Asociación Española
de
Ingeniería Mecánica

XIX CONGRESO NACIONAL
DE INGENIERÍA MECÁNICA

Simulación del mecanismo molecular con incrementos optimizados en los ángulos diédricos

Mikel Diez Sánchez [(1)], **Víctor Petuya Arcocha**[(1)], **Mónica Urízar Arana**[(1)], **Erik Macho Mier**[(1)]

(1) Dpto. Ingeniería Mecánica. Universidad del País Vasco (UPV/EHU)
mikel.diez@ehu.es

La simulación de los posibles movimientos que puede presentar una proteína a la hora de llevar a cabo su función sigue siendo hoy en día un campo lleno de incógnitas. La mayoría de los métodos existentes en la bibliografía, utilizan simplificaciones que si bien posibilitan la simulación del proceso, no obtienen trayectorias cinemáticamente posibles. En este artículo se presenta un algoritmo para la simulación del mecanismo molecular de la proteína, el cual posee la capacidad de modificar los valores de los incrementos en los ángulos diédricos en función de la información recopilada a lo largo de la simulación. El algoritmo se ha utilizado para la simulación del mecanismo molecular de 4 proteínas, 1zac, 1k20, 1k9p y 3cln, cada una de las cuales posee un tipo de movimiento diferente.

1. INTRODUCCIÓN

La simulación del comportamiento de las proteínas en sus diversas funciones es una de las tareas más difíciles de la bioquímica. Los ingentes costes computacionales asociados a los procesos y algoritmos de simulación hace complicada la obtención de información relativa a los mecanismos mediante los cuales las proteínas llevan a cabo sus funciones o, más importante aún, mediante los cuales se construyen. Estos estudios son de vital importancia ya que las proteínas toman parte en casi todos los procesos vitales de los seres vivos. Una mejor comprensión de su funcionamiento facilitará la correcta interpretación de dichos procesos.

La simulación del mecanismo molecular de las proteínas esta enfocada a el análisis de los movimientos macroscópicos que ciertas proteínas requieren para llevar a cabo su función [1]. La proliferación de procedimientos de simulación basados en teorías o metodologías ya utilizadas en robótica o cinemática de mecanismos es cada vez mayor. El uso de métodos basados en el análisis de modos y frecuencias [2] o en la generación de trayectorias [3,4] son ya conocidos para este tipo de simulaciones. Nuevos métodos [5] utilizan procesos de minimización de los esfuerzos que aparecen en los enlaces de la estructura para refinar las trayectorias entre dos posiciones conocidas de una proteína.

El procedimiento presentado en este artículo es una evolución del presentado en [6]. Este algoritmo evolucionado posee la capacidad de modificar tanto el valor de los incrementos, como el sentido de giro de los grados de libertad de la proteína, permitiendo la exploración de un mayor espacio configuracional. Pese a la mayor complejidad del algoritmo y a seguir considerando la totalidad de átomos presentes en la proteína, el algoritmo sigue manteniendo una buena relación coste computacional-precisión. Al igual que en el trabajo anterior, la continuidad cinemática del movimiento esta asegurada al no realizar procesos intermedios de minimización energética. Del mismo modo los Ramachandran plots obtenidos demuestran la viabilidad biológica de los resultados.

2. MODELO CINEMATICO DE LA PROTEINA

Las proteínas son unas cadenas polipéptidas formadas por la unión de aminoácidos. Todos los aminoácidos comparten una estructura principal, compuesta por un átomo de nitrógeno y dos de carbono. Del átomo de carbono central, denominado Cα, nace una cadena de átomos, particular de cada aminoácido, denominada cadena secundaria o radical (R_i) (figura 1). Esta cadena es la responsable de definir las características químicas particulares de cada aminoácido.

