Universidad de Buenos Aires

Facultad de Ciencias Exactas y Naturales

Departamento de Computación

Desarrollo e implementación de un algoritmo para determinar la co-evolución de residuos en proteínas y análisis del impacto del uso de alfabetos reducidos

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Jonathan Javier Zaiat

Director: Dr. Adrián Turjanski

Codirector: Dr. Marcelo Martí

Ciudad Autónoma de Buenos Aires, 2015

DESARROLLO E IMPLEMENTACIÓN DE UN ALGORITMO PARA DETERMINAR LA CO-EVOLUCIÓN DE RESIDUOS EN PROTEÍNAS Y ANÁLISIS DEL IMPACTO DEL USO DE ALFABETOS REDUCIDOS

La presente tesis tiene como objetivo desarrollar y evaluar un algoritmo para detectar eficientemente Coevolución de residuos en proteínas basado en los principios de entropía de Shannon e Información Mutua.
Se llama Coevolución en proteínas al fenómeno por el cual se pueden observar mutaciones correlacionadas entre pares de aminoácidos al analizar un alineamiento múltiple de secuencias de una proteína. Encontrar estos pares sirve para identificar residuos que interactúan en una proteína, residuos estructural o funcionalmente importantes y posibles sitios de interacción entre la proteína y su sustrato o con otra proteína.
En este trabajo se analiza la capacidad predictiva del método desarrollado utilizando alfabetos reducidos, clusters por identidad de secuencia, distintas alternativas para definir la existencia de un contacto entre residuos y el uso de distintas distancias mínimas en secuencia.
Los resultados demuestran la eficacia del algoritmo desarrollado para detectar residuos que coevolucionan. Asimismo se concluye que los alfabetos reducidos aportan precisión, que los clusters por identidad de secuencia eliminan redundancias y que uno de los métodos propuestos para calcular contacto entre residuos obtiene mejores resultados que la medida usualmente utilizada.

Palabras claves: Coevolución, proteínas, información mutua, entropía, alfabetos reducidos, identidad de secuencia, distancia de contacto.

DEVELOPMENT AND IMPLEMENTATION OF AN ALGORITHM TO DETERMINE THE COEVOLUTION OF RESIDUES IN PROTEINS AND THE ANALYSIS ON THE IMPACT OF USING REDUCED ALPHABETS

The goal of the present thesis is to develop and evaluate an algorithm for an efficient detection of coevolution in protein residues based on the principles of Shannon's entropy and Mutual Information.
Coevolution in proteins is the phenomenon for which correlated mutations between pairs of aminoacids can be seen when analizing a multiple sequence alignment for a given protein family. Finding these pairs could be useful to identify residues interacting inside a protein, residues structural or functionally important and potential interaction sites between a protein and a sustrate or with another protein.
In this work we analize our method predictive capacity using reduced alphabets, clusters by sequence identity, different alternatives to define the existence of contact between residues and the use of several minimal distance between residues in the sequence.
The results prove the accuracy of the algorithm developed to detect coevolving residues. Also, we conclude that reduced alphabets increase the precision, that the clusters by sequence identity eliminate redundancies and that one of the proposed methods to calculate contact between residues obtains better results than the measure most commonly used.

Palabras claves: Coevolution, proteins, mutual information, entropy, reduced alphabets, sequence identity, contact distance.

AGRADECIMIENTOS

Desde la profesional quiero agradecer en primer lugar a Adrián y a Marcelo, quienes me tuvieron una paciencia infinita y demostraron confiar en mí más de lo que yo mismo podría hacerlo, es un placer poder trabajar con uds y aprender tanto.
A los chicos de BIA, por el ultimo empujón, por ayudarme a resolver mis dudas, por hacerme cumplir mi promesa (¿quién se podía imaginar que Racing iba a salir campeón después de todo?).
A Juan Ma por ayudarme a avanzar y a Lanza que me dió una mano al principio.

Desde lo personal, quiero agradecer a mis papás por TODO lo que me ayudan y me apoyan, este trabajo no sería posible sin la educación que me brindaron, sin el amor incondicional y sin todo lo que hacen día a día por mi.
A mi hermano Matías, a quien extraño en demasía, quien fuera mi modelo a seguir, mi principal crítico y también el primero en estar orgulloso de mí.
A Vane, que se creyó que me "faltaba poco para terminar la tesis" cuando nos conocimos, que me aguanta en todas, que me apoya, que me insiste hasta el cansancio, que me ayudó a seguir adelante cuando más lo necesitaba, con quien construimos una familia hermosa y cuyo ejemplo de perseverancia, estudio y dedicación me resulta imposible de imitar.
A mi hijo Juli, por ser la persona más tierna y hermosa del mundo, por darme una razón para intentar mejorar día a día, espero que aprendas de mis aciertos y de mis errores para que puedas cometer los tuyos propios.
A mis amigos, por estar, por compartir, por crecer y hacerme crecer a su lado.

Y a todos los que alguna vez me preguntaron "¿todavía no terminaste la tesis?".

A Matías.
A Juli.

Índice general

1. INTRODUCCIÓN

La presente tesis es un trabajo interdisciplinario encuadrado en el marco de la Bioinformática, pues intenta resolver un problema de la Biología utilizando un enfoque computacional.

En particular en este trabajo busqué detectar eficientemente la coevolución de residuos en proteínas, utilizando una combinación de técnicas bioinformáticas con métodos estadísticos que permiten un mejor análisis de los resultados. Para llevar adelante este objetivo, durante el presente trabajo de tesis:

- requerí comprender los conceptos biológicos involucrados y el objeto de estudio, principalmente a las proteínas, los aminoácidos que las componen y la coevolución entre ellos,
- convertí los datos biológicos (el conjunto de proteínas y sus aminoácidos) en estructuras de datos lógico-matemáticas y computacionales,
- programé un algoritmo para determinar el contenido de Información Mutua en un alineamiento múltiple de secuencias, y
- generé y analicé una serie cálculos estadísticos para determinar los resultados significativos.

1.1. Conceptos biológicos

A continuación definiremos brevemente algunos conceptos fundamentales de biología necesarios para la comprensión de este trabajo de tesis.

1.1.1. Aminoácido

Un aminoácido es una molécula orgánica con un grupo amino (-NH2) y un grupo carboxilo (-COOH; ácido). Los aminoácidos más frecuentes y de mayor interés son aquellos que forman parte de las proteínas, los cuales están formados por un carbono alfa unido a un grupo carboxilo, a un grupo amino, a un hidrógeno y a una cadena de estructura variable (habitualmente denominada residuo o cadena lateral), que determina la identidad y las propiedades de cada uno de los diferentes aminoácidos; existen cientos de posibles residuos por lo que se conocen cientos de aminoácidos diferentes, pero sólo 20 forman parte de las proteínas y son codificados por el ADN mediante el código genético (ver Tabla 1.1). Estos 20 aminoácidos se diferencian en la estructura de la cadena lateral y pueden ser clasificados según las propiedades físico químicas de la misma:

- **Alifáticos (Glicina, Alanina, Valina, Leucina e Isoleucina)**: En este grupo se encuadran los aminoácidos cuya cadena lateral es alifática, es decir una cadena

hidrocarbonada. Tienen carácter hidrófóbico, tanto más marcado cuanto mayor es la longitud de la cadena. La glicina tiene un tamaño muy pequeño, y permite con su presencia la formación de estructuras particulares, como la triple hélice del colágeno. Participan de interacciones hidrofóbicas.

- **Prolina**: También tiene una cadena lateral de naturaleza alifática pero es un iminoácido, es decir, su grupo amino no es un grupo amino primario, como los de los demás aminoácidos, sino secundario. La presencia del anillo impide el giro sobre ese enlace, y consecuentemente la organización de la estructura secundaria de la proteína. Las prolinas no son toleradas en alfa-hélices.

- **Aromáticos (Fenilalanina, Triptófano y Tirosina)**: Son aquellos cuya cadena lateral contiene un anillo aromático. La fenilalanina es una alanina que lleva unida un grupo fenílico. La tirosina es como la fenilalanina con un hidroxilo en su anillo aromático, lo que lo hace menos hidrofóbico y más reactivo. El triptófano tiene un grupo indol. Además de formar parte de las proteínas, son precursores de otros compuestos biológicos. En las proteínas, son responsables de la absorción de rayos UV. Participan de interacciones aromáticas (o sea, entre ellos) e hidrofóbicas.

- **Azufrados (Metionina y Cisteína)**: Son aquellos aminoácidos cuya cadena lateral posee un átomo de azufre. Ambos son bastante inestables frente a condiciones de oxidación. La cisteína es muy importante en el mantenimiento de la estructura terciaria y cuaternaria de la mayoría de las proteínas mediante la formación de puentes disulfuro en dímeros de la cisteína formados por oxidación.

- **Alcoholes (Treonina y Serina)**: Son aquellos que tienen cadenas alifáticas hidroxiladas. El grupo hidroxilo hace que estos aminoácidos sean más hidrofílicos y reactivos. Ambos aminoácidos, especialmente la serina, pueden estar modificados por fosforilación, o por glicosilación en el caso de las glicoproteínas. Participan en interacciones polares, por ejemplo con el agua en la superficie proteica.

- **Ácidos (Ácido aspártico y Ácido glutámico)**: Son los aminoácidos con cadenas laterales de naturaleza ácida. Ambos tienen un grupo carboxilo en la cadena lateral, además del que forma el enlace peptídico. Este grupo carboxilo puede estar o no ionizado en función del pH del medio. Son aminoácidos hidrófilos y los responsables de las cargas negativas de la proteína.

- **Con grupo amida (Asparagina y Glutamina)**: Estos aminoácidos, de carácter hidrófilo, son las amidas del amonio del aspártico y glutámico. Son excelentes donores y aceptores de puentes de hidrógeno.

- **Básicos (Lisina, Arginina e Histidina)**: Estos son hidrofílicos, teniendo o no carga positiva en función del pH del medio. Son relativamente inestables, especialmente la lisina.

Las cadenas laterales determinan diferencias en tamaño, carga, capacidad de formación de puentes de hidrógeno y en reactividad química. La versatilidad de las funciones de las proteínas es el resultado de la diversidad de las cadenas laterales de los aminoácidos.

Tabla 1.1: Aminoácidos

Aminoácido	Código de tres letras	Código de una letra
Alanina	Ala	A
Arginina	Arg	R
Asparagina	Asn	N
Ácido aspártico	Asp	D
Cisteína	Cys	C
Glutamina	Gln	Q
Ácido glutámico	Glu	E
Glicina	Gly	G
Histidina	His	H
Isoleucina	Ile	I
Leucina	Leu	L
Lisina	Lys	K
Metionina	Met	M
Fenilalanina	Phe	F
Prolina	Pro	P
Serina	Ser	S
Treonina	Thr	T
Triptófano	Trp	W
Tirosina	Tyr	Y
Valina	Val	V

1.1.2. Proteína

Las proteínas son largas cadenas no ramificadas de aminoácidos unidas por enlaces peptídicos entre el grupo carboxilo (-COOH) y el grupo amino (-NH2) de residuos de aminoácidos adyacentes. La secuencia de aminoácidos en una proteína está codificada en su gen (una porción de ADN) mediante el código genético. Las proteínas ocupan un lugar de máxima importancia entre las moléculas constituyentes de los seres vivos pues son encargadas de cumplir diversas funciones, como mantener su estructura, regular procesos, recibir señales de factores externos o mediar la respuesta inmune. Prácticamente todos los procesos biológicos dependen de la presencia o la actividad de este tipo de moléculas.

Las proteínas poseen distintos niveles de organización:

- **Estructura primaria**: está determinada por la secuencia de aminoácidos de la cadena proteíca, es decir, el número y tipo de aminoácidos presentes y el orden en que están enlazados por medio de enlaces peptídicos. Por convención, (coincidiendo con el sentido de síntesis natural) el orden de escritura es siempre desde el grupo amino-terminal hasta el carboxi-terminal. Un ejemplo de estructura primaria se muestra en la Figura 1.1 para la insulina.
- **Estructura secundaria**: es el tipo de estructura (o forma) que adoptan segmentos de la cadena proteica en el espacio. Se adopta gracias a la formación de puentes de hidrógeno entre los grupos carbonilo (-CO-) y amino (-NH-) de los carbonos

involucrados en las uniones peptídicas de aminoácidos cercanos en la cadena. Las dos conformaciones más comunes son hélices alfa (α-helix) y láminas beta (β sheets), las cuales se muestran en la Figura 1.2.

- **Estructura terciaria**: es la forma en que las estructuras secundarias de la cadena polipeptídica se acomodan en el espacio dando lugar a la estructura o forma tridimensional. Sus principales tipos son globular y fibrosa. Esta estructura hace posible la función de la proteína. Un ejemplo de estructura terciaria se muestra en la Figura 1.3 para la insulina.

- **Estructura cuaternaria**: deriva de la conjunción de varias cadenas peptídicas que, asociadas, conforman un ente, un multímero, que posee propiedades distintas a la de sus monómeros componentes. Dichas subunidades se asocian entre sí mediante interacciones no covalentes, como pueden ser puentes de hidrógeno, interacciones hidrofóbicas o puentes salinos. Para el caso de una proteína constituida por dos monómeros, un dímero, éste puede ser un homodímero, si los monómeros constituyentes son iguales, o un heterodímero, si no lo son.

Fig. 1.1: Estructura primaria de la insulina. La misma tiene dos cadenas, A-chain y B-chain, unidas por puentes disulfuro (indicados con una línea conteniendo dos azufres).

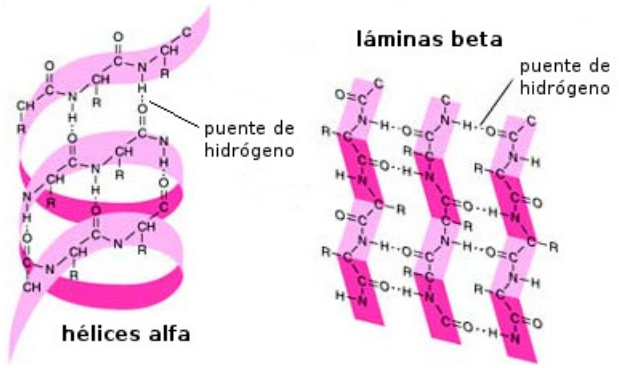

Fig. 1.2: Estructura secundaria de las proteínas

El Dogma central de la Biología Molecular y el flujo de información biológica

El ADN es la molécula donde se encuentra codificada la información genética. Se trata de una molécula larga en forma de hélice y que puede representarse como dos largos filamentos moleculares enrollados y unidos por las bases o nucleótidos. Hay cuatro tipos de bases (A, C, G y T) y cada filamento está unido al otro por las bases complementarias del otro.

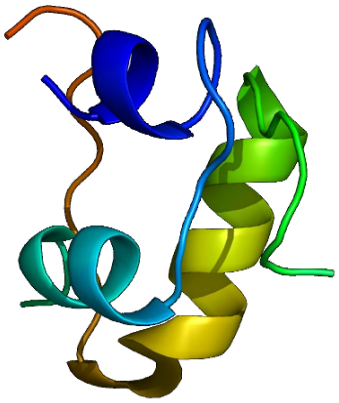

Fig. 1.3: Estructura terciaria de la insulina

La información genética está contenida en los genes, segmentos de ADN que llevan información para fabricar un producto funcional determinado. Sólo una pequeña parte del genoma es codificante; la mayor parte corresponde a secuencias cortas móviles no codificantes o a secuencias regulatorias.

