
dm02g	1	2000	5
dm03g	3	2000	9
dm04g	4	2000	9
dm05g	3	2500	14
dm06g	1	3000	7
dm07g	3	1500	0
dm08g	3	2000	0

Tabla XI: Casos de prueba ficticios de la Mosca.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
dm01m	4	1500	7

Continuación en la siguiente página.

Tabla XI – continúa de la página anterior

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
dm02m	1	2000	5
dm03m	3	2000	9
dm04m	4	2000	9
dm05m	3	2500	14
dm06m	1	3000	7
dm07m	3	1500	0
dm08m	3	2000	0

C.1.2 Humano

Tabla XII: Casos de prueba reales del Humano.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
hm01r	18	2000	16
hm02r	9	1000	11
hm03r	10	1500	15
hm04r	13	2000	11
hm05r	3	1000	11
hm06r	8	500	9
hm07r	5	1000	6
hm08r	14	500	13
hm09r	10	1500	10

Continuación en la siguiente página.

Tabla XII – continúa de la página anterior

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
hm10r	6	500	11
hm11r	8	1000	19
hm12r	2	500	5
hm13r	6	1000	9
hm14r	2	1000	4
hm15r	4	2000	4
hm16r	7	3000	7
hm17r	11	500	10
hm18r	5	3000	7
hm19r	5	500	4
hm21r	5	1000	7
hm22r	6	500	5
hm23r	4	500	5
hm24r	8	500	8
hm25r	2	500	5
hm26r	9	1000	10

Tabla XIII: Casos de prueba genéricos del Humano.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
hm01g	18	2000	16
hm02g	9	1000	11

Continuación en la siguiente página.

Tabla XIII – continúa de la página anterior

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
hm03g	10	1500	15
hm04g	13	2000	11
hm05g	3	1000	11
hm06g	8	500	9
hm07g	5	1000	6
hm08g	14	500	13
hm09g	10	1500	10
hm10g	6	500	11
hm11g	8	1000	19
hm12g	2	500	5
hm13g	6	1000	9
hm14g	2	1000	4
hm15g	4	2000	4
hm16g	7	3000	7
hm17g	11	500	10
hm18g	5	3000	7
hm19g	5	500	4
hm21g	5	1000	7
hm22g	6	500	5
hm23g	4	500	5
hm24g	8	500	8

Continuación en la siguiente página.

Tabla XIII – continúa de la página anterior

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
hm25g	2	500	5
hm26g	9	1000	10

Tabla XIV: Casos de prueba ficticios del Humano.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
hm01m	18	2000	16
hm02m	9	1000	11
hm03m	10	1500	15
hm04m	13	2000	11
hm05m	3	1000	11
hm06m	8	500	9
hm07m	5	1000	6
hm08m	14	500	13
hm09m	10	1500	10
hm10m	6	500	11
hm11m	8	1000	19
hm12m	2	500	5
hm13m	6	1000	9
hm14m	2	1000	4
hm15m	4	2000	4
hm16m	7	3000	7

Continuación en la siguiente página.

Tabla XIV – continúa de la página anterior

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
hm17m	11	500	10
hm18m	5	3000	7
hm19m	5	500	4
hm21m	5	1000	7
hm22m	6	500	5
hm23m	4	500	5
hm24m	8	500	8
hm25m	2	500	5
hm26m	9	1000	10

C.1.3 Ratón

Tabla XV: Casos de prueba reales del Ratón.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
mus01r	3	500	6
mus02r	9	1000	12
mus03r	5	500	9
mus04r	7	1000	14
mus05r	4	500	6
mus06r	3	500	5
mus07r	4	1500	4

Continuación en la siguiente página.

Tabla XV – continúa de la página anterior

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
mus08r	3	1500	3
mus09r	2	500	2
mus10r	13	1000	15
mus11r	12	500	15
mus12r	3	500	7

Tabla XVI: Casos de prueba genéricos del Ratón.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
mus01g	3	500	6
mus02g	9	1000	12
mus03g	5	500	9
mus04g	7	1000	14
mus05g	4	500	6
mus06g	3	500	5
mus07g	4	1500	4
mus08g	3	1500	3
mus09g	2	500	2
mus10g	13	1000	15
mus11g	12	500	15
mus12g	3	500	7

Tabla XVII: Casos de prueba ficticios del Ratón.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
mus01m	3	500	6
mus02m	9	1000	12
mus03m	5	500	9
mus04m	7	1000	14
mus05m	4	500	6
mus06m	3	500	5
mus07m	4	1500	4
mus08m	3	1500	3
mus09m	2	500	2
mus10m	13	1000	15
mus11m	12	500	15
mus12m	3	500	7

C.1.4 Levadura

Tabla XVIII: Casos de prueba reales de la Levadura.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
yst01r	9	1000	7
yst02r	4	500	5
yst03r	8	500	18

Continuación en la siguiente página.

Tabla XVIII – continúa de la página anterior

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
yst04r	7	1000	7
yst05r	3	500	4
yst06r	7	500	7
yst08r	11	1000	14
yst09r	16	1000	13

Tabla XIX: Casos de prueba genéricos de la Levadura.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
yst01g	9	1000	7
yst02g	4	500	5
yst03g	8	500	18
yst04g	7	1000	7
yst05g	3	500	4
yst06g	7	500	7
yst07g	6	500	0
yst08g	11	1000	14
yst09g	16	1000	13
yst10m	5	1000	0

Tabla XX: Casos de prueba ficticios de la Levadura.

Caso de prueba	No de Secuencias	Largo	Sitios de Pegado
yst01m	9	1000	7
yst02m	4	500	5
yst03m	8	500	18
yst04m	7	1000	7
yst05m	3	500	4
yst06m	7	500	7
yst07m	6	500	0
yst08m	11	1000	14
yst09m	16	1000	13
yst10m	5	1000	0

C.2 Documentación bioinformática

Para los casos de prueba reales, también se documentó la ubicación de la región promotora (secuencia) dentro de la base de datos GenBank (Benson *et al.*, 2008), el nombre del gen que regula, la ubicación del gen que regula y el número de identidad de GenBank como referencia.