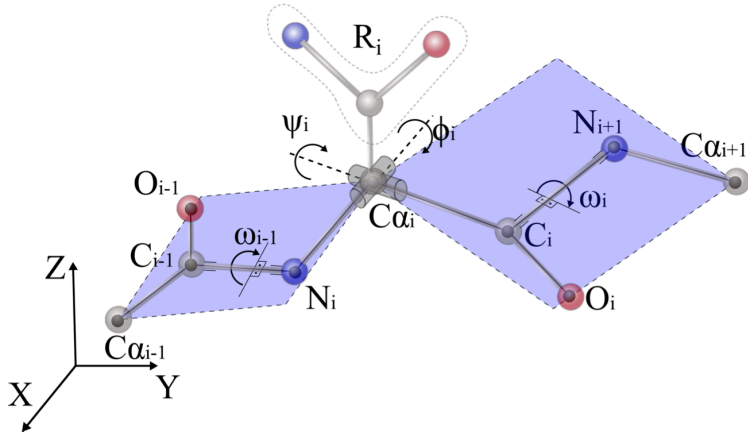


Figura 1. Modelización de la estructura de la cadena de una proteína con sus grados de libertad

La cadena polipéptida se forma mediante la unión covalente del último átomo de carbono C_i de un aminoácido con el átomo de nitrógeno N_{i+1} del siguiente aminoácido. Este enlace covalente, denominado enlace peptídico, tiene propiedades de doble enlace, por lo que la rotación alrededor del el mismo esta impedida. Lo enlaces propios del aminoácido, $N_i - C\alpha_i$ y $C\alpha_i - C_i$ sí tienen libertad de rotación, limitada únicamente por las interferencias de las cadenas secundarias. Los ángulos relativos en dichos enlaces se denominan ϕ_i y ψ_i (figura 1).

Debido a que el algoritmo requiere los valores de los ángulos diédricos para el cálculo de los incrementos, estos han de ser lo más exactos posibles. En una proteína, las fuerzas que mantienen unidos a los átomos también los hacen vibrar alrededor de una posición de equilibrio. Debido a que los métodos experimentales obtienen imágenes estáticas de las proteínas, es prácticamente imposible que en dos instantáneas de una misma proteína, pese a que la estructura global sea la misma, los átomos mantengan sus posiciones relativas. Estos errores producen leves modificaciones en los valores de las longitudes y ángulos de enlace. Si esto lo trasladamos al cálculo de los incrementos angulares en los ángulos diédricos de la proteína la consecuencia es que los incrementos calculados contienen un error. Para minimizar este error se ha aplicado el algoritmo de normalización presentado en [7]. Este algoritmo utiliza restricciones de distancia y plano para normalizar la estructura (figura 2).

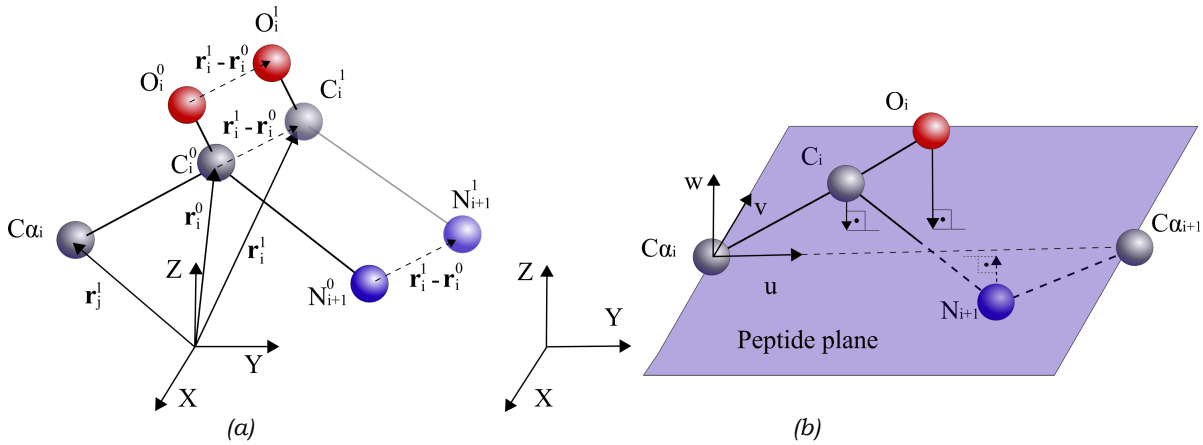


Figura 2. (a) Restricción de distancia aplicada al átomo C_i. Todos los átomos posteriores en la cadena son desplazados junto con el C_i para no distorsionar la estructura. (b) Normalización de los planos peptídicos. El cálculo del plano medio se realiza mediante coordenadas locales.