Para que la información pase de una molécula a otra, primero debe copiarse, en un proceso que se llama replicación y que ocurre en el núcleo. Pero como el ADN se encuentra en el núcleo y las proteínas son sintetizadas en el citoplasma, debe existir una molécula que funcione como intermediaria. Este papel lo cumple el ácido ribonucleico mensajero (ARNm). El ADN se copia en ARNm en el núcleo, en un proceso denominado transcripción. Luego la información contenida en el ARNm es empleada para construir proteínas en el proceso de traducción, que tiene lugar en el citoplasma.

Estos tres procesos secuenciales constituyen el llamado *Dogma central de la Biología Molecular*, que establece que la información fluye desde el ADN al ARN y de este a las proteínas (ver Fig. 1.4). Además, las proteínas controlan el proceso de replicación del ADN uniéndose a una secuencia específica en el mismo, activando o inhibiendo así la transcripción de un gen determinado.

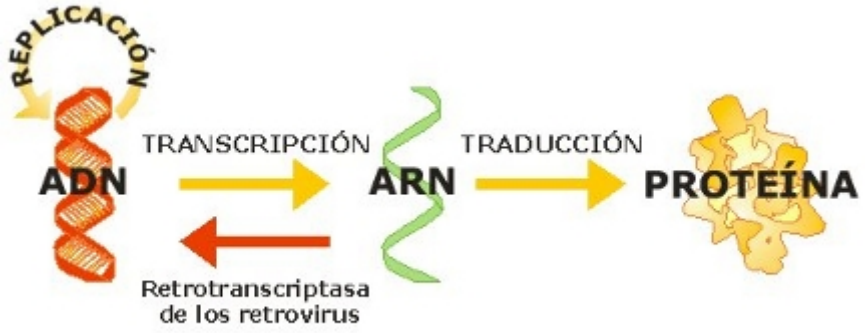

Fig. 1.4: Dogma central de la Biología Molecular

1.1.3. Alineamiento de secuencias

Un alineamiento de secuencias es una forma de representar y comparar dos o más secuencias de aminoácidos o nucleótidos para encontrar similitudes, que podrían indicar relaciones funcionales o evolutivas entre ellas. Las secuencias alineadas se escriben con las letras (representando aminoácidos o nucleótidos) en las filas de una matriz en las que, si es necesario, se insertan espacios (gaps) para que las zonas con idéntica o similar estructura se alineen.

Fig. 1.5: Alineamiento simple de secuencias. Cada aminoácido se representa con un color diferente y los gaps con guiones.

Si dos secuencias en un alineamiento comparten un ancestro común (o sea, son homólogas), las no coincidencias pueden interpretarse como mutaciones puntuales (sustituciones), y los huecos como indels (mutaciones de inserción o deleción) introducidas en uno o ambos linajes en el tiempo que transcurrió desde que divergieron. En el alineamiento de secuencias proteicas, el grado de similitud entre los aminoácidos que ocupan una posición concreta en la secuencia puede interpretarse como una medida aproximada de conservación en una región particular, o secuencia motivo, entre linajes. La ausencia de sustituciones, o la presencia de sustituciones muy conservadas (la sustitución de aminoácidos cuya cadena lateral tiene propiedades químicas similares) en una región particular de la secuencia suele indicar que esta zona tiene importancia estructural o funcional.

Existen varios algoritmos para obtener alineamientos de secuencias, entre los cuales cabe mencionar a los siguientes:

- Dot-matrix: consiste en disponer dos secuencias en una matriz (una como filas y la otra como columnas) y dibujar un punto donde existe una coincidencia entre ellas. Es un método simple y visual pero poco preciso, el cual se utiliza generalmente con fines didácticos o visuales.
- Needleman-Wunsch: es un algoritmo de programación dinámica que genera un alineamiento global. Para ello utiliza una matriz de sustitución (generalmente se utiliza una matriz blosum) y penalizaciones para la apertura o extensión de un gap. Su complejidad algorítmica es $O(n^3)$. Este método garantiza que, dada una función de scoring particular (determinada por la matriz de sustitución), se obtiene un alineamiento óptimo. El mismo es extensible a más de dos secuencias, pero resulta prohibitivamente lento para un gran número de secuencias (o para secuencias muy largas).
- Smith-Waterman: es un algoritmo de programación dinámica que genera un alineamiento local. Para ello utiliza una matriz de sustitución (generalmente se utiliza una matriz blosum) y penalizaciones para la apertura o extensión de un gap. Su complejidad algorítmica es $O(n^2)$. Este método garantiza que, dada una función de scoring particular (determinada por la matriz de sustitución), se obtiene un alineamiento óptimo. El mismo es extensible a más de dos secuencias, pero resulta prohibitivamente lento para un gran número de secuencias (o para secuencias muy largas).

- BLAST: utiliza un método heurístico no exhaustivo (llamado *word method* o método de k-tuplas) para hallar alineamientos locales, los cuales son extendidos usando una matriz de sustitución (blosum62). Consiste en armar una tabla de look-up con subsecuencias y luego realizar búsquedas a gran escala en diversas bases de datos.

En la actualidad, Needleman-Wunsch y Smith-Waterman son los métodos más comúnmente utilizados para alinear y comparar pares de secuencias.

Alineamiento múltiple de secuencias

El alineamiento múltiple de secuencias (MSA, multiple sequence alignment) es una extensión del alineamiento de secuencias de a pares (o alineamiento simple) que incorpora más de dos secuencias al mismo tiempo. Los métodos de alineamiento múltiple intentan alinear todas las secuencias de un conjunto dado. Los alineamientos múltiples son usados a menudo en la identificación de regiones conservadas en un grupo de secuencias que hipotéticamente están relacionadas evolutivamente. Estos motivos conservados pueden ser usados, en conjunto con la estructura y con información estadística, para localizar sitios activos catalíticos de las enzimas. Los alineamientos son también utilizados para ayudar al establecimiento de relaciones evolutivas mediante la construcción de árboles filogenéticos. Los alineamientos múltiples de secuencias son computacionalmente difíciles de producir y la mayoría de las formulaciones del problema conducen a problemas de optimización combinatoria NP-completos. Sin embargo, la utilidad de estos alineamientos en la bioinformática ha dado lugar al desarrollo de una variedad de métodos adecuados para la alineación de tres o más secuencias, entre los que podemos destacar a los métodos progresivos (los cuales comienzan alineando secuencias o regiones similares y progresivamente adicionan las menos relacionadas, como ejemplo podemos mencionar Clustal), a los métodos iterativos (una mejora de los anteriores que incluye una función objetivo a ser optimizada) y al descubrimiento de motivos (motif finding). Por otro lado, los algoritmos de programación dinámica son teóricamente aplicables pero resultan altamente costosos en tiempo y memoria para el alineamiento múltiple de secuencias.
Un alineamiento múltiple de secuencia se representa como una matriz, donde las filas son las secuencias de la familia en estudio, y las columnas son las posiciones equivalentes en todas las secuencias. A partir de estas columnas, se realiza el análisis de la variación en cada columna, lo que permite determinar la frecuencia de aparición de los distintos residuos, el contenido de información de una posición y la conservación del aminoácido asociado a la misma, entre otras medidas.

1.1.4. Interacción entre aminoácidos

Además de los enlaces covalentes, entre los aminoácidos existen otras fuerzas débiles que los atraen. Consideramos entonces que dos aminoácidos están en contacto y que existe interacción entre ellos si alguna de estas fuerzas se manifiesta de manera significativa entre los mismos.

- Fuerzas iónicas: se produce entre aminoácidos cargados positiva y negativamente por atracción de cargas diferentes.

- Fuerzas polares: es una interacción no covalente entre dos moléculas neutras polares, debido a la atracción eléctrica entre dipolos opuestos. Las moléculas que son dipolos se atraen entre sí cuando la región positiva de una está cerca de la región negativa de la otra. Una molécula polar puede interaccionar con un ión, o bien con otra molécula polar.
- Puente de Hidrógeno: es un caso especial de dipolo-dipolo entre un átomo electronegativo y un átomo de hidrógeno unido covalentemente a otro átomo electronegativo. Es un enlace débil.
- Fuerzas de Van Der Waals: son fuerzas inespecíficas y muy débiles, que generan dipolos instantáneos
- Interacciones hidrofóbicas: son las fuerzas que mantienen juntas las regiones no polares de las moléculas, pues estas regiones son hidrofóbicas y tienden a agruparse en medio acuoso para evitar el contacto con el agua.
- Puente Disulfuro: es el tipo de enlace que se establece al oxidarse dos cisteínas para formar una cistina mediante la unión de los dos azufres.

En su caracter de atracción, estas fuerzas mantienen a los aminoácidos a una distancia muy cercana, la cual varía de acuerdo a la fuerza en cuestión, la naturaleza de los aminoácidos y la disponibilidad para ubicarse en el espacio tridimensional. Por este motivo, conociendo la distancia entre dos aminoácidos, podemos determinar si estos están interactuando o no. En este trabajo, usaremos dos formas de calcular la distancia entre aminoácidos:

- obteniendo el centro geométrico de cada aminoácido y calculando la longitud del segmento que los une,
- calculando la distancia entre el átomo más cercano de un aminoácido a algún átomo del otro aminoácido (en ambos casos exceptuando a átomos hidrógeno).

Usualmente se considera que existe interacción (o contacto) si los centros geométricos se encuentran a menos de 8 Å, o si los átomos más cercanos se encuentran a menos de 6 Å. Asimismo, en la presente tesis analizaremos distintos valores de distancia mínima de interacción (también llamada distancia de contacto).

1.1.5. Familia de proteínas

Una familia de proteínas es un grupo de proteínas relacionadas evolutivamente. Las proteínas de una familia descienden de un antepasado común y típicamente poseen estructuras tridimensionales, funciones y secuencias similares. Las familias de proteínas pueden ser caracterizadas mediante los alineamientos múltiples de las secuencias que los componen, ya que estos revelan la conservación y divergencia de las diferentes posiciones. En este trabajo de tesis usaremos los MSAs de familias proteícas para inferir qué pares de aminoácidos coevolucionan dentro de la misma.

1.1.6. Evolución

Actualmente la teoría evolutiva más aceptada es la Síntesis Evolutiva Moderna (también llamada Teoría Sintética), la cual surge, entre los años 30 y 40, de la integración de la teoría de la evolución de las especies por selección natural de Charles Darwin, la teoría genética de Gregor Mendel como base de la herencia biológica, la mutación genética aleatoria como fuente de variación y la genética de poblaciones. Sus principales artífices fueron Julian Huxley, Theodosius Dobzhansky, Ernest Mayr, Sewell Wright, Ledyard Stebbins, George Gaylord Simpson y Bernard Rensch.

De acuerdo a esta teoría, la variación genética de las poblaciones surge por azar mediante la mutación (causada por errores en la replicación del ADN) y la recombinación (la mezcla de los cromosomas homólogos durante la meiosis). La evolución consiste básicamente en los cambios en la frecuencia de los alelos entre las generaciones, como resultado de la deriva genética, el flujo genético y la selección natural. La especiación podría ocurrir gradualmente cuando las poblaciones están aisladas reproductivamente, por ejemplo por barreras geográficas (especiación alopátrica), o por cambios dentro de una misma población (especiación simpátrica).

A nivel molecular, el proceso evolutivo se ve reflejado con pequeños y graduales cambios en la secuencia de nucleótidos del ADN. Estos cambios, al momento de la traducción de los genes en proteínas, pueden llevar aparejadas modificaciones en la secuencia de aminoácidos, las cuales a su vez pueden tener un impacto en la estructura y función de la proteína que codifican. El conjunto de proteínas afectadas por este tipo de cambios podrá otorgar al individuo ventajas o desventajas sobre el resto (o serán neutros) y aportar a su adaptación al entorno, lo cual en cierta forma determinará las posibilidades de que estos cambios sean transmitidos o no a su descendencia.

1.1.7. Coevolución en proteínas

Comparando las secuencias de una familia de proteínas entre sí mediante un alineamiento múltiple (MSA), podemos ver que los aminoácidos presentes en algunas posiciones coinciden o varían levemente, mientras que en otras posiciones existe una variabilidad mayor. Se puede inferir entonces que las posiciones que coinciden (o tienen poca variabilidad) contienen residuos conservados que suponemos son importantes para la estructura o la función de la proteína, pues no fueron toleradas mutaciones en las mismas (es decir, a largo plazo, los individuos con estas mutaciones fueron menos eficientes o aptos para transmitir sus genes que aquellos que no las tenían). Asimismo, podemos notar que la variabilidad en una posición está dada por la capacidad de las proteínas de tolerar la presencia de distintos residuos en la misma posición, y que esta depende fuertemente del entorno del residuo (o sea de quienes son sus residuos vecinos) por lo que muchas veces la variabilidad de residuos vecinos en la estructura está relacionada. Entonces se deduce que lo que ocurre es que las mutaciones en una posición pueden ocurrir porque están acompañadas o precedidas de cambios compensatorios en otra posición relacionada. Esta compensación resulta en un emparejamiento entre los cambios en las dos posiciones, y es llamado **Coevolución de residuos**.

Aquellos residuos que se encuentran cercanos en la estructura terciaria se encuentran interactuando entre sí (también se dice que están en contacto), y es esta interacción la que

puede generar la necesidad de cambios compensatorios entre ellos, ya sea por restricciones estructurales o funcionales. Por lo cual podemos aseverar que **dos residuos que están en contacto tienen tendencia a coevolucionar**. Dado que tendremos a disposición la secuencia de las proteínas y la distribución tridimensional de todos los átomos involucrados, a lo largo de este trabajo validaremos los resultados que arroje el algoritmo utilizando el siguiente **control positivo (gold standard)**:

Si dos residuos están en contacto, entonces coevolucionan.

Cabe aclarar que coevolución no implica contacto, es decir pueden existir residuos distantes que coevolucionan debido a otros factores [1, 2] (como ser acumulación de interacciones más débiles, cercanía en otras conformaciones de la misma proteína o en contacto en conformación homo-oligomérica), por lo cual es posible que se presenten casos que no podamos validar con la información que disponemos.

Teniendo en cuenta que las proteínas regulan y ejecutan la mayoría de los procesos bioquímicos que ocurren dentro de las células, resulta primordial comprender el mecanismo por el cual ocurre la relación entre proteínas. Dado que esta relación se da por la interacción entre residuos, encontrar pares de posiciones que coevolucionan dentro de una secuencia puede servir para identificar residuos que interactúan dentro de la proteína, residuos estructural o funcionalmente importantes y posibles sitios de interacción entre la proteína y su sustrato o con otra proteína.

2. MÉTODOS

2.1. Representación computacional de los datos biológicos

Para poder tratar el problema biológico con herramientas computacionales se debe encontrar modelos discretos que representen los datos y puedan ser utilizados en los algoritmos necesarios.

