Este capítulo pertenece al libro

Bioinformática con Ñ

Edición y coordinación:
Álvaro Sebastián y Alberto Pascual-García

Autores:
F. Abascal, J. Aguirre, E. Andrés-León, D. Bajic, D. Baú, J. A. Bueren-Calabuig,
Á. Cortés-Cabrera, I. Dotu, J. M. Fernández, H. G. D. Santos, B. García-Jiménez,
R. Guantes, I. Irisarri, N. Jiménez-Lozano, J. Klett, R. Méndez, A. Morreale,
A. Pascual-García, A. Perona, A. Sebastian, M. Stich, S. Tarazona, I. Yruela y R. Zardoya

Portada:
Enrique Sahagún (`http://www.scixel.es`)

Maquetación:
Álvaro Sebastián
Usando LaTeX(`http://www.latex-project.org`) y Texmaker (`http://www.xm1math.net/texmaker`)

Editorial:
Libro autoeditado e impreso por CreateSpace (`http://www.createspace.com`)

Depósito legal: SE-NNNNNNN

ISBN: NNNNNNN

Índice general

Capítulo 1

Alineamiento de estructura de proteínas

Alberto Pascual-García

1.1. Introducción

En el presente capítulo vamos a presentar una introducción a algunos métodos computacionales orientados a *comparar de manera objetiva estructuras de proteínas*. La motivación reside en que sabemos que la estructura de la proteína nos puede ayudar a inferir la función de la misma. La secuencia de aminoácidos determina el plegamiento nativo, y a su vez el plegamiento determinará los movimientos posibles de la proteína con la consecuente exposición de los distintos tipos de aminoácidos, y por extensión de sus posibilidades funcionales. Esta cuestión es tratada en mayor detalle en el **??**.

La observación de que *la divergencia estructural entre proteínas que comparten un ancestro común es proporcional al tiempo que hace que se separaron de dicho ancestro* [2], sugiere que la comparación estructural entre homólogos podría permitir inferir funciones desconocidas (ver **??**). Pero en realidad el escenario es bastante más complejo, pues podemos contemplar prácticamente todas las posibilidades en la terna secuencia, estructura y función. Es posible por ejemplo encontrar proteínas con la misma función en estructuras claramente distintas (y sin homología reconocible) pero que comparten una región funcional muy local de elevada similitud estructural. También encontramos conjuntos de proteínas homólogas, con elevada similitud estructural a nivel global y funciones distintas [1, 5, 6, 11]. En la Figura 1.1A se muestra el alineamiento entre dos proteínas sin homología reconocible, pero con una similitud global clara si bien también con diferencias en los motivos periféricos. Del mismo modo mostramos dos proteínas con aparente similitud global pero cuyos plegamientos son considerados distintos en las clasificaciones estructurales, e incluso con contradicciones entre ellas en la Figura 1.1B (ver **??**).

Pero esta introducción a la relación secuencia-estructura-función a la luz de la evolución, nos da una idea del escenario en el que tiene que moverse una persona interesada en desarrollar métodos computacionales para comparar estructuras. Encontrar similitudes estructurales entre dos estructuras cualesquiera implica que somos capaces de *identificar similitudes tanto locales como globales*, incluyendo posibles reordenamientos de sus elementos de estructura secundaria. Por ejemplo, si tuviéramos los motivos estructurales $a - b - c$ a lo largo de la secuencia de la proteína A y los motivos $b - a - d$ a lo largo de la proteína B, pero estuvieran espacialmente colocados de manera equivalente, deberíamos de ser capaces de llevar a cabo el alineamiento (ver Figura 1.2).

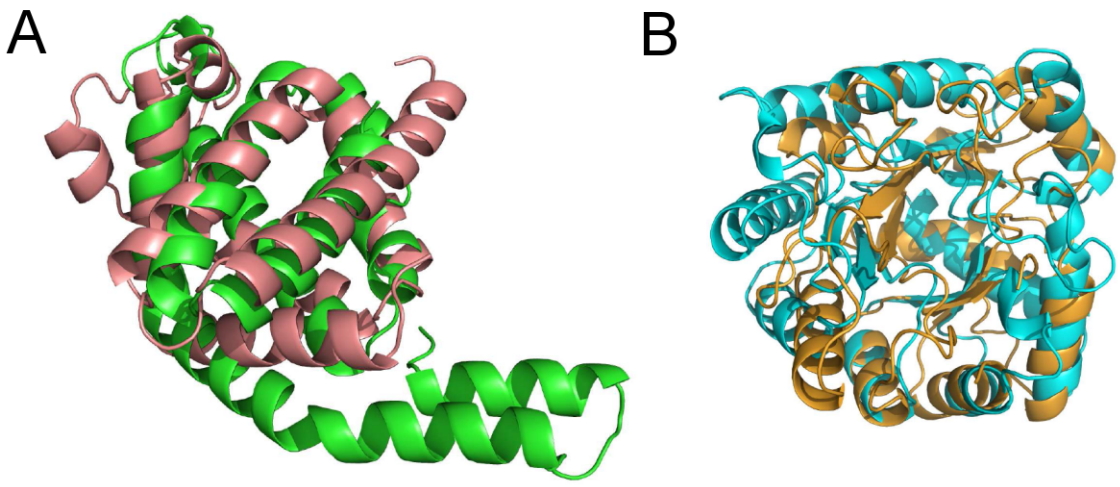

Figura 1.1: A. Alineamiento entre la subunidad alpha de la aloficocianina (en verde, con PDB 1all y clasificada como "Globin-like" en SCOP) y la oxi-mioglobina (en marrón, con PDB 1a6m y clasificada como "Globin"). A pesar de no ser proteínas con homología reconocible sus plegamientos son globalmente similares. B. Lo mismo ocurre con la hidrolasa dependiente de metal (en azul, con PDB 1j6o y clasificada como "TIM barrel") y la hipotética proteína YcdX de Escherichia Coli (con PDB 1m65 y clasificada como "7-stranded beta/alpha barrel") [22].

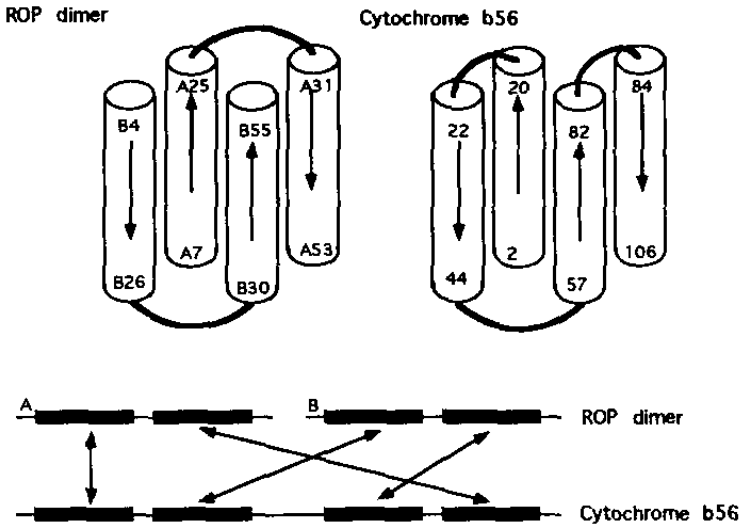

Figura 1.2: Representación esquemática del dímero ROP y el citocromo b56 [8]. Ambas proteínas comparten cuatro alpha hélices con conectividades distintas. En el caso del dímero, está formado por dos hebras idénticas (cadenas A y B) mientras que en el citocromo tenemos una única cadena. Necesitaremos por tanto una búsqueda local no secuencial para poder reconstruir el alineamiento. Figura reproducida con permiso del editor.

Cabría preguntarse además hasta qué punto los algoritmos deberían desarrollarse siguiendo *criterios puramente geométricos* o deberían de considerar *información evolutiva*, como en el caso de los algoritmos de alineamiento de secuencias [25]. La mayoría de los algoritmos de comparación estructural consideran medidas geométricas '*ad-hoc*'. Considerar medidas estrictamente geométricas permiten generar medidas objetivas, pero exige una interpretación evolutiva y funcional posterior. Este hecho explica las *discrepancias entre las denominadas clasificaciones estructurales*, que en algunos casos utilizan información evolutiva y funcional, con respecto a los intentos de clasificación puramente geométricos. Esta cuestión es detallada también en la sección **??**.

El número y variedad de *algoritmos de alineamiento estructural* se ha duplicado cada cinco años durante los últimos treinta años [4], y sin embargo no existe consenso sobre qué método y medidas es el mejor ya que no existe un '*golden standard*' que sirva de referencia para mejorar dichos métodos, y se llegan a menudo a conclusiones contradictorias (ver por ejemplo las comparaciones que se encuentran en [20] y [25]). Un buen lugar para encontrar métodos disponibles *online* con una descripción sencilla de los mismos es la *Wikipedia*[1], así como *software*[2]. Aquí hemos seleccionado algunos algoritmos que creemos son representativos de las aproximaciones generales más importantes a tener en cuenta, para hacernos una idea aproximada de toda la variedad que se esconde tras el problema de alineamiento estructural.

