

Plegamiento de Proteínas 2D

Álvaro Úbeda Ruiz
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
alvuberui@alum.us.es

Estefanía Ganfornina Triguero
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
estgantri@alum.us.es

El objetivo principal de este trabajo es implementar uno de los primeros modelos de plegamientos de proteínas que construyeron la base para abordar el histórico problema del plegamiento de las proteínas desde la computación y la inteligencia artificial. Para ello se debe comprender la teoría básica detrás del plegamiento de proteínas y la paradoja de Levinthal y comparar estos conceptos con un problema de optimización. Además, se deberá identificar las distintas disciplinas necesarias para estudiar la paradoja de Levinthal, entender la importancia de la multidisciplinariedad y aprender a implementar un algoritmo de plegamiento de proteínas en 2D, usar la librería `pyplot` y usar la base de datos UNIPROT.

En conclusión, con este proyecto, vamos a poder principalmente comprobar lo complejo que es el problema del plegamiento de proteínas, debido al gran número de posibilidades que hay y de lo costoso que puede llegar a ser en la mayoría de los casos. Además, también aprenderemos más sobre la teoría de las proteínas, plegamientos, implementación de algoritmos, etc.

Palabras Clave— Proteínas, plegamiento, algoritmo, enfriamiento simulado.

I. INTRODUCCIÓN

Las proteínas son unas de las macromoléculas más importantes en la biología molecular. Están presentes en todas las formas de vida conocidas (e incluso en los virus) y son las encargadas de llevar a cabo casi todos los procesos necesarios para que la vida siga adelante.

Las proteínas están formadas por cadenas de aminoácidos que se doblan y acoplan entre sí espontáneamente, en un proceso llamado plegamiento de proteínas, para formar una estructura tridimensional relacionada con la función biológica de la proteína.

Uno de los desafíos de la biología molecular, conocido como el “problema del plegamiento de las proteínas” consiste en entender cómo la secuencia de aminoácidos determina la estructura tridimensional. Para solucionar este problema, es necesario comprender la termodinámica de las fuerzas interatómicas que resultan en una estructura estable y el mecanismo por el que las proteínas alcanzan su configuración final con extrema rapidez.

Las estructuras de las proteínas se determinan habitualmente de forma experimental mediante técnicas que son costosas y pueden requerir mucho tiempo. Durante los últimos sesenta años solo se han identificado las estructuras de

unas 170 000 proteínas, de las más de doscientos millones que se calcula que existen en todas las formas de vida conocidas. El poder predecir la estructura de las proteínas sin más información que la secuencia de aminoácidos sería de gran ayuda para avanzar en la investigación científica. Sin embargo, la paradoja de Levinthal muestra que, si bien una proteína se puede plegar en milisegundos, el tiempo que lleva calcular todas las estructuras posibles al azar para determinar la estructura óptima es más largo que la edad del universo conocido. En resumen, se trata de un problema de optimización en el que hay innumerables soluciones con una puntuación distinta cada una.

A lo largo de los años, los investigadores han aplicado numerosos métodos computacionales al problema de la predicción de la estructura de las proteínas, pero la precisión de los modelos generados jamás se había acercado a la de las estructuras determinadas por técnicas experimentales, excepto para proteínas pequeñas y simples. Hasta la entrada en escena de AlphaFold2 (2020), un programa de inteligencia artificial (IA) desarrollado por DeepMind de Alphabets/Google que realiza predicciones de la estructura de las proteínas mediante Deep Learning. Este hito multidisciplinar ha marcado un antes y un después en la comunidad científica.

En este trabajo se va a implementar uno de los primeros modelos de plegamiento de proteínas, menos complejos que AlphaFold2, pero que construyeron la base para abordar el histórico problema del plegamiento de las proteínas desde la computación y la inteligencia artificial.

Para ello, se buscará alcanzar los siguientes objetivos específicos, comprender la teoría básica detrás del plegamiento de proteínas y la paradoja de Levinthal. Comparar estos conceptos con un problema de optimización, identificar las distintas disciplinas necesarias para estudiar la paradoja de Levinthal. Entender la importancia de la multidisciplinariedad, aprender a implementar un algoritmo en base a un modelo de plegamiento de proteínas 2D, realizar gráficos con `pyplot` para representar los plegamientos de proteínas 2D y aprender a usar la base de datos de proteínas UNIPROT.

