RESEARCH

Un algoritmo paralelo para la reducción de trayectorias de plegamiento usando una estrategia rápida de agrupamiento

Luis Garreta[1†], Mauricio Martinez[2] and Pedro A Moreno[1*]

Resumen

Background: La simulación del proceso de plegamiento de proteínas es una de las principales herramientas para estudiar y comprender los mecanismos subyacentes en este proceso. Hoy en día estas simulaciones están llegando a unos tiempos de simulación que hasta hace algunos años eran imposibles de alcanzar y como consecuencia las trayectorias generadas son muy grandes. Analizar este tipo de trayectorias trae complicaciones debido a su tamaño y por lo tanto se necesita crear herramientas que logren reducirlas de tal manera que se logren preservar tanto los eventos principales como el orden temporal en el que ellos ocurren.

Results: Introducimos aquí un algoritmo de reducción para trayectorias grandes de plegamiento de proteínas que se caracteriza por dividir la trayectoria en segmentos y mediante una estrategia rápida de agrupamiento tomar los eventos más disimilares para luego seleccionar entre ellos a los k eventos más representativos. El algoritmo aprovecha el orden temporal implícito en la trayectoria para realizar en cada segmento comparaciones locales, entre eventos vecinos, y así evitar realizar una comparación de todos contra todos que es muy costosa computacionalmente.

Conclusions: El esquema anterior permite que el algoritmo sea muy rápido y que los eventos seleccionados conserven su orden temporal dentro de la trayectoria. Además, el particionamiento en segmentos permite al algoritmo realizar la reducción por cada segmento de forma independiente y por lo tanto realizarse las reducciones de forma paralela lo que lo vuelve aún más rápido cuando se ejecuta en máquinas con procesadores de múltiples cores, como los PCs que se consiguen en el mercado hoy en día. Para mostrar la efectividad del algoritmo propuesto realizamos reducciones sobre tres conjuntos de trajectorias disponibles públicamente: las del supercomputador Anton, las del proyecto folding@home, y las del servidor de desplegamiento de Parasol.

Keywords: Protein folding simulations; Protein structure comparison; Protein structure clustering

Background

En este artículo presentamos un algoritmo para reducir trayectorias de plegamiento de proteínas el cual obtiene rápidamente conformaciones representativas conservando tanto su estructura tridimensional (3D) como su orden temporal, y que además es altamente paralelizable. Las proteínas desempeñan funciones fundamentales en todos los seres vivos, pero para ser funcionales deben a partir de su cadena de aminoácidos (AA) plegarse hasta alcanzar una forma 3D única o estado nativo, lo que se conoce como el proceso de plegamiento de las proteínas. Entender los mecanismos y reglas de este proceso ha sido uno de los objetivos más perseguidos dentro de la biología y una herramienta teórica importante para estudiarlo son las trayectorias de plegamiento, que describen la evolución del plegamiento de una proteína mediante la secuencia de estados que esta atraviesa en función del tiempo durante su proceso de plegamiento (Figura).

Estas trayectorias son simuladas principalmente por el método de dinámica molecular (DM), el cual por su costo computacional está limitado a proteínas pequeñas (< 100 AA) y a tiempos muy cortos (pico o microsegundos). Sin embargo, nuevos avances tecnológicos evidencian un progreso notable en estas simulaciones. Recientemente en el 2016 se puso en operación la supercomputadora Anton-2 [1], diez veces más rápida

*Correspondence: pedro.moreno@correounivalle.edu.co
[1]Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Santiago de Cali, Colombia
Full list of author information is available at the end of the article
[†]Equal contributor

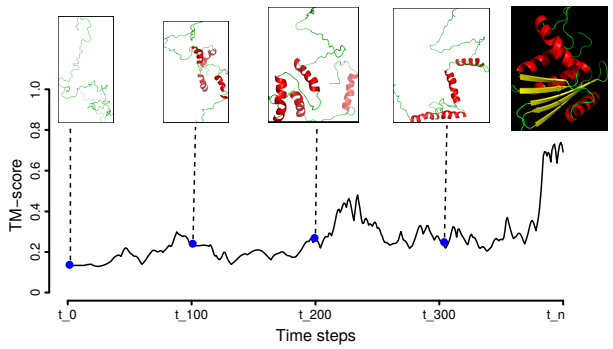

Figura 1 Trayectoria de plegamiento de una proteína. La evolución del plegamiento se mide a cada paso comparando la estructura actual (imágenes fondo blanco) frente a la estructura nativa (imágen fondo negro) mediante la métrica TM-Score [?]. Eje horizontal: tiempo de simulacion desde t_0 (inicial) hasta t_n (final). Eje vertical: valor del TM-Score, desde 0 (diferentes) hasta 1 (similares).

que su predecesora Anton-1 [2], diseñada especialmente para el plegamiento de proteínas y de la cual ya se reportó en el 2011 las simulaciones completas de 12 proteínas [3], varias en el orden de los milisegundos. Como una alternativa más económica a estas supercomputadoras, en el 2014 se usó unidades de procesamiento gráfico (GPUs) y se reportó las simulaciones de 17 proteínas en el orden de los microsegundos [4]. Y años antes, en el 2007 utilizando computadoras de escritorio unidas a través de computación distribuida en el proyecto folding@home se realizaron varias simulaciones en el orden de los microsegundos del plegamiento de la proteína villin headpiece [5].

