

UNIVERSIDAD DE ZARAGOZA

FACULTAD DE CIENCIAS

DEPARTAMENTO DE FÍSICA TEÓRICA  
TRABAJO DE FIN DE GRADO

---

**Estudio del comportamiento de las proteínas  
modulares en función del número de módulos repetidos**

---

*Author:*  
Laura Clara JOVER GALTIER

*Supervisor:*  
Dr. Pierpaolo BRUSCOLINI



# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. El plegamiento de las proteínas . . . . .	1
1.2. Proteínas modulares diseñadas . . . . .	2
1.3. Modelado teórico . . . . .	2
1.4. Objetivos . . . . .	4
<b>2. Métodos</b>	<b>4</b>
2.1. Un modelo de mecánica estadística para el plegamiento de proteínas . . . . .	4
2.2. Diseño de las proteínas artificiales . . . . .	5
2.3. Caracterización termodinámica de las proteínas . . . . .	6
<b>3. Resultados</b>	<b>9</b>
3.1. Estudio del aumento del número de módulos centrales . . . . .	9
3.2. Partes iniciales y finales . . . . .	12
3.3. Alargamiento de los bucles . . . . .	16
<b>4. Conclusiones</b>	<b>22</b>
<b>5. Códigos</b>	<b>26</b>
5.1. modificar_proteina . . . . .	26
5.2. alargar_bucles . . . . .	34
5.3. pesos_contactos_nn . . . . .	42
5.4. k_van't_hoff . . . . .	45



# 1. Introducción

## 1.1. El plegamiento de las proteínas

A pesar de las décadas dedicadas a su estudio, el plegamiento de proteínas sigue proponiendo retos a la investigación, debido también a la variedad de proteínas diferentes por estructura y comportamiento[9]. Las proteínas son polímeros lineales, formados por la unión con enlaces peptídicos de un número muy variable de aminoácidos (desde pocas decenas a varios centenares). Los 20 tipos de aminoácidos diferentes, que se distinguen por su cadena lateral, permiten un número asombroso de secuencias diferentes, de las cuales sólo una pequeña parte existe en realidad en la naturaleza. Las proteínas desarrollan un gran número de funciones diferentes en la célula; sin embargo, cada una ejerce su función exclusivamente cuando asume una estructura compacta que la caracteriza, el llamado “estado nativo”. Esta conformación está codificada en la secuencia, ya que las proteínas se pueden desplegar y volver a plegar en el estado nativo *in vitro*, en disolución acuosa, lejos de la acción de la maquinaria celular, sencillamente cambiando la temperatura o añadiendo agentes caotrópicos, como la urea.

Esta observación sugiere que el estado nativo, que en muchos casos es suficientemente rígido como para poderlo organizar en cristales y estudiarlo por difracción de rayos X, corresponde a un macroestado termodinámicamente estable: la desestabilización del estado nativo y el desplegamiento de la proteína es el equivalente, para un sistema de tamaño finito, de una transición de fase; la cinética de relajación después de un salto en temperatura corresponde a la relajación de un sistema físico al equilibrio termodinámico.

La naturaleza de la transición y de la cinética (por ejemplo, la presencia o ausencia de estados metaestables de equilibrio o dinámicos, la rapidez de la relajación, la robustez frente a mutaciones o a perturbaciones del entorno químico) dependen no sólo de la secuencia sino también de la estructura nativa, y no cualquier conformación polimérica vale como estructura nativa de una proteína.

Por ejemplo, es característica común de las estructuras nativas la presencia de patrones estructurales (“estructura secundaria”) estabilizados por una red de enlaces de hidrógeno, con forma de hélice ( $\alpha$ -hélice) o estructuras elongadas (láminas  $\beta$ ), que se ensamblan jerárquicamente en una estructura global (“terciaria”) que representa la forma de la proteína. Las proteínas más comunes, y las más estudiadas, son proteínas globulares cuyo estado nativo se parece a un ovillo más o menos regular. Sin embargo, nuestro interés en este trabajo se centra en las proteínas modulares, que presentan una mayor regularidad, que las hace objeto ideal para estudios de modelado teórico.

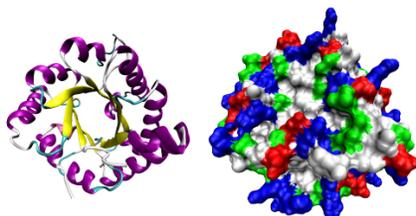


Figura 1: *Izquierda*: Estructura de una proteína globular natural en la que se muestran las hélices  $\alpha$  (hélices moradas) y las láminas  $\beta$  (flechas amarillas). *Derecha*: Representación de todos los átomos que forman esa estructura globular. Fuente: Wikipedia.

## 1.2. Proteínas modulares diseñadas

Las proteínas modulares están constituidas por repeticiones de la misma estructura mínima, típicamente de longitud 20÷40 aminoácidos, y formada por dos  $\alpha$ -hélices, dos láminas  $\beta$  o una combinación  $\alpha/\beta$ , organizadas en una estructura alargada, sustancialmente lineal (figura 2). Estas estructuras están caracterizadas por interacciones locales (entre aminoácidos no muy distantes a lo largo de la cadena), a diferencia de lo que pasa en las proteínas globulares, de forma más esférica, donde se pueden dar contactos entre puntos a cualquier distancia a lo largo de la cadena. En la naturaleza, las proteínas modulares desarrollan varias funciones, mediando la interacción entre proteínas, participando en la regulación de la transcripción y en el transporte de proteínas. Las proteínas modulares naturales tienen estructura regular, pero secuencia variable; en cambio, ha sido posible diseñar y sintetizar proteínas artificiales donde una misma secuencia se repite en cada módulo estructural (posiblemente, con excepción de los módulos extremos). La clase de proteínas que estudiamos en este trabajo pertenece a una clase de proteínas modulares diseñada: la CTPR (“Consensus Tetratricopeptide”)[14], donde el módulo está constituido por un patrón  $\alpha - \alpha$  de 34 aminoácidos. Actualmente hay dos estructuras cristalizadas, la 1NA3, con 5 hélices (2 módulos más una hélice final que estabiliza la estructura en solución[14]) y la 1NA0, con 7 hélices. Sin embargo, se han estudiado y caracterizado experimentalmente (aunque sin cristalizarlas) la termodinámica y la cinética de varias proteínas artificiales CTPR de hasta 41 hélices (CTPR20)[8], sobre todo a través de técnicas espectroscópicas y con desnaturalización química[], pero también con técnicas calorimétricas[8] y de intercambio de hidrógeno[7][15].

## 1.3. Modelado teórico

El estudio del plegamiento de proteínas se ha beneficiado enormemente de la evolución de los ordenadores, cuya potencia de cálculo hace ahora posible la simulación del plegamiento de proteínas pequeñas, con dinámica molecular con detalle atómico y agua explícita[13]. Sin embargo, el estudio del equilibrio, para el cual se necesita un muestreo significativo del espacio de conformaciones, y

la necesidad de tratar proteínas más grandes en tiempos de cálculo razonablemente cortos, hace interesante el uso de modelos simplificados (“de grano grueso”), donde un conjunto de átomos, un aminoácido o estructuras más grandes aún se consideran como variables fundamentales del modelo, con una importante reducción de los grados de libertad. Por ejemplo, las proteínas modulares se han estudiado también con el modelo de Ising, considerando una hélice[7] o un módulo entero [12] como “spin”, ajustando los parámetros de interacción y el campo ( $J_{i,i+1}$  y  $h_i$  en su escritura tradicional) a los resultados experimentales. Sin embargo, la identificación de las variables elementales de espín con un fragmento de la estructura secundaria dificulta la posibilidad de investigar el papel de los contactos entre los residuos individuales, y de estudiar el origen de la cooperatividad y la cinética de multiestados en la escala de los residuos.

Otra clase de modelos es la que reconoce el papel esencial de la estructura nativa imponiéndola como estructura de mínima energía. Este tipo de modelos, llamado de Go por el autor que los propuso[19], precisan el conocimiento de la estructura nativa como requisito previo y, por lo tanto, no sirven para predecirla, sino sólo para describir la termodinámica y la cinética del plegamiento, bajo la hipótesis fundamental de que las proteínas son “diseñadas” evolutivamente para tener muy poca frustración energética. Con esta hipótesis, los modelos de Go sencillamente premian energéticamente los contactos entre aminoácidos presentes también en la estructura nativa, e ignoran todos los demás, así que su paisaje de energía es totalmente “cuesta-abajo” hacia el estado nativo, y sólo la entropía involucrada en el tamaño del espacio de configuraciones puede crear barreras al plegamiento. Estos modelos a menudo tienen en cuenta la estructura nativa en su hamiltoniano, a través del “mapa de contacto” de la proteína, una representación bidimensional de la estructura nativa: se trata de la matriz  $\Delta$ , tal que  $\Delta_{i,j} = 1$  si los aminoácidos  $i,j$  están “en contacto” en la estructura nativa (con una oportuna definición de contacto, basado en su distancia), y  $\Delta_{i,j} = 0$  si no lo están. El modelo que usaremos en este estudio pertenece a esta última clase de modelos: se trata del modelo WSME, que trataremos con más detalle en el apartado de los métodos.

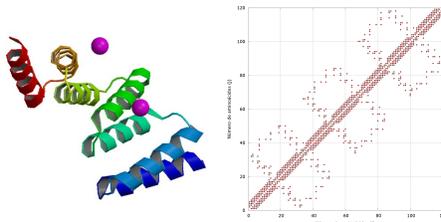


Figura 2: *Izquierda*: Imagen esquemática de la proteína estudiada (1NA0), donde se puede ver la estructura secundaria, los módulos formados por parejas de hélices, los bucles entre las hélices y la hélice final en rojo. Fuente: Protein Data Bank *Derecha*: Mapa de contacto de la proteína CTPRa3 (1NA0). Las  $\alpha$ -hélices, caracterizadas por contactos  $i, i + 3$ , se presentan como segmentos más “anchos” a lo largo de la diagonal; las regiones de contactos perpendiculares a la diagonal corresponden a uniones entre las hélices  $j, j + 1$ ; las regiones paralelas a la diagonal representan contactos entre las hélices  $j, j + 2$ .