3. ALGORITMO PARA LA SIMULACIÓN DEL MECANISMO MOLECULAR

El algoritmo que presentamos en este artículo es la evolución del presentado en [6,7]. La anterior versión del algoritmo optaba por bloquear los grados de libertad del algoritmo que habían generado mayores incrementos de energía. Este procedimiento produce un modelo cinemático de la proteína excesivamente rígido. Por este motivo se ha decidido modificar este algoritmo para lograr un modelo cinemático más flexible. Al igual que en el trabajo anterior el algoritmo sigue haciendo uso de los grados de libertad reales, es decir, los ángulos diédricos, para avanzar a lo largo de la simulación. Los datos de las estructuras iniciales de las proteínas se han obtenido del *protein data bank* PDB. Para obtener una referencia de la evolución del error se han comparado las estructuras obtenidas con las estructuras dato del Morph server. Las simulaciones se han realizado mediante el software GIMPRO desarrollado por nuestro grupo de investigación.

Los resultados se han validado utilizando tres indicadores diferentes. En primer lugar el error cuadrático medio o rmsd. Este error hace referencia a la similitud global entre las estructuras obtenidas en la simulación y las estructuras utilizadas como dato. En segundo lugar se ha evaluado la evolución de la energía potencial de la proteína. Este indicador es particularmente sensible a distorsiones locales en la estructura de la proteína aumentando su valor cuando dos o más átomos están a punto de colisionar. De entre todos los campos potenciales disponibles en la bibliografía se ha seleccionado el campo de AMBER con los parámetros propuestos por Cornell [8]. El último indicador son los Ramachandran Plots de las estructuras obtenidas. Los Ramachandran Plots indican el sentido biológico de las estructuras obtenidas.

El algoritmo evolucionado está presentado en Algoritmo 1. Este algoritmo, antes de bloquear el giro de un grado de libertad trata de aplicar su rotación con un incremento menor. El algoritmo comienza realizando el cálculo de la energía inicial de la proteína E^0. Una vez calculado, el proceso de simulacion comienza rotando de forma secuencial los grados de libertad de la proteína, desde el primer aminoácido al último. Una vez rotados, se obtiene el valor de la energía potencial de la configuración actual E^k. Este valor no puede exceder el máximo admitido para la iteracion k actual, $E^0 + E^0 \cdot \epsilon_k$ donde ϵ_k es una tolerancia energetica cuyo objetivo es distribuir uniformemente los cambios en la energía potencial de la proteína. El cálculo de ϵ_k se realiza mediante la siguiente expresion:

$$\epsilon_k = \frac{k \cdot \epsilon}{p} \tag{1}$$

donde ϵ es el cambio porcentual de la energía a lo largo de toda la simulación y p es el número de pasos que se quieren realizar. Ambos valores son definidos por el usuario. En caso de que E^k supere el limite establecido el algoritmo busca el grado de libertad que ha generado el mayor incremento energetico. La rotación en este grado de libertad es deshecha y vuelta a aplicar reduciendo su valor a la mitad *(*$\Delta\psi_i/2 \,||\, \Delta\phi_i/2$*).* Si resulta que esta nueva rotación no logra reducir el valor de la energía hasta el limite establecido el algoritmo procede a deshacer el nuevo giro y a bloquear el grado de libertad en la actual iteración. El proceso se va repitiendo con todos los grados de libertad hasta que el valor de la energía es menor que en la iteración anterior. A lo largo de la simulación el algoritmo guarda un registro del número de veces que cada grado de libertad ha sido frenado o bloqueado. En el evento de que un grado de libertad sea bloqueado m veces el algoritmo considera que el grado de libertad no puede girar en dicha dirección y cambia su sentido de rotación ($\Delta\psi_i = -\Delta\psi_i \,||\, \Delta\phi_i = -\Delta\phi_i$) durante las próximas n iteraciones. Ambos valores m y n son definidos por el usuario.