Estos avances muestran un crecimiento notable en estas simulaciones con tiempos en el orden de los micro y milisegundos, y con trayectorias de millones de conformaciones. Muchas de estas trayectorias ya se están colocando a disposición pública, pero debido al gran número de conformaciones, su procesamiento y análisis en computadoras convencionales es muy costoso en tiempo computacional. Por lo tanto se necesiten nuevos algoritmos capaces de reducir estas trayectorias de una forma rápida, aprovechando eficientemente los recursos de este tipo de máquinas, y buscando conservar la mayor información posible tanto a nivel de representación como a nivel de orden temporal.

Para realizar estas reducciones se han usado dos enfoques: la reducción de la dimensionalidad [6] y el agrupamiento [7]. En el primer enfoque se transforma una conformación a un conjunto reducido de variables que la representan lo mejor posible. Para esto se han usado tanto técnicas lineales como no-lineales (e.g. análisis de componentes principales (PCA), escalamiento multi-dimensional [8], Isomap [9], diffusion maps [10]). Sin embargo, aunque se logra la reducción de las conformaciones, se pierde su representatividad como estructuras 3D (Figura 2.A). Además, estas técnicas consumen mucho tiempo cuando las trayectorias son muy grandes, ya que tienen que transformar todas sus conformaciones.

En el segundo enfoque, agrupamiento, se asignan las conformaciones a grupos que comparten las mismas características (e.g. similaridad con la estructura nativa) y se toma de cada grupo ya sea un representante promedio ó sus características generales. Aquí se se han usado tanto agrupamientos particionales como jerárquicos (e.g. k-means [11], linkage [12]). Sin embargo, los grupos pierden su orden temporal ya que pueden abarcar conformaciones que ocurren a tiempos muy distintos (Figura 2.B). Además, se tienen que comparar todos los pares de conformaciones, lo cual es una operación costosa y más aún cuando las trayectorias son muy grandes.

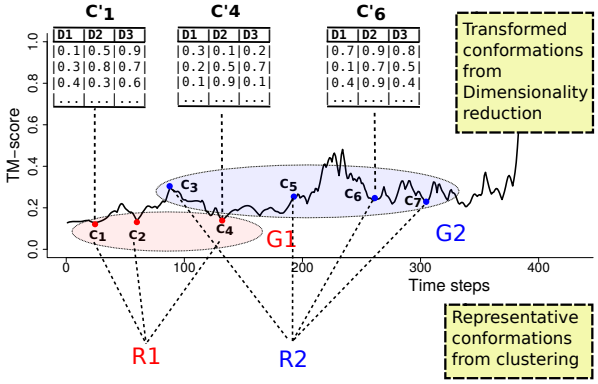

Figura 2 Perdida de información en las reducciones. La reducción de la dimensionalidad (Parte de arriba) transforma las conformaciones a un nuevo conjunto de valores y pierden su información estructural. Mientras que en el agrupamiento (parte de abajo), las representantes R1 y R2 pierden su orden temporal ya que los grupos G1 y G2 contienen conformaciones C3 y C4 que se traslapan en el tiempo.

Nuestro algoritmo reduce una trayectoria de plegamiento en tres fases (Figura 3): primero divide la trayectoria en pequeñas subtrayectorias que luego las reduce de manera individual, independiente y paralela. Segundo toma cada subtrayectoria y extrae de forma muy rápida sus conformaciones características y elimina las redundantes utilizando la estrategia de agrupamiento rápido de Hobohm and Sander (1992). Y tercero toma la conformaciones características y selecciona las más representativas mediante una estrategia tipo k-medoides [13], la cual al trabajar sobre pocas conformaciones, mejora sustancialmente su desempeño. Al final, los resultados de cada reducción se unen para obtener la reducción total de la trayectoria.

Además, a diferencia de otras técnicas de reducción de trayectorias, nuestro algoritmo tiene la ventaja de no cambiar la representación de las conformaciones como lo hacen las técnicas de reducción de dimensionalidad, ni de perder el orden temporal como lo hacen las técnicas de agrupamiento. El resultado de nuestro algoritmo es un conjunto de conformaciones representativas de la trayectoria que siguen conservando tanto su estructura 3D como su orden temporal.

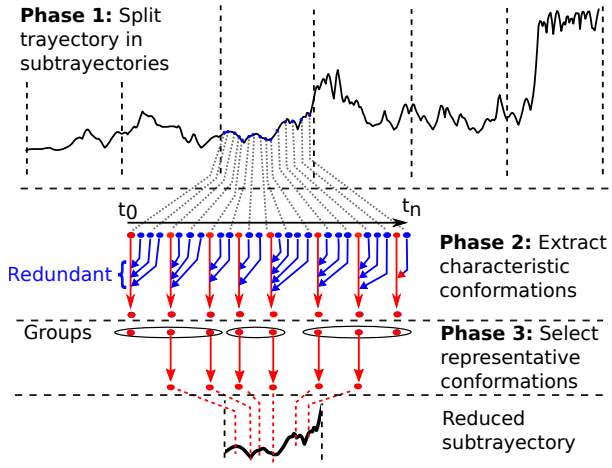

Figura 3 Algoritmo de reducción. La trayectoria se divide en subtrayectorias (líneas discontinuas) que se reducen individualmente. Mediante una estrategia de agrupamiento rápido [14] se seleccionan las conformaciones características (puntos rojos) y se ignoran las redundantes (puntos azules). Después, mediante un agrupamiento tipo k-medoides [13] se extraen las representativas que forman la subtrayectoria reducida.

