Título en progreso: Reducción de secuencias de proteínas mediante clustering

Mauricio Martínez Jiménez[1]

***Abstract*— El abstract se realizará al finalizar la composición del artículo.**

I. INTRODUCTION

Se escribirá al finalizar el artículo.

II. Estado del Arte

Cada vez, especialmente en el campo de la biología, la cantidad de datos disponibles para ser analizados es mayor. Actualmente existen varias técnicas disponibles para extraer información manejable de grandes cantidades de datos. Entre ellas se encuentran el clustering y PCA, las cuales han sido de uso común en varias situaciones, aunque no existe una guía definitiva que permita determinar el algoritmo específico para cada caso (Datta S Datta S).

A. Clustering

El clustering es un método no supervisado para la clasificación de observaciones en grupos. El problema de clustering ha sido abordado desde diferentes contextos y tiene repercusión en distintas áreas del conocimiento, ya que se ha convertido en un paso muy importante en la etapa de análisis de datos exploratorios en muchas investigaciones. Representa un problema combinatorio y por ello se considera difícil (Gan, Ma, & Wu, 2007).

Clustering es una técnica de minería de datos que busca agrupar un conjunto de datos en clusters o grupos de tal manera que los objetos dentro del cluster tienen gran similitud pero son muy disímiles a los objetos de otros clusters. Dichas similitudes o disimilitudes son obtenidas a partir de atributos que describen los objetos. En general, los algoritmos de clustering son usados para organizar y categorizar datos, detectar valores atípicos, entre otros. Algo común a todas las técnicas de clustering es la tarea de encontrar un centro que represente cada cluster. Son varias los algoritmos que se han desarrollado y se categorizan de acuerdo a su enfoque. Entre ellos se encuentran los métodos de particionamiento, los jerárquicos, basados en densidad y los basados en cuadrícula (Joshi & Kaur, 2013).

Dado que en el campo de la bioinformática uno de las tareas más comunes consiste en detectar grupos de objetos cercanamente relacionados, el clustering es un tema de gran importancia en esta disciplina.

El clustering es útil en varios análisis exploratorios de patrones, agrupamiento, toma de decisiones y aprendizaje de máquinas. En muchos de estos problemas hay poca información preliminar, por lo que se hace necesario hacer algunas asunciones sobre los datos, cuando es posible.

En general se pueden identificar varios pasos en un proceso de clustering: . Representación del patrón: Número de clases, patrones disponibles, número, tipo y escala de los elementos. . Definición de una medida de proximidad apropiada para el dominio de datos: Generalmente se mide como una función de distancia entre pares de elementos. Es posible usar una medida de distancia simple como la Euclidiana. . Agrupamiento: Existen varias maneras de ejecutar el agrupamiento. La salida puede ser dura, es decir, que los datos sean particionados en grupos; o difusa, donde cada patrón tiene un grado variable de pertenencia a los clusters de salida. . Abstracción de datos: Consiste en extraer una representación simple y compacta de un conjunto de datos (Jain, Murty, & Flynn, 1999).

1) Clustering jerárquico: Los algoritmos de clustering duros se dividen en jerárquicos y par-

[1]Mauricio Martínez Jiménez, Escuela de Ingeniería y Ciencias de la Computación

ticionales. Un algoritmo particional divide un conjunto de datos en una única partición, mientras que un algoritmo jerárquico divide el conjunto de datos en una secuencia de particiones anidadas. Los métodos jerárquicos, a su vez, se dividen en aglomerativos y divisivos.

Los algoritmos jerárquicos aglomerativos comienzan con cada elemento en un cluster separado. Luego empieza a repetir mezclas con el par más cercano de acuerdo a algún criterio de similaridad hasta que todos los datos se encuentran en un mismo cluster. Las principales desventajas de éste método es que una mala agrupación temprana no se puede reacomodar, y al usar distintas medidas de similaridad se obtienen resultados muy distintos.

