Algoritmo Rápido de Reducción de Trajectorias de Plegamiento de Proteínas

Luis Ernesto Garreta U.

9 de septiembre de 2019

Índice

Resumen

Gracias a los avances en hardware y software, las simulaciones de plegamiento de proteínas están experimentando grandes progresos, alcanzando tiempos de simulación sin precedentes, en el orden de los microsegundos y milisegundos. Como consecuencia, las trayectorias generadas por estas simulaciones son muy extensas, con miles y millones de conformaciones, lo que genera problemas tanto en tiempo como en espacio para procesarlas y analizarlas. Una manera de sobrepasar estos problemas es desarrollar algoritmos rápidos para generar trayectorias reducidas que preserven tanto como sea posible las características de las trayectorias originales, especialmente el orden temporal y la estructura de las conformaciones.

En este artículo presentamos un algoritmo para reducir este tipo de trayectorias que divide la trayectoria en segmentos y por cada segmento extrae los eventos más disimilares, mediante una estrategia rápida de agrupamiento, y luego selecciona a los más representativos, mediante una estrategia de agrupamiento global. El algoritmo aprovecha el orden temporal implícito en la trayectoria para realizar en cada segmento comparaciones locales y evitar la comparación de todos contra todos, que se vuelve impráctica computacionalmente cuando son muchas conformaciones. De esta manera, el algoritmo reduce muy rápidamente la trayectoria y las conformaciones seleccionadas conservan tanto su estructura como su orden temporal. Además, la partición por segmentos permite al algoritmo reducir cada segmento de forma independiente y paralela, lo que lo vuelve aún más rápido cuando se ejecuta en máquinas multi-core, muy comunes hoy en día.

1. Introducción

Las simulaciones del plegamiento de proteínas han demostrado ser de gran utilidad para estudiar las mecanismos subyacentes de este proceso el cual permite a una cadena de aminoácidos plegarse hasta alcanzar su estructura tridimensional única y convertirse en una proteína activa habilitada para ejecutar una función biológica. Gracias a los avances en hardware y software, estas simulaciones han experimentado grandes progresos, con tiempos de simulación en el orden de los milisegundos, y llevadas a cabo usando diferentes tecnologías, desde costosas supercomputadoras especializadas [9], hasta, económicos arreglos de tarjetas gráficas [11], e incluso PCs distribuidos alrededor del mundo [4]. Muchas de estas simulaciones alcanzan tiempos que antes no se lograban debido a las limitaciones en los recursos computacionales, y las trayectorias generadas por estas simulaciones se caracterizan por abarcar miles o millones de conformaciones, lo cual a pesar de ser una gran ventaja porque se tiene más detalle del proceso, así mismo es un problema debido al tiempo y recursos computacionales necesarios para analizarlas. Por esta razón, se necesitan nuevos algoritmos capaces de reducir estas trayectorias de una forma rápida, y que logren preservar la mayor información posible tanto a nivel de representación como a nivel de orden temporal de las conformaciones de la trayectoria.

Para realizar estas reducciones se han utilizado diferentes técnicas que básicamente caen en dos enfoques: la reducción de dimensionalidad [2] y el agru-

pamiento [13], las cuales más que reducir las trayectorias realizan un análisis sobre ellas. Mientras que en el primer enfoque se transforma las conformaciones a una forma simplificada para poder interpretar los resultados; en el segundo, se obtienen conformaciones representativas de grupos de conformaciones con propiedades comunes. Sin embargo, aunque en el primer enfoque se conserva el orden temporal de las conformaciones, su estructura se pierda ya que su representación se modifica; por el contrario, aunque en el segundo enfoque se conserva la estructura de las conformaciones, el orden temporal se pierde ya que los grupos pueden contener conformaciones de tiempos muy diferentes.

En este artículo presentamos un algoritmo rápido para reducción de trayectorias largas de plegamiento de proteínas que toma como base la estrategia de Hobohm&Sander [5] para agrupamientos rápidos y que se basa en tres estrategias: primero una partición de la trayectoria en múltiples segmentos; segundo, una reducción local muy rápida sobre cada una de ellos que aprovecha el orden temporal de las conformaciones; y tercero, una reducción global que busca encontrar las conformaciones más representativas de cada segmento. Estas tres estrategias permiten que este algoritmo sea rápido, fácilmente paralelizable, y produzca trayectorias reducidas que conservan tanto la estructura como el orden temporal de las conformaciones.

2. Antecedentes

2.1. Técnicas de reducción de trayectorias de plegamiento

Para la reducción de trayectorias de plegamiento de proteínas se han utilizado varios métodos como los basados en agrupamientos, basados en transformaciones lineales como el análisis de componentes principales (PCA) y el escalamiento multidimensional (MDS), y los que cambia la representación de la estructura como los basados en mapas de contactos.

Los basados en agrupamientos de estructuras se implementa en varias herramientas de simulación de plegamiento como el algoritmo de agrupamiento de GROMACS [**?**] donde toman todas las estructuras, miden la distancia entre ellas, toman como representativa la que más vecinos tenga de acuerdo a un valor de corte *(cutoff),* la eliminan junto a sus vecinos, y repiten el prtooceso para las restantes. Sin embargo, este tipo de algoritmos generalmente dependen de distintos parámetros tales como la especificación inicial del rádio o número de grupos, o la medida de similaridad para comparar las estructuras. Estos parámetros tienden a hacer artificial el agrupamiento donde los cambios del valor de alguno de los parámetros, pueden producir resultados que varian de forma considerable.