La implementación del algoritmo está en lenguaje R excepto la comparación entre conformaciones, que usa la métrica TM-Score para comparar pares de estructuras de proteínas [15], y que por ser la parte que más se repite está implementada en lenguaje Fortran.

Métodos

Conjuntos de datos de proteínas

Para evaluar nuestro algoritmo tanto en las reducciones que realiza como en su desempeño tomamos las trayectorias de tres proteínas de diferentes proyectos: la trayectoria de la proteína Trp-cage, simulada con dinámica molecular en la supercomputadora Anton por D.E Shaw Research [3], tiempo de simulación de 208 μs, y pasos de tiempo de 200 ps. La trayectoria de la proteína villin-headpiece, simulada con dinámica molecular utilizando computación distribuida en el proyecto folding@home [16], tiempo de simulación 8 μs, y pasos de tiempo de 50 ps; y la trayectoria de la proteína Ribonuclease H, simulada con el método Probabilistic Roadmap Method [17] con 429 pasos de simulación, que corresponden a eventos de desplegamiento y no a pasos de tiempo.

Comparaciones con nMDS, PCA, y clustering

Comparamos los resultados de nuestro algoritmo frente a los resultados de tres métodos comúnmente utilizados en reducción de trayectorias de plegamiento [11]: escalamiento multidimensional no-métrico (nMDS), análisis de componentes principales (PCA), y agrupamientos (Figura **??**). Para la reducción con nMDS, calculamos la matriz de *disimilaridades* entre las conformaciones mediante la métrica TM-score [15], con esta matriz calculamos los nuevos puntos para un espacio geométrico de 2D mediante la función *monoMDS* del paquete *vegan* del sistema R [18], y los desplegamos sobre un plano 2D. Para la reducción con PCA caracterizamos cada conformación con las coordenadas XYZ de sus átomos, calculamos los componentes principales mediante la función *pca.xyz* del paquete *Bio3D* del sistema R [19], y seleccionamos los dos primeros componentes que explican la mayor varianza. Y para el agrupamiento, caracterizamos cada conformación con sus dos primeros componentes principales y realizamos un agrupamiento jerárquico con el método *complete linkage* de la función *hclust* del paquete *stats* del sistema R [20]. El número de grupos k=7 lo seleccionamos mediante un enfoque de promedios Silhouette al variar k desde 1 hasta 10 utilizando la función *fviz_nbclust* del paquete *factoextra* del sistema R [21].

Implementación

El algoritmo reduce una trayectoria de plegamiento de proteínas en tres fases: particionamiento, selección, y extracción (Figura 3). Cada fase conlleva una estrategia para mejorar la eficiencia del algoritmo cuando las trayectorias de plegamiento son muy grandes.

Fase 1: Particionamiento y Paralelización

Dividimos la trayectoria en subtrayectorias con el objetivo de reducirlas de forma independiente y paralela (líneas verticales punteadas, Figura 3).

Esta estrategia de particionamiento tiene un doble objetivo: primero, reducir localmente cada subtrayectoria y así enfocarnos en sus características particulares, lo que al final resulta en la obtención de las características globales de toda la trayectoria. Y segundo,

que sus reducciones se puedan realizar en paralelo y así mejorar notoriamente la eficiencia del algoritmo a la hora de ejecutarlo en una máquina con tecnología multi-core (Figura 4).

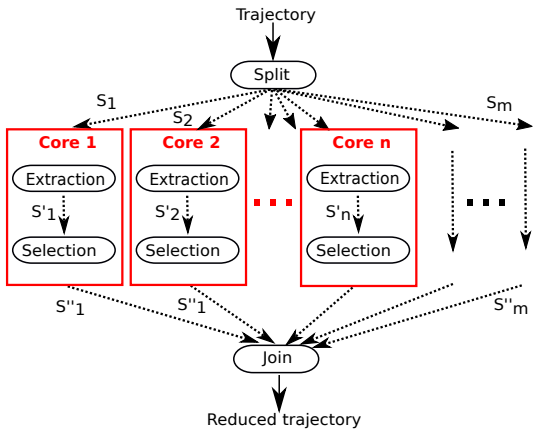

Figura 4 Parallel Reduction. La reducción de cada subtrayectoria S1...Sm se ejecuta en paralelo aprovechando la tecnología multi-core de las máquinas actuales. Los resultados tanto del proceso de selección de características (S'1...S'm) como los del de extracción de representativas (S''1...S''m) son independientes de los de las otras subtrayectorias. Al final los resultados de cada procesamiento se unen para obtener la trayectoria total reducida.

Fase 2: Extracción y Filtración

Esta fase del algoritmo extrae rápidamente de cada subtrayectoria las conformaciones características y filtra las redundantes. Para hacerlo de manera eficiente, modificamos la estrategia de agrupamiento rápido de Hobohm and Sander (1992) para trabajar con estructuras de proteínas y en vez agruparlas busque las más disimilares.

El algoritmo aprovecha el orden temporal implícito en las subtrayectorias para organizar las conformaciones en orden creciente de tiempo de simulación (flecha horizontal negra, Figura 3). Se asigna la primera conformación como la primera representante característica (punto rojo en t0, Figura 3), y se toma la siguiente conformación y se comparan. Si son diferentes, entonces se convierte en una nueva representante (puntos rojos, Figura 3), de lo contrario es redundante y se filtra (puntos azules, Figura 3). Después, se toma la siguiente conformación y se continua el mismo proceso hasta terminar con todas.