## 1.4. Objetivos

Aplicando el modelo WSME a proteínas modulares diseñadas (CPTR), nos planteamos averiguar cuál será el comportamiento de estas proteínas si introducimos modificaciones en su estructura.

En primer lugar, queremos comprobar qué sucede si aumentamos el número de módulos centrales de las proteínas, es decir, si modificamos su tamaño, prestando atención a los cambios en las distintas propiedades que vamos a estudiar.

A continuación, estudiamos la función de las partes inicial y final y su influencia en el comportamiento de la proteína. Se sabe, por ejemplo, que la última hélice se introduce en el diseño para aumentar la solubilidad de la proteína y, dado que las proteínas TRP naturales tienen esa hélice extra, nos preguntamos si también puede tener efectos sobre la estabilidad. También nos planteamos si las modificaciones producidas por el aumento del tamaño de la proteína son más relevante que las que se producen por la presencia de estas partes iniciales y finales.

Por último, estudiaremos el efecto sobre la proteína de introducir aminoácidos en las regiones entre las hélices. Es importante destacar que la variación de estabilidad por el alargamiento de las regiones interhélice tiene importantes implicaciones biotecnológicas: las proteínas modulares se estudian también como posible herramienta para el desarrollo de moléculas bioactivas y para la construcción de nanomateriales, siendo los bucles la parte que más fácilmente se puede manipular en tamaño y funcionalizar.

## 2. Métodos

### 2.1. Un modelo de mecánica estadística para el plegamiento de proteínas

Para nuestro estudio de las proteínas modulares utilizamos el modelo introducido por Wako y Saito [20, 21], y redescubierto independientemente dos décadas después por Muñoz and Eaton [17, 16]. Se trata de un modelo fenomenológico y de tipo Go: en la función energía del modelo se considera conocida la estructura nativa de la proteína para describir su equilibrio y cinética. El estado de cada residuo  $i$  se describe mediante una variable binaria  $m_i = 0, 1$ , representando los estados desplegado y nativo respectivamente. Dado que el estado final permite un número mucho mayor de procesos microscópicos que el inicial, cada residuo  $k$  tiene un coste entrópico  $q_k$ .

Formalmente, el modelo difiere del modelo de Ising en que las interacciones no se limitan a los vecinos próximos, sino que se extienden a cualquier distancia, pero siempre en el interior de una región completamente nativa. El equilibrio del modelo puede ser calculado de manera exacta [20, 21, 2, 18], de manera que las energías, energías libres y fracciones de residuos nativos pueden ser fácilmente estudiadas.

El modelo viene descrito por el Hamiltoniano efectivo

$$H = - \sum_{i=1}^{N-1} \sum_{j=i+1}^N \epsilon_{i,j} \Delta_{i,j} \prod_{k=i}^j m_k + T \sum_{k=1}^N q_k m_k$$

donde  $N$  es el número de residuos de la proteína y  $T$  la temperatura absoluta. El producto  $\prod_{k=i}^j m_k$  toma valor 1 únicamente si todas las cadenas de residuos desde  $i$  hasta  $j$  se encuentran en el estado nativo. Las interacciones no nativas son despreciadas, mientras que las nativas son incluidas en la matriz de contacto  $\Delta_{ij}$ , que da información del número de contactos entre los átomos de residuos no consecutivos  $i$  y  $j$  en la estructura nativa, definidos de acuerdo con un criterio de distancia oportuno.

El modelo se ha aplicado para describir el plegamiento de gran variedad de proteínas [22, 5, 11, 1, 6, 3, 4]

A pesar de que se han introducido variantes más “realistas“ del modelo WSME, con parámetros de interacción dependientes de la temperatura y relacionados con una descripción más rigurosa de la superficie expuesta al disolvente en el proceso de desplegamiento, para este estudio adoptamos la formulación más sencilla del modelo, con  $\epsilon_{ij} = \epsilon$ ,  $q_i = q$ , y con un mapa de contacto donde todas las interacciones tienen el mismo peso. La razón es que queremos sacar conclusiones generales, que, aunque en buena medida cualitativas, serán robustas frente a posibles diferencias en la elección de los parámetros.

Con el mismo espíritu, fijamos el parámetro entrópico a un valor obtenido en estudios previos con otras proteínas modulares  $q = 0,00938$  kJ/K/mol, y ajustamos el parámetro  $\epsilon$  para que el pico del calor específico del modelo coincida con el correspondiente experimental ( $T = 334,5$  K)

## 2.2. Diseño de las proteínas artificiales

Nuestros objetivos involucran el diseño de proteínas artificiales, obtenidas alargando la secuencia original de la CTPRa3 (código PDB (Protein Data Bank): 1NA0). En este proceso, asumimos que la proteínas alargadas siguen presentando un estado nativo estable, y que toda la información relevante sobre éstos se pueda resumir en sus mapas de contacto correspondientes.

En primer lugar, partimos de la proteína inicial, que consta de dos hélices iniciales, dos hélices centrales y tres hélices finales. Construimos el mapa de contacto correspondiente: definimos que dos residuos  $i$  y  $j$  están en contacto:  $\Delta_{i,j} = 1$ , si existen por lo menos dos de sus átomos (excluyendo a los hidrógenos) que distan entre ellos menos que un umbral de  $4.5 \text{ \AA}$ ; excluimos las interacciones entre átomos del mismo residuo y de dos residuos primeros vecinos a lo largo de la cadena, estando sus contactos siempre presentes:  $\Delta_{i,i} = \Delta_{i,i+1} = 0$ . Como se puede comprobar en la figura 2, es fácil identificar las 7 hélices en el mapa de contacto, así como las interacciones entre ellas, que definen la estructura nativa.

El criterio para la división en 2, 2, 3 hélices (frente, por ejemplo, a la alternativa 2,(2x)2,1) reside en la presencia de algunas diferencias en las secuencias y en los patrones de contacto del módulo inicial y final frente al módulo central.

Al espacio que se produce entre distintas hélices se le denomina bucle, refiriéndonos a bucles inter-módulo cuando se encuentra entre dos hélices del mismo módulo y a bucles intra-módulo cuando las hélices pertenecen a módulos distintos.

**Aumento del número de módulos** En la primera parte del trabajo, estudiamos los efectos de aumentar artificialmente el número de módulos centrales (replicando su patrón de contactos un número arbitrario de veces), o de quitar parcial o totalmente las estructuras de los dos extremos. En la figura 3 izquierda se ve un ejemplo de una proteína diseñada con este criterio.

**Alargamiento de bucles** Otra forma de modificar la proteína es añadir aminoácidos en las regiones entre las hélices. Los aminoácidos añadidos presentan contactos únicamente con sus segundos vecinos ( $i, i+2$ ), mientras que los que existían en la proteína inicial mantienen sus contactos a pesar de que la distancia entre ellos ha aumentado.

Un ejemplo se puede encontrar en la figura 3 derecha, en el que se han alargado en 5 aminoácidos todos los loops intra-módulo.

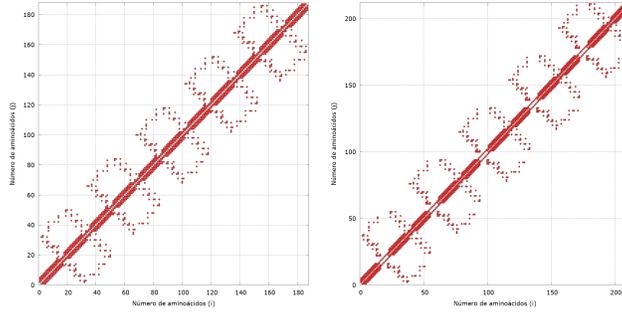


Figura 3: *Izquierda*: Mapa de contacto de una proteína artificial, obtenida añadiendo tres módulos centrales. *Derecha*: Mapa de contacto de una proteína artificial, obtenida alargando las regiones intra-módulo.

### 2.3. Caracterización termodinámica de las proteínas

Una vez diseñada la proteína, estudiamos sus propiedades termodinámicas con el modelo WSME:

- El calor específico nos permite conocer dónde está la transición <sup>1</sup> unfolded-nativo, calculándolo como

$$C_v = \frac{\partial E}{\partial T}$$

donde E representa la energía interna, que viene dada por la derivada de la función de partición Z de la siguiente manera

$$E = -\frac{\partial}{\partial \beta} \ln Z$$

donde  $Z = \sum_{\{m_i=0,1, i=1, \dots, N\}} \exp(-\beta H)$

<sup>1</sup>Hablamos impropriadamente de transición, siendo que las proteínas tienen tamaño finito, y además sus interacciones son de alcance finito.

- La cooperatividad nos da idea de ante qué tipo de transición nos encontramos. Para cuantificarla, introducimos el parámetro  $\kappa$  de van't Hoff, dado por la relación entre la variación de energía interna de un sistema de dos estados equivalente al dado y la variación de energía interna real (calorimétrica) durante la transición:

$$\kappa = 2T_{max} \sqrt{\frac{k_B C_v(T_{max})}{\Delta H_{cal}}} = 2T_{max} \sqrt{\frac{k_B C_v(T_{max})}{E_{T \gg T_m} - E_{T \ll T_m}}}$$

donde  $T_{max}$  es la temperatura del pico del calor específico, y  $E_{T \gg T_m}$ ,  $E_{T \ll T_m}$  son las energías internas a temperaturas mucho más grandes o pequeñas que la temperatura de transición  $T_m$ . Un sistema de dos estados es un caso límite de los casos reales, ya que en éstos posiblemente la presencia de estados metaestables ensanchará la transición y provocará un cambio de energía mayor del que se da si sólo dos estados están poblados. Por lo tanto  $\kappa \in [0, 1]$ , con  $\kappa = 1$  indicando un sistema completamente cooperativo, de dos estados.

- El perfil de energía libre muestra la estructura de mínimos, expresando esta energía como

$$F(M) = -\frac{1}{\beta} \ln Z_M,$$

con  $Z_M = \sum_{\{m_i=0,1,i=1,\dots,N\}} \delta(\sum_i m_i - M) \exp(-\beta H)$  las funciones de partición parciales, calculadas sobre los estados con exactamente  $M$  aminoácidos nativos.

