Algoritmo Rápido de Reducción de Trajectorias de Plegamiento de Proteínas

1 de septiembre de 2018

Resumen

En los últimos años se ha logrando realizar simulaciones de plegamiento de proteínas mucho más largas que llegan al orden de los milisegundos, lo que antes no se había realizado principalmente debido a las limitaciones en los recursos computacionales. Muchas de estos datos de trayectorias de plegamiento se están colocando a disposición pública para que sean analizados, sin embargo, debido a la inmensa cantidad de conformaciones de proteínas que resultan en estas trayectorias, su análisis se vuelve complejo. Por lo tanto, se vuelve necesario el desarrollo de métodos que logren obtener las conformaciones o grupos de conformaciones más representativas de la trayectoria teniendo en cuenta dos aspectos claves: el tiempo computacional y la calidad de los datos. En este trabajo, nosotros presentamos un algoritmo y una herramienta que realiza esta reducción de las trayectorias de plegamiento de proteínas teniendo en cuenta los dos aspectos anteriores.

Introducción

Actualmente se están liberando con más frecuencia datos de simulaciones de plegamiento de proteínas para que la comunidad científica los analice y avance en el entendimiento de este proceso. Estas simulaciones alcanzan tiempos de simulación que antes no se lograban debido a las limitaciones en los recursos computacionales. Hace algunos años el proyecto folding@home [2] liberó varias trayectorias de la simulación de la proteína Villin Headpiece la cual alcanzó el orden de los microsegundos utilizando computación distribuida. Más recientemente, el grupo de David Shaw liberó simulaciones de varias proteínas en el orden de los milisegundos utilizando la supercomputadora Anton diseñada especialmente para simular dinámica molecular [7, 9]. Todas estas simulaciones se caracterizan por generar trayectorias que abarcan miles o millones de conformaciones, lo cual es una gran ventaja porque se tiene más detalle del proceso, pero así mismo es un problema debido al tiempo y recursos computacionales necesarios para analizarlas.

Para reducir estas trayectorias los métodos actuales buscan conjuntos de conformaciones representativas, que generalmente utilizan métodos de agrupamiento donde se construye una matriz con las distancias entre cada una de las conformaciones, usualmente se usa la distancia conocida como RMSD o *Root Mean Square-Deviation*. Estos agrupamientos se vuelven muy costosos en tiempo y recursos computacionales cuando se trata de muchas conformaciones y por esta razón los algoritmos buscan simplificar estos costos, como por ejemplo, reducir el número de átomos que comparar en las conformaciones (solo carbonos alfa).

Otra forma de reducir estas trayectorias es crear agrupamientos rápidos que no tengan que comparar todas las conformaciones, parecido a lo que realiza el algoritmo de Hobohm&Sander [4] para comparar secuencias de ADN. En este trabajo presentamos un algoritmo rápido para reducción de trayectorias de plegamiento de proteínas que toma como base la idea del algoritmo de Hobohm&Sander y que se basa en tres estrategias: primero una partición de la trayectoria en múltiples secciones; segundo, una reducción local muy rápida sobre cada una de ellas que aprovecha el tiempo de ocurrencia de las conformaciones; y tercero, una reducción global que busca encontrar las conformaciones más representativas de cada partición. Estas tres estrategias permiten que este algoritmo sea fácilmente paralelizable, obtenga unos resultados previos de forma rápida, y de esos resultados seleccione los más importantes.

1. Antecedentes

En esta sección describiremos los elementos básicos que se manejan en este trabajo, principalmente hablaremos de plegamiento de proteína, simulaciones de plegamiento de proteínas, trayectorias de plegamiento y métodos de

reducción de datos biológicos:

1.1. Métodos de Comparación de Estructuras de Proteínas (MMart)

Revisar [[5]]. La idea es definir que es RMSD, GDS, TM-score con las referencias. Después empieza uno contras de RMSD (sensible), GDS (umbrales), y unos pros sobre TM-score (no tiene umbrales, idenpendiente del nro de aminos).

