Tesis de Licenciatura en Ciencias de la Computación

Departamento de Ciencias de la Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Argentina

Inferencia de Interacciones Proteína-Proteína basadas en Interacciones de Dominios

Directores
Dr. Mariano Levin - Dr. Martin Vazquez
Instituto de Investigaciones en Ingeniería Genética y Biología Molecular (INGEBI).
Facultad de Ciencias Exactas y Naturales.
Universidad de Buenos Aires.
Argentina
(mlevin@dna.uba.ar - mvazquez@dna.uba.ar)

Co-Directora
Lic. Irene Loiseau
Departamento de Computación.
Facultad de Ciencias Exactas y
Naturales.
Universidad de Buenos Aires.
Argentina
(irene@dc.uba.ar)

Tesistas
Hugo Iván Coustau - Diego Sebastián Villarino
Departamento de Computación.
Facultad de Ciencias Exactas y Naturales.

Universidad de Buenos Aires.
Argentina
(dvillari@dc.uba.ar LU 416/95 - hcoustau@dc.uba.ar LU230/95)

Resumen

Dentro de los avances en la Biología Molecular se halla la posibilidad de secuenciar genomas de distintos organismos. Esto permite a los biólogos enfocar sus investigaciones en cuestiones más complejas que el estudio de genes individuales. Uno de los nuevos desafíos es conocer cómo interactúan las proteínas entre sí para comprender cómo se desarrollan los procesos biológicos. Conocer las interacciones de proteínas de un organismo nos permite definir una "Red de Interacciones". Los biólogos trabajan sobre estas redes para, por ejemplo, desarrollar nuevos medicamentos interfiriendo en una interacción para impedir el desarrollo de una enfermedad. La importancia de conocer nuevas interacciones de proteínas y lo dificultoso de obtenerlas en forma experimental, ya sea por su costo como por su complejidad, motiva a los biólogos a buscar formas alternativas de determinarlas. Una de estas formas es inferir interacciones de proteínas en un determinado organismo en base a las interacciones ya conocidas en otro. Este método se basa en que algunas interacciones se mantienen entre proteínas de diferentes organismos. Por tratarse de inferencias, estas se deberán verificarse en el laboratorio en forma experimental, pero gracias a este procedimiento la búsqueda de interacciones esta guiada y permite reducir el número de experimentos a realizar.

Los términos “dominio conservado”, “dominio” o “motivo” se utilizan para definir una región de la secuencia de aminoácidos de una proteína que puede ser identificada en otra a pesar de la falta de similitud global entre ellas. Estas regiones son de especial interés, ya que pueden determinan una función de esa región de la proteína. Estos dominios se utilizan para definir familias de proteínas que comparten el mismo motivo y por lo tanto pueden cumplir la misma función. Una proteína puede contener uno o más dominios y/o repeticiones del mismo. Las interacciones entre proteínas suelen darse entre los dominios que la componen, y pueden estar involucrados más de un dominio al mismo tiempo. Este trabajo comenzó luego de reuniones mantenidas con el equipo de biólogos del INGEBI dedicados a la investigación del *Tripanosoma cruzi* (Parásito que causa el Mal de Chagas), quienes estudian interacciones proteína-proteína. Para apoyarlos en sus investigaciones se desarrolló una herramienta que permita inferir interacciones de proteínas a partir de las interacciones ya conocidas de otro organismo. La predicción de estas interacciones se realizará basándonos en interacciones de dominios.

Abstract

Advances in Molecular Biology allow to obtain the genome of many organisms. As a result, biologists can focus theirs investigations on the study of complete genomes instead of individual genes. Nowadays, one of the goals in biology is to understand how proteins interact between each other, in order to understand how biological processes occur. Knowing the interactions of an organism, we can define an “Interaction Network”. These networks can help scientists to develop new drugs. This is done by breaking up an interaction to restrict the evolution of a disease. The importance of finding new interactions related to the difficulties to obtain them and the high cost of the experiments give rise to find alternative ways to determinate them. One possibility is to infer new protein interactions based on known interactions of another organism. This method is based on the fact that the same interactions are preserved across different organisms. Inferences must be confirmed in the laboratory performing experiments, but thanks to this kind of methods, the search of new interactions is driven reducing the number of required experiments.

The terms “Conserved Domain”, “Domain” or “Motif” are used to define a region of the protein’s amino acid sequence that can be identified in another protein despite the lack of global similarity between them. These regions are very important because they may define a function on a specific region. Domains can be used to define protein families that share the same function. A protein may contain one or more domains, and a domain may be present more than once. Protein interactions occur between the domains of the proteins. This work begun after several meetings maintained with *Trypanosome cruzi* (Chagas disease parasite) biologist research team at INGEBI which are researching protein-protein interactions. In order to collaborate with their research, we developed an application to infer protein interactions based on already known interactions in another organism. New interactions will be inferred based on domain interactions.

Contenido

Capítulo 1 - Introducción a la biología computacional

En esta sección se verán algunos conceptos de biología molecular indispensables para la compresión de este trabajo. Estas nociones se explicaran de manera elemental reduciendo su complejidad sin entrar en demasiados detalles biológicos. Una explicación más profunda sobre estos temas puede encontrarse en [Setubal97], [Hunter] y en la bibliografía referenciada a lo largo del trabajo.

- ADN (ácido desoxirribonucleico)

  En cada célula de un organismo se encuentra una molécula de ADN que esta constituida por una cadena de moléculas simples denominadas nucleótidos. Existen cuatro nucleótidos o bases nitrogenadas, llamadas Adenina, Guanina, Citosina y Timina, cuya respectiva abreviación es A, G, C y T. La información genética está codificada dentro de esta cadena. Las cadenas de ADN se representan con una secuencia de caracteres, donde cada uno corresponde a un nucleótido. El ADN está compuesto por una doble cadena de moléculas que están unidas formando una estructura helicoidal. La unión entre las dos cadenas se basa en que cada nucleótido tiene su par o complemento, al cual se une. La base A siempre se complementa con la T y la base C con la G.

  Ejemplo:

  [...AGTTCAAACGTACAT...]

- Gen

  Existen regiones en el ADN que codifican información y otras que no. Se llama gen a cada una de las regiones codificantes. En estas regiones puede haber información sobre la construcción de una proteína.

- ARN (ácido ribonucleico)

  Es una molécula como el ADN donde se reemplaza la molécula Timina (T) por Uracilo (U). Además, el ARN está formado por una única cadena. La forma de representar esta cadena es la misma que para el ADN.

  Ejemplo:

  [...AGUUCAAACGUACAU...]

- Proteína

Es una cadena de moléculas simples llamadas aminoácidos. La forma de escribirlas es igual que una cadena de ADN o ARN. Estas moléculas también se representan por abreviaciones.

Ejemplo:
[…DVATAKIISKEVSDGVIAPGYEPEALNILSKKKNGKYCILQIDPNYVPGQMES…]

Código de una letra	Código de tres letras	Nombre
A	Ala	Alanine
C	Cys	Cysteine
D	Asp	Aspartic Acid
E	Glu	Glutamic Acid
F	Phe	Phenylalanine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
K	Lys	Lisien
L	Leu	Leucine
M	Met	Methiomine
N	Asn	Asparagine
P	Pro	Proline
Q	Gln	Glutamine
R	Arg	Arginine
S	Ser	Serine
T	Thr	Threonine
V	Val	Valine
W	Trp	Tryptophan
Y	Tyr	Tyrosine

Tabla 1: Abreviaciones de proteínas

- Dogma Central de la Biología Molecular

En el siguiente esquema muestra el flujo de información genética en una célula y los diferentes procesos que relacionan el ADN, ARN y Proteína.

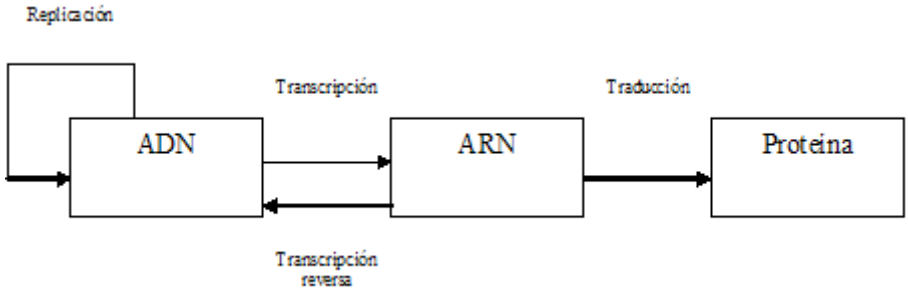

Figura 1: Dogma Central de la Biología Molecular

- Codones

  Un codón es una terna cualquiera de nucleótidos que codifica a un aminoácido. Ej.: AGT, TCC, GCA

- Transcripción

  Es el proceso biológico por el cual se genera una copia de un gen (ADN) en una molécula de ARN. Durante esta copia se transcribe al nucleótido T como U.

- Traducción

  Es el proceso biológico por el cual se genera una proteína a partir del ARN. La especificación de cada aminoácido se realiza en base a codones. Comúnmente se representan en una tabla donde se indica los posibles codones y el aminoácido resultante.

Primera posición	**Segunda posición**				**Tercera posición**
	G	A	C	U	
G	Gly	Glu	Ala	Val	G
	Gly	Glu	Ala	Val	A
	Gly	Asp	Ala	Val	C
	Gly	Asp	Ala	Val	U
A	Arg	Lys	Thr	Met	G
	Arg	Lys	Thr	Ile	A
	Ser	Asn	Thr	Ile	C
	Ser	Asn	Thr	Ile	U
C	Arg	Gin	Pro	Leu	G
	Arg	Gin	Pro	Leu	A
	Arg	His	Pro	Leu	C
	Arg	His	Pro	Leu	U
U	Trp	STOP	Ser	Leu	G
	STOP	STOP	Ser	Leu	A
	Cys	Tyr	Ser	Phe	C

	Cys	Tyr	Ser	Phe	U

Tabla 2: Mapeo entre codones y aminoácidos

- Alineamientos de secuencias

  Las técnicas de alineamiento son útiles para realizar búsquedas en bases de datos. Dada una secuencia de Nucleótidos o Aminoácidos se obtiene una lista de secuencias que tengan algún grado de similitud. Este proceso puede ser muy costoso para bases de datos grandes por lo que se utilizan heurísticas que permiten hacer la búsqueda en forma eficiente aunque los resultados no sean exactos. Uno de los programas más utilizados para realizar este tipo de búsquedas es BLAST (Basic Local Alignment Search Tool) [Setubal97] y sus variantes [Altschul].

  Un alineamiento global de dos o más secuencias se realiza mediante la inserción de espacios (gaps) en las mismas con el objetivo de que sean todas del mismo tamaño y puedan ser puestas una arriba de otra para poder ver cuán similares son. Existen muchas formas de alinear dos secuencias. A cada alineamiento se le asigna un store, que estará determinado por una función que indica cuánto se parecen entre sí. Esta función premia las coincidencias de caracteres (match) y penaliza los espacios (gaps) y las diferencias (mismatch). El score máximo de todos los alineamientos posibles entre un conjunto dado de secuencias se denomina **similitud**. En general, puede haber varios alineamientos distintos que tengan un score máximo.

  Ejemplos:

  GA-CGGAGG**AG**
  GAT**CGGA**TT**AG**

  CAG**CA-C**T**TGG**ATTCTCGG
  ---**CA**G**C**G**TGG**--------

  CAGCACTTGGATTCTCGG
  CAGC-----G-T----GG

  Dadas 2 se secuencias s y t, se realizan tres tipos de alineamientos:

  - **Global**: utiliza toda la cadena s y toda la cadena t para realizar el alineamiento.

- **Local** : encuentra una alineación entre una sub-cadena de s y una sub-cadena de t

- **Semi-global**: se calcula el score del alineamiento ignorando los gaps en los extremos de las secuencias

- Alineamientos múltiples de secuencias

  El alineamiento múltiple es una generalización del alineamiento de dos secuencias.

  Sea S={s1,...,sk} un conjunto de secuencias sobre el mismo alfabeto (nucleótidos o aminoácidos), un alineamiento múltiple de S se obtiene insertando espacios en las secuencias de manera de hacerlas del mismo tamaño. Es costumbre ubicar las secuencias extendidas en una lista de manera que los caracteres (o espacios) ocupen su correspondiente posición en la misma columna. Además se requiere que ninguna columna este constituida por espacios.

  Ejemplo:

```
MQPILLL
MLR-LL-
MK-ILLL
MPPVLIL
```

- Dominio Conservado (Dominio o Motivo)

  Se utiliza este término para definir una secuencia de aminoácidos que puede ser identificada en una o más proteínas. La aparición de dominios en distintas proteínas es consecuencia de las restricciones estructurales que sufren estas regiones, y su conservación a lo largo de la evolución responde a la necesidad de mantener su propiedad biológica o función. Los dominios se pueden utilizan para definir familias de proteínas que comparten el mismo motivo a pesar de la falta de similitud global entre ellas. Una proteína puede contener uno o más dominios y/o repeticiones del mismo.

  Para representar los motivos, se utilizan diferentes métodos basados en 3 técnicas básicas. Estas son Expresiones Regulares, Perfiles o Matrices de Pesos y Modelos Ocultos de Markov.

  Expresiones Regulares

Este ha sido el primer sistema de codificación de motivos. Veremos cómo se define una expresión regular mediante un ejemplo. Asumamos que poseemos un alineamiento regular de tres secuencias alrededor de un residuo activo (en nuestro caso Histidina, H)

ALRDFAT**H**DDF
SMTAEAT**H**DST
ECDQAAT**H**EAS

Una expresión regular para representar el patrón común de estas secuencias sería ATH[D o E]. En este caso el patrón es pequeño y podría generar falsos positivos (indicar erróneamente que estamos en presencia de un dominio) al comparar con otras secuencias.
Las siguientes son convenciones para describir a estos patrones:

- El código estándar IUOAC de una letra para notar los aminoácidos
- El símbolo x es usado para especificar una posición donde cualquier aminoácido es aceptado
- Las ambigüedades se resuelven listando los aminoácidos aceptables para esa posición dentro de corchetes. Por ejemplo [ALT] indica Ala , Leu o Thr.
- También se resuelven las ambigüedades utilizando llaves indicando los aminoácidos no aceptados en cierta posición. Ej.: {AM} indica cualquier aminoácido excepto Ala y Met.
- Cada elemento de un patrón se separa de su vecino por un guión '-'.
- Las repeticiones de un elemento dentro del patrón se indican siguiendo al elemento con un valor numérico o un rango numérico entre paréntesis. Ej.: x(3) corresponde a x-x-x, x(2,4) corresponde a x-x o x-x-x o x-x-x-x.

Ejemplos:
Patrón: [AC]-x-V-x(4)-{ED}
Representa: **[Ala or Cys]-any-Val-any-any-any-any-{any but Glu or Asp}**

Patrón: A-x-[ST](2)-x(0,1)-V.
Representa: **Ala-any-[Ser or Thr]-[Ser or Thr]-(any or none)-Val**

Perfiles o matrices de peso
Es una tabla que contiene pesos asociados a aminoácidos y penalizaciones

asociadas a la no-aparición (gap). Estos valores (o scores) son utilizados para calcular un valor de similitud para cualquier alineamiento entre un perfil y una secuencia. Una alineación con un valor de similitud superior o igual a cierto umbral de corte constituye una aparición del motivo. Los perfiles se pueden construir de varias maneras. El método más usado [Gribskov90] requiere un alineamiento múltiple a partir del cual, usando una matriz de puntuaciones para los cambios de aminoácidos, se obtienen los perfiles a partir de las distribuciones de frecuencias de residuos. Al contrario que en las expresiones regulares, los perfiles no están confinados a pequeñas regiones con elevada similitud de secuencia, sino que se intenta caracterizar familias de proteínas o dominios completos. Los perfiles son más sensibles y robustos que las expresiones regulares ya que a partir de los alineamientos se puede obtener una puntuación discriminatoria no sólo para los residuos presentes en una posición determinada, sino para aquellos que no aparecen.

Una ampliación sobre este tema puede leerse en [Gribskov90], [Gribskov87] y [Luethy94].

Modelos Ocultos de Markov

Son modelos estadísticos que permiten modelar el consenso de la estructura primaria de una familia de secuencias. Los modelos utilizados para representar dominios se generan a partir de un alineamiento de secuencias que contienen un motivo. Este motivo o dominio comúnmente esta asociado a una función de la proteína. La ventaja de usar HMMs es que estos tienen una base probabilística y aunque esto parezca restringido a un estudio académico, el sustento probabilístico permite identificar propiedades que otras heurísticas no permiten.

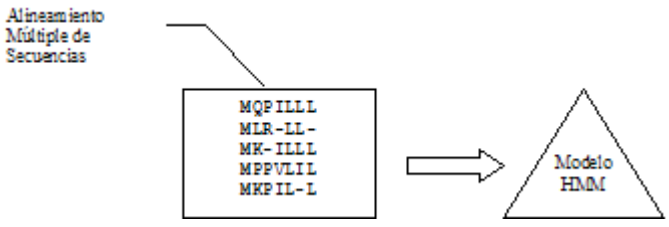

Figura 2: Esquema de la obtención de un modelo HMM

Una vez construido el modelo se lo aplica a una secuencia cualquiera para determinar si ésta cumple con el modelo, si así fuera, decimos que la secuencia contiene el dominio representado por el modelo.
Una explicación más amplia sobre HMM (Hidden Markov Model) y la teoría subyacente puede leerse en [Durbin98] y [Krogh94].

Existen varias bases de datos de dominios que utilizan alguna de las técnicas anteriormente mencionadas. Una de las más conocidas que utiliza HMM para representar dominios es PFAM [Sonnhammer97].

- Perfil Hidrofóbico / Hidrofílico

  Se utiliza para saber qué partes de la proteína generaran una atracción al agua (Hidrofílico) o repulsión (Hidrofóbico).
  El perfil hidrofílico de una proteína se calcula asignando a cada aminoácido un valor numérico (un valor Hidropático) determinado por alguna de las tabla (Hopp-Woods o Kyte-Doolittle) y posteriormente promediando estos valores a través de la cadena de aminoácidos. Un valor superior a 0 significa que esa región es hidrofóbica y un valor inferior a 0 indica que estamos en una región hidrofílica. Para realizar este cálculo se debe definir de antemano la longitud de la ventana en la secuencia de aminoácidos sobre la cual se calculará el promedio.
  Para una ampliación sobre este tema puede leerse [Hopp81] y [Kyte82]

- Compartimientos celulares

  Dentro de las células existen varios compartimientos que poseen diversas funciones. Algunas bases de datos identifican el compartimiento donde se encuentra la proteína. Se sabe que las interacciones proteína-proteína se desarrollan dentro del mismo compartimiento.
  La siguiente es una lista de los principales compartimientos: Núcleo, Citoplasma, Mitocondria, Membrana, Golgi, Retículo endoplasmático, Peroxisoma, Flagelo, Lisosoma, Vacuola y Ribosoma.

Capítulo 2 - Problema biológico a tratar

Las funciones de las proteínas se basan en su estructura de tres dimensiones. Esta estructura está determinada por la secuencia de aminoácidos de la proteína. Se han utilizado diferentes técnicas para determinar estas estructuras, como la cristalografía de rayos–X o resonancia magnética nuclear, que demostraron ser complejas, costosas y no siempre funcionaron.

La mayoría de las funciones biológicas dentro de una célula la realizan las proteínas y la mayoría de los procesos biológicos y eventos bioquímicos finalmente se realizan por interacciones entre proteínas. Conociendo estas interacciones se puede formar una “red de interacciones de proteínas”. Estas redes son de especial interés para los investigadores y se utilizan para entender el funcionamiento de diferentes organismos. Por ejemplo, supongamos que podemos descubrir la red de interacción de un organismo que genera cierta enfermedad, los investigadores podrían desarrollar medicamentos que interfieran en una interacción esencial de ese organismo y así impedir su evolución.
Actualmente se realizan numerosos estudios para determinar qué proteínas interactúan en determinados organismos. El alto costo que implica realizar estos estudios llevó a los biólogos a buscar formas alternativas para determinarlas. Al estudiar un nuevo organismo es útil aplicar el conocimiento que se tiene de otros ya estudiados. Pero el mayor problema consiste en que no se pueden aplicar en forma directa los resultados del estudio de un organismo en otro.
El objetivo de nuestro trabajo es predecir la red de interacciones de un organismo simple unicelular utilizando interacciones de proteína-proteína ya conocidas en otros organismos.