Fase 3: Búsqueda y Selección

Esta última fase del algoritmo toma como entrada las conformaciones características de la fase anterior y realiza una búsqueda completa de las conformaciones que más las representen. Hablamos de completa porque la búsqueda implica comparar todas las conformaciones entre si, es decir calcular su matriz de disimilaridades. Por esta razón es que esta búsqueda es factible hacerla ahora y no antes ya que se hace sobre un conjunto mucho menor de conformaciones que el que se tiene al inicio por cada subtrayectoria, .

Para encontrar estas conformaciones representantes calculamos las k conformaciones cuya disimilaridad media a todas las demás integrantes del grupo es mínima, lo que se conoce como *medoides* y el algoritmo que usamos para realizar esto es el particionamiento alrededor de medoides PAM [13].

Esta fase necesita tres datos de entrada: el conjunto de conformaciones características de la fase anterior (C), el umbral mínimo de TM-score para aceptar dos conformaciones como similares (T), y el número deseado de representantes seleccionadas (*K*).

Comparación entre Estructuras de Proteínas

Para la comparación de las estructuras de las conformaciones utilizamos la métrica TM-score (Template Modeling score) [15]. El TM-score es más preciso que otras métricas usadas en comparación de estructuras, como el Root Mean Square-Deviation (RMSD), ya que es más robusta a variaciones locales.

Nuestro algoritmo requiere como uno de sus parámetro de entrada un valor de puntaje mínimo de TM-score para aceptar como similares a dos conformaciones. Este parámetro se usa después tanto en la fase de extracción de características, al comparar las conformaciones para encontrar disimilares y remover redundantes, como en la fase de selección de representativas, al calcular la matriz de distancias de todas las conformaciones.

Para tener una aproximación del rango de valores de este puntaje mínimo, los puntajes del TM-score varian de 0 a 1, donde 1 indica un emparejamiento perfecto. Además, las estadísticas hechas por sus autores [22] muestran que un puntaje < 0.17 indica dos estructuras aleatorias, sin relación de similaridad, y un puntaje > 0.5 indica que las estructuras tienen un grado de similaridad que no está dado por el azar.

Resultados y Discusión

Reducciones de tres trayectorias de plegamiento

Realizamos la reducción de las trayectorias de tres proteínas tomadas de distintos proyectos de simulación: Trp-cage (Supercomputador Anton [3], villin-headpiece (folding@home [16]), y la Ribonuclease H (Folding server [17]) (ver detalles de las simulaciones en la sección de Métodos). Los resultados se muestran en la figura 5 donde se presenta para cada proteína en el recuadro izquierdo sus detalles, y en el derecho

