Enfriamiento Simulado para plegamiento de proteínas 2D.

***Resumen*— El objetivo principal del trabajo consiste en implementar un sencillo algoritmo de plegamiento de proteínas en 2D en base a la hidrofobicidad de los aminoácidos. Para ello implementaremos un algoritmo de Enfriamiento Simulado (Simulated Annealing) y de esa manera dar una solución a los problemas del plegamiento de las proteínas.**

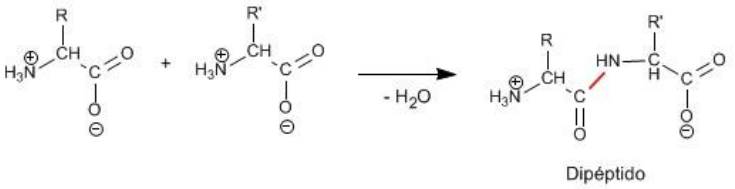
A lo largo de este documento se recogen los pasos y la información sobre cómo hemos generado un sistema de aprendizaje inteligente que en base al algoritmo de enfriamiento simulado es capaz de concluir resultados satisfactorios y cercanos al objetivo final que se busca.

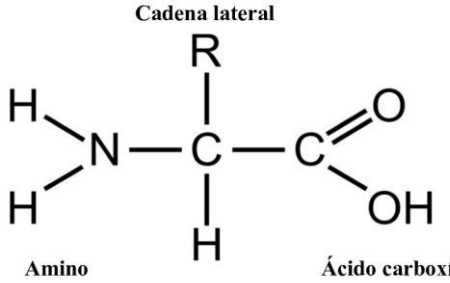
I. Introducción

Para situarnos en un enfoque general del problema al que nos vamos a centrar es importante realizar una pequeña introducción acerca del mismo.

Las proteínas son grandes moléculas que se encuentran en todos los seres vivos y que contribuyen a la vida de estos organismos de manera fundamental. Estas macromoléculas, a su vez, se encuentran formadas por sub-estructuras denominadas aminoácidos (AA) que se unen formando largas cadenas, es aquí donde se centra nuestro estudio.

Los aminoácidos de una proteína se unen unos con otros a través de un enlace peptídico (-CO-NH-), que consiste en la unión del grupo ácido por parte del AA1 y del grupo amino del AA2 desprendiéndose en la formación de este enlace una molécula de agua (H2O). Los grupos moleculares que forman los aminoácidos y que no intervienen en los enlaces peptídicos están dotados de una cierta movilidad que les permite girar (en sentido o no de las agujas del reloj) para formar otros enlaces, conocidos como puentes de hidrógeno, con el resto de grupos libres de aminoácidos de la cadena, dando lugar a plegamientos en la estructura. Nótese llegados a este punto la importancia de la hidrofobicidad de cada una de estas unidades funcionales para la formación de los enlaces.





Predecir la estructura tridimensional que adoptará una proteína basándose únicamente en su secuencia de aminoácidos (el componente de predicción de la estructura del "problema del plegamiento de proteínas") ha sido un importante problema abierto de investigación durante más de 50 años. [1]

La estructura de una proteína determina la función de la misma, y es este detalle el que supone un reto para muchos campos de estudio en la actualidad (biomedicina, fisiología molecular, biología informática, biotecnología…), ya que el poder predecir o estudiar la estructura de las proteínas supone una gran ventaja.

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 es más largo que la edad del universo conocido.

Para entender el papel fundamental que desempeña el algoritmo de Enfriamiento Simulado (Simulated Annealing) en la construcción o implementación de un sistema de IA capaz de predecir la estructura final de una proteína, lo primero que debemos conocer es el esquema general que sigue este algoritmo, así como los pasos que nos llevan a desarrollarlo.

Este algoritmo se basa en el enfriamiento que se utiliza con algunos metales y cerámicas. Esta técnica consiste en calentar un material para posteriormente dejarlo enfriar de manera lenta

y alterar sus propiedades físicas con el objetivo de manipular la estructura de dicho material.

A grandes rasgos, posteriormente nos centraremos más en la explicación de este algoritmo, se parte de un estado inicial o solución inicial a partir de la cual se hace una exploración de de los diferentes vecinos a la solución inicial, y por cada iteración se decide probabilísticamente si emigrar a un nuevo estado/solución o mantener el actual.