Capítulo 3 - Problema computacional

Una de las formas más intuitivas para predecir estas interacciones es utilizar un algoritmo de alineación (Ej. BLAST) para determinar qué proteínas encontramos semejantes entre los organismos. Sabiendo que las proteínas A y B interactúan, y encontrando un par de proteínas similares en el organismo estudiado, se podría suponer que estas proteínas deberían interactuar ya que su estructura es similar. El siguiente esquema muestra esta idea:

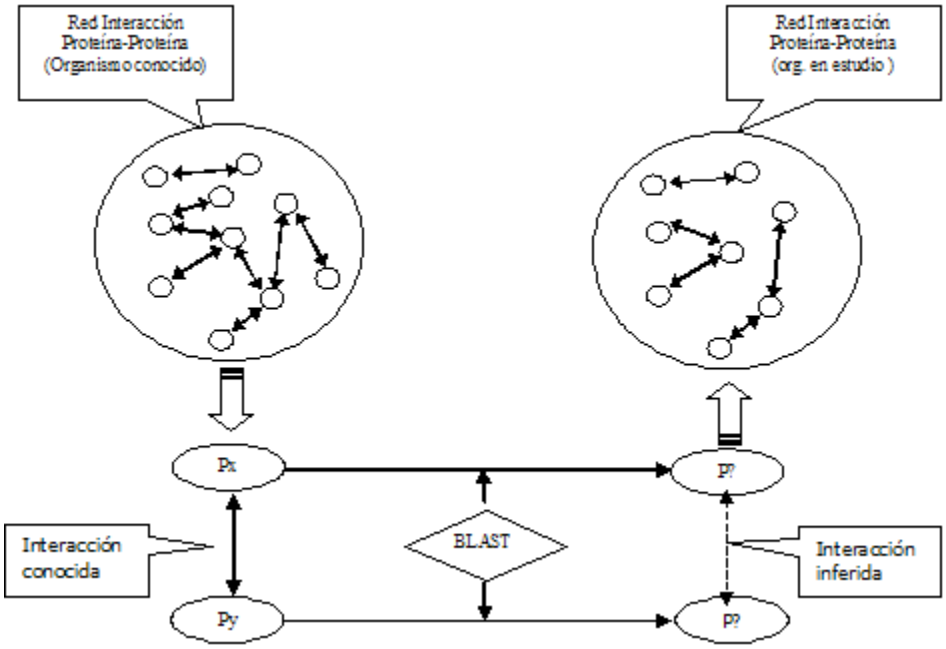

Figura 3: Inferencia de interacciones en forma directa

Los resultados obtenidos con este método son claramente limitados. Forzar la semejanza de dos proteínas en organismos diferentes es restrictivo. La red de interacciones inferida entre proteínas será pobre, porque no necesariamente se encontrará un par de proteínas homólogas en el organismo en estudio. Además no se utiliza información sobre los dominios de las proteínas.

Los dominios o motivos conservados son regiones de las proteínas que permanecen con pocas alteraciones a lo largo de la evolución. El estudio de estos dominios determinó que cumplen funciones similares en diferentes proteínas e incluso en diferentes organismos. Dado que las proteínas interactúan entre ellas a través de dominios específicos, si lográramos predecir gran parte de las interacciones dominio-dominio, se podrán predecir las interacciones proteína-proteína. Teniendo esto en cuenta podemos reformular el problema original y construir una “Red de Interacciones de Dominios”. La construcción de esta red se

basa en el análisis de los dominios que poseen las proteínas que interactúan. Para inferir una interacción en un organismo, es necesario determinar los dominios de cada proteína y luego evaluar si existe alguna interacción dominio-dominio.
Utilizaremos el siguiente esquema para desarrollar este trabajo:

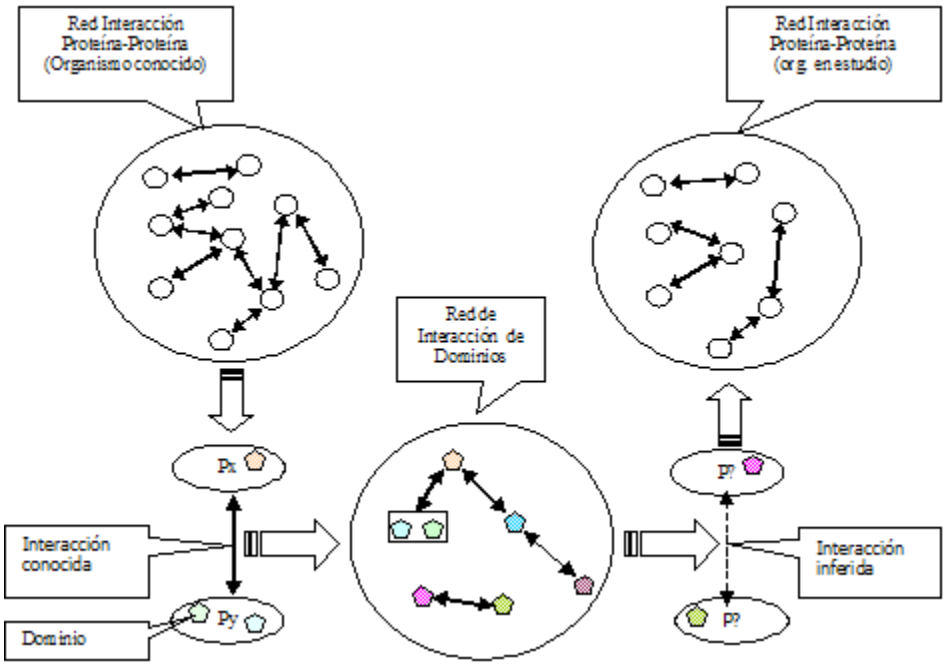

Figura 4: Inferencia de interacciones utilizando dominios

Este tipo de problemas presenta las siguientes dificultades desde el punto de vista computacional:

1) Información incompleta – La información disponible ya sea de dominios como de interacciones es incompleta. Sólo se conoce un porcentaje de la red de interacciones proteína-proteína.
2) Ambigüedad de la falta de información. Las bases de datos de interacciones sólo especifican que dos proteínas interactúan. El hecho de que dos proteínas no estén relacionadas en esa base se puede interpretar de la siguiente manera:
   a) No interactúan porque biológicamente no deben interactuar (información que sería de suma importancia para mejorar los modelos).
   b) No se realizaron los experimentos para determinar si estas proteínas interactúan.
3) Sesgo de la información. Los experimentos que se desarrollan están orientados para descubrir ciertos grupos de interacciones proteína-proteína

que los biólogos consideran que pueden brindar información importante. Lo ideal sería hacer un estudio exhaustivo de todas las proteínas, pero debido a los costos que ello implica no es prácticamente factible.

Representación de las redes de interacciones.

Una red de interacciones (de proteínas o dominios) puede representarse de varias maneras, por ejemplo, mediante un grafico o enumerando cada interacción en forma individual.

El ejemplo siguiente muestra una red de interacción de dominios representada gráficamente y una alternativa enumerando sus interacciones.

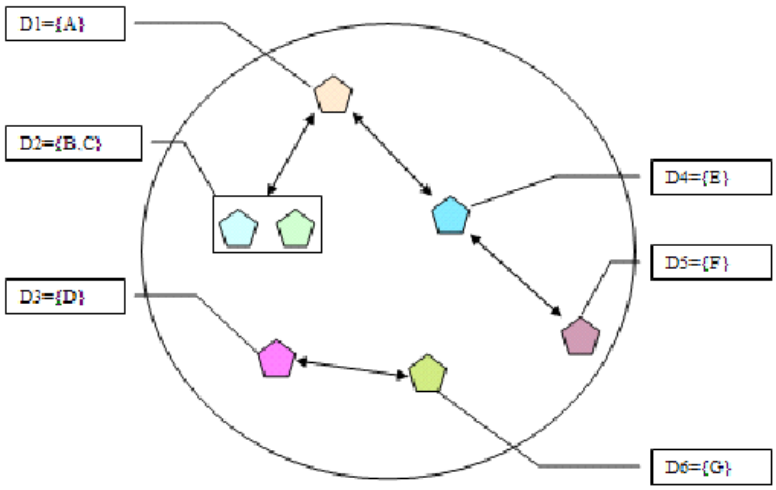

Figura 5: Representación gráfica de una red de interacciones de dominios

Interacciones			Interactuantes	
D1	D2		D1	{A}
D1	D4		D2	{B,C}
D4	D5		D3	{D}
D3	D6		D4	{E}
			D5	{F}
			D6	{G}

Tabla 3: Representación de una red de interacciones de dominios

Capítulo 4 - Trabajos Previos

El desarrollo de métodos para predicción de interacciones proteína-proteína e interacciones dominio-dominio es un tema de investigación de actualidad. Todos estos métodos (inclusive el propuesto en este trabajo) se enfrentan con la problemática de contar con datos insuficientes y con el sesgo que poseen los datos debido a que los experimentos realizados se centran en determinadas proteínas o interacciones y no llevan a cabo un estudio exhaustivo [Mrowka01].
Los métodos tienen diferente grado de complejidad, comenzando por los más simples que se basan en lógicas inductivas [Thierry01], los que realizan búsquedas de pares de proteína homólogas utilizando algún método de alineación [Matthews01] y los que tratan de predecir interacciones basados en que algunos pares de proteínas (o dominios) interactuantes poseen homólogos en otro organismo fusionados en una proteína (dominio) simple [Marcotte99] (Rostta stone method).
Existen técnicas más complejas basadas en el cálculo de cuán probable puede ser una red de interacciones de proteínas en un organismo dado. Estas técnicas tratan de completar la red utilizando simulaciones estadísticas (Markov Chain Monte Carlo simulation) [Gomez02] [Gomez01]. El principal problema de estas técnicas es que requieren un gran poder de cómputo restringiendo su utilización práctica a problemas reducidos.
Otras técnicas utilizan la región de la proteína que está involucrada en la interacción para realizan un cluster de proteínas basado en el patrón de interacción y en la secuencia de aminoácidos involucrada. Una vez definidas las particiones, se construye un modelo para cada una de ellas que representa el fragmento que tienen en común todas las proteínas de una partición (por ejemplo, un modelo HMM) [Wojcik01]. Estos métodos son interesantes porque no se basan en dominios predefinidos, por el contrario se van creando a medida que se descubren. Lamentablemente, para muy pocas proteínas se conoce cuál es su región interactuante, por lo que no se puede emplear como método general para inferir interacciones.
Por último, se encuentran las técnicas estadísticas. Éstas definen un modelo general para el problema y luego buscan los parámetros para que ese modelo se ajuste a los datos observados. Estas técnicas también consideran que puede haber falta de información, como es el caso de las base de datos de interacciones que contienen las proteínas que interactúan pero no tiene aquellas que no lo hacen [Deng02].

En este trabajo mostraremos dos de estos algoritmos y veremos, como mencionamos anteriormente, que los métodos estadísticos requieren importantes recursos computacionales para generar resultados prácticos. Estos algoritmos fueron informados previamente en [Deng03].

Las técnicas que infieren interacciones dominio-dominio están limitadas a la información disponible en las bases de datos de dominios (PROSITE, PRINTS, BLOCKS, Pfam, etc.). A medida que se descubran más dominios, más precisas resultaran estas técnicas. Muchas de ellas generan resultados pobres debido a que todavía existe un gran número de proteínas que no contienen dominios conocidos y por lo tanto quedan excluidas del análisis.
La falta de un adecuado conocimiento de los dominios genera serias dificultades, ya que los modelos necesitan por lo menos una instancia de una interacción dominio-dominio en el conjunto de entrenamiento para ser inferida.

Método de Asociación

Comenzaremos el estudio de un algoritmo existente para inferir interacciones basado en la siguiente suposición.

> *Dos proteínas interactúan sí y sólo sí al menos un par de dominios de ambas proteínas interactúan.*

Basándose en esta suposición el método utiliza la frecuencia relativa para calcular la probabilidad de que dos dominios D_m y D_n interactúen. Este valor será calculado como la proporción de interacciones tales que una proteína del par interactuante contiene el dominio D_m y la otra el dominio D_n, sobre la cantidad total de proteínas que contienen el dominio D_m multiplicado por la cantidad total de proteínas que contienen el dominio D_n.
Sea I_{mn} la cantidad de interacciones que contienen los dominios (D_m, D_n) y N_{mn} la cantidad total de pares de proteínas que contienen los dominios D_m y D_n. Este método define la medida de asociación como:

$$A(D_m, D_n) = \frac{I_{mn}}{N_{mn}}$$

Ejemplo:
Sea $P_I=\{D_A,D_X\}$, $P_J=\{D_Y,D_B\}$, $P_K=\{D_Y,D_C\}$, $P_L=\{D_X\}$

Sea I= {(P_I,P_J) , (P_I,P_K) }

$$A(D_A, D_Y) = \frac{I_{AY}}{N_{AY}} = \frac{2}{2} = 1$$

$$A(D_X, D_Y) = \frac{I_{XY}}{N_{XY}} = \frac{2}{4} = 0.5$$

Una vez calculado el valor de asociación se selecciona un umbral para determinar si los dominios interactúan.

Selección del umbral:

El umbral puede variar entre [0..1]. Seleccionando un umbral alto (mayor a 0.5) estaríamos privilegiando la exactitud de las reglas inferidas respecto de encontrar interacciones que, en el conjunto de entrenamiento, se den con poca frecuencia. Esta selección favorecerá las interacciones de dominios que se repitan siendo más probable que éstas se den en la realidad. Una selección así puede resultar desventajosa ya que infiere poca cantidad de IDD y podría estar pasando por alto interacciones que no se sean muy frecuentes.

Por el contrario, seleccionar un umbral bajo permite tener una mayor sensibilidad y detectar IDD que su aparición no sea tan frecuente en el conjunto de entrenamiento. Recordemos que los datos disponibles son incompletos y sesgados hacia un conjunto de proteínas por lo que es factible que esto suceda. Como desventaja, se obtiene un aumento de los falsos positivos (interacciones dominio-dominio que se predicen pero no son reales).

Este método se basa en la precisión de los datos observados, ya que no se contempla que contengan errores. Además, ignora otros dominios que pueden estar presentes dentro del par de proteínas.

Pseudocódigo del Método de Asociación:

Entrada:
Lista de interacciones.
Lista de proteínas con sus dominio asociados.
Salida:
Conjunto de reglas de interacción de dominios

1. Obtener la lista de dominios del conjunto de entrenamiento
2. **Para** cada par de dominios D_m, D_n calcular A(D_m, D_n)
3. **Si** A(D_m, D_n) >= umbral **entonces**
4. inferencias.agregar(D_m, D_n)

5. Fin Si
6. **Fin Para**

7. **Devolver** inferencias

El Paso 1 puede pensarse como una optimización al algoritmo ya que no tiene sentido recorrer todos los pares de dominios, sin analizar si estos están incluidos en proteínas que interactúan, ya que para estos casos, $I_{mn} = 0$ lo que implica que $A_{mn} = 0$.
En el Paso 2 se analizan las n(n-1) posibles combinaciones de agrupar dos dominios y se calcula su valor de asociación.
En el Paso 3 se compara el valor de asociación con el umbral de selección y si el valor de asociación es mayor al umbral definido se agrega una regla indicando que estos dominios interactúan (Paso 4).

Una vez realizada la inferencia de interacciones, disponemos de una Red de Interacción de dominios con la que podemos generar la predicción de interacciones proteína-proteína.
En función de la red obtenida, se establece que dos proteínas (P_I,P_J) interactúan si algún par de dominios (D_X,D_Y), con $D_X \in$ dominios(P_I) y $D_Y \in$ dominios(P_J), se encuentra en la red obtenida por el algoritmo.

Cálculo de orden

- El paso 1 requiere O(#interacciones * DominioMax^2)
- El paso 2 itera O(#dominios^2) veces
- El paso 3 requiere O(#interacciones * #proteínas + #proteínas) pasos

De esta forma se obtiene que el orden del algoritmo es O(#dominios^2 * #proteínas * #interacciones).

Estimación de Máxima Esperanza

Este algoritmo se basa en un método estadístico utilizado para la estimación de los parámetros de una función de distribución. El mismo propone una función de probabilidad de las interacciones de proteínas en base a sus dominios, luego en función de los datos de entrenamiento realiza la estimación de sus parámetros y por último, para inferir interacciones, definirá un valor mínimo de probabilidades

para decidir si dos proteínas interactúan.
Este algoritmo fue desarrollado bajo las siguientes suposiciones:

> *Suposición 1: Las interacciones de dominios son independientes, es decir, el hecho de que dos dominios interactúen no depende de otros dominios*.
>
> *Suposición 2: Dos proteínas interactúan sí y sólo sí al menos un par de dominios de ambas proteínas interactúan.*

Sean $D_1,..., D_m$ M dominios y $P_1,..., P_n$ N proteínas. Sea P_{ij} el par de proteínas P_i, P_j y D_{ij} el par de dominios D_i y D_j.
Definamos P_{ij} como los pares de dominios formados por las proteínas P_i y P_j.
Se tomaran las interacciones proteína-proteína y dominio-dominio como variables aleatorias. Sea $P_{ij} = 1$ si las proteínas i y j interactúan y $P_{ij} = 0$ en otro caso. De la misma forma, $D_{mn} = 1$ si D_m interactúa con el dominio D_n y $D_{mn} = 0$ si esto no sucede.

En función de las suposiciones, se define

$$\Pr(P_{ij} = 1) = 1.0 - \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) \quad (1)$$

donde $\lambda_{mn} = \Pr(D_{mn} = 1)$

Se deben considerar dos tipos de errores experimentales en los métodos biológicos utilizados para encontrar interacciones. Los falsos positivos que se producen cuando dos proteínas no interactúan en la realidad pero se observa esta interacción en los experimentos y los falsos negativos que se producen cuando dos proteínas interactúan pero este fenómeno no se detecta en los experimentos. Notaremos estos valores como fp y fn respectivamente.
Sea O_{ij} la variable aleatoria de las interacciones observadas. $O_{ij}=1$ si se observa la interacción y $O_{ij}=0$ en otro caso. Se puede definir:

$$fp = \Pr(O_{ij}=1 \mid P_{ij} =0)$$
$$fn = \Pr(O_{ij}=0 \mid P_{ij} =0)$$

$$\begin{aligned} \Pr(O_{ij} = 1) & \quad (2) \\ &= \Pr(O_{ij} = 1, P_{ij} = 1) + \Pr(O_{ij} =1 , P_{ij} =0) \\ &= \Pr(Oij = 1 \mid Pij = 1)\Pr(Pij = 1) \end{aligned}$$

$$+ \Pr(Oij = 1 \mid Pij = 0)(1 - \Pr(Pij = 1))$$
$$= \Pr(P_{ij} = 1)(1 - fn) + (1 - \Pr(P_{ij} = 1))fp$$

La función de esperanza de todo el conjunto de interacciones se define como :

$$L = \prod_{P_i, P_j} (\Pr(O_{ij} = 1))^{O_{ij}} (1 - \Pr(o_{ij} = 1))^{1 - O_{ij}} \quad (3)$$

donde O_{ij} : $\begin{cases} 1 \text{ si se observa la interacción } P_i, P_j \\ 0 \text{ en otro caso} \end{cases}$

La función L depende de los parámetros $\theta = (\lambda_{mn}, f_p, f_n)$ donde f_p y f_n estarán fijos y dependen de los datos.

El parámetro θ podría ser estimado mediante el método MLE (Maximum Likelihood estimation) pero debido a la gran cantidad de datos con los que se trabaja se torna complejo maximizar L. Esta restricción llevó a desarrollar el método de Maximización de Esperanza (Expectation Maximization EM) para resolver el problema MLE [Collins97], [Neng99], [Bilmes98].