A. Paradoja de Levinthal

Si suponemos que cada enlace peptídico tiene sólo tres grados de libertad, entonces para una proteína pequeña de unos 101 aminoácidos existirán $3^{100} = 5 \times 10^{47}$ conformaciones. Aún si la proteína fuera capaz de explorar estas conformaciones a

una enorme velocidad, digamos de 10^{13} conformaciones por segundo, lo que significa 3×10^{20} por año, entonces le tomaría 10^{27} años probar todas las posibilidades, es decir, si la edad del Universo se estima en 1.37×10^{10} años (13 mil 700 millones de años), una proteína requeriría un tiempo de 17 órdenes de magnitud mayor que la edad del Universo para plegarse. Por lo tanto, empleando esta lógica parece imposible que las proteínas encuentren su estructura óptima, a pesar de que el tamaño promedio de las cadenas polipeptídicas oscila alrededor de los 300 aminoácidos. Sin embargo, una proteína se pliega en una escala de tiempo de segundos o menos.

Levinthal propuso que la búsqueda conformacional estocástica no es el mecanismo real del plegamiento, sino que éste debe ser un proceso dirigido por las condiciones fisicoquímicas existentes durante la biosíntesis de la proteína y las rutas bioquímicas que experimentan a posteriori, o sea una vez plegadas (Levinthal 1968 y 1969).

Para comprender la paradoja de Levinthal necesitamos comprender la cinética y estructura de las proteínas.

II. ALGORITMOS DE PLEGAMIENTO DE PROTEÍNAS 2D Y COMO CONTRIBUYERON A LOS ALGORITMOS ACUALES

El problema del plegamiento de proteínas consiste en la predicción de la estructura a partir de la secuencia de aminoácido. Este problema puede definirse como:

Dadas una determinada secuencia de aminoácidos y una función de energía a evaluar, encontrar una conformación de mínima energía.

Para resolver el problema del plegamiento de proteínas simplificado se han propuesto varios métodos a través de los años. Dentro de estos métodos encontramos la Búsqueda Tabú (Morales et al., 2000), el Recocido Simulado (Garduño et al., 1990; Morales et al., 1991), los métodos de Monte Carlo (Ramakrishnan et al., 1997; Bastolla et al., 1998; Chikenji et al., 1999; Liang et al., 2001; Zhang y Liu, 2002), los Algoritmos Evolutivos (Unger y Moulton, 1993a; Unger y Moulton, 1993b; Patton y Goldman, 1995; Krasnogor et al., 1998; Garduño et al., 2003) y la Optimización por Colonia de Hormigas (ACO) (Shmygelska et al., 2002; Shmygelska y Hoos, 2003; Chu et al., 2005; Shmygelska y Hoos, 2005; Fidanova, 2006; Song et al., 2006).

El problema de estos algoritmos convencionales es que son muy lentos en comparación con la complejidad que tiene la resolución de los plegamientos de las proteínas.

En el año 2018, DeepMind desarrolló el programa AlphaFold y a finales de 2020, anunció un avance significativo con AlphaFold2. Dicho programa ha exhibido el poder del aprendizaje automático en el reconocimiento de patrones en secuencias primarias para determinar estructuras tridimensionales con alta precisión. El componente central de AlphaFold es una red neuronal que se entrena con un gran número de estructuras en el banco de datos de proteínas para predecir distribuciones de distancias entre los átomos C β de pares de residuos de una proteína y construir un campo de fuerza artificial para dirigir el plegamiento sin utilizando una plantilla individual, pero en patrones derivados de muchas

proteínas. También se ha basado en gran medida en bases de datos de secuencias y múltiples alineaciones de secuencias.

El programa AlphaFold2 ha superado significativamente a otros métodos produciendo modelos de una calidad cercana a la de la determinación experimental. Lo que hace es comparar, mediante aprendizaje profundo haciendo uso de una red neuronal artificial, estructuras y secuencias de aminoácidos para las ciento setenta mil proteínas para las que se cuenta con el conocimiento necesario (de los doscientos millones que existen en la naturaleza). Y a partir de esa comparación es capaz de predecir la forma de proteínas cuya estructura se desconoce, pero de las que se sabe su secuencia. Los creadores de AlphaFold sostienen que este desarrollo es la puerta que abrirá el paso al diseño y producción de fármacos con la forma adecuada para actuar sobre dianas específicas, así como a otros avances en áreas de medicina y biología en las que es importante conocer la estructura de las proteínas que participan en procesos de interés. Quizás esas pretensiones sean prematuras, pero el logro es impresionante, y marcará un antes y un después en el conocimiento de las estructuras de los seres vivos y en sus posibles aplicaciones.