Entre los basados en transformaciones tanto lineales como no-lineales están los que usan PCA y MDS. Los que usan PCA [2] transforman la estructura de la proteína desde un espacio N-dimensional–dado por los puntos de datos de las N coordenadas de sus átomos–a un espacio lineal K-dimensional ($K < D$) que corresponde a un nuevo sistema de coordenadas llamado componentes prin-

cipales. Estos componentes representan los vectores tangentes que describen un hiperplano que pasa a través de los puntos de datos tanto como sea posible cuando se evalúan sus mínimos cuadrados. Los componentes se ordenan de acuerdo a su varianza y los primeros (los de mayor varianza) son los que resumen mejor los cambios conformacionales globales de la proteína. Sin embargo PCA tiene problemas cuando los espacios son no-lineales, como se piensa que es el espacio conformacional de la proteína y por lo tanto el nuevo espacio K-dimensional puede resultar distorsionado [1].

Para evitar el problema de la linearidad con el PCA, Rajan et al. 2010 [14] adaptan un método de escalamiento multidimensional no métrico (*nMDS)* para obtener una representación reducida 2D de toda la trayectoria. Inicialmente transforman la estructura 3D de la proteina a sus respectivos ángulos dihédricos para luego aplicarles el método de escalamiento y obtener un conjunto de puntos que representa las estructuras de proteínas. Estos puntos se despliegan sobre un espacio métrico (generalmente 2D) que representa la trayectoria de tal manera que la distancia cada par de puntos *x,y* es consistente con las distancias de cada par de estructuras *X,Y* representadas por los respectivos puntos. Aunque esta forma de reducción simplifica a 2D las estructuras N dimensionales (N coordenadas XYZ de sus átomos), la información de la estructura se pierde y la reducción se vuelve específica para ciertos análisis como el de analizar visualmente la ocurrencia de eventos en el tiempo.

Entre los que cambian la representación de la estructura Yang et al. 2007 [17] transforman la estructura a un mapa de contacto (matriz 2D binaria) a través de lo que definen como SOAPs o patrones 2D no locales que se encuentran en los mapas de la estructura. Se encuentran los SOAPs de todas las estructuras, se los agrupa por SOAPs comúnes de acuerdo a una medida de distancia, y se obtienen los más frecuentes. Así, las principales partes de la estructura se representa con los SOAPs más frecuentes y las otras partes se eliminan, lo que lleva a una representación más concisa. Sin embargo, la reducción cambia sustancialmente los elementos de la trayectoria al trasladar las estructuras 3D a representaciones 2D, perdiendo información implicita en la estructura.

2.2. Simulaciones de Plegamiento

Las simulaciones del plegamiento de proteínas son complejas y demandan gran cantidad de tiempo y recursos computacionales. Debido a estas limitaciones tecnológicas, hasta hace unos años estas simulaciones se realizaban para proteínas pequeñas y los tiempos simulados eran muy cortos, en el orden de los microsegundos mientras que una proteína se pliega en el orden de los milisegundos [6]. Sin embargo, \\en los últimos años los avances en el hardware han logrado avances de tal manera que se empiezan a mostrar resultados de simulaciones mucho más largas y de proteínas más grandes. Dos ejemplos de estos avances son los proyectos de folding@home y de la supercomputadora Anton. El proyecto foldin@home logró realizar hace algunos años una de las primeras simulaciones largas utilizando computación distribuida. Una de sus simulaciones alcanzó el orden de los microsegundos para plegar completamente una pro-

teína pequeña, la Villin Headpiece de 36 residuos [10]. Más recientemente, la supercomupatadora Anton ha usado computación paralela y hardware especializado para simular dinámica molecular [15]. Con está máquina se ha logrado plegar completamente varias proteínas medianas (10-80 residuos), alcanzando tiempos de simulación del orden de los milisegundos. En ambos proyectos los resultados de las trayectorias están disponibles para que la comunidad científica los descargue y los analice para avanzar en el entendimiento del plegamiento de las proteínas.

2.3. Algoritmos Rápidos de Agrupamiento

Muchos de los algoritmos para realizar agrupamientos rápidos de secuencias biológicas se basan en las ideas del algoritmo de Hobohm y Sander [5] que creó inicialmente para agrupar de forma rápida secuencias de proteínas. El algoritmo determina las secuencias más representativas a través de dos actividades: un ordenamiento y una selección rápida. En el ordenamiento, las secuencias se organizan por longitud en orden descendiente, luego se toma la primera secuencia (la más larga) como representativa del primer grupo. En la selección rápida, se compara el resto de secuencias con la representativa y se las incorpora al grupo si son cercanas (ejemplo, si son similares a nivel de secuencias), de lo contrario, pasa a ser la representativa de un nuevo grupo y se hace lo mismo con el resto de secuencias hasta terminar. Los aspectos determinantes del éxito del algoritmo son la relación de orden que se establezca al inicio y las propiedades que se tomen para comparar las secuencias. En secuencias de ADN y de proteínas estos aspectos funcionan bien ya que dos secuencias de más o menos de igual longitud tienen mayor probabilidad de ser similares que dos secuencias de longitudes completamente diferentes. Sin embargo en estructuras tridimensionales de proteínas que pertenecen a una misma trayectoria, la longitud y la similaridad de la secuencia va a ser la misma para todas la conformaciones, lo que implica redefinir estos aspectos en términos de las características de las estructuras 3D de proteínas de una misma trayectoria, como vamos a describir más adelante cuando mostremos nuestro algoritmo de reducción de trayectorias de plegamiento.