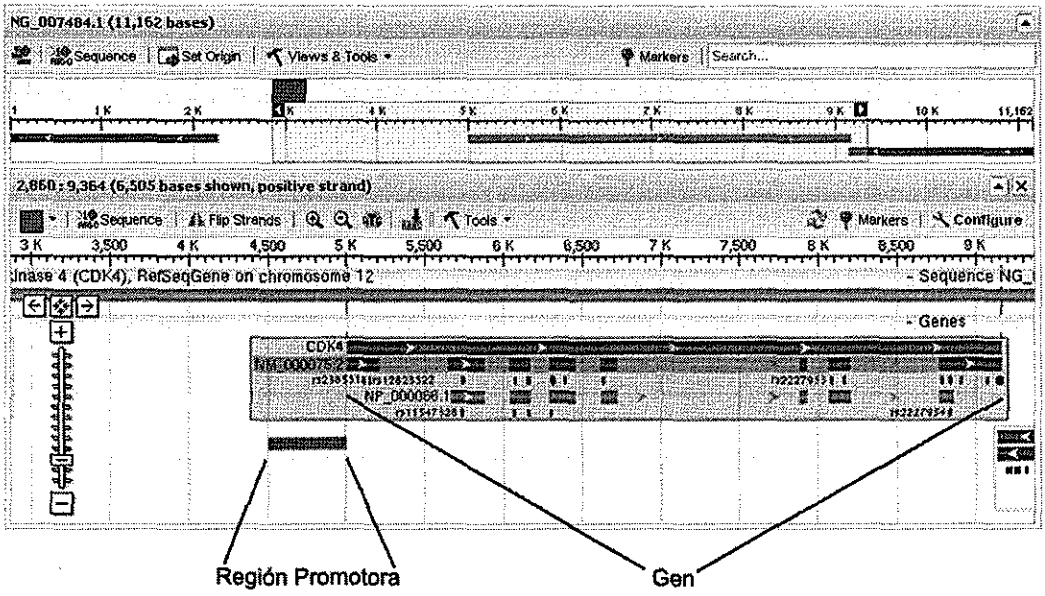


Figura 51. Ejemplo gráfico de una secuencia promotora, donde se muestra la ubicación de la región promotora y del gen que regula.

Como se muestra en la Figura 51, la secuencia 0 del caso de prueba hm06r, se encuentra en GenBank bajo el registro NG_007484.1, su región promotora está ubicada entre los nucleótidos 4501-5000, la cual regula el gen $CDK4$, ubicado entre los nucleótidos 5001-9162. En caso que dentro de la secuencia de GenBank, no se encuentre el gen, es decir, se trate únicamente de la región promotora, se encontrará un ND (No Disponible) dentro del campo "Ubicación del Gen".

Tabla XXI: *Documentación de las regiones promotoras de los casos de prueba reales.*

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
hm01	0	NG_004113.1	43905-45904	CCL4L1	45908-47715
	1	AF111167.2	21814-23813	CFOS	23969-26410
	2	AL136985.11	1314-3197	C_JUN	2780-3198
	3	NG_011740.1	3073-5072	MMP1	5001-13326
	4	NG_007161.1	3189-5188	MYC	5000-10365
	5	NG_016798.1	3001-5000	POLA1	5001-308040
	6	NG_016196.1	3131-5130	EDN1	5001-11899
	7	NG_008401.1	3047-5046	GFAP	5001-14922
	8	NG_015840.1	3019-5018	IFNG	5001-9972
	9	NG_016148.1	3003-5002	NPY	5001-12678
	10	M22469.1	84-2083	PLANH2	2082-2464
	11	NM_032223.2	4478-6477	PCNXL3	1..6584

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	12	AC023464.11	146774-148773	*preproenkephalin A*	148406-153461
	13	NG_012306.1	3020-5019	IRF5	5001-17096
	14	AL023883.6	130996-132995	PRL	132480-142730
	15	NG_012100.1	3003-5002	MMP3	5001-12815
	16	NG_012533.1	3115-5114	TIMP1	5001-9501
	17	NG_011904.1	2450-4449	PLAU	5001-11398
hm02	0	DQ989182.1	511-1510	CEACAM2	1470-22653
	1	AC096649.1	8094-8993	ATF2	7874-7993
	2	AY338491.1	112-1111	CCNB1	1112-11721
	3	NG_008624.1	4001-5000	KRT14	5001-9617
	4	NG_011676.1	4001-5001	GH1	5001-6636
	5	NG_015840.1	4019-5018	IFNG	5001-9972
	6	NG_021397.1	3994-4993	MIP	5001-10150
	7	DQ370420.1	934-1933	MT2A	1934-2818

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	8	J00122.1	1-825	PENK	823-963
hm03	0	AY948115.1	322-1921	ADH1A	1822-16441
	1	NG_011435.1	3515-5014	ADH1B	5001-19732
	2	NG_011718.1	3515-5015	ADH1C	5001-21233
	3	NG_009291.1	3535-5034	ALB	5001-22158
	4	NG_011793.1	3501-5000	APOB	5001-47645
	5	NG_009557.1	4186-5002	C3	5001-47817
	6	NG_012651.1	3519-5018	HP	5001-11448
	7	X15399.1	1-511	Hemoxopin	ND
	8	NG_008852.1	3501-5000	INSR	5001-186746
	9	NG_013080.1	3759-5258	TF	5001-37874
hm04	0	NG_012926.1	3484-5005	NPPA	5001-7075
	1	X59744.1	1-690	C-JUN	ND
	2	NG_011740.1	3073-5072	MMP1	5001-13326

Continuación en la siguiente página.