En la elaboración de este trabajo, se ha seguido el patrón general que implementa el algoritmo de Enfriamiento Simulado para extrapolar su funcionalidad a la predicción de la estructura proteica en 2D.

II. FUNCIONAMIENTO DE LOS ALGORITMOS DE PLEGAMIENTO DE PROTEÍNAS 2D Y CÓMO CONTRIBUYEN A LOS ALGORITMOS ACTUALES (ALPHAFOLD2)

AlphaFold2 desarrollada en 2020 es un programa de inteligencia artificial desarrollado por DeepMind de Alphabets/Google que realiza predicciones de la estructura de las proteínas mediante Deep Learning (Aprendizaje profundo).[2]

Anteriormente, y como fruto de su desarrollo vino la aparición del actual algoritmo, la empresa DeepMind lanzó el proyecto AlphaFold1, cuya implementación se basaba en el trabajo realizado por varios equipos en 2010.

De esta forma, y tal y como definió Mirko Torrisi[3]:

El elemento central de AlphaFold es un predictor de mapa de distancia implementado como redes neuronales residuales muy profundas con 220 bloques residuales que procesan una representación de dimensionalidad 64 × 64 × 128, correspondiente a las características de entrada calculadas a partir de dos fragmentos de 64 aminoácidos. Cada bloque residual tiene tres capas, incluida una capa convolucional dilatada de 3 × 3; los bloques pasan por la dilatación de los valores 1, 2, 4 y 8. En total, el modelo tiene 21 millones de parámetros. La red utiliza una combinación de entradas 1D y 2D, incluidos perfiles evolutivos de diferentes fuentes y características de coevolución. Junto con un mapa de distancias en forma de histograma de distancias muy detallado, AlphaFold predice ángulos Φ y Ψpara cada residuo que se utilizan para crear la estructura 3D prevista inicial. Los autores de AlphaFold concluyeron que la profundidad del modelo, su gran tamaño de cultivo, el gran conjunto de entrenamiento de aproximadamente 29,000 proteínas, las técnicas modernas de Deep Learning y la riqueza de información del histograma predicho de distancias ayudaron a AlphaFold a lograr una alta precisión de predicción de mapas de contacto.

A. Explicación y gráfica de cada uno de los plegamientos obtenidos

Para la explicación de las diferentes estructuras que componen este algoritmo vamos a hacerlo de manera deductiva, es decir, vamos a ir de lo general a lo más específico.

```
función SA(problema, esquema) ret (estado_solución)
    act:= ESTADO_INICIAL(problema);
    T:= temperatura_inicial(esquema); tope:=num_iteraciones(esquema); t:=0;
    repetir
        est:= act; it:=0;
        repetir
            sig:= sucesor_aleatorio (est);
            ΔE := valor(sig) – valor(est);
            si ΔE<0 entonces est:=sig;
            sino    q:= min{1, e^(-ΔE/T)};
                    si aleatorio(0,1)<q entonces est:=sig;
            it:= it+1;
        hasta it=tope;
        act:= est; T:=ENFRIAR(T, esquema); t:=t+1;
    hasta T≅0
    ret act;
```

[4]

Como puede apreciarse en la imagen (número imagen), la función SA (Simulated Annealing) recibe unos parámetros **problema** y **esquema**. El primero hace referencia al input del que partiremos para generar nuestra solución, en nuestro caso, la secuencia de aminoácidos que conforman la proteína; el segundo define una ruta en base a la cual configuraremos las variables que hagan iterar nuestro algoritmo..

La variable **'act'** (actual) se inicializa con la función **'ESTADO_INICIAL'**, que a partir del input problema, genera una posible solución inicial.

La variable **'T'** toma una temperatura inicial siguiendo el esquema que se pasa como parámetro.

La variable **'tope'** toma un valor numérico entero para almacenar el número de iteraciones para el bucle externo y nos ayuda a fijar la condición de equilibrio.

La variable se inicializa como contador **'t'** a 0.

A continuación se abre el bucle externo, que tiene como condición de parada como 'T' aproximadamente o igual a 0, éste actúa como un esquema de descenso de temperatura, enfriamiento, que garantiza la convergencia a óptimos globales, con las siguientes variables:

La variable **'est'**, que toma el estado o solución de 'act'.