En este caso los datos biológicos y los modelos utilizados son:

- los aminoácidos, los cuales se representan con el código de una o tres letras correspondiente (ver Tabla 1.1). En consecuencia, el alfabeto de residuos se representa como un conjunto de caracteres (con un máximo de 20 posibles).
- la estructura primaria de una proteína, la cual se representa como una cadena de caracteres (*string*), donde cada caracter es un aminoácido. El orden de los caracteres en la cadena refleja el orden los aminoácidos en la secuencia, lo cual permite utilizar las posiciones de la cadena para indicar posiciones en la estructura primaria.
- la estructura terciaria de una proteína, la cual se representa como un conjunto de átomos que conforman los aminoácidos con sus coordenadas cartesianas (x, y, z) en el espacio tridimensional.
- la familia de una proteína, la cual se representa con el alineamiento múltiple de las secuencias de la familia expresado como una matriz donde las filas son las secuencias y las columnas son las posiciones equivalentes en todas las secuencias.

2.2. Conceptos matemáticos y estadísticos

2.2.1. Entropía de Shannon

La entropía de Shannon es una medida de incertidumbre para una fuente de información. Fue introducida en 1948 por Claude E. Shannon en un trabajo fundamental [3] en el marco de la Teoría de la Comunicación. Permite medir cuánta información es producida por una fuente, pues determina qué grado de "elección" está involucrado en la ocurrencia de un evento particular dentro de un conjunto de posibles eventos (cada uno con una probabilidad propia). De esta manera, es posible medir el desorden (o ruido) de la fuente de información.

Dada una variable aleatoria discreta X, cuyos valores posibles están definidos por el alfabeto $\{x_1, x_2, \ldots, x_K\}$ y cuya distribución de probabilidad asociada es $p(X) = \{p(x_1), p(x_2), \ldots, p(x_K)\}$, donde $\sum_{i=1}^{K} p(x_i) = 1$ y $b \in \mathbb{N}_{>0}$, la entropía de Shannon H(X) está definida como:

$$H(X) = -\sum_{i=1}^{K} p(x_i) \log_b p(x_i) \tag{2.1}$$

Esta ecuación expresa que la entropía de una variable aleatoria X es el valor medio ponderado de la cantidad de información de los diversos estados de la misma, y representa una medida de la incertidumbre acerca de una variable aleatoria y por tanto de la cantidad de información. Puede observarse que cuando X puede tomar sólo un valor, es decir $X = x$ con probabilidad $p(x) = 1$, $H(X) = 0$, es decir no hay incertidumbre. Cuando la probabilidad está distribuida equitativamente en todos los elementos del alfabeto, $X = x_i$ con probabilidad $p(x_i) = \frac{1}{K} \forall i$, la entropía $H(X)$ está maximizada. La elección del logaritmo en base b permite escalar la entropía; por ejemplo con $b = K$ se obtiene que $maxH(X) = 1$.

La entropía de Shannon será utilizada en este trabajo para medir el contenido de información de cada columna del alineamiento múltiple, a través del cálculo de la frecuencia de aparición de cada aminoácido en dicha columna en cada secuencia del MSA. En este caso, el contenido de información de una posición se refiere a la variabilidad biológica de la misma, a cuales son las variantes aceptadas en dicha posición de manera que la proteína sigue siendo viable y funcional, a cuan conservada evolutivamente está la posición, y, por ende, a la importancia que tiene la misma para la estructura y/o función de la proteína en cuestión. En la Figura 2.1 se muestra un diagrama de la frecuencia de un extracto de la familia Globinas, en la cual se muestra la frecuencia relativa de los aminoácidos presentes en cada posición.

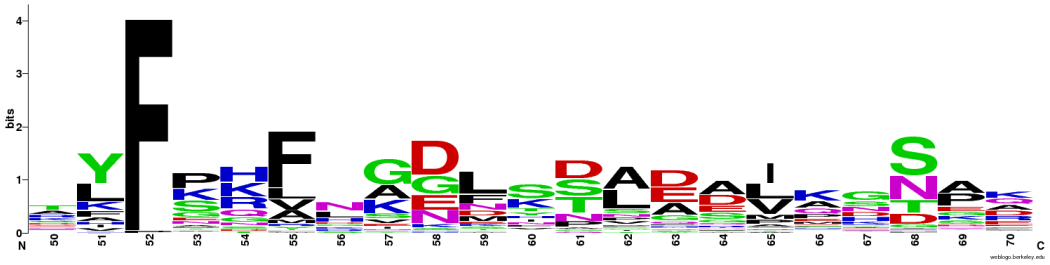

Fig. 2.1: Frecuencia de un extracto de la secuencia de la familia Globinas (PF00042). El tamaño de cada letra es relativa a la frecuencia de aparición del aminoácido correspondiente en esa columna. Por ejemplo, en la posición 52 se ve que la fenilalanina (F) está conservada pues no acepta otras opciones, mientras que la posición 58 contiene por igual cuatro residuos: ácido aspártico (D), glicina (G), ácido glutámico (E) y asparagina (N).

El concepto de entropía puede ser extendido a dos variables aleatorias, las cuales constituyen un par ordenado (x, y) pertenecientes al alfabeto extendido que se obtiene por el producto del alfabeto de cada uno (por lo cual sus elementos son todos los posibles pares distintos). Esta extensión se llama entropía conjunta y está definida como:

$$H(X,Y) = -\sum_{i=1}^{K}\sum_{j=1}^{L} p(x_i, y_j) \log_b p(x_i, y_j) \tag{2.2}$$

Asimismo, la entropía de una variable aleatoria dado el conocimiento del valor de otra, llamada entropía condicional, se define como:

$$H(X|Y) = H(X,Y) - H(Y) \tag{2.3}$$

2.2.2. Mutual Information

Información Mutua (Mutual Information, MI) es una medida utilizada en Teoría de la Información basada en la entropía de Shannon. Mide la reducción de incertidumbre de una variable aleatoria X, dado el conocimiento del valor de otra variable aleatoria Y.

$$MI(X,Y) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y) \tag{2.4}$$

Dado que $p(X = x, Y = y)$ es equivalente a $p(Y = y, X = x)$ vemos que, por definición de entropía conjunta, $H(X,Y) = H(Y,X)$, es decir la entropía conjunta es simétrica. En consecuencia y dado que la definición de MI involucra la suma de las entropías individuales y de la conjunta, vemos que también es válido que $MI(X,Y) = MI(Y,X)$, o sea MI es simétrica. Por otro lado, la entropía condicional no es simétrica, pues $H(X|Y)$ es la resta entre la entropía conjunta y la entropía de Y, mientras que $H(Y|X)$ es la resta de la conjunta con la entropía de X, por lo cual sólo son equivalentes cuando $H(X) = H(Y)$.

Mutual Information en proteínas

La Mutual Information puede ser utilizada para estimar la relación evolutiva existente entre pares de aminoácidos en una proteína. Dada una proteína podemos considerar a cada aminoácido como una variable aleatoria X y a cada columna del alineamiento múltiple con proteínas homólogas como el conjunto de observaciones de cada una de esas variables. De esta manera, un estimado de la entropía $H(X)$ se obtiene reemplazando las probabilidades $p(x_i)$ con la frecuencia observada de aminoácidos $f(x_i)$ en una columna del MSA. Un estimado similar puede ser utilizado para $H(X,Y)$, usando otra columna del MSA y la frecuencia de aparición de todos los pares ordenados (x_i, y_j) en las filas de ambas columnas analizadas. Luego, MI depende de $H(X)$ y $H(X,Y)$, por lo cual el estimado se obtiene trivialmente en base a ellos aplicando la ecuación 2.4. **Conceptualmente la MI entre dos columnas refleja el grado de correlación en el patrón de las dos columnas, es decir, refleja la relación presente entre dos aminoácidos dados de la proteína en base a la observación de aparición de aminoácidos en proteínas homólogas.**

2.2.3. Curva ROC

La Curva ROC (Receiver Operating Characteristic) es una representación gráfica de la sensibilidad frente a (1 – especificidad) para un sistema clasificador binario según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo).

Los resultados de un método utilizando el clasificador binario serán divididos en positivos y negativos (por encima o por debajo del umbral especificado), mientras que la observación de esos valores en la realidad los clasificará como verdaderos o falsos. De esta manera se obtienen cuatro conjuntos: verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN).

La sensibilidad de un método, para un dado umbral, se mide a través de la tasa de verdaderos positivos (TPR), la cual se define como la fracción de verdaderos positivos encontrados sobre todos los positivos que el método debió haber hallado (que incluye tanto a los TP

como a los FN).
Por otro lado, la tasa de falsos positivos (FPR) es la fracción de falsos positivos que el método halló sobre todos los negativos que debió haber encontrado (tanto los FP como los TN). La especificidad se mide como 1 - FPR.

Asimismo, se define el valor predictivo positivo (PPV) como la fracción de verdaderos positivos hallados sobre todos los positivos reportados por el método (sean estos verdaderos o falsos). De manera análoga, el valor predictivo negativo (NPV) se define como la fracción de verdaderos negativos sobre todos los negativos informados por el método (tanto verdaderos como falsos).
Los valores PPV y NPV permiten medir la eficacia diagnóstica del clasificador.

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN} \tag{2.5a}$$

$$PPV = \frac{TP}{TP + FP} \qquad NPV = \frac{TN}{TN + FN} \tag{2.5b}$$

La Curva ROC se obtiene graficando la tasa de verdaderos positivos (TPR) contra la tasa de falsos positivos (FPR) variando el umbral de discriminación.

El área de la curva ROC, llamada AUC (Area Under the Curve), se utiliza para medir la capacidad discriminativa del sistema clasificador (la habilidad de maximizar la detección de TP y minimizar la de FP).
Se dice que el clasificador es aleatorio cuando el valor del área de la curva es 0.5, pues es el área del triángulo determinado por la recta $y = x$ y los ejes, es decir para cada valor del umbral el clasificador encuentra la misma cantidad de TP y FP. Esta línea se llama de no-discriminación, pues en este caso el método es incapaz de discriminar verdaderos de falsos.
En consecuencia, un clasificador tendrá mayor capacidad discriminativa en la medida que su curva ROC se ubique lo más lejos posible por encima de la línea de no-discriminación (maximizando la TPR y minimizando la FPR), lo que es equivalente a que su área sea mayor. Asimismo, una curva ROC por debajo de la línea de no-discriminación indica que el clasificador obtiene peores resultados que el azar.

2.3. Algoritmo

En el presente trabajo desarrollé un algoritmo para el cálculo de Coevolución en proteínas basado en los conceptos de entropía de Shannon y los trabajos de Dunn et al [4, 5] y Marino Buslje et al [6]. Además de calcular la coevolución para un conjunto de familias de proteínas, realizo diversos cálculos estadísticos que enriquecen los resultados y permiten obtener información adicional.

2.3.1. Descripción del algoritmo

El algoritmo identifica pares de residuos que coevolucionan en familias de proteínas utilizando métodos de eliminación de ruido (generado por azar y por filogénesis). Para ello se basa en el cálculo de la Mutual Information entre pares de residuos y la corrección de la misma con Average Product Correction (APC) y Low Count Correction (LCC, [6]).

Los datos de entrada que el algoritmo requiere son:

- uno o más MSAs (donde cada uno es un archivo en formato FASTA[1])
- para cada MSA, un archivo con las distancias en la estructura terciaria entre cada par de aminoácidos de la secuencia de referencia. Como veremos más adelante, está información será utilizada para estimar la capacidad predictiva del algoritmo.

Para cada MSA se calcula la Mutual información de cada par posible de aminoácidos mediante el cálculo de la entropía conjunta de los mismos. Esta MI se corrige con APC y LCC y se obtiene el Z score asociado. Luego esta información se utiliza para realizar diversos cálculos estadísticos.

Recordemos que para validar que la Mutual Information calculada refleje la coevolución subyacente entre los pares de aminoácidos nos basamos en la premisa que asevera que aquellos residuos que están en contacto (es decir, existe alguna interacción a nivel molecular debido a su cercanía espacial) tienen propensión a coevolucionar. Se define entonces la *distancia de contacto* como aquella a la cual dos residuos interaccionan. En la sección **Análisis de la distancia entre residuos** se ahonda este tema y se explican los distintos métodos para definir distancia de contacto. Entonces, utilizando esta distancia de contacto, se validan los Z scores calculados y la distancia entre pares de aminoácidos en la estructura terciaria con el objetivo de verificar si la coevolución calculada condice con la cercanía en distancia.

Como resultado el algoritmo obtiene:

- MI y Z score de cada par de aminoácidos (junto con otros datos estadísticos como H, H conjunta, media y desvío standard)
- Gráficos estadísticos: ROC, distribución, Rank, web logos, sensibilidad vs cobertura, contact network, entre otros.

El algoritmo es parte de un conjunto de scripts programados en Perl, los cuales lo ejecutan para un conjunto de MSAs, realizan cálculos estadísticos y permiten configurar la ejecución con diversos parámetros.

La estructura de los scripts es la siguiente:

1. run_mi_dir.pl: punto de entrada. Recorre una carpeta determinada y ejecuta el algoritmo para cada familia (archivo FASTA) allí contenida.

[1] FASTA es un formato de texto que permite enumerar una o más secuencias de aminoácidos (o nucleótidos). Consiste de una línea de encabezado por cada secuencia seguida por una o más líneas donde se escriben de manera consecutiva los residuos de la secuencia usando el código de una letra.

2. before_exec.pl: se asegura que se cumplan los requisitos previos para la ejecución.
3. MIp_wrapper.pl: encapsula la ejecución del algoritmo para una determinada familia.
4. MI.pl: realiza el cálculo de la MI, a través del cálculo de la entropía y las frecuencias de pares de aminoácidos.
5. peerZ.pl: aplica la corrección APC a la MI, y calcula la media, el desvío standard y el Z score)
6. graphZ.pl: genera graficos de conectividad de los pares de aminoácidos.
7. get_stats.pl: ejecuta el cálculo de las métricas estadísticas.
8. after_exec.pl: se ejecutan cálculos estadísticos que requieren la finalización de todas las familias.
9. Aminoacid_alphabet.pm: clase para el manejo de alfabetos de aminoácidos.
10. Contact_distance.pm: clase para el manejo de distancia de contacto.
11. Coev_Statistics.pm: clase para el cálculo de diversas métricas estadísticas.
12. roc_auc.pm: clase para el cálculo de curvas ROC.
13. Structure_adjustment.pm: clase para ajustar los índices entre pdb y fasta.
14. incremental_coverages.pl: genera el grafico de cobertura para todas las familias adicionando pares detectados en alfabetos reducidos.