Tabla XXI – *continúa de la página anterior*

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	3	NG_016196.1	3292-5130	EDN1	5001-11899
	4	NG_012196.1	2999-4834	FN1	5001-80615
	5	DQ370420.1	1..1933	MT2A	1934-2818
	6	AC023464.11	146774-148773	preproenkephalin A	148406-153461
	7	NG_007462.1	3001-5000	TNF	5001-7763
	8	AC009311.3	94399-96398	TNFAIP6	96399-118854
	9	AL162734.15	30943-32876	YY1AP1	2001-30931
	10	AL390122.16	32483-34117	FBXO28	34118-82054
	11	NG_011904.1	3074-4949	PLAU	5001-11398
	12	AL133356.10	6642-8641	VIP	8642-17609
hm05	0	AF512554.1	1995-2994	CDC2	3008-16817
	1	NG_012330.1	3999-4998	MYB	5001-42859
	2	X75755.1	62-1061	HPR4	1108-4233
hm06	0	NG_007484.1	4501-5000	CDK4	5001-9162

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	1	NG_011587.1	4508-5007	CXCR4	5001-8807
	2	NM_001130679.1	1006-1505	EIF4E	1524-2270
	3	NG_007503.1	16504-16854	ERBB2	16862-45523
	4	NG_012105.1	4501-5000	ODC1	5001-12946
	5	NG_009492.1	4505-5004	IGF1R	5001-319999
	6	AC109809.3	53832-54291	MER91A	54297-63267
	7	AL365502.57	12239-12739	NELF	12238-2271
hm07	0	NG_009065.1	4051-5050	CYBB	5001-38445
	1	NG_016798.1	4001-5001	POLA1	5001-308040
	2	AL450320.4	38555-39554	B3GAT2	37964-39163
	3	NG_000007.3	46759-47758	HBG1	47759-49344
	4	DQ451402.1	1085-1941	HSPA1A	1942-4324
hm08	0	NG_016421.1	4521-5020	ADRB2	5001-7042
	1	NG_009097.1	28283-28782	CNGA3	28785-57447

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	2	AY212879.1	654-1037	FOS	1038-4420
	3	NG_009111.1	51263-51762	CCNA2	51555-53008
	4	NG_008645.1	4501-5000	DBH	5001-27982
	5	NG_012196.1	4499-4998	FN1	5001-80615
	6	AL138827.16	56093-56592	CGA	46484-56092
	7	NG_007114.1	4626-5000	INS	4986-6416
	8	NG_008962.1	4542-5041	PTH	5001-8967
	9	NG_012890.1	4647-5147	RPL10	5001-9110
	10	NG_023264.1	4501-5000	PLAT	5001-37959
	11	AL133356.10	8142-8641	VIP	8642-17609
	12	AL132768.15	72857-73356	C20ORF172	50905-72928
	13	NM_016206.2	8349-8848	VGLL3	8951-10315
hm09	0	AF111167.2	22314-23386	unknown	23388-26410
	1	AC096649.1	8094-9493	ATF2	7874-7993

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	2	J04201.1	1-1283	POLB	1285-1728
	3	NG_012196.1	3499-4998	FN1	5001-80615
	4	AF109152.1	204-1637	GHP	1637-1686
	5	X00187.1	135-1013	preproenkephalin A	ND
	6	AC068889.35	101-1353	BAC	ND
	7	NG_027721.1	4693-6192	TGFB2	5001-104286
	8	NG_023264.1	3501-5000	PLAT	5001-37959
	9	AL133356.10	7142-8641	VIP	8642-17609
hm10	0	AL133539.14	40597-41096	RAB32	41098-52370
	1	AJ009560.1	1035-1534	CDC6	ND
	2	NG_007161.1	4689-5188	MYC	5000-10365
	3	AL021154.1	80987-81486	E2F2	56195-80985
	4	AL136172.16	5581-6080	RBL1	5346-5568
	5	M15205.1	1-456	TK	451-13417

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
hm11	0	NG_000007.3	26473-27472	HBE	27473-29266
	1	AL355305.9	58019-58899	RP11	57620-57940
	2	NG_000007.3	26473-27472	HBE1	27473-29266
	3	NG_008331.1	4001-5000	ITGA2B	5001-22324
	4	NG_007483.2	4001-5001	GYPB	5001-28240
	5	AY040820.1	401-1016	SIGLEC5	1017-19569
	6	NG_007479.1	4001-5000	GYPC	5001-45563
	7	NG_000006.1	12777-13717	HBZ	13718-15368
hm12	0	AL138827.16	56592-56093	CGA	46484-56092
	1	NG_011640.1	4555-5054	IL6	5001-9856
hm13	0	NG_009291.1	4035-5034	ALB	5001-22158
	1	NG_012043.1	4001-5000	APOA2	5001-6336
	2	NG_013007.1	4016-5015	CRP	5001-7301
	3	NG_008331.1	4001-5000	ITGA2B	5001-22324

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	4	AC091524.4	6445-7444	IGFBP1	7445-12617
	5	NG_017043.1	4042-5001	SI	5001-104598
hm14	0	NG_009291.1	4035-5034	ALB	5001-22158
	1	NG_012043.1	4001-5000	APOA2	5001-6336
hm15	0	NG_012196.1	2999-4834	FN1	5001-80615
	1	NG_008401.1	3047-5046	GFAP	5001-14922
	2	AL138827.16	56093-57526	CGA	46484-56092
	3	NG_011676.1	3001-5001	GH1	5001-6636
hm16	0	AC109583.2	80348-83347	RP11	ND
	1	NG_012083.1	2262-5262	ICAM1	5001-20775
	2	NG_008851.1	2001-5000	IL1B	5001-12020
	3	NG_011640.1	2055-5054	IL6	5001-9856
	4	NG_015990.1	2001-5000	CCL5	5001-13883
	5	M26698.1	1-1138	SAA	1123-1157