Para encontrar los parámetros de máxima esperanza (MLE) se agregan a los datos observados, datos que no fueron observados (datos faltantes). El conjunto de los datos observados Y junto con los faltantes será el nuevo conjunto de datos Z.

El algoritmo genérico trabaja en dos pasos, en el primero (esperanza) se calcula la esperanza del conjunto completo de datos Z en función de los datos observador Y.

$$\hat{Z} : E(Z \mid Y, \theta^{(t-1)})$$

Luego en la etapa de maximización, se obtiene el MLE de θ , $\theta^{(t)}$ basándose en $\hat{Z}$.

Estos pasos dan la fórmula recursiva para estimar los parámetros θ.

Adaptando el algoritmo a este problema biológico en particular, tenemos que el conjunto de datos observados es el conjunto de las interacciones $O = \{O_{ij} = o_{ij}, i \leq j\}$. El conjunto completo de datos contiene todas las interacciones dominio-dominio de cada par de proteínas.

Definimos A_m como el conjunto de proteínas que contienen el dominio D_m, A_n como el conjunto de proteínas que contienen el dominio D_n y N_{mn} como el número total de pares de proteínas entre A_m y A_n.

Para estimar λ_{mn} se utilizará la información del estado de las interacciones para los pares de proteínas A_m y A_n. El conjunto de datos completos es (O,D) donde O es el conjunto de las interacciones observadas y $D = \{D_{mn}^{(ij)} P_i \in A_m, P_j \in A_n, \forall m, n\}$. $D_{mn}^{(ij)} = 1$ Si los dominios D_m y D_n interactúan en los pares de proteínas P_i y P_j y $D_{mn}^{(ij)} = 0$ en otro caso.

El cálculo de la esperanza se realiza de la siguiente forma:

$$
\begin{aligned}
& E(D_{mn}^{(ij)} \mid O_{kl} = o_{kl}, \forall k, l, \theta^{(t-1)}) \\
& = E(D_{mn}^{(ij)} \mid O_{ij} = o_{ij}, \theta^{(t-1)}) \\
& = \frac{\Pr(D_{mn}^{(ij)} = 1, O_{ij} = o_{ij} \mid \theta^{(t-1)})}{\Pr(O_{ij} = o_{ij} \mid \theta^{(t-1)})} \\
& = \frac{\Pr(D_{mn}^{(ij)} = 1 \mid \theta^{(t-1)}) \Pr(O_{ij} = o_{ij} \mid D_{mn}^{(ij)} = 1, \theta^{(t-1)})}{\Pr(O_{ij} = o_{ij} \mid \theta^{(t-1)})} \\
& = \frac{\lambda_{mn}^{(t-1)} (1 - fn)^{o_{ij}} fn^{1-o_{ij}}}{\Pr(O_{ij} = o_{ij} \mid \theta^{(t-1)})}
\end{aligned}
$$

Donde el denominador puede ser calculado utilizando la ecuación 2. El estimador de máxima esperanza de λ_{mn} es la fracción de $\{D_{mn}^{(ij)}, P_i \in A_m, P_j \in A_n\}$ tal que $D_{mn}^{(ij)} = 1$. Obtenemos de esta forma una fórmula recursiva para el paso de la maximización.

$$
\lambda_{mn}^{(t)} = \frac{1}{M_{mn}} \sum_{i \in A_m, j \in A_n} E(D_{mn}^{(ij)} \mid O_{kl} = o_{kl}, \forall k, l, \theta^{(t-1)})
$$

$$
\lambda_{mn}^{(t)} = \frac{\lambda_{mn}^{(t-1)}}{N_{mn}} \sum_{i \in A_m, j \in A_n} \frac{(1 - fn)^{o_{ij}} fn^{1-o_{ij}}}{\Pr(O_{ij} = o_{ij} \mid \theta^{(t-1)})} \qquad (4)
$$

El Pseudocódigo es el siguiente:

Entradas :
Lista de interacciones
Proteínas con sus dominios asociados
ε umbral para detener las iteraciones del algoritmo
nmax Cantidad máxima de iteraciones

Salida :
Salida matriz λ donde λ_{ij} indica la probabilidad que dos dominios interactúen

1. $\forall_{m,n}: \lambda_{mn} = 0.5$
2. Lant = MAX_FLOAT
3. L = MIN_FLOAT
4. I = 0
5. **Mientras** | Lant – L | < ε **e** i < nmax
6. i = i + 1
7. Lant = L
8. Calcular $Pr(P_{ij} = 1)$ utilizando la ecuación 1
9. Calcular $Pr(O_{ij} = 1)$ utilizando la ecuación 2
10. Actualizar los parámetros $\{ \lambda_{mn}, \forall_{m,n}\}$ utilizando la ecuación 4
11. L = Calcular la función de esperanza mediante la ecuación 3
12. **Fin Mientras**

Devolver λ

El algoritmo obtiene como resultado una matriz λ donde λ_{ij} indica la probabilidad que los dominios i, j interactúen. Luego para inferir si un par de proteínas interactúa, se deben obtener sus dominios y aplicar la siguiente fórmula.

$$\Pr(P_i, P_j) = \Pr(P_{ij} = 1) = 1.0 - \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn})$$

La última fórmula puede ser calculada mediante el siguiente algoritmo:

Entradas :
Matriz λ obtenida por el algoritmo
Proteína1
Proteína2
Salida :
Probabilidad de que la Proteína1 interactúe con la Proteína2

1. res = 1
2. Dominios1 = Proteina1.obtenerDominios()
3. Dominios2 = Proteina2.obtenerDominios()
4. **Para cada** d1 **en** Dominios1
5. **Para cada** d2 **en** Dominios2
6. res = res * $\lambda_{i,j}$
7. **Fin Para**
8. **Fin Para**

Devolver res

Cálculo de orden

En el cálculo de orden se debe tener en cuenta la cantidad de pasos necesarios para realizar una iteración y el orden de convergencia. Este último es complejo de calcular y excede a este trabajo, por lo tanto, nos manejaremos con resultados empíricos que están volcados en la en la sección de resultados. Nos limitaremos a analizar la complejidad de cada iteración.

La cantidad de operaciones estará dada en función de las variables:
#Dominios = Cantidad total de dominios que poseen las proteínas analizadas
DominioMax = Cantidad máxima de dominios que se puede encontrar en una proteína del conjunto
#Proteínas = cantidad de Proteínas en el conjunto de entrenamiento

- El paso 1 tiene O(#Dominios^2)
- Los pasos 2,3 y 4 tienen O(1)
- El paso 5 determinará la cantidad de iteraciones, este se resuelve en O(1) operaciones. Para este cálculo asumiéremos que el bucle **mientras** itera sólo una vez
- Los pasos 6 y 7 tienen O(1)
- La ecuación 1 y por lo tanto el paso 8 requieren O(DominioMax^2) pasos
- El paso 9 requiere O(#Proteínas^2) pasos
- El paso 10 depende de la ecuación 4 donde se actualiza cada posición de la matriz λ utilizando una doble sumatoria sobre las proteínas. De esta forma se realizan realiza O(Dominios^2) actualizaciones a la matriz donde

cada paso requiere O(#Proteínas^2) operaciones.

- Por último el paso 11 realiza una doble productoria sobre las proteínas lo que requiere O(#Proteínas^2) operaciones.

En resumen, la cantidad de operaciones necesarias para la inicialización del algoritmo es O(#Dominios^2), que debe ser sumado a la cantidad de operaciones de cada iteración donde tenemos: O(DominioMax^2) + O(#Proteínas^2) + O(#Dominios^2)* O(#Proteínas^2) + O(#Proteínas^2).
En total obtenemos O(#Dominios^2) * O(#Proteínas^2) para cada iteración.

Capítulo 5 - Algoritmo Basado en Reglas

Evolución del algoritmo

En esta sección describiremos el algoritmo propuesto para la resolución de este problema.
En primer lugar comentaremos el origen del problema y luego nuestra solución. Este trabajo de tesis surgió de reuniones que mantuvimos con un equipo de biólogos del INGEBI [INGEBI] dirigidos por el Dr. Mariano Levin [http://proteus.dna.uba.ar/levin.htm] interesados en el estudio de las interacciones proteína-proteína.
Originalmente el problema planteado consistía en inferir interacciones proteína-proteína en *Tripanosoma cruzi* a partir de las interacciones de Levadura (*Saccharomyces Cerevisiae*).
Al plantearnos el problema también nos presentaron un posible método que estaban considerando. Este consistía en utilizar la base de proteínas de Levadura y su base de datos de interacciones para inferir posibles interacciones de *Tripanosoma cruzi*. El primer paso del algoritmo toma cada proteína que interviene en interacciones y realiza una búsqueda con Blast en la base de T. cruzi. Luego se infiere que las proteínas similares a cada proteína interactuante lo harán también en T. cruzi.

Pseudocódigo:

Entradas:
Interacciones de Levadura
Proteínas Homólogas de Levadura en T.cruzi
Salida:
Inferencias realizadas en T.cruzi

1. **Para** cada interacción A,B de Levadura
2. A_1,A_2, ..., A_n <- Buscar proteínas homólogas a A en T.cruzi
3. B_1,B_2, ..., B_n <- Buscar proteínas homólogas a B en T.cruzi
4. **Si** existen A_1,A_2, ..., A_n y B_1,B_2,...,B_m **entonces**
5. **Para** i = 1 a n
6. **Para** j=1 a m
7. inferencias.agregar(Ai, Bj)
8. **Fin Para**
9. **Fin Para**
10. **Fin Si**
11. **Fin Para**

12. **Devolver** inferencias

Se trata de un algoritmo simple que obtiene resultados limitados. Como explicamos anteriormente, forzar que dos proteínas se asemejen en organismos diferentes suele ser demasiado restrictivo. La red de interacciones de proteínas inferida es muy pobre, principalmente por no poder encontrar un par de proteínas homólogas en el organismo en estudio. Además, no se está utilizando la información sobre los dominios subyacentes que generalmente representan funciones dentro de la proteína.

Definición del Algoritmo Basado en Reglas

A continuación presentaremos nuestra propuesta para resolver este problema, se trata de un algoritmo basado en reglas para predecir interacciones de dominios. El objetivo de esta clase de algoritmos es aprender un conjunto de reglas de primer orden que describen la solución. El tipo de algoritmo que usaremos es de la forma “Aprender una regla a la vez” que dará como resultado un conjunto de reglas del tipo los dominios i,j,k interactúan con los dominios l,m,n.
El algoritmo que proponemos utiliza la estrategia de aprender una regla, eliminar del conjunto de datos todos los casos que cubre la regla y repetir el proceso hasta que no queden más datos por recorrer.
Cada una de las reglas cubrirá un conjunto de casos de los datos, algunos de los casos serán aciertos y otros serán predicciones que no se corresponden con los

datos observados. El objetivo será encontrar reglas que generen la mayor cantidad de aciertos y que minimicen la cantidad de fallos. Esta clase de condiciones generará un conjunto de reglas precisas que, dependiendo de la precisión que se le otorgue al algoritmo mediante el parámetro umbral, provoque un bajo cubrimiento de los datos, es decir, si se selecciona una precisión muy alta, podría suceder que no hubiera reglas para todos los datos del conjunto de entrenamiento.
A este tipo de algoritmos se los denomina de cubrimiento secuencial, ya que secuencialmente descubren reglas que en su conjunto cubren los datos de entrenamiento. Mas detalle sobre este tipo de algoritmo puede encontrarse en [Mitchell97].

Sabemos que las interacciones de proteínas en gran medida se producen entre los dominios de la proteína. Es decir, las interacciones sólo se producen en una sub-secuencia de aminoácidos de la proteína. Este dato nos da una idea sobre cómo podrían inferirse las interacciones de proteínas.
Conociendo los dominios que se encuentran en las proteínas que interactúan, y asumiendo que el motivo por el cual se produce dicha interacción es la presencia de estos dominios, fácilmente obtenemos el siguiente criterio para inferir reglas (interacciones dominio-dominio).

> Criterio: “Si un par de conjuntos de dominios A y B se observan en un par de proteínas que interactúan, entonces el motivo por el cual estas proteínas interactúan es la presencia de estos dominios“

Utilizando este criterio podemos analizar las interacciones, obtener los conjuntos de dominios interactuantes y definir que la presencia de estos dominios en las proteínas es la razón por la cual se produce una interacción.
Al desarrollar un algoritmo con estas características, comprobamos que este tipo de relaciones no producen resultados satisfactorios. Esto se produce por la gran cantidad de falsos positivos que se generan. Llamaremos falsos positivos a las interacciones inferidas que no están en el conjunto de las interacciones observadas. Los falsos positivos aparecen donde hay conjuntos de dominios que se encuentran en muchas proteínas, pero éstas no interactúan.
Siguiendo este razonamiento concluimos que no es posible seleccionar cualquier conjunto de dominios y proponerlo como regla de inferencia, sino que es necesario imponer otras condiciones al elegir una regla y verificar que la

proporción de interacciones inferidas sobre la cantidad de interacciones reales (que se observan al elegir esta regla) se mantenga elevada. Es decir, encontrar reglas que al aplicarlas generen la mayor cantidad posible de interacciones dadas como dato (o conjunto de entrenamiento) y generen una baja cantidad de falsos positivos.
Reformularemos el criterio del algoritmo para mejorar su eficiencia :

> Criterio: “Si un par de conjuntos de dominios A y B se observan en un par de proteínas que interactúan, entonces el motivo por el cual estas proteínas interactúan es la presencia de estos dominio” además, “se deben seleccionar los conjuntos de dominios de manera tal que, al agregar una regla, la cantidad de interacciones conocidas que cumplen la regla sea alta y la cantidad posible de interacciones que cumplen la regla sea baja”.

Este criterio describe claramente un nuevo método a utilizar para inferir interacciones de dominios, pero ¿Qué significa que la cantidad de interacciones conocidas que cumplen la regla sea alta y la cantidad posible de interacciones que cumplen la regla sea baja ?

Para precisar el algoritmo asignaremos un valor de confiabilidad a cada interacción de dominios inferida. Este valor lo definimos como la proporción de interacciones conocidas que se infieren mediante esta regla (aciertos), sobre la cantidad total de posibles interacciones.
Además definiremos un umbral de corte mediante el cual se decidirá si una regla será agregada o no.

$$r(A,B) = regla\ en\ estudio$$
$$P_A = \Pr\ oteinas\ que\ poseen\ el\ conjunto\ de\ do\min io\ (A)$$
$$P_B = \Pr\ oteinas\ que\ poseen\ el\ conjunto\ de\ do\min io\ (B)$$
$$IO_{AB} = Interaccion\ observadas\ con\ los\ conjuntos\ de\ do\min io(A,B)$$
$$Confiabilidad(r) = \frac{IO_{AB}}{P_A P_B}$$

Con estos elementos tenemos un algoritmo con el cual podemos inferir interacciones entre conjuntos de dominios.
Una de las características deseables para este tipo de algoritmos es que las reglas inferidas sean lo más genéricas posibles, por lo que reformularemos el criterio

para considerarlo.

> Criterio: “Si un par de conjuntos de dominios A y B se observan en un par de proteínas que interactúan, entonces el motivo por el cual estas proteínas interactúan es la presencia de estos dominios” además “se deben seleccionar los conjuntos de dominios de manera tal que, al agregar una regla, la cantidad de interacciones conocidas que cumplen la regla sea alta y la cantidad posible de interacciones que cumplen la regla sea baja”. “Serán privilegiadas aquellas reglas que contengan la menor cantidad de dominios”

Este nuevo criterio condicionará que el algoritmo comience la búsqueda de reglas que tengan la menor cantidad de dominios. Además, irá generalizando en cada paso infiriendo que una regla califica para una proteína si el conjunto de dominios de la regla está contenido en el conjunto de dominios de la proteína.
Con este último criterio ya es posible formular un algoritmo y lo llamaremos Algoritmo Basado en Reglas. Buscamos ahora fortalecerlo utilizando información biológica.

> Criterio: ... además “Las interacciones se producen entre proteínas que se alojan en el mismo compartimiento de la célula”

Esta regla impone restricciones biológicas sobre la inferencia de interacciones. Se sabe que las proteínas que interactúan lo hacen dentro del mismo compartimiento celular. Por ejemplo, una proteína alojada en el núcleo de la célula no podrá interactuar con otra que forme parte de la mitocondria.
Las bases de datos de las proteínas generalmente poseen anotaciones, las cuales permiten buscar un conjunto de palabras clave para determinar en qué región de la célula se encuentran. Son ejemplos de éstas palabras clave: nuclear, cytoplasmic, mitochondria, etc.
De esta forma, catalogamos las proteínas para que, al momento de analizar si una proteína puede interactuar con otra (ya que posee los mismos dominios), se verifique también que pertenecen al mismo compartimiento celular.
En un estudio que realizamos sobre las interacciones de proteínas en los distintos compartimientos, se observa una particularidad con el compartimiento membrana, ya que las estadísticas sobre las interacciones muestran que las proteínas alojadas en este compartimiento interactúan en forma muy frecuente

con otras de otros compartimientos. Entendemos que esta característica tiene también un significado biológico, ya que al estar hablando de membrana, este compartimiento se encuentra en contacto con otros. El apéndice I presenta el análisis y los resultados de la cantidad de interacciones existentes entre dos compartimientos.
Por lo tanto, este criterio se agrega al algoritmo por dos razones. Por un lado para incorporar conocimiento biológico y buscar mayor precisión, y también porque estudiando los resultados intermedios del algoritmo, se puede ver que muchas posibles reglas son descartadas debido a que, por ejemplo, existen dominios que aparecen en una gran cantidad de proteínas. Notar que el denominador de la variable confiabilidad de una regla se calcula en forma de producto cartesiano, por lo cual, la confiabilidad disminuye rápidamente en estos casos.

Con el agregado de nuevos filtros buscamos acotar el producto cartesiano que se obtiene al contar la cantidad de interacciones que se agregarían en caso de utilizar una regla, considerando información biológica.

Otro criterio que será agregado al algoritmo tiene relación con la forma de las proteínas y la posibilidad que tiene para interactuar una región de las mismas.

> Criterio: ...“Las regiones de la proteína donde se produce la interacción deben estar expuestas al solvente”

Esta regla también tiene significado biológico y tiene relación con el plegamiento de las proteínas. Asumiendo que las interacciones se producen entre dominios, si el dominio que se supone interactuante en una proteína está en una región interna de la proteína, es decir se encuentra en una región plegada hacia el interior de la misma, éste no tendrá posibilidades de interactuar con un dominio de otra proteína.
Para analizar esta característica de las proteínas, debemos contemplar el perfil hidrofóbico de la proteína. Éste nos indicará si una cierta región de la proteína repele el agua (hidrofóbico) o no (hidrofílico). Se sabe que las regiones que son hidrofóbicas tienden a plegarse hacia adentro de la proteína, por lo cual no interactuará con otra. Utilizaremos este tipo de información en el algoritmo para restringir la selección de aquellas proteínas que podrían interactuar. Cabe aclarar que podrían obtenerse mejores resultados conociendo el plegamiento exacto de una proteína y a partir de allí determinar las regiones expuestas, pero

actualmente no se conoce una solución polinomial para calcular el plegamiento exacto de cualquier proteína. Fue demostrado que el problema de plegamiento de proteínas pertenece al grupo de problemas llamado NP-Hard [Unger93].

Utilizando este criterio, el algoritmo buscará las proteínas que tienen ciertos dominios y las clasificará como candidatas a interactuar. Luego, analizará si la región donde se encuentra el dominio es hidrofóbica o no para determinar si ese dominio puede interactuar.