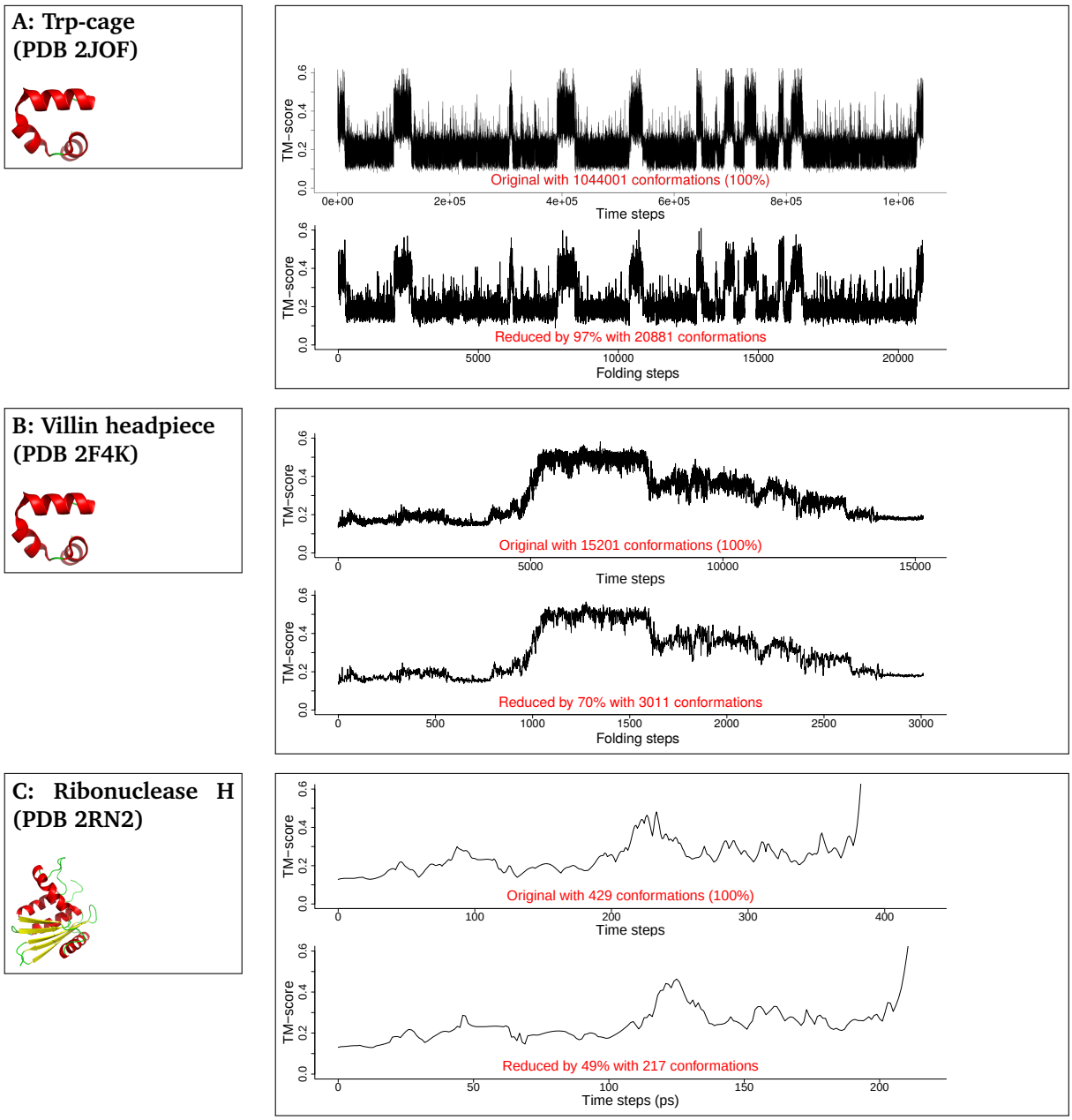

Figura 5 Reducciones de tres trayectorias de plegamiento. Recuadro izquierdo: nombre, código PDB y estructura nativa de la proteína. Recuadro derecho: trayectoria original en la parte de arriba y la trayectoria reducida en la parte inferior. La trayectoria original muestra en el eje horizontal los pasos de tiempo en que ocurren las conformaciones en la simulación, mientras que la trayectoria reducida muestra los pasos de plegamiento o secuencia de eventos que resumen la trayectoria. (A) Trayectoria de la proteína Trp-cage reducida en un 98 %, de 1044001 a 20881 conformaciones. (B) Trayectoriade la proteína villin-headpiece reducida en un 80 %, de 15201 a 3011conformaciones. (C) Trayectoria de la proteína Ribonuclease H reducida en un 49 %, de 429 a 217 conformaciones.

sus dos trayectorias: la original (arriba) y la reducida (abajo).

Como se puede observar de la figura 5, los resultados de las reducciones son conformaciones de la misma trayectoria, las cuales siguen conservando tanto su estructura como su orden temporal. Este resultado es importante ya que estas reducciones, al ser un resumen de la trayectoria original, se pueden usar enteramente como entrada para análisis más complejos que pueden volverse imprácticos cuando tratan con trayectorias muy grandes. Otros técnicas de reducción usadas en análisis de trayectorias o bien transforman las conformaciones en estructuras de menos dimensiones, solo interpretables cuando se observan en conjunto, como el caso de MDS, Isomap, y diffusion maps [23, 10]; o crean grupos de ellas que resaltan alguna similaridad ya sea estructural o energética, sin importar su orden temporal, como en el caso de los agrupamientos [7]. Además, debido a que varias de estas técnicas se basan en el cálculo de las distancias entre pares de conformaciones, el alto costo computacional de realizar esos cálculos para millones o incluso miles de conformaciones, las puede volver imprácticas sino se utilizan trayectorias reducidas como las que produce nuestro algoritmo.

Sin embargo, aunque las conformaciones de las trayectorias reducidas conservan el orden temporal que tienen en la trayectoria original, el tiempo de simulación en que suceden no se conserva explícitamente. Es decir, las reducciones no describen pasos de tiempo sino pasos de plegamiento, que se refieren a la secuencia de eventos destacados que resumen el plegamiento de la proteína y no al tiempo exacto en que estos ocurren. No obstante, para obtener estos tiempos, se puede tomar el nombre o identificador de la conformación de interés en la trayectoria reducida y localizar su tiempo en la trayectoria original.

Comparación frente a otros métodos de reducción

Para las comparaciones utilizamos los datos de la simulación de plegamiento de la proteína villin-headpiece del proyecto folding@home [5]. Tomamos la trayectoria original y calculamos su reducción por los métodos de nMDS y PCA. Luego, calculamos dos reducciones con nuestro algoritmo sobre esta trayectoria y a los datos resultantes le calculamos nuevamente las reducciones por nMDS y PCA. Los resultados se muestran en la figura 6, donde cada fila contiene tres despliegues en 2D: de la trayectoria, del patrón resultante de la reducción por nMDS, y del agrupamiento al proyectar los dos primeros componentes del PCA.