1.2. Plegamiento de Proteínas (MMart)

Revisar sección 2.2 Tesis LG[3].

1.3. Simulaciones de Plegamiento

Revisar secciones 2.7, especialmente la 2.7.1. de Tesis LG [3]

1.4. Simulaciones Largas de Plegamiento (MMart)

Las simulaciones del plegamiento de proteínas son complejas y demandan gran cantidad de tiempo y recursos computacionales. Debido a estas limitaciones tecnológicas, las simulaciones del plegamiento de proteínas hasta hace unos años 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 [**?**]. Sin embargo, en los últimos años los avances en el hardware han logrado algunos avances de tal manera que se empiezan a mostrar resultados de simulaciones 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 proteína pequeña, la Villin Headpiece de 36 residuos [8]. La supercomupatadora Anton es un proyecto más reciente (2010) que usa computación paralela y hardware especializado para simular dinámica molecular. 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.

1.5. Algoritmos de Agrupamiento Clásicos (MMart)

Hablar de k-means y k-medoides.

1.6. Algoritmos Rápido de Agrupamiento de Hobohn y Sander

El algoritmo de Hobohm y Sander [4] se creó inicialmente para agrupar de forma rápida secuencias de proteínas, determinando 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. Luego, 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, la secuencia que no es muy similar 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 y UCLUST. El programa CD-HIT [6] 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 [1] utiliza para comparar 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.

2. Algoritmo de Reducción de Trayectorias de Plegamiento (LuisG)

La primera parte del algoritmo realiza un agrupamiento local rápido donde se aprovecha el ordenamiento temporal de las conformaciones implícito en la trayectoria. Pare esto, se toma la idea del algoritmo propuesto por Hobohm et al. [1] 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.

3. Datos y Métodos (LuisG)

3.1. Trayectorias de Plegamiento de Proteínas

Para mostrar los resultados del algoritmo de reducción propuesto, aplicamos las reducciones 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 [10]* 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 [7] 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).

3.2. Trayectorias de Plegamiento de Proteínas generada por Anton

Estas simulaciones fueron realizadas por en una supercomputador especialmente diseñado para resolver problemas de Dinámica Molecular, llamada Anton [9], que es el método de más utilizado para simular el plegamiento de proteínas. Las trayectorias puestas a disposición corresponden a las simulaciones del plegamiento completo en solvente explicito de 12 proteínas de 10 a 02 residuos [**?**]. Por cada

4. Detalles de Implementación (LuisG)

4.1. Descripción de Programas

El algoritmo está implementado a través de tres scripts:

- pr00_main.py: Script principal en lenguaje Python que toma los parámetros iniciales y llama a los otros scripts enviándoles los parámetros necesarios.
- pr01_createBins.py: Script en lenguaje Python que realiza las particiones
- pr02_localReduction.R : Script en lenguaje R que realiza la reducción local.
- pr03_globalReduction.R: Script en lenguaje R que realiza la reducción global.

4.2. Ejecución

La ejecución se realiza llamando al script *reduction.py* así:

```
$ ./reduction.py <InputDir> <Ouputdir> <BinSize> <Threshold> <K> <nCores>
```

Donde:

- **Input Dir:** Nombre del directorio de entrada donde están las conformaciones de la trayectoría de la proteína a reducir.
- **Output Dir:** Nombre del directorio donde quedarán los resultados de la reducción. Si el directorio ya existe lo renombra automáticamente y crea uno nuevo. Dentro del directorio se crean cuatro subdirectorios:
  - bins: donde se crean las particiones con las conformaciones correspondientes a cada bin
  - binsLocal: donde se crean las nuevas particiones con los resultados de la reducción local
  - pdbsLocal donde se copian todas las conformaciones de la nueva trayectoria producto de la reducción local.
  - pdbsGlobal: donde se copian todas las conformaciones de la nueva trayectoria producto de la reducción global.
  - tmp: donde se coloca los archivos temporales resultantes de la creación de las matrices de distancia con TM-score
- **Bin Size:** El tamaño de conformaciones por partición o *bin*. El algoritmo crea el número de *bins* dependiendo del tamaño de la trayectoria.
- **Threshold:** Umbral usado por el TM-score para comparar dos conformaciones y decidir si son similares.
- **K:** Número de conformaciones a seleccionar por el agrupamiento global
- **nCores:** Número de *cores* a utilizar para el procesamiento en paralelo.