Dos de las implementaciones más usadas de este algoritmo para agrupamiento rápido de secuencias son los programas CD-HIT [8] y UCLUST [3]. El programa CD-HIT realiza un ordenamiento por longitud de la secuencia como lo plantea el algoritmo de Hobhon, y para la selección utiliza un filtro de palabras cortas para comparar si dos secuencias son similares–evitando el alineamiento de las mismas–y así asignarlas a un mismo grupo o crear uno nuevo. En el caso de secuencias de proteínas el programa usa por defecto una palabra de 10 aminoácidos o *decapeptido*. En cambio el programa UCLUST utiliza para com parar las secuencias una función creada por los mismos autores que la llaman como USEARCH y que calcula la similitud entre las secuencias a partir de un alineamiento global.

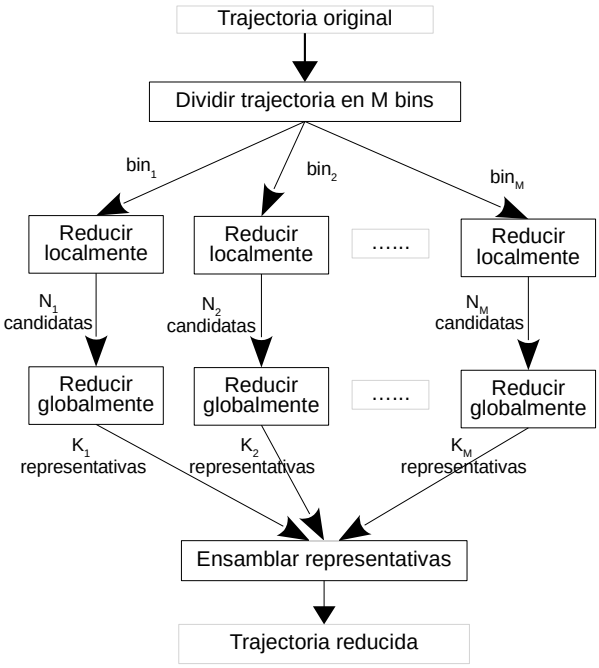


Figura 1: Flujo de datos algoritmo

3. Algoritmo Propuesto

La primera parte del algoritmo realiza un agrupamiento local rápido donde se aprovecha el ordenamiento temporal de las conformaciones implícito en la trayectoria. Para esto, se toma la idea del algoritmo propuesto por Hobohm et al. [5] para la selección de conjuntos de proteínas. Se particiona la trayectoria en *M bins* o secciones de *N* conformaciones contiguas en el tiempo de simulación. Para cada uno de los *bins* se toma la primera estructura como cabeza del primer grupo y se la compara con la siguiente en orden de tiempo de simulación. Si presentan similaridad se adicionan al grupo; de lo contrario si es disimilar se crea un nuevo grupo y se toma a esta última estructura como cabeza del nuevo grupo. El proceso continua hasta terminar con todas las estructuras del *bin* y esto mismo se realiza para los demás *bins*. En la segunda parte del algoritmo, toma cada conjunto de conformaciones cabeza de grupo seleccionadas en cada *bin* y se crea una matriz de similaridades que se la usa para realizar un agrupamiento para seleccionar las *K* estructuras más representativas de cada conjunto tomando los *k-medoides*. La unión de estas *K* estructuras por *bin* crea un nuevo conjunto mucho más reducido que el creado en el agrupamiento local. El orden temporal no se pierde ya que las *K* estructuras seleccionadas por cada conjunto se las ordena de acuerdo a su tiempo original de simulación.

4. Datos y Métodos

4.1. Comparación de Estructuras de Proteínas

Tanto para la primera y segunda fase de reducción utilizamos la medida de similitud entre estructuras de proteína llamada TM-score (Template Modeling score) [**?**]. Esta medida de similitud a diferencia de otras medidas ampliamente usadas en comparación de estructuras como el RMSD (Root Mean Square-Deviation) es más precisa ya que en el TM-score influyen poco sobre el puntaje final las secciones pequeñas de la proteína que alinean incorrectamente, tales como giros simples o términos flexibles, lo que reduce el chance de evaluaciones sesgadas.

4.2. Selección de Estructuras Representativas

Las estructuras representativas de cada grupo o *bin* resultante de la selección rápida de la primera fase se obtienen aplicando en cada uno de ellos un algoritmo de particionamiento alrededor de medoides (PAM) [12]. El algoritmo selecciona como representativa la estructura media o central de cada subgrupo resultante para la cual la suma de las distancias entre esta y las demás estructuras del subgrupo es mínima. Así, al final se obtienen por cada grupo o *bin* inicial un conjunto reducido de estructuras que representan los eventos principales de esta sección de la trayectoria de plegamiento.