III. PRELIMINARES

En esta sección se hará una breve introducción a las técnicas empleadas para la resolución del problema de plegamiento de proteínas.

A. Métodos empleados

Para resolver el histórico problema de plegamiento de proteínas se ha usado un sencillo algoritmo en base a la hidrofobicidad de los aminoácidos.

Para ello, se ha implementado un algoritmo de Enfriamiento Simulado. Este algoritmo, como otros muchos, se trata de un método de optimización inspirado en un proceso natural, es decir, fue creado basándose en la observación de una excelente optimización de un fenómeno que tiene lugar en la naturaleza. El Enfriamiento o Recocido Simulado es una de la metaheurísticas más clásicas. Se ha convertido en una herramienta muy popular gracias a su simplicidad y buenos resultados en numerosos problemas. Aunque es muy fácil hacer que el Recocido Simulado funcione, es complejo hacer que funcione correctamente. Esto se debe a que no es propiamente un algoritmo, sino una estrategia heurística que necesita de la toma de varias decisiones que influyen de gran manera en la calidad de las soluciones que genera.

El algoritmo de Enfriamiento Simulado elige elementos del entorno de manera aleatoria, aceptando automáticamente todas las soluciones que aporten una mejora, y aceptando soluciones peores de manera probabilística. Esta probabilidad irá en descenso a medida que disminuya la temperatura.

Para poder implementar este algoritmo es necesario tomar una serie de decisiones cómo controlar la temperatura (definición de su valor inicial y función de descenso), número de enfriamientos, número de iteraciones y cómo se obtendrá la solución inicial. Además, se debe establecer la función de coste o energía.

IV. METODOLOGÍA

En esta sección se explicarán los algoritmos y métodos empleados para la resolución del problema del plegamiento de las proteínas.

Como se ha indicado anteriormente, en este proyecto se ha implementado un sencillo algoritmo de plegamiento de proteínas en 2D en base a la hidrofobicidad de los aminoácidos. Utilizando para ello un algoritmo de Enfriamiento Simulado (Simulated annealing) para resolver este problema de optimización.

Esta implementación se ha llevado a cabo resolviendo varios problemas propuestos más sencillos:

- Problema 1: Se ha implementado la función `get_spatial_dic` que recibe una cadena representando una proteína (letras de aminoácidos) y otra cadena representando su estructura (I para el aminoácido inicial, N, S, E o W según la posición relativa de un aminoácido con respecto al anterior) y devuelve un diccionario o un diccionario vacío si existen solapamientos. Las claves de dicho diccionario son tuplas de dos números enteros que representan coordenadas espaciales y los valores son letras de aminoácidos. Esta función no tiene en cuenta si las letras de los aminoácidos son correctas.
- Problema 2: En este problema se han implementado dos funciones, `is_hydrophobic` y `get_score`. La primera recibe una letra representando un aminoácido y devuelve True si es hidrofóbica, es decir, si su variación energía es mayor a -1,5 y False si es hidrofílica. La segunda función recibe un diccionario representando la estructura espacial de una proteína y devuelve su puntuación. La puntuación de un aminoácido será $\Delta G * N$ (si el aminoácido no es hidrofóbico) y $\Delta G * N + 10 * N$ (si el aminoácido es hidrofóbico). Siendo N el número de posiciones adyacentes libres. Para ello, se recorre el diccionario que se le pasa como parámetro, y para cada valor se comprueba cuantas coordenadas libres hay alrededor suya para posteriormente calcular su puntuación dependiendo de si son hidrofóbicas o no. Esta será la función de coste que deberemos minimizar en el algoritmo de Enfriamiento Simulado.
- Problema 3: En este caso se han creado las funciones `fold` y `get_successors`. La función `fold` devuelve la estructura tras aplicar el plegado indicado por la posición y el ángulo que se le pasan como parámetros. En primer lugar, se crea una copia de la estructura

inicial que se va actualizando con los valores nuevos tras el plegado. Esta actualización se realiza desde la posición de plegado hasta el final de la estructura y dependerá del ángulo que se haya indicado.