Pseudocódigo del Algoritmo Basado en Reglas:

Entradas:
Lista de interacciones
Proteínas con sus dominios asociados
Salidas:
Reglas de inferencia Dominio-Dominio

1. DominioMax = Cantidad máxima de dominios que tienen las proteínas
2. **Para** i = 1 **hasta** DominioMax
3. **Para** j = 1 **hasta** i
4. $interacciones_{IJ}$ = Obtener las interacciones que en su lado izquierdo tengan i dominios y en su lado derecho j dominios
5. **Para cada** $interaccion_{IJ}$ **en** $interacciones_{IJ}$
6. dominiosIzq = obtener los dominios que contiene el interactor izquierdo
7. dominiosDer = obtener los dominios que contiene el interactor derecho
8. interaccionesConocidas = contar la cantidad de interacciones que tienen en su parte izquierda contienen los dominios dominiosIzq y en su parte derecha contiene los dominiosDer.
9. protIzq = proteínas que contienen los dominiosIzq
10. protDer = proteínas que contienen los dominiosDer
11. ip = IntearaccionesPosibles (protIzq, protDer)
12. confiabilidad = interaccionesConocidas / ip
Si confiabilidad > umbral **entonces**
13. inferencias.agregar(dominiosIzq, dominiosDer)
14. eliminar todas las interacciones tales que dominisIzq esté incluido en el conjunto de dominios del interactor izquierdo
15. ídem dominios de la derecha
16. **Si no** descartar la regla
17. **Fin Para**
18. **Fin Para**
19. **Fin Para**

20. **devolver** inferencias

El algoritmo comienza calculando el número máximo de dominios que tienen las proteínas del conjunto de entrenamiento, ya que a lo sumo se inferirán reglas que tengan esa cantidad de dominios.

Los pasos 2 y 3 son los encargados de iterar buscando las reglas de dominios e

implementan el criterio bajo el cuál fue pensado el algoritmo. Se privilegiarán aquellas reglas que contengan la menor cantidad de dominios.
En el paso 4 se obtienen todas las interacciones que tienen una cierta cantidad de dominios y luego en el paso 5 se analizan una a una. Los pasos 6 y 7 obtienen los dominios que contienen las proteínas que forman esta interacción. Buscar los dominios de una proteína es un proceso costoso en tiempo de procesamiento. La implementación realizada asume que los dominios están precalculados y almacenados en una base de datos.
En el paso 8 ya disponemos una posible regla de dominios, (Ej.: {A,B}↔{C,D}) y se comienza a analizar la cantidad de interacciones que contienen estos dominios y la cantidad de proteínas que los contienen. En este paso se cuentan todas las interacciones que contienen estos dominios. Notar además que en este paso se consideran aquellas interacciones que tienen exactamente los dominios analizados y las interacciones que contienen estos dominios incluidos.
Los pasos 9 y 10 cuentan la cantidad total de proteínas que contienen los dominios analizados. Este valor se utilizará para calcular los falsos positivos que se estarían generando en caso de incluir la regla analizada.

En la línea 11 del algoritmo se realiza el cálculo de las interacciones posibles. Este valor se utilizará para el cálculo de la confiabilidad y significa la cantidad de interacciones que se estarían agregando en el caso de definir una regla con estos dominios. Como mencionamos anteriormente, existen varias formas de realizar esta estimación y la cantidad puede cambiar dependiendo del tipo de información que se utilice.

Diferentes formas para el cálculo de las posibles interacciones

i. No se utiliza información adicional

Ésta es la forma simple de calcular la cantidad posible de interacciones, ya que sin aplicar información adicional, las posibles interacciones se deben calcular como el producto cartesiano de las proteínas que contienen los dominios analizados.

```
IntearaccionesPosibles (protIzq, protDer)
        Devolver protIzq.cantidad * protDer.cantidad
```

ii. Analizando compartimientos celulares

El uso de compartimientos celulares agrega información biológica al cálculo de las interacciones posibles. El algoritmo analiza la descripción de cada proteína y obtiene el compartimiento celular donde ésta se encuentra (si este está informado). Luego calcula las interacciones posibles considerando aquellas que se encuentran en el mismo compartimiento.

Existen dos implementaciones realizadas de este algoritmo para considerar aquellas proteínas que no informan el compartimiento celular donde se encuentran. Una posibilidad consiste en no aplicar el control cuando el compartimiento no está informado y la segunda consiste en descartarlas.

Para el conjunto de datos utilizado desaconsejamos el uso de la última opción ya que se desperdicia gran cantidad de datos de entrenamiento debido a que la base de proteínas utilizada tiene sólo el veinte por ciento de las proteínas clasificadas.

Otra modificación implementada para este algoritmo fue el manejo del compartimiento membrana, que puede interactuar con todos los demás, por las razones expresadas en el apéndice I.

El siguiente es el algoritmo de considera los compartimientos e ignora las proteínas desconocidas.

IntearaccionesPosibles (protIzq, protDer, InteractorIzq, InteractorDer)

Entrada:
ProtIzq, ProtDer lista de proteínas
InteractorIzq, InteractorDer proteína

Salida:
Cantidad de proteínas que podrían interactuar ya que se encuentran en el mismo compartimiento celular

1. Izquierda = 0
2. Derecha = 0
3. CompartimientoIzq = interactorIzq.compartimiento()
4. CompartimientoDer = interactorDer.compartimiento()
5. **Para cada** proteína **en** protIzq
6. **Si** proteína.compartimiento() = CompartimientoIzq **entontes**
7. Izquierda = Izquierda + 1
8. **Fin Si**
9. **Fin Para**
10. **Para cada** proteína **en** protDer
11. **Si** proteína.compartimiento() = CompartimientoDer **entonces**
12. Derecha = Derecha + 1
13. **Fin Si**
14. **Fin Para**

15. **Devolver** Izquierda * Derecha

Este algoritmo es simple, ya que suma en forma condicional las proteínas que se encuentran en el mismo compartimiento celular que los interactores. En los pasos 3 y 4 se obtiene el compartimiento celular de una proteína, este se calcula buscando un conjunto de palabras claves dentro de la anotación de la proteína. Para simplificar el pseudocódigo se asume que todas las proteínas tienen anotado su compartimiento, ya que esto dista bastante de la realidad y para contemplar estas situaciones, se ha implementado el tratamiento de desconocidos. En la implementación del algoritmo se puede especificar qué sucede al encontrar proteínas que no informan el compartimiento. Una posibilidad es descartarlas y la otra consiste en omitir este análisis y asumir que pueden interactuar con cualquier otra proteína (como si no se realizara este análisis).

La implementación también considera el caso espacial del compartimiento membrana, u otros que se definan que pueden interactuar con cualquier proteína.

iii. Analizando perfil hidrofóbico

InteraccionesPosibles (protIzq, protDer, TamañoVentana, TablaPerfilHidrofóbico)

Entrada:
ProtIzq, ProtDer lista de proteínas
TamañoVentana para el cálculo del perfil hidrofóbico
TablaPerfilHidrofóbico tabla parámetro para el cálculo del perfil

Salida:
Cantidad de proteínas que podrían interactuar ya que el perfil hidrofóbico de los
dominios no repelen el agua

1. posiblesInteractuantesIzq = CantidadNoHidrofóbico (protIzq,TamañoVentana, TablaPerfilHidrofóbico)
2. posiblesInteractuantesDer = CantidadNoHidrofóbico (protDer, TamañoVentana, TablaPerfilHidrofóbico)
3. **devolver** posiblesInteractuantesIzq * posiblesInteractuantesDer

Esta función analizará cada una de las proteínas (que contienen los dominios de la regla analizada) contando cuántas de ellas tendrán posibilidades de interactuar ya que su perfil hidrofóbico no impone restricciones para hacerlo.

CantidadNoHidrofóbico (Proteínas, TamañoVentana, TablaPerfilHidrofóbico, dominiosAnalizados. Porcentaje)

Entrada:
Proteínas lista de proteínas
TamañoVentana para el cálculo del perfil hidrofóbico
TablaPerfilHidrofóbico tabla parámetro para el cálculo del perfil
LímiteHidrofóbico si el valor del perfil es superior a este número se considera
que la parte de la proteína analizada es hidrofóbica
DominiosAnalizados dominios de la regla que se están analizando
Porcentaje = porcentaje del dominios que debe estar abajo del LímiteHidrofóbico

Salida:
Cantidad de proteínas pasadas como parámetro que no son hidrofóbicas (que no repelen el agua)

```
1.  cantidad = 0
2.  Para cada proteína en Proteínas
3.      pi = proteína.obtenerPerfilHidrofóbico
4.      encontré = verdadero
5.      Para cada dominio en DominiosAnalizados y mientras encontré
6.          Para cada detalleDeDominio en dominio
7.              p = pi.CalcularPorcentajeBajo (detalleDeDominio ,
8.                                              límiteHidrofóbico )
9.              Si (p > porcentaje) entonces
10.                 encontré = falso
11.             Fin Si
12.         Fin Para
13.     Fin Para
14.     Si encontré entonces
15.         cantidad = cantidad + 1
16.     Fin Si
17. Fin Para

18. devolver cantidad
```

Este algoritmo analizará el perfil hidrofóbico de la proteína en las que se encuentran los dominios de la regla que se está analizando. Dado que cada dominio puede aparecer dentro de la proteína más de una vez, es necesario hacer referencia al detalleDeDominio (línea 11) que representa cada una de las apariciones del mismo y por lo tanto se buscará que al menos una ocurrencia del dominio en la proteína no tenga un perfil hidrofóbico.

Cálculo de orden

Como se puede apreciar en el pseudocódigo, existen funciones que no son primitivas y consideraremos un orden para su cálculo. Estas funciones pertenecen

a una parte de la aplicación desarrollada para disponer de un método eficiente de acceso a los datos y consultas frecuentes. Algunas de estas funciones son obtener todas las proteínas que contienen un conjunto de dominios, obtener las interacciones que tienen cierta cantidad de dominios, etc.
Vale aclarar que esta implementación fue utilizada para el cálculo de todos los algoritmos en pos de mejorar al máximo su rendimiento y serán consideradas en el estudio del orden de los algoritmos.

A continuación realizaremos el estudio del orden de complejidad del algoritmo basado en reglas tomando como base el pseudocódigo presentado. Utilizaremos el término $DominioMax^2$ para indicar la mayor cantidad de dominios que tiene una proteína.

- El paso 1 tiene O(1)
- Los ciclos de las líneas 2 y 3 iteran $O(DominioMax^2)$ veces
- La línea 4 requiere O(#interacciones) pasos
- El paso 5 requiere O(#interacciones) pasos
- Los pasos 6 y 7 tienen O(1)
- El paso 8 requiere 2 * O(#dominios) * O(#proteínas) operaciones + O(#interacciones) operaciones
- Los pasos 9 y 10 tienen un costo de O(#interacciones) * O(#proteínas) + O(#proteínas)
- El paso 11 tiene O(1) si no se utiliza información adicional de las proteínas. Más adelante se analizará el orden de incluir estas operaciones
- Los pasos 12, 13, 14 tienen O(1)
- El paso 15 requiere O(#interacciones) pasos

Orden total:

$DominioMax^2$ * [O(#interacciones) +
O(#interacciones) * (
O(2 * #dominios * #proteínas + #interacciones) +
O(#interacciones * #proteínas + #proteínas) +
O(#interacciones))]

= $O(DominioMax^2)$ * $O(\#interacciones^2)$ * O(#proteínas (#dominios + #proteínas))

Capítulo 6 - Implementación

Arquitectura de la aplicación

La implementación fue desarrollada íntegramente en lenguaje Java utilizando la versión 1.4.1 de Sun [java]. Se utilizó la versión 4.0.12 de la base de datos MySql [mysql]. Además, para facilitar las pruebas de los algoritmos, se desarrolló un pequeño front end con tecnología Web que permitió ejecutar los diferentes algoritmos con los diferentes juegos de datos. La interfaz de pruebas ejecuta en un servidor de web Apache Tomcat 4.1 [tomcat] y se utilizó Apache Cocoon 2.0.3 [cocoon] como generador de contenidos. Tanto la base de datos como las herramientas de Apache gozan de licencia Open Source y java, si bien no es un producto con código abierto, se distribuye con una licencia gratuita para el desarrollo de aplicaciones.
El desarrollo de la aplicación fue realizado sobre una plataforma Windows, no obstante, el uso de Java, Tomcat y Cocoon (que están desarrollados en Java) y la disponibilidad de MySql en distintos sistemas operativos, permiten instalar y ejecutar la aplicación en forma directa en otras plataformas como ser Linux, Unix o MacOs y lograr una escalabilidad acorde a las necesidades.
El desarrollo fue dividido en etapas que describimos a continuación.

Estandarización de datos

Al comienzo de la implementación nos encontramos con una gran cantidad de datos de proteínas, dominios e interacciones de distintas bases de datos y en diferentes formatos. Para manejar esta información fue necesario diseñar un modelo de datos estándar para unificar la información que permitiera ejecutar los diferentes algoritmos independientemente del origen de los datos. Fue así que modelamos una base de datos que permitiera almacenar toda la información utilizada y luego generamos distintos componentes específicos para procesar los datos y tener una carga inicial del modelo. El detalle de los datos cargados es:

- Proteínas y Nucleótidos
  - Base NR del NCBI para obtener las proteínas de Levadura y T. cruzi en formato fasta
  - Bases de TIGR provistas por el INGEBI donde obtuvimos secuencias de nucleótidos recientes (no disponibles en otras bases) en formato fasta

- Interacciones
  - Base de datos de interacciones de BIND en formato XML
  - Base de datos de interacciones de DIP en formato XML (diferente al de BIND)
- Dominios

  La base de dominios utilizada es PFAM FS [pfam] y se usaron dos programas para realizar la búsqueda de dominios. HmmSearch de Pfam [hmmer] y MAST [mast] que fue utilizado para obtener los dominios de las secuencias de nucleótidos de TIGR en la primera etapa del trabajo ya que la utilidad HmmSearch sólo permite buscar dominios en proteínas y solo se contaba con nucleótidos para T. cruzi.
  - Archivos de dominios obtenidos con HmmSearch
  - Archivos de dominios obtenidos con MAST

El modelo de datos resultante tiene el siguiente esquema físico.

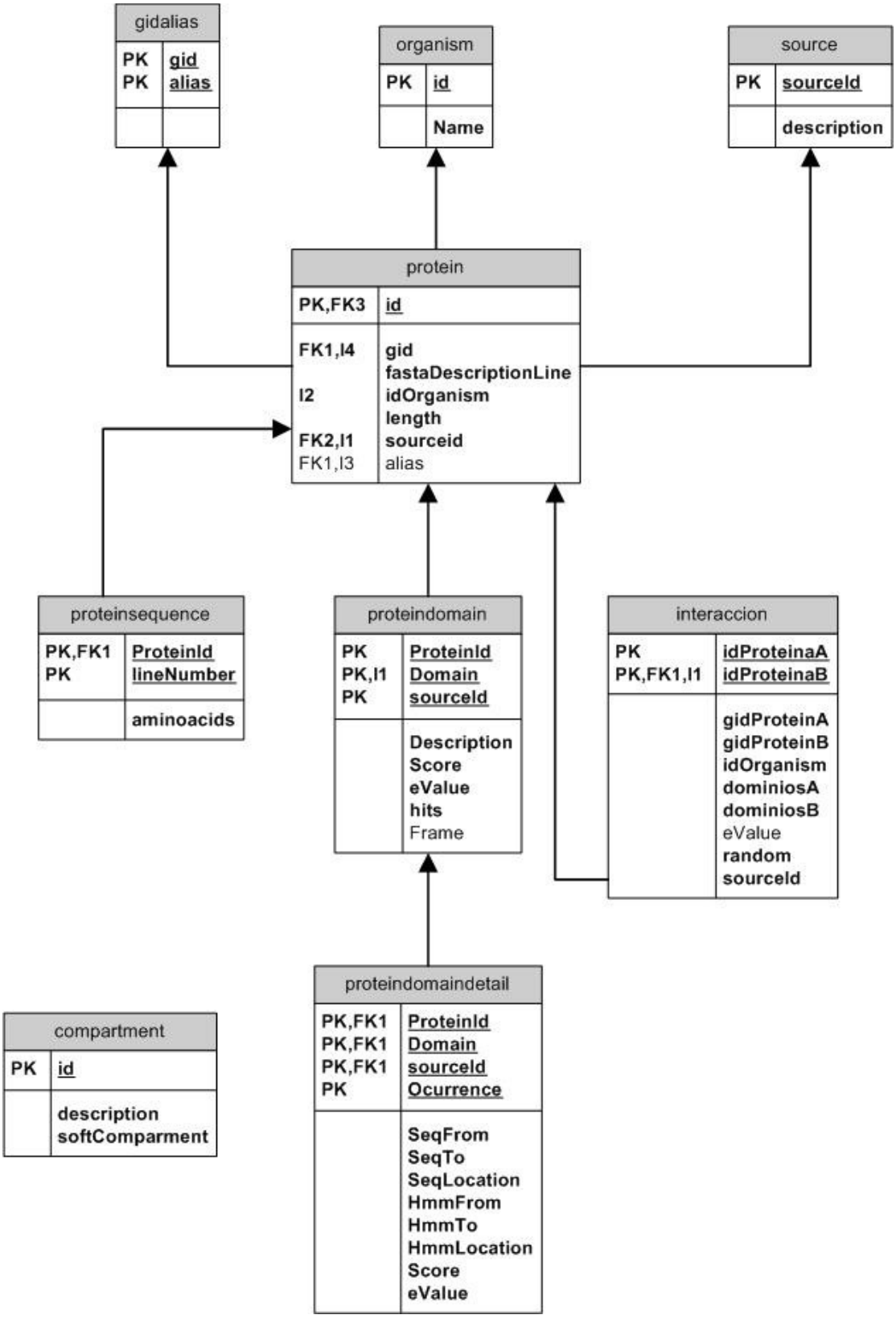

Figura 6: Esquema físico del Modelo de Datos

Modelado del problema

Para comenzar el desarrollo en Java modelamos el problema con objetos. Las clases principales del modelo se presentan en el siguiente esquema.

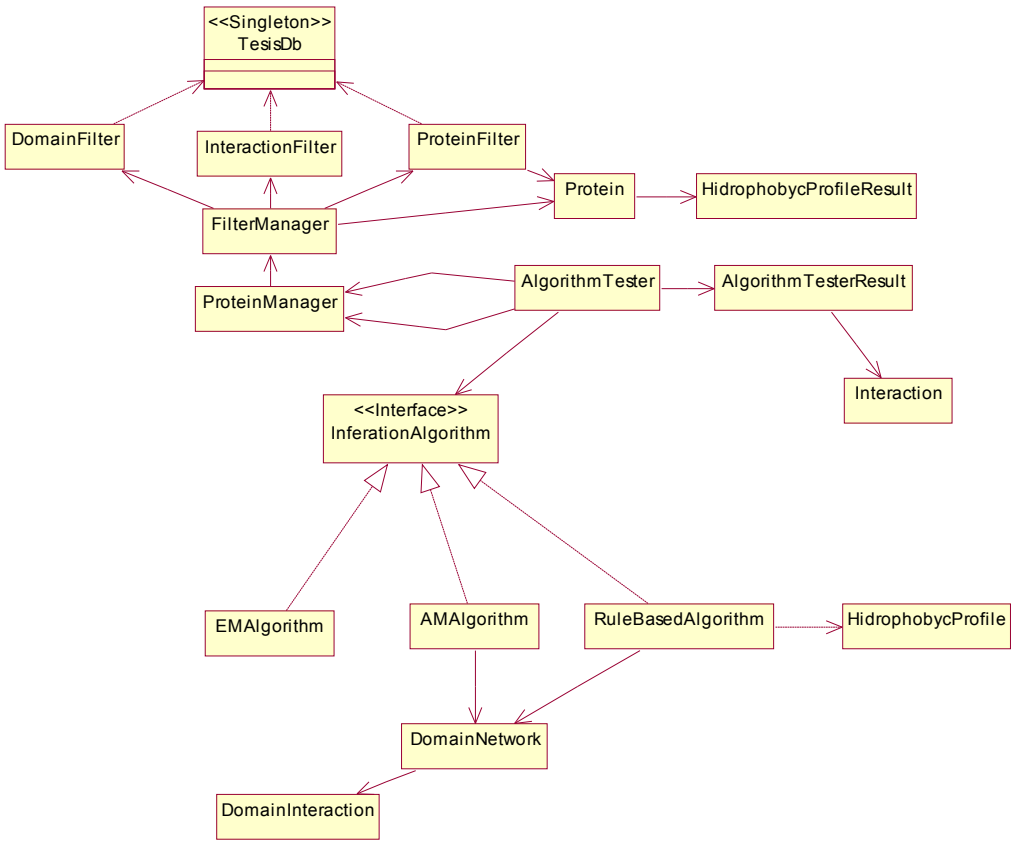

Figura 7: Principales Clases del Modelo

El modelo realizado nos permitió realizar un framework de trabajo para todos los algoritmos. Por un lado, se desarrollaron parsers para procesar los distintos orígenes de datos.