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	6	AC009311.3	93399-96398	TNFAIP6	96399-118854
hm17	0	U03018.1	502-1001	MGSA alpha	1081-1180
	1	U03019.1	547-1046	MGSA beta	1122-1221
	2	U03020.1	506-1005	MGSA gamma	1083-1182
	3	NG_012083.1	4763-5262	ICAM1	5001-20775
	4	EF064725.1	1607-2046	IFNB1	1972-2811
	5	AB061825.1	1053-1552	RPL18	1167-5033
	6	NG_007403.1	4501-5000	IL2RA	4940-56616
	7	NG_011640.1	4555-5054	IL6	5001-9856
	8	M28130.1	983-1481	IL8	1482-4236
	9	NG_012124.1	4542-5041	SELE	5001-16440
	10	AF129756.1	177730-178229	LYMPHOTOXIN alpha	175726-177647
hm18	0	Z95114.19	145427-151012	APOL3	126037-145264
	1	NG_007161.1	2189-5188	MYC	5000-10365

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	2	NG_000007.3	44759-47758	HBG1	47759-49344
	3	NG_000827.2	5302-8302	H2B	4822-5302
	4	NG_016779.1	1771-4770	*IL2*	5001-10026
hm19	0	NG_012196.1	4499-4834	FN1	5001-80615
	1	X15723.1	4161-4659	FUR	4660-12253
	2	HM151000.1	815-1314	Gastrin	1348-1374
	3	AL162727.20	33401-33900	RGS3	33901-50979
	4	NG_011676.1	4501-5001	GH1	5001-6636
hm21	0	AL118506.27	1489-2488	C20orf135	2161-4254
	1	NG_007553.1	4981-5918	ACTC1	5897-12631
	2	NG_006672.1	4003-5002	ACTA1	5001-7852
	3	AF111167.2	22814-23813	CFOS	23969-26410
	4	NG_007555.1	4547-5042	MYL3	5001..10617
hm22	0	DQ989182.1	1001-1510	CEACAM2	1470-22653

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	1	Z21818.1	10190-10550	carcinoembryonic antigen	10694-11288
	2	NG_012022.1	4633-5110	LOC80054	2468-4668
	3	AY338491.1	612-1111	CCNB1	1112-11721
	4	NG_023030.1	4528-5027	HMOX1	5001-18148
	5	NG_023187.1	4545-5044	HLA-B	5001-8341
hm23	0	NG_008645.1	4501-5000	DBH	5001-27982
	1	NG_000007.3	26973-27472	HBE1	27473-29266
	2	AF373868.2	1453-1952	CSF2	1977-4326
	3	NG_015840.1	4519-5018	IFNG	5001-9972
hm24	0	AF527417.1	111-610	CDC25A	611-30465
	1	DQ300360.1	1114-1613	CYC1	1614-4042
	2	AY326400.1	1565-2064	ENSA	2187-2596
	3	NG_007375.1	4501-5000	CCND1	5001-18370
	4	AL355310.20	9232-9731	GNAI3	9664-55406

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	5	AF516106.1	1409-1908	E2F1	1909-12613
	6	NG_000009.2	767-1266	HIST1H2BO	561-1027
	7	NG_009009.1	4529-5028	RB1	5001-183144
hm25	0	AL138827.16	56593-56592	CGA	46484-56092
	1	NG_011640.1	4555-5054	IL6	5001-9856
hm26	0	DQ989182.1	511-1510	CEACAM2	1470-22653
	1	NG_008927.1	4049-5048	CRELD2	1..4824
	2	AC096649.1	8094-8993	ATF2	7874-7993
	3	AY338491.1	112-1111	CCNB1	1112-11721
	4	NG_011676.1	4001-5001	GH1	5001-6636
	5	NG_015840.1	4019-5018	IFNG	5001-9972
	6	NG_021397.1	3994-4993	MIP	5001-10150
	7	DQ370420.1	934-1933	MT2A	1934-2818
	8	AC023464.11	147774-148773	preproenkephalin A	148406-153461

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
dm01	0	M17837.1	3859-5358	ADH	5359-6264
	1	BT050430.1	1558-2377	CG15335-RB	2380-2467
	2	M14497.1	1549-2864	ANTP	3000-3964
	3	NM_078876.4	1-1048	DDC	1049-2476
dm02	0	U32088.1	1-1824	EVE	1825-2071
dm03	0	S59100.1	1-690	TWI	ND
	1	NM_137166.4	1-505	PCS	506-1939
	2	AE001572.2	300678-302577	zerknullt 1	302627-303975
dm04	0	AE013599.4	3629372-3631371	CG30497	3625273-3629683
	1	NM_165841.1	22-540	EN	541-2199
	2	X78903.1	1-662	even-skipped	ND
	3	X13331.1	1-1829	Knirps	1846-4590
dm05	0	NM_141274.2	1-1924.	HD	ND
	1	U31961.1	241584-243834	UBX	243835-318526

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	2	AL133506.2	38313-40237	BACN33B1.1	32733-38312
dm06	0	AE014297.2	14059219-14062218	Repo	14062219-14064472
mus01	0	AL596122.14	160701-161200	CCL3	159187-160698
	1	AL683828.8	57684-58183	ORM1	58178-61781
	2	AL592185.35	128999-129498	NOS2	129427-168894
mus02	0	M20497.1	1-890	AFABP	869-966
	1	AF033102.1	1278-2284	EKLF	2326-5297
	2	AF195956.1	3754-4514	SINE	4562-9749
	3	AL645741.15	5715-6714	IL4	5524-5718
	4	AL645741.16	106843-107842	IL5	107843-112155
	5	X61655.1	1-624	MYOD1	654-2539
	6	X16009.1	166-1165	MRP	1170-1200
	7	J05511.1	228-860	SCC	841-885
	8	AL671885.13	110843-111842	TIMP1	111838-115537

Continuación en la siguiente página.