Algoritmo 1 Algoritmo para el paso k

foreach Grado de libertad de la proteína **do**
  Rotar el grado de libertad *i* ($\Delta\psi_i||\Delta\phi_i$)
  E_i^k ←Evaluación de la energía potencial después de la rotación del grado de libertad *i*
  $\Delta E_i^k = E_i^k - E_{i-1}^k$ ←Incremento de energía asociado al grado de libertad *i*
end foreach
  $E^k = E_i^k$ ←Energía potencial después de la iteración *k*
while $\left[\frac{E^k - E^0}{E^0}\right] \geq \epsilon^k$ **do**
  Deshacer la rotación el grado de libertad *i* ($\Delta\psi_i||\Delta\phi_i$) asociado al máximo ΔE_i^k
  Rotar el grado de libertad *i* con incremento reducido ($\Delta\psi_i/2||\Delta\phi_i/2$)
  E^k ←Evaluación de la energía potencial
  If $\left[\frac{E^k - E^0}{E^0}\right] \geq \epsilon^k$ **then**
    Deshacer el giro del el grado de libertad *i* ($-\Delta\psi_i/2||-\Delta\phi_i/2$)
  end if
  $\Delta E_i^k = 0$
  If el grado de libertad *i* se ha frenado mas de *m* veces **then**
    $\Delta\psi_i = -\Delta\psi_i||\Delta\phi_i = -\Delta\phi_i$
  end if
  E^k ←Evaluación de la energía potencial
end while

Ya que el algoritmo posee la capacidad de modificar los valores de los incrementos en los ángulos diédricos e incluso cambiar su sentido de giro, es más que probable que una vez alcanzadas las p iteraciones definidas por el usuario éstos no hayan alcanzado sus valores finales. Por este motivo, cada vez que el algoritmo frena o bloquea un grado de libertad automáticamente p es incrementado en 1. Esto también hace que el incremento energético definido para cada iteración por la ecuación 1 se ajuste dinámicamente a lo largo de la simulación.

La simulación puede terminar debido a tres factores. Primero, los ángulos diédricos han alcanzado sus valores finales. Segundo, el algoritmo ha consumido completamente la

energía disponible para la simulación. Y tercero, debido a que el algoritmo tiene la capacidad de incrementar el número máximo de iteraciones, con el proposito de no realizar simulaciones excesivamente largas, se ha definido un máximo de $10 \cdot p$ iteraciones.

4. EJEMPLOS DE APLICACIÓN

Para evaluar la eficacia del algoritmo propuesto se han seleccionado cuatro proteínas von diferentes tipos de movimientos. La proteína 1k20 posee un movimiento de pinza en la que una zona de la proteína se desplaza respecto a la otra. La proteína 1zac posee el mismo movimiento aunque de menor magnitud. El movimiento de la proteína 1k9p esta definido por la posición relativa de dos hélices α. Por último, en la proteína 3cln el movimiento esta definido por la unión de dos hélices α en una única hélice (figura 3).

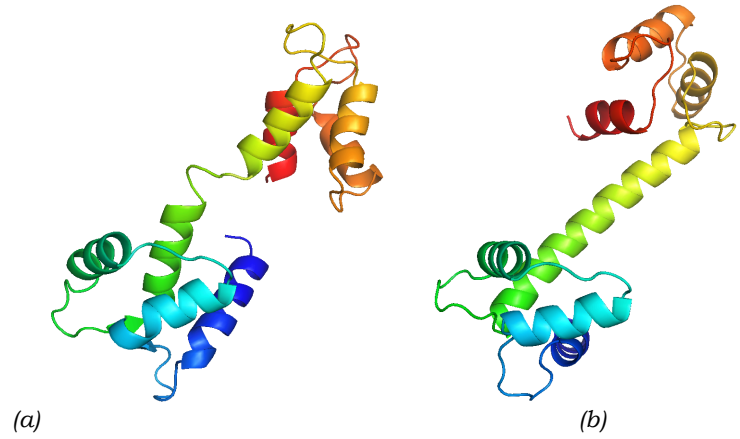


Figura 3. (a) Configuración incial y (b) final del movimiento de la proteína 3cln. Se aprecia perfectamente como se forma la hélice central arrastrando el bloque superior completo. Representadas mediante Pymol

Como se ha dicho, el algoritmo presentado en este artículo posee la capacidad de modificar los valores de los incrementos de los ángulos diédricos de la proteína. Este comportamiento se controla mediante los parámetros m y n. En aras de comprobar la influencia de estos parámetros se han realizado dos simulaciones, una con los valores $m = n = 2$ y otra con los valores $m = 3, n = 1$. El resto de parámetros se han definido con los siguientes valores; número máximo de iteraciones $p = 100$, tolerancia de energía máxima $\epsilon = 10\%$. Los resultados obtenidos para las diferentes proteínas se muestran en la tabla 1. Para comparar los resultados obtenidos con este algoritmo también se han realizado las mismas simulaciones con la versión previa del mismo utilizando los mismos valores de p y ϵ, los resultados de estas simulaciones se muestran en la tabla 2.