Para el caso de las proteínas fue necesario procesar la base NR en formato fasta obtenida del NCBI. Ésta es una base de proteínas unificada y presenta una codificación para las proteínas que se encuentran en distintas bases de datos. Para obtener toda la información del archivo, fue necesario identificar las proteínas similares de otras bases y contemplarlas en nuestro modelo de datos,

ya que la información disponible de interacciones hacía referencia a proteínas de cualquier base y de otra forma no hubieran sido encontradas.
Respecto a las interacciones, las bases de Bind y Dip se encontraban en formato XML y cada una utilizaba un esquema propio. Para unificar la información se realizaron parsers que procesaran cada una de las bases y las interacciones se almacenaron en un repositorio común.
Con respecto a los dominios, fue necesario desarrollar parsers para procesar las salidas dejadas por lo programas Mast y Hmmer. Al igual que sucedió con las interacciones, las salidas de ambos algoritmos eran diferentes y fue necesario desarrollar los parsers a medida y se cargó la información de dominios unificada en la base de datos.
Vale aclarar que los modelos de datos son lo suficientemente genéricos como para incluir nuevas fuentes de datos desarrollando los parsers adecuados y con un mínimo esfuerzo.

Selección de datos y consultas básicas

Al disponer de un modelo unificado para todos los datos con los que trabajamos fue necesario implementar clases que permitieran generar conjuntos de datos de entrenamiento y test a partir del conjunto de datos unificados. De esta forma podemos entrenar los algoritmos con las proteínas de levadura y testear con T. cruzi, entrenar y testear con levadura o entrenar con el 75% de las proteínas de levadura y testear con el 25% restante, seleccionar distintos organismos, distintas bases de datos, etc.
Todas estas características fueron implementadas en clases que llamamos filtros. Para ello se especifica cuáles serán los parámetros a utilizar en el conjunto de entrenamiento y test de proteínas, interacciones y dominios. Con estos tres filtros, generamos el conjunto de datos sobre el cual ejecutarán los algoritmos. Las clases que implementan los filtros son FilterManager, ProteinFilter, DomainFilter e InteractionFilter.
El gran volumen de datos, y el hecho de que estuvieran almacenados en una base de datos, provocó que nuestras primeras implementaciones de los algoritmos no fueran eficientes, ya que constantemente realizaban consultas a la base de datos. Para solucionar este problema decidimos utilizar buffers y limitar los accesos a la base sólo para la carga inicial y luego trabajar con los datos en memoria.

Además fue necesario construir una estructura adecuada para almacenar los datos en memoria para hacer un uso eficiente de la misma y responder las consultas más frecuentes en forma óptima. Algunos de las más utilizadas son:

- Obtener todas las proteínas que contienen un conjunto de dominios.
- Obtener todas las interacciones cuyas proteínas contienen una cantidad d_1 y d_2 de dominios.

La implementación de estas consultas se encuentra en la clase ProteinManager.

Algoritmos

Con las clases mencionadas hasta el momento ya disponíamos de un framework para escribir algoritmos, teníamos los datos, una forma eficiente de tomar conjuntos de datos desde una base unificada y un método eficiente para manipularlos.

La implementación de los algoritmos fue realizada de manera tal que fueran independiente del conjunto de datos que estuvieran utilizando. Además se buscaron características similares entre los procedimientos para ejecutar un algoritmo y se realizó un modelo abstracto con las operaciones que realiza cada algoritmo. Como resultado se generó una interfaz llamada InferationAlgorithm, que describe todas las operaciones básicas que debe implementar un algoritmo de inferencia. Estas operaciones son build (crear un modelo), posibleInteractions (devolver un vector de posibles interacciones inferidas) y getModelProperties (obtener las propiedades del modelo generado). El uso de este modelo abstracto simplificó el desarrollo de la aplicación y nos permitirá agregar nuevos algoritmos de inferencia, con un esfuerzo moderado.

El resultado de la corrida de un algoritmo es un conjunto de reglas del tipo {A,B,C} -> {X,Y} para el Algoritmos Basado en Reglas y el Método de Asociación. En el caso del algoritmo Maximización de Esperanza, el resultado es una matriz de probabilidades donde la posición i,j indica la probabilidad que los dominios I y J interactúen.

Por último, fue necesario desarrollar un esquema para probar estos algoritmos para un conjunto de datos con uno o varios parámetros y obtener sus resultados. De esta forma surgieron las clases AlgorithmTester y BatchAlgorithmTester.

Todos los algoritmos corrían desde la línea de comando y no había forma de cambiar los parámetros en entrada, lo cual implicaba tener que realizar cambios en el código y recompilar la aplicación cada vez que queríamos realizar una prueba.

Con el fin de eliminar estos problemas y tener una aplicación más amigable, desarrollamos un front-end Web que permitiera customizar la carga de datos para distintas ejecuciones. El uso de Cocoon para el desarrollo del front-end permitió reutilizar el modelo de objetos realizado en Java e ingresar parámetros de los algoritmos utilizando este front end.

Capítulo 7 - Estudio de los algoritmos

En este capítulo realizaremos un análisis de los algoritmos en base a estas características:

- Precisión del los datos inferidos
- Poder de expresión de los algoritmos

Las mismas serán analizadas ejecutando los algoritmos seleccionando umbrales entre 0 y 1 a intervalos de 0.05 utilizando los conjuntos de entrenamiento disponibles para levadura y testeando contra el mismo conjunto.

Notación

Se empleara la siguiente notación para diferenciar las distintas corridas

AA – ABR – [Opciones] = Algoritmo Basado en Reglas

AA - MA = Método de Asociación

AA - ME - λ = Maximización de la Esperanza

λ = número de iteraciones realizadas para obtener la matriz.

AA = Año del conjunto de datos utilizado

[Opciones] = opciones o parámetros especiales utilizados en el algoritmo.

Ej: Compartimientos

PH – 50 (Perfil Hidrofóbico – Límite de 50%)

Definiciones

Para realizar el análisis de los algoritmos vamos a definir los siguientes términos

$I_o = \{o_1, o_2, \ldots, o_n\}$ conjunto de interacciones observadas

$I_i = \{i_1, i_2, \ldots, i_m\}$ conjunto de interacciones inferidas

$M(I_o, I_i) = I_o \cap I_i$ conjunto de interacciones inferidas correctamente

- *Especificidad:* es la proporción de interacciones inferidas correctamente sobre el total de interacciones predichas.

$$\mathrm{E} = \frac{\#M(I_o, I_i)}{\#I_i}$$

Este indicador medirá la cantidad de falsos positivos que genera el algoritmo, es decir, las interacciones inferidas que no se verifican en el conjunto de entrenamiento. Un valor cercano a cero indicará mayor cantidad de falsos positivos.

- *Sensitividad:* es la proporción del número de interacciones inferidas correctamente sobre el total de interacciones observadas.

$$\mathrm{S} = \frac{\#M(I_o, I_i)}{\#I_o}$$

Este indicador analiza la cantidad de reglas que el algoritmo predice en base al conjunto de entrenamiento. En este caso el valor de S indica el porcentaje que cubrimiento que tienen las interacciones inferidas sobre el conjunto de entrenamiento.

- *Tamaño del modelo (T)*: En el caso del Método de Asociación y el Algoritmo Basado en Reglas el tamaño del modelo esta definido por la cantidad de interacciones dominio-dominio generadas. La variación de los parámetros de entrada de los algoritmos produce que los modelos construidos sean de tamaño diferente. El algoritmo ME utiliza una matriz para representar el modelo de interacciones dominio-dominio, el tamaño en este caso es fijo.

- *Poder de Expresión*: es la proporción entre el tamaño del modelo y la cantidad de interacciones inferidas correctamente.

$$\mathrm{PE} = \frac{\#M(I_o, I_i)}{T}$$

Este es un indicador de sobre ajuste, el valor PE indica la cantidad de interacciones promedio que infiere cada regla. Valores cercanos a uno indican sobre ajuste, ya que cada interacción se infiere con una regla. A medida que el valor de PE aumenta, las reglas generadas son más genéricas.
Evaluar la precisión del modelo generado a nivel de dominio (o sea, evaluar las interacciones dominio-dominio) para determinar si son correctas o no es muy dificil debido al escaso conocimiento que se tiene sobre este tipo de relación.

Debido a esta dificultad realizaremos un estudio sobre las interacciones proteína-proteína generadas a partir de las interacciones dominio-dominio.

En este estudio utilizaremos datos de *Saccharomyces Cerevisiae* (Levadura) para evaluar los algoritmos de la siguiente manera:

1) A partir de las proteínas, dominios e interacciones proteína-proteína conocidas de Levadura se construye el modelo de interacciones dominio-dominio.
2) Luego, se infieren las interacciones proteína-proteína para Levadura utilizando solamente el modelo generado.
3) Por último se comparan las interacciones inferidas con las conocidas, y así poder evaluar la precisión del modelo.

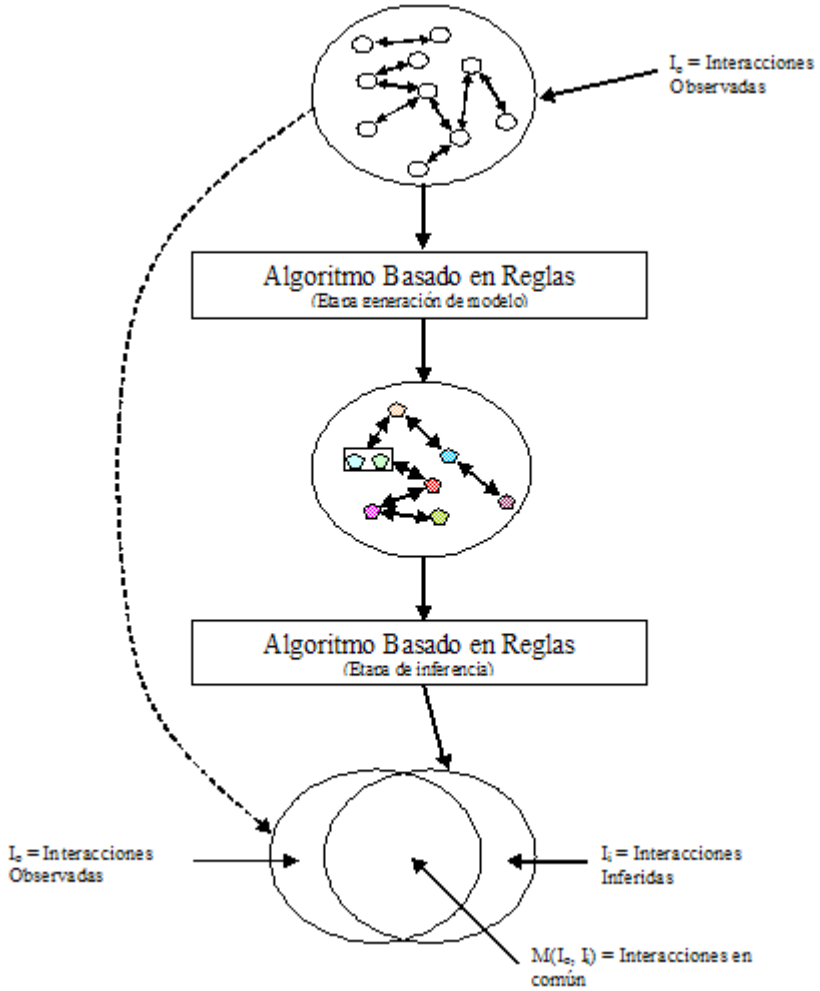

Figura 8: Procedimiento para el estudio de los algoritmos

Los datos a utilizar son los siguientes:

Saccharomyces Cerevisiae		
	Fuente	**Cantidad**
Proteínas	NCBI	8428
	BIND	76
	DIP	2434
Interacciones	BIND	5044
	DIP	10784
Dominios	PFAM	7258

Saccharomyces Cerevisiae	
	Cantidad
Proteínas con Dominios	8081 (%73 del total)
Interacciones entre proteínas con dominios	11153 (%70 del total)

Comparación entre algoritmos

Este gráfico muestra los resultados de los diferentes algoritmos con respecto a la sensitividad y especifidad cuando se varía el umbral.

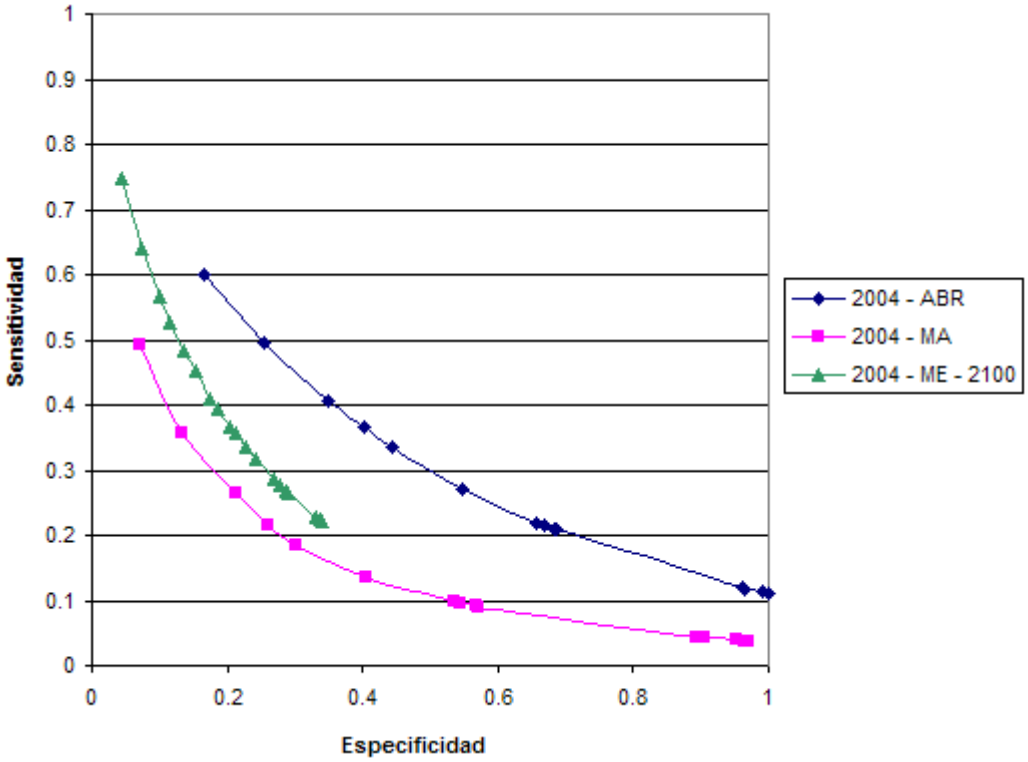

Gráfico 1: Comparación de sensitividad y especificidad entre algoritmos

Se puede apreciar una baja especificidad y sensitividad de los tres algoritmos. Los siguientes puntos identifican las principales causas que generan este comportamiento.

- Los modelo usados simplifican la complejidad del problema real y no se consideran ciertos factores, como ser :

  - *Interacciones a traves de dominios desconocidos*: la base de estos algoritmos es considerar que una porción de la proteína, con una estructura especial, es la responsable de la interacción. Estas porciones pueden no coincidir con los dominios definidos de PFAM obtenidos por alineamiento múltiple.

  - *Independencia de las interacciones dominio-dominio*: el ME y MA suponen en sus modelos que las interacciones dominio-dominio son independientes. Esto no es cierto, ya que la interacción de dos dominios puede depender de otros dominios en la misma proteína o de factores del entorno.
    - ✓ Nuestro algoritmo no asume dicha restriccion y considera interacciones multi-dominio – multi-dominio.

  - *Limitaciones espacio-tiempo*: es conocido que las interacciones proteína-proteína poseen restricciones de tiempo y espacio. Esto significa que dos proteínas que contienen dominios que potencialmente interactúan pueden no interactuar entre sí, ya que se expresan en diferentes momentos durante el ciclo de vida de la célula, o porque estan ubicadas en diferentes compartimientos.
    - ✓ El algoritmo propuesto considera las limitaciones de espacio evaluando, cuando se dispone de la informacion, los compartimientos en donde se ubican las proteínas.

  - *Datos faltantes*:
    - El conjunto de datos conocidos de interacciones proteína-proteína es una fracción de la red de interacciones proteína-proteína. La falta de información genera dificultades para derivar una red de interacciones dominio-dominio.
    - Esta falta de informacion se puede apreciar al analizar que los

datos de diferentes fuentes (Ej.: [BIND][DIP]) poseen poco solapamiento.

Algoritmo Basado en Reglas

En este algoritmo podemos ver la variación de los resultados utilizando las alternativas de evaluar los compartimientos celulares y el perfil Hidrofófico de la proteína.

El siguiente gráfico muestra los resultados utilizando los compartimientos celulares

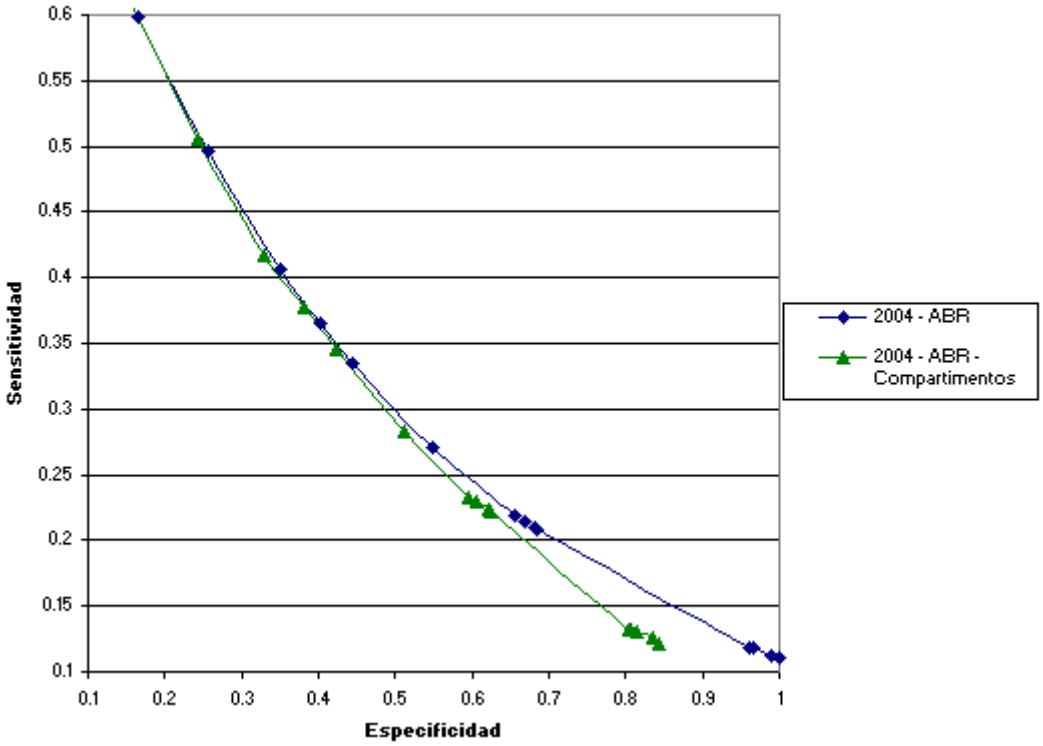

Gráfico 2: Algoritmo Basado en Reglas utilizando Compartimientos Celulares

Recordemos que dos proteínas pueden interactuar si pertenecen al mismo compartimiento (Existen excepciones a esta regla, ver Apéndice I).
Lamentablemente sólo una pequeña porción de las proteínas tienen indicado el compartimiento celular (alrededor del veinte porciento). Debido a esto, los resultados aparentemente no son mejores. Más adelante analizaremos el *nivel de expresión* del modelo donde se podrá apreciar el beneficio de utilizar compartimientos.