Tabla XXI – *continúa de la página anterior*

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
mus03	0	M62362.1	13-515	EBP	516-1703
	1	AL596122.14	160701-161200	CCL3	159187-160698
	2	AL683828.8	57684-58183	ORM1	58178-61781
	3	M29660.1	419-812	3T3	786-955
	4	M21280.1	286-717	SDS	710-885
mus04	0	AL683828.8	57684-58183	ORM1	58178-61781
	1	J05246.1	1-859	AFP	860-900
	2	J04738.1	1228-2045	HH3	2041-2376
	3	M20497.1	1-890	AFABP	869-966
	4	AC153910.6	110897-111501	RP23	111505-200166
	5	X03480.1	1-479	SAA	ND
	6	M21280.1	1-717	SDS	710-885
mus05	0	AF195956.1	4015-4514	SINE	4562-9749
	1	AL645741.15	5715-6214	IL4	5524-5718

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	2	NM_010592.4	915-1227	JUND	1146-2345
	3	X16009.1	666-1165	MRP	1170-1200
mus06	0	AF033102.1	1778-2284	EKLF	2326-5297
	1	M38133.1	1154-1628	EPO	1496-1775
	2	M76557.1	3-163	MC-CPA	170-172
mus07	0	X13959.1	1-1132	AChR	1133-1499
	1	M22381.1	1543-3042	AChR	3058-3157
	2	AL596096.7	34594-36093	Alox15	27121-34583
	3	AF188002.1	1850-3350	MCK	3351-3357
mus08	0	AL928918.8	24987-26464	St8sia6	24714-24810
	1	AL592185.35	127999-129498	NOS2	129427-168894
	2	AF109719.2	90135-91634	LTA	91875-92696
mus09	0	AF195956.1	4015-4514	SINE	4562-9749
	1	NM_010592.4	915-1227	JUND	1146-2345

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
mus10	0	AC151999.5	193927-194411	BAC	ND
	1	AC137525.3	46831-47755	RP23	ND
	2	AF079309.1	101-1101	Hdac3	1046-3102
	3	L10295.1	78-861	Rep-3	976-1324
	4	M38133.1	655-1628	EPO	1496-1775
	5	M29660.1	1-812	3T3	786-955
	6	X53530.1	405-1483	Metallothionein-I	ND
	7	X61655.1	1-624	MYOD1	654-2539
	8	M32612.1	600-981	NCAM-C	947-1195
	9	AC149606.6	162595-163595	RP23	ND
	10	AB064489.1	2516-3519	MP41	3519-3525
	11	J05605.1	233-1100	THBS-1	1145-3466
	12	M61011.1	8-915	alpha-1	ND
mus11	0	X54876.1	1148-1645	COL1A1	1628-3980

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	1	M62362.1	13-515	EBP	516-1703
	2	AL591490.12	125169-125546	ADA	102429-124960
	3	L10295.1	78-861	Rep-3	976-1324
	4	AC159631.2	23726-24225	RP24	ND
	5	X60961.1	993-1492	EX1	1493-1667
	6	M38133.1	1154-1628	EPO	1496-1775
	7	NM_010592.4	915-1227	JUND	1146-2345
	8	S59359.1	1-270	MB-1	280-331
	9	X15128.1	86-584	MRE	596-663
	10	M35824.1	465-965	PI	964-980
	11	AL671917.8	85617-86116	Gpsm2	86117-117021
mus12	0	AF195956.1	4015-4514	SINE	4562-9749
	1	AL645741.15	6214-6714	IL4	5524-5718
	2	AL645741.16	107343-107842	IL5	107843-112155

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
yst01	0	Z28113.1	1115-2114	RAD27	272-1420
	1	X59720.2	16881-17886	CHA1	15798-16880
	2	Z46260.1	8293-9292	Delta	7216-8292
	3	U00027.2	28115-28988	ENO2	29114-30427
	4	Z71781.1	35633-36632	MPS1	33338-35632
	5	Z38059.1	2474-3473	POT1	1220-2473
	6	U12980.3	68539-69061	CDC19	69196-70698
	7	Z75025.1	1-1000	YTA1	884-2188
	8	U32445.1	18398-19397	RPC40	17390-18397
yst02	0	U00062.1	13126-13625	PUT2	13626-15353
	1	Z73253.1	462-940	GAL2	962-2686
	2	X81324.1	4773-5272	GAL7	3672-4772
	3	Z46729.1	20069-20568	GAL80	20569-20951
yst03	0	Z49212.1	2660-3159	ADE4	1127-2659

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	1	Z75107.1	1381-1880	PET56	405-1380
	2	X59720.2	68349-68833	HIS4	65934-68333
	3	Z73564.1	1-500	ORF	249-2000
	4	X72833.1	136-635	YPT31	304-975
	5	X01466.1	278-777	ILV1	778-2508
	6	Z49702.1	12971-13268	ILV2	13269-15332
	7	X04273.1	169-668	TRP4	669-1811
yst04	0	FJ415226.1	8-668	PGK1	ND
	1	Z49209.1	8764-9642	TPI1	7896-8642
	2	Z48613.1	18813-19812	PLB1	19813-21030
	3	Z74828.1	1958-2957	ADH1	911-1957
	4	X99228.1	2552-3551	ENO1	3552-4865
	5	U00027.2	28115-28988	ENO2	29114-30427
	6	U12980.3	68539-69061	CDC19	69196-70698

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
yst05	0	X03010.1	35-534	AA1-431	535-1830
	1	X74151.1	33614-34113	ORF15	32201-33613
	2	Z28209.1	4059-4449	STE6	186-4058
yst06	0	Z73543.1	498-997	MFalpha1	998-1495
	1	Z72611.1	570-1069	MFalpha2	207-569
	2	X03010.1	35-534	AA1-431	535-1830
	3	X74151.1	33614-34113	ORF15	32201-33613
	4	Z72675.1	928-1427	ORF16	596-1273
	5	Z28209.1	4059-4449	STE6	186-4058
	6	Z73514.1	2646-3145	ORF YPL158c	366-2642
yst08	0	X99228.1	2552-3551	ENO1	3552-4865
	1	U00027.2	28115-28988	ENO2	29114-30427
	2	Z73217.1	418-1417	PDC1	1-417
	3	U12980.3	68539-69061	CDC19	69196-70698