4.3. Trayectorias de Plegamiento de Proteínas

Para mostrar los resultados del algoritmo de reducción propuesto, aplicamos las reducciónes a tres trayectorias de plegamiento de proteínas. La dos primeras corresponden a trayectorias cortas (200-300 conformaciones) para las proteínas: ferredoxina desde clostridium acidurici (PDB: 1FCA) y del Cyt férrico de levadura (iso-1-Cytc, PDB: 2YCC) que fueron simuladas por el grupo de Amato mediante el método *Probabilistic Roadmap Method [16]* y que se caracterizan por ser trayectorias cortas que tratan de incluir los eventos principales de la simulación. Por el contrario, la tercera trayectoria corresponde a la simulación de plegamiento mediante la técnica de Dinámica Molecular [9] para la proteína Trp-cage (PDB: 2JOF) y se caracteriza por ser una trayectoria mucho más extensa y detallada (más de 1 millón de conformaciones).

5. Resultados y Discusión

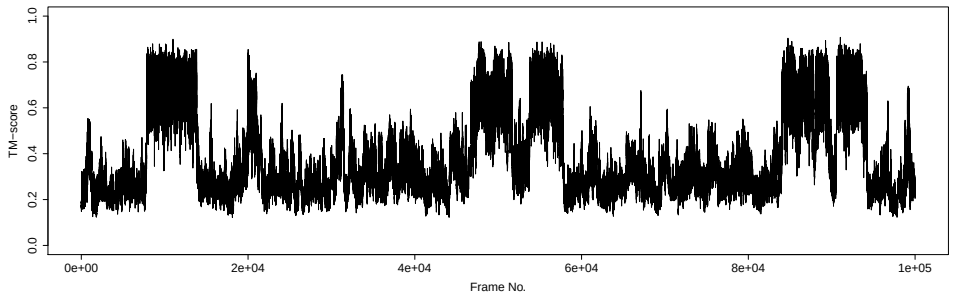
Para mostrar las reducciónes que realiza nuestro algoritmo, presentamos aquí los resultados de la reducción realizada a tres trayectorias de proteínas. Las dos primeras son trayectorias cortas de menos de 300 conformaciones, mientras que la tercera es mucho mas larga con más de 1 millón de conformaciones.

5.1. Reducción sobre un segmento de una trayectoría

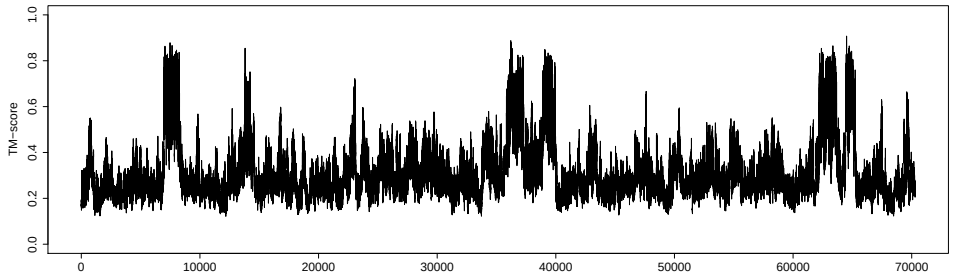
En la figura 2 se observa los resultados de aplicar nuestro algoritmo a un segmento de 1000000 conformaciones de la trayectoria de la proteína villin-headpiece (PDB 2F4K). En la figura A se presenta la trayectoria sin reducir, mientras que en la figura B se presenta los resultados de la primera reducción ó reducción local después del agrupamiento rápido, y en la figura C se observa la reducción final después de aplicar el agrupamiento global a los resultados de la reducción local.

En la primera parte del algoritmo, la reducción local es de alrededor del 30 % (de 100000 a 70000 conformaciones) y se observa que se conservan la mayoría de eventos principales aunque la amplitud de los mismos no es tan evidente. Sin embargo, en la segunda parte del algoritmo, la reducción global ya es alrededor del 90 % (de 100000 a 10000 conformaciones), los principales eventos se siguen conservando, y la amplitud de los mismos se mejora considerablemente respecto al número de conformaciones de la nueva trayectoria.

A. Trayectoria sin reducir (Original)



B. Reducción Parcial (Local)



C. Reducción Final (Global)

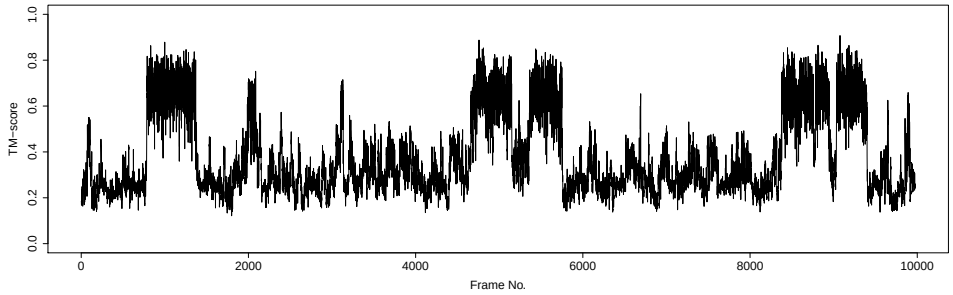
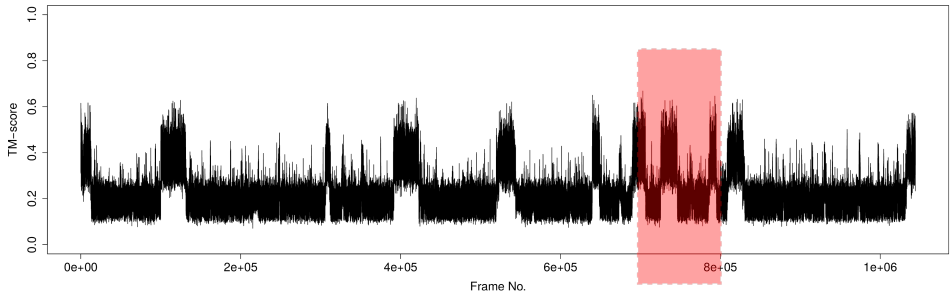