2.3.2. Low Count Correction

Low Count Correction (LCC), es una corrección propuesta por Marino Buslje et al [6], la cual reajusta el cálculo de la entropía de manera de eliminar el sesgo debido a la falta de datos suficientes. Esta consiste en normalizar la frecuencia de aparición de los residuos (con la que se aproxima la probabilidad en el cálculo la entropía) tal como se expone en las ecuaciones 2.6, donde $freq(x_i, y_j)$ significa la frecuencia de aparición del aminoácido x en la posición i al mismo tiempo que y está en la posición j. La constante λ es quien logra lidiar con la falta de datos, al influir significativamente cuando hay pocas observaciones y al diluirse en muestreos más grandes.

$$freq_{LCC}(x_i, y_j) = \frac{\lambda + freq(x_i, y_j)}{N_{ij}} \; donde \; N_{ij} = \sum_{x_i, y_j} \lambda + freq(x_i, y_j) \tag{2.6a}$$

$$freq_{LCC}(x_i) = \sum_{y} freq_{LCC}(x_i, y) \tag{2.6b}$$

2.3.3. Average Product Correction

La MI entre un par de residuos en una proteína está compuesta por información mutua debida a interacciones estructurales, a restricciones funcionales, a ruido aleatorio y a la herencia de ancestros comunes [7]. Average Product Correction (APC) es una corrección propuesta por Dunn et al [4] para separar la señal deseada (la causada por restricciones funcionales y estructurales, denotada MI_{sf}) del ruido de fondo (MI_b, compuesto por el ruido aleatorio y la herencia en común). Para ello postulan una fórmula que aproxima MI_b entre dos posiciones, asumiendo que éste está dado por el producto de la propensión de cada posición, relacionada con su entropía y su historia filogenética, hacia MI_b.

$$APC(x_i, y_j) = \frac{MI(x_i, \overline{a})MI(y_j, \overline{a})}{\overline{MI}} \tag{2.7}$$

donde $MI(x_i, \overline{a})$ denota la media de la MI de la columna i y todas las demás columnas, y $\overline{MI}$ es la MI media total entre todos los pares de columnas.

Restando este componente de la MI total se logra una aproximación de MI_{sf}, denominada MI_p.

$$MI_p(x_i, y_j) = MI(x_i, y_j) - APC(x_i, y_j) \tag{2.8}$$

También se contempla la posibilidad que las propensiones sean aditivas en vez de multiplicativas (ASC, Average Sum Correction).

$$ASC(x_i, y_j) = MI(x_i, \bar{a}) + MI(y_j, \overline{a}) - \overline{MI} \tag{2.9}$$

2.3.4. Alfabetos reducidos

Una de las variables que decidimos explorar en esta tesis consiste en utilizar alfabetos reducidos: un alfabeto reducido es una simplificación del alfabeto original (en este caso los 20 aminoácidos) que agrupa dos o más caracteres en uno, lo que implica que sean el mismo a nivel interpretativo. Los alfabetos reducidos nos permiten explorar otros enfoques: la coevolución puede emerger debido a alguna característica de los residuos involucrados (tamaño, estructura, polaridad, carga, etc), y no necesariamente a los residuos específicos. Consideramos 5 alfabetos reducidos (ver Tabla 2.1) y se pueden agregar nuevos fácilmente mediante una expresión regular.

En el primer alfabeto (alpha1) se agrupan los aminoácidos por su características físico-químicas principales, pero siendo muy estricto, ya que sólo se agrupan los alifáticos I, L de similar tamaño, dejándose los aromáticos fuera de la agrupación. Los reemplazos son:

- reducción de R con K, dado que ambos son básicos
- reducción de E con D, dado que ambos son ácidos con carga negativa
- reducción de T con S, por ser ambos alcoholes

Tabla 2.1: Alfabetos reducidos

• all = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}
• alpha1 = {A, C, D, F, G, H, I, K, M, N, P, S, V, W, Y} *donde* $(R \to K)$, $(E \to D)$, $(T \to S)$, $(L \to I)$, $(Q \to N)$
• alpha2 = {A, C, D, F, G, H, K, M, N, P, S, W, Y} *donde* $(R \to K)$, $(E \to D)$, $(T \to S)$, $(Q \to N)$, $(\{I, L, V\} \to A)$
• alpha3 = {A, C, D, F, G, H, N} *donde* $(\{R, K\} \to H)$, $(E \to D)$, $(\{T, S\} \to C)$, $(\{W, Y\} \to F)$, $(\{I, L, M, P, V\} \to A)$, $(Q \to N)$
• alpha4 = {A, C, D, F, H, I, K, N, S} *donde* $(\{G, P\} \to A)$, $(T \to S)$, $(E \to D)$, $(R \to K)$, $(\{W, Y\} \to F)$, $(\{L, M, V\} \to I)$, $(Q \to N)$
• alpha5 = {A, C, D, F, G, H, I, K, M, N, P, S} *donde* $(E \to D)$, $(R \to K)$, $(T \to S)$, $(Q \to N)$, $(\{W, Y\} \to F)$, $(\{L, V\} \to I)$

- reducción de L con I, por ser ambos alifáticos y muy similares (I es una isoforma de L)
- reducción de Q con N, por tener ambos grupo amida

La longitud de este alfabeto es de 15 aminoácidos.

El segundo alfabeto (alpha2) es una extension de alpha1 con la diferencia que se agrupan más alifáticos. Ademas de los reemplazos de alpha1 se cambia L x I con:

- reducción de I, L y V con A, por ser todos alifáticos

La longitud de este alfabeto es de 13 aminoácidos.

El tercer alfabeto (alpha3) se base en alpha1 pero se profundiza en las reducciones, se realizan reemplazos en base a características secundarias por lo cual se agrupan todos los alifáticos y todos los aromáticos:

- reducción de E con D, igual que en alpha1
- reducción de Q con N, igual que en alpha1
- reducción de R y K con H, son los tres básicos, se agrega H, dado que a veces se lo encuentra como HisH+, por lo cual tiene carga positiva y es muy polar

- reducción de T y S con C, los tres son polares, C tiene un grupo tiol el cual tiene similitudes funcionales con el grupo alcohol de T y S
- reducción de W e Y con F, dado que los tres son aromáticos
- reducción de I, L, M, P y V con A, por ser todos alifáticos, aunque de diferente tamaño

La longitud de este alfabeto es de 7 aminoácidos.

El cuarto alfabeto (alpha4) es una extension de alpha1, conservadora como alpha2, con pocos reemplazos. A los cambios de alpha1 se adicionan los aromáticos y los alifáticos se dividen por tamaño:

- reducción de W e Y con F, dado que los tres son aromáticos
- reducción de G y P con A, por ser todos alifáticos y pequeños, por lo cual no tienen impedimento estérico
- reducción de L, M y V con I, por ser todos alifáticos y de tamaño similar (un poco más grandes que los anteriores)

La longitud de este alfabeto es de 9 aminoácidos.

Por último, el quinto alfabeto (alpha5) es una extension de alpha1 más conservadora aún. A los cambios de alpha1 se adicionan los aromáticos y no se explora demasiado con los alifáticos:

- reducción de W e Y con F, dado que los tres son aromáticos
- reducción de L y V con I, por ser todos alifáticos y de tamaño similar

La longitud de este alfabeto es de 12 aminoácidos.

2.3.5. Análisis de la distancia física entre residuos en la estructura de la proteína

Con el objetivo de analizar la relación entre la MI de cada par de residuos y su distancia en la estructura terciaria de la proteína, obtuvimos el archivo *PDB* (Protein Data Bank, el cual contiene la disposición de los residuos en el espacio tridimensional) de la secuencia de referencia de cada familia. Para poder relacionar la estructura terciaria (el archivo PDB) con la estructura primaria (la secuencia en el MSA), nos aseguramos que en cada MSA la primera secuencia (que es la tomada como referencia) sea la misma que está representada por el PDB.

Para obtener las distancias entre los residuos de cada PDB, creamos un script en el lenguaje de programación Tcl, el cual es interpretado por el software VMD y retorna un archivo con la distancia entre todos los pares (se puede utilizar el centro geométrico o la distancia entre átomos no-hidrógeno).

Definición de residuos en contacto físico

Definimos que dos residuos están en contacto físico si se encuentran a una distancia tal que exista entre ellos alguna interacción a nivel atómico. Podemos entonces considerar que todo par de residuos en contacto tendrá cierta disposición a coevolucionar debido a estas interacciones (cabe aclarar que no podemos afirmar que toda coevolución se deba a estas interacciones, y por lo tanto, pueden también existir pares que no estén en contacto y coevolucionen).

Una convención muy utilizada considera que dos residuos están en contacto si sus centros geométricos se encuentran como máximo a una distancia de 8 Å. Dado que los aminoácidos difieren en tamaño y no tienen estructuras regularmente esféricas, tomar una medida fija para todos los residuos resulta conveniente pero, al mismo tiempo, puede conllevar una pérdida de precisión en los resultados. Por este motivo decidimos experimentar diversas alternativas para determinar la distancia a la que consideramos que dos residuos están en contacto:

fijo geométrico para todo par de residuos utilizamos un valor fijo de distancia (en particular, 8 y 10 Å), que se mide entre sus centros geométricos.

fijo no-hidrógeno para todo par de residuos utilizamos un valor fijo de distancia (en particular, 6 Å), que se mide entre sus átomos no-hidrógeno más cercanos. Es decir, dos residuos están en contacto si al menos un par de átomos no-hidrógeno están en contacto.

estadístico geométrico para cada par de residuos utilizamos un valor de distancia distinto de acuerdo a los aminoácidos en cuestión, medido entre sus centros geométricos y determinado por un cálculo estadístico.

Para obtener los valores del método que denominamos *estadístico geométrico* creamos un script Perl y una base de datos MySQL en los cuales calculamos la distribución de distancia entre todo par posible de aminoácidos de una muestra de 1000 secuencias tomadas al azar de la base de datos PFAM [8, 9]. Los gráficos de estas distribuciones, las cuales normalizamos dividiendo cada valor por la superficie de la esfera generada al tomar el mismo como radio, diviendo por la densidad y fijando que la integral de la curva sea igual a 1, nos permitieron estimar la distancia más probable para cada par de aminoácidos (dado que son 20 los aminoácidos y las relaciones son simétricas, obtuvimos 200 gráficos; en el gráfico 2.2 se muestra un ejemplo). Para cada gráfico obtuvimos dos distancias: la más probable (considerada como el primer mínimo local que aparece luego del máximo global, a la cual llamamos *strict*) y una menos estricta (tomada como el punto donde finaliza la curva del máximo global, a la cual denominamos *not strict*). Con estos valores armamos una matriz de distancias entre aminoácidos, con los valores estrictos ubicados en la mitad superior de la misma (ver Tabla 2.2).

Por otro lado, agregamos más opciones para normalizar las distribuciones: dividir cada valor por el volumen de la esfera, dividir cada valor por la background distribution del segundo aminoácido, y dividir por la distribución aleatoria de obtener cualquier residuo a una distancia dada tomando el segundo aminoácido como referencia (por promedio o sumatoria).

Fig. 2.2: Distribución de Distancia ALA - GLN

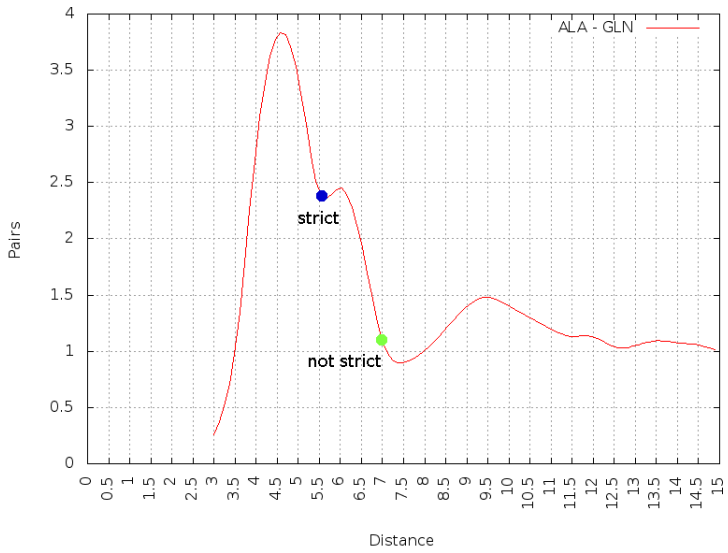

Gráfico de distancias de ALA y GLN

Tabla 2.2: Matriz de distancias entre aminoácidos

	ALA	ARG	ASN	ASP	CYS	GLN	GLU	GLY	HIS	ILE	LEU	LYS	MET	PHE	PRO	SER	THR	TRP	TYR	VAL
ALA	4	6.5	5	5	4.5	5.5	5.5	4	7	5	5	7	5.5	7	4.5	4	4.5	7.5	7.5	4.5
ARG	6.5	8	7	6.5	7	8	7.5	7	7.5	7	7.5	8	8.5	8	7	7	7	6	7.5	7
ASN	6.5	7	6.5	6	5.5	6	6	5	6.5	5.5	6	6.5	6	7	5	6	6.5	7.5	7	6.5
ASP	6.5	6.5	6	6	5	6	6	5	6.5	5.5	6	6	6	7	5	5.5	5.5	8	8	5.5
CYS	4.5	7	6	6	6	6	6	4	6	5	6.5	6.5	6.5	7	4.5	4.5	4.5	6.5	7	4.5
GLN	7	8.5	6.5	7	7	8	6.5	6	6.5	7	7	7	7	8.5	6.5	6	8	6.5	7	7.5
GLU	7	7.5	6.5	7	7	8.5	8	5.5	7	6	6	7	6.5	8.5	6	5.5	6	8.5	7.5	6.5
GLY	4	8	6	6	4.5	7.5	8	4	6.5	4.5	6	6	5.5	7.5	5	4	5	6.5	6.5	4.5
HIS	7	9.5	7	7.5	7	8	8.5	7	7	7	7	6.5	6.5	7.5	6	6	6	8	7	7
ILE	5	8.5	7	7	7	8	8	7.5	8	7.5	8	8	7.5	8.5	7.5	7.5	8	8.5	8	7.5
LEU	6.5	9	7	7	7	8	8	7.5	8	8	9	8	8	8	8	8	8	9	8.5	8
LYS	7	8.5	7	7	7.5	8.5	8.5	8.5	7	8	8	8	8	8.5	8	7.5	7.5	8.5	6.5	8
MET	7	8.5	6.5	7	7	8	8	7	6.5	7.5	8	8	8	8.5	8	8	8	8	9	8
PHE	7	9.5	8	8	8	8.5	8.5	7.5	7.5	8.5	8	8.5	8.5	8	8	8	8	8.5	8	8.5
PRO	4.5	8.5	6	6	5	8.5	8	7	8	7.5	8	8	8	8	5	5	5	7.5	8.5	4
SER	4	7	6.5	6	5.5	7.5	7.5	5	7	7.5	8	7.5	8	8	5	4.5	4.5	8.5	7.5	4.5
THR	4.5	8	7	6.5	4.5	8	7.5	5	6	8	8	7.5	8	8	5	4.5	5.5	7.5	8.5	4.5
TRP	7.5	8	8	8	7	9.5	9.5	8.5	8	8.5	9	8.5	8	8.5	7.5	8.5	7.5	8.5	8.5	8.5
TYR	7.5	8.5	8	8	8	8.5	7.5	8	9.5	8	8.5	8	9	8	8.5	7.5	8.5	8.5	7	8.5
VAL	4.5	8.5	7	6.5	5	7.5	8	5.5	8	7.5	8	8	8	8.5	4	4.5	4.5	8.5	8.5	8

Cabe aclarar que al utilizar alfabetos reducidos con el método *estadístico geométrico* se deben considerar las distancias entre todos los apareamientos posibles de los residuos representados por el par en cuestión (para lo cual se debe revertir la traducción), y finalmente quedarse con un valor de distancia representativo. En particular, decidimos utilizar la máxima distancia obtenida, pues consideramos que de esta manera representamos a todos los pares posibles.