Los algoritmos jerárquicos divisivos tienen una aproximación contraria al aglomerativo. Empiezan con todos los elementos en un cluster y comienza con una serie de divisones repetitivas para obtener clusters más pequeños.

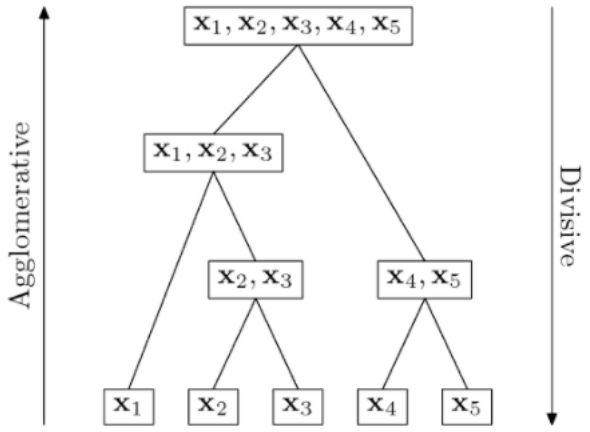

FIG 1: Diagrama comparativo de un algoritmo aglomerativo y uno divisivo.

Los resultados de un algoritmo jerárquico son representados gráficamente en un dendrograma.

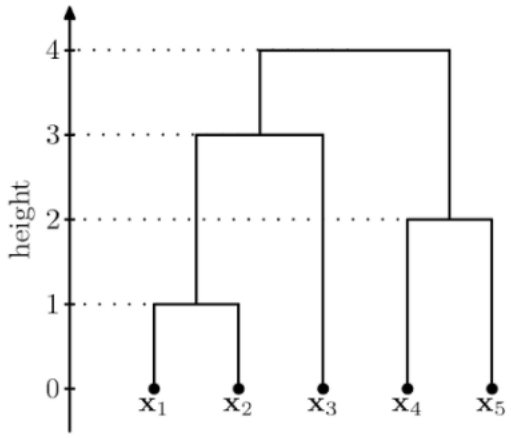

FIG 2: Dendograma.

2) Problemas de escalamiento: Estas técnicas, sin embargo, presentan problemas que en ocasiones hacen su utilización poco viable. Uno de los más notorios es la poca escalabilidad frente a conjuntos de datos demasiado grandes. Los datos pueden ser tan grandes que no caben en la memoria, o incluso pueden estar almacenados de forma distribuída. Los algoritmos de clustering convencionales requieren generalmente acceso repetido a todos los datos, lo que lo hace un proceso muy costoso (Bolaños, Forrest, & Hahsler, 2014).

Los investigadores han usado diversas aproximaciones para abordar el problema. Entre ellas se pueden encontrar las técnicas de computación paralela (como el uso, pr ejemplo, del framework MapReduce de Google (Li, Hu, Li, Wu, & Yang, 2016) o la reducción de datos por muestreo, cómo el caso de CLARA (Kaufman & Rousseeuw, 2005), que usa muestreo y luego aplica Partitioning Around Medoids (PAM) en las muestras y retorna el cluster que mejor se ajusta. Otras investigaciones en este campo se orientan al uso de flujos de datos (Data Streaming), concepto que ha tenido un auge importante en investigación en los últimos años. Los algoritmis de flujo de datos han sido diseñados para procesar grandes volúmenes de datos de una manera eficiente usando una única pasada por los datos mientras se tiene un mínimo de sobrecosto en los requerimientos de almacenamiento (Bolaños et al., 2014).

B. PCA

Karl Pearson desarrolló los fundamentos matemáticos de PCA en 1901 (Pearson, 1901). Sus aplicaciones aparecieron inicialmente en las

ciencias sociales, donde fue usada para cuantificar e identificar fenomenos que no se podían medir directamente.

PCA es una técnica para reducir la dimensionalidad de conjuntos de datos. Permite que estos se puedan interpretar mas fácilmente, mientras se minimiza la perdida de información. Esto lo logra al crear nuevas variables no correlacionadas para las cuales se maximiza sucesivamente su varianza (Jolliffe & Cadima, 2016).