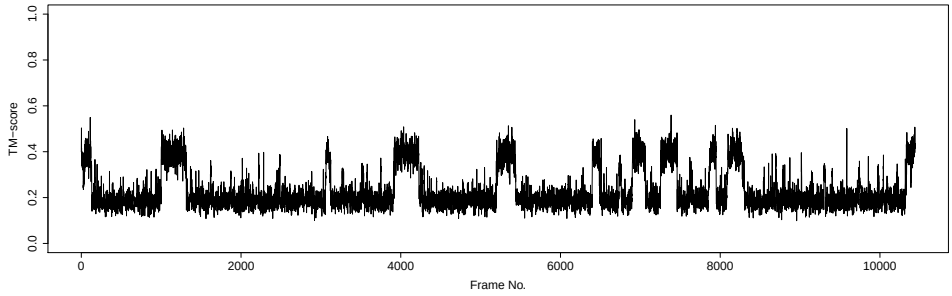
Figura 2: Reducción de una trayectoria larga de plegamiento.

5.2. Reducción sobre una trayectoría completa

Ahora mostramos en la Figura 3 la reducción realizada sobre la trayectoria completa de la proteína TRP-Cage (PDB 2JOF) con más de 1 millón de conformaciones, donde la trayectoría se reduce en un 99 % (de 1044004 a 10000 conformaciones). Observamos en general que las oscilaciones en el plegamiento mostradas en la trayectoria original (figura A), se pueden observar claramente en la reducción (figura B).



A. Trayectoria sin reducir (Original)



C. Reducción Final (Global)

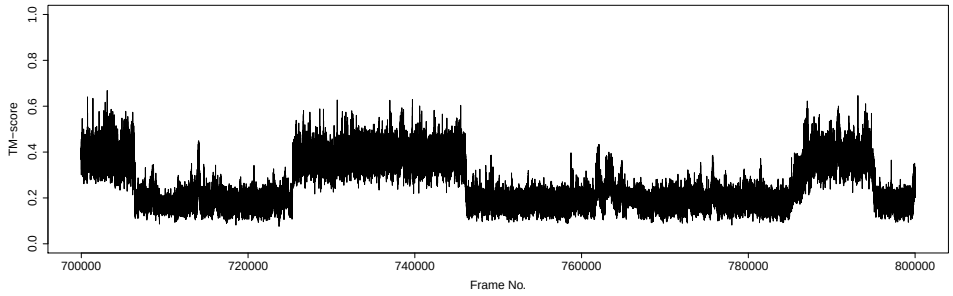
Figura 3: Reducción de una trayectoria larga de plegamiento.

5.2.1. Reducción dada por Gromacs

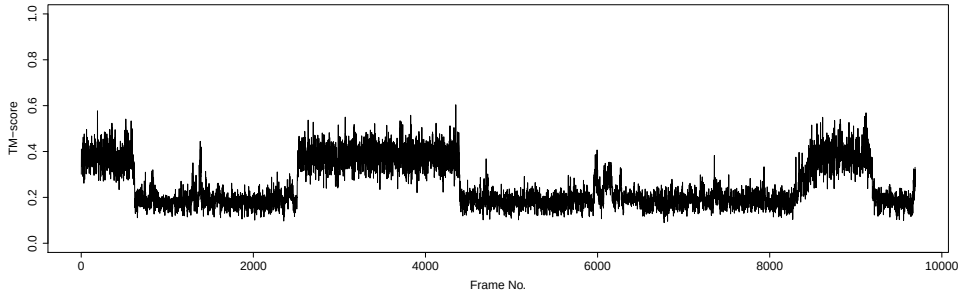
Los resultados de nuestro algoritmo de reducción los comparamos con los resultados del agrupamiento producidos por la herramienta Gromacs, la cual es muy usada en ánalisis de trayectorias de plegamiento de proteínas. Especificificamente utilizamos el algoritmo de agrupamiento *gromos* [XXX] que crea grupos representativos basados en la métrica RMSD entre las conformaciones de la trajectoria. Gromos forma grupos primero encontrando estructuras de vecinos más cercanos de acuerdo a un umbral *(cutoff)* previamente definido, y luego seleccionando la estructura de mayor número de conformaciones como primer grupo. Después, elimina este grupo junto con sus conformaciones y repite el proceso con las restantes. Al final, el algoritmo retorna una lista con el centroide de cada grupo.