- A partir de la probabilidad de islas nativas podemos conocer qué tipo de estructuras encontramos en esos mínimos, dándonos información de cómo de estable es una región nativa aislada. Llamamos isla nativa a una región con todos los residuos nativos entre la posición  $i$  y la  $j$ , con  $i - 1$  y  $j + 1$  desnaturalizados. Su probabilidad coincide con

$$\nu_{ij} = \left\langle (1 - m_{i-1}) \left( \prod_{k=i}^j m_k \right) (1 - m_{j+1}) \right\rangle$$

donde

$$\langle X \rangle = \frac{1}{Z} \sum_{\{m\}} (X) \exp(-\beta H)$$

La expresión que se representa en gráficas como la de la figura 8 es  $-\log \nu_{ij}$ , de manera que, cuanto más oscura sea una región, mayor será la probabilidad de encontrar islas nativas.

- La probabilidad de residuos nativos  $\langle m_i \rangle$  representa gráficamente qué partes de la proteína son más nativas, es decir, la fracción de nativos a diferentes temperaturas o situaciones.

Para obtener estas características y las temperaturas a las que nos interesa calcularlas, empleamos distintos parámetros:

- La fracción de residuos nativos,  $m$ , que se calcula como

$$m = \frac{1}{N} \sum_{i=1}^N \langle m_i \rangle$$

- El parámetro de orden,  $p$ , calculado como

$$p = \frac{m(T) - m(T = \infty)}{m(T = 0) - m(T = \infty)}$$

El perfil de energía libre y las probabilidades de islas nativas y de residuos nativos dependen de la temperatura a la que se calculan. Para comparar fiablemente proteínas de longitud y comportamiento diferente, elegimos utilizar, para cada proteína, su temperatura de transición, eligiendo que esta sea aquella para la que el parámetro  $p = 0,5$  que incidentalmente coincide con la temperatura del máximo del calor específico. Esta temperatura se presenta en la figura 4.

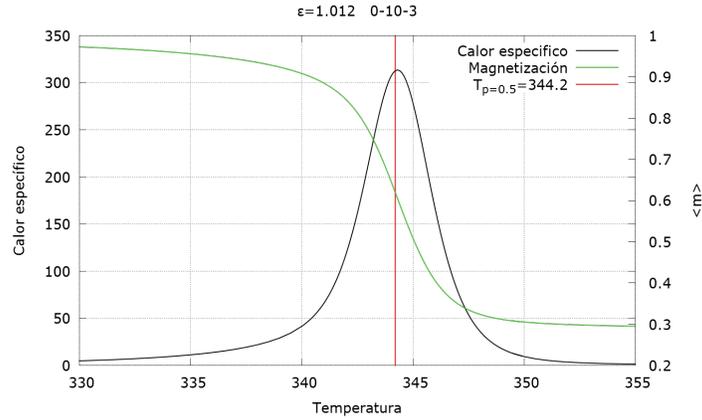


Figura 4: Localización de la temperatura de transición empleada en la curva del calor específico.

### 3. Resultados

#### 3.1. Estudio del aumento del número de módulos centrales

En primer lugar, estudiamos el comportamiento de la proteína, representando su calor específico al ir añadiendo módulos a la parte central.

En la figura 5 observamos que, conforme aumenta el número de módulos, la proteína se vuelve más estable, es decir, la temperatura necesaria para desnaturalizarla es más elevada.

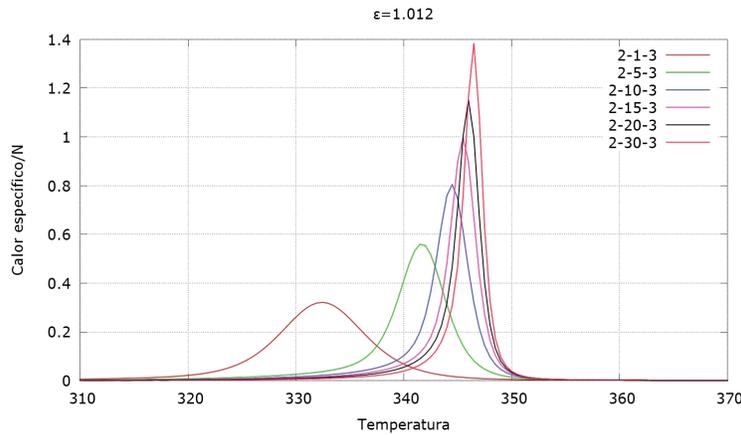


Figura 5: Curva del calor específico para distinto número de módulos centrales.

Además, comprobamos que, conforme aumenta el número de hélices, la proteína se vuelve menos cooperativa (disminuye el parámetro de van't Hoff)

Hélices	7	15	25	35	45
Estructura	2-1-3	2-5-3	2-10-3	2-15-3	2-20-3
$\kappa$	0,86	0,76	0,70	0,65	0,62

Cuadro 1: La primera columna se refiere al número de hélices de la proteína estudiada, con la estructura indicada en la segunda columna, donde el primer dígito representa el número de hélices iniciales, el segundo al número de módulos centrales y el tercero el número de hélices finales. La tercera columna presenta el valor de la  $\kappa$  de van't Hoff para cada estructura.

Comprobamos que el valor de la relación de energías disminuye conforme aumenta el tamaño de la proteína, lo que implica una disminución de la cooperatividad, debido a que, conforme se añaden módulos, la estructura interna gana importancia.

Estos resultados obtenidos conciden con los observados experimentalmente por Cortajarena y Regan [8], quienes, estudiando el mismo tipo de proteínas, han obtenido como conclusiones que, al aumentar el número de módulos, aumenta la temperatura a la que se alcanza el máximo de calor específico y que, cuanto mayor es el número de módulos, la curva del calor específico se estrecha. También han estudiado la relación de energía, encontrando que ésta disminuye conforme aumenta el número de módulos, aunque sus resultados son menos cooperativos que los obtenidos con nuestro modelo.

Encontramos una dependencia del calor específico con el número de aminoácidos de la proteína, como se observa en la figura 6. Aunque no alcanzamos a ver la saturación, tratando con un sistema unidimensional con interacciones de alcance finito, nos esperamos que esta se produzca para algún tamaño de la proteína.

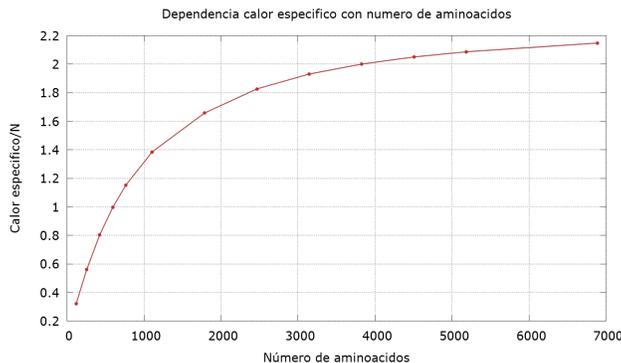


Figura 6: Curva de saturación del calor específico al aumentar el número de aminoácidos.

En la figura 5 se ha observado que, al representar el calor específico de las proteínas, la curva presenta una asimetría con respecto al pico que también obtienen Cortajarena y Regan en su estudio experimental [8]. Esta asimetría observada hacia la izquierda sugiere que lo primero que se excita son los estados intermedios próximos al estado nativo, antes de que la temperatura sea suficiente para cruzar la barrera que separa la región nativa de la desnaturalizada.

Este escenario está corroborado por el análisis de los perfiles de energía libre de proteínas con diferentes tamaños, donde se ve un aumento del número de mínimos metaestables en la cuenca nativa, que compiten en población con el estado nativo al crecer el número de módulos.

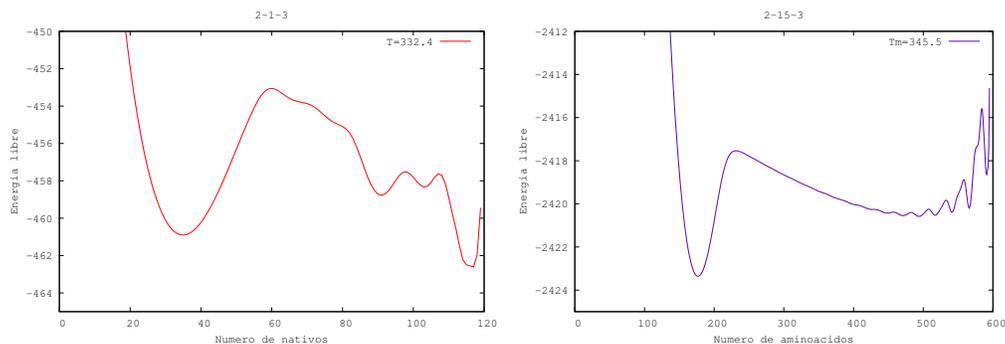


Figura 7: *Izquierda:* Perfil de energía libre de la proteína 2-1-3. *Derecha:* Perfil de energía libre de la proteína 2-15-3

Estudiamos también la probabilidad de islas nativas para la proteína original.

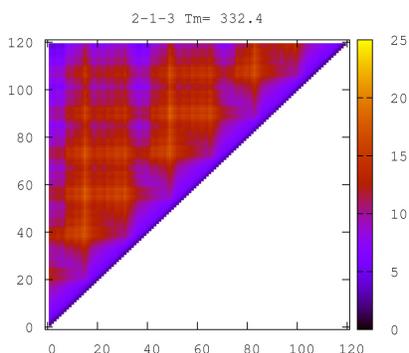


Figura 8: Gráfica de probabilidad de islas nativas de la proteína 1NA0. Las zonas más oscuras presentan una mayor probabilidad que las claras. Las regiones oscuras con forma triangular próximas a la diagonal representan las hélices aisladas de la proteína estudiada.

En las regiones próximas a la diagonal encontramos formas triangulares más oscuras, cuyo significado es que, a esta temperatura, cada hélice aislada es bastante estable, permitiéndonos así ver claramente la estructura interna. Además, se observa que las parejas de módulos (por ejemplo, la región 1-30 o la 70-100) también presentan una probabilidad elevada, por lo que estas estructuras también son estables, aunque en menor medida que las hélices por separado.