- Si el ángulo es de 90:
 - N -> W
 - S -> E
 - E -> N
 - W -> S
- Si el ángulo es de -90:
 - N -> E
 - S -> W
 - E -> S
 - W -> N

Por otro lado, la función `get_successors` va realizando todos los posibles plegamientos desde la posición 1 hasta la última posición de la estructura con ambos ángulos (90 y -90). Tras realizar cada plegamiento obtiene el diccionario espacial y si es válido lo guarda. Finalmente devuelve un diccionario con todos los diccionarios espaciales válidos.

- Problema 4: Usando las funciones creadas en los problemas anteriores, se ha implementado el algoritmo de Enfriamiento Simulado (Simulating Annealing) para resolver el problema del plegado 2D como un problema de optimización.

Para implementar dicho algoritmo se han creado dos funciones auxiliares:

- `initial_state`: recibe una proteína y devuelve una estructura inicial.
- `get_successor`: recibe una proteína y su estructura y aplica un plegamiento aleatorio. Finalmente devuelve la estructura tras el plegamiento y el diccionario espacial de esa proteína.

Tras crear las funciones auxiliares se ha implementado el algoritmo de Enfriamiento Simulado. Este recibe como parámetros la proteína, una temperatura inicial, factor de descenso, número de enfriamientos y número de iteraciones.

Enfriamiento Simulado (Simulating Annealing)

Entrada:

- Proteína (p)
- Temperatura inicial (to)
- Factor de descenso (fd)
- Número enfriamientos (enf)
- Número iteraciones (iter)

Salidas:

- Un diccionario espacial obtenido a partir del mejor estado. (dic)

Algoritmo:

```
temp = to
act_state = estructura inicial p
act_value = puntuación act_state
1. Mientras i < enf
    1.1 Mientras j < iter
        i. sucesor de p
        ii. candidate = estructura del sucesor
        iii. candidate_value = puntuación del sucesor
        iv. incr= valor del candidato – valor actual
            1. Si valor del candidato < valor actual
                a. Actualizar la estructura y el valor actual con el del candidato.
            2. Sino
                a.  $q = \text{mínimo entre } 1 \text{ y } e^{(-\text{incr}/\text{temp})}$ 
                b.  $r = \text{valor aleatorio entre } 0 \text{ y } 1, \text{ menor que } q.$ 
                c. Si  $r = 1$ 
                    i. Actualiza estructura y valor actual con el del candidato.
            3. Incrementar j en 1
    1.2 Disminuir temperatura por el factor de descenso
    1.3 Incrementar i en 1
2. Devolver dic
```

- Problema 5: La resolución de este problema se detallará en el siguiente apartado del documento. En este se pide:
 - Probar los algoritmos con al menos otras tres proteínas sencillas cuya secuencia de aminoácidos y descripción sea extraída desde *UNIPROT*.
 - Hacer una representación gráfica de las estructuras 2D resultante con *pyplot*. Coloreando de distinto color los aminoácidos dependiendo de si son hidrofóbicos o no.
 - Implementar un gradiente de color dependiendo del valor ΔG del

correspondiente aminoácido. (Siendo este último apartado opcional)

V. RESULTADOS

En esta sección se detallarán tanto los experimentos realizados como los resultados conseguidos.

Para comprobar su correcto funcionamiento se han realizado varias pruebas con proteínas sencillas cuya secuencia de aminoácidos y descripción han sido extraídas desde UNIPROT como se indicaba en el primer apartado del problema 5. Estas proteínas son:

- Glucose-6-phosphatase del organismo del Homo Sapiens (Humano)
- Insulin del organismo 'Homo sapiens (Human)'
- Tat protein del organismo 'Human immunodeficiency virus 1'

Para hacer la representación gráfica de las estructuras 2D resultante con *pyplot* se ha implementado una función *represent* que recibe una estructura y un título (nombre de la proteína a representar). En primer lugar, recorre toda la estructura dividiendo los puntos a representar en dos diccionarios dependiendo de si el aminoácido es hidrofóbico o hidrofílico. Posteriormente, representa la estructura completa para que se pueda mostrar la secuencia y, a continuación, se representan los puntos hidrofóbicos en rojo e hidrofílicos en azul. Para que estos puntos queden por encima de la estructura pintada al inicio se utiliza la propiedad *zorder*.