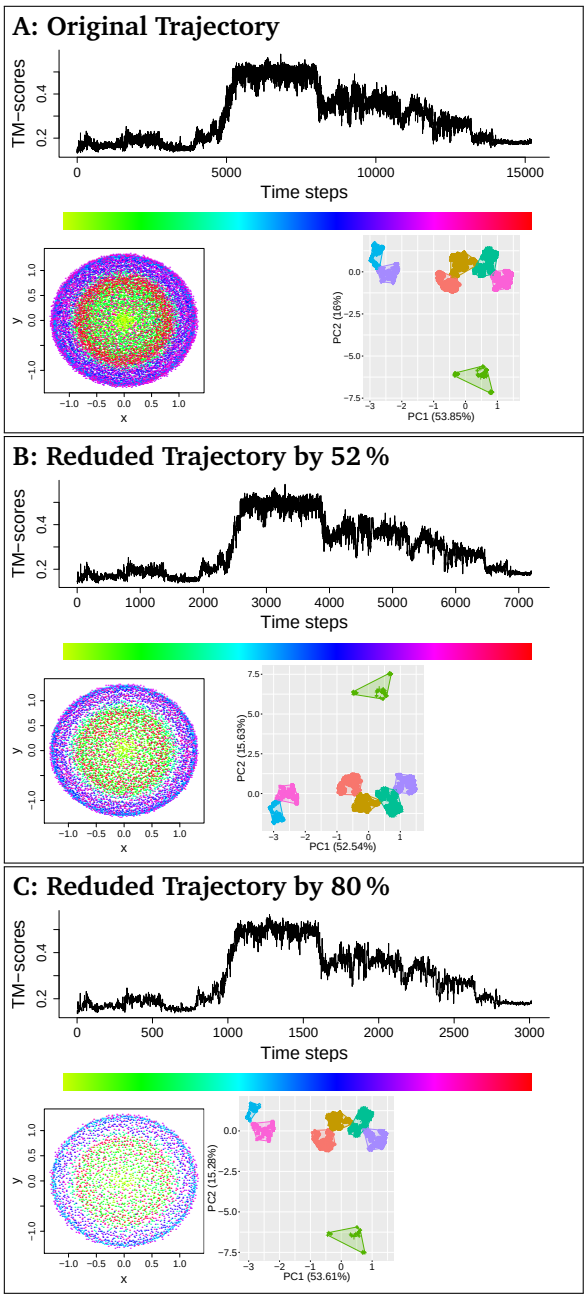

Figura 6 Comparación frente a otros métodos.

Observamos que las reducciones de la trayectoria original producen un despliegue en 2D característico en ambos métodos de reducción: un patrón de círculos de puntos, para el nMDS; y una estructura de 7 grupos, para el agrupamiento por PCA (fila superior, figura 6). Así mismo, este mismo despliegue se repite en gran medida en las dos reducciones calculadas por nuestro algoritmo, la de compresión media del 52 % y la de compresión alta del 80 % (filas central e inferior de la figura 6, respectivamente).

Lo anterior nos indica que nuestras reducciones preservan en gran medida los eventos principales de la trayectoria al observar que tanto las reducciones con nMDS y PCA siguen conservando el mismo patrón y la misma estructura de grupos. Además, nuestro algoritmo presenta ventajas adicionales sobre los otros métodos de reducción. Primero, el cálculo de las reducciones es más eficiente que el de nMDS ya que no necesita la matriz de disimilaridades, que es sumamente costosa de calcular cuando el número de conformaciones es grande. Segundo, la interpretación de los resultados es directa ya que los resultados son conformaciones de la proteína y no transformaciones de los datos, como en el caso del nMDS y PCA, o grupos de conformaciones, como en el caso de los agrupamientos. Y tercero, el orden temporal se conserva ya que el resultado es una nueva trayectoria, a diferencia del agrupamiento en donde los grupos resultantes pueden contener conformaciones que ocurren en tiempos muy distintos.

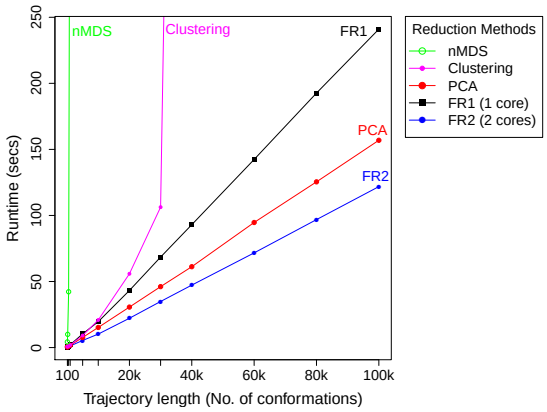

Figura 7 Desempeño del algoritmo frente a otros métodos. Comparación del nuestro algoritmo FR1 con nMDS, PCA, y agrupamiento. PCA y FR1 son los más eficientes, pero si nuestro algoritmo utiliza dos núcleos (FR2), el tiempo se disminuye a la mitad y se vuelve más eficiente que PCA. Por el contrario, nMDS y clustering toman demasiado tiempo, aún con trayectorias pequeñas.

Desempeño del algoritmo

El desempeño de nuestro algoritmo lo evaluamos en dos situaciones: comparándolo frente a otros métodos de reducción (Figura 7) y ejecutándolo en paralelo usando múltiples núcleos de procesamiento (Figura 8). Para esto utilizamos las 100K primeras conformaciones de la trayectoria de la proteína Trp-cage (ver Métodos). Para la primera evaluación ejecutamos los métodos con diferentes tamaños de subtrayectorias, desde 100 hasta 100K conformaciones, y en la segunda evaluación ejecutamos nuestro algoritmo con diferente número de núcleos de procesamiento.

En la comparación con otros métodos de reducción, la figura 7 muestra que PCA es el más eficiente seguido de nuestro algoritmo FastReduction cuando se ejecuta con un solo núcleo de procesamiento. Sin embargo, si lo ejecutamos en paralelo con 2 núcleos, este se vuelve más eficiente que PCA. Por el contrario, nMDS y clustering se vuelven imprácticos con subtrayectorias medianamente largas. Ahora, si ejecutamos nuestro algoritmo en paralelo con 2 cores (FR2, línea azul), este se vuelve más eficiente que PCA.