Es interesante destacar la estabilidad de estructuras formadas por un módulo entero y la mitad del siguiente, mientras que si lo que tenemos es la mitad de un módulo y el siguiente entero, la estructura es inestable. Este resultado se ha observado experimentalmente en proteínas modulares[10], aunque en una proteína que presenta dos módulos con distintas configuraciones, dando lugar a la

idea de que se trata de unidades cooperativas que se solapan.

Se observa también que bloques de cuatro hélices son estables, siempre que se trate de dos módulos completos, presentando inestabilidad en estructuras formadas por la mitad de un módulo, un módulo completo y la mitad del siguiente.

Comprobamos cómo afectan las variaciones en la estabilidad a la probabilidad de residuos nativos.

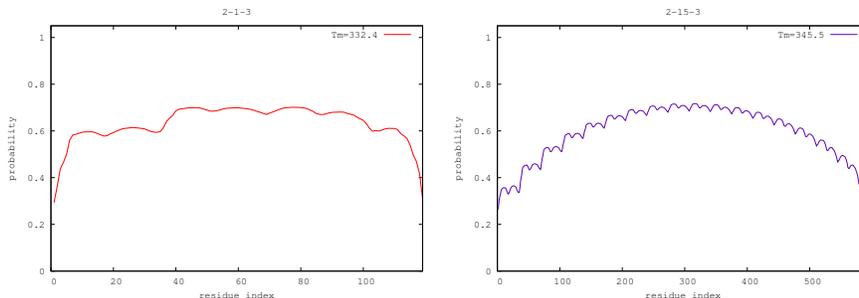


Figura 9: *Izquierda*: Probabilidad de residuos nativos de la proteína 2-1-3. *Derecha*: Probabilidad de residuos nativos de la proteína 2-15-3

El análisis del perfil de probabilidad nativa  $\langle m_i \rangle$  revela cómo el segundo y el tercer módulo de la proteína inicial 1NA0 son los más estables, mientras que las hélices inicial y final son parcialmente desnaturalizadas incluso en el estado nativo, de acuerdo con las observaciones experimentales [7]. Para proteínas muy largas, se produce una situación similar, con estabilidad decreciente yendo desde el centro hacia los extremos de la proteína.

### 3.2. Partes iniciales y finales

El comportamiento diferente de las hélices inicial y final en la gráfica 9 nos anima a estudiar el papel de estas partes y su influencia en el comportamiento de la proteína. Para ello realizamos distintas comparaciones entre proteínas con estructuras diferentes, estudiando sus propiedades termodinámicas.

**Proteínas sin el módulo inicial** En primer lugar, comparamos proteínas completas frente a proteínas con el mismo número de hélices pero sin parte inicial. Indicamos en la gráfica el número de módulos de cada estructura, siendo comparables los que tienen la misma longitud.

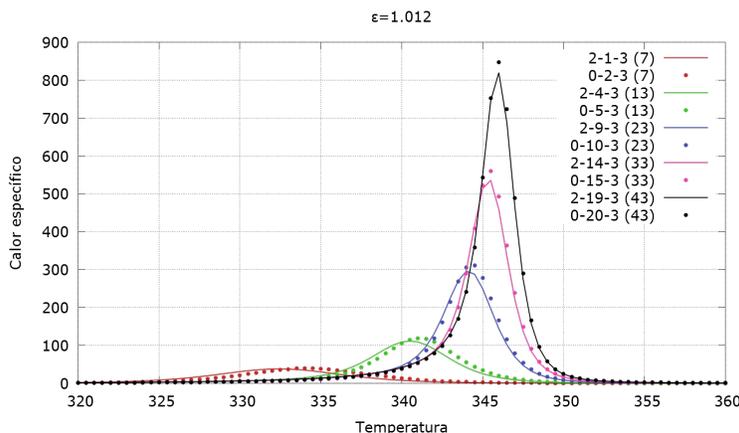


Figura 10: Comparación de las curvas de calor específico.

Observamos que, para proteínas pequeñas, aquellas que no tienen parte inicial son más estables que las que sí la presentan, pero esta diferencia va desapareciendo conforme el número de módulos centrales aumenta. Esto nos permite deducir que el tamaño de la proteína es más relevante que la presencia o no de partes iniciales y finales. La cooperatividad de estas estructuras revela que las configuraciones que carecen de parte inicial presentan una mayor cooperatividad.

Hélices	7	7	13	13	23	23	33	33	43	43
Estructura	2-1-3	0-2-3	2-4-3	0-5-3	2-9-3	0-10-3	2-14-3	0-15-3	2-19-3	0-20-3
$\kappa$	0,86	0,87	0,78	0,80	0,71	0,72	0,66	0,67	0,62	0,63

Para profundizar en el estudio de estas estructuras, representamos la energía libre de dos parejas de proteínas con el mismo número de hélices, una con parte inicial y otra sin, para las temperaturas a las que alcanzan su máximo de calor específico (figura 11).

Se comprueba que la existencia de un módulo inicial da lugar a una energía libre mayor, aunque la diferencia es porcentualmente pequeña.

Además, observando su perfil se ve que en la parte final los picos son más bajos cuando la proteína no presenta parte inicial, lo que significa que tenemos un segundo estado poblado que no existe en la proteína completa, e incluso domina sobre el estado nativo conforme aumenta el número de módulos centrales.

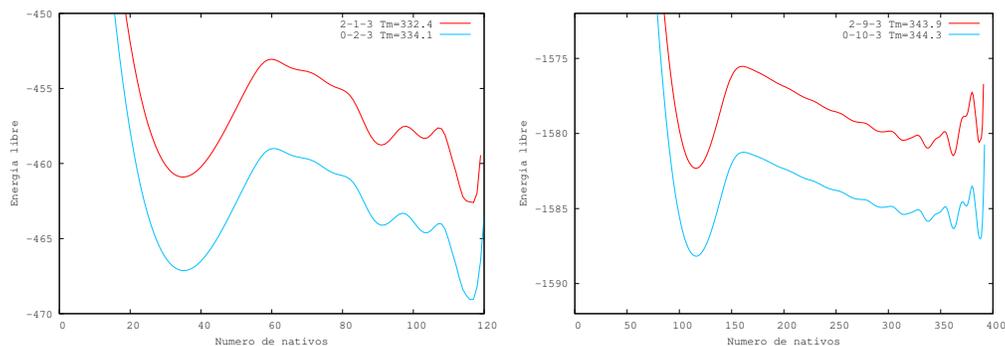


Figura 11: *Izquierda*: Perfil de energía libre para dos proteínas con 7 hélices. *Derecha*: Perfil de energía libre para dos proteínas con 23 hélices.

La comparación con la figura 11, donde se ve que las principales diferencias se encuentran entre el mínimo nativo y otro a aproximadamente 30 residuos nativos de distancia, sugiere que la sustitución del módulo inicial por un módulo central hace la proteína más cooperativa.

**Proteínas sin el módulo final** Análogamente al caso anterior, realizamos el estudio comparativo del calor específico, quitando en este caso las partes finales. Sin embargo, encontramos un problema, y es que la única manera de conseguir tener el mismo número de hélices es, o bien comparándolo con tener la parte inicial incompleta, o bien con que la incompleta sea la final. Comprobamos qué sucede en cada uno de estos casos, comenzando por comparar la parte final incompleta frente su sustitución por un módulo central.

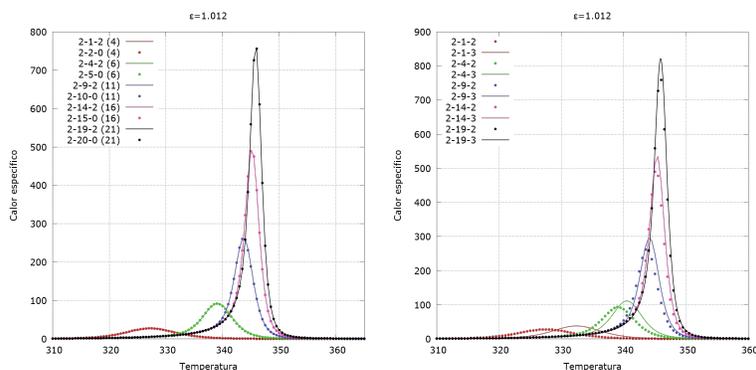


Figura 12: *Izquierda*: Curvas de calor específico sin el módulo final.

Como se observa en la figura 12 izquierda, los resultados apenas difieren de tener parte final incompleta a no tenerla, por lo que una posible conclusión es que la hélice que añade estabilidad a la proteína es la última y no el conjunto de las tres últimas que hemos denominado parte final. Esto se evidencia en la figura 12 derecha, donde se han representado proteínas con la parte final completa frente a otras con ausencia de la última hélice. Estos resultados coinciden con los observados experimentalmente [14].

**Proteínas con dos hélices libres** Para averiguar si lo relevante es el número total de hélices, independientemente de su organización en módulos, realizamos la comparación entre los casos de ausencia de parte final frente a tener la parte inicial incompleta.

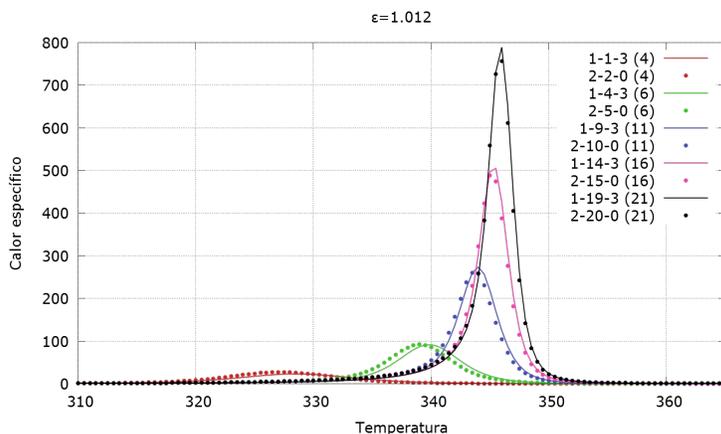


Figura 13: Curvas del calor específico para proteínas con distinta organización de sus hélices.

Y comprobamos que en este caso la diferencia sí es apreciable, obteniendo una mayor estabilidad cuando la proteína tiene la parte final completa, aunque esta diferencia vuelve a perder importancia conforme aumenta el tamaño de módulos centrales.