1.2. Descripción general del método

La primera división a tener en cuenta es entre métodos que consideran o no información evolutiva. Aquí vamos a tratar de métodos que consideran criterios puramente geométricos por ser claramente mayoritarios, si bien al final expondremos un ejemplo de medida que pretende incorporar información evolutiva, y remitimos al lector a algunos de los intentos de incorporar información evolutiva en las referencias [3, 13]. La mayoría de los *métodos geométricos* no hacen uso de la secuencia de las proteínas, sino que trabajan directamente con las *coordenadas de los átomos en el espacio* y en la mayoría de los casos únicamente con las coordenadas de los carbonos *alpha* (C_α). Por tanto definiremos una proteína genérica A mediante el conjunto de coordenadas x, y, z de cada uno de los carbonos *alpha* de sus residuos a_i, es decir: $A = \{a_i\}_{i=1}^n = \{(a_{x_i}, a_{y_i}, a_{z_i})\}_{i=1}^n$, donde n es el número de residuos. Esta descripción reducida de la proteína es lo que se conoce como *esqueleto* (del inglés *backbone*) de las proteínas que, aun siendo una simplificación importante es necesaria, ya que veremos que el problema sigue siendo de una elevada complejidad. Queremos entonces comparar el esqueleto de la proteína A con el de otra proteína $B = \{b_i\}_{i=1}^m$ donde m es su número de residuos y, en el caso más general, es distinto de n. Nos gustaría obtener como salida del algoritmo un valor numérico indicando la similitud (o disimilitud) entre ambas estructuras.

Hay dos obstáculos que rápidamente salen a la luz en cuanto planteamos el problema. El primero es que las coordenadas de cada proteína están en *sistemas de referencia* distintos, por lo que se ha de encontrar una transformación entre los sistemas. El segundo es que las *longitudes de las proteínas* son distintas. Por tanto, nuestro problema consiste en encontrar una operación $f : A \to B$ tal que para cada residuo $a_i \in A$ encontramos una *correspondencia inyectiva* con un residuo $b_k \in B$. Además, el conjunto de todas las correspondencias encontradas debería ser tal que dicha correspondencia maximiza una función que consideremos que mide la información estructural que ambas proteínas comparten. El problema consistente en decidir si existe una cierta correspondencia cuya mejor *transformación rígida* aproxime las dos estructuras a una distancia prefijada entre los residuos en correspondencia. Su solución requiere la búsqueda y evaluación de todas las combinaciones posibles entre residuos, lo cuál

[1] Métodos de alineamiento estructural. `http://en.wikipedia.org/wiki/Structural_alignment`
[2] *Software* de alineamiento estructural. `http://en.wikipedia.org/wiki/Structural_alignment_software`

es un problema no resoluble en tiempo polinomial, por lo que *podríamos diferenciar a los distintos algoritmos esencialmente por*: 1) en el modo en que reducen la búsqueda de combinaciones con una *aproximación heurística inicial* y 2) en el *tipo de función que evalúa la información estructural* que las proteínas comparten. Y, como es de esperar, la heurística utilizada va de la mano de la función que se pretende evaluar y viceversa, por lo que desde el punto de vista didáctico no es sencillo decidir qué exponer en primer lugar. Nosotros vamos a mostrar en primer lugar cómo reducir la combinatoria, que en la mayoría de los casos se resuelve mediante comparaciones locales entre ambas proteínas, y después veremos el modo en que la correspondencia se reconstruye optimizando una determinada función global.

Vamos a precisar de manera más formal estos pasos y a tratar algunos ejemplos. En los ejemplos vamos a presentar principalmente las aproximaciones seguidas por los *algoritmos de alineamiento Dali* [8], y *MAMMOTH* [20], si bien examinaremos otros algoritmos para explicar otras metodologías por su relevancia u originalidad. Elegimos estos algoritmos no solamente por ser ampliamente utilizados sino porque el primero se considera uno de los mejores algoritmos de alineamiento (es uno de los más antiguos y ha sido mejorado en sucesivas versiones) y el segundo es uno de los más rápidos, lo que lo hace apropiado para análisis masivos como el estudio del universo de estructuras de proteínas. Este *balance entre calidad del alineamiento y velocidad* queda patente en los distintos pasos que presentaremos, lo que nos resulta bastante ilustrativo.

1.3. Comparaciones locales

El primer paso para realizar un alineamiento de proteínas es la *descomposición de cada una de las proteínas problema en fragmentos* seguido de la comparación de todos los fragmentos contra todos de ambas proteínas. Los fragmentos son típicamente hexámeros o heptámeros, que es el tamaño mínimo para tener sensibilidad suficiente para *detectar similitudes locales*. La información que se utiliza típicamente es estrictamente estructural, si bien proliferan los métodos en los que alguna información adicional reduce el número de comparaciones a tener en cuenta. Por ejemplo, en Matras [9] se utiliza también información sobre la exposición al solvente del residuo.

Comencemos analizando cómo trabaja MAMMOTH. Este algoritmo considera todos los heptámeros en la proteína A $\{(a_i, ..., a_{i+6})\}_{i=1}^{n-6}$ y en la proteína B $\{(b_k, ..., b_{k+6})\}_{k=1}^{m-6}$ realizando una comparación de todos los heptámeros contra todos. Para realizar esta comparaciones considera una esfera de radio unidad y, para cada par de heptámeros a comparar, crea vectores unitarios centrados en el origen de dicha esfera en la dirección de $C_\alpha \rightarrow C_{\alpha+1}$. Como la separación entre dos carbonos α consecutivos es aproximadamente fija (3,84Å) esta representación contiene la información que necesitamos sobre el esqueleto de la proteína. Tendremos por tanto 6 vectores unitarios $u_i = \{\bar{u}_{j,j+1}\}_{j=i}^{j=i+6}$ para cada heptámero u_i de la proteína A y otros 6 vectores unitarios, $v_k = \{\bar{v}_{l,l+1}\}_{k=l}^{k=l+6}$, para cada heptámero de la proteína B, y podemos imaginar que introducimos los 12 vectores en la esfera de radio unidad centrados en el origen. A continuación, manteniendo fijos los vectores asociados a una de las proteínas, buscamos una única rotación solidaria de los vectores de la segunda (es decir, mantenemos fijos sus ángulos relativos), de modo que minimicemos la distancia cuadrática con respecto a los vectores de la primera proteína, que no es más que *minimizar la distancia entre todos los pares de vectores correspondientes para cada proteína*, donde los superíndices denotan las respectivas componentes de los vectores unitarios en las direcciones de x, y, z. Esta medida se denomina $URMS$ del inglés *Unit-vector Root Mean Square* y tiene aplicaciones también como coordenada de reacción en el plegamiento de proteínas o para

monitorizar trayectorias en dinámica molecular [10].

$$URMS_{uv} = \sqrt{\sum_{i,k=1}^{6} \left(u^x_{i,i+1} - v^x_{k,k+1}\right)^2 + \left(u^y_{i,i+1} - v^y_{k,k+1}\right)^2 + \left(u^z_{i,i+1} - v^z_{k,k+1}\right)^2} \tag{1.1}$$

Es interesante destacar de esta medida que se puede dar una estimación del valor esperado para dos polímeros construidos con direcciones al azar: $URMS_R = \sqrt{2{,}0 - \frac{2{,}84}{\sqrt{L}}}$, donde L es el número de vectores unitarios. Esto nos permite derivar una medida de similitud entre cada par de heptámeros u y v como:

$$S_{uv} = \frac{(URMS_R - URMS_{uv})}{URMS_R} \triangle (URMS_{uv}, URMS_R) \tag{1.2}$$

donde $\Delta\left(URMS_{uv}, URMS_R\right) = 0$ si $URMS_R < URMS_{uv}$ y tendrá un valor cualquiera, que determinará la escala de la medida de similitud, por ejemplo 10 en el caso de MAMMOTH, en caso contrario. De modo que si nuestra proteína tiene longitud n, este procedimiento nos permitirá determinar una matriz S_{uv} de tamaño $n-6 \times n-6$ que será nuestra entrada para reconstruir el alineamiento global de la proteína, como veremos en el siguiente apartado.