2.3.6. Complejidad del algoritmo

Para determinar la complejidad del algoritmo analizaremos la implementación de la funcionalidad principal, la cual consiste en el cálculo de la entropía, la información mutua y la

APC de cada par de residuos. El tamaño de la entrada del algoritmo depende básicamente de dos variables: la cantidad de secuencias del MSA (a la cual llamaremos m) y la longitud de la secuencia de referencia (a la que llamaremos n), medida por la cantidad de residuos excluyendo gaps. Si bien el algoritmo también depende de la longitud del alfabeto utilizado, como este no puede crecer asintóticamente podemos considerarlo como una constante (k, acotada por 20, la cantidad de aminoácidos posibles).

Algoritmo 1: Cálculo de Mutual Information con APC y LCC

```
Input: MSA, ref-seq, alphabet

foreach sequence s of MSA do                    // count residue pairs ocurrences
    foreach residue r1 of ref-seq do
        count r1 occurrence;
        foreach residue r2 of ref-seq do
            count r1 and r2 occurrence;

foreach residue r of ref-seq do                                  // get H and MI
    foreach aminoacid a in alphabet do
        calculate r entropy and frequency;

foreach residue r1 of ref-seq do                                 // get N for LCC
    foreach residue r2 of ref-seq do
        foreach aminoacid pair (a, b) in alphabet do
            calculate N value for LCC;

foreach residue r1 of ref-seq do                            // get joint H and MI
    foreach residue r2 of ref-seq do
        foreach aminoacid pair (a, b) in alphabet do
            calculate co-frequency of r1 and r2;
        calculate joint entropy and mutual information of r1 and r2;

foreach residue r of ref-seq do                                        // get mean
    calculate mean of r;

foreach residue r1 of ref-seq do                              // get APC and MIp
    foreach residue r2 of ref-seq do
        calculate APC and get MIp;
```

Como podemos observar en el pseudo-código (ver Algoritmo 1), el cálculo de la MI de cada par de residuos consta de una serie de ciclos sobre la cantidad de secuencias en el MSA (m) y la cantidad de residuos en la secuencia de referencia (n). Acotando la suma de estos ciclos concluimos que el algoritmo tiene el orden de complejidad polinomial $\mathrm{O}(mn^2)$.

$$f(n,m) \in \mathrm{O}\big(mn^2 + nk + n^2k^2 + n^2k^2 + n + n^2\big) \in \mathrm{O}\big(mn^2\big) \tag{2.10}$$

Análisis del tiempo de ejecución

Complementamos el cálculo teórico de la complejidad del algoritmo con el análisis del tiempo de ejecución del algoritmo en función del tamaño de la entrada (tanto en n y m). Para ello utilizamos los alineamientos artificiales creados por Gouveia-Oliveira [10], adaptando la cantidad de secuencias de los alineamientos y la longitud de la secuencia de referencia de cada uno para contar con un conjunto de datos significativo.

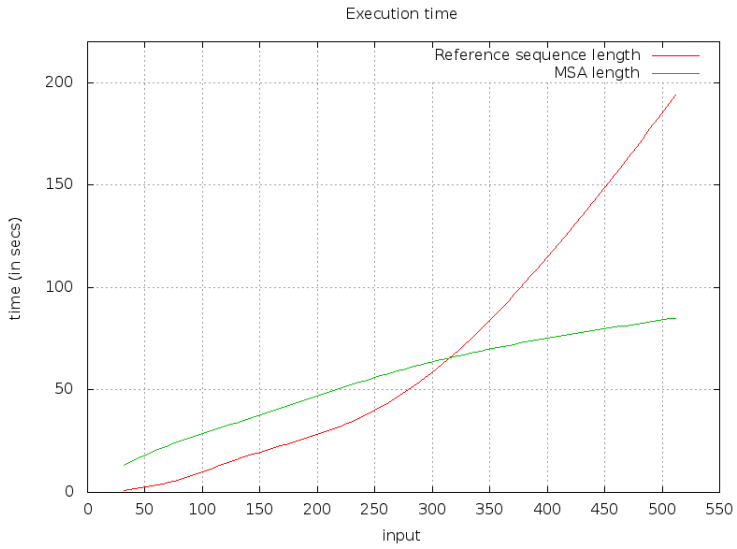

Fig. 2.3: Tiempo de ejecución en función del tamaño de la entrada

Como podemos observar en el gráfico 2.3. comprobamos experimentalmente la complejidad previamente enunciada: el tiempo de ejecución varía linealmente sobre la cantidad de secuencias en el MSA, m, y cuadráticamente en la longitud de la secuencia de referencia, n. Esto indica que el algoritmo es relativamente eficiente y aborda el problema en un tiempo de ejecución razonable (para datos de entrada de longitud razonable).

2.3.7. Análisis de datos

Además de calcular la coevolución de los pares de residuos en un MSA con diversos parámetros, el algoritmo desarrollado genera una serie de gráficos y parámetros estadísticos que permiten analizar su capacidad predictiva y comparar el impacto de las variables estudiadas.

Análisis de identidad de secuencias

El grado en que las secuencias se asemejan está relacionado cualitativamente con la distancia evolutiva entre ellas. Una alta identidad entre dos secuencias sugiere que tienen un ancestro común más reciente, mientras que una baja identidad sugiere que la divergencia es más remota. Dentro de un MSA pueden existir secuencias redundantes, tanto idénticas como con un alto grado de similitud. Estas secuencias no sólo implican un costo extra

de procesamiento sino que además pueden alterar el resultado al aumentar la frecuencia de aparición de sus residuos por la sobrerepresentación de los mismos (por consiguiente impactan sobre la entropía, que se basa en la frecuencia, y en la Mutual Information, que se basa en la entropía).

Para analizar el impacto de la identidad de secuencias de un MSA se generaron grupos para cada familia (llamados clusters) en base a la similitud de identidad entre las secuencias que los conforman. Por lo tanto, cada una de las secuencias del MSA resultante es representante de sólo uno de los clusters, y cada cluster es representado sólo por una secuencia.

Para armar los clusters utilizamos el software CD-HIT [11], utilizando valores entre 65 y 90 % de similitud entre secuencias (aumentando cada 5 %).

A modo de ejemplo para ilustrar la identidad de secuencias, en la Figura 2.4 vemos un MSA con 6 secuencias y los distintos clusters que se conforman variando el porcentaje de identidad de secuencias aceptado para agruparlas por semejanza.

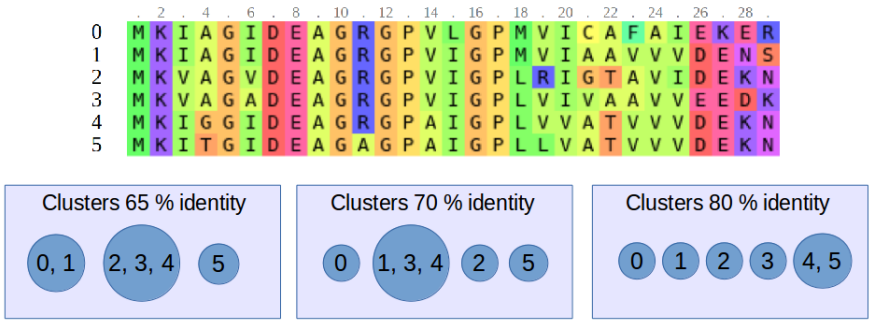

Fig. 2.4: Ejemplo de clusterización de MSA: se muestra un MSA de 6 secuencias y se esquematizan los clusters resultantes con diferentes porcentajes de identidad de secuencias. Los círculos representan los clusters obtenidos para el porcentaje de identidad indicado y su contenido es el id de las secuencias según la tabla. Por ejemplo, para 65 % de identidad se obtienen tres clusters, el primero con las secuencias 0 y 1, el segundo contiene las secuencias 2, 3 y 4, y el último sólo a la secuencia número 5.

Precisión y Cobertura

El algoritmo analiza la precisión y la cobertura del método (accuracy y coverage) para cada familia y cada cluster, y retorna un gráfico con las dos curvas que los representan. La precisión está dada por el porcentaje (o cantidad) de pares identificados como en coevolución y que están en contacto, respecto del total de pares que coevolucionan; mientras que la cobertura está dada por el porcentaje (o cantidad) de pares identificados como en coevolución y están en contacto, respecto del total de pares en contacto. Se considera que un par de residuos está en contacto si se encuentra dentro de una determinada distancia (de acuerdo al método de distancia seleccionado), y se considera que coevolucionan si su valor de coevolución (MIp o Z score) es mayor o igual a un valor de corte arbitrario (que varía en el eje x).
La precisión se calcula con la TP Rate y la cobertura con el PPV, ambas explicadas anteriormente junto a la Curva ROC (ver Figura 2.5).

Distribución

El algoritmo calcula la distribución de todos los pares en contacto y no en contacto variando el valor de Z score considerado como corte, y retorna un gráfico con ambas curvas para cada MSA y cada cluster. Para calcular la distribución se obtiene la cantidad de pares que están dentro de cada rango de Z score (determinado por buckets de 0.5). Minimizar la intersección entre ambas distribuciones implica obtener resultados más precisos.

Análisis de la precisión mediante rangos

El algoritmo realiza un análisis que puede ser utilizado para comparar la precisión del método en diferentes condiciones, a través del cálculo de la proporción de pares que identifica correctamente como que coevolucionan (usando la distancia de contacto como indicador), considerando subconjuntos de pares con los mayores Z scores. Para ello de cada familia se obtienen los n pares con mayor Z score y se obtiene qué fracción de estos están en contacto (con n = 1, 2, 3, 4, 5, 10 y 20), se promedian todas las fracciones del mismo valor de n de todas las familias y se grafica el resultado para cada n.

Curva ROC

El algoritmo genera la Curva ROC y calcula el AUC de cada familia y de todas la familias juntas. Se utiliza un valor específico de distancia de contacto de los aminoácidos (determinado por el método de distancia seleccionado) como medida para determinar la veracidad del par, y se varía sobre el Z score para determinar si el método lo reconoce como positivo.

Otros gráficos

El método genera otros gráficos adicionales, entre los que cabe destacar:

- **Weblogo** [12]: para cada MSA recibido como entrada se genera un weblogo, el cual es una representación gráfica del MSA desarrollada por Tom Schneider y Mike Stephens. Un logo consiste en una serie de símbolos apilados en cada posición de la secuencia. La altura total de cada pila indica la conservación de la secuencia en esa posición, mientras que la altura de cada símbolo indica la entropía (o la frecuencia relativa) de cada aminoácido en esa posición.
- **Cytoscape** [13]: se generan dos grafos de conectividad para cada MSA y cada cluster, uno de MI y otro de contactos, los cuales se pueden comparar en el programa Cytoscape (obteniendo la intersección y la diferencia) con el objetivo de visualizar las diferencias entre residuos en contacto y residuos coevolucionando.

3. RESULTADOS

3.1. Cómo entender los resultados de MI

Resultado e interpretación de la MI para una familia de proteínas

El resultado principal del algoritmo es una tabla donde se indica la Mutual Information y el Z score para cada par posible de aminoácidos de la familia dada, ordenada decrecientemente por Z score (ver Tabla 3.1). Cada par de aminoácidos se representa en una fila, donde se detallan la posición de cada uno en la secuencia, la letra del residuo correspondiente a cada uno, la entropía (individual y conjunta), la Mutual Information y el Z score.

Tabla 3.1: Extracto del resultado del algoritmo para la familia X1DF

pos_1	pos_2	aa_1	aa_2	H_1	H_2	H_{12}	MI_{12}	MIp	Z
109	222	L	V	0.44028	0.40641	0.63346	0.21322	0.11517	6.40256
90	241	Y	A	0.60066	0.64543	0.87219	0.37391	0.11243	6.25000
179	315	G	D	0.31419	0.31267	0.42764	0.19923	0.11141	6.19321
151	308	G	K	0.57182	0.70761	0.92002	0.35941	0.10573	5.87695
115	141	T	N	0.44618	0.56224	0.73764	0.27078	0.10398	5.77951
369	379	I	A	0.59174	0.48903	0.82786	0.25291	0.09964	5.53786

con mean MI = 0.00018 y stdev MI = 0.01796.

Relacionando el Z score obtenido con la estructura primaria de la proteína puede observarse que la cercanía en secuencia no es una condición necesaria para la coevolución entre pares de aminoácidos. Esto se debe a que la interacción entre los aminoácidos se da a nivel tridimensional, por lo cual lo relevante es la distancia en la estructura terciaria de la proteína. Por otro lado, cabe aclarar que uno de los parámetros del algoritmo permite ignorar pares muy cercanos en secuencia (menos de 5 aa de distancia) con el objetivo de descartar pares que coevolucionan trivialmente (por ejemplo, por formar parte de un mismo giro de las alfa-hélices).

Se muestra un ejemplo en las Figuras 3.1 y 3.2, en la cuales puede observarse que si bien el par se encuentra distante en secuencia (a 154 aminoácidos), está en contacto en estructura y su Z score es relativamente alto (muy cercano a 4).

En los diagramas tridimensionales de las proteínas los residuos se representan con la estructura de sus átomos y sus uniones, mientras que las alfa hélices están dibujadas de manera tubular y las hojas beta como flechas anchas.

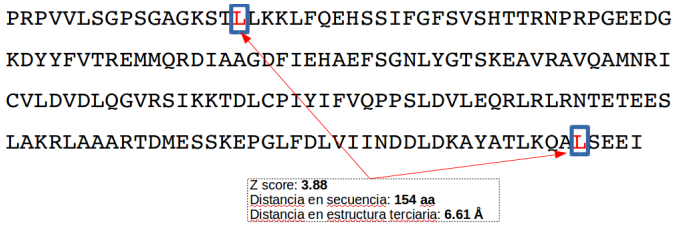

Fig. 3.1: Secuencia de referencia de la familia de proteínas Guanylate kinase (PF00625), se indica el Z score, la distancia en secuencia y en la estructura terciaria de un par de aminoácidos de tipo leucina

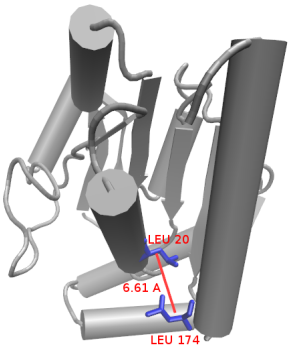

Fig. 3.2: Estructura terciaria de la proteína 1LVG de la familia Guanylate kinase, se indica la distancia del mismo par de residuos resaltado en la estructura primaria.

3.1.1. Resultado relacionado con la estructura terciaria

En esta sección se muestran ejemplos de estructuras terciarias en las cuales se encuentran pares que son TP, FP, FN o TN, dado un umbral de corte de Z score arbitrario igual a 4 y utilizando el método de distancia de contacto que denominamos *strict*.

En la Figura 3.3 se muestra la proteína GTP Ciclohidrolasa I, resaltando un par de aminoácidos que está en contacto y coevoluciona (o sea, es un par True Positive). Su distancia de contacto es 6,67 Å (menor a la distancia Phe-Phe, según matriz estricta, de 8 Å) y su Z score es 4.275 (mayor al umbral).