Volvemos a profundizar estudiando los perfiles de energía libre de dos parejas de proteínas con el mismo número de hélices, una con la parte inicial completa y la otra sin parte final (figura 14).

En este caso, observamos que, cuando la proteína presenta dos hélices (inicial y final) libres, aparecen unos mínimos de energía para estados en los que la proteína aún no es completamente nativa. El análisis de las islas nativas, figura 15 izquierda, revela la presencia de unas estructuras estables no nativas, especialmente aquella que tiene los primeros 20 aminoácidos desplegados. Esta configuración es más estable que el estado nativo, por lo que en este caso la estructura de equilibrio tiene la primera hélice desnaturalizada.

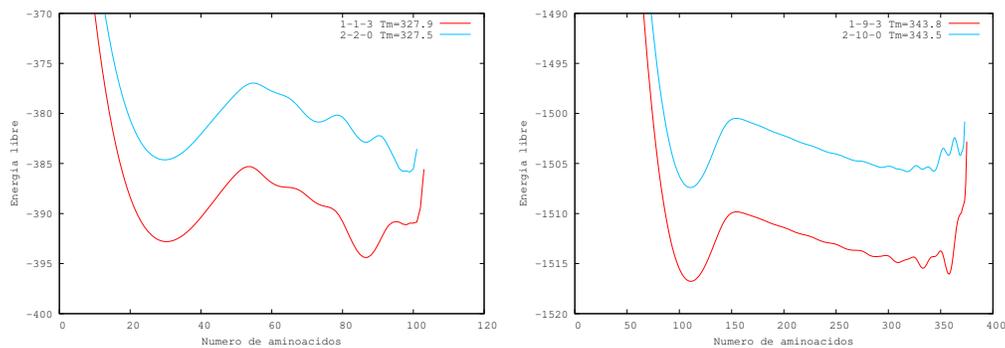


Figura 14: *Izquierda*: Perfil de energía libre para dos proteínas con 6 hélices. *Derecha*: Perfil de energía libre para dos proteínas con 22 hélices.

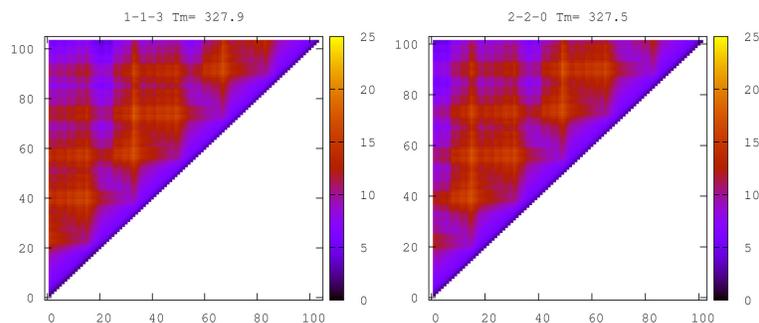


Figura 15: *Izquierda*: Probabilidad de islas nativas de la proteína con la parte inicial incompleta y un módulo central. *Derecha*: Probabilidad de islas nativas de la proteína sin parte final y dos módulos centrales.

### 3.3. Alargamiento de los bucles

En esta tercera parte del trabajo, nos interesa estudiar lo que sucede al añadir aminoácidos entre las distintas hélices de la proteína.

La pregunta que nos proponemos es: ¿hay diferencia entre bucles inter- e intra-módulo, en lo que concierne a la estabilidad, la cooperatividad y la distribución de estructuras a lo largo de la cadena? Para averiguarlo, tomamos como proteína original la que consta de las dos hélices iniciales, tres módulos centrales y las tres hélices finales, cuyo mapa de contacto está representado en la figura 3 izquierda.

**Alargamiento de 5 aminoácidos en bucles distintos** En primer lugar, comparamos el calor específico de esta proteína inicial con los obtenidos al añadir cinco aminoácidos cada vez a un bucle diferente, tanto intra-módulos como inter-módulos.

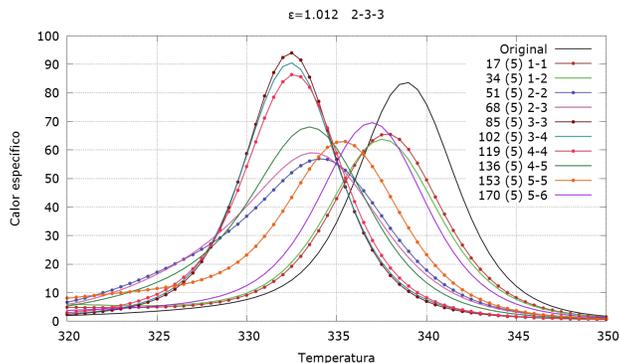


Figura 16: Curvas de calor específico para proteínas con un bucle alargado 5 aminoácidos

En la figura 16 observamos que la proteína original es la más estable, seguida por aquellas estructuras en las que los bucles alargados se encuentran en los extremos de la proteína, aunque estas presentan un calor específico menor. Sin embargo, encontramos tres casos diferentes, en los que el calor específico alcanzado por la proteína es mayor que en la proteína original. En dos de los tres casos, correspondientes al aminoácido 85 y 102, la proteína es más cooperativa que la original, y en el tercero el parámetro  $\kappa$  es sólo ligeramente inferior al original.

Estudiamos los perfiles de energía libre para dos de estos casos, el que se produce en el aminoácido 85, un intra-módulo, y el encontrado en el aminoácido 102, un inter-módulo, comparándolos con el perfil de la proteína original y con otro caso de alargamiento de bucle, esta vez en el aminoácido 136, un inter-módulo. Para las cuatro estructuras, tomamos la temperatura para la cual la proteína alcanza su máximo calor específico (figura 17)

Se observa que el caso intra-módulo tiene barreras ligeramente más altas que el correspondiente inter-módulo. Encontramos también que en todos los casos aparece un nuevo mínimo ausente en la estructura original. Para entender lo que sucede, estudiamos la probabilidad de islas nativas para cada una de estas proteínas.

La proteína original presenta una estructura regular de islas relativamente estables, como se observa en la figura 18 superior izquierda. Por lo tanto, es difícil identificar con precisión a qué estructuras se corresponde el mínimo que se encuentra en  $M=160$  en el perfil de energía libre 17, contribuyendo a él varias combinaciones (1-120, 40-180, ..., cualquier isla de tamaño menor que 160, junto a otros residuos nativos del resto de la proteína); en cambio, está clara la interpretación del mínimo en  $M=180$ , que corresponde al estado completamente nativo 1-180, y resulta plausible que el mínimo desplegado en  $M=60$  tenga contribuciones de residuos aislados o helices aisladas parcialmente formadas.

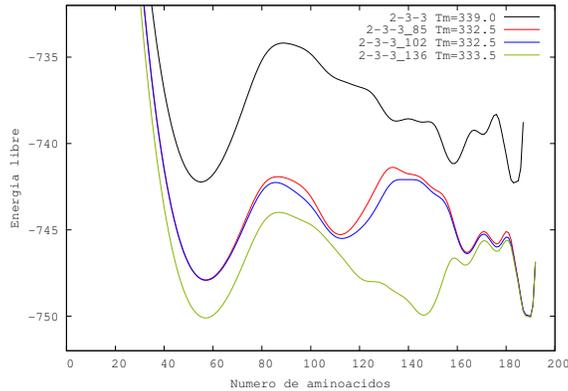


Figura 17: Perfiles de energía libre para distintas estructuras.

Por una parte, en la proteína en la que alargamos el quinto bucle (aminoácido 85, figura 18 superior derecha), desaparece buena parte de la estructura de islas anterior, reforzándose la probabilidad de las partes iniciales en la región 1-80 y finales 100-185 por separado, así como la probabilidad del estado completamente nativo 1-185. El mínimo en  $M=160$  ahora es mucho menos relevante con respecto al nativo, y es compatible con la presencia de estructura en ambas regiones inicial y final; también aparece otro mínimo en  $M=110$  aminoácidos, que podemos suponer tendrá contribuciones de los dos dominios inicial 1-80 y final 100-185, posiblemente con uno completamente formado. De una manera más precisa, hemos analizado el perfil de la fracción nativa, comprobando como se refuerza la estructura final, más que la inicial, por el alargamiento en el residuo 85. La información apenas cambia con respecto a la figura 19 izquierda, por lo que no se incluye la gráfica.

Es posible, por lo tanto, que las barreras encontradas en el perfil de energía libre (figura 17) correspondan a la formación de una o dos islas nativas separadas, que luego se juntan para formar el estado nativo, aunque sea imposible inferir un camino de plegamiento a partir de estos datos de equilibrio.

El comportamiento de la proteína en la que el bucle alargado es el sexto (aminoácido 102, figura 18 inferior izquierda) es prácticamente el mismo, aún tratándose de un inter-módulo. Sin embargo, encontramos que la parte inicial tiene mayor probabilidad y la parte final menor frente a lo obtenido para el quinto bucle.

En cambio, observamos que en el caso del alargamiento en el octavo bucle (aminoácido 136, figura 18 inferior derecha), el añadir los aminoácidos genera un mínimo en  $M = 145$  en la figura 17, compatible con la formación de estructuras independientes en la región 1-135 y 140-180, como se puede ver en la figura 18 inferior derecha.

La figura 19 derecha revela como las partes inicial y central se vuelven más estructuradas que la final, lo que es compatible con un mecanismo de formación de estructura entre 0 y 130, superando una primera barrera, y con un plegamiento completo hacia la parte final, superando una segunda barrera de energía libre.