En el caso del *programa de alineamiento Dali*, la aproximación que se realiza consiste en considerar también fragmentos de proteínas, pero primero calculando una *representación en dos dimensiones de cada proteína denominada matriz de distancias*. La matriz de distancias de una proteína genérica A que denotamos por d^A_{ij} es simplemente una matriz en la que, en cada celda, tenemos la distancia euclídea entre los dos residuos i y j. Esta representación "astuta" de la proteína, contiene toda la información estructural (salvando la quiralidad) y nos permite comparar fácilmente dos estructuras. Dali subdivide la matriz de distancias de la primera proteína d^A_{ij} en submatrices de distancias de tamaño 6×6: $\{(a_i, ..., a_i + 5;\ a_j, ..., a_j + 5)\}$ (con $i = 1, ..., n-13$ y $j > i+6, ..., n$) y realiza la misma descomposición sobre la matriz de distancias d^B_{kl}: $\{(b_k, ..., b_k + 5;\ b_l, ..., b_l + 5)\}$ en la proteína B. A continuación, *binariza* la matriz de distancias convirtiéndola en una *matriz de contactos* C_{ij}(ver Figura 1.3), es decir, la convierte en una matriz de unos y ceros en donde:

$$C_{ij} = \begin{cases} 1\ si & d_{ij} \leq d_0 \\ 0\ si & d_{ij} > d_0 \end{cases}$$

donde $d_0 = 4$Å es la distancia mínima para considerar que dos residuos están en contacto. La ventaja de considerar matrices de contactos es que podemos rápidamente capturar la información estructural relevante ya que, como se puede ver en la sección de plegamiento de proteínas, los contactos que tienen lugar entre residuos alejados en la secuencia de la proteína son determinantes en el plegamiento de la misma.

La similitud entre dos submatrices de contactos se calculará simplemente como el número de contactos compartidos, que llamamos *solapamiento de contactos*. Como contrapartida, y dado que el tamaño de la matriz de distancias crece como el cuadrado de la longitud de la proteína (n^2), el número de comparaciones a realizar crecerá como ($n^2 m^2$), lo que hace esta aproximación computacionalmente mucho más intensa. Por ello Dali realiza una serie de filtros para proporcionar una lista de comparaciones reducida y no redundante (que no vamos a tratar) que será la entrada para el apartado que discutiremos a continuación en el que optimizamos una función global. Simplemente mencionar que esta reducción es tal que el orden de magnitud de la matriz es aproximadamente el mismo (para comparaciones de proteínas de cualquier longitud) que el que gestionaríamos con MAMMOTH para una comparación de dos proteínas de 200 residuos.

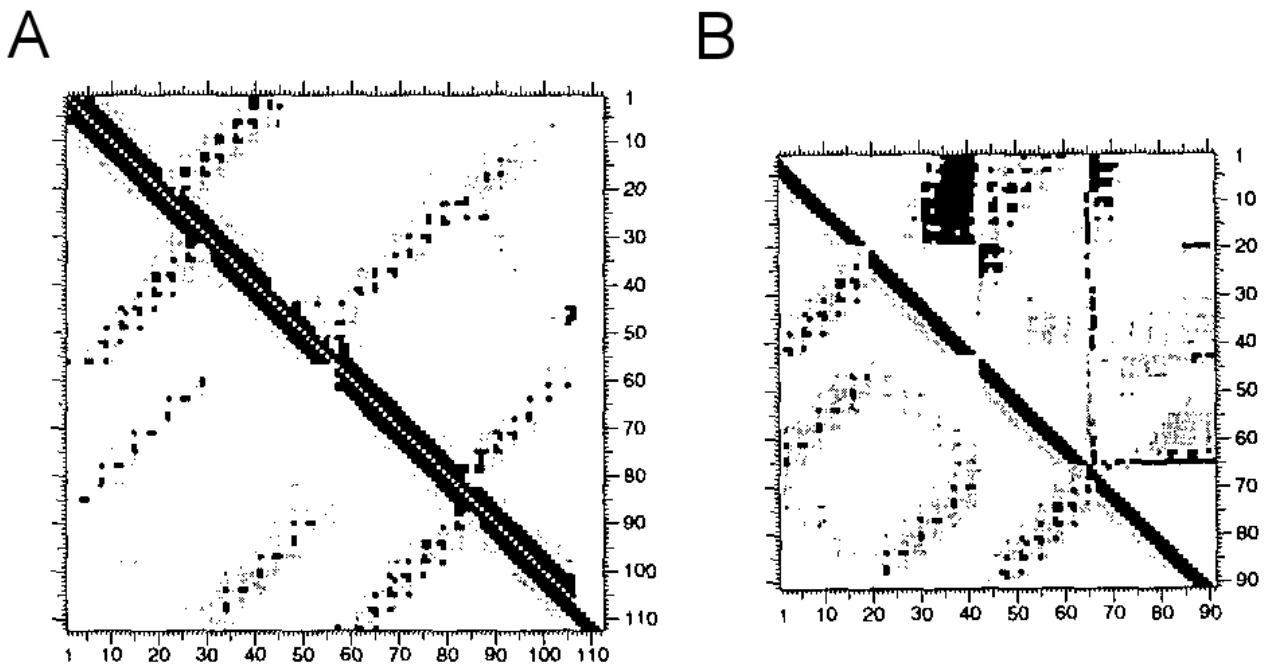

Figura 1.3: A. En la figura se representan las matrices de distancias de las proteínas del ejemplo mostrado en la Figura 1.2 en una única matriz (pues son simétricas y de longitud similar). Las distancias están discretizada en contactos a menos de 8Å (negro), 12Å (gris oscuro) y 16Å (gris claro). El dímero ROP se representa en la parte triangular inferior y el citocromo b56 en la superior. B. Se representa en la triangular superior la matriz de diferencias de distancias entre ambas proteínas, manteniendo como referencia a ROP en la inferior tras elinimar la región no alineada. Para la matriz de diferencia de distancias el código de colores va del blanco (diferencia de menos de 1Å), al negro (más de 4Å). Figura reproducida con permiso del editor.

1.4. Construcción del alineamiento: superposición rígida, flexible y una medida elástica.

Nuestro siguiente objetivo consiste en, a partir de la información que hemos obtenido de las comparaciones locales, obtener un alineamiento global óptimo desde el punto de vista estructural. En general la mayoría de los algoritmos construyen una primera matriz de similitudes locales como hemos mostrado en el caso de MAMMOTH, que permite realizar con Programación Dinámica un primer alineamiento global usando un método como el Needlman-Wunsch [18]. Discutiremos después el caso de Dali, que se diferencia en lo que vamos a exponer a continuación del resto.

En general, la actualización del alineamiento inicial utilizando información estructural global pasa por un proceso de superposición óptima. Para entender este proceso, consideramos el caso más simple de superposición, que es aquél que tiene lugar entre dos proteínas de la misma longitud con un elevado parecido entre ellas. En este caso no es necesario realizar alineamiento alguno pues existe una relación uno a uno entre los residuos a_i de la primera proteína A, y los residuos b_i de la segunda proteína que llamamos B. En este caso, llamamos *superposición óptima* a la posición relativa entre dos proteínas que maximiza (o minimiza) una medida global de similitud entre ambas cuando movemos una respecto a la otra únicamente mediante operaciones de cuerpo rígido, a saber, traslaciones y rotaciones. Si llamamos

T a esta transformación de cuerpo rígido, una posible medida a optimizar sería la raíz cuadrada de la *desviación cuadrática media* ($RMSD$ del inglés *root mean square deviation*) *entre los residuos de ambas proteínas*. Si llamamos D a su desviación cuadrática:

$$D = \sum_{i=1}^{n} \|a_i - T(b_i)\|^2$$

Definimos el valor del $RMSD$ como:

$$RMSD = \sqrt{\frac{D}{n}} \tag{1.3}$$

La idea de la optimización del alineamiento inicial, consiste en aplicar este tipo de transformaciones de cuerpo rígido (superposiciones por tanto) a un primer conjunto de residuos que consideramos bien alineados, para progresivamente ir incorporando nuevos residuos a la vez que evaluamos la bondad de la superposición con una medida como el $RMSD$. Expongamos el problema más en detalle. Supongamos que nuestro primer alineamiento $f : A \to B$ nos ha mapeado N_c residuos $a_i \in A$ con el mismo número de residuos $b_k \in B$. Es conveniente para discutir las siguientes medidas y dado que tenemos identificados los pares y podemos ordenarlos, el unificar entonces la notación de nuestros subíndices y considerar el mismo subíndice para cada par alineado, es decir el par (a_i, b_k) pasaremos a llamarlo (a_i, b_i). Pues bien, a continuación aplicaremos una transformación T de cuerpo rígido que maximice una función $F(D_c)$ que contiene típicamente también un término de desviación cuadrático D_c:

$$D_c = \sum_{i \in N_c} \|a_i - T(b_i)\|^2$$

que ahora vemos está particularizado únicamente a los residuos alineados en común. Se procederá a modificar iterativamente el alineamiento inicial buscando que la función $F(D_c)$ se minimice. Una forma sencilla de $F(D_c)$ sería por ejemplo el denominado $RMS(N_c)$:

$$RMS(N_c) = \sqrt{\frac{D_c}{N_c}} \tag{1.4}$$

Hay que observar aquí que, si quisiéramos optimizar esta medida, tendríamos que elegir entre, o bien restringir su valor a un umbral máximo e intentar maximizar el número de residuos alineados (N_c), o bien intentar minimizar su valor restringiendo el valor de N_c. La primera opción es la que siguen algoritmos como MAMMOTH utilizando programación dinámica con la heurística de MaxSub [27], o los conocidos algoritmos CE [26] y ProSup [12]. La segunda opción es la elegida por ejemplo en el algoritmo LOVOalign [15]. La Figura 1.4 ejemplifica los distintos casos posibles.