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	4	NM_001181461.1	1-417	YJL027C	ND
	5	Z46659.1	22994-23993	RP51A	23994-24802
	6	U51033.2	15411-16410	TEF1	16411-17787
	7	Z35985.1	2745-3582	TKL2	466-2511
	8	Z46861.1	26761-27760	REG1	25510-26760
	9	X59720.2	136740-137629	ADP1	133718-136867
	10	Z49209.1	8764-9642	TPI1	7896-8642
yst09	0	U10399.1	926-1925	DED81	1925-3081
	1	U33057.1	33336-34335	YDR512C	32772-33335
	2	U43503.2	29087-30086	tM(CAU)P	28991-29062
	3	Z73217.1	418-1417	PDC1	1-417
	4	Z70202.1	7092-8091	CTA1	5544-7091
	5	Z72873.1	2219-3218	CTT1	3219-4940
	6	Z46729.1	16284-17283	CYB2	14508-16283

Continuación en la siguiente página.

Tabla XXI – continúa de la página anterior

Caso	Secuencia	ID GenBank	Ubicación	Gen	Ubicación del Gen
	7	X99228.1	2552-3551	ENO1	3552-4865
	8	X95644.1	10127-11126	PHO2	8447-10126
	9	Z49509.1	1236-2235	TDH2	237-1235
	10	M22040.1	18-1016	HSF1	1017-3518
	11	Z49702.1	12269-13268	ILV2	13269-15332
	12	Z73051.1	1-795	ORF	598-2703
	13	U12980.3	68539-69061	CDC19	69196-70698
	14	L22015.2	7879-8878	SSA1	5950-7878
	15	Z74747.1	262-1000	RPB11	426-788

Apéndice D

Tablas de Sensibilidad, Especificidad y Rendimiento

A continuación, se presentan las tablas de los resultados de los algoritmos en cuanto a su sensibilidad, especificidad y rendimiento; donde BP: BioProspector, RM+BP: RepeatMasker + BioProspector, MM: MEME, RM+MM: RepeatMasker + MEME, WD: Weeder, RM+WD: RepeatMasker + Weeder, RM+BP+MM: RepeatMasker + BioProspector + MEME, RM+BP+WD+MM: RepeatMasker + BioProspector + Weeder + MEME.

Tabla XXII: Sensibilidad.

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
dm01r	0.0000	0.0000	0.7143	0.7143	0.4286	0.4286	0.0000	0.1429
dm02r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dm03r	0.0000	0.1111	0.0000	0.0000	0.1111	0.1111	0.2222	0.2222
dm04r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dm05r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dm06r	0.0000	0.0000	0.0000	0.0000	0.4286	0.4286	0.0000	0.1429
hm01r	0.0000	0.0000	0.0000	0.0000	0.0625	0.0625	0.0625	0.0625
hm02r	0.0909	0.1818	0.0000	0.0000	0.0000	0.1818	0.4545	0.3636
hm03r	0.0667	0.1333	0.2000	0.2667	0.0000	0.0667	0.4000	0.2667
hm04r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
hm05r	0.0909	0.0909	0.0909	0.1818	0.3636	0.3636	0.3636	0.3636
hm06r	0.6667	0.6667	0.0000	0.0000	0.6667	0.6667	0.7778	0.6667
hm07r	0.0000	0.0000	0.1667	0.1667	0.0000	0.0000	0.1667	0.1667

Continuación en la siguiente página.

Tabla XXII – continúa de la página anterior

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
hm08r	0.8462	0.8462	0.0000	0.0000	0.6923	0.6923	0.9231	0.9231
hm09r	0.6000	0.6000	0.0000	0.1000	0.9000	0.9000	0.4000	0.5000
hm10r	0.0000	0.0909	0.0000	0.0000	0.5455	0.5455	0.4545	0.3636
hm11r	0.2632	0.2632	0.0526	0.0526	0.0526	0.1053	0.2632	0.2632
hm12r	0.0000	0.0000	0.0000	0.0000	0.2000	0.2000	0.6000	0.4000
hm13r	0.1111	0.1111	0.0000	0.0000	0.0000	0.0000	0.2222	0.2222
hm14r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2500	0.2500
hm15r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2500	0.2500
hm16r	0.0000	0.2857	0.0000	0.1429	0.0000	0.1429	0.2857	0.2857
hm17r	0.9000	0.9000	0.7000	0.7000	0.9000	0.9000	0.8000	0.8000
hm18r	0.0000	0.0000	0.1429	0.1429	0.1429	0.1429	0.1429	0.2857
hm19r	0.5000	0.5000	0.5000	0.5000	0.0000	0.0000	1.0000	0.7500
hm21r	0.1429	0.1429	0.4286	0.4286	0.0000	0.2857	0.8571	1.0000
hm22r	0.4000	0.4000	0.0000	0.0000	0.4000	0.4000	0.4000	0.4000

Continuación en la siguiente página.

Tabla XXII – continúa de la página anterior

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
hm23r	0.4000	0.4000	0.0000	0.2000	0.2000	0.2000	0.6000	0.6000
hm24r	0.1250	0.1250	0.3750	0.3750	0.6250	0.6250	0.1250	0.2500
hm25r	0.2000	0.4000	0.4000	0.4000	0.2000	0.2000	0.6000	0.6000
hm26r	0.2000	0.1000	0.0000	0.0000	0.0000	0.0000	0.3000	0.2000
mus01r	0.3333	0.3333	0.1667	0.1667	0.5000	0.5000	0.3333	0.3333
mus02r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0833	0.0833
mus03r	0.1111	0.2222	0.1111	0.2222	0.0000	0.0000	0.3333	0.3333
mus04r	0.2857	0.2857	0.0714	0.0714	0.0714	0.0714	0.3571	0.2857
mus05r	0.1667	0.1667	0.1667	0.1667	0.1667	0.3333	0.1667	0.1667
mus06r	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	0.0000	0.2000
mus07r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
mus08r	0.0000	0.0000	0.0000	0.0000	0.6667	0.6667	0.0000	0.0000
mus09r	0.5000	0.5000	0.5000	0.5000	1.0000	1.0000	1.0000	1.0000
mus10r	0.4000	0.4000	0.2000	0.2000	0.0000	0.0000	0.4000	0.2667

Continuación en la siguiente página.