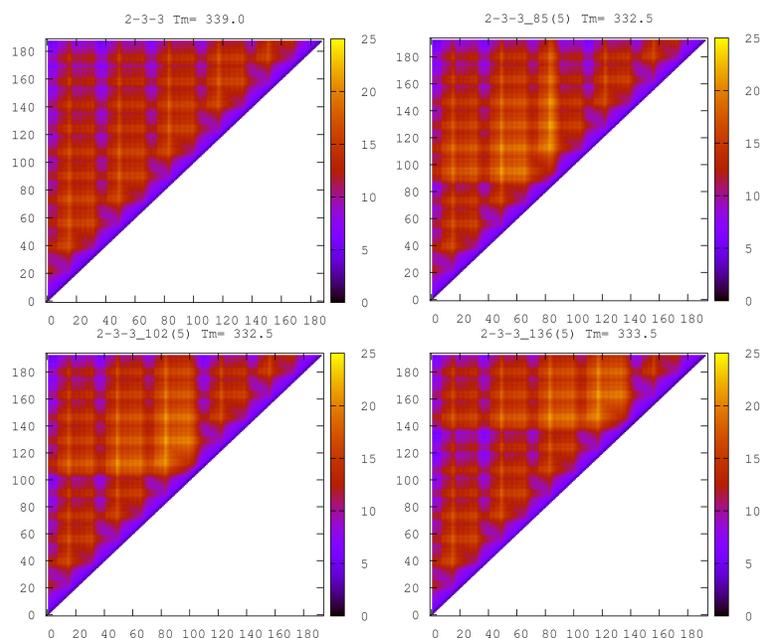


Figura 18: *Superior izquierda*: Probabilidad de islas nativas de la proteína inicial, con las partes inicial y final completas y tres módulos centrales. *Superior derecha*: Probabilidad de islas nativas de la proteína con el intra-módulo 85 alargado. *Inferior izquierda*: Probabilidad de islas nativas de la proteína con el inter-módulo 102 alargado. *Inferior derecha*: Probabilidad de islas nativas de la proteína con el inter-módulo 136 alargado.

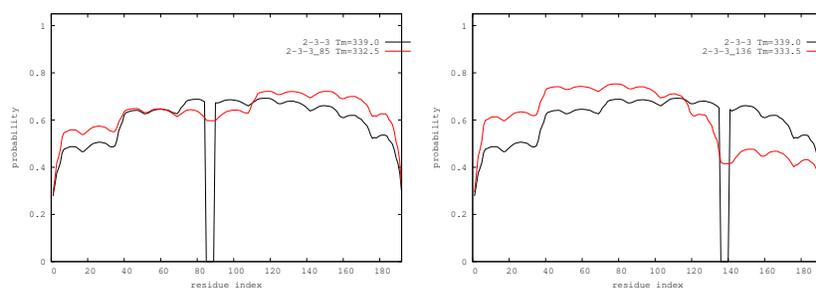


Figura 19: *Izquierda*: Probabilidad de residuos nativos alargando el quinto bucle. *Derecha*: Probabilidad de residuos nativos alargando el octavo bucle.

**Alargamiento de 5 aminoácidos en todos los bucles** Para analizar si hay diferencia entre bucles intra-módulo o inter-módulo, comprobamos cómo se comporta la proteína si alargamos todos los inter-módulos, todos los intra-módulos o todos los bucles, añadiendo en todos los casos 5 aminoácidos en los lugares correspondientes. Para poder comparar los calores específicos de cada caso, dividimos cada proteína por su número de aminoácidos.

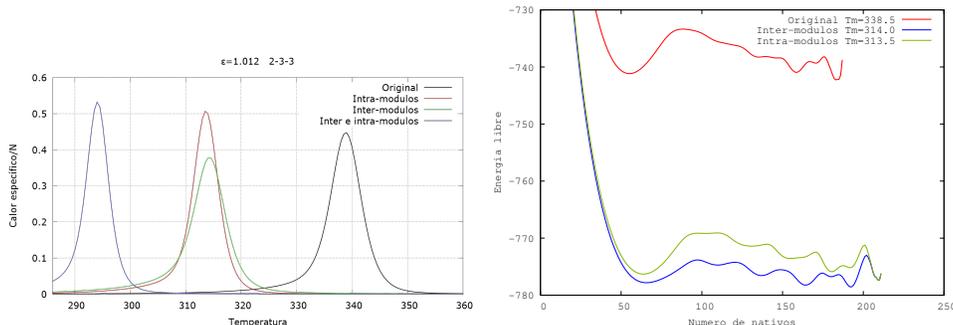


Figura 20: *Izquierda*: Curvas de calor específico al añadir 5 aminoácidos en distintas combinaciones de bucles. *Derecha*: Perfil de energía libre de diferentes estructuras.

Estudiamos la cooperatividad de estas estructuras

Estructura	Original	Intra-módulos	Inter-módulos	Inter e intra
$\kappa$	0,80	0,79	0,69	0,80

Comprobamos que, en todos los casos, añadir aminoácidos implica una pérdida de estabilidad de la proteína. Además, se observa que aunque la estabilidad de los inter e intra-módulos es prácticamente la misma, el calor específico que alcanzan es distinto. Esto sugiere que al alargar los intra-módulos aumenta la barrera de energía libre que se debe superar para plegar cada módulo y llegar al estado completamente nativo, mientras que alargar los inter-módulos favorece la independencia de los módulos, lo que disminuye las barreras.

En efecto, el estudio de los perfiles (figura 20 derecha) en este caso revela que, al alargar los intra-módulos, el estado más estable es el nativo completo, mientras que al alargar los inter-módulos el estado plegado más estable no es el nativo completo.

Una vez comprobado lo que sucede al añadir el mismo número de aminoácidos en distintos bucles de la proteína, pasamos a estudiar el caso complementario, en el que en el mismo bucle añadimos cada vez un número distinto de aminoácidos.

**Alargamiento con un número variable de aminoácidos** Comenzamos estudiando el caso de alargar el quinto bucle (aminoácido 85), añadiendo distinto número de aminoácidos y comparando el comportamiento del calor específico para cada caso, dividiendo entre el número de aminoácidos de cada estructura para que sean comparables (figura 21 izquierda).

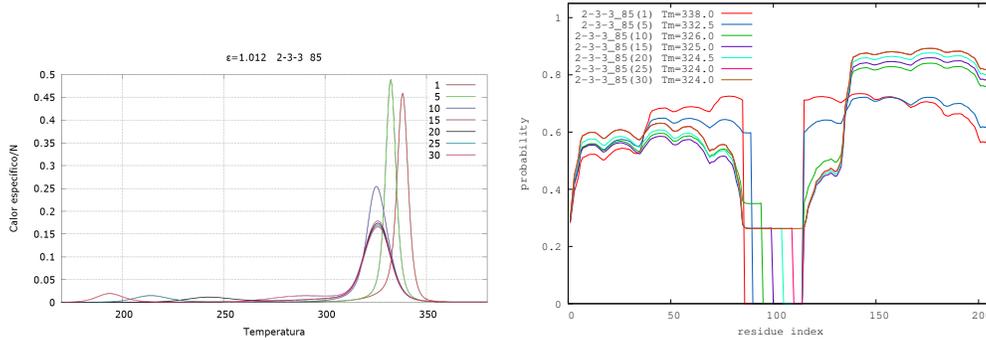


Figura 21: *Izquierda*: Curvas del calor específico al alargar el quinto bucle con distinto número de residuos. *Derecha*: Probabilidad de residuos nativos al alargar el quinto bucle con distinto número de residuos. Se han introducido artificialmente huecos en las proteínas para que haya correspondencia entre los mismos residuos en proteínas de tamaño diferente.

Observamos que, al ir aumentando la separación de las hélices, la proteína se va desestabilizando. La altura del pico sube al principio, señal de que los residuos centrales pertenecen al núcleo del plegamiento que determina la barrera, y luego baja al aumentar el número de aminoácidos añadidos; al añadir los suficientes aminoácidos, aparecen segundos picos a baja temperatura debidos a que la proteína se divide en dos dominios independientes, y el dominio más pequeño tiene menor estabilidad y desnaturaliza a temperaturas más bajas.

Aminoácidos añadidos	0	1	5	10	15	20	25	30
$\kappa$	0,80	0,79	0,81	0,57	0,49	0,48	0,47	0,47

La tabla anterior evidencia que la cooperatividad se mantiene e incluso aumenta al añadir unos pocos residuos y comienza a disminuir cuando el número de aminoácidos va aumentando. Este decrecimiento se estanca cuando la proteína se divide en dos proteínas independientes.

Los perfiles de probabilidad nativa (figura 21 derecha) revelan que, al añadir aminoácidos, se resta estabilidad a las hélices contiguas. Además, al alargar el bucle central, aumenta la estabilidad de la parte final de la proteína y se desestabiliza la inicial, debido a que la influencia de una sobre otra disminuye. Añadiendo el número suficiente de aminoácidos, la proteína se separa, comportándose como dos dominios distintos.

Comprobamos de la misma manera lo que ocurre al añadir distinto número de aminoácidos en un bucle del extremo final, el noveno bucle (aminoácido 153).

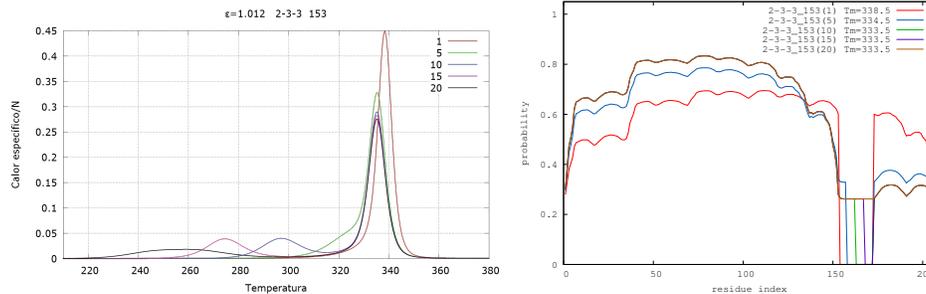


Figura 22: *Izquierda*: Curvas de calor específico alargando el noveno bucle con distinto número de residuos. *Derecha*: Probabilidad de residuos nativos al alargar el noveno bucle con distinto número de residuos.

El calor específico por aminoácido (figura 22 izquierda) revela que, al aumentar la distancia entre las hélices, disminuye la estabilidad, aunque en este caso no encontramos un pico de calor específico como en el quinto bucle.

Al igual que en el caso anterior (figura 21 izquierda), el pico de calor específico va disminuyendo, y llega un momento en el que la proteína se divide en dos dominios diferentes, apareciendo un nuevo pico a baja temperatura del calor específico.

Aminoácidos añadidos	0	1	5	10	15	20
$\kappa$	0,80	0,79	0,67	0,63	0,63	0,62

La cooperatividad disminuye conforme aumenta el número de residuos añadidos. Esta disminución se ralentiza cuando la proteína se divide en dos dominios independientes.