Tenemos además que hacer notar en este punto que estas medidas tendrán una dependencia con la longitud de la proteína. Esto es debido a que la probabilidad de encontrar un mapeo entre dos pares de residuos al azar aumenta con la diferencia de la longitud de las proteínas que se está comparando, lo cual estará presente en el alineamiento que realizamos inicialmente y como también sucede en los alineamientos de secuencia. Y este hecho está aún presente en la superposición estructural óptima que estamos tratando, por lo que habrá que eliminar esta dependencia en la medida de salida que obtengamos como veremos en la sección siguiente. Sin embargo, se puede realizar una *normalización* en este paso *para evitar la dependencia con la longitud*, como se propone en el algoritmo TM-align [29], donde se incluye una normalización en la medida que proponen:

$$TMscore = \frac{1}{n_{min}} \sum_{i=1}^{N_c} \frac{1}{1 + \left(\|a_i - T(b_i)\| / g(n_{min})\right)^2}$$

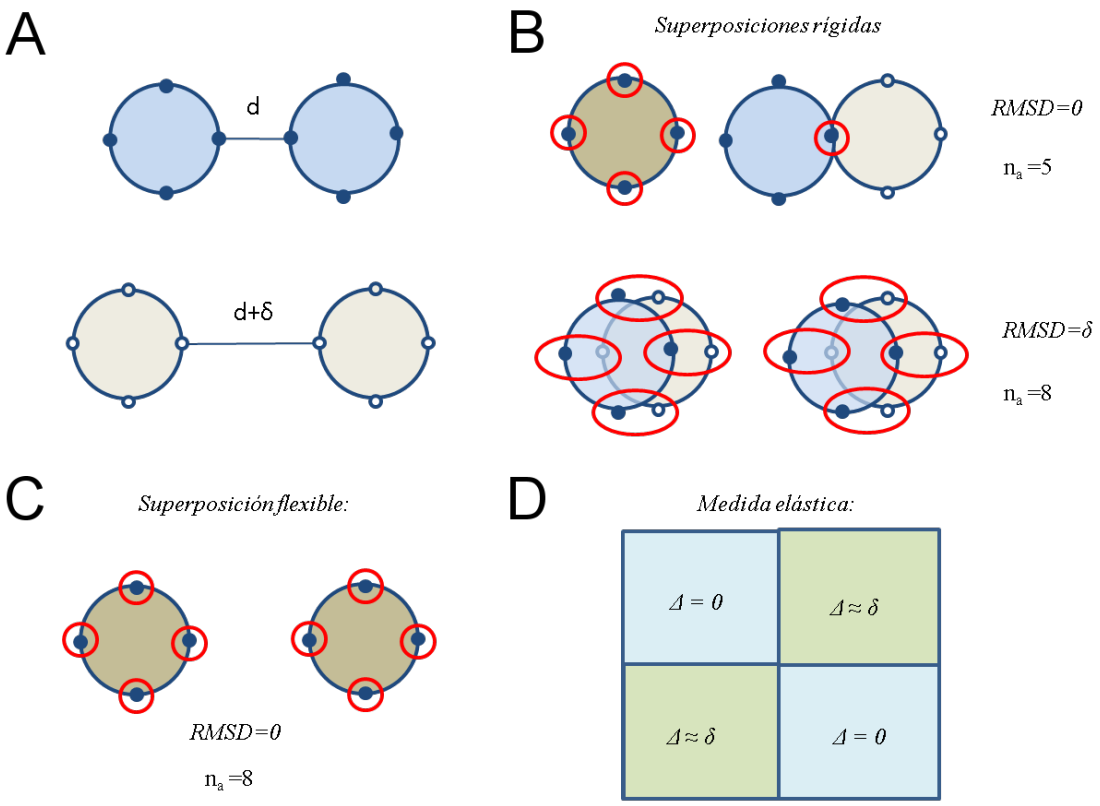

Figura 1.4: Superposición de dos estructuras ideales (A). En la superposición rígida (B) se maximiza o bien N_c fijando un $RMSD$ mínimo o viceversa. Las elipses rojas muestran residuos superpuestos y vemos que, o bien no superponemos todos o lo conseguimos a costa de un $RMSD$ poco satisfactorio. En la superposición flexible (C) la proteína se puede fragmentar (hecho que hay que controlar) y después superponer cada fragmento de manera independiente, lo que en este caso permite una superposición óptima. Por último, la base de las medidas elásticas residen en la matriz de diferencia de distancias, que es independiente del sistema de referencia, y permite también encontrar la superposición óptima (D). Figura adaptada de [4] con permiso del editor.

donde $g(n_{min}) = 1{,}24\sqrt[3]{n_{min} - 15} - 1{,}8$ siendo n_{min} la longitud mínima de las proteínas comparadas y los distintos valores son parámetros optimizados experimentalmente. De este modo, se elimina la dependencia con la longitud mínima de las dos proteínas ya que la normalización está implícita en el proceso de superposición óptima. Existiría aún cierta dependencia con respecto a la longitud de cada una de ellas por separado, pero es importante únicamente para rangos de *scores* bajos [23]. Como veremos más adelante en la sección en la que tratamos las *normalizaciones*, una de las propiedades deseables de un *score* es que se aproxime a alguna distribución estadística para poder ser interpretado en términos de un $P - valor$, con el objeto de establecer un umbral de significatividad. En el caso de TM-score, valores por encima de $0{,}4 - 0{,}5$ se consideran significativos [28], y hay una probabilidad alta de que se pueda considerar en el mismo *fold* por encima de 0,56 [23].

Hasta ahora, en el proceso de superposición óptima intentamos maximizar el número de residuos alineados considerando toda la proteína en los movimientos de cuerpo rígido. Pero entonces, si tuviéramos una proteína que posee dos conformaciones muy distintas, por ejemplo como consecuencia de que su función así lo requiere, tendríamos serias dificultades para encontrar un alineamiento y superposición

óptima, como se muestra en la Figura 1.4 esquemáticamente y un caso real en la Figura 1.5. Esto podría suceder por ejemplo en aquellas proteínas que tienen un cambio conformacional denominado de bisagra (del inglés hinge), en el que dos subdominios de la proteína se mueven relativamente uno respecto del otro como dos cuerpos rígidos con un eje (o bisagra) común.

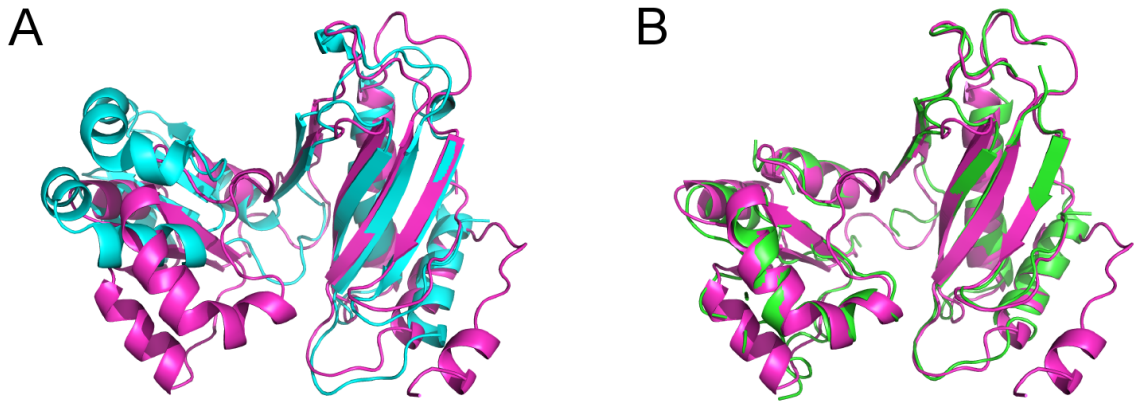

Figura 1.5: Alineamiento estructural con superposición rígida (izquierda) y flexible (derecha), obtenida con ProtDeform [24], entre la proteína ribosomal L1 de la arquea metanógena *janaschii* (azul y verde respectivamente, con PDB '1cjs') y un mutante (PDB '1ad2') cuyo cambio fundamental es el reemplazamiento de la serina 179 por una cisteína. Este organismo es uno de los microbios que es capaz de vivir en condiciones más extremas, por lo que el estudio de su genoma despierta un gran interés. El alineamiento flexible permite encontrar una similitud estructural mucho mayor que el que obtenemos en el alineamiento rígido, observamos como contrapartida la introducción de gaps que se hace patente en las regiones con discontinuidades. Los alineamientos han sido gentilmente cedidos por Jairo Rocha.