Tabla XXII – continúa de la página anterior

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
mus11r	0.6667	0.6667	0.2000	0.2000	0.4667	0.4667	0.6667	0.6000
mus12r	0.0000	0.0000	0.1429	0.1429	0.0000	0.0000	0.0000	0.0000
yst01r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
yst02r	0.8000	0.8000	0.8000	0.8000	0.6000	0.8000	1.0000	1.0000
yst03r	0.2222	0.4444	0.0556	0.0556	0.1111	0.2222	0.2222	0.2778
yst04r	0.2857	0.5714	0.1429	0.2857	0.0000	0.0000	0.5714	0.4286
yst05r	0.0000	0.0000	0.7500	0.7500	0.5000	0.5000	0.0000	0.0000
yst06r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
yst08r	0.5714	0.5714	0.6429	0.6429	0.0000	0.0000	0.6429	0.5714
Promedio	0.1989	0.2262	0.1544	0.1715	0.2400	0.2642	0.3340	0.3258

Tabla XXIII: Especificidad.

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
dm01r	0.0000	0.0000	0.2000	0.1923	0.0118	0.0122	0.0000	0.0139
dm02r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dm03r	0.0000	0.0455	0.0000	0.0000	0.0208	0.0217	0.0400	0.0377
dm04r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dm05r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dm06r	0.0000	0.0000	0.0000	0.0000	0.1579	0.1667	0.0000	0.0909
hm01r	0.0000	0.0000	0.0000	0.0000	0.0114	0.0114	0.0081	0.0071
hm02r	0.0161	0.0323	0.0000	0.0000	0.0000	0.0107	0.0704	0.0563
hm03r	0.0145	0.0299	0.0833	0.0952	0.0000	0.0204	0.0690	0.0513
hm04r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
hm05r	0.0385	0.0435	0.0625	0.1429	0.0784	0.0784	0.0769	0.0727
hm06r	0.1200	0.1111	0.0000	0.0000	0.0330	0.0339	0.0778	0.0706
hm07r	0.0000	0.0000	0.0435	0.0455	0.0000	0.0000	0.0111	0.0110

Continuación en la siguiente página.

Tabla XXIII – continúa de la página anterior

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
hm08r	0.1358	0.1358	0.0000	0.0000	0.0481	0.0497	0.1319	0.1319
hm09r	0.0870	0.1091	0.0000	0.0213	0.0600	0.0621	0.0345	0.0400
hm10r	0.0000	0.0256	0.0000	0.0000	0.1176	0.1200	0.1064	0.0769
hm11r	0.1000	0.1042	0.0294	0.0278	0.0370	0.0769	0.1000	0.0962
hm12r	0.0000	0.0000	0.0000	0.0000	0.0109	0.0110	0.0349	0.0230
hm13r	0.0270	0.0286	0.0000	0.0000	0.0000	0.0000	0.0274	0.0267
hm14r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0667	0.0526
hm15r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0149	0.0139
hm16r	0.0000	0.0417	0.0000	0.0286	0.0000	0.0125	0.0200	0.0215
hm17r	0.1406	0.1500	0.2000	0.2000	0.0692	0.0732	0.1569	0.1356
hm18r	0.0000	0.0000	0.0370	0.0435	0.0050	0.0047	0.0083	0.0163
hm19r	0.0606	0.0571	0.0833	0.0870	0.0000	0.0000	0.0870	0.0612
hm21r	0.0270	0.0270	0.1250	0.1364	0.0000	0.0339	0.0800	0.0875
hm22r	0.0455	0.0417	0.0000	0.0000	0.0116	0.0119	0.0435	0.0408

Continuación en la siguiente página.

Tabla XXIII – continúa de la página anterior

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
hm23r	0.0408	0.0465	0.0000	0.0500	0.0082	0.0086	0.0500	0.0429
hm24r	0.0217	0.0208	0.0833	0.0882	0.0641	0.0641	0.0118	0.0235
hm25r	0.0227	0.0476	0.1818	0.2000	0.0108	0.0110	0.0429	0.0395
hm26r	0.0333	0.0192	0.0000	0.0000	0.0000	0.0000	0.0326	0.0217
mus01r	0.0488	0.0513	0.0385	0.0400	0.0441	0.0455	0.0377	0.0351
mus02r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0127	0.0141
mus03r	0.0256	0.0571	0.0400	0.0870	0.0000	0.0047	0.0455	0.0441
mus04r	0.0889	0.0870	0.0333	0.0313	0.0128	0.0128	0.0781	0.0667
mus05r	0.0244	0.0238	0.0313	0.0303	0.0043	0.0043	0.0156	0.0154
mus06r	0.0000	0.0000	0.0000	0.0000	0.0862	0.0893	0.0000	0.0179
mus07r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
mus08r	0.0000	0.0000	0.0000	0.0000	0.0290	0.0299	0.0000	0.0000
mus09r	0.0400	0.0435	0.0357	0.0357	0.0465	0.0465	0.0488	0.0435
mus10r	0.0822	0.0896	0.0682	0.0652	0.0000	0.0000	0.0690	0.0500

Continuación en la siguiente página.