Si pasamos a estudiar el perfil de probabilidad nativa (figura 22 derecha) comprobamos que, añadiendo 10 aminoácidos o más, las dos últimas hélices se desacoplan.

## 4. Conclusiones

En este trabajo hemos aplicado un sencillo modelo de mecánica estadística, el WSME, a las proteínas modulares diseñadas (CTPR), utilizando la estructura cristalizada 1NA0, con tres módulos, como estructura de partida. A pesar de su extrema sencillez, el modelo permite reproducir y racionalizar los principales resultados experimentales presentes en la literatura.

En particular, en el estudio de proteínas artificiales obtenidas aumentando el número de módulos centrales, reproducimos los resultado de [8], obteniendo evidencias de que, al aumentar el número de módulos centrales, las curvas de calor específico se estrechan y se desplazan hacia una temperatura de máximo mayor, mientras disminuye la cooperatividad de la proteína, medida por la relación  $\kappa$  entre la variación de energía interna de un sistema de dos estados que mejor reproduce la proteína en cuestión, y la variación de energía interna real durante la transición. Otro resultado obtenido

en ambos estudios es la asimetría en la curva del calor específico, evidencia de que existe gran número de estados metaestables cercanos al estado nativo que empiezan a estar poblados antes de que se alcance la temperatura de transición. De acuerdo con los datos experimentales, hemos visto que las partes más propensas a desplegarse de la proteína inicial son la primera y última hélice, y esto se generaliza a proteínas largas en la observación de que, en la temperatura de transición, la probabilidad de que un residuo esté ordenado como en el estado nativo aumenta suavemente moviéndose desde los extremos de la proteína hacia los módulos centrales.

Por otra parte, inspirados por los resultados anteriores, hemos estudiado el papel que tienen el módulo inicial y la parte final (módulo final más hélice extra) en la estabilidad de la proteína, puesto que estas regiones tienen un patrón de contactos ligeramente diferente del patrón del módulo central. Sorprendentemente, hemos encontrado que la parte inicial de las proteínas estudiadas la desestabiliza, con respecto a la sustitución por el módulo central. Esto no es habitual, puesto que las hélices introducidas al comienzo de las proteínas diseñadas suelen emplearse para estabilizarlas, por lo que nuestro resultado podría deberse a la extrema sencillez del modelo. Sin embargo, encontramos que la parte final sí estabiliza la proteína, llegando a la conclusión de que es sobre todo la última hélice la que tiene un papel fundamental en este sentido (trabajando sin la última hélice, la sustitución del módulo final por el módulo central no proporciona ningún cambio significativo).

También, como era de esperar, hemos visto que el papel de las partes iniciales y finales es importante cuando las proteínas que estudiamos tienen un bajo número de módulos centrales, pero si el tamaño aumenta encontramos que el número de módulos total es más relevante que la presencia o ausencia de partes iniciales y finales con diseño especial.

En la segunda parte del estudio, hemos investigado los efectos de alargar los bucles de la proteína, añadiendo regiones con interacciones homogéneas y completamente locales ( $i, i + 2$ ). Como era de esperar, el alargamiento de un bucle en cualquier posición aumenta la entropía sin aportar nada en energía, así que en todos casos la temperatura de transición, definida por la posición del pico de calor específico, baja y la proteína se desestabiliza. Perturbar los bucles en las regiones centrales de la proteína aumenta el pico y la cooperatividad con respecto a la proteína original, sugiriendo que estos bucles pertenecen al núcleo de plegamiento, y su perturbación aumenta la barrera eficaz entre estado nativo y desnaturalizado. En cambio, perturbar bucles en regiones más próximas a los extremos de la proteína disminuye la cooperatividad, generando nuevos mínimos en el paisaje de energía libre, que en algunos casos hemos podido relacionar con la presencia de estructuras ("islas" nativas) en la primera o la segunda parte de la proteína.

También hemos comprobado que, al añadir el número suficiente de aminoácidos en los bucles, las proteínas se dividen en dos dominios independientes, como es razonable, con la aparición de dos picos en el calor específico. Notablemente, esta división se produce con un número menor de residuos añadidos cuando los bucles se encuentran más cerca de los extremos, siendo, en cambio, necesario añadir un gran número de aminoácidos para dividir los dominios si se alargan los bucles centrales, lo que apunta, una vez más, a un papel más importante de las partes centrales como núcleo de plegamiento.

No hemos podido, en cambio, encontrar una diferencia clara entre el papel de los bucles intra-módulo e inter-módulo, considerados por separado. Sin embargo, el análisis de la construcción con

todos los bucles inter-módulo alargados resulta tener menor cooperatividad y menores barreras de energía libre que la proteína con todos los intra-módulos alargados. Todas estas observaciones podrían ser relevantes a la hora de elegir dónde y cómo funcionalizar estas proteínas modulares, de cara a aplicaciones biotecnológicas.

Este trabajo puede dar pie a numerosos desarrollos, que no hemos podido tratar por límites de tiempo, como por ejemplo: un estudio más riguroso del límite termodinámico, con la construcción y análisis de una matriz de transferencia para calcular de forma exacta el valor asintótico de las variables termodinámicas; un análisis más realista de la termodinámica, utilizando versiones del modelo que mejor reproduzcan los detalles de la fenomenología del equilibrio, para alcanzar predicciones más cuantitativas de los resultados experimentales; un estudio de la cinética del plegamiento, a través de simulaciones de Monte-Carlo o de aproximaciones como la LEA ("Local Equilibrium Approximation"), para tratar de reproducir la gran cantidad de datos experimentales sobre las velocidades y los caminos de plegamiento.

## Referencias

- [1] H. Abe and H. Wako. Folding/unfolding kinetics of lattice proteins studied using a simple statistical mechanical model for protein folding, I: Dependence on native structures and amino acid sequences. *Physica A*, 388(17):3442–3454, SEP 1 2009.
- [2] Pierpaolo Bruscolini and Alessandro Pelizzola. Exact solution of the Muñoz-Eaton model for protein folding. *Phys. Rev. Lett.*, 88(25 Pt 1):258101, Jun 2002.
- [3] Pierpaolo Bruscolini, Alessandro Pelizzola, and Marco Zamparo. Downhill versus two-state protein folding in a statistical mechanical model. *J. Chem. Phys.*, 126(21):215103, Jun 2007.
- [4] Pierpaolo Bruscolini, Alessandro Pelizzola, and Marco Zamparo. Rate determining factors in protein model structures. *Phys. Rev. Lett.*, 99(3):038103, Jul 2007.
- [5] T. Cellmer, E. R. Henry, J. Hofrichter, and W. A. Eaton. Measuring internal friction of an ultrafast-folding protein. *Proc. Natl. Acad. Sci. USA*, 105(47):18320–18325, NOV 25 2008.
- [6] H. S. Chung and A. Tokmakoff. Temperature-dependent downhill unfolding of ubiquitin. II. Modeling the free energy surface. *Proteins*, 72(1):488–497, JUL 2008.
- [7] Aitziber L Cortajarena, Simon GJ Mochrie, and Lynne Regan. Mapping the energy landscape of repeat proteins using NMR-detected hydrogen exchange. *Journal of molecular biology*, 379(3):617–626, 2008.
- [8] Aitziber L Cortajarena and Lynne Regan. Calorimetric study of a series of designed repeat proteins: modular structure and modular folding. *Protein Sci.*, 20(2):336–40, February 2011.
- [9] Ken A Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046, 2012.

- [10] Dominika T Gruszka, Fiona Whelan, Oliver E Farrance, Herman KH Fung, Emanuele Paci, Cy M Jeffries, Dmitri I Svergun, Clair Baldock, Christoph G Baumann, David J Brockwell, et al. Cooperative folding of intrinsically disordered domains drives assembly of a strong elongated protein. *Nature communications*, 6, 2015.
- [11] K. Itoh and M. Sasai. Cooperativity, connectivity, and folding pathways of multidomain proteins. *Proc. Natl. Acad. Sci. USA*, 105(37):13865–13870, SEP 16 2008.
- [12] Tommi Kajander, Aitziber L. Cortajarena, Ewan R. G. Main, Simon G. J. Mochrie, and Lynne Regan. A new folding paradigm for repeat proteins. *J. Am. Chem. Soc.*, 127(29):10188–10190, Jul 2005.
- [13] Thomas J Lane, Diwakar Shukla, Kyle A Beauchamp, and Vijay S Pande. To milliseconds and beyond: challenges in the simulation of protein folding. *Current opinion in structural biology*, 23(1):58–65, 2013.
- [14] Ewan R G Main, Yong Xiong, Melanie J Cocco, Luca D’Andrea, and Lynne Regan. Design of stable alpha-helical arrays from an idealized TPR motif. *Structure*, 11(5):497–508, May 2003.
- [15] Ewan RG Main, Katherine Stott, Sophie E Jackson, and Lynne Regan. Local and long-range stability in tandemly arrayed tetratricopeptide repeats. *Proceedings of the National Academy of Sciences of the United States of America*, 102(16):5721–5726, 2005.
- [16] V Mu noz, E R Henry, J Hofrichter, and W A Eaton. A statistical mechanical model for beta-hairpin kinetics. *Proc. Natl. Acad. Sci. USA*, 95(11):5872–9, May 1998.
- [17] V Mu noz, P A Thompson, J Hofrichter, and W A Eaton. Folding dynamics and mechanism of beta-hairpin formation. *Nature*, 390(6656):196–9, November 1997.
- [18] A. Pelizzola. Exactness of the cluster variation method and factorization of the equilibrium probability for the Wako-Saito-Munoz-Eaton model of protein folding. *J. Stat. Mech.*, page P11010, 2005.
- [19] Hiroshi Taketomi, Yuzo Ueda, and N. Go. Studies on Protein Folding, Unfolding and Fluctuations by Computer-Simulation .1. Effect of Specific Amino-Acid Sequence Represented by Specific Inter-Unit Interactions. *Inter. J. Prot. Pep. Res.*, 7(6):445–459, 1975.
- [20] H. Wako and N. Saito. STATISTICAL MECHANICAL THEORY OF PROTEIN CONFORMATION .2. FOLDING PATHWAY FOR PROTEIN. *J. Phys. Soc. Jpn.*, 44(6):1939–1945, 1978.
- [21] H. Wako and N. Saito. Statistical Mechanical Theory of Protein Conformation .2. Folding Pathway for Protein. *J. Phys. Soc. Japan*, 44(6):1939–1945, 1978.
- [22] Marco Zamparo and Alessandro Pelizzola. Nearly symmetrical proteins: folding pathways and transition states. *J. Chem. Phys.*, 131(3):035101, Jul 2009.