Para encontrar una solución a este tipo de situaciones se han desarrollado los denominados *algoritmos de alineamiento flexible* como ProtDeform [24] o el alineamiento múltiple Matt [16]. La idea general de estos algoritmos es muy similar a la de los algoritmos que hemos presentado, pero su diferencia fundamental reside en el proceso de superposición óptima. En este proceso, se permite separar los movimientos de cuerpo rígido en regiones independientes, de modo que si existen cambios conformacionales podremos encontrar la superposición óptima como concatenación de superposiciones. La dificultad residirá entonces en determinar automáticamente en qué situación hay que dividir el problema, ya que una excesiva fragmentación podría permitir valores muy altos de la función de optimización considerada, pero sin sentido biológico. Por tanto, la determinación del tamaño de los dominios óptimo implica en sí mismo un proceso adicional de optimización que hace que estos algoritmos sean computacionalmente más pesados. Pero progresivamente vemos que las mejoras en estos algoritmos los convierten en alternativas muy atractivas respecto de otros algoritmos más establecidos. Por ejemplo, en el caso de ProtDeform se han realizado modificaciones orientadas a incrementar la velocidad y conseguir un *score* también independiente de la longitud, consiguiendo muy buenos resultados en relación al problema de la clasificación [23].

Terminamos esta sección comentando la metodología que usa Dali para construir el alineamiento global, ya que su aproximación es distinta a la del resto de los algoritmos. El hecho de que Dali considere matrices de contactos hace la comparación independiente del sistema de referencia, lo cual es una ventaja muy importante ya que evita realizar el proceso de superposición óptima mediante movimientos de cuerpo rígido. Para reconstruir el alineamiento en Dali se propone una medida en el que se comparan las distancias internas entre ambas proteínas, a través de las matrices de distancias respectivas.

Sean dos pares de residuos alineados entre las proteínas A y B :(a_i, b_i) y (a_j, b_j). Si consideramos las

distancias internas entre los residuos a_i y a_j, que recordamos hemos llamado d_{ij}^A y la correspondiente distancia entre los residuos b_i y b_j, que llamamos d_{ij}^B, definimos el *score* ϕ entre las posiciones alineadas i y j como:

$$\phi(i,j) = \begin{cases} \left(\theta - \frac{|d_{ij}^A - d_{ij}^B|}{\bar{d}_{ij}}\right) w(\bar{d}_{ij}) & i \neq j \\ \theta & i = j \end{cases}$$

donde $\bar{d}_{ij} = (d_{ij}^A + d_{ij}^B)/2$, el parámetro $\theta = 0{,}2$ nos indica que hay un 20 % de tolerancia en las distancias de contacto típicas que podemos encontrar, y $w(\bar{d}_{ij}) = exp(-\bar{d}_{ij}^2/\alpha^2)$ es una función que pesa la importancia de la diferencia que hemos encontrado en relación al tamaño típico de un dominio estructural, que es de $\alpha = 20\mathring{A}$. De este modo los contactos muy alejados en distancia se consideran, pero se infravaloran a través de esta función. A esta medida se le califica como *medida elástica*, porque permite ser tolerante con la acumulación de variaciones geométricas en la exploración de la reconstrucción global óptima.

Equipados con esta función, *Dali reconstruye el alineamiento global explorando al azar las posibles combinaciones de submatrices con un algoritmo de Montecarlo* con criterio de Metrópolis. En esta búsqueda, el criterio de aceptación se construye alrededor de una medida de similitud global que se define como:

$$S = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} \phi(i,j)$$

de modo que la probabilidad de aceptación será mayor si esta medida global crece. Se consideran distintos mecanismos de control en la reconstrucción del alineamiento tanto para que el mapeo entre residuos sea inyectivo (evitando redundancias) como para encontrar la reconstrucción óptima, ya que el carácter estocástico de la búsqueda en principio podría dar resultados diferentes cada vez que se alinean dos proteínas. Esto hace que *este algoritmo, aunque es probablemente el más preciso, sea también el más pesado*. Por este motivo se han buscado modificaciones que lo hacen más ligero creando además una base de datos para almacenar alineamientos muy útil para cálculos masivos, ya que el ejercicio se reduce a una búsqueda en la base datos en vez de el cálculo de nuevo de los alineamientos, el denominado DaliLite [7].

1.5. Medidas de similitud

1.5.1. Medidas crudas y normalizaciones

En general, la mayoría de los algoritmos de alineamiento tienen como salida algunas medidas típicas. Una de estas medidas suele ser el *porcentaje de residuos alineados*, conocido como PSI (del inglés *percentage of structural identity*) definido como $PSI = N_c/n_{min}$. También podemos encontrar el $RMS(N_c)$ definido en la Ecuación 1.4 o el *solapamiento de contactos* entre ambas proteínas, que denominamos q. Como hemos indicado anteriormente, estas medidas que podríamos denominar "crudas", tienen la desventaja de contener una dependencia importante con la longitud de las proteínas alineadas, con lo que todos los algoritmos dan alguna medida adicional que evita este tipo de sesgos y permite además darnos una idea de la significatividad de la medida. Ponemos a continuación algunos ejemplos de estas medidas.

En algoritmos como MAMMOTH, existe una dependencia clara de estas medidas con respecto a la longitud. Por ejemplo, si consideramos todos los pares de proteínas que tienen un valor de n_{min} determinado y pintamos el *histograma de su PSI*, observaremos que en todos los casos se aproxima a una

distribución denominada de valores extremos (**??**) para comparaciones entre pares de proteínas que no tienen homología reconocible, lo que sugiere un patrón similar para comparaciones al azar. Encontraríamos un comportamiento similar para otras medidas como el solapamiento de contactos. Podemos entonces normalizar nuestra medida si encontramos los parámetros de la distribución para cada valor de n_{min}. Supongamos que, en vez de seguir una distribución de valores extremos, sigue una distribución gaussiana. Esta suposición no es en absoluto cierta porque son distribuciones bien diferenciadas. Pero simplificará la explicación y la diferencia en los valores que obtendríamos tras normalizar no es muy importante desde el punto de vista práctico si los valores que consideramos de PSI son suficientemente altos, es decir, en el extremo de la cola superior de la distribución de datos observados.

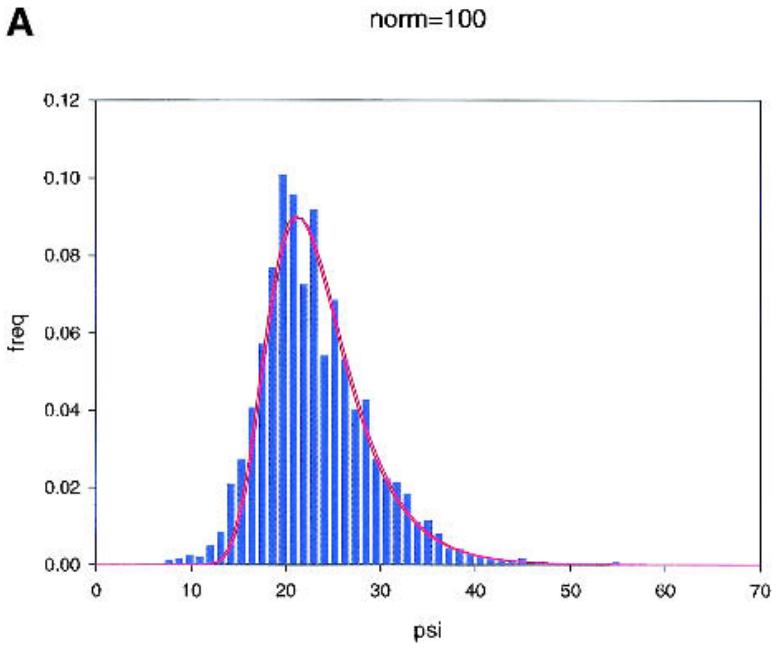

Figura 1.6: Distribución del PSI para pares de proteínas con $n_{min} = 100$ obtenido de [20]. Los datos se ajustan bien a una distribución de valores extremos (curva roja) para todos los valores de n_{min}. Al igual que una distribución gaussiana, esta distribución tiene dos parámetros que podremos obtener como función de n_{min} y que llamamos $\overline{PSI}(n_{min})$ y $\sigma_{PSI}(n_{min})$. En nuestro caso, estas funciones se ajustan bien a funciones analíticas (en particular a leyes de potencias) lo que permite el cálculo directo de una medida como un $Zscore$ para cualquier par. Figura representada con permiso del editor.