Al analizar el perfil hidrofóbico, se observó un aumento de la sensibilidad y una disminución en la especificidad. Esto se debe a que el algoritmo descarta muchas posibles interacciones dentro de la función *PosiblesInteracciones*, ya que hay una gran cantidad que no cumplen los valores requeridos del perfil hidrofóbico. Esta disminución en la cantidad de posibles proteínas genera que en la ecuación de la línea 12 del ABR calcula un valor alto para la *confiabilidad*. Al agregar un mayor número de reglas al modelo se infiere más cantidad de interacciones pero con menos precisión. Los biólogos determinarán si este tipo de estudio les será de utilidad.

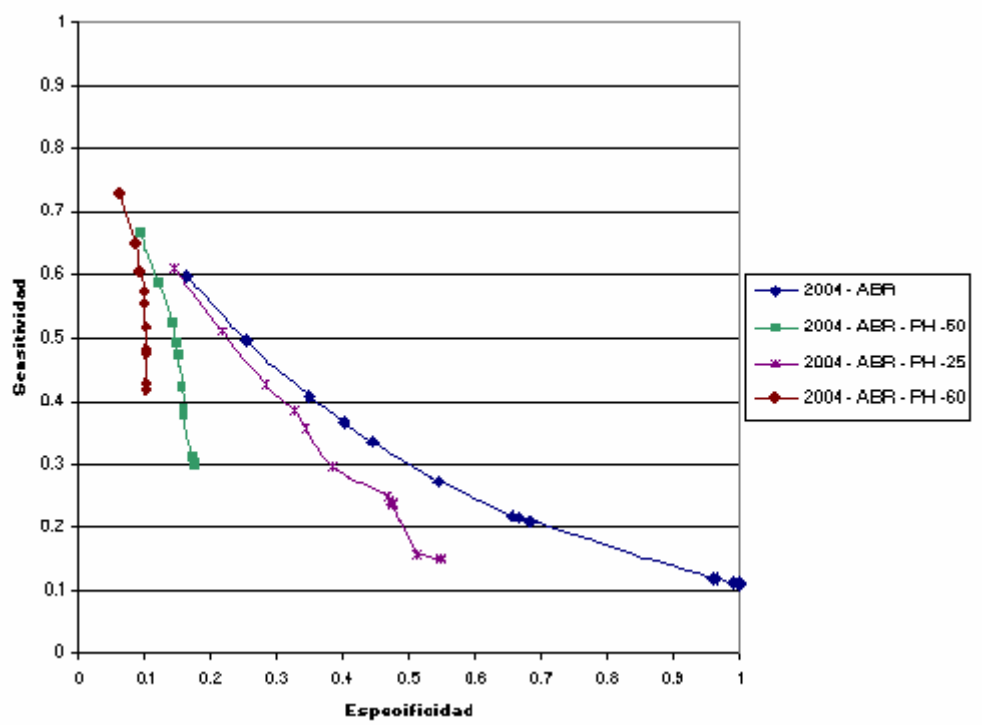

Gráfico 3: Algoritmo Basado en Reglas utilizando Perfil Hidrofóbico

Algoritmo ME

Uno de los principales problemas que tiene el algoritmo ME es la lentitud en converger a resultados precisos. En el siguiente gráfico se puede observar la mejora en la especificidad a medida que se ejecutan las iteraciones. Cada iteracion consume aproximadamente 4 minutos[1] y por lo tanto es necesario dejar correr varias horas o días para que converja. Otros de los inconvenientes que

1 Computadora de referencia : PC Intel Pentium III de 500Mhz. con 512Mb. de memoria Ram.

tiene el algoritmo es que la velocidad de convergencia esta determinada por la matriz inicial de interacciones dominio-dominio (que es elegida arbitrariamente), incrementando sustancialmente el tiempo de corrida si se elige una matriz que difiera significativamente con la matriz real. Como esta matriz no se conoce de antemano se hace necesario disponer de alguna estimacion previa. Tambien es dificultoso definir el ΔL (diferencia de la funcion de esperanza L entre dos interacciones) que determina la finalizacion del algoritmo por lo que es necesario agregar otro criterio de corte como cantidad de iteraciones . En el siguiente grafico se aprecia la velocidad de convergencia del algoritmo, este necesito por lo menos cien iteraciones para estabilizarse. En este caso la convergencia fue veloz ya que se utilizo como matriz inicial λ la iteracion 2300 calculada previamente con los datos del 2003 (ver apéndice II)

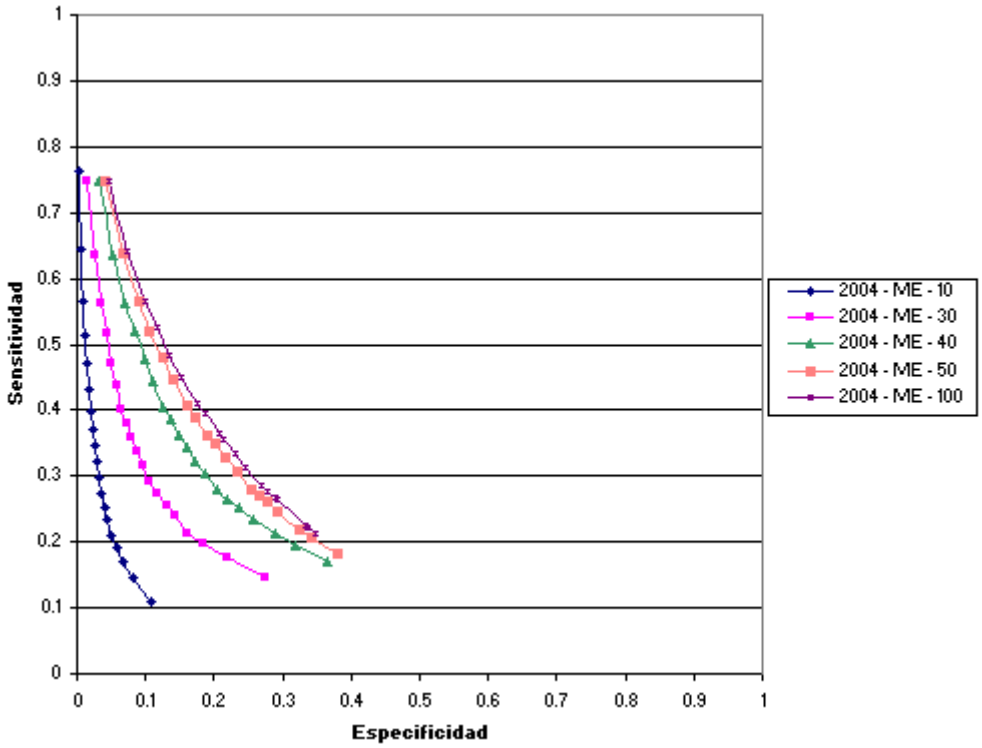

Gráfico 4: Convergencia del algoritmo ME

Comparación del poder de expresión

Una caracteristica deseada es generar un modelo que sea lo más expresivo posible. Por expresivo se entiende tratar de minimizar el tamaño del modelo sin perder poder de inferencia. En nuestro caso se pueden representar las

interacciones inferidas dominio-dominio como el tamaño del modelo

Cuando el poder de expresion esta cercano a 1 indica que cada inferencia es proviene de una regla (interacción dominio-dominio) del modelo

Cuanto mayor sea el nivel de expresión, el modelo será más genérico. Como mencionamos anteriormente el modelo en ME tiene un tamaño fijo y por lo tanto su poder de expresion no varia. Veremos como el Algoritmo Basado en Reglas supera al Método de Asociacion

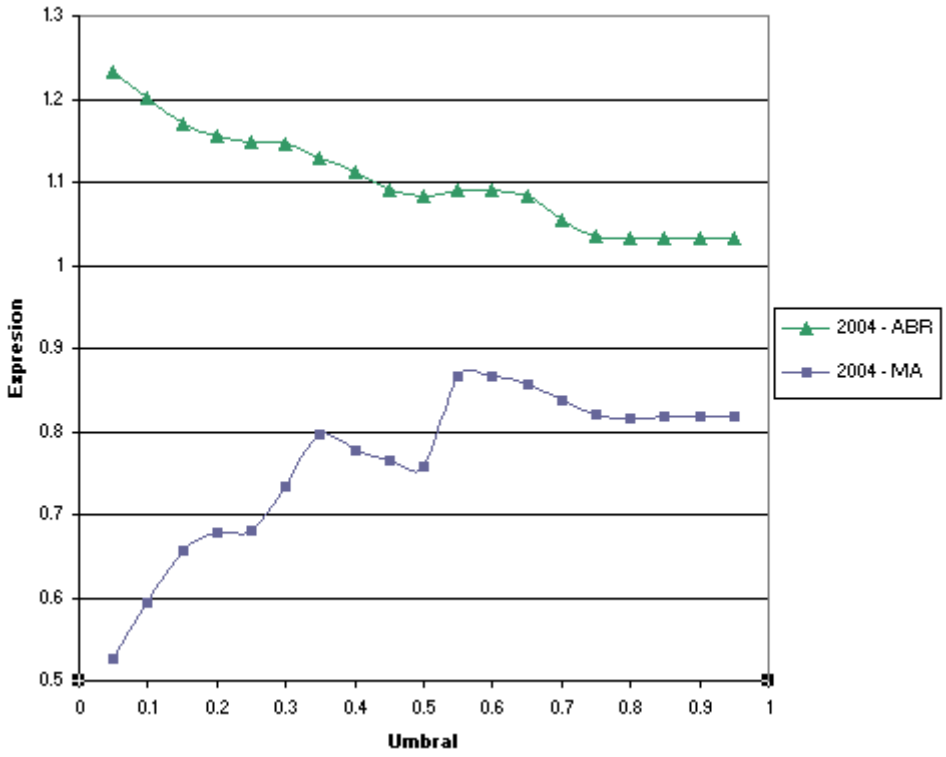

Gráfico 5: Comparación del poder de expresión entre en ABR y el MA

La diferencia que observamos se poduce porque el AM trabaja únicamente con interacciones mono-dominio que suelen no ser apropiadas para definir correctamente una interacccion dominio-dominio. El algoritmo propuesto contempla la existencia de reglas multi-dominio mejorando la precision de los datos inferencia.

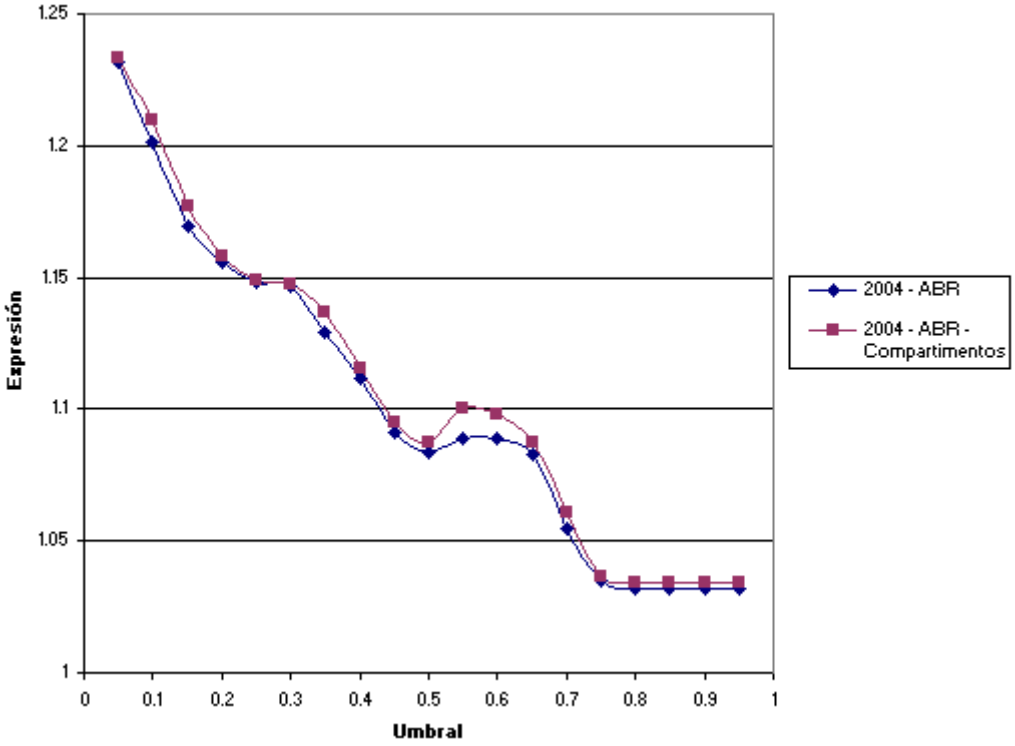

Gráfico 6: Poder de expresión del ABR utilizando compartimientos

Se nota una pequeña mejora al considerar los compartimientos de cada proteína. La mejora es mínima porque no se puede aplicar en forma generalizada debido a que hay pocas proteínas cuya descripcion indica el compartimiento celular donde se encuentra alojada pero igualmente notamos un tendencia a mejorar el poder de expresión

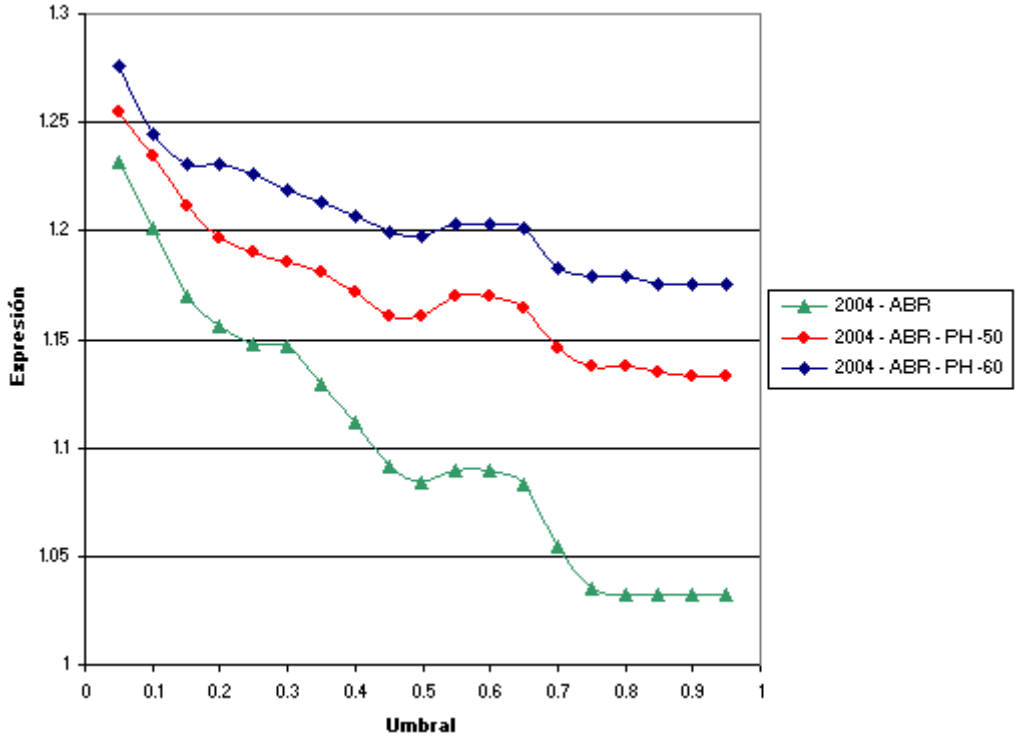

Gráfico 7: Poder de expresión del ABR utilizando Perfil Hidrofóbico

Comparación entre el Algoritmo Basado en Reglas y el resto de los algoritmos

Supongamos que existe una interacción (comprobada biologicamente) entre los siguientes dominios

$$\{A,B\} \leftrightarrow \{C\}$$

Por lo tanto si una proteína contiene el dominio C y otra contiene los dominios A y B, entonces podríamos afirmar que deberían interactuar. Cuando se analiza el conjunto de interacciones deberían aparecer varias proteínas con contengan esos dominios, por ejemplo:

Proteina1: $\{A,B\} \leftrightarrow \{C\}$

Proteina2: $\{A,B\} \leftrightarrow \{C\}$

Proteina3: $\{A,B\} \leftrightarrow \{C\}$

Proteina4: $\{A,B\} \leftrightarrow \{C\}$

El MA al no manejar interacciones multi-dominio generará dos reglas, una para {A}↔{C} y otra para {B}↔{C}. Inferir de esta forma genera las siguientes situaciones que disminuyen el poder expresión y además llevan a resultados incorrectos.
Por otro lado el método de maximización de experanza no tiene posibilida de asociar ocurrencias de dominos. Recordar que la matriz λ del método ME contiene la probabilidad que dos dominios interactúen. En este caso se dará como probable la interaccion entre los dominios {A}↔{C} y {B}↔{C} cuando en realidad la interaccion con C depende de A y B

En resumen,

- **Inferir interacciones incorrectas**: Inferir interacciones en proteínas que sólo contengan el dominio A ó el domino B es incorrecto ya que la interacción se produce únicamente cuando los dominios se encuentran en forma simultánea.

- **Generar reglas excesivas en el modelo (solo en MA)**: Por cada interacción multi-dominio se deben generar varias reglas para poder representarla (con las desventajas ya mencionadas) mientras que en el ABR sólo es necesaria una.

Resumen de ventajas y desventajas de cada método.

Método	Ventajas	Desventajas
Método de Asociación	- Es un algoritmo simple y veloz.	- Asume que las interacciones se realizan únicamente entre un dominio de cada proteína (no contempla interacciones multi-dominio). - Asume independencia en las interacciones dominio-dominio. - Genera excesivas reglas - No contempla datos faltantes
Maximización de esperanza	- Implementa un método estadístico que contempla datos faltantes para determinar la probabilidad que dos dominios interactúen - Asigna una probabilidad a la interacción de dos dominios. - Permite asignar una probabilidad al hecho de que dos proteínas interactúen.	- Poco eficiente - Es un método iterativo que puede converger a máximos locales. - Asume independencia en las interacciones dominio-dominio.
Algoritmo basado en reglas	- Es un algoritmo eficiente - No asume independencia en las interacciones dominio-dominio. - Contempla interacciones multidominio - Analiza el compartimiento celular donde se encuentra la proteína. - Analiza el perfil hidrofóbico de una proteína.	- No contempla datos faltantes

Tabla 4: Ventajas y desventajas de cada método

Capítulo 8 - Aplicación: Interacciones proteína-proteína en Tripanosoma cruzi

Este trabajo fue desarrollado para ser aplicado al estudio del *Tripanosoma cruzi* (Parásito que causa el Mal de Chagas) que se está desarrollando en el INGEBI bajo la dirección del Dr. Mariano Levin. Durante el transcurso de esta tesis se produjeron diferentes avances en el secuenciamiento del genoma de *Tripanosoma cruzi*.

Objetivo:
Detectar posibles interacciones en *Tripanosoma cruzi* basadas en las interacciones de *Saccharomyces Cerevisiae* (Levadura).

Versión preliminar

Originalmente se utilizaron las secuencias aleatorias de nucleótidos de TIGR que provenían en forma directa del secuenciamientode T. cruzi. Estos fragmentos todavía no se habían ensamblado para formar un genoma. Esta base presentaba dos problemas, el primero es que la determinación de los dominios requiere una secuencia de aminoácidos y el segundo es que existía gran cantidad de secuencias repetidas.
La búsqueda de los dominios se realizó agregando un nivel más de procesamiento que consistió en traducir la secuencia de nucleótidos a las posibles proteínas y a partir de ellas determinar los dominios. Los resultados deben verificarse experimentalmente, ya que parte de estas posibles proteínas puedrían no existir en la realidad. No obstante tuvo la gran ventaja de estar analizando toda la secuencia de nucleótidos de T. cruzi sin estar disponibles sus proteínas.
El programa MAST [MAST] permite obtener los dominios a partir de una secuencia de nucleótidos. Internamente genera las posibles proteínas y a partir de estas determina los dominios.
MAST utiliza el modelo de representación de Matrices de Peso (ver introducción) mientras que PFAM utiliza la representación de Modelos Ocultos de Harkov. Debido a esto se tuvo que realizar una conversión de los modelos de PFAM (HMM) a MAST.