PCA permite la identificación de grupos de variables que están interrelacionadas a través de fenómenos no observables directamente. Esto se logra al asumir que cualquier variable observada (manifiesta) está correlacionada con un número pequeño de fenómenos subyacentes que no se pueden medir de manera directa (variables latentes). PCA es una examinación automática y sistemática de correlaciones entre variables manifiestas, cuyo propósito es el de hacer inferencias sobre las identidades de cualquier variable latente. La elección de las variables manifiestas es un paso crucial en un análisis de PCA, y deben seleccionarse aquellas que reflejan distintos aspectos de fenómenos subyacentes (Burstyn, 2004).

PCA se puede generalizar a análisis de correspondencia (CA) de manera que pueda manejar variables cualitativas y también a análisis de factores múltiples (MFA) para manejar conjuntos heterogéneos de variables (Abdi & Williams, 2010).

Las aplicaciones de PCA son variadas. Se pueden citar, por ejemplo, el filtro de imágenes (Vargas, Villa, & Gonzáles, 2016), reconocimiento de rostros (Riaz, Gilgiti, & Mirza, 2004), descomposición de señales de electrocardiográma (Kanaujia, 2015), clasificación de imágenes de cristalización de proteínas (Dinc et al., 2014) y similitud en estructuras de proteínas (Chen, Chang, & Tian, 2010).

C. Data reduction in protein trajectories

La dinámica molecular es una de las técnicas computacionales usadas para la simulación de plegamiento de proteínas. Es un proceso que demanda gran cantidad de recursos computacionales, por lo que las simulaciones se ven restringidas a escalas de tiempo de microsegundos (Towse & Daggett, 2013). Como ejemplo se puede tomar los estudios realizados con la villina, una proteína de 35 residuos que se pliega en aproximadamente 4.5 microsegundos (Freddolino & Schulten, 2009). Hasta 1998, la simulación de 1 microsegundo por Duan y Kollman había sido la más larga hasta la época (Duan & Kollman, 1998) Con el tiempo, este límite se ha ido extendiendo poco a poco, como el ejemplo de la simulación de aproximadamente 6 microsegundos ejecutada por Freddolino y Schulten (Freddolino & Schulten, 2009). Las trayectorias obtenidas con estas simulaciones contienen millones de frames, siendo cada uno como una fotografía en cada momento de las coordenadas en tercera dimensión de los átomos de la proteína.

Una de las técnicas usadas para esta compresión de las trayectorias es nMSD. Puede alcanzar gran compresión de los datos y al mismo tiempo conserva las características destacadas de las trayectorias. La técnica (non-metric multidimensional scaling method) reduce la dimensionalidad de los datos conservando las interrelaciones entre los mismos. Se trata de un método de geometrización no supervisado que coloca N puntos que representan los objetos de estudio (como pueden ser los frames de una trayectoria) en un cierto espacio E, de tal manera que las distancias por parejas de los puntos en E sean consistentes con las disimilitudes por parejas de los objetos en los datos de entrada. La característica de ”no métrico” se da porque en lugar de manejar distancias numéricas, usa rangos, y lo que importa es conocer las relaciones entre distancias (por ejemplo, que la distancia entre A y B es menor que la distancia entre A Y C) (Kruskal, 1964). Como una crítica a las técnicas de clustering, se ha dicho que estas producen clústeres inestables o clústeres con interrelaciones no conocidas. Aunque la compresión obtenida con nMSD es alta, su costo computacional es también elevado, y este es un aspecto que se debe considerar a la hora de elegir la técnica de reducción adecuada. Otro punto a tener en cuenta es que la técnica es aún muy dependiente de una interpretación visual de los resultados, lo que dificulta su aplicación en procesos que se desean lo más automáticos posible (Rajan, Freddolino, & Schulten, 2010).

III. Métodos

...