Para esta comparación seleccionamos de la anterior trayectoría (proteína 2JOF, Figura 3.A) un segmento de 100000 conformaciones, desde el paso 700000 al paso 800000, que presenta varios oscilaciones visibles gráficamente (segmento marcado en la caja roja). A este segmento le realizamos la reducción usando dos algoritmos: nuestro algoritmo y el algoritmo de Gromacs. Los resultados se presentan en la figura 4: el segmento sin reducir se presenta en la figura 4A, la reducción dada por nuestro algoritmo en la figura 4B, y la reducción dada por Gromacs en la figura 4C. Se observa claramente la diferencia de nuestros

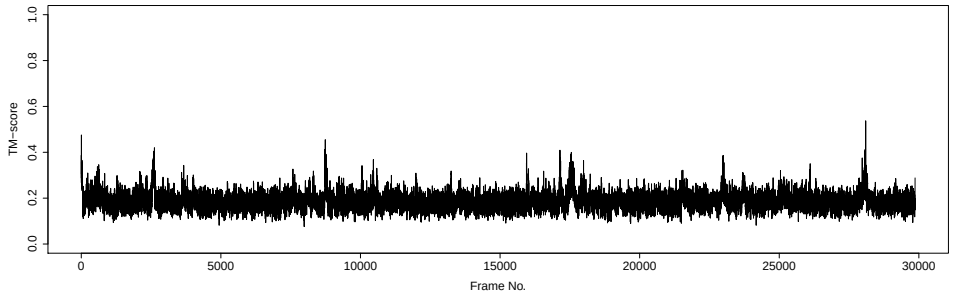
resultados frente a los de Gromacs, en nuestra reducción se conservan los eventos y su amplitud, lo que no sucede en la reducción dada por Gromacs y eso que nuestra reducción es el del 90 % (de 100000 a 10000) mientra que la de Gromacs es de solo el 70 % (de 100000 a 30000)



A. Segmento sin reducir de 100000 conformaciones



B. Segmento reducido por nuestro algoritmo de 10000 conformaciones.



C. Segmento reducido por Gromacs de 30000 conformaciones.

Figura 4: Reducción de una trayectoria larga de plegamiento.

5.2.2. Reducción sobre trayectorias cortas

En la Figura 5 mostramos las reducciónes de dos trayectorias cortas correspondientes a las proteínas 1FCA1 y 2YCC (ver sección 4.3). En la parte superior está la trayectoria original completa; en la parte intermedia la trayectoria después de la reducción local; y en la parte inferior la trayectoria final después de

la reducción global. Las reducciónes logradas son del orden de más del 76 % para la proteína 1FCA1 (de 239 a 57 conformaciones) y más del 90 % para la proteína 2YCC (de 268 a 26 conformaciones). Observamos que los eventos principales en ambas trayectorias se conservan claramente (recuadros rojos en las trayectorias original y final) lo que prueba visualmente que nuestro algoritmo realiza reducciónes que reflejan la dinámica de la trayectoria. Además, destacamos que en la primera reducción, la local (figura intermedia), los eventos principales tienden a desplazarse frente a los originales (recuadros azules), lo cual se logra después corregir en la reducción final. Esto se debe a que la reducción local por ser rápida incluye conformaciones tanto de eventos principales como de eventos secundarios, mientras que la global se enfoca en dejar solo los eventos principales y por lo tanto el desplazamiento se reduce.

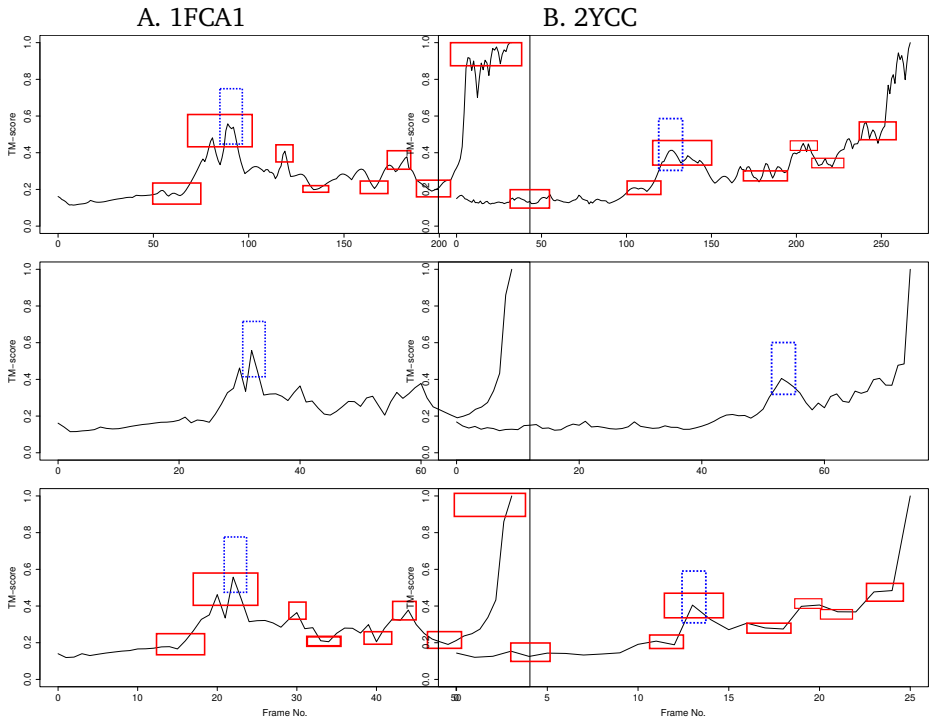


Figura 5: **Reducción de las trayectorias cortas de plegamiento para las proteínas 1FCA1 y 2YCC.** En recuadros rojos se resaltan los eventos principales que se conservan tanto en la trayectoria original como en la final. Los recuadros rojos muestran como algunos eventos principales se desplazan en la reducción local, pero logran ajustarse al final en la reducción global. Para la proteína 1FCA1 la reducción se realizó con los parámetros de 40 bins, un umbral de TMscore de 0.5 y un K de 10. Mientra que para la proteína 2YCC se usaron 50 bins, un TMscore de 0.5 y un K de 5