4.3. Requisitos

Los programas están en python y en R. Del sistema R se necesita instalar las librería para agrupamientos y paralelización: *cluster* y *parallel*, respectivamente.

5. Resultados y Discusión

Para mostrar las reducciones 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.
En la Figura 1 mostramos las reducciones de las dos trayectorias cortas correspondientes a las proteínas 1FCA1 y 2YCC (ver sección 3.1). 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 reducciones 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 reducciones que reflejan la dinámica de la trayectoria. Además, destaquemos 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, lo cual va a ser más evidente en el caso de la trayectoria larga descrita a continuación.

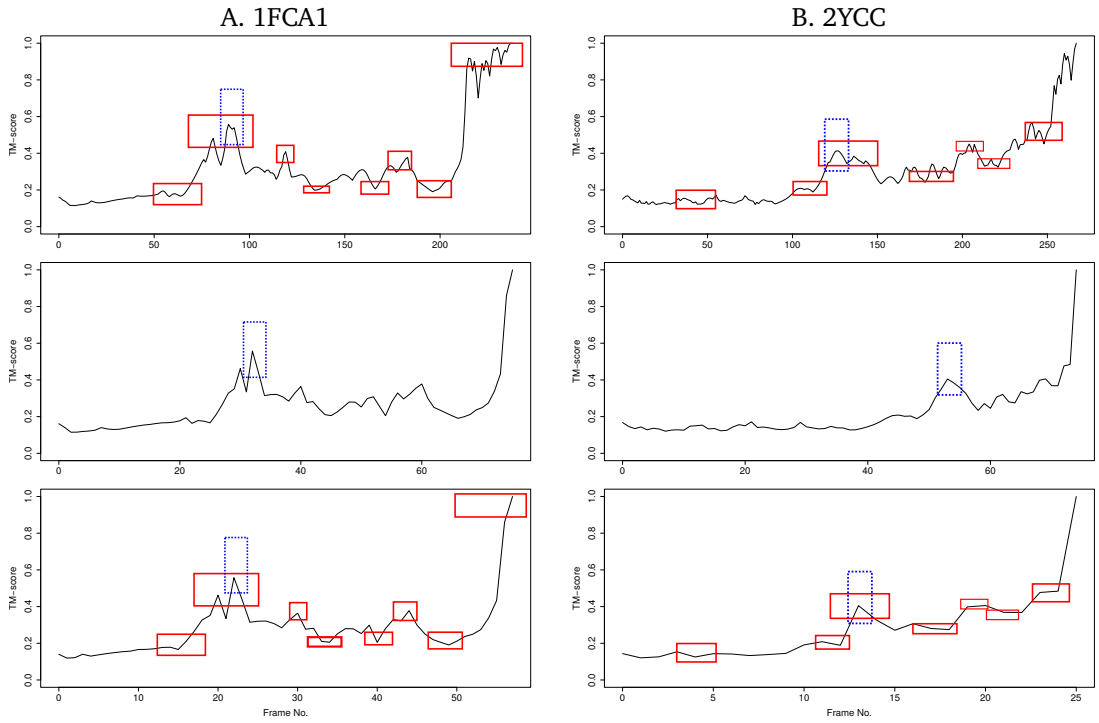

Figura 1: **Reducción de las trayectorias cortas de plegamiento para las proteínas 1FCA1 y 2YCC.** En recuadros rojos se resaltan los eventos principales que se conservan tanto en la trayectoria original como en la final. Los recuadros rojos muestran como algunos eventos principales se desplazan en la 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

Ahora, en la Figura 2 observamos la reducción hecha sobre una trayectoria larga de más de 1 millón de conformaciones para la proteína 2FOF (ver sección 3.1). La reducción final fue de más del 97 % (de 1044004 a 20883 conformaciones). Observamos que a pesar de que la simulación presenta bastantes oscilaciones en el plegamiento, en general los eventos principales al final de la reducción global se conservan. Es importante notar aquí que la reducción local no describe claramente los eventos principales, como lo destacamos en las reducciones anteriores, sin embargo la reducción global que toma los datos de la local, logra destacarlos cuando selecciona las conformaciones más representativas de cada partición.