La variable **'it'**, que viene a ser un contador para el bucle interno.

Seguidamente se abre el bucle interno, cuya condición de parada se cumplirá cuando la variable 'it' tome el valor de 'tope'. Este bucle nos permite obtener nuevas configuraciones o soluciones a partir de la actual, siguiendo un esquema de exploración del espacio de configuraciones[4]. Aquí se asignan las siguientes variables y funciones:

Se crea una variable sig que toma una solución a partir de la función **'sucesor_aleatorio'** que recibe como entrada a **'est'**.

Se crea una variable **'ΔE'** que calcula la diferencia entre las funciones de coste **'valor'** entre sig y est.

Acto seguido, aún dentro del bucle interno se declara un **si/sino** (if/else), donde se chequea si 'ΔE' es menor que 0, en cuyo caso 'est' toma valor de 'sig'; en otro caso:

Se crea una nueva variable **'q'** que toma el mínimo valor entre 1 y e^(-ΔE/T). Es dentro de este 'sino' (else) donde además se declara un nuevo si (if) para evaluar si el número generado aleatoriamente entre 0 y 1 es menor que 'q', en cuyo caso se asigna a 'sig' el valor de 'est'.

Una vez cerrado el bucle interno y antes de cerrar el externo, se realizan una serie de asignaciones:

La variable 'act' toma el valor de 'est'.

La variable 'T' recibe el valor de la función **'ENFRIAR'**.

El contador 't' se incrementa en 1.

Por último, queda devolver el valor que almacena 'act'.

La reducción de la probabilidad de aceptar malas soluciones a medida que avanzamos en la exploración del espacio de configuraciones supone una facilidad para hallar valores óptimos.

De esta forma y para implementar nuestro algoritmo, se hemos implementado una serie de funciones auxiliares propuestas y que se detallan a continuación:

aa_deltaG Diccionario

formado por:

- Clave:

Aminoácido. (A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y)

- Valor:

Valor de energía libre para cada aminoácido. (Ej: -2.6)

get_spatial_dic(protein, structure)

- ➢ entrada:

  protein = secuencia de aminoácidos (Ej: PEPTIDE)
  structure = secuenciación de la orientación de los aminoácidos de una proteína donde cada aminoácido toma dicha orientación con respecto al anterior en la cadena de la secuencia, a excepción del primero que toma el valor 'I' de inicial. (N,S,E,W).

- ➢ salida: diccionario o un diccionario vacío si existen solapamientos. Las claves de dicho diccionario serán tuplas de dos números enteros representando coordenadas espaciales y los valores serán letras de aminoácidos.

is_hydrophobic(aa)

- ➢ entrada:

  aa = aminoácido representado por su letra correspondiente en el diccionario aa_deltaG.

- ➢ salida : Devuelve verdadero si el aminoácido evaluado es hidrofóbico; falso en caso contrario.

get_score(dic)

- ➢ entrada:

  dic = diccionario representando la estructura espacial de una proteína.

- ➢ salida: devuelve su puntuación. La puntuación de un aminoácido será ΔG * N (si el aminoácido no es hidrofóbico) y ΔG * N + 10 * N (si el aminoácido es hidrofóbico). Siendo N el número de posiciones adyacentes libres.

fold(structure, pos, angle)

- ➢ entrada:

  structure = secuenciación de la orientación de los aminoácidos de una proteína donde cada aminoácido toma dicha orientación con respecto al anterior en la cadena de la secuencia, a excepción del primero que toma el valor 'I' de inicial. (N,S,E,W).

  pos = posición a partir de la que se comienza a plegar.

  angle = ángulo de plegado (90º,-90º).