Por otro lado, se ha creado una función *represent_grad* que representa gráficamente la estructura 2D de la proteína coloreando de azul los aminoácidos hidrofílicos y de rojo los hidrofóbicos, pero con un gradiente que depende del valor ΔG de cada aminoácido. Para ello, se ha recorrido toda la estructura separando los puntos a representar en dos diccionarios dependiendo del tipo de aminoácido que sea, al igual que en la función *represent*, sin embargo, en este diccionario se crean 3 claves más, *minim*, *maxim* y *color*. Las dos primeras sirven para almacenar el valor mínimo y máximo del valor ΔG de cada tipo de aminoácido y, el color, se usa para almacenar todos los valores de ΔG . En el caso de los hidrofóbicos se almacena el valor ΔG aumentado en 3 unidades para que sean todos los valores positivos y en el de los hidrofílicos se aumenta en 8 unidades.

Tras esta separación se representa la estructura completa y, cómo en la función *represent*, se utiliza la propiedad *zorder* para mostrar los puntos con colores por encima de la estructura base. En este caso se representan los puntos hidrofóbicos en rojo e hidrofílicos en azul, pero con un gradiente que depende de los valores ΔG que se han ido almacenando en color. Para ello, se han normalizado estos valores para que estén entre 0 y 1 utilizando los valores máximos y mínimos del diccionario de

cada tipo de aminoácido. Si todos los aminoácidos de un tipo tienen el mismo valor, el color tomará valor 1.

Para encontrar los parámetros que mejor se adapten a cada problema se han realizado pruebas variando la temperatura

inicial, el factor de descenso, el número de enfrentamientos y el número de iteraciones. Con los mismos valores se pueden obtener diferentes valores de puntuación ya que se escogen los sucesores de forma aleatoria.

A continuación, se mostrarán los resultados obtenidos en cada proteína.

A. *Glucose-6-phosphatase*

MEEGMNVLHDFG				
Temperatura inicial	Factor de descenso	Número de enfrentamientos	Número de iteraciones	Puntuación
100	0.95	100	100	41.94
100	0.95	100	100	44.34
100	0.95	100	200	41.94
100	0.95	200	100	41.94
50	0.95	100	100	43.14
50	0.95	200	200	43.14
50	0.95	100	100	41.94
50	0.95	200	200	41.94

Tabla 1. Resultados Glucose-6-phosphatase

Tras realizar las pruebas, se ha concluido que los mejores valores de los parámetros para resolver esta proteína son:

- Temperatura inicial = 100
- Factor de descenso = 0.95
- Número de enfrentamientos = 100
- Número de iteraciones = 200

Con estos valores se ha obtenido la mejor puntuación y menor tiempo de ejecución.

A continuación, se muestra la gráfica del plegamiento obtenido a partir del algoritmo de enfriamiento simulado.

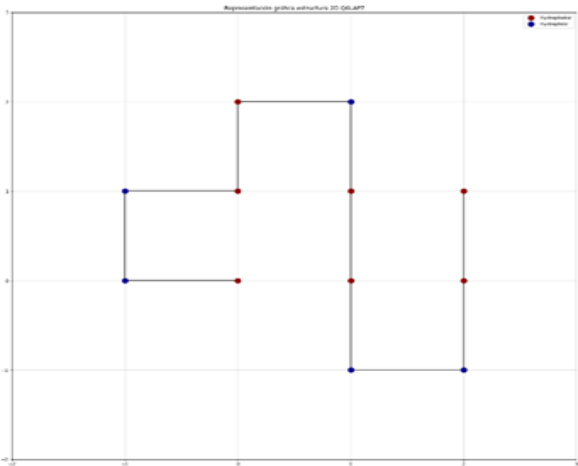


Ilustración 1. Glucose-6-phosphatase

Como se observa en la imagen, los aminoácidos polares o hidrofílicos que están representados de color azul se encuentran en el exterior y los apolares o hidrofóbicos se encuentran en el interior. Además, tiene una estructura compacta.

En la siguiente imagen se muestra la misma estructura, pero con un gradiente de color dependiendo del valor ΔG del correspondiente aminoácido.