Tabla XXIII – continúa de la página anterior

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
mus11r	0.1333	0.1299	0.0638	0.0698	0.0391	0.0393	0.1031	0.0928
mus12r	0.0000	0.0000	0.0370	0.0455	0.0000	0.0000	0.0000	0.0000
yst01r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
yst02r	0.0784	0.0816	0.0909	0.0851	0.0390	0.0411	0.0758	0.0735
yst03r	0.0702	0.1356	0.0270	0.1081	0.0211	0.0396	0.0482	0.0568
yst04r	0.0435	0.0851	0.0313	0.0857	0.0000	0.0230	0.0625	0.0429
yst05r	0.0000	0.0000	0.1154	0.1250	0.0377	0.0377	0.0000	0.0000
yst06r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
yst08r	0.1176	0.1250	0.1765	0.2093	0.0000	0.0000	0.1098	0.0941
Promedio	0.0337	0.0405	0.0384	0.0420	0.0223	0.0262	0.0422	0.0402

Tabla XXIV: Rendimiento.

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
dm01r	0.0000	0.0000	0.1852	0.1786	0.0116	0.0120	0.0000	0.0128
dm02r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dm03r	0.0000	0.0333	0.0000	0.0000	0.0179	0.0185	0.0351	0.0333
dm04r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dm05r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dm06r	0.0000	0.0000	0.0000	0.0000	0.1304	0.1364	0.0000	0.0588
hm01r	0.0000	0.0000	0.0000	0.0000	0.0097	0.0097	0.0072	0.0064
hm02r	0.0139	0.0282	0.0000	0.0000	0.0000	0.0102	0.0649	0.0513
hm03r	0.0120	0.0250	0.0625	0.0755	0.0000	0.0159	0.0625	0.0449
hm04r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
hm05r	0.0278	0.0303	0.0385	0.0870	0.0690	0.0690	0.0678	0.0645
hm06r	0.1132	0.1053	0.0000	0.0000	0.0324	0.0333	0.0761	0.0682
hm07r	0.0000	0.0000	0.0357	0.0370	0.0000	0.0000	0.0105	0.0104

Continuación en la siguiente página.

Tabla XXIV – continúa de la página anterior

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
hm08r	0.1325	0.1325	0.0000	0.0000	0.0471	0.0486	0.1304	0.1304
hm09r	0.0822	0.1017	0.0000	0.0179	0.0596	0.0616	0.0328	0.0385
hm10r	0.0000	0.0204	0.0000	0.0000	0.1071	0.1091	0.0943	0.0678
hm11r	0.0781	0.0806	0.0192	0.0185	0.0222	0.0465	0.0781	0.0758
hm12r	0.0000	0.0000	0.0000	0.0000	0.0104	0.0105	0.0341	0.0222
hm13r	0.0222	0.0233	0.0000	0.0000	0.0000	0.0000	0.0250	0.0244
hm14r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0556	0.0455
hm15r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0143	0.0133
hm16r	0.0000	0.0377	0.0000	0.0244	0.0000	0.0116	0.0190	0.0204
hm17r	0.1385	0.1475	0.1842	0.1842	0.0687	0.0726	0.1509	0.1311
hm18r	0.0000	0.0000	0.0303	0.0345	0.0049	0.0046	0.0079	0.0156
hm19r	0.0571	0.0541	0.0769	0.0800	0.0000	0.0000	0.0870	0.0600
hm21r	0.0233	0.0233	0.1071	0.1154	0.0000	0.0313	0.0789	0.0875
hm22r	0.0426	0.0392	0.0000	0.0000	0.0114	0.0117	0.0408	0.0385

Continuación en la siguiente página.

Tabla XXIV – continúa de la página anterior

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
hm23r	0.0385	0.0435	0.0000	0.0417	0.0079	0.0083	0.0484	0.0417
hm24r	0.0189	0.0182	0.0732	0.0769	0.0617	0.0617	0.0109	0.0220
hm25r	0.0208	0.0444	0.1429	0.1538	0.0103	0.0105	0.0417	0.0385
hm26r	0.0294	0.0164	0.0000	0.0000	0.0000	0.0000	0.0303	0.0200
mus01r	0.0444	0.0465	0.0323	0.0333	0.0423	0.0435	0.0351	0.0328
mus02r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0111	0.0122
mus03r	0.0213	0.0476	0.0303	0.0667	0.0000	0.0045	0.0417	0.0405
mus04r	0.0727	0.0714	0.0233	0.0222	0.0110	0.0110	0.0685	0.0571
mus05r	0.0217	0.0213	0.0270	0.0263	0.0042	0.0042	0.0145	0.0143
mus06r	0.0000	0.0000	0.0000	0.0000	0.0862	0.0893	0.0000	0.0167
mus07r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
mus08r	0.0000	0.0000	0.0000	0.0000	0.0286	0.0294	0.0000	0.0000
mus09r	0.0385	0.0417	0.0345	0.0345	0.0465	0.0465	0.0488	0.0435
mus10r	0.0732	0.0789	0.0536	0.0517	0.0000	0.0000	0.0625	0.0440

Continuación en la siguiente página.

Tabla XXIV – continúa de la página anterior

Caso	BP	RM+BP	MM	RM+MM	WD	RM+WD	RM+BP+MM	RM+BP+WD+MM
mus11r	0.1250	0.1220	0.0508	0.0545	0.0374	0.0376	0.0980	0.0874
mus12r	0.0000	0.0000	0.0303	0.0357	0.0000	0.0000	0.0000	0.0000
yst01r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
yst02r	0.0769	0.0800	0.0889	0.0833	0.0380	0.0400	0.0758	0.0735
yst03r	0.0563	0.1159	0.0185	0.0784	0.0180	0.0348	0.0412	0.0495
yst04r	0.0392	0.0800	0.0263	0.0769	0.0000	0.0217	0.0597	0.0405
yst05r	0.0000	0.0000	0.1111	0.1200	0.0364	0.0364	0.0000	0.0000
yst06r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
yst08r	0.1081	0.1143	0.1607	0.1875	0.0000	0.0000	0.1034	0.0879
Promedio	0.0306	0.0365	0.0329	0.0357	0.0206	0.0239	0.0393	0.0369