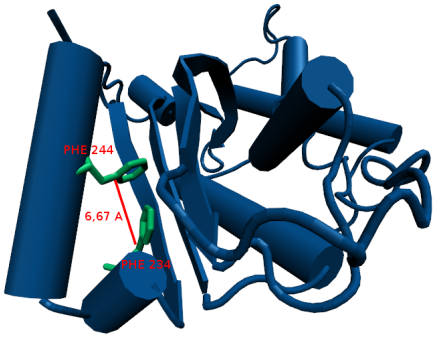

Fig. 3.3: Estructura terciaria de la proteína GTP Ciclohidrolasa I. Se resaltan dos residuos, Phe 234 y Phe 244, y su distancia desde el centro de masa.

Para ejemplificar un par de residuos FP (el método determina que coevoluciona pero no está en contacto), en la Figura 3.4 se muestra la proteína NDP Kinasa de Myxococcus xanthus. Su distancia de contacto es 8.46 Å (mayor a la distancia Leu-Phe, según matriz estricta, de 8 Å) y su Z score es 6.072 (mayor al umbral).

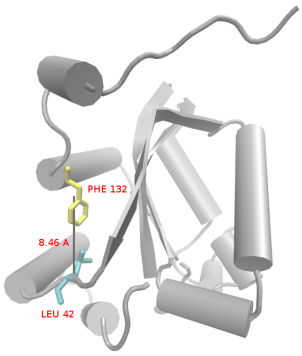

Fig. 3.4: Estructura terciaria de la proteína NDP Kinasa. Se resaltan dos residuos, Leu 42 y Phe 132, y su distancia desde el centro de masa.

Por otro lado, un ejemplo de un par de aminoácidos FN (no coevoluciona pero está en contacto) lo encontramos en la proteína Peptidil-tRNA hidrolasa de la E. Coli, esquematizado en la Figura 3.5. Su distancia de contacto es 5.69 Å (menor a la distancia Arg-Asp, según matriz estricta, de 6.5 Å) y su Z score es 3 (menor al umbral).

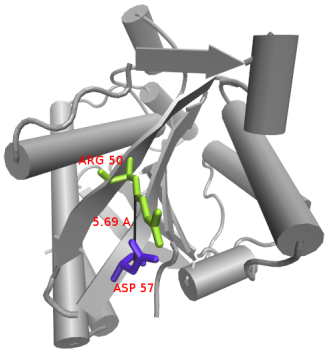

Fig. 3.5: Estructura terciaria de la proteína Peptidil-tRNA hidrolasa. Se resaltan dos residuos, Arg 50 y ASP 57, y su distancia desde el centro de masa.

Por último, un ejemplo TN (no coevoluciona y no está en contacto) lo vemos en la proteína Chorismate synthase, en la Figura 3.6. Su distancia de contacto es 12.02 Å (mayor a la distancia Leu-Tyr, según matriz estricta, de 8.5 Å) y su Z score es 2.19 (menor al umbral).

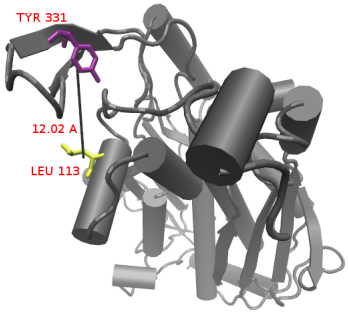

Fig. 3.6: Estructura terciaria de la proteína Chorismate synthase. Se resaltan dos residuos, Leu 113 y Tyr 331, y su distancia desde el centro de masa.

3.1.2. Distribución de valores de MI para todas las familias

En la Figura 3.7 se muestra la distribución de valores de Z para todos los pares de residuos, agrupados y normalizados para todas las familias, discriminada en dos curvas de acuerdo a si los pares están o no en contacto. Como puede observarse, las curvas tienen un alto solapamiento para valores bajos de Z score, y la curva de pares en contacto está desplazada hacia la derecha, lo que indica que hay mayor proporción de pares en contacto para valores altos de Z score.

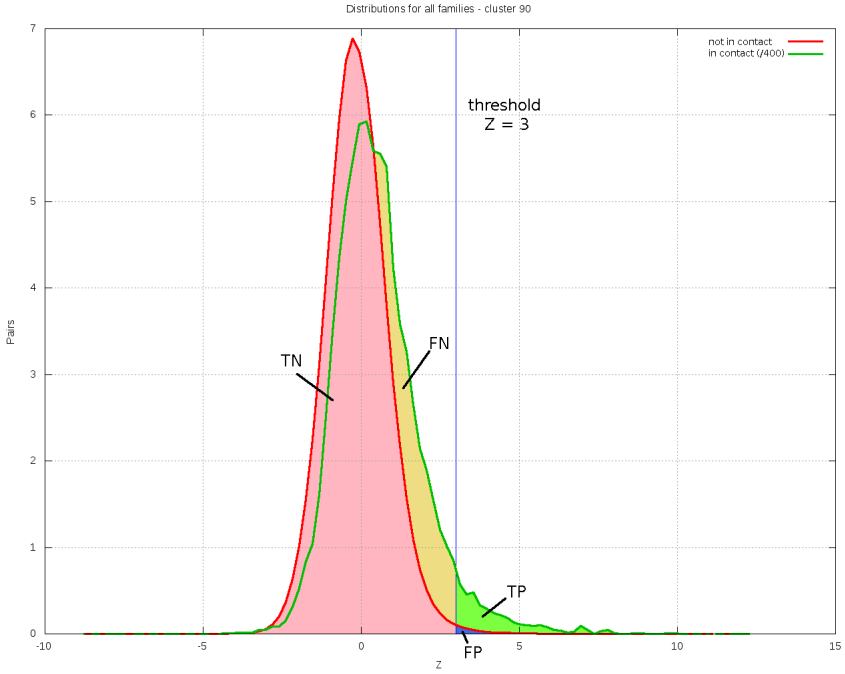

Fig. 3.7: Distribución de todas las familias variando el Z de corte utilizado tanto para pares en contacto como para aquellos que no lo están. Asimismo, utilizando un corte arbitrario (línea azul, Z=3) se resaltan las áreas donde se encuentran los TPs, FPs, TNs y FNs. Se indica también la normalización utilizada para modificar la escala de las curvas.

3.1.3. Curva ROC

En la Figura 3.8 se muestra la Curva ROC de la familia CD00657. Se utilizan los distintos valores de Z score como umbral de discriminación, el contacto entre los pares (de acuerdo al método de distancia de contacto usado) como medida de la observación en la realidad (en contacto = verdadero, no en contacto = falso), y se clasifica a los pares como positivos o negativos comparando su valor de Z score con el umbral de discriminación utilizado en cada paso.

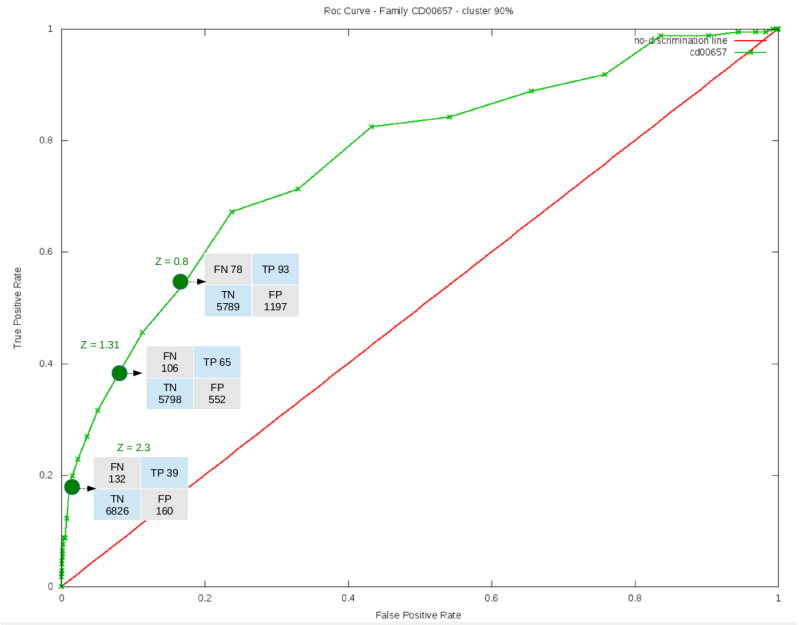

Fig. 3.8: Curva ROC de la familia CD00657. Para tres cortes arbitrarios de Z score se muestra la cantidad de TP, FP, TN y FN hallados. El valor de AUC es 0.766.

En la curva se observa que a medida que se incrementa la tasa de falsos positivos (FPR, eje x), se disminuye el valor de Z score representado. De esta manera se observa también que los valores de Z score representativos (mayores a 3) se encuentran muy cercanos al (0, 0). Esta situación se puede comprender claramente observando la figura 3.7, en la cual si movemos el corte de Z score podemos ver como varían y se relacionan los cuatro conjuntos (TP, FP, TN y FN):

- a medida que se incrementa el corte de Z score, algunos de los pares que se encontraban en el conjunto TP pasan a pertenecer al conjunto de FN, decrementando la aridad del primero y acrecentando la del segundo, pues están a distancia de contacto pero no logran alcanzar el umbral de Z score para ser considerados en coevolución.
- en análoga situación, el aumento del umbral de Z score, traslada pares del conjunto de FP a TN, pues estos no están en contacto y dejan de ser considerados en coevolución.
- en ambas relaciones, TP-FN y FP-TN, se observa que la cantidad de pares positivos (TP, FP) es significativamente inferior a la cantidad de pares negativos (FN, TN), especialmente para valores significativos de Z score (mayores a 3). Siendo que estos

últimos, FN y TN, se encuentran en el divisor de las ecuaciones TPR y FPR se deduce que los resultados de estas ecuaciones tienden a ser cercanos a 0 a medida que se incrementa el Z score analizado.

3.2. Análisis de los parámetros

En está sección se analizan los resultados obtenidos al variar los diferentes parámetros del algoritmo: LCC, Clusters por identidad de secuencia, Alfabetos reducidos y Distancia de contacto. Para analizar el efecto de los mismos en relación a su poder predictivo se analizaron las curvas ROC, la precisión (medida con el PPV) y la cobertura (medida con la TPR).

3.2.1. Análisis de la variación del parámetro LCC

En la Figura 3.9 se muestran las curvas ROC obtenidas con y sin el uso de la corrección LCC. Se observa que la curva que representa el método con el uso de LCC comienza por arriba de la curva del método sin LCC, y luego se invierte el orden quedando la segunda por arriba de manera significativa, incluso de manera tal que su AUC es mayor. Si bien esto pareciera indicar que sin el uso de LCC se obtienen mejores resultados, cabe destacar que justamente es en el tramo donde el uso de LCC da mejores resultados en donde se hallan los valores de Z score más altos (desde de la intersección de las curvas en adelante los valores de Z score son menores a 1) y por lo tanto más significativos.

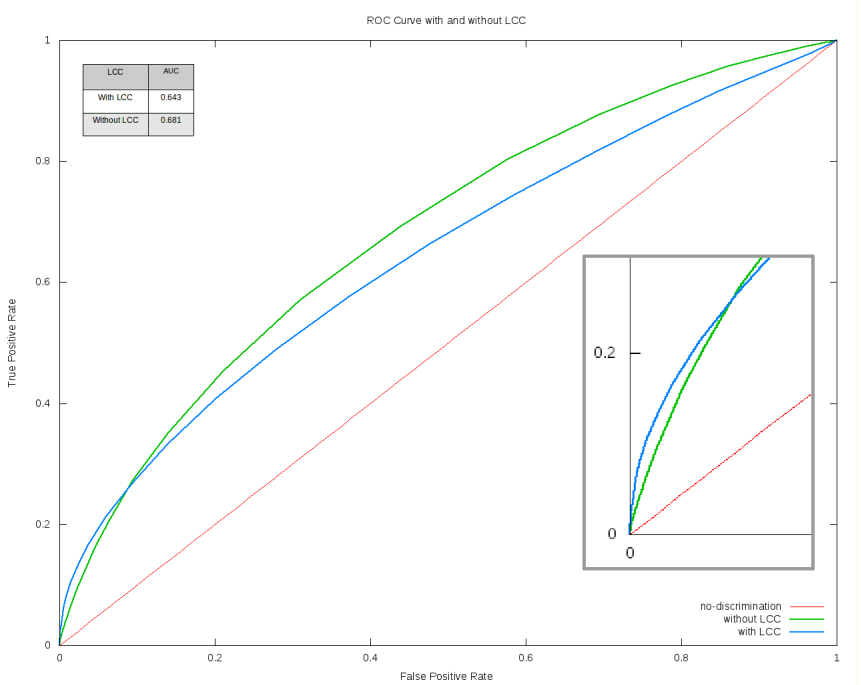

Fig. 3.9: Curvas ROC obtenidas con y sin el uso de la corrección LCC. En el área enmarcada se muestra el zoom del área más significativa, la cual contiene los valores más altos de Z score y donde se observa la intersección de las curvas.

Respecto a la cobertura (o sensibilidad, medida por la TPR), en la Figura 3.10 se observa que para ambos casos la misma es muy similar, con una leve predominancia para valores de Z score significativos (entre 3 y 5) con el uso de LCC. Por lo tanto, se concluye que para el rango representativo de Z scores, el uso de LCC tiene mayor cobertura de TP.

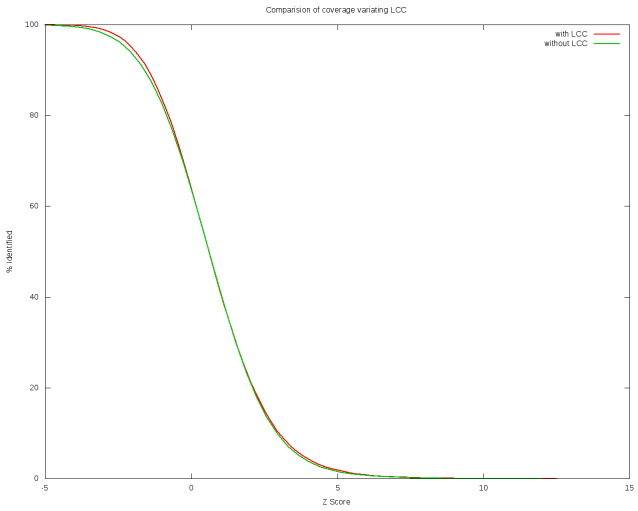

Fig. 3.10: Cobertura (TP Rate) con y sin el uso de la corrección LCC

En la Figura 3.11 se observa la precisión (medida con la PPV) de ambos casos, y puede verse que utilizando la corrección LCC el método incrementa su precisión significativamente. Asimismo, se observa que la curva que refleja el método sin LCC, para los valores medidos de Z score no logra alcanzar el 100 % de predicción; esto se debe a que incluso para valores muy elevados de Z score el método sin LCC sigue encontrando falsos positivos, mientras que con el uso de la corrección logra reconocerlos correctamente como verdaderos negativos (cabe la aclaración que para estos valores muy elevados, Z score más de 6, la cantidad de pares no excede los 4 o 5, en general).