IV. Resultados (quizas esto sean los métodos)

La solución propuesta permite obtener un porcentaje determinado de la población total de secuencias, aprovechando las ventajas de la programación paralela. Este porcentaje se obtiene de una manera tal que las secuencias resultantes representen las características estructurales del conjunto inicial. De esta manera, los análisis que posteriormente se realicen sobre este subconjunto, generarán resultados similares a los que se obtendrían si se analizara la totalidad de las secuencias.

Las simulaciones de sistemas físicos, como es el caso de la dinámica molecular, generan "snapshots" del sistema simulado. Los steps en una simulación corresponden al cálculo del siguiente estado del sistema, y el timestep es el intervalo de tiempo para pasar de un estado a otro.

La longitud de los timesteps determina por lo tanto la cantidad de snapshots que se obtienen en una simulación dada, suponiendo constante el tiempo de la simulación. Por ejemplo, los datos analizados en este artículo corresponden a una simulación de cerca de 200 microsegundos con timesteps de 200 picosegundos. Con esta configuración se tiene alrededor de 1.000.000 snapshots. La reducción que se realiza corresponde a un aumento en la longitud del timestep.

De esta manera, si se decide partir de un timestep de 2000 picosegundos en lugar de 200 picosegundos, el aumento de 10 veces en dicha longitud corresponde con una reducción de 10 veces en la cantidad de snapshots, es decir, se obtienen 10.000 snapshots. Esta cantidad es la que se define como el número de elementos de la población original que se tomarán como elementos "representativos" y permitirá que sea más manejable el procesamiento posterior.

A. *Descripción del proceso*

A continuación se describen los pasos llevados a cabo para la reducción de las secuencias. El proceso se muestra de manera gráfica en la imagen siguiente.

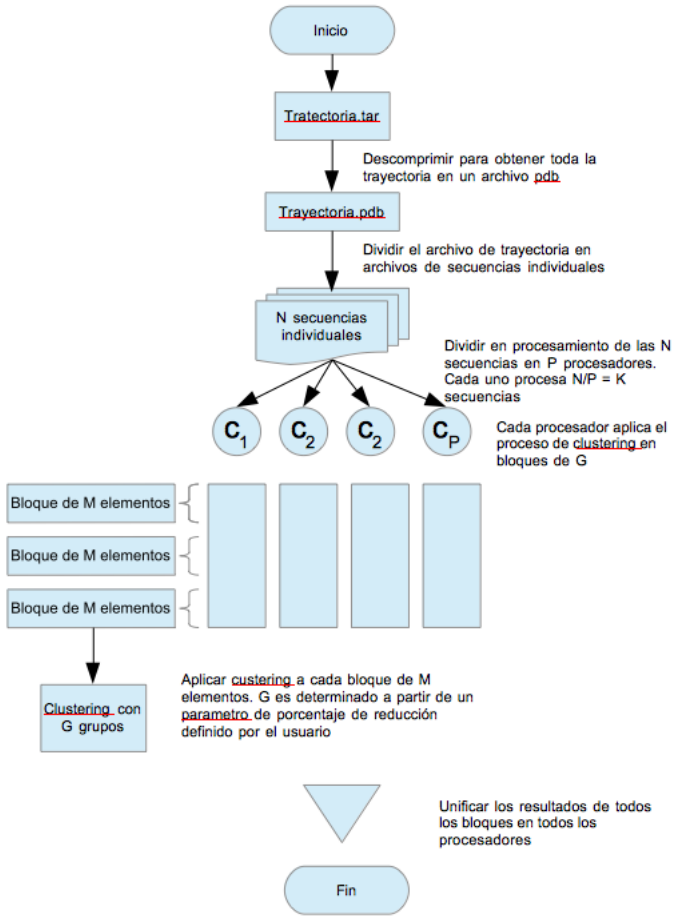

FIG 1: Diagrama comparativo de un algoritmo aglomerativo y uno divisivo.