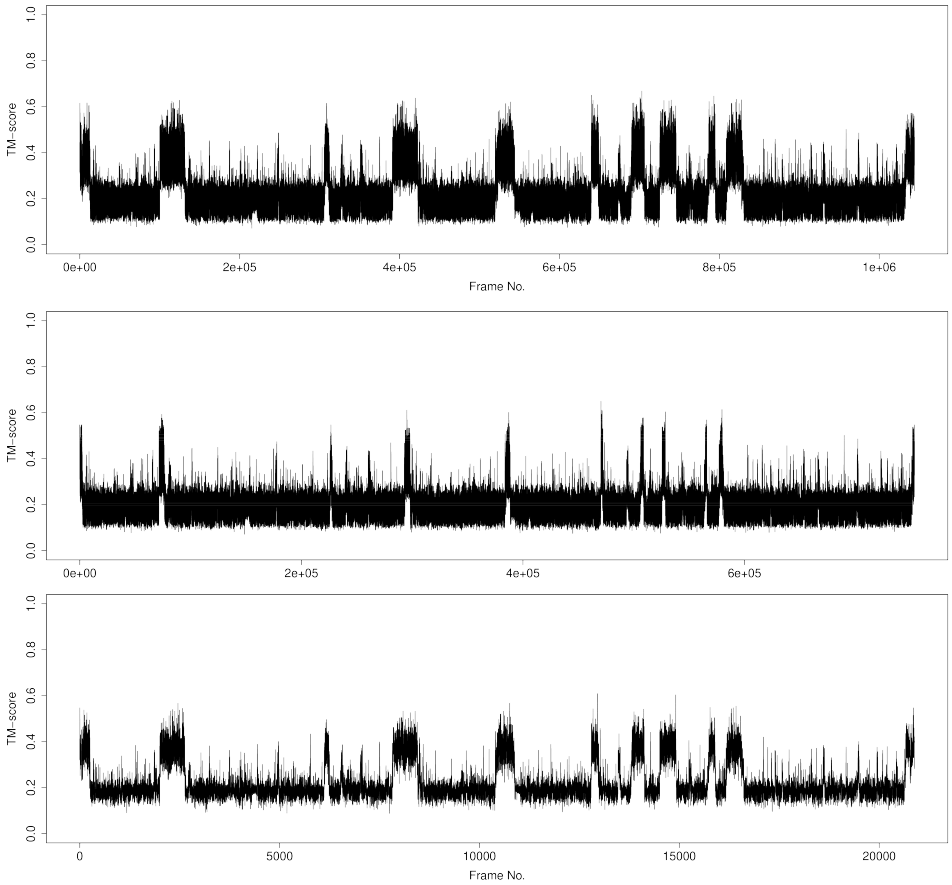

Figura 2: Reducción de una trayectoria larga de plegamiento.

6. Conclusiones

En este trabajo presentamos un algoritmo de reducción de trayectorias que visualmente produce reducciones que logran preservar la dinámica de la trayectoria 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] Robert C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 2010.

[2] 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.

[3] Luis Garreta. *Conformational Folding Status and Folding Levels Based on Global Protein Properties : a Computational Approach*. PhD thesis, Universidad del Valle, 2015.

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

[5] Irina Kufareva and Ruben Abagyan. Methods of protein structure comparison. pages 231–257, 2015.

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

[7] K Lindorff-Larsen, S Piana, R O Dror, and D E Shaw. How Fast-Folding Proteins Fold. *Science*, 334(5):517–520, oct 2011.

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

[9] 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.

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