Proteína	**m=n=2**			**m=3, n=1**		
	Rmsd(Å)	**Energía (%)**	**RP(% de átomos en las zonas favorables)**	**Rmsd(Å)**	**Energía (%)**	**RP(% de átomos en las zonas favorables)**
1k9p	-	-	-	3.78*	6*	97*
1k20	5.48	6.2	92	5.29	4.8	93
1zac	-	-	-	3.08	4	98
3cln	5.52	3.7	93	5.34	3	92

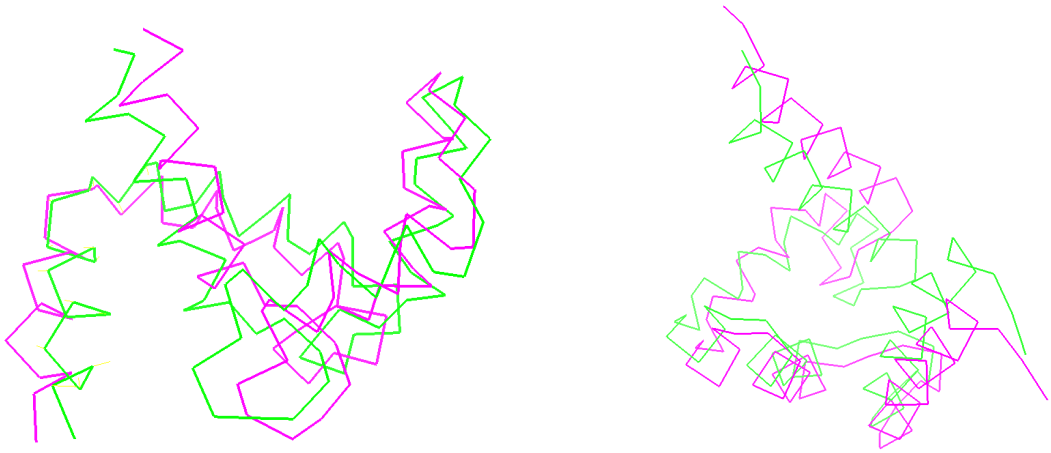
*Estos resultados no representan un movimiento real de la proteina

Tabla 1. Resultados de la simulación del mecanismo molecular con el algoritmo presentado

Proteína	Rmsd(Å)	Energía (%)	RP(% de átomos en las zonas favorables)
1k9p	4.01	7	96
1k20	6.18	7.6	92
1zac	3.44	6.9	97
3cln	6.88	0	95

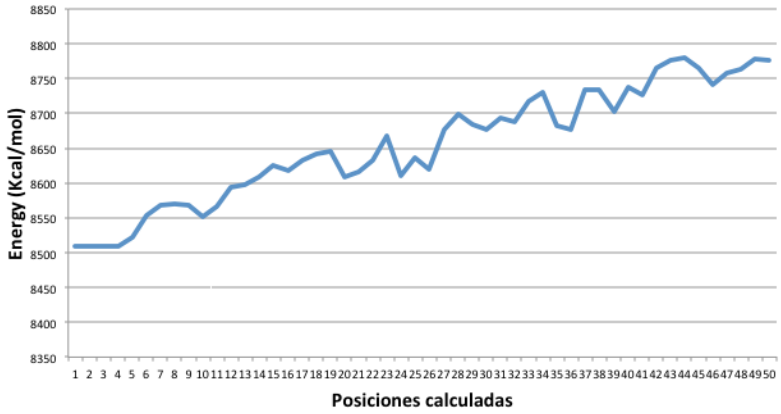
Tabla 2. Resultados obtenidos por la versión anterior del algoritmo