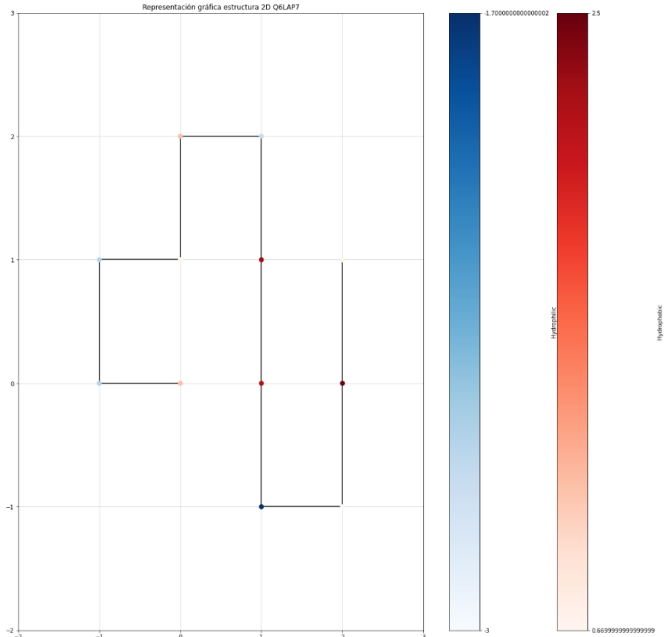


Ilustración 2. Glucose-6-phosphatase Gradiente

B. *Insulin*

MALWMRLLPLLALLALWGPDPAAAFVNQHLGSHLVEALYLVCGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQCCTSIQLENYCN				
Temperatura inicial	Factor de descenso	Número de enfrentamientos	Número de iteraciones	Puntuación
100	0.95	100	100	794,669999
100	0.95	100	100	886,34
100	0.95	100	200	757,37999
100	0.95	200	100	359,40999
50	0.95	100	100	900,54999
50	0.95	200	200	374,55
50	0.95	100	100	817,5899
50	0.95	200	200	618,45999
100	0.95	500	500	302.83
100	0.95	500	500	238.8
200	0.75	500	500	558.20
200	0.95	500	500	556.9

Tabla 2. Insulin

Tras realizar las pruebas, se ha concluido que los mejores valores de los parámetros para resolver esta proteína son:

- Temperatura inicial = 100
- Factor de descenso = 0.95
- Número de enfrentamientos = 500
- Número de iteraciones = 500

Con estos valores se han obtenidos las mejores puntuaciones con un tiempo de ejecución de aproximadamente de 2 minutos.

A partir del algoritmo de enfriamiento simulado implementado se obtiene el plegamiento de dicha proteína y se representa la gráfica obtenida.

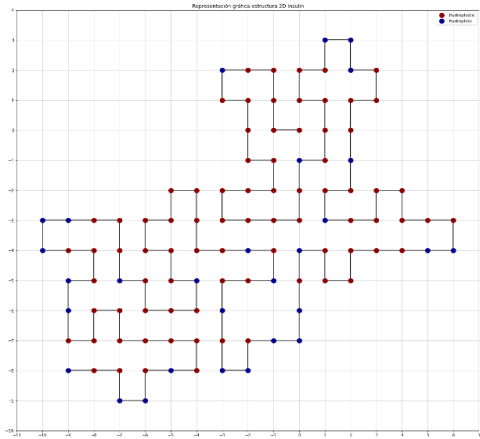


Ilustración 3. Insulin

En la imagen se puede comprobar que no todos los aminoácidos polares están en el exterior ni todos los apolares en el interior esto se debe a que no se ha obtenido el mínimo global a partir de este algoritmo, pero si tiene una estructura bastante compacta.

En la siguiente imagen se muestra la misma estructura, pero con un gradiente de color dependiendo del valor ΔG del correspondiente aminoácido.