1) Adición de átomos: Dependiendo de la fuente, las secuencias estarán "completas" o no. Cuando no están completas, las secuencias contienen únicamente los carbonos alfa, resultando esto en archivos mucho más livianos y fáciles de almacenar y distribuir. Otros servidores pueden proveer las secuencias con todos los átomos, de una manera similar a la que se puede encontrar en un repositorio como el PDB. Este paso de adición de átomos, por lo tanto, se realiza para aquellas secuencias que contienen únicamente los carbonos alfa. En este caso, se hace uso de algún algoritmo que complete la secuencia, esto es, que basándose en información de posiciones de átomos en estrucutras completas, reconstruyen la secuencia asignando las coordenadas más probables para los átomos faltantes. Pulcra es un ejemplo de una herramienta que implementa la funcionalidad de reconstrucción

para secuencias que contienen únicamente carbonos alfa.

2) Paralelización del procesamiento: Ya que la cantidad de proteínas a analizar es de alrededor un millón, es conveniente aprovechar las infraestructuras multinúcleo que cada vez son más fáciles de acceder y de ésta manera dividir la carga de trabajo. Otra razón por la cuál se usa esta paralelización es la naturaleza misma del problema. Al ser la cantidad de proteínas un número considerablemente alto, procesar todos los datos de una sola vez conduce a una demanda excesiva en recursos, principalmente en memoria, ya que se deben calcular y comparar las distancias de cada secuencia con las demás. Por eso, se cuenta con un parámetro que indica el número de procesadores que se pueden usar para el procesamiento. El total de proteínas a analizar se divide entre el número de procesadores y esa cantidad es la que cada uno de ellos procesa.
3) Aplicación del método de clustering: Cada núcleo debe aplicar el método de clustering al bloque de proteínas que le corresponde debido a la división del trabajo. Sin embargo, cada bloque puede contener un número arbitrariamente alto de proteínas a procesar, ya que el conjunto inicial de proteínas no tiene un límite en cuanto a cantidad. Por eso, se hace una nueva división, esta vez generando bloques de un número fijo de proteínas, de tal manera que se vayan procesando secuencialmente y no se tengan problemas de memoria. Estos bloques de tamaño fijo son los que finalmente se usan como entrada para el algoritmo de clustering, el cual analiza las secuencias y genera tantos grupos como se ha especificado a través de un parametro dependiente del porcentaje que el usuario desea obtener de la población total. La salida de este paso es un archivo de texto con un representante de cada grupo generado. Cada representante se obtiene eligiendo el primer elemento de cada grupo.
4) Unificación de los resultados: En este paso se leen todos los archivos generados en el paso anterior y se copian a un directorio los archivos pdb correspondientes a los representantes de cada grupo obtenido. Este directorio contiene la cantidad de secuencias que representa el porcentaje indicado por el usuario, del total de proteínas.

V. Conclusiones

...

References

Abdi, H., & Williams, L. J. (2010, jul). *Principal component analysis* (Vol. 2) (No. 4). John Wiley & Sons, Inc. Retrieved from http://doi.wiley.com/10.1002/wics.101 doi: 10.1002/wics.101

Bolaños, M., Forrest, J., & Hahsler, M. (2014). Clustering large datasets using data stream clustering techniques. In *Studies in classification, data analysis, and knowledge organization* (Vol. 47, pp. 135–143). Springer, Cham. Retrieved from http://link.springer.com/10.1007/978-3-319-01595-8_15 doi: 10.1007/978-3-319-01595-8-15

Burstyn, I. (2004). Principal component analysis is a powerful instrument in occupational hygiene inquiries. *Annals of Occupational Hygiene*, *48*(8), 655–661. doi: 10.1093/annhyg/meh075

Chen, Y., Chang, S., & Tian, X. (2010, sep). 2nd PCA on 3D protein structure similarity. In *2010 ieee fifth international conference on bio-inspired computing: Theories and applications (bic-ta)* (pp. 253–257). IEEE. Retrieved from http://ieeexplore.ieee.org/document/5645321/ doi: 10.1109/BICTA.2010.5645321