Como se puede observar en la tabla 1, el algoritmo presentado en este artículo mejora los resultados obtenidos por el primer algoritmo en las proteínas 1k20, 1zac y 3clnl, reduciendo los errores rmsd en un 14%, 10% y 22% respectivamente. Los valores de los Ramachandran Plots muestran cómo se ha mantenido el sentido biológico de las proteínas en todo momento. En lo referente a la energía se puede observar como los incrementos obtenidos han sido bajos demostrando la ausencia de choques entre átomos. En la figura 4(a) se puede ver una superposición de la estructura obtenida (morado) y la estructura dato (verde) de la proteína 1zac. Por otro lado en la figura 5 se muestran las graficas de la evolución de la energía potencial y el rmsd para la simulación de la proteína 3cln. Por lo que respecta a la proteína 1k9p el algoritmo presentado en este artículo no logra obtener una solución válida con ninguna de las combinaciones de parámetros propuestas. Esto se debe en gran medida a las interferencias entre las cadenas secundarias de la proteína, las cuales, son tratadas por el algoritmo como sólidos rígidos trasladándolas junto con la cadena principal. Por este motivo el algoritmo despliega parcialmente la proteína para alcanzar la posición final, obteniendo un movimiento no válido de la proteína. En todo caso, a pesar de ello, el algoritmo sí logra alcanzar una posición final con un error rmsd de 3.78 Å, un incremento energético del 6% y colocando el 97% de los átomos en las zonas preferentes de los Ramachandran Plots. En la figura 4(b) se muestra la superposición de esta estructura final obtenida con respecto a la estructura dato.

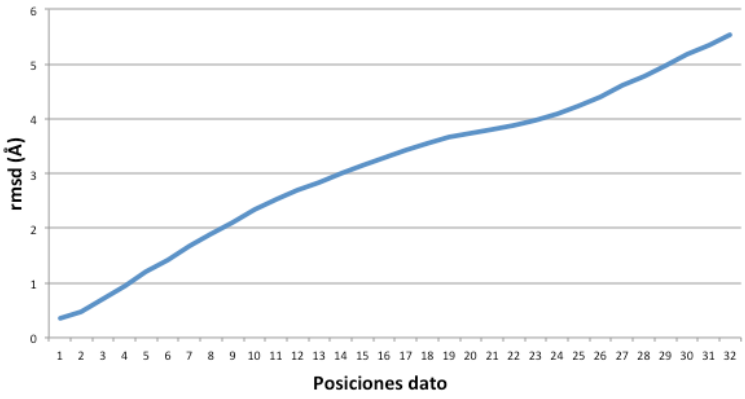


(a) (b)

Figura 4. Superposición de la estructura obtenida (morado) y dato (verde) de la proteína 1zac (a) y 1k9p (b).



(a)



(b)

Figura 5. Resultados para la proteína 3cln con los parámetros m=3 y n=1. (a) Evolución de la energía potencial de las posiciones obtenidas por el algoritmo evolucionado. (b) Evolución del rmsd respecto a las posiciones dato del Morph server.

Centrándonos en los resultados obtenidos por el algoritmo presentado en este artículo podemos observar notables diferencias entre las dos combinaciones de parámetros. Los mejores resultados han sido obtenidos para los valores $m = 3, n = 1$. El parámetro m controla las veces que un grado de libertad es bloqueado antes de que se cambie su sentido de rotación. Por otro lado el parámetro n define el número de veces que dicho grado de libertad rota en sentido contrario. Desde un punto de vista cinemático esta combinación de parámetros produce un modelo más rígido de la estructura de la proteína, reduciendo el número de veces que un grado de libertad puede rotar en sentido contrario. Por el contrario con los valores $m = n = 2$ el cambio de rotación se produce antes y durante mayor intervalo de tiempo. Esto provoca que los ángulos diédricos tengan más facilidad de retroceso y no alcancen la posición final deseada. Se observa que para la simulación del mecanismo molecular de estas proteínas el modelo cinemáticamente más rígido aporta mejores resultados. De todas formas, la selección de los parámetros dependerá de las características del mecanismo molecular de cada proteína. Por ejemplo, en simulaciones que exijan una mayor libertad de movimiento, como podría ser la simulación del plegado de proteínas, sería más conveniente una selección de parámetros que permita la exploración de un mayor espacio configuracional.