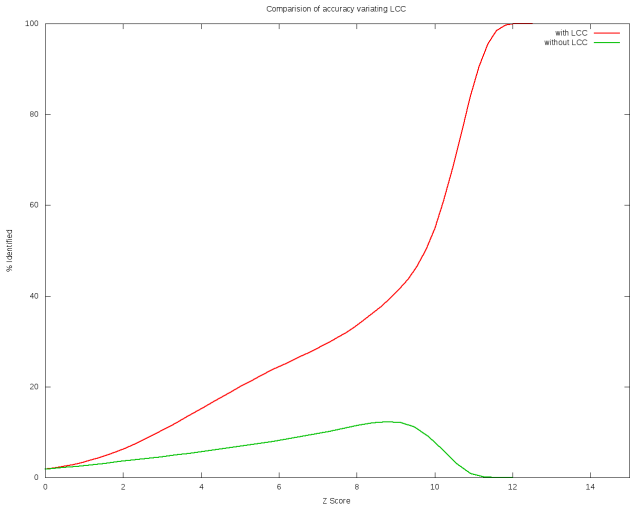

Fig. 3.11: Precisión (PPV) con y sin el uso de la corrección LCC

3.2.2. Análisis de la variación del parámetro Clusters por identidad de secuencia

En la Figura 3.12 se muestran las curvas ROC obtenidas variando el cluster por identidad de secuencia utilizado. Se observa que las diferencias entre las mismas no resultan significativas, teniendo el cluster de 65 % una menor AUC y el de 90 % una mayor AUC.

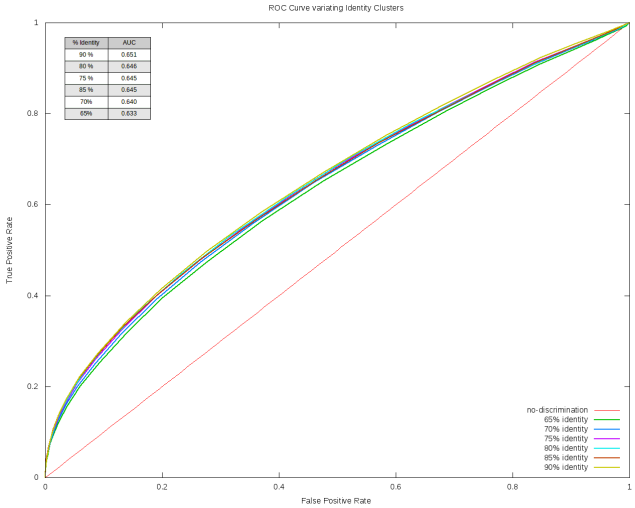

Fig. 3.12: Curvas ROC obtenidas con distintos clusters por identidad de secuencia.

De manera análoga tampoco se observan diferencias significativas en la cobertura, Figura 3.13, incrementándose la misma de manera directamente proporcional al porcentaje de identidad del cluster.

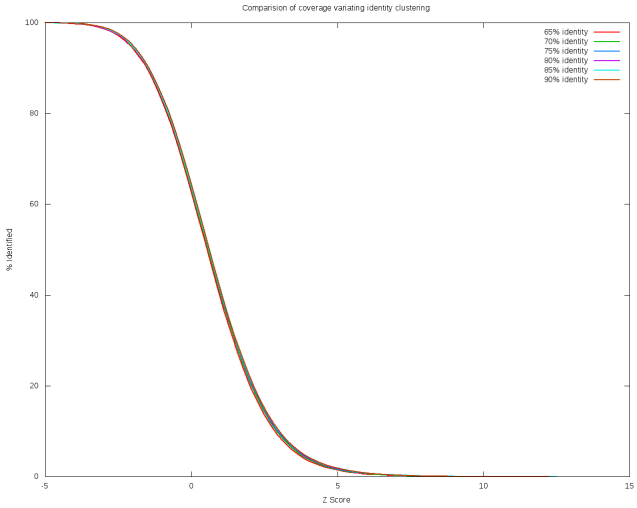

Fig. 3.13: Cobertura (TP Rate) de los distintos clusters por identidad de secuencia

Sin embargo, en la Figura 3.14 se observa que la precisión sí se ve afectada por el uso de los clusters por identidad. Es el cluster de 85 % de identidad el que se destaca por tener mayor precisión que el resto en el rango de Z score de interés (3 a 5).

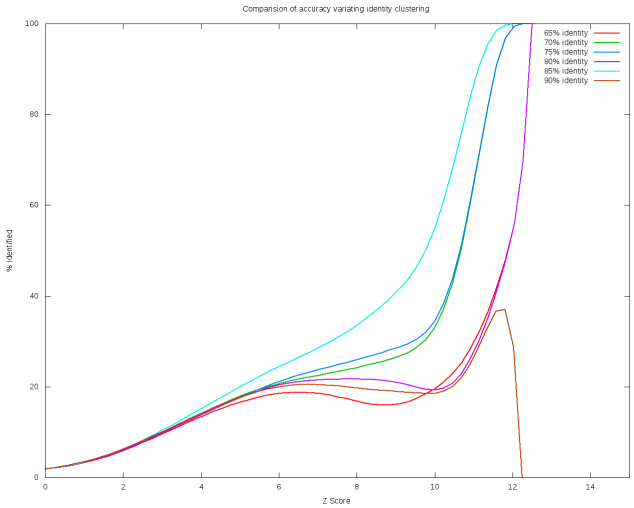

Fig. 3.14: Precisión (PPV) de los distintos clusters por identidad de secuencia

3.2.3. Análisis de la variación del parámetro Distancias de contacto

En la Figura 3.15 se muestran las curvas ROC obtenidas variando el método para calcular las distancias de contacto.
A diferencia de los demás parámetros, que impactan sobre el cálculo de coevolución alterando el valor de Z score, variar el método para calcular las distancias de contacto impacta sobre la forma de predecir la validez de los resultados, pues modifica la definición misma de predicción correcta o incorrecta de los resultados (es decir la definición de TP, FP, TN y FN). Por lo tanto, en esta figura lo que se observa es a qué modelo de contacto entre residuos se adapta mejor el método desarrollado.

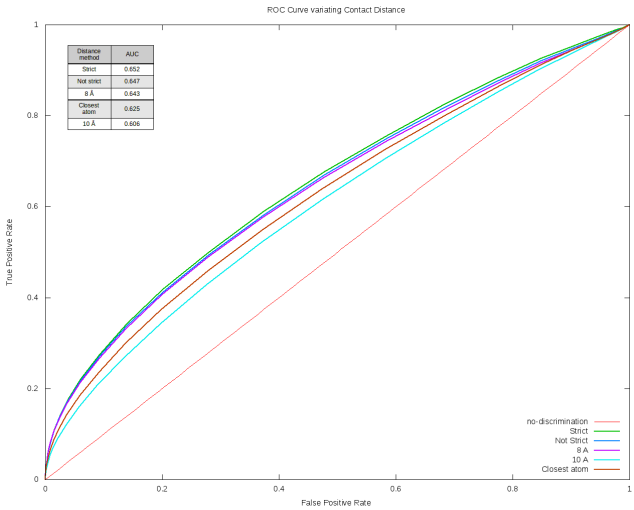

Fig. 3.15: Curvas ROC obtenidas con distintas formas de calcular distancia de contacto.

En particular, vemos que las dos medidas desarrolladas en este trabajo -*strict* y *not strict*-,

son las que mejores resultados retornan, seguidas muy de cerca por *8 Å*, que es la medida utilizada usualmente para definir contacto.
Consideramos que el hecho de que *8 Å* brinde resultados similares a las medidas desarrolladas (las cuales tienen en cuenta la relación entre diferentes tipos de aminoácidos), puede deberse a que esta medida está compensando el cambio favorable de FPs a TPs con un cambio desfavorable de TNs a FNs (recordar que la TP Rate depende de TP y FN, mientras que FP Rate depende de FP y TN). En otras palabras, pares que por los tipos de aminoácidos involucrados no deberían estar en contacto, pasan a ser considerados en contacto por estar en un radio menor a 8 Å. Esto deriva en que existen pares con un Z score mayor al threshold, que con las otras medidas no se consideraban en contacto y por ende eran FP, que en esta medida de distancia de contacto son tomados como TP por estar en contacto; mientras que, de manera análoga, pares que tienen Z score por debajo del umbral pasan de ser TN (en otras medidas) a FN por incrementar el radio de contacto a 8 Å.
De las otras dos medidas, *10 Å* da peores resultados que las demás, lo cual era esperable debido a que implica expandir el radio de contacto de manera excesiva incrementando así la cantidad de FN por demás. El argumento es similar al expresado para *8 Å* con la diferencia que en este caso, el radio de contacto es suficientemente grande como para que el incremento de FN no pueda ser contrarrestado proporcionalmente por el incremento de TP.

En la Figura 3.16 se muestra la cobertura de los distintos métodos para calcular distancia de contacto, y en la misma se ve reflejado lo expresado anteriormente respecto al TP Rate: *strict*, *not strict* y *8 Å* tienen tasas similares, mientras que *10 Å* y *closest atom* tienen tasas más bajas.

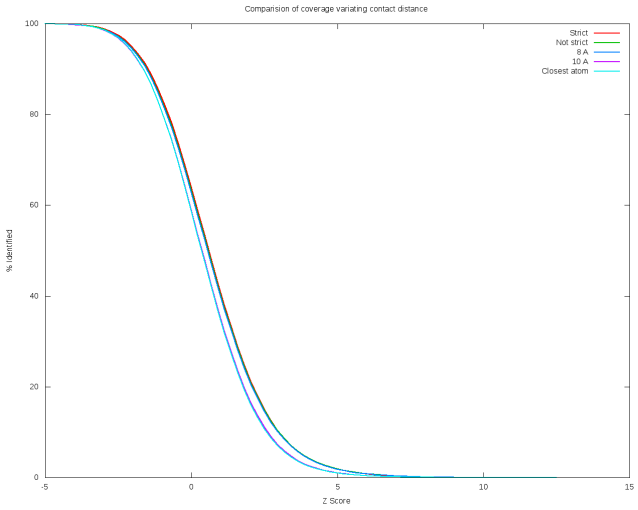

Fig. 3.16: Cobertura (TP Rate) de las distintas distancia de contacto

En resumen, si bien *8 Å* es un buen compromiso, un criterio que contempla el tamaño de los residuos, como *strict* y *not strict*, pareciera ser el más adecuado.

3.2.4. Análisis del uso de alfabetos reducidos

Respecto al uso de alfabetos reducidos en la Figura 3.17 se muestran las curvas obtenidas variando los mismos. Se observa que el alfabeto completo obtiene mejores resultados que las otras variantes analizadas, aunque cabe aclarar que este análisis se realizó contemplando un conjunto amplio de familias. En consecuencia, procedimos a probar de manera individual con algunas familias y encontramos algunas en que los alfabetos reducidos obtienen mejores resultados, lo que parece indicar que no conviene utilizar las reducciones de manera general sino en casos particulares en los cuales las características agrupadas estén manifestadas.

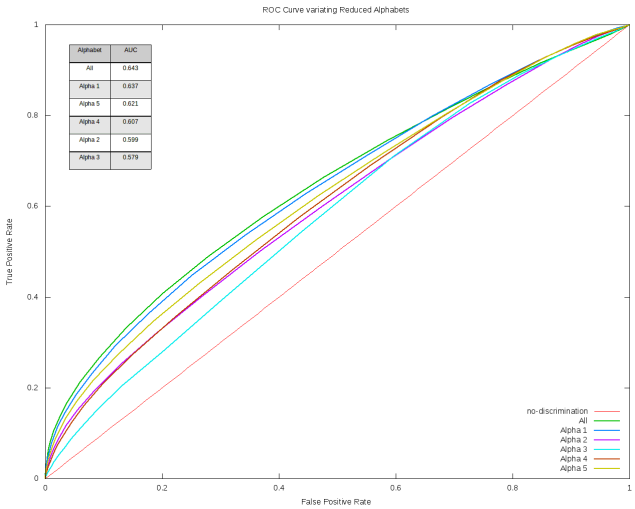

Fig. 3.17: Curvas ROC obtenidas con distintos alfabetos reducidos

De todas formas, observando la mencionada figura, vemos que el alfabeto alpha1 se comporta de forma similar al alfabeto completo, lo que indica que sus reemplazos reflejan la realidad correctamente y son generalmente aplicables. Con el resto de los alfabetos los resultados empeoran, creemos que debido al agregado de mas reducciones que no son viables de manera global. Cabe mencionar en particular al alfabeto alpha3, pues es el que peores resultados arroja, consideramos que se debe a que es el más arriesgado, en el que se realizan mayor cantidad y variedad de reducciones, tal vez en exceso dado que es un alfabeto de sólo 7 elementos lo cual no puede ser suficiente para representar las particularidades de todos los residuos en variados contextos.

La cobertura (TP Rate) de los alfabetos reducidos se muestra en la Figura 3.18. En la misma se observa que el alfabeto completo tiene mayor cobertura que el resto, y que estos se ubican de manera incremental.

Dado que, en general, utilizar el alfabeto completo obtiene mejores resultados que usar los reducidos, procedimos a analizar de qué manera pueden los alfabetos reducidos ser utilizados para mejorar los resultados. Para ello consideramos obtener la cobertura para todas las familias partiendo del alfabeto completo y adicionando alfabetos reducidos de a uno (ordenados de acuerdo a los que tienen mejor AUC).

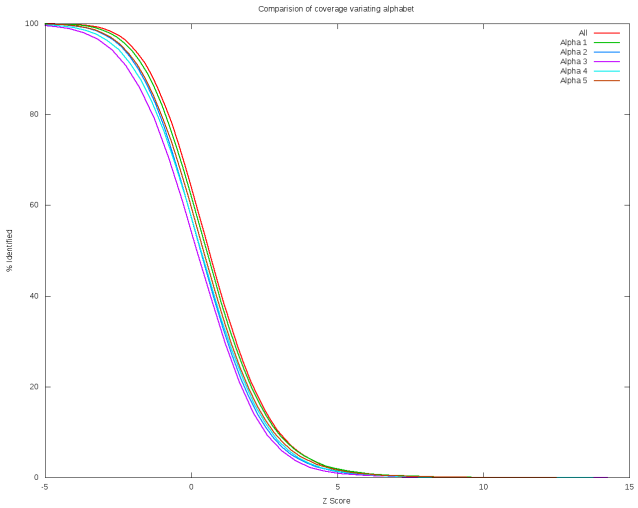

Fig. 3.18: Cobertura (TP Rate) del uso de Alfabetos reducidos

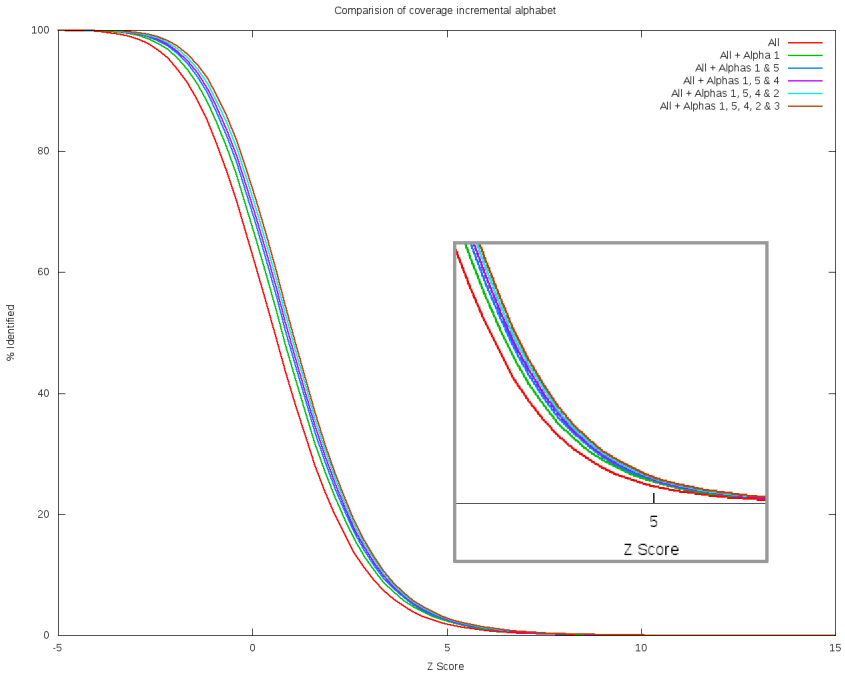

Fig. 3.19: Cobertura (TP Rate) para todas las familias, adicionando alfabetos de manera incremental (ordenados por mejor ROC). En el área enmarcada se muestra el zoom del área más significativa con el objeto de permitir una visualización del carácter aditivo de la incorporación de los alfabetos reducidos.