Con esta consideración procederíamos calculando para cada valor de n_{min} la media $\overline{PSI}(n_{min})$ y desviación estándar $\sigma_{PSI}(n_{min})$, que vemos son funciones de n_{min}. En la Figura 1.6 podemos observar la distribución de PSI para pares no relacionados con $n_{min} = 100$. Podemos por tanto proponer una medida que evalúe la significatividad de un determinado valor del PSI, conocido el valor de n_{min} del par analizado, como un $Zscore$ (**??**):

$$Zscore = \frac{PSI - \overline{PSI}(n_{min})}{\sigma_{PSI}(n_{min})} \tag{1.5}$$

Y sabemos que un valor de $Zscore > 2$ se puede considerar como significativo. Sin embargo tenemos que notar aquí que, cuando comparamos dos proteínas, no sabemos si están o no relacionadas a priori evolutivamente. Por tanto, aplicar este procedimiento podría penalizar la significatividad de aquellas proteínas que, estando relacionadas tienen un valor de $\sigma_{PSI}(n_{min})$ grande, ya que esta normalización la vamos a calcular a ciegas, es decir, sin hacer suposiciones previas sobre si están o no relacionadas evolutivamente.

Un modo de evitar esta penalización es realizar un estudio para ver si existe cierto valor ϵ de la medida de similitud que estamos utilizando (sea el porcentaje de identidad estructural, el solapamiento de contactos u otro similar) por encima del cual sabemos que existe una relación evolutiva. Si encontramos este valor, que dependerá también de la longitud, $\epsilon = \epsilon(n_{min})$, podemos evitar la normalización sugerida en la Ecuación 1.5. ¿Qué expresión podemos seguir en este caso y qué normalización habrá que realizar? Podemos proponer aquí una hipótesis evolutiva.

1.5.2. Una medida con motivación evolutiva

Sabemos que la probabilidad $P\{A_i = B_i\}$ de que dos residuos de las proteínas A y B, que son homólogas, sean iguales en secuencia en la posición i decae con el tiempo con un ritmo $1/\tau$ de manera exponencial: $exp(-t/\tau)$. También podemos estimar la probabilidad condicionada de que, si ha habido al menos un cambio en una de las dos secuencias, suceso que tiene probabilidad $(1 - exp(-t/\tau))$, las dos secuencias tengan el mismo residuo. La probabilidad p del suceso "tener el mismo residuo" se puede estimar a partir de las frecuencias relativas f de cada tipo de residuo a como $p = \sum_a f(a)^2$. Por tanto la probabilidad condicionada será: $p\,(1 - exp(-t/\tau))$ y la probabilidad final que buscamos será:

$$P\{A_i = B_i\} = e^{-t/\tau} + p\left(1 - e^{-t/\tau}\right) \tag{1.6}$$

Si asumimos que cada posición evoluciona de manera independiente, lo cual no es cierto pero es una asunción casi inevitable, podemos relacionar esta probabilidad con la identidad en secuencia (porcentaje de identidad entre dos secuencias) $P\{A_i = B_i\} \approx SI$, de modo que en un instante de tiempo t la *divergencia evolutiva entre las dos secuencias* se puede calcular, despejando la Ecuación 1.6 como:

$$Divergencia\_secuencia = t/\tau = -ln\left(\frac{SI - p}{1 - p}\right) \tag{1.7}$$

Esta medida, cuando $SI \gg p$ coincide con la fórmula que se obtendría para estimar la divergencia evolutiva con un proceso de Poisson [19], luego estamos considerando un proceso de Poisson corregido. Chothia y Lesk [2] observaron que la divergencia en estructura es también proporcional al tiempo que ha pasado desde que ambas estructuras comenzaron a diverger (**??**). Así que la idea para trabajar con una similitud estructural es que, si la medida que utilizamos es mayor que el umbral $\epsilon(n_{min})$ que determinemos y dada la relación entre la divergencia en secuencia y en estructura encontrada encontrada por Chothia y Lesk, es legítimo proponer una medida similar a la divergencia en secuencia para la *divergencia en estructura* [21]:

$$Divergencia\_estructura = -ln\left(\frac{q - q_\infty(n_{min})}{1 - q_\infty(n_{min})}\right) \tag{1.8}$$

Donde q es el solapamiento de contactos y $q_\infty(n_{min})$ es el análogo de p en el análisis que hemos hecho de secuencia, y hemos colocado el subíndice ∞ para enfatizar que la función proporciona valores del solapamiento de contactos en el límite de tiempos evolutivos muy largos, que también es una función de n_{min}. Por tanto, si nuestra medida de similitud q cumple que $q > \epsilon(n_{min})$ utilizaremos la Ecuación 1.8, mientras que en caso contrario utilizaremos la Ecuación 1.5, que sabemos que funciona bien para pares no relacionados.

La parte positiva de esta medida es que, a pesar de que hay varios parámetros y asunciones a tener en cuenta, nos permite cuantificar la relación entre la divergencia en secuencia y en estructura que, si bien sus valores absolutos hay que tomarlos con precaución, sus valores relativos entre familias de proteínas distintas nos pueden permitir hacer estudios evolutivos [21].

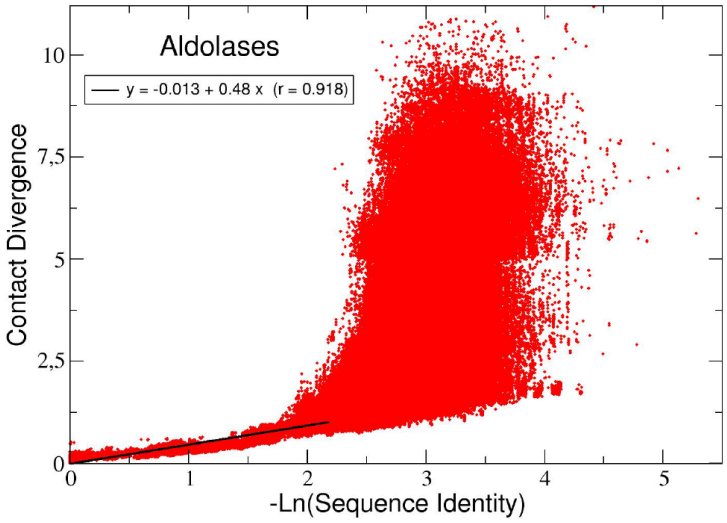

Figura 1.7: Relación entre la divergencia en secuencia y estructura para la superfamilia de las aldolasas. La definición de la divergencia de contactos nos permite estimar el ritmo de divergencia entre ambas, para lo cual se realiza un ajuste en la región compatible con una ordenada en el origen significativamente cercana a cero. Más detalles en [21].

Esta medida tiene un comportamiento similar al denominado TM-score que vimos en la sección en la que discutíamos el proceso de superposición típico de los algoritmos rígidos (ver Figura 1.4). Sin embargo no se construye con ninguna hipótesis evolutiva y su interpretación es menos directa.

1.6. Alineamiento múltiple

Nuestro siguiente objetivo consiste en alinear estructuralmente un conjunto arbitrario de proteínas. Queremos encontrar el alineamiento común que contenga el máximo número de residuos con la menor distancia espacial posible entre sus residuos, medida con alguna expresión como el RMS mostrado en la Ecuación 1.4. De este modo, si tenemos ya no un par sino todo un conjunto de proteínas homólogas, el encontrar el mayor número de posiciones espaciales conservadas implica que podemos calcular la variabilidad en secuencia que se observa en dichas posiciones, lo cual nos permite establecer hipótesis sobre la evolución y funciones de los miembros del conjunto. Por tanto sus aplicaciones son numerosas, desde la mejora de los mismos alineamientos en secuencia (**??**) a la mejora de la clasificación de estructura de proteínas (**??**), puntos que giran a su vez alrededor de los métodos de predicción de estructura.

Los *algoritmos de alineamiento múltiple* utilizan aproximaciones variadas y no vamos a repasar aquí las diferencias entre los distintos métodos con el mismo detalle que hemos realizado el análisis de los métodos de alineamiento de pares. Vamos a aprovechar la descripción que ya hemos realizado del alineamiento de pares MAMMOTH para ver cómo se generaliza el algoritmo de manera natural a su versión múltiple: MAMMOTH-mult [14]. El lector que haya seguido la explicación de la sección de alineamiento de pares podrá, a la vista de la generalización de este algoritmo, seguir fácilmente otras aproximaciones como el ya mencionado Matt [16], o cómo se determinan los clusters estructurales en la base de datos HOMSTRAD [17].

Consideremos un conjunto P de proteínas que queremos alinear. Un comienzo razonable podría ser el mismo que realiza la versión de pares de MAMMOTH, a saber, se divide cada proteína en heptámeros y se comparan los heptámeros todos contra todos de cada par de proteínas del grupo. Hasta aquí, realizaríamos el mismo análisis local que en dicha versión si la corriéramos para cada par de proteínas. Pero recordemos que tenemos que encontrar después el alineamiento óptimo para todas las proteínas, optimizando globalmente el mismo con operaciones de cuerpo rígido. Y hay muchas combinaciones posibles para ir seleccionando proteínas y optimizando dicha superposición, lo que nos lleva a un problema computacionalmente demasiado pesado si quisiéramos considerar todos los modos en que seleccionamos subconjuntos de proteínas. Así que MAMMOTH-mult realiza un par de pasos previos a la construcción del alineamiento múltiple.