Este comportamiento lo podemos ver más claramente en la figura 8, donde se muestran los tiempos y la aceleración que alcanza el algoritmo a medida que se ejecuta con más núcleos. Cada que duplicamos el número de núcleos, el tiempo de ejecución se disminuye casi a la mitad, hasta los 8 núcleos esta relación se conserva y luego la disminución es menor hasta volverse mínima pasados los 30 núcleos.

Todo lo anterior nos muestra que el algoritmo presenta un buen desempeño comparado con los otros métodos, y que este mejora más cuando aprovecha su

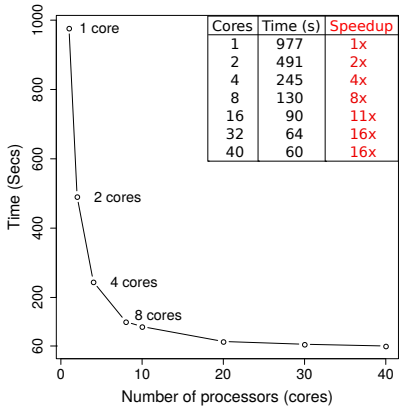

Figura 8 Desempeño del algoritmo cuando se ejecuta en paralelo con múltiples núcleos. Al duplicar el número de núcleos de procesamiento el tiempo se disminuye casi a la mitad y por lo tanto la aceleración crece casi de forma lineal, por lo menos hasta los 8 núcleos (8x). Después de esto, la aceleración sigue siendo apreciable hasta casi después de los 30 núcleos.

paralelismo y se ejecuta con más de un núcleo. Como consecuencia, la aceleración de nuestro algoritmo escala de forma lineal con el número de núcleos que utiliza, por lo menos hasta 8x, es decir, la velocidad de ejecución cuando utiliza 8 núcleos es 8 veces más que cuando utiliza solo uno. Además, con 32 núcleos todavía se logra una aceleración de 16x, después de lo cual esta se mantiene sin mayor aumento (ver recuadro figura **??**.B). Ahora, considerando que la tecnología multi-core ya está presente en muchas de los computadores de hoy día, el algoritmo tiene la capacidad de aprovechar esta tecnología para reducir tra-

yectorias largas en tiempos cortos, cercanos e incluso mejores que los que toman algunos de los métodos cómunes usados en reducción de trayectorias de plegamiento.

Conclusiones

Las simulaciones de plegamiento de proteínas están avanzando significativamente y cada vez se realizan más para nuevas proteínas, con tiempos de duración más largos, y llevadas a cabo sobre diversas tecnologías. Como consecuencia, las trayectorias generadas por estas simulaciones cada vez son más extensas, del orden de millones de conformaciones, lo cual hace difícil su procesamiento y análisis. Para simplificarlas se han planteado diferentes técnicas que más bien son técnicas de análisis que transforman las conformaciones o crean grupos de ellas y sus resultados tienen sentido solo cuando se observan en conjunto.

Aquí, nosotros hemos planteado un algoritmo para simplificar trayectorias de plegamiento que divide la trayectoria en segmentos y extrae de ellos sus eventos principales o conformaciones destacadas en dos fases: primero extrae rápidamente las conformaciones disimilares y luego una selecciona de estas a las más representativas. El algoritmo se caracteriza por ser rápido y fácilmente paralelizable, y por lo tanto ejecutable en máquinas ordinarias con múltiples cores, disponibles ya en la mayoría de laboratorios de investigación.

De acuerdo a los resultados, el algoritmo produce simplificaciones de las trayectorias originales con una compresión alta y con los eventos principales visualmente conservados. Así mismo, estos resultados conservan en gran medida los patrones y la estructura que producen las reducciones hechas por otras técnicas de reducción y análisis de trayectorias. En cuanto al desempeño del algoritmo, este se aproxima al mostrado por algunas de las técnicas más eficientes y mejora mucho cuando se ejecuta en paralelo.

Sin embargo, las simplificaciones producidas por el algoritmo están limitadas a crear resúmenes de las trayectorias sin realizarles ningún tipo de análisis, como lo hacen otras técnicas. Por esta misma razón, estas trayectorias resumidas pueden servir de entrada tanto a técnicas de análisis complejas como a otras técnicas de reducción que empiezan a tener problemas a medida que las trayectorias se vuelven más grandes.

Author details
[1]Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Santiago de Cali, Colombia. [2]The European Bioinformatics Institute (EMBL-EBI), Hinxton, Cambridgeshire, UK.