5. CONCLUSIONES

El estudio de las proteínas se está convirtiendo en una ciencia cada vez más interdisciplinar. No solo matemáticas, física o biología tienen cabida sino también la

ingeniería. Las similitudes entre las proteínas y los mecanismos de cadena abierta o con las estructuras de barras permiten aplicar técnicas y conocimientos de problemas cinemáticos y dinámicos a su estudio.

En este artículo se ha descrito un algoritmo para la simulación de mecanismo molecular de las proteínas mediante el cual llevan a cabo su función. Este algoritmo proporciona a la proteína mayor libertad de movimiento permitiendo la exploración de nuevas trayectorias. Esta libertad está determinada por la capacidad del algoritmo de modificar los valores de los incrementos en los ángulos diédricos de la proteína así como de su sentido de giro. Este comportamiento esta controlado por dos parámetros cuya modificación altera la libertad de movimiento de la proteína obteniendo un modelo mas o menos flexible de la cadena principal de la proteína. Los resultados muestran como este nuevo algoritmo mejora los resultados obtenidos por su versión previa.

El trabajo actual se centra en proporcionar a la proteína de una mayor libertad de movimiento añadiendo grados de libertad asociados a las cadenas secundarias. Esto permitirá mejorar los resultados en movimientos de carácter interno como el que presenta la proteína 1k9p. Por otro lado y de cara a reducir todavía más el coste computacional del método se esta trabajando en un algoritmo para detectar estructuras secundarias a lo largo del proceso de simulación para eliminar los grados de libertad asociados a las mismas.

6. AGRADECIMIENTOS

Los autores desean agradecer el soporte financiero recibido por parte del gobierno mediante Ministerio de Educación y Ciencia (Proyecto DPI2008-00159), La Unión Europea (Proyecto FP7-CIP-ICT-PSP-209-3) y el Gobierno regional del País Vasco mediante el Departamento de Educación, Universidades e Investigación (Proyecto IT445-10).

7. REFERENCIAS

[1] Lucas M, Encinar JA, Arribas EA, Oyenarte I, García IG, Kortazar D, Fernández JA, Mato JM, Martínez-Chantar ML, Martínez-Cruz LA. *Binding of S-methyl-5'-thioadenosine and S-adenosyl-L-methionine to protein MJ0100 triggers an open-to-closed conformational change in its CBS motif pair*. J Mol Biol. (2010) 396(3):800-20.

[2] Adam D. Schuyler, Gregory S. Chirikjian. *Efficient determination of low-frequency normal modes of large protein structures by cluster-NMA*. Journal of Molecular Graphics and Modelling (2005), 24, 46–58.

[3] Thomas, S., Song, G., Amato, N. M.. *Protein folding by motion planning*. Physical Biology (2005), 2(4), S148–S155.

[4] Singh, A. P., Latombe, J. C., & Brutlag, D. L.. *A motion planning approach to flexible ligand binding. Proceedings*. ISMB International Conference on Intelligent Systems for Molecular Biology (1999), 252–261.

[5] C Madden, P Bohnenkamp, K Kazerounian, H T Ilies. Residue Level Three-dimensional Workspace Maps for Conformational Trajectory Planning of Proteins. The International Journal of Robotics Research (2009), vol. 28 (4) pp. 450-463.

[6] M. Diez, V. Petuya, Ch. Pinto, A. Hernandez. Un algoritmo con feedback energético para la simulación cinemática del movimiento de las proteínas. Anales de Ingeniería Mecánica Año 17 (2010).

[7] Mikel Diez, Víctor Petuya, Luis Alfonso Martínez-Cruz, Alfonso Hernández. *A biokinematic approach for the computational simulation of proteins molecular mechanism*. Mechanism and Machine Theory (2011), vol. 46 (12) pp. 1854-1868.

[8] WD Cornell, P Cieplak, CI Bayly, IR Gould, KM Merz, DM Ferguson, DC Spellmeyer, T Fox, JW Caldwell, PA Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. Journal of the American Chemical Society, vol. 117 (19) pp. 5179-5197, 1995.