1.6.1. Primeros pasos

El *primer paso* consiste efectivamente en realizar un *alineamiento de pares de todas las proteínas contra todas* con la versión estándar de pares. El *segundo paso* consistirá en, a partir de la medida de similitud que obtenemos del algoritmo bien normalizada, *utilizar un algoritmo aglomerativo* (*average linkage* (**??**)) *para relacionar jerárquicamente las estructuras en un árbol*. De este modo tenemos una aproximación razonable al ordenamiento ideal en el que tendríamos que ir incorporando estructuras para la reconstrucción del alineamiento múltiple. Obviamente observamos ya un par de elecciones arbitrarias en este primer paso (el algoritmo de alineamiento de pares con una medida concreta asociada, y el algoritmo aglomerativo) y otras opciones podrían determinar resultados distintos. Por este motivo en los algoritmos es importante hablar de la robustez del método, es decir, cuánto dependen sus resultados de las elecciones que se realizan. Vamos a asumir que el método es robusto ante estas elecciones y veamos en qué consiste específicamente el alineamiento múltiple.

1.6.2. Construcción del alineamiento

Una vez que tenemos el árbol resultado del proceso de aglomeración, lo recorremos desde las hojas hasta la raíz, parándonos en cada nudo en donde dos ramas se separan. Si llamamos A y B a las ramas (ya no a las proteínas), que ahora contendrán P_A y P_B proteínas ($P_A + P_B \leq P$), se van a dividir los alineamientos ya definidos en cada rama en heptámeros, y se van a utilizar los $URMS$ calculados en la comparación previa de pares, pero donde ahora queremos pesar la contribución de cada proteína a cada heptámero. Siguiendo la notación de la Ecuación 1.1, de los heptámeros u y v que pasan por las posiciones $i \in A$ y $k \in B$ asignamos a cada par de posiciones el valor $URMS$ más pequeño, es decir $URMS_{ik} = min(URMS_{uv});\ i \subset u,\ k \subset v$. Y entonces calculamos el promedio para cada par de posiciones como:

$$URMS_{ik}^{AB} = \frac{1}{P_A}\frac{1}{P_B}\sum_{a=1}^{P_A}\sum_{b=1}^{P_B} URMS_{ikab} \tag{1.9}$$

cuyos valores se pueden normalizar del mismo modo que en la Ecuación 1.2 para obtener una matriz de similitudes. Los siguientes pasos, al igual que MAMMOTH de pares, consiste en realizar un alineamiento local con Needleman-Wunsch [18] para conseguir las correspondientes asignaciones entre residuos y se realiza una superposición tridimensional utilizando también MaxSub [27].

La novedad consiste en que, a partir de la superposición, queremos obtener una nueva *matriz de similitud* al estilo de la construida en la Ecuación 1.9, pero ahora teniendo en cuenta las posiciones espaciales obtenidas mediante la superposición:

$$S_{ik}^{AB} = \frac{1}{P_A}\frac{1}{P_B}\sum_{a=1}^{P_A}\sum_{b=1}^{P_B}\left(w_e URMS_{ikab} + w_d e^{-\alpha d_{ikab}^2}\right) \tag{1.10}$$

donde d_{ikab} es la distancia euclídea entre las posiciones i y k de las proteínas a y b, y se introducen los parámetros α que controla el decaimiento en función de la distancia. w_e junto con w_d controlan el peso relativo que daremos a la componente relacionada con el esqueleto (*backbone*) y la distancia cartesiana, respectivamente. Dichos parámetros se optimizan utilizando alineamientos estructurales curados manualmente, utilizando un algoritmo de Monte Carlo por ejemplo.

El último paso consiste en forzar al conjunto de residuos alineados para todas las proteínas, el denominado *core* del alineamiento compuesto por N_c residuos, a tener una *superposición óptima* para todas las proteínas. Para ello se van seleccionando las proteínas que comprenden el alineamiento de manera ordenada y, para cada una de ellas, se realizan movimientos de cuerpo rígido manteniendo a todas las demás fijas hasta encontrar un valor de $RMS(N_c)$ mínimo, calculado según la Ecuación 1.4. Este proceso se itera sistemáticamente hasta que la siguiente función de error converge:

$$\varepsilon_{core} = \sum_{m=1}^{N_c}\sum_{a\neq b}^{P_A+P_B} d_{mab}^2$$

donde nuevamente d_{mab} es la distancia euclídea para la posición alineada m entre las proteínas a y b.

Una vez hemos conseguido que converja el error, iremos recorriendo el árbol hasta el siguiente nudo, hasta que lleguemos a la raíz en donde $P_A + P_B = P$. Como salida obtendremos todos los cálculos realizados en los primeros pasos por el algoritmo de comparación de pares y algunas medidas propias del alineamiento múltiple. Algunas de estas medidas son el *tamaño del core* denominado *estricto*, que es aquél con un 100 % de conservación de sus residuos en secuencia que están además alineados (cualesquiera de las parejas de proteínas consideradas) a menos de 4Å. También se define un *core* denominado *laxo*, que es aquél con un 66 % de conservación y donde las proteínas están alineados

a menos de $3\mathring{A}$ respecto del promedio del alineamiento en cada una de las posiciones. Sobre estas definiciones se pueden realizar medidas generales como el RMS promedio de cada tipo de *core* o su desviación típica.

En la Figura 1.8 presentamos el alineamiento múltiple de 23 proteínas de la familia de las globinas según están clasificadas en la clasificación estructural manual de SCOP (ver sección de evolución de estructura de proteínas). Esta superfamilia está subdividida en familias correspondientes a las globinas canónicas de unión al grupo hemo (código 46463), las globinas truncadas de protozoo/bacteria (46459) y las hemoglobinas neuronales (74660). El alineamiento estructural y el dendograma que se puede reconstruir del protocolo de alineamiento, muestra una clara separación de los tres grupos. Se representa además el alineamiento en secuencia consecuencia del alineamiento estructural. De este último alineamiento es interesante observar la conservación en la posición estructural de la histidina situada en la posición 113. Esta histidina está situada en el sitio activo, y establece la unión con el átomo de hierro del grupo hemo.

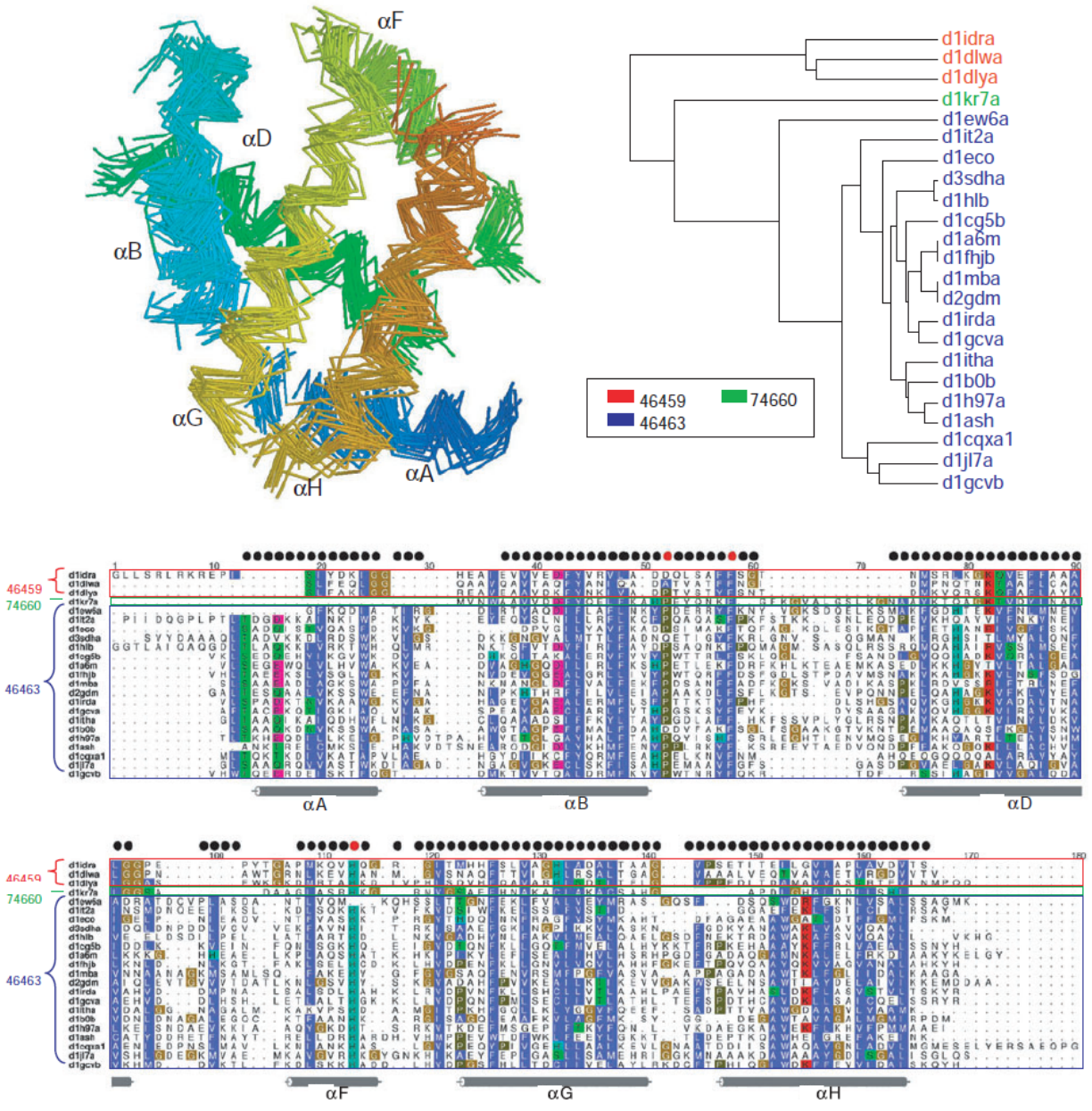

Figura 1.8: Alineamiento estructural de la superfamilia de las globinas (arriba izquierda) obtenido con MAMMOTH múltiple [14]. Arriba a la derecha se muestra el dendograma asociado, y abajo el alineamiento en secuencia. Más detalles en el texto. Figura reutilizada con permiso del editor.