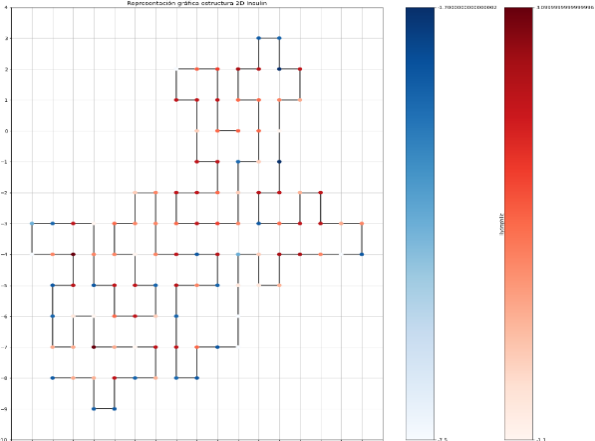


Ilustración 4. Insulin Gradiente

c) *Tat protein*

RRQRRSAPPSSSEHDHQNPSEQPSPQTRGNSTGSEESKKKVESKTET DPCD				
Temperatura inicial	Factor de descenso	Número de enfrentamientos	Número de iteraciones	Puntuación
100	0.95	100	100	71,05
100	0.95	100	100	44,16
100	0.95	100	200	52,944
100	0.95	200	100	18,3000
50	0.95	100	100	46,2200
50	0.95	200	200	11,57
50	0.95	100	100	36,200
50	0.95	200	200	-18,9

Tabla 3. Resultados Tat protein

Tras realizar las pruebas, se ha concluido que los mejores valores de los parámetros para resolver esta proteína son:

- Temperatura inicial = 50
- Factor de descenso = 0.95
- Número de enfrentamientos = 200
- Número de iteraciones = 200

Con estos valores se ha obtenido la mejor puntuación y menor tiempo de ejecución.

A continuación, se muestra la gráfica del plegamiento obtenido a partir del algoritmo de enfriamiento simulado.

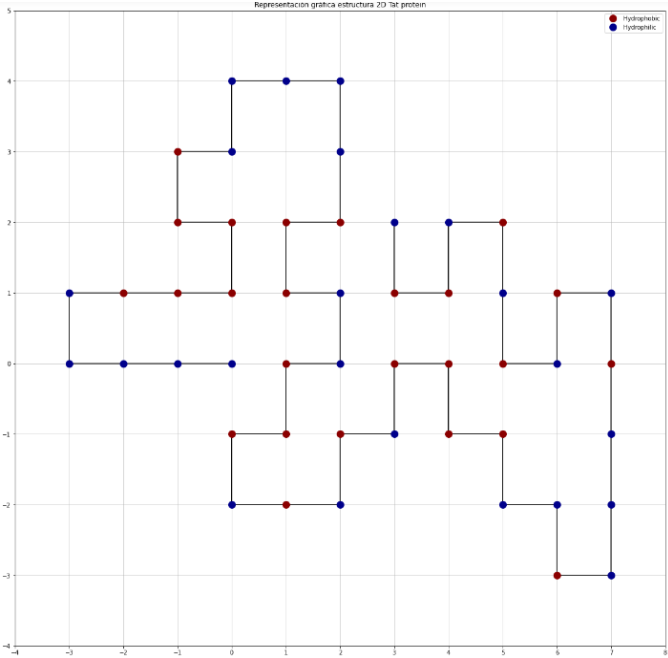


Ilustración 5. Tat protein

Al igual que en la proteína anterior se puede observar que no todos los aminoácidos polares están en el exterior ni todos los apolares en el interior esto es debido a que no se ha obtenido el mínimo global a partir de este algoritmo, pero si tiene una estructura bastante compacta.

En la siguiente imagen se muestra la misma estructura, pero con un gradiente de color dependiendo del valor ΔG del correspondiente aminoácido:

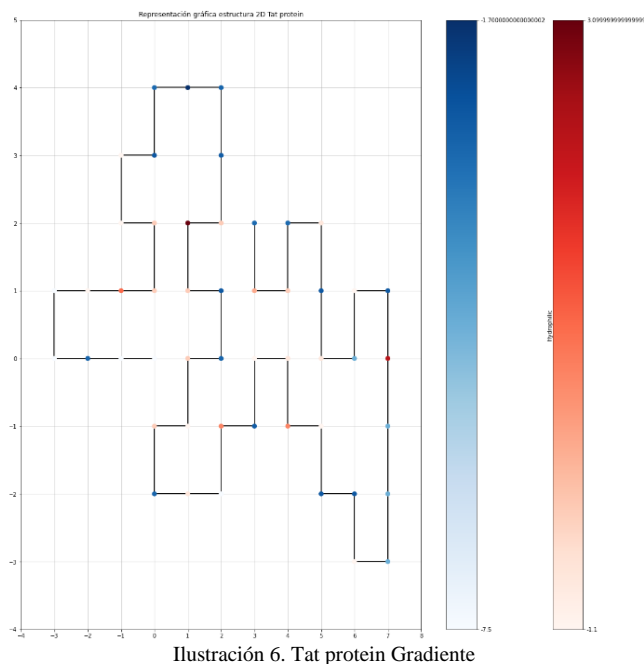


Ilustración 6. Tat protein Gradiente

VI. CONCLUSIONES

Una de las tareas más importantes para la biología molecular es resolver el problema de plegamiento de proteínas. En este proyecto hemos implementado un algoritmo de optimización, en este caso, el algoritmo de enfriamiento simulado, para intentar resolver dicho problema. Para ello, hemos tenido que investigar su funcionamiento para poder adaptarlo a dicho problema y finalmente implementarlo.