La base de TIGR original estaba compuesta por 612903 secuencias. La búsqueda de dominios requirió un elevado poder de procesamientos y se utilizó el cluster Beowulf de 16 nodos “Speedy Gonzales” [Speedy] de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires.

Trabajo final

Durante la etapa final de esta tesis TIGR consiguió ensamblar los fragmentos de las secuencias utilizando diferentes programas y técnicas. A partir de esto consiguieron predecir las posibles proteínas de T. cruzi. El avance logrado por TIGR permitió aplicar nuevamente nuestro trabajo para un número mucho menor y más preciso de secuencias (ahora proteínas) produciendo resultados más exactos.

- **Fuentes de datos**

  Durante la carga de datos se elimino la información duplicada. La siguientes tablas muestran únicamente los datos eliminando los duplicados.

Saccharomyces Cerevisiae		
Tipo	**Fuente**	**Cantidad**
Proteínas	NCBI	8428
	BIND	76
	DIP	2434
Interacciones	BIND	5044
	DIP	10784

Tripanosoma cruzi		
Tipo	**Fuente**	**Cantidad**
Proteínas Predichas	TIGR	23354

Dominios		
Dominios	PFAM	7258

Procedimiento

Comentaremos a continuacion el procedimiento realizado para la obtencion de la Red de Interacciones en T. cruzi.

1- Obtener los dominios de las proteínas de *Saccharomyces Cerevisiae* utilizando HMMER [HMMER]

2- Unificar las interacciones proteína-proteína de BIND y DIP de Levadura
3- Ejecutar el **Algoritmo Basado en Reglas** con los datos obtenidos en 1 y 2 para generar reglas de interacción dominio-dominio para *Saccharomyces Cerevisiae*
4- Obtener los dominios de las proteínas de *Tripanosoma cruzi* utilizando HMMER
5- Ejecutar el **Algoritmo Basado en Reglas** con los datos obtenidos de 3 y 4 para *inferir* interacciones Proteína-Proteína en *Tripanosoma cruzi*.

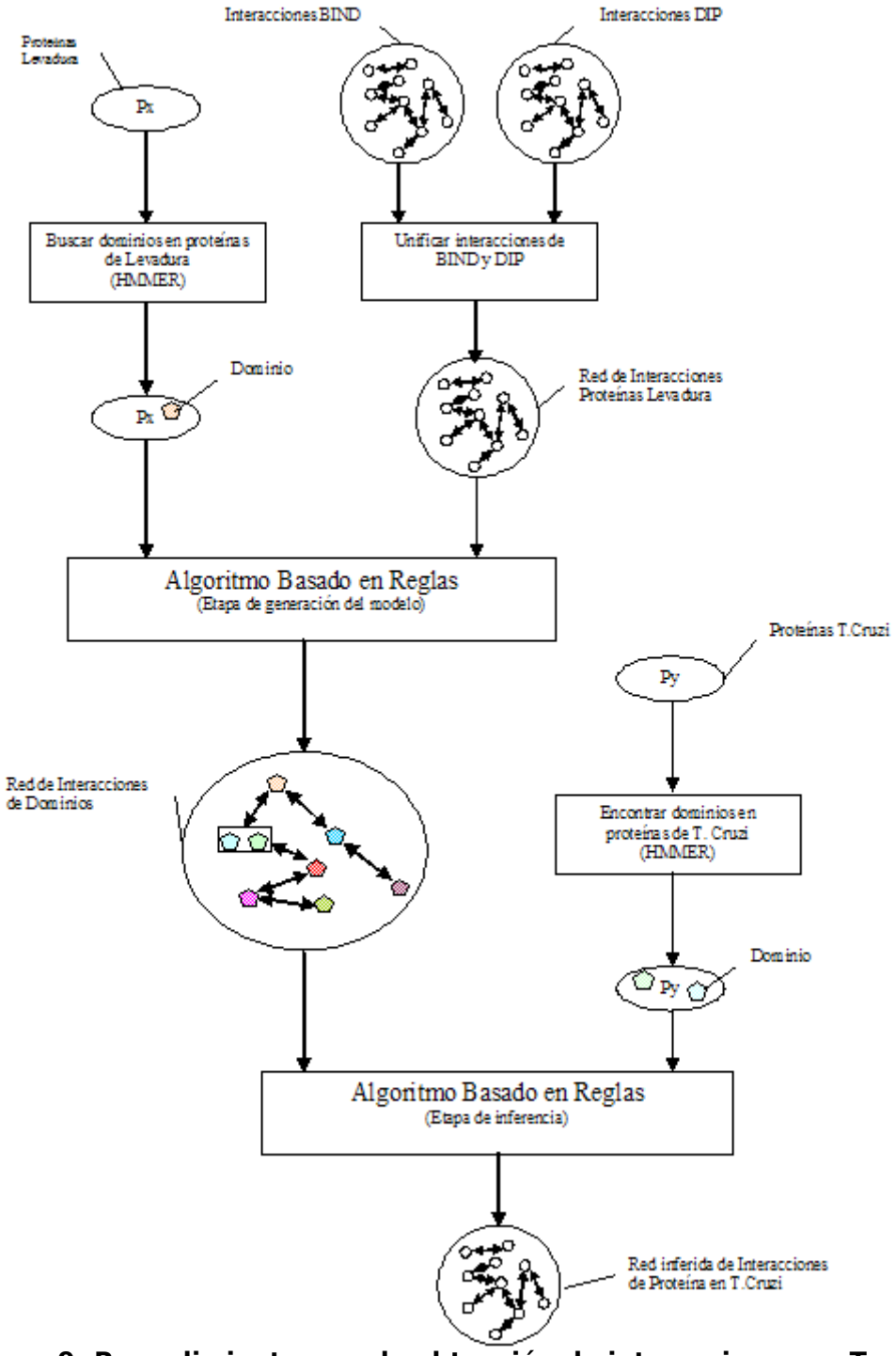

Figura 9: Procedimiento para la obtención de interacciones en T. cruzi

Interacciones y Dominios

Distribución de las interacciones conocidas de *Saccharomyces Cerevisiae* (Levadura)

El siguiente gráfico muestra la distribución de las interacciones de Levadura con respecto a la cantidad de dominios de las proteínas involucradas. Ej.: Dada una interacción dos proteínas

- Si las proteínas son mono-dominio la representaremos con 1-1.
- Si alguna de ellas posee más de un dominio la representaremos como 1-N donde N indica la cantidad de dominios.
- Si las dos proteínas son multi-dominio las representaremos como M-N con M y N cantidad de dominios que posee cada proteína respectivamente.

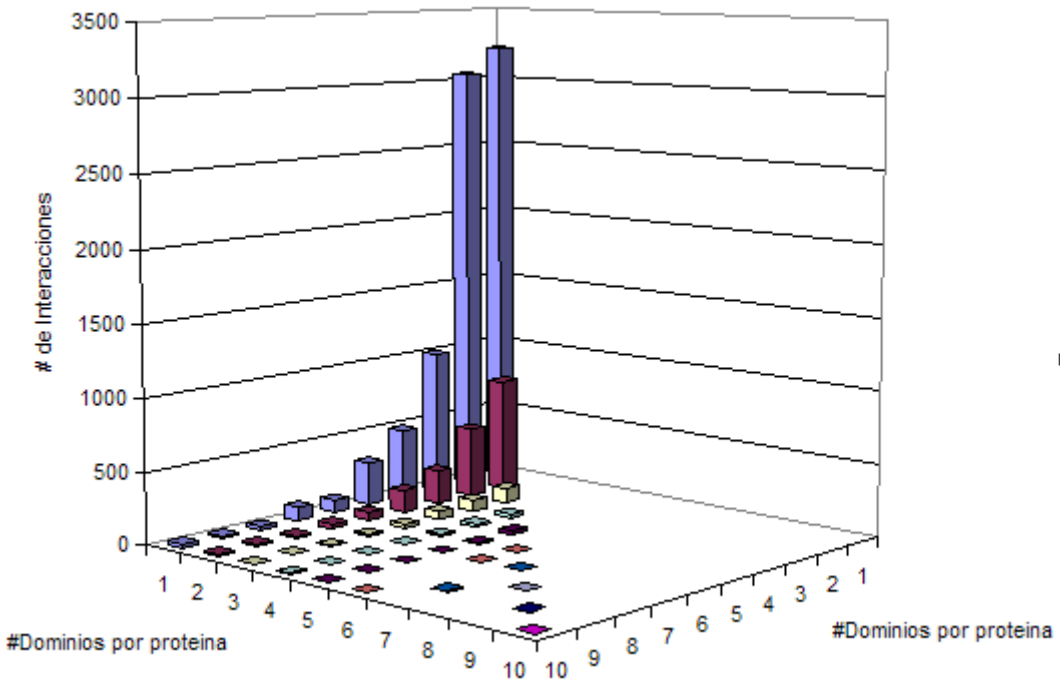

Gráfico 8: Distribución de interacciones por cantidad de dominios de Levadura

La mayor cantidad de interacción se da entre proteínas mono-dominio contra multi-dominios. Se destacan entre ellas las interacciones monodomino – monodominio y existen pocas interacciones multidominios – multidominios.

Resultados

El modelo generado

El siguiente gráfico muestra la distribución de las reglas generadas con el algoritmo propuesto.

Ej.: Dada una regla entre dos conjunto de 1 o más dominios

- Si los dos conjuntos poseen 1 domino cada uno lo representaremos con 1-1.
- Si uno de ellos posee más de un dominio lo representaremos como 1-N donde N indica la cantidad de dominios.
- Si los dos conjuntos poseen más de un dominio lo representaremos como M-N con M y N cantidad de dominios que poseen cada conjunto respectivamente.

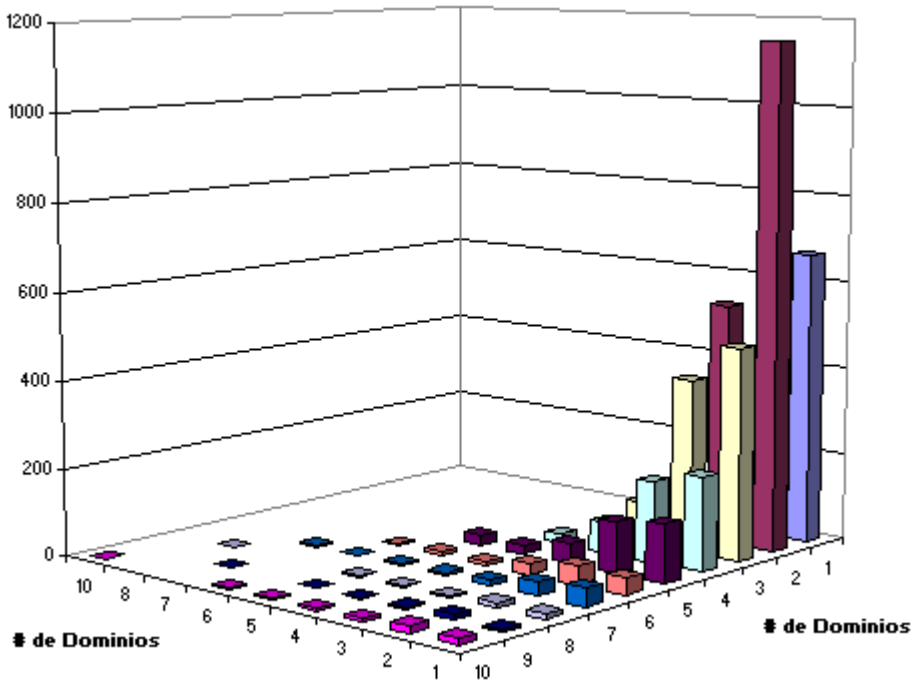

Gráfico 9: Distribución de la cantidad de dominios por regla en el modelo generado

Nota: en este gráfico sólo se consideran las reglas menores a 10 dominios

Interacciones inferidas

La siguiente tabla muestra un resumen de los resultados obtenidos.

Cantidad de dominios involucrados en las interacciones										
		#Dominios								
		1	2	3	4	5	6	7	10	Total
#Dominios	1	6145	2105	133	2	39		29	7	8460
	2		334	53	70	6	4	7		474
	3			3						3
	4								1	1
	5					3		2		5
Total		6145	2439	189	72	48	4	38	8	**8943**

Tabla 5: Tipo de interacciones inferidas en T.cruzi

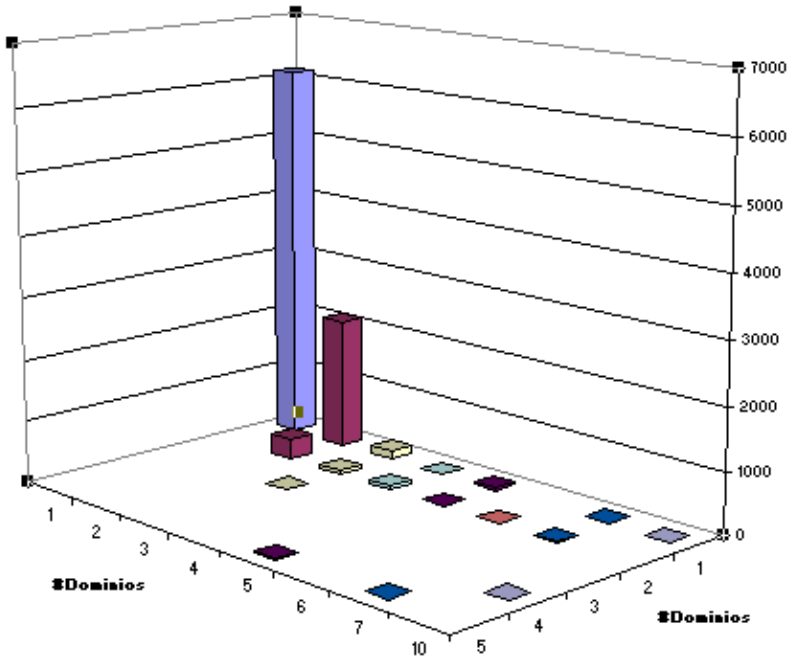

Gráfico 10: Cantidad de dominios involucrados en las interacciones

Significado Biológico

Los resultados de este proyecto fueron analizados por el Dr. Martín Vazquez del INGEBI quien encontró de gran utilidad tener una predicción de las interacciones en T. cruzi, ya que les permite pensar nuevos comportamientos que no se habían considerado. Al momento de finalizar esta tesis continúan analizando la información entregada.

Capítulo 9 - Conclusiones

Las técnicas que infieren interacciones dominio-dominio están limitadas a la información disponible en las bases de datos de dominios (PROSITE, PRINTS, BLOCKS, Pfam, etc). A medida que se descubren más dominios mejores resultarán estas técnicas. Muchas de ellas generan resultados pobres debido a que todavía existe una gran cantidad de proteínas que no contienen al menos un dominio conocido, y por lo tanto quedan fuera del análisis.
La falta de un adecuado conocimiento de los dominios presenta serias dificultades, ya que los modelos necesitan por lo menos 1 instancia de una interacción dominio-dominio particular en el conjunto de entrenamiento para ser predicha.

Otro de los principales inconvenientes es que los datos se encuentran esparcidos en diferentes bases de datos y con diferente formato. Esto genera una de las mayores complicaciones al tener que unificar y validar los datos en las diferentes fuentes.

El algoritmo propuesto mejora notablemente al Método de Asociación al predecir reglas multi-dominio. El método ME obtiene buenos resultados pero necesita mucho más tiempo de procesamiento que nuestro algoritmo y al igual que el MA, no genera reglas multi-dominio

Nuestro algoritmo considera la ubicación (compartimiento) de la proteína dentro de la célula para inferir interacciones dominios-dominios que tengan mayor posibilidad de existencia. Como hemos visto, sólo un veinte por ciento de las proteínas analizadas contienen esta información y no es consistente con las interacciones publicadas (ver Apéndice I). Si se mejoraran las anotaciones de las proteínas o existiera una base de datos detallando el compartimiento donde se encuentra cada proteína, se podría obtener un mayor porcentaje de las mismas para las que se conoce el compartimiento y a partir de esto mejorar los resultados.
Otro concepto analizado fue el perfil hidrofóbico. Éste se basa en un cálculo para determinar si una región de la proteína tiene una tendencia a interactuar con otra, o bien estas nunca podrían hacerlo ya que por sus características, se repelen entre sí. A diferencia del análisis de compartimientos, éste tiene la ventaja de poder ser calculado debido a que se basa en la secuencia de aminoácidos. Sin

embargo, el análisis de los resultados no convalida las hipótesis preliminares sobre la influencia del perfil hidrofóbico en las interacciones.

La posibilidad de agregar conocimiento biológico es fundamental para mejorar los resultados de estos algoritmos. Ya no alcanza con desarrollar métodos estadísticos que analicen los datos en forma numérica o cualitativa. Es importante desarrollar algoritmos que analicen características biológicas para mejorar los conceptos a aprender. La mejora de este tipo de algoritmos vendrá de la mano de un trabajo interdisciplinario donde la relación biólogo – computador será la clave de este progreso.

Capítulo 10 - Futuros trabajos

Las técnicas vistas en este trabajo se basan en que los dominios conocidos son los que efectivamente producen la interacción entre proteínas, descartando que la interacción puede ser realizada por otra región de la proteína.
Algunos trabajos [Wojcik01] se basan en conocer la porción de la secuencia de aminoácidos que efectivamente está involucrada en la interacción, sin limitarse a dominios previamente conocidos. De esta manera, superan las limitaciones de la falta de información de los dominios. Lamentablemente esta información es muy escasa, por lo que no puede ser utilizada en la mayoría de los estudios.
En la actualidad se están desarrollando técnicas para determinar qué sección de la proteína se encuentra involucrada en las interacciones (Interaction sites). Estas técnicas se basan en generar un modelo a partir de estructuras de tres dimensiones de proteínas en donde se conoce precisamente cuál es la región que interactúa. Con estos modelos se puede predecir el lugar donde sucede la interacción (interaction site) de una proteína, conociendo sólo la secuencia de aminoácidos y sin conocer sobre su estructura de tres dimensiones.
Para llevar adelante estas técnicas se utilizan redes neuronales [Ogran03], [Zhow01], [Fariselli02] o un clasificador SVM (support vector machine) [Yan02], produciendo resultados significativos.
Uniendo estas técnicas y utilizando el algoritmo propuesto creemos que se pueden alcanzar mejores resultados. Explicaremos brevemente la idea a seguir.

1) Para cada proteína calcular su posible interaction site.
2) Generar clusters de proteínas según su patrón de interacción. Ej.: agrupando proteínas cuyos pares interactuantes sean los mismos.
3) Para cada cluster definir un dominio (utilizando por ejemplo HMM) basado en la alineación de las secuencias definidas en 1).
4) Correr el algoritmo ABR utilizando como entrada los dominios generados en 3).

Apéndice I – Compartimientos Celulares e Interacciones

En esta sección realizaremos un análisis sobre las interacciones de levadura donde estudiaremos los compartimientos celulares donde se encuentra cada interactor. Mediante este estudio justificaremos el uso del análisis de los compartimientos celulares en el Algoritmo Basado en Reglas. La siguiente tabla presenta los compartimientos analizados y la cantidad de proteínas que están clasificadas dentro de cada compartimiento. Como se puede apreciar, la mayor cantidad de proteínas no tiene asociado un compartimiento celular al cual pertenecen y perjudica la utilización de esta técnica en nuestro algoritmo.

Compartimiento	Cantidad	Porcentaje
Desconocido	6439	79.67%
Membrane	774	9.58%
Mitochondria	360	4.45%
Ribosomal	165	2.04%
Nuclear	159	1.97%
Vacuolar	50	0.62%
Cytoplasmic	45	0.56%
Golgi	42	0.52%
Endoplasmatic r.	32	0.40%
Peroxisoma	15	0.19%
Lysosomal	1	0.01%

Tabla 6: Distribución de los compartimientos en Levadura

La siguiente tabla muestra los compartimientos celulares y la cantidad de interacciones que existen entre ellos. En la misma se observa que la mayor cantidad de interacciones se encuentran en la diagonal, es decir, entre los mismos compartimientos. Además podemos observar que la membrana se comporta diferente al observar una cantidad de proteínas de otros compartimientos que interactúan con este. Esta característica nos resultó llamativa y fue consultada con el equipo de biólogos, quienes comentaron que los datos son coherentes ya que las proteínas alojadas en el compartimiento membrana tienden a interactuar con otras que se encuentran en otros compartimientos dado que se encuentran en contacto.