- ➢ salida: secuenciación de aminoácidos ya plegados.

get_successors(protein,structure)

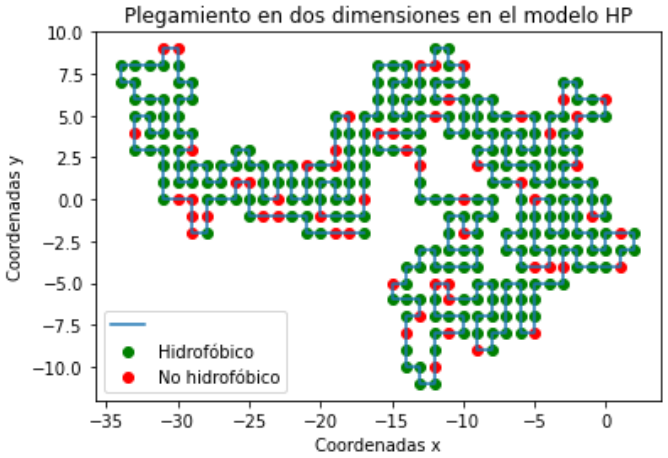
- ➢ entrada:

  protein = secuencia de aminoácidos (Ej: PEPTIDE)
  structure = secuenciación de la orientación de los aminoácidos de una proteína donde cada aminoácido toma dicha orientación con respecto al anterior en la cadena de la secuencia, a excepción del primero que toma el valor 'I' de inicial. (N,S,E,W).

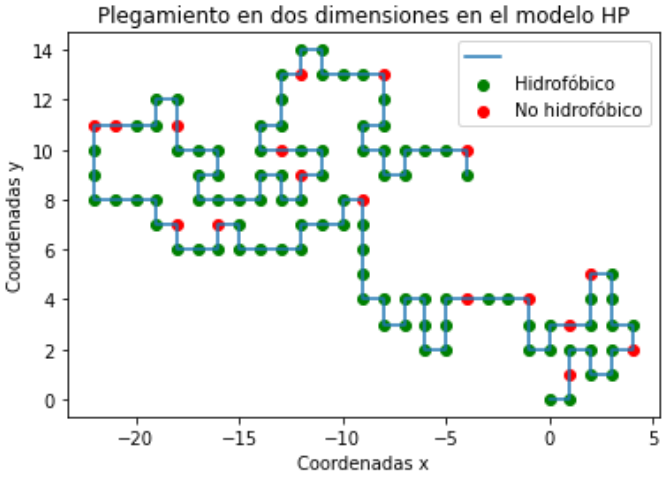
- ➢ salida: diccionario cuyas claves son todas las posibles estructuras válidas tras aplicar todos los posibles plegamientos y cuyos valores sean los correspondientes diccionarios espaciales obtenidos con get_spatial_dic.

A continuación, se muestran las gráficas obtenidas después de aplicar a las proteínas: Q8NHC7, A0A5P8I0X0, A1L190 y P01308, (extraídas de la base da datos UNIPROT) el algoritmo SA que hemos implementado para el plegamiento de proteínas 2D.

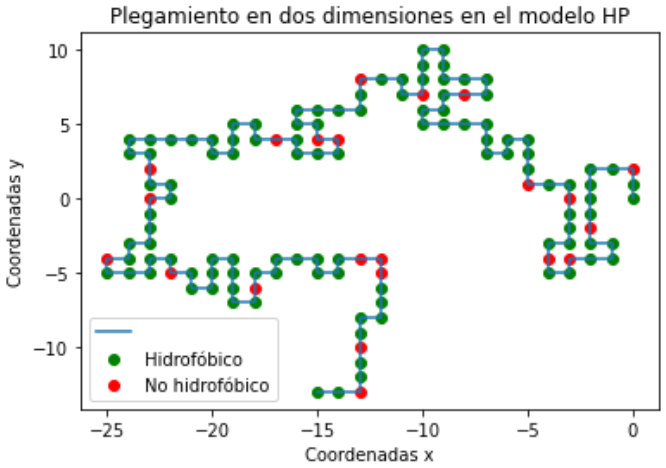
● Q8NHC7



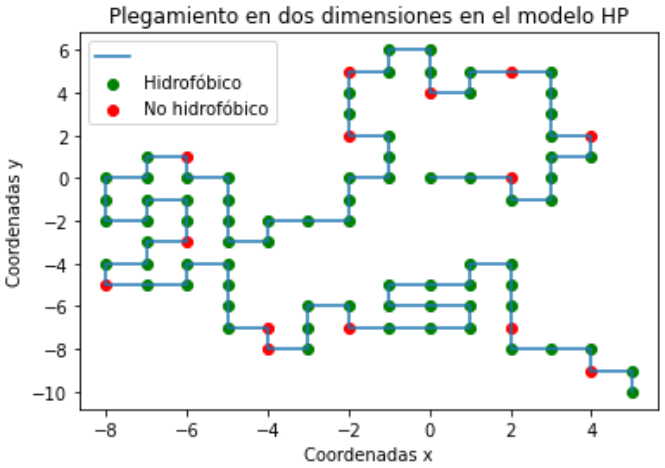
● A1L190



● P01308



● A0A5P8I0X0



B. Trabajos Relacionados

En 1993 Shakhnovich & Gutin realizaron un artículo de información llamado **"Engineering of stable and fast-folding sequences of model proteins"**.