La metodología para adicionar alfabetos consiste en utilizar el mejor Z score de cada par de aminoácidos en los alfabetos que se están estudiando. De esta manera los alfabetos reducidos sólo pueden convertir pares, que con el alfabeto completo son FP, en pares que con el alfabeto incrementado son TP. Se muestra, entonces, en la Figura 3.19 la cobertura de estos nuevos alfabetos ampliados incrementalmente, y puede observarse que

la cobertura del alfabeto completo se ve mejorada con estas ampliaciones a medida que se van incorporando más alfabetos reducidos.

3.2.5. Análisis de la precisión de los parámetros

En la Tabla 3.2 se muestra la precisión de las distintas variantes de los mencionados parámetros, medidas a través de la PPV, la cantidad de pares TPs y la cantidad de pares FPs (estos últimos dos normalizados por la cantidad de pares posibles con todas las combinaciones posibles en 100 aminoácidos), utilizando como umbral de corte $Zscore > 4$.

Tabla 3.2: PPV, cantidad de TPs y cantidad de FPs para cada variación de cada parámetro. Threshold $Zscore > 4$.

Método	# TP x 100 aa	# FP x 100 aa	PPV
lcc	1.1312	6.5636	0.1470
no lcc	1.3890	12.8564	0.0975
sequence dist 5	1.1312	6.5636	0.1470
sequence dist 1	5.2172	5.5992	0.4823
cluster 65	1.7490	5.1123	0.2549
cluster 70	1.7959	5.0773	0.2613
cluster 75	1.9772	5.6673	0.2586
cluster 80	2.2478	6.7257	0.2505
cluster 85	2.1365	5.3701	0.2846
cluster 90	2.2052	6.2559	0.2549
all	1.1312	6.5636	0.1470
alpha1	1.9265	7.6304	0.2016
alpha2	1.4039	10.5061	0.1179
alpha3	0.8240	16.9623	0.0463
alpha4	1.2278	11.9625	0.0931
alpha5	1.7119	9.5241	0.1524
strict	1.1312	6.5636	0.1470
not strict	1.5109	6.1839	0.1964
d8	2.0185	5.6763	0.2623
d10	2.7904	4.9044	0.3626
closest atom	2.3815	5.3132	0.3095

Se observa que:

- No utilizar la corrección LCC incrementa levemente la cantidad de TPs pero duplica la cantidad de FPs.
- Considerar los pares muy cercanos en secuencia (menos de 5 aa de distancia) mejora muy significativamente la cantidad de TPs, abonando la hipótesis de que estos son pares triviales y conviene ignorarlos.
- Los clusters por identidad dan resultados similares, pero se observa que el cluster de 85 % de identidad obtiene el mejor compromiso entre las cantidades de TPs y FPs.

- En cuanto a los alfabetos reducidos, el alfabeto 3 tiene muy baja cantidad de TPs; mientras que el alfabeto 1, si bien tiene mayor cantidad de FPs que el alfabeto completo, encuentra TPs en mayor proporción logrando una mejor precisión.
- En el caso del parámetro referente a la distancia de contacto se observa que los resultados son inversos a los obtenidos en la cobertura, es decir las distancias que parecen reflejar la realidad de manera mas laxa son los que tienen mayor cantidad de TPs y menor cantidad de FPs. Esto se debe a que incrementar el radio de contacto incrementa los TPs en desmedro de los FPs de las distancias más cercanas. Cabe recordar que el problema de las distancias 10 Å y closest atom está en el exceso de FNs, lo cual no se está analizando en esta tabla.

4. CONCLUSIONES

Como conclusión de la presente tesis de Licenciatura puedo remarcar tres hitos importantes: por un lado el aprendizaje de diversos conceptos y herramientas, por otro el desarrollo de una aplicación que puede resultar de utilidad a otros profesionales del área y finalmente la realización de un análisis detallado de cómo diferentes parámetros afectan la performance de la predicción de coevolución.

En cuanto al aprendizaje, la presente tesis me permitió aprender y profundizar en conceptos biológicos y estadísticos; me permitió involucrarme en el campo de la bioinformática conociendo lenguajes de programación, herramientas y, más importante aún, el trabajo interdisciplinario con profesionales de las áreas de computación, biología y matemática; logré abstraer el problema biológico a uno matemático para luego resolverlo con herramientas computacionales, las cuales retornan resultados que debí corroborar que mantuvieran una coherencia dentro del marco biológico inicial.

Pude desarrollar una aplicación integral, que parte de las secuencias de una familia y la disposición espacial de los átomos de la proteína de referencia para la familia, y calcula la coevolución subyacente para todos los pares de aminoácidos posibles y diversas métricas asociadas. Asimismo implementé diversos métodos para enriquecer los resultados de coevolución, de manera de poder ofrecer no sólo el valor de Mutual Information y su Z score, sino también un conjunto de análisis que se muestran tanto gráfica como textualmente.

En base a los resultados, demostré que el método desarrollado detecta eficientemente coevolución entre residuos de proteínas. También observé que los parámetros brindan flexibilidad y permiten realizar la pesquisa desde distintos enfoques, y determiné cuales son las configuraciones óptimas de estos para obtener mejores predicciones de coevolución.

4.1. Trabajo futuro

Aún quedan áreas por explorar en cuanto a la coevolución de residuos en proteínas que escaparon el alcance de esta tesis de Licenciatura. Entre ellas podemos mencionar las siguientes:

- Crear una aplicación web con el objetivo de brindar el servicio abiertamente a través de una interfaz gráfica amigable. Los usuarios se verán beneficiados dado que podrán acceder a la aplicación independientemente del sistema operativo, podrán observar los resultados gráficamente y no deberán poseer conocimientos sobre el uso de línea de comandos.

- Evaluar coevolución entre residuos que no están en contacto. Si bien se ha demostrado que existe coevolución entre residuos alejados en la estructura terciaria [1][2], debido a la complejidad que conlleva determinarlo, en este trabajo decidimos utilizar un enfoque más directo (asumimos que contacto implica coevolución, como ya fue explicado), sabiendo que esto implica incrementar los FP y por ende actúa en desmedro de la precisión del método.
- Implementar una interfaz de programación de aplicaciones (API) para permitir la integración con otros sistemas del equipo de trabajo del Director y el Co-director de está tesis. El objetivo es brindar el cálculo de coevolución entre residuos como servicio para que otras aplicaciones enriquezcan sus resultados.

Apéndice

A. INSTRUCCIONES DE USO

A continuación se detallan los pasos requeridos para utilizar el sistema.

1. Obtener el MSA de la familia en formato FASTA.
2. Obtener el PDB correspondiente a la secuencia de referencia, la cual debe ser la primera en el MSA.
3. Generar el archivo de distancias en base al PDB, para ello se debe ejecutar en VMD el script *difference_matrix.tcl*.
   a) Modificar el nombre del archivo para que tenga la siguiente estructura: *family_subname*.dat.
   b) Alternativa: ejecutar el script *download_msa.pl*. Dada una lista de pdbs, genera un script para correr en vmd y obtener el archivo de distancia de todos los pdbs de la lista.
4. Asegurarse que la secuencia de referencia contenga el índice inicial y final de los residuos (por ejemplo, "/5-232").
5. Calcular el ajuste entre los índices del MSA y del PDB y crear una entrada en el archivo *structure_adjustments.txt* para la familia.
   a) Alternativa: ejecutar el script *run_adj_dir.pl*. Dadas las carpetas de distancia y los MSAs, calcula el ajuste entre el PDB y la secuencia de referencia de cada MSA. Devuelve los ajustes para revisar y un archivo con los valores de los ajustes.
   b) Asimismo, en modo debug la clase Coev_Statistics genera un log detallando la correlación entre posiciones del MSA y el PDB.
6. Calcular los clusters de 65 a 90 incrementando de a 5, usando CD-HIT. Los archivos deben tener la siguiente nomenclatura *family_subname_clusterID*.FASTA, donde *clusterID* es el porcentaje utilizado para el cluster (*subname* lo utilizamos para indicar si el MSA es full o seed). Para reducir el tiempo de ejecución se pueden eliminar los clusters iguales.
   a) Alternativa: ejecutar el script *run_cdhit_dir.pl*. Dada una carpeta con los MSAs, calcula los cluster para 65 a 90 % de identidad (de a 5 %).
7. Crear una carpeta dentro del directorio **data** y copiar los archivos resultantes (los FASTA y el archivo de distancias).
8. Modificar los parámetros que se requieran en el archivo *config.yml*.
9. Ejecutar por linea de comandos: *perl run_mi_dir.pl -id* nombre_carpeta. Donde nombre_carpeta es el nombre de la carpeta creada en el paso 7.

A.1. Parámetros disponibles

El archivo *config.yml* contiene diversos parámetros que permiten modificar el comportamiento del algoritmo. Los mismos se dividen en cuatro grupos:

- *general*: parámetros generales del algoritmo (particularmente la ruta de las carpetas necesarias).
- *method*: parámetros que se utilizan para configurar el comportamiento de los métodos.
- *values*: parámetros que indican valores utilizados en los métodos.
- *alphabets*: expresiones regulares que definen los alfabetos posibles.

Asimismo, la matriz de distancias de contacto se obtiene del archivo *distances.yml*. Por default, este archivo contiene la matriz estricta (strict), mientras que la matriz no estricta (not strict) se encuentra en el archivo *distances_no_strict.yml* (para utilizarla se debe renombrar el archivo a *distances.yml*).

A.1.1. Parámetros del grupo *method*

Los parámetros de los métodos existentes son los siguientes:

- *alphabet_type*: indica el alfabeto utilizado, los valores posibles son *all* y los nombres de las entradas del grupo *alphabet*.
- *debug*: valor binario que indica si se ejecuta en modo debug.
- *APC_or_ASC*: indica si se utiliza la corrección APC, ASC, APC_gap o ASC_gap.
- *use_gap*: valor binario que indica si se usa el gap como un caracter del alfabeto.
- *use_lcc*: valor binario que indica si se utiliza la corrección LCC.
- *force_contact_distance*: valor binario que indica si se ignora el archivo de distancias (*distances.yml*) y se toma un valor fijo de distancia especificado en *values*.
- *analysis_plots*: valor binario que indica si se desea obtener gráficos adicionales de análisis del algoritmo.
- *value_used_in_stats*: indica si se utiliza Z o MI para los gráficos estadísticos.
- *extra_stats*: valor binario que indica si se desea obtener gráficos estadísticos adicionales.

A.1.2. Parámetros del grupo *values*

Los parámetros que indican valores utilizados en los métodos son los siguientes:

- *contact_distance*: valor fijo de distancia de contacto que se utiliza si *force_contact_distance* está en 1.
- *ignore_sequence_distance*: mínima distancia en secuencia aceptable.
- *lcc_lambda*: λ utilizado en LCC.
- *max_Z*, *mid_Z* y *min_Z*: rango de Z utilizado para graficar las redes de conectividad.
- *min_H*: filtro de entropía para graficar las redes de conectividad.
- min_entropy: filtro de entropía para calcular MI.
- *mi_score_step*: paso de MI utilizado para graficar estadísticas.
- *z_score_step*: paso de Z utilizado para graficar estadísticas.
- *acceptable_gap_percent*: porcentaje aceptable de gaps en una columna para que esta sea tenida en cuenta.

A.1.3. Alfabetos reducidos

Para definir los alfabetos reducidos se utiliza una expresión regular que permite indicar que residuos se reemplazan y por cuales. Para ello se debe indicar en el parámetro *remove_chars* aquellos residuos que se eliminan, y en el parametro *replace_with* los residuos que los reemplazan en orden (el primer residuo de *remove_chars* es reemplazado por el primero de *replace_with*, el segundo por el segundo y así hasta el final). Si *remove_chars* contiene mas residuos que *replace_with*, los restantes son reemplazados por el último residuo de la lista de reemplazo.

Bibliografía

[1] L. Burger and E. van Nimwegen, "Disentangling direct from indirect co-evolution of residues in protein alignments," *PLoS Comput Biol*, vol. 1, January 2010.

[2] M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, and T. Hwa, "Identification of direct residue contacts in protein–protein interaction by message passing," *PNAS*, vol. 1, no. 106, pp. 67–72, 2009.

[3] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, p. 379–423, 1948.

[4] S. D. Dunn, L. M. Wahl, and G. B. Gloor, "Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction," *Bioinformatics*, vol. 24, no. 3, pp. 333–340, 2008.

[5] L. C. Martin, G. B. Gloor, S. D. Dunn, and L. M. Wahl, "Using information theory to search for co-evolving residues in proteins," *Bioinformatics*, vol. 21, no. 22, pp. 4116–4124, 2005.

[6] C. Marino Buslje, J. Santos, J. M. Delfino, and M. Nielsen, "Correction for phylogeny, small number of observations and data redundancy improves the identification of coevolving amino acid pairs using mutual information," *Bioinformatics*, vol. 25, pp. 1125–1131, 2009.

[7] K. R. Wollenburg and W. R. Atchley, "Separation of phylogenetic and functional associations in biological sequences by using the parametric bootstrap," *Proceeding of the National Academy of Sciences USA*, vol. 97, no. 7, pp. 3288–3291, 2000.

[8] M. Punta, P. Coggill, R. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E. Sonnhammer, S. Eddy, A. Bateman, and R. Finn, "The pfam protein families database," *Nucleic Acids Research*, vol. Database Issue 40, pp. D290–D301, 2012.

[9] E. Lanzarotti, R. Biekofsky, D. Estrin, M. Marti, and A. Turjanski, "Aromatic-aromatic interactions in proteins: beyond the dimer," *J Chem Inf Model*, vol. 51, pp. 1623–1633, Jul 2011.

[10] R. Gouveia-Oliveira and A. G. Pedersen, "Finding coevolving amino acid residues using row and column weighting of mutual information and multi-dimensional amino acid representation," *Algorithms for Molecular Biology*, vol. 2, no. 12, 2007.

[11] L. Weizhong and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[12] G. Crooks, G. Hon, J. Chandonia, and S. Brenner, "Weblogo: A sequence logo generator," *Genome Research*, vol. 14, pp. 1188–1190, 2004.

[13] M. Smoot, K. Ono, J. Ruscheinski, P. Wang, and T. Ideker, "Cytoscape 2.8: new features for data integration and network visualization," *Bioinformatics*, vol. 27, p. 431–432, February 2011.