1.7. Discusión

En la presente sección hemos abordado el problema del alineamiento de estructura de proteínas. Hablando en abstracto, *el problema del alineamiento se podría interpretar como la búsqueda de un sistema de referencia común a un conjunto de entidades, con el objeto de identificar la información que tienen en común.* En nuestro caso, esta información está codificada en la estructura de las proteínas, y su identificación hemos visto que nos abre numerosas puertas. Los distintos eventos evolutivos que conocemos, son los mecanismos mediante los cuales está información ha fluido en el universo de estructura de proteínas que podemos observar actualmente. Por otra parte, el modo en el que la información es capaz de transmitirse y modificarse a lo largo del proceso evolutivo está condicionado a los requerimientos biológicos derivados de la funcionalidad de las proteínas. La posibilidad de que un cambio evolutivo se fije en una proteína, vendrá determinado por su viabilidad termodinámica y cinética, que son condiciones necesarias para que la proteína realice su función. Esto nos llevará a interesarnos por la física del plegamiento de proteínas. Pero no son condiciones suficientes, porque pueden existir aminoácidos cuya localización concreta es esencial para realizar la función, como vimos con la histidina en el ejemplo de las globinas, lo que nos lleva a interesarnos por las particularidades químicas de las mismas.

El alineamiento estructural es, por tanto, una herramienta muy potente en tanto en cuanto nos permite abrir *una ventana en la cual asomarse a todo el proceso evolutivo.* Qué eventos son dominantes, qué estructuras son las más relevantes o qué motivos locales determinan la función, son estudios que pueden fundamentarse a partir de los análisis obtenidos mediante el alineamiento estructural junto con la información proveniente de la secuencia y de las funciones conocidas. Con todas estas posibilidades en mente, invitamos al lector a afrontar las secciones de plegamiento de proteínas (**??**), evolución de estructura de proteínas (**??**) y modelización. El recorrido que proponemos entrará en la física con los modelos de plegamiento para después asomarse al pasado en la sección de evolución. Allí llegaremos incluso a conjeturar sobre los eventos dominantes en instantes de tiempo incluso anteriores a los dominios estructurales tal y como los conocemos, con una aproximación fuertemente basada en la comparación mediante alineamientos estructurales. Todo este recorrido nos hará llegar finalmente a la modelización, que será nuestro objetivo desde el punto de vista práctico más potente, pues su aplicación permite ampliar nuestras posibilidades de comprender la función de proteínas cuyo rol es desconocido, con la potencialidad que conlleva el comprender dicha función desde el punto de vista de las aplicaciones biomédicas.

Agradecimientos

El autor agradece especialmente la revisión crítica del manuscrito a Jairo Rocha. Este trabajo ha sido posible gracias a la financiación del Ministerio de Economía y Competitividad a través de una beca FPI (BES-2009-013072). La investigación en el CBMSO es apoyada por la Fundación Ramón Areces.

1.8. Bibliografía

[1] P. Bork, C. Sander, and A. Valencia. Convergent evolution of similar enzymatic function on different protein folds: The hexokinase, ribokinase, and galactokinase families of sugar kinases. *Protein Science*, 2(1):31–40, 1993.

[2] C. Chothia and A. M. Lesk. The relation between the divergence of sequence and structure in proteins. *The EMBO Journal*, 5(4):823–826, Apr. 1986. PMID: 3709526 PMCID: PMC1166865.

[3] G. Csaba, F. Birzele, and R. Zimmer. Protein structure alignment considering phenotypic plasticity. *Bioinformatics*, 24(16):i98–i104, Aug. 2008.

[4] H. Hasegawa and L. Holm. Advances and pitfalls of protein structural alignment. *Current Opinion in Structural Biology*, 19(3):341–348, June 2009.

[5] H. Hegyi and M. Gerstein. Annotation transfer for genomics: Measuring functional divergence in multi-domain proteins. *Genome Research*, 11(10):1632–1640, Oct. 2001.

[6] F. G. Hoffmann, J. C. Opazo, and J. F. Storz. Gene cooption and convergent evolution of oxygen transport hemoglobins in jawed and jawless vertebrates. *Proceedings of the National Academy of Sciences of the United States of America*, 107(32):14274–14279, Aug. 2010. PMID: 20660759.

[7] L. Holm and J. Park. DaliLite workbench for protein structure comparison. *Bioinformatics*, 16(6):566–567, June 2000.

[8] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233(1):123–138, Sept. 1993.

[9] T. Kawabata. MATRAS: a program for protein 3D structure comparison. *Nucleic Acids Research*, 31(13):3367–3369, July 2003.

[10] K. Kedem, L. P. Chew, and R. Elber. Unit-vector RMS (URMS) as a tool to analyze molecular dynamics trajectories. *Proteins: Structure, Function, and Bioinformatics*, 37(4):554–564, 1999.

[11] L. N. Kinch and N. V. Grishin. Evolution of protein structures and functions. *Current Opinion in Structural Biology*, 12(3):400–408, June 2002.

[12] P. Lackner, W. A. Koppensteiner, M. J. Sippl, and F. S. Domingues. ProSup: a refined tool for protein structure alignment. *Protein Engineering*, 13(11):745–752, Nov. 2000.

[13] X. Liu, Y.-P. Zhao, and W.-M. Zheng. CLEMAPS: multiple alignment of protein structures based on conformational letters. *Proteins: Structure, Function, and Bioinformatics*, 71(2):728–736, 2008.

[14] D. Lupyan, A. Leo-Macías, and Á. R. Ortiz. A new progressive-iterative algorithm for multiple structure alignment. *Bioinformatics*, 21(15):3255–3263, Aug. 2005.

[15] L. Martínez, R. Andreani, and J. M. Martínez. Convergent algorithms for protein structural alignment. *BMC Bioinformatics*, 8(1):306, Aug. 2007.

[16] M. Menke, B. Berger, and L. Cowen. Matt: Local flexibility aids protein multiple structure alignment. *PLoS Comput Biol*, 4(1):e10, Jan. 2008.

[17] K. Mizuguchi, C. M. Deane, T. L. Blundell, and J. P. Overington. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Science*, 7(11):2469–2471, 1998.

[18] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, Mar. 1970.

[19] M. Nei and S. Kumar. *Molecular Evolution and Phylogenetics*. Oxford University Press, June 2000.

[20] A. R. Ortiz, C. E. Strauss, and O. Olmea. MAMMOTH (matching molecular models obtained from theory): An automated method for model comparison. *Protein Science*, 11(11):2606–2621, 2002.

[21] A. Pascual-García, D. Abia, R. Méndez, G. S. Nido, and U. Bastolla. Quantifying the evolutionary divergence of protein structures: The role of function change and function conservation. *Proteins: Structure, Function, and Bioinformatics*, 78(1):181–196, 2010.

[22] A. Pascual-García, D. Abia, Á. R. Ortiz, and U. Bastolla. Cross-over between discrete and continuous protein structure space: Insights into automatic classification and networks of protein structures. *PLOS Computational Biology*, 5(3):e1000331, Mar. 2009.

[23] J. Rocha and R. Alberich. The significance of the ProtDeform score for structure prediction and alignment. *PLoS ONE*, 6(6):e20889, June 2011.

[24] J. Rocha, J. Segura, R. C. Wilson, and S. Dasgupta. Flexible structural protein alignment by a sequence of local transformations. *Bioinformatics*, 25(13):1625–1631, July 2009.

[25] M. I. Sadowski and W. R. Taylor. Evolutionary inaccuracy of pairwise structural alignments. *Bioinformatics*, 28(9):1209–1215, May 2012.

[26] I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 11(9):739–747, Sept. 1998.

[27] N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer. MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, 16(9):776–785, Sept. 2000.

[28] Y. Zhang and J. Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

[29] Y. Zhang and J. Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, Jan. 2005.