Referencias

1. Shaw, D.E., Grossman, J.P., Bank, J.A., Batson, B., Butts, J.A., Chao, J.C., Deneroff, M.M., Dror, R.O., Even, A., Fenton, C.H., Forte, A., Gagliardo, J., Gill, G., Greskamp, B., Ho, C.R., Ierardi, D.J., Iserovich, L., Kuskin, J.S., Larson, R.H., Layman, T., Lee, L.-S., Lerer, A.K., Li, C., Killebrew, D., Mackenzie, K.M., Mok, S.Y.-H., Moraes, M.A., Mueller, R., Nociolo, L.J., Peticolas, J.L., Quan, T., Ramot, D., Salmon, J.K., Scarpazza, D.P., Schafer, U.B., Siddique, N., Snyder, C.W., Spengler, J., Tang, P.T.P., Theobald, M., Toma, H., Towles, B., Vitale, B., Wang, S.C., Young, C.: Anton 2: Raising the Bar for Performance and Programmability in a Special-Purpose Molecular Dynamics Supercomputer. In: Shaw2014 (ed.) SC14: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 41–53. IEEE, Los Alamitos, CA, USA (2014). doi:10.1109/SC.2014.9. http://ieeexplore.ieee.org/document/7012191/
2. Shaw, D.E., Chao, J.C., Eastwood, M.P., Gagliardo, J., Grossman, J.P., Ho, C.R., Lerardi, D.J., Kolossváry, I., Klepeis, J.L., Layman, T., McLeavey, C., Deneroff, M.M., Moraes, M.A., Mueller, R., Priest, E.C., Shan, Y., Spengler, J., Theobald, M., Towles, B., Wang, S.C., Dror, R.O., Kuskin, J.S., Larson, R.H., Salmon, J.K., Young, C., Batson, B., Bowers, K.J.: Anton, a special-purpose machine for molecular dynamics simulation. Communications of the ACM 51(7), 91 (2008). doi:10.1145/1364782.1364802
3. Lindorff-Larsen, K., Piana, S., Dror, R.O., Shaw, D.E.: How fast-folding proteins fold. Science 334(6055), 517–520 (2011). doi:10.1126/science.1208351. arXiv:1011.1669v3
4. Nguyen, H., Maier, J., Huang, H., Perrone, V., Simmerling, C.: 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 (2014). doi:10.1021/ja5032776
5. Larson, S.M., Snow, C.D., Shirts, M., Pande, V.S.: Folding@Home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology (2009). 0901.0866
6. Duan, M., Fan, J., Li, M., Han, L., Huo, S.: Evaluation of Dimensionality-reduction Methods from Peptide Folding-unfolding Simulations. Journal of chemical theory and computation 9(5), 2490–2497 (2013). doi:10.1021/ct400052y
7. Peng, J.-h., Wang, W., Yu, Y.-q., Gu, H.-l., Huang, X.: Clustering algorithms to analyze molecular dynamics simulation trajectories for complex chemical and biological systems. Chinese Journal of Chemical Physics 31(4), 404–420 (2018). doi:10.1063/1674-0068/31/cjcp1806147
8. Rajan, A., Freddolino, P.L., Schulten, K.: Going beyond clustering in MD trajectory analysis: An application to villin headpiece folding. PLoS ONE 5(4), 9890 (2010). doi:10.1371/journal.pone.0009890
9. Das, P., Moll, M., Stamati, H.: Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. Proceedings of the ... 103(26) (2006)
10. Kim, S.B., Dsilva, C.J., Kevrekidis, I.G., Debenedetti, P.G.: Systematic characterization of protein folding pathways using diffusion maps: Application to Trp-cage miniprotein. The Journal of Chemical Physics 142(8), 85101 (2015). doi:10.1063/1.4913322
11. Doerr, S., Ariz-Extreme, I., Harvey, M.J., De Fabritiis, G.: Dimensionality reduction methods for molecular simulations (2017). 1710.10629
12. Shao, J., Tanner, S.W., Thompson, N., Cheatham, T.E.: Clustering Molecular Dynamics Trajectories: 1. Characterizing the Performance of Different Clustering Algorithms. Journal of chemical theory and computation 3(6), 2312–34 (2007). doi:10.1021/ct700119m
13. Kaufman, L., Rousseeuw, P.: Finding Groups in Data. Wiley-Interscience; New York, ??? (1990)
14. Hobohm, U., Scharf, M., Schneider, R., Sander, C.: Selection of representative protein data sets. Protein Science 1(3), 409–417 (1992). doi:10.1002/pro.5560010313
15. Zhang, Y., Skolnick, J.: Scoring function for automated assessment of protein structure template quality. Proteins: Structure, Function, and Bioinformatics 68(4), 1020 (2007). doi:10.1002/prot.21643
16. Ensign, D.L., Kasson, P.M., Pande, V.S.: 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)
17. Amato, N.M., Tapia, L., Thomas, S.: A Motion Planning Approach to Studying Molecular Motions. Communications in Information and Systems 10(1), 53–68 (2010). doi:10.4310/CIS.2010.v10.n1.a4
18. Oksanen, J., Blanchet, F.G., Friendly, M., Kindt, R., Legendre, P., McGlinn, D., Minchin, P.R., O'Hara, R.B., Simpson, G.L., Solymos, P., Stevens, M.H.H., Szoecs, E., Wagner, H.: vegan: Community Ecology Package. (2019). https://cran.r-project.org/package=vegan
19. Grant, B.J., Rodrigues, A.P.C., ElSawy, K.M., McCammon, J.A., Caves, L.S.D.: Bio3D: An R package for the comparative analysis of protein structures. Bioinformatics 22(21), 2695–2696 (2006). doi:10.1093/bioinformatics/btl461
20. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2018). R Foundation for Statistical Computing. https://www.r-project.org/
21. Kassambara, A., Mundt, F.: factoextra: Extract and Visualize the Results of Multivariate Data Analyses. (2017). https://cran.r-project.org/package=factoextra
22. Xu, J., Zhang, Y.: How significant is a protein structure similarity with TM-score = 0.5? Bioinformatics 26(7), 889–895 (2010). doi:10.1093/bioinformatics/btq066
23. Duan, M., Han, L., Rudolph, L., Huo, S., Carlson, G.H.: Geometric Issues in Dimensionality Reduction and Protein Conformation Space. (2014)