En 1995 se publicó **"Principles of protein folding: A perspective from simple exact models. Protein Science"** por Kenn A. Dill.

Varios años despues, en 2021 se publicó **"Highly accurate protein structure prediction with AlphaFold. Nature"** por Jumper, J., Evans, R., Pritzel, A.

III. Resultados

Para nuestro proyecto, hemos utilizado la secuencia de aminoácidos de las proteínas: Q8NHC7, A0A5P8I0X0, A1L190 y P01308. Todas ellas extraídas de UNIPROT.

Resultados experimentales del algoritmo SA

Proteína	*Puntuación*	*Estructura*
Q8NHC7	1656.5700000000004	IESWWSSWNWSWNWNNWSSWSEESEEEESWSSWWNENWWSSSWSEENESENENNEEESENENNWSWWWNEENWWNEENENWNNESESSSWSEESSEEESWWWSEEESESWWNWWSESSWSEENNESSEEESWSWWWNWSSWNWNWNENNNWSSWWNENWNWSSSSSWWWSSSWNNNWNNENWWSSWNNWSWSESEESWWNWSSEEEEEEEEESSSWWNENWWWWWSEEESSEESWWSEESWWSWSESESWSESWWNWSWWNWSWWSSENENNEEENWWNEENWWNNWNEENNNNENNNN

Resultados experimentales del algoritmo SA

Proteína	*Puntuación*	*Estructura*
A0A5P8I0X0	1685.0500000000006	INNWWSSSSSESWWSWNNENNNNWWNNNWSWNNWWWNENEENWWNNWSSSWNWWSSWWWSESESWWNWWNWSSWNWWWWSESSESWSSSWSWSEENESESSENNESSSENNENEESENEESSSSWSSSSSWW
A1L190	987.5099999999998	IEESENNENWNNNWWSWNNWSWSSSESSWSSWWSWNNNWNWSWSSENESSWSWSEENESSSESENNESEEENWWNEENESSSSEESES
P01308	1455.0999999999997	IENNESENENWNNWSSWWSWNNWWWWSSWNNWSWNWNNNNWSWWSWWWNWSWWNWNWWWNNNEEENESSEESWSEEENESENENWWWNENNENESEEESSWSESENEEES

IV. Conclusiones

Una vez realizado las distintas pruebas experimentales, bajo nuestra propia implementación del algoritmo de enfriamiento simulado, sobre las estructuras de algunas proteínas, y habiendo obtenido resultados satisfactorios; podemos concluir que:

Hemos conseguido el objetivo principal de este trabajo al implementar el algoritmo SA para el plegamiento de proteínas 2D.

El algoritmo de plegamiento de proteínas 2D supone una de las bases para el posterior desarrollo de estructuras y mecanismos de generación de proteínas 3D, en concreto para los desarrollados por DeepMind en AlphaFold2.

Se puede demostrar que si el esquema de la temperatura, para el algoritmo de enfriamiento simulado, disminuye T lo bastante despacio, el algoritmo encuentra el óptimo global con probabilidad cercana a 1.

.

V. Bibliografía

Referencias

[1] https://www.nature.com/articles/s41586-021-03819-2

[2] https://es.wikipedia.org/wiki/AlphaFold

[3] Torrisi, Mirko y col. (22 de enero de 2020), Métodos de aprendizaje profundo en la predicción de la estructura de proteínas . Revista de Biotecnología Computacional y Estructural vol. 18 1301-1310. doi : 10.1016 / j.csbj.2019.12.011 (CC-BY-4.0)

[4] https://bmcbioinformatics.biomedcentral.com/articles/10.1 186/1471-2105-6-30

[5] https://www.cs.upc.edu/~mabad/IA/SIMULATED%20ANNEALING.pdf

[6] https://hmong.es/wiki/AlphaFold