5.3. Evaluación de Desempeño

5.3.1. Procesamiento paralelo

La estructura de nuestro algoritmo rápido de reducción es altamente paralelizable y por lo tanto su desempeño mejora bastante a medida que utiliza más de un procesador. Para ver este desempeño en la figura 6 mostramos los resultados de reducir un segmento de una trayectoria de 100000 conformaciones variando el número de procesadores desde 1 hasta 40 procesadores. En la gráfica puede observarse claramente la disminución de tiempo cuando de un procesador pasa a ajecutarse en paralelo con 2 y 5 procesadores. La ejecución se reduce de más de 16 minutos a la mitad del tiempo con dos procesadores y a casi menos de 1 minuto con 40 procesadores.

Nro. Procesadores	Tiempo (Segundos)
1	977
2	491
5	197
10	166
20	73
30	70
40	59

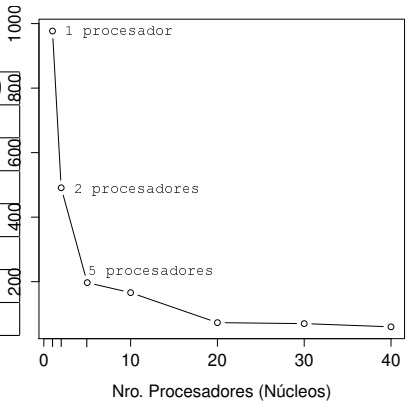


Figura 6: Tiempos de ejecución algoritmo de agrupamiento rápido.

De acuerdo a la estructura del algortimo, primero se divide la trayectoria en multipĺes *bins,* para después realizar sobre cada uno de ellos una reducción local, y sobre estos resultados realizar una reducción global. La división en *bins* es la que permite paralelizar el algoritmo ya que cada *bin* se toma como un trozo de trayectoria independiente de los demás y por lo tanto se puede envíar cada uno de ellos a ejecutarse en un procesador distinto.

5.3.2. Procesamiento por tamaño de los *bins*

El tiempo total del algoritmo se incrementa significativamente (de 58 a 500 segundos) a medida que se selecciona un *bin* de mayor tamaño, como se aprecia en la figura 7 (línea continua de color negro). Este tiempo se reparte en las tres fases del algoritmo: particionamiento, reducción local, y reducción global. De estas fases, la de reducción global es la que mas tiempo consume, proporcional al tiempo total (línea discontinua color rojo, figura 7). Mientras que la reducción local gasta un tiempo casi constante (alrededor de 12 segundos, línea de puntos

color azul, figura 7), y así mismo la fase de particionamiento (alrededor de 1 segundo, no mostrado en la figura 7).

La parte que más consume cómputo en el algoritmo es la comparación entre pares de proteínas que se realiza con las conformaciones de cada *bin* tanto en la reducción local como en la global. Sin embargo, mientras que la reducción local solo realiza algunas comparaciones debido al agrupamiento selectivo propuesto por Hobohm et al. [5] (Sección 2.3); la reducción global realiza todas las posibles comparaciones debido al agrupamiento por k-medoides PAM (partitions around medoids)[7], el cual cálcula la matriz de distancias de las conformaciones de cada *bin*.

Tamaño *Bin*	**Tiempo (Segundos)**			
(Nro conformaciones)	**Total**	**Part.**	**Local**	**Global**
100	58	1	12	45
200	104	1	12	91
400	207	1	12	194
600	319	1	12	306
800	391	1	12	378
1000	512	1	15	496

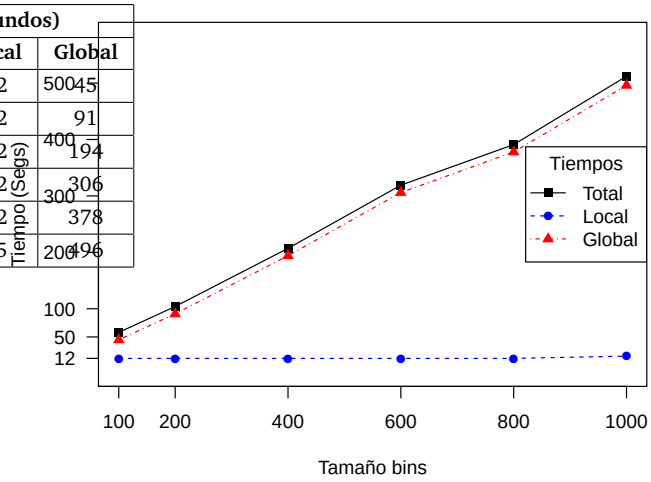


Figura 7: Tiempos de ejecución por tamaño de bins. El tiempo de ejecución se incrementa significativamente a medida que se aumenta el tamaño de los *bins*. Con un *bin* de 100 conformaciones el algoritmo toma alrededor de 1 minuto (59 segundos), sin embargo con un *bin* de 1000 conformaciones el algoritmo toma más de 8 minutos (496 segundos).

6. Implementación

Casi todo el algoritmo está implementado en el lenguaje R excepto la comparación entre pares de proteínas, que es la parte que más veces se ejecuta y que está implementada en el lenguaje Fortran tomando como base el programa TM-score de Zhang&Skolnick [**?**]. Estas comparaciones se realizan tanto en la fase de reducción local como en la global.