Dinc, I., Sigdel, M., Dinc, S., Sigdel, M. S., Pusey, M. L., & Aygun, R. S. (2014, mar). Evaluation of normalization and PCA on the performance of classifiers for protein crystallization images. In *Conference proceedings - ieee southeastcon* (pp. 1–6). IEEE. Retrieved from http://ieeexplore.ieee.org/document/6950744/ doi: 10.1109/SECON.2014.6950744

Duan, Y., & Kollman, P. A. (1998, oct). Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science (New York, N.Y.)*, *282*(5389), 740–4. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/9784131

Freddolino, P. L., & Schulten, K. (2009, oct). Common structural transitions in explicit-solvent simulations of villin headpiece folding. *Biophysical journal*, *97*(8), 2338–47. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/19843466 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2764099 doi: 10.1016/j.bpj.2009.08.012

Gan, G., Ma, C., & Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications.* SIAM, Society for Industrial and Applied Mathematics. doi: 10.1017/CBO9781107415324.004

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999, sep). Data clustering: a review. *ACM Computing Surveys*, *31*(3), 264–323. Retrieved from http://portal.acm.org/citation.cfm?doid=331499.331504 doi: 10.1145/331499.331504

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, *374*(2065), 20150202. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/26953178%5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4792409 doi: 10.1098/rsta.2015.0202

Joshi, A., & Kaur, R. (2013). A Review: Comparative Study of Various Clustering Techniques in Data Mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(3), 2277–128.

Kanaujia, M. (2015). ECG Signal Decomposition Using PCA and ICA. *National Conference on Recent Advances in Electronics & Computer Engineering*, 301–305.

Kaufman, L., & Rousseeuw, P. J. (2005). *Finding Groups in Data: An Introduction to Cluster Analysis* (Vol. 33) (No. 1). Wiley. Retrieved from http://www.amazon.com/Finding-Groups-Data-Introduction-Analysis/dp/0471878766 doi: 10.1007/s00134-006-0431-z

Kruskal, J. B. (1964, jun). Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, *29*(2), 115–129. Retrieved from http://link.springer.com/10.1007/BF02289694 doi: 10.1007/BF02289694

Li, R., Hu, H., Li, H., Wu, Y., & Yang, J. (2016, aug). *MapReduce Parallel Programming Model: A State-of-the-Art Survey* (Vol. 44) (No. 4). Springer US. Retrieved from http://link.springer.com/10.1007/s10766-015-0395-0 doi: 10.1007/s10766-015-0395-0

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(1), 559–572. Retrieved from http://dx.doi.org/10.1080/14786440109462720 doi: 10.1080/14786440109462720

Rajan, A., Freddolino, P. L., & Schulten, K. (2010, apr). Going beyond Clustering in MD Trajectory Analysis: An Application to Villin Headpiece Folding. *PLoS ONE*, *5*(4), e9890. Retrieved from http://dx.plos.org/10.1371/journal.pone.0009890 doi: 10.1371/journal.pone.0009890

Riaz, Z., Gilgiti, A., & Mirza, S. (2004). Face Recognition: A review and comparison of HMM, PCA, ICA and Neural Networks. *Pakistan Institute of Engineering and Applied Sciences*, 41–46. doi: 10.1109/ETECH.2004.1353842

Towse, C.-L., & Daggett, V. (2013). Protein Folding: Molecular Dynamics Simulations.

In *Encyclopedia of biophysics* (pp. 2020–2025). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-16712-6_607 doi: 10.1007/978-3-642-16712-6-607

Vargas, d. l. R. A. d. Z., Villa, J. U. A. d. Z., & Gonzáles, E. U. A. d. Z. (2016). A tour of nonlocal means techniques for image filtering. *Electronics, Communications and Computers (CONIELECOMP)*. doi: 10.1109/CONIELECOMP.2016.7438548