	Cytoplasmic	Golgi	Membrane	Mitochondria	Nuclear	Peroxisoma	Ribosomal	Vacuolar
Cytoplasmic	3	1	10	6	4		1	
Golgi		4	17	7	4		3	2
Lysosomal			4	1	1			
Membrana			248	142	110	4	51	36
Mitochondria				43	31		17	2
Nuclear					21		12	8
Peroxisoma						1		
Ribosomal							6	3
Vacuolar								4

Tabla 7: Cantidad de interacciones entre compartimientos

El siguiente gráfico corresponde a la tabla anterior.

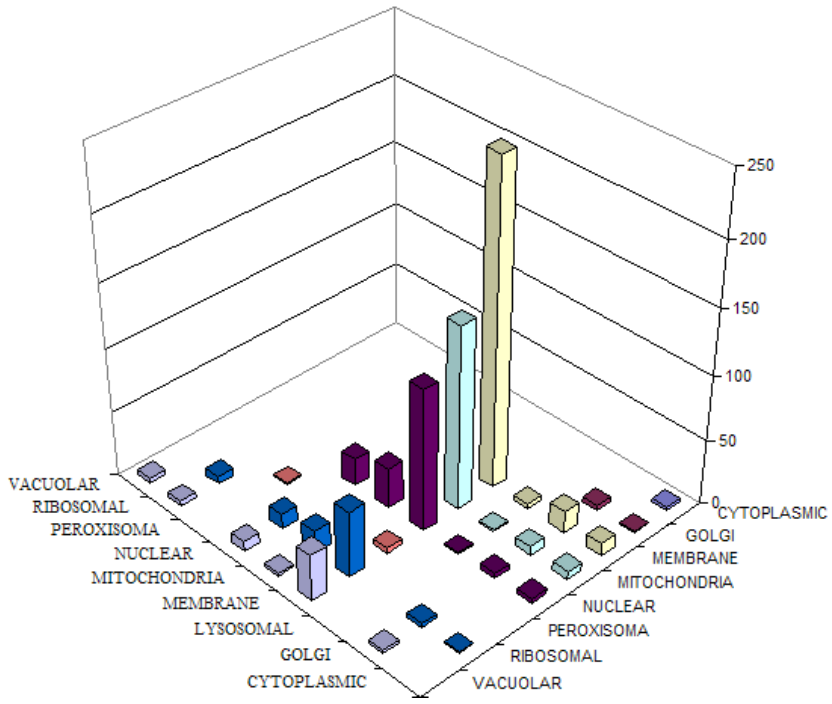

Gráfico 11: Cantidad de interacciones entre compartimientos

Para apreciar mejor el hecho la cantidad de proteínas que se encuentran sobre la diagonal, veamos el mismo gráfico anterior, pero sin las interacciones que se producen en la membrana.

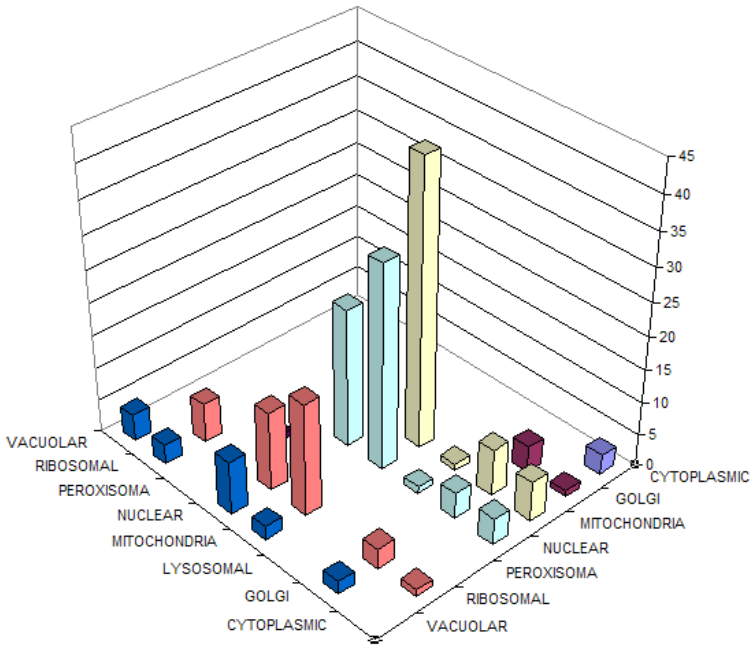

Gráfico 12: Cantidad de interacciones entre compartimientos sin membrana

Al eliminar membrana se ve que la diagonal predomina sobre el resto de los datos lo que confirma la hipótesis de trabajo. También se puede apreciar que existen interacciones que se producen entre distintos compartimientos, nuevamente realizamos la consulta a los biólogos y comentaron que existen muchas proteínas que migran de compartimiento a lo largo de su ciclo de vida y es probable que cuando fue identificada y anotada, se encontrara en algún lugar de la célula y al momento de interactuar en otro.

El siguiente gráfico muestra la diferencia en el comportamiento del algoritmo al considerar las interacciones con proteínas que se encuentran en el compartimiento Membrana. Originalmente el algoritmo evaluaba que el compartimiento celular sea el mismo para poder llevar a cabo la interacción, luego fue modificado para que considere que una interacción entre dos proteínas es posible si alguna de ellas se encuentra en el compartimiento Membrana o las dos se encuentran en el mismo compartimiento.

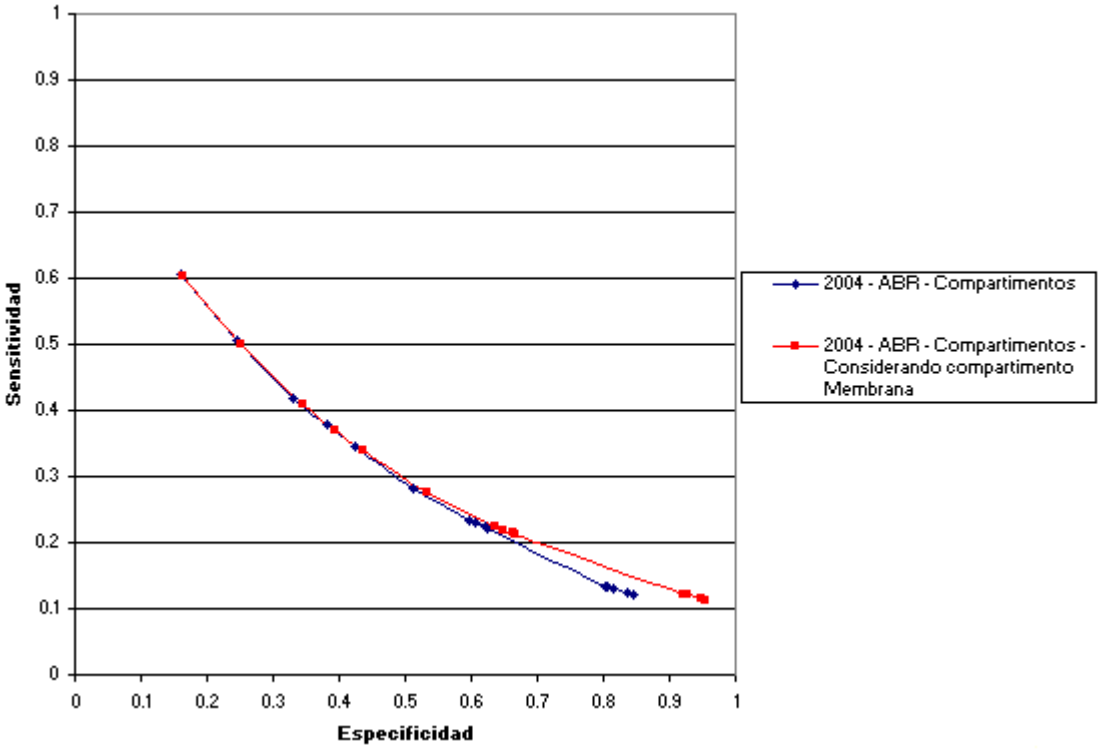

Gráfico 13: Compartimiento Membrana - Variación en los resultados

Apéndice II – Comparación de distintos juegos de datos

Durante la versión preliminar de este trabajo se emplearon los dominios registrados en la base de datos Pfam de Marzo de 2003. Esta base contiene dominios que provienen de experimentos y continuamente se está incorporando nueva información

Por esta razón en Enero de 2004 tomamos una nueva versión de la base de dominios para actualizar los datos utilizados y comparar el comportamiento de los algoritmos ante diferentes juegos de datos. Pero no sólo la base de dominios se encuentra en evolución, también hemos actualizado las bases de interacciones y proteínas. En el caso de las proteínas incorporamos datos que no utilizamos en Marzo de 2003 y son las bases de proteínas provistas por Bind y DIP.

Los datos que contábamos tenían las siguientes características:

Saccharomyces Cerevisiae			
	Fuente	**Cantidad-2003**	**Cantidad-2004**
Proteínas	NCBI	8297	8428
	BIND	0	76
	DIP	0	2434
Interacciones	BIND	5219	5044
	DIP	9856	10784
Dominios	PFAM	5196	7258

Se puede apreciar que las interacciones de Bind han bajado en 2004, pero en realidad este fenómeno se produce ya que se realizó en primer lugar la carga de las interacciones de DIP, por lo cual, algunas de Bind fueron rechazadas por estar duplicadas.

Saccharomyces Cerevisiae		
	Cantidad-2003	**Cantidad-2004**
Proteínas con Dominios	5827 (%70 del total)	8081 (%73 del total)
Interacciones entre proteínas con dominios	9535 (%63 del total)	11153 (%70 del total)

Presentaremos a continuación una comparación de los algoritmos con los diferentes juegos de datos.

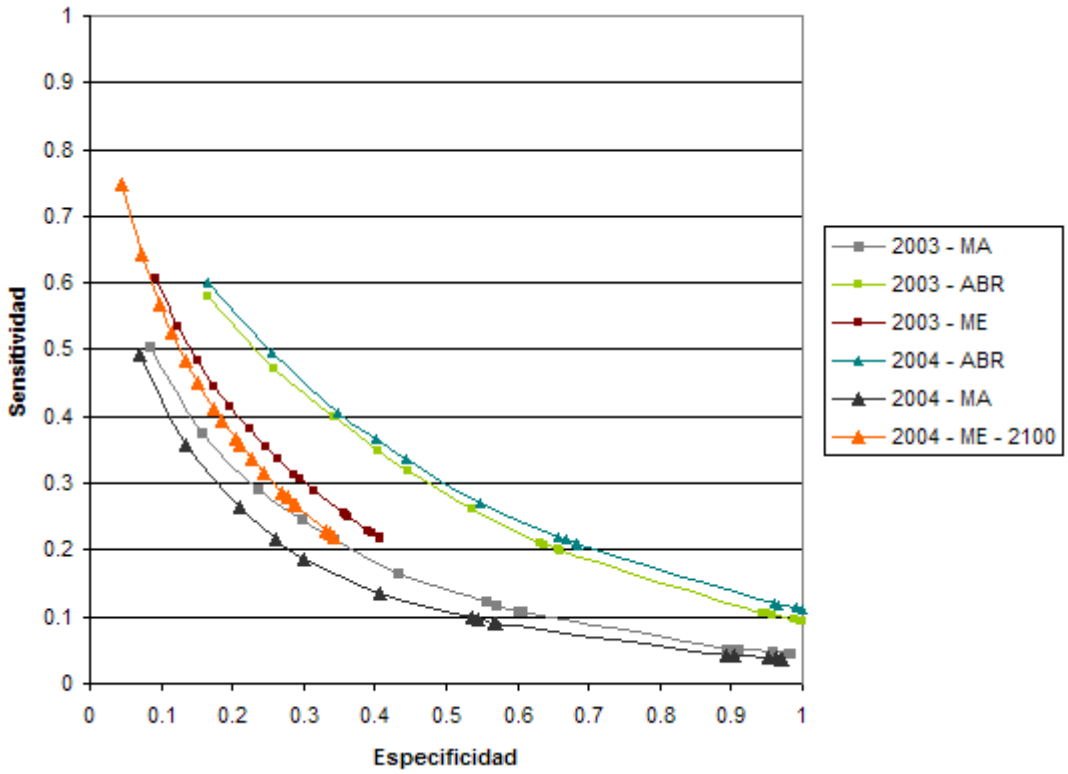

Gráfico 14: Comparativa datos 2003-2004

Podemos ver que el algoritmo Basado en Reglas mejora sensiblemente los resultados, mientras que el método MA y ME empeoran sus resultados.

La posibilidad de detectar interacciones multi-dominio aumenta debido a que incorporamos una mayor cantidad de dominios. El MA sólo trabaja con interacciones mono-dominio por lo cual debe generar múltiples reglas para representar los nuevos datos. Estas reglas empeoran los resultados ya que tienen poca precisión. El algoritmo ME posee un comportamiento similar.

Índice de Figuras

Índice de Tablas

Índice de Gráficos

Bibliografía

[Altschul97] Altschul S., Madden T., Schaffer A., Zhang J., Zhang Z., Miller W. and
Lipman D. Gapped BLAST and PSI-BLAST: a new generation of protein database
search programs (1997).

[Bilmes98] Bilmes J. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. International Computer Science Institute and Computer Science Division – Department of Electrical Engineering and Computer Science (1998).

[Collins97] Collins M. The EM Algorithm (1997).

[Deng02] Deng M., Mehta S., Sun F., Chen T. Inferring Domain-Domain inteaction from protein-protein interaction. Department of Biological Sciences, University of Surthen California (2002).

[Deng03] Deng M., Zhang K., Mehta S., Chen T., Sun F. Prediction of protein function using protein-protein interaction data. Molecular and Computational Biology Program, Department of Biological Sciences. University of Southern California (2003).

[Durbin98] Durbin R., Eddy S., Krogh A., Mitchison G., Biological Sequence analysis. Probabilistic models of proteins and nucleic acids (1998).

[Fariselli02] Fariselli P., Pazos F., Valencia A., Casadio R. Prediction of protein-protein interaction sites in heterocomplexes with neural networks. Biochem 269:1356-1361 (2002).

[Gomez01] Gomez S., Lo S., Rzhetsky A. Probabilistic Prediction of Unknown Metabolic and Signal-Trnsduction Networks. Columbia Genome Center, Department of Statistics and Department of Medical Informatics. (2001).

[Gomez02] Gomez M., Rzhetsky A. Towards The Prediction Of Complete Protein- Protein Interaction Networks. Columbia Genome Center, and Department of Medical Informatics, Columbia University (2002).

[Gribskov87] Gribskov M., McLachlan AD, Eisenberg D. Proc. Natl. Acad. Sci. U.S.A. 4:4355-4358(1987).

[Gribskov90] Gribskov M., Luethy R., Eisenberg D. Profile análisis. Meth. Enzymol. 183:146-159 (1990).

[Hopp81] Hopp TP and Woods KR Prediction of protein antigenic determinants from amino acid sequences. Proc. Nat. Acad. Sci. 6: 3824-3828 (1981).

[Hunter] Hunter L. Molecular Biology for Computer Scientist.

[Krogh94] Krogh A., Haussler D., Introduced profile HMMs. UC Santa Cruz (1994).

[Kyte82] Kyte J and Doolittle RF A Simple Method for Displaying the Hydropathic Character of a Protein . Journal of Molecular Biology 6: 105-142 (1982).

[Luethy94] Luethy R., Xenarios I., Bucher P. Protein Sci. 3:139-146(1994).

[Marcotte99] Marcotte E., Pellegrini M., Ng H., Rice D., Yeates T., Eisenberg D. Detecting Protein Function and Protein-Protein Interactions from Genome Sequences. Science 285:30 (1999).

[Matthews01] Matthews L., Vaglio P., Reboul J., Ge H., Davis B., Garrels J., Vincent S. and Vidal M. Genome Research 11:2120-2126 (2001).

[Mitchell97] Mitchell T., Machine Learning. McGraw-Hill (1997).

[Mrowka01] Mrowka R., Patzak A., Herzel H. Is there a Bias in Proteome Research? Berlin University. Genoma Research 11:19711973 (2001).

[Neng99] Neng S. Supervised Learning from Incomplete Data using Expectation Maximization (EM) Algorithm. University of Hawaii (1999).

[Ogran03] Ogran Y., Rost B. Predict protein-protein interaction site from local sequence information. Columbia University (2003).

[Oria02] Oria N., Garg A. MMIHMM: Maximum Mutual Information Hidden Markov Models. Univ. Ilinois (2002).

[Saito02] Saito R., Suzuki H., Hayashizaki Y. Interaction generality, a measurement to assess the realiability of a protein-protein interaction. Nucleic Acids Research 30-5:1163-1168 (2002).

[Setubal97] Setubal J., Meidanis J. Introduction to computational molecular biology. University of Campinas, Brazil. PWS Publishing company (1997).

[Smyth96] Smyth P., Heckerman D., Jordan M., Probabilistic Independence Network for Hidden Markov Probability Models. (1996).

[Sonnhammer97] Sonnhammer L., Eddy S. R., Durbin R. Pfam: a Comprehensive Database of Protein Domain Families Based on Seed Alignments. Department of Genetics, Washington University School of Medicine (1997).

[Thierry01] Thierry-Mieg N. Protein-Protein Interaction Prediction for C. elegans. Laboratoire LSR-IMAG (2001).

[Unger93] Unger R. and Moult J. Finding the lowest free energy conformation of a protein is an NP-hard problem: proof and implications. Bulleting of Mathematical Biology, 55(6): 1183-1198, 1993

[Wojcik01] Wojcik J., Schachter V. Protein-Protein interaction map inference using interaction domain profile pairs. Bioinformatics 17:296-305 (2001).

[Yan02] Yan C. Honavar V., Dobbs D. Predicting Protein-Protein Interaction Site from Amino Acid Sequence. Iowa State University (2002).

[Yan02a] Yan C., Baker L. H. Predicting Protein-Protein Interaction Sites From Amino Acid Sequence. Technical Report ISU-CS-TR-02-11. Department of Computer Science. Iowa State University (2002).

[Zhow01] Zhow, H., Shan Y. Prediction of protein Interaction Sites from Sequence profile and Residue Neighbor List. Proteins, Function and Genetics 44:336343 (2001).

Referencias

[BIND] Bader GD, Betel D, Hogue CW. (2003) BIND: the Biomolecular Interaction Network Database. Nucleic Acids Res. 31(1):248-50 PMID: 12519993.
http://www.bind.ca/

[DIP] Database of Interacting Proteins. http://dip.doe-mbi.ucla.edu/dip/Main.cgi

[HMMER] HMMER - Biological sequence analysis with profile hidden Markov models
Copyright (C) 1992-2003 HHMI/Washington University School of Medicine http://hmmer.wustl.edu/

[INGEBI] INGEBI - CONICET: Instituto de Investigaciones en Ingeniería Genética y Biología Molecular - Consejo Nacional de Investigaciones Científicas y Técnicas. Facultad de Ciencias Exactas y Naturales – Universidad de Buenos Aires http://proteus.dna.uba.ar.

[Java] http://java.sun.com

[MAST] Motif Alignment & Search Tool. Timothy L. Bailey and Michael Gribskov, "Combining evidence using p-values: application to sequence homology searches", Bioinformatics, 14(48-54), 1998. http://meme.sdsc.edu/meme

[Mysql] http://www.mysql.com

[NCBI] National Center for Biotechnology Information – http://www.ncbi.nlm.nih.gov

[Pfam] A database of Protein Families - http://pfam.wustl.edu

[Speedy]
http://www.dc.uba.ar/people/proyinv/fra/cluster_speedy/homepage.html

[TIGR] The Institute for Genomic Research. http://www.tigr.org