7. Conclusiones

En este trabajo presentamos un algoritmo de reducción de trayectorias que visualmente produce reducciónes que logran preservar la dinámica de la trayec-

toria original en cuanto a los eventos principales y la relación de tiempo en la que estos ocurren. El algoritmo tiene cuatro fases: particionamiento, reducción local, y reducción global.

Nuestro algoritmo es altamente configurable, se puede escoger el número de conformaciones de estructuras de proteínas por partición, el umbral de comparación entre dos conformaciones, y el número K para seleccionar las más representativas por partición. Además, el enfoque de particiones que tiene el algoritmo lo vuelve altamente paralelizable ya que cada reducción (local y global) se aplica de forma independiente, tanto local como, sobre cada una de ellas.

Usamos la métrica de TM-score en vez del RMSD para comparar las estructuras de proteínas. Aunque tradicionalemente se ha usado el RMSD, se conoce muy bien que esta métrica es muy sensible a pequeñas diferencias (grupos de átomos) entre las estructuras. Esas pequeñas diferencias dan como resultado grandes valores de RMSD que sugieren que las estructuras comparadas son muy diferentes. El TM-score es una métrica más robusta que el RMSD y produce mejores resultados a la hora de comparar estructuras de conformaciones muy cercanas, que es exactamente lo que se tiene cuando se comparan estructuras de conformaciones consecutivas en una línea de tiempo.

La implementación del algoritmo se realizó en el lenguaje R y Fortran para las librerías de agrupamiento y la fácil paralelización de tareas. En R están implementados los tres módulos: particionamiento, clustering local, y clustering global, mientras que en Fortran está implementada la rutina de evaluación del TM-score, que es la que más se llama tanto en el agrupamiento rápido de la reducción local, como en el agrupamiento detallado de la reducción local.

Referencias

[1] Payel Das, Mark Moll, and H Stamati. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proceedings of the . . .* , 103(26), 2006.

[2] Mojie Duan, Jue Fan, Minghai Li, Li Han, and Shuanghong Huo. Evaluation of Dimensionality-reduction Methods from Peptide Folding-unfolding Simulations. *Journal of chemical theory and computation*, 9(5):2490–2497, may 2013.

[3] Robert C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 2010.

[4] Daniel L Ensign, Peter M Kasson, and Vijay S Pande. Heterogeneity even at the speed limit of folding : Large-scale molecular dynamics study of a fast-folding variant of the villin headpiece. *Journal of molecular biology*, 374(3):806–816, 2007.

[5] Uwe Hobohm, Michael Scharf, Reinhard Schneider, and Chris Sander. Selection of representative protein data sets. *Protein Science*, 1(3):409–417, mar 1992.

[6] Fabien P E Huard, Charlotte M Deane, and Graham Wood. Modelling sequential protein folding under kinetic control.

[7] Leon Kaufman and Peter Rousseeuw. *Finding Groups in Data*. Wiley-Interscience; New York, 1990.

[8] W. Li, L. Jaroszewski, and A. Godzik. Tolerating some redundancy significantly speeds up clustering of large protein databases. *Bioinformatics*, 18(1):77–82, 2002.

[9] Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, oct 2011.

[10] A. Marsden. M. Lougher, M. Lücken, T Machon, M. Malcomson. Computational Modelling of Protein Folding. Technical report.

[11] Hai Nguyen, James Maier, He Huang, Victoria Perrone, and Carlos Simmerling. Folding simulations for proteins with diverse topologies are accessible in days with a physics-based force field and implicit solvent. *Journal of the American Chemical Society*, 136(40):13959–13962, oct 2014.

[12] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, mar 2009.

[13] Jun-hui Peng, Wei Wang, Ye-qing Yu, Han-lin Gu, and Xuhui Huang. Clustering algorithms to analyze molecular dynamics simulation trajectories for complex chemical and biological systems. *Chinese Journal of Chemical Physics*, 31(4):404–420, aug 2018.

[14] Aruna Rajan, Peter L. Freddolino, and Klaus Schulten. Going beyond clustering in MD trajectory analysis: An application to villin headpiece folding. *PLoS ONE*, 5(4):e9890, jan 2010.

[15] David E Shaw, Jack C. Chao, Michael P. Eastwood, Joseph Gagliardo, J. P. Grossman, C. Richard Ho, Douglas J. Lerardi, István Kolossváry, John L. Klepeis, Timothy Layman, Christine McLeavey, Martin M. Deneroff, Mark A. Moraes, Rolf Mueller, Edward C. Priest, Yibing Shan, Jochen Spengler, Michael Theobald, Brian Towles, Stanley C. Wang, Ron O. Dror, Jeffrey S. Kuskin, Richard H. Larson, John K. Salmon, Cliff Young, Brannon Batson, and Kevin J. Bowers. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM*, 51(7):91, 2008.

[16] Guang Song and Nancy M Amato. Using Motion Planning to Study Protein Folding Pathways. *Journal of Computational Biology*, pages 287–296, 2001.

[17] Hui Yang, Srinivasan Parthasarathy, and Duygu Ucar. A spatio-temporal mining approach towards summarizing and analyzing protein folding trajectories. *Algorithms for Molecular Biology*, 2(1):3, 2007.