Durante la realización del proyecto, hemos llegado a comprender la teoría básica de las proteínas y de sus plegamientos, además, hemos observado que algo que tarda segundos en formarse, puede ser algo muy costoso de calcular, como es el plegamiento de proteínas.

Una vez investigado e implementado todo lo necesario para aplicar el algoritmo, realizamos varios experimentos para comprobar que todo funcionaba correctamente. Para obtener las secuencias de proteínas necesarias y sus descripciones utilizamos UNIPROT. La solución obtenida para cada una de ellas han tenido

resultados satisfactorias, desde nuestro punto de vista. Sin embargo, no han podido encontrarse los valores óptimos.

Además, durante el desarrollo y la experimentación del proyecto, hemos ido marcando con un temporizador el tiempo que tardaba cada prueba en ejecutarse para ir mejorando la eficiencia del algoritmo.

Como conclusión, podemos afirmar que, con lo aprendido con el estudio en profundidad del problema de plegamiento de proteínas, este se trata de un problema muy complejo de resolver pero con el Algoritmo de Enfriamiento Simulado se pueden obtener soluciones de buena calidad de una manera eficiente. Sin embargo, la parte más compleja de este algoritmo es encontrar el conjunto de parámetros que optimicen los resultados de cada proteína.

Para futuros trabajos, se propone investigar otros algoritmos de optimización, para poder resolver de manera más eficiente el problema de plegamientos de proteínas, utilizar otro tipo de métodos que reduzcan el tiempo de ejecución y comprobar los métodos con proteínas más grandes y complejas.

REFERENCIAS

- Jumper, J., Evans, R., Pritzel, A. et al. Highly accurate protein structure prediction with AlphaFold. Nature (2021). <https://doi.org/10.1038/s41586-021-03819-2>
- Shakhnovich & Gutin et al. Engineering of stable and fast-folding sequences of model proteins. PNAS (1993). <https://doi.org/10.1073/pnas.90.15.7195>
- Dill et al. Principles of protein folding: A perspective from simple exact models. Protein Science (1995) <https://doi.org/10.1002/pro.5560040401>
- Moreno-Hernandez & Levitt. Comparative modeling and protein-like features of hydrophobic-polar models on a two-dimensional lattice. Proteins. (2012) <https://doi.org/10.1002/prot.24067>
- Berg, J. M., Tymoczko, J. L., Stryer, L., & Stryer, L. Biochemistry. New York: W.H. Freeman. (2002).
- Kapoor, A. Hands-On Artificial Intelligence for IoT : Expert Machine Learning and Deep Learning Techniques for Developing Smarter IoT Systems, Packt Publishing, Limited, (2019). <https://ebookcentral.proquest.com/lib/uses/detail.action?docID=5675583>
- Kochenderfer, & Wheeler, T. A. Algorithms for optimization. The MIT Press. (2019) ISBN 9780262039420.
- KALIVAS, J.H. Adaption of simulated annealing to chemical optimization problems. New York: Elsevier. (1995) ISBN 1-281-05813-0.
- Dowsland, K. A., & Díaz, B. A. (2003). Diseño de heurística y fundamentos del recocido simulado. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 7(19), 0.
- MONTERO, M. R. (2009). Métodos basados en optimización por colonia de hormigas aplicados al modelo hidrofóbico-polar del plegamiento de proteínas. <http://inaoe.repositorioinstitucional.mx/jspui/handle/1009/446>
- Fersht, A. R. (2021). AlphaFold—A Personal Perspective on the Impact of Machine Learning. Journal of Molecular Biology, 433(20), 167088.
- Alas-Guardado, S. D. J., Rojo, A., & Merino, G. (2011). La paradoja de Levinthal: cuando una contradicción se vuelve lógica. Educación química, 22(1), 51-54.