## 5. Códigos

Hemos empleado diversos códigos en lenguaje C a lo largo del estudio para modificar la proteína y calcular la relación de energías. Los programas creados para ello son los siguientes:

- **modificar\_proteina**: permite elegir la estructura de la proteína a partir de la original 1NAO.
- **alargar\_bucles**: lo empleamos para alargar los bucles que se encuentran entre las hélices, estando también preparado para acortarlos.
- **pesos\_contactos\_nn**: desprecia el peso de los contactos y permite elegir tener en cuenta las interacciones con primeros vecinos o no.
- **k\_van't\_hoff**: calcula la relación de energías mediante la  $\kappa$  de van't Hoff.

### 5.1. modificar\_proteina

```
#include <stdio.h>
#include <stdlib.h>

#define N_i 34
#define N_i_m 17
#define N_c 34
#define N_f 51
#define N_t 120

#define max 20

int datos[N_t][N_t], datos_c[N_t][N_t], centro[N_t][N_t], centro_c[N_t][N_t],
repetir[N_t][N_t], repetir_c[N_t][N_t], helice_ini[max],
helice_fin[max];

void inicia_vector (int v[N_i][N_t]); //inicializa las matrices de datos a 0
int leeLineas (FILE *f); //lee el número de líneas de los ficheros "1NAOA_4.5.map"
(mapa de contacto de la proteína original) y "helixlist1NAO.txt" (localización de las hélices
dentro de la proteína original)
void estructura (FILE *f, int F); //lee los contactos y su peso del fichero "1NAOA_4.5.map"
y los copia en vectores
void leeHelice (FILE *f, int F); //lee la estructura de hélices del fichero
"helixlist1NAO.txt" y la copia en un vector

int main()
```

```

{
    int i, j, aux, aux2, b, F, c, K;

    int ini, cent, fin;

    FILE *datos_iniciales , *datos_finales, *helice_inicial, *helice_final;

    datos_iniciales=fopen ("1NAOA_4.5.map","r");
    datos_finales=fopen ("proteina_modificada.map","w");

    helice_inicial=fopen ("helixlist1NAO.txt","r");
    helice_final=fopen ("helixlist_modificada.txt","w");

    inicia_vector(datos);
    inicia_vector(datos_c);

    inicia_vector(helice_ini);
    inicia_vector(helice_fin);

    F=leeLineas(datos_iniciales);
    K=leeLineas(helice_inicial);

    rewind (datos_iniciales); //volvemos al principio del fichero
    rewind (helice_inicial);

    estructura(datos_iniciales, F);
    leeHelice(helice_inicial, K);

    printf("Numero de modulos en la estructura inicial (0, 1, 2): \n");
    scanf("%d", &ini);

    printf("Numero de repeticiones de la estructura central (numeros enteros a partir
    del 1): \n");
    scanf("%d", &cent);

    printf("Estructura final (0, 1, 2, 3): \n");
    scanf("%d", &fin);

    //Definimos la presencia o ausencia de la parte inicial
    if (ini==0) b=N_i;
    else if (ini==1) b=N_i_m;

```

```

else if (ini==2) b=0;

//Definimos la presencia o ausencia de la parte final
if (fin==1) c=N_i+N_c+N_i_m;
if (fin==2) c=N_i+(2*N_c);
if (fin==3) c=N_t;
if (fin==0) c=N_i+N_c;

//Escribimos la estructura inicial, con tantas hélices como se haya indicado al programa
for (i=b;i<N_i;i++)
{
    for (j=b;j<c;j++)
    {
        if (datos[i][j]==1)
        {
            if (b==0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n", i-b, j-b,
                datos_c[i][j], datos[i][j]);
            if (b!=0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n", i-b+1, j-b+1,
                datos_c[i][j], datos[i][j]);
        }
    }
}

//Escribimos la estructura central, con tantos módulos como se haya indicado al programa
if(cent==1)
{
    inicia_vector(centro);
    inicia_vector(centro_c);
    central(b, c);
    for(i=N_i;i<(N_i+N_c);i++)
    {
        for(j=b;j<c;j++)
        {
            if(centro[i][j]==1)
            {
                if (b==0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n", i-b, j-b,
                    centro_c[i][j], centro[i][j]);
                if (b!=0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n", i-b+1, j-b+1,
                    centro_c[i][j], centro[i][j]);
            }
        }
    }
}

```

```

    }
}

if(cent>1)
{
    inicia_vector(centro);
    inicia_vector(centro_c);
    central(b, c);
    for (i=N_i;i<(N_i+N_c);i++)
    {
        for (j=b;j<N_t;j++)
        {
            if (centro[i][j]==1)
            {
                if (b==0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n", i-b,
                    j-b, centro_c[i][j], centro[i][j]);
                if (b!=0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n", i-b+1,
                    j-b+1, centro_c[i][j], centro[i][j]);
            }
        }
    }
    for (aux=1;aux<cent-1;aux++)
    {
        for (i=N_i;i<(N_i+N_c);i++)
        {
            for (j=0;j<N_t;j++)
            {
                if (centro[i][j]==1)
                {
                    if (b==0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n",
                        i+(N_c*aux)-b, j+(N_c*aux)-b, centro_c[i][j], centro[i][j]);
                    if (b!=0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n",
                        i+(N_c*aux)-b+1, j+(N_c*aux)-b+1, centro_c[i][j], centro[i][j]);
                }
            }
        }
    }
    aux=cent-1;
    for (i=N_i;i<(N_i+N_c);i++)
    {
        for (j=0;j<c;j++)

```

```

        {
            if (centro[i][j]==1)
            {
                if (b==0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n",
                    i+(N_c*aux)-b, j+(N_c*aux)-b, centro_c[i][j], centro[i][j]);
                if (b!=0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n",
                    i+(N_c*aux)-b+1, j+(N_c*aux)-b+1, centro_c[i][j], centro[i][j]);
            }
        }
    }
}

//Escribimos la estructura final, con tantas hélices como se haya indicado al programa
if(fin!=0)
{
    for(i=(N_i+N_c);i<c;i++)
    {
        for(j=b;j<c;j++)
        {
            if(datos[i][j]==1)
            {
                if (b==0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n",
                    i+(N_c*(cent-1))-b, j+(N_c*(cent-1))-b, datos_c[i][j], datos[i][j]);
                if (b!=0) fprintf(datos_finales, "%d \t %d \t %d \t %d \n",
                    i+(N_c*(cent-1))-b+1, j+(N_c*(cent-1))-b+1, datos_c[i][j],
                    datos[i][j]);
            }
        }
    }
}

//Escribimos en fichero la estructura de hélices de la proteína modificada
fprintf(helice_final, "offset 0\n");

if (ini==2)
{
    for(i=0;i<4;i++)
    {
        fprintf (helice_final,"%d %d\n", helice_ini[i]-b, helice_fin[i]-b);
    }
}

```

```

if(ini==1)
{
    for(i=1;i<4;i++)
    {
        fprintf (helice_final,"%d %d\n", helice_ini[i]-b, helice_fin[i]-b);
    }
}

if(ini==0)
{
    for(i=2;i<4;i++)
    {
        fprintf (helice_final,"%d %d\n", helice_ini[i]-b, helice_fin[i]-b);
    }
}

if(cent!=1)
{
    i=2;
    for(j=1;j<cent;j++)
    {
        fprintf (helice_final, "%d %d\n", helice_ini[i]-b+34*j, helice_fin[i]-b+34*j);
        fprintf (helice_final, "%d %d\n", helice_ini[i+1]-b+34*j,
            helice_fin[i+1]-b+34*j);
    }
}

if(fin==3)
{
    for(i=4;i<7;i++)
    {
        fprintf (helice_final, "%d %d\n", helice_ini[i]-b+34*(cent-1),
            helice_fin[i]-b+34*(cent-1));
    }
}

if(fin==2)
{
    for(i=4;i<6;i++)
    {

```

```

        fprintf (helice_final, "%d %d\n", helice_ini[i]-b+34*(cent-1),
                helice_fin[i]-b+34*(cent-1));
    }
}

if(fin==1)
{
    for(i=4;i<5;i++)
    {
        fprintf (helice_final, "%d %d\n", helice_ini[i]-b+34*(cent-1),
                helice_fin[i]-b+34*(cent-1));
    }
}

fclose(datos_iniciales);
fclose(datos_finales);
fclose(helice_inicial);
fclose(helice_final);
}

void inicia_vector(int v[N_i][N_t])
{
    int i, j;

    for (i=0;i<N_i;i++)
    {
        for (j=0;j<N_t;j++)
        {
            v[i][j]=0;
        }
    }
}

int leeLineas (FILE *f)
{
    int i,N=0;
    char c;

    do
    {
        c=getc(f);

```

```

        if(c=='\n')
        {
            N++;
        }
    }
    while (c!=EOF);

    return(N);
}

void estructura (FILE *f, int F)
{
    int i, j, contactos;
    int m;

    for (m=0;m<F;m++)
    {
        fscanf(f, "%d %d %d", &i,&j, &contactos);

        datos[i][j]=1; //contactos sin pesos
        datos_c[i][j]=contactos; //contactos con pesos
    }
}

void leeHelice (FILE *f, int F)
{
    int i, j, m;
    char c;

    do
    {
        c=getc(f);
    }
    while (c!='\n');

    for (m=0;m<F-1;m++)
    {
        fscanf(f, "%d %d", &i, &j);

        helice_ini[m]=i;
        helice_fin[m]=j;
    }
}

```

```

    }
}

void central (int b, int c)
{
    int i, j;

    for(i=N_i;i<(N_i+N_c);i++)
    {
        for(j=0;j<(N_i+2*N_c);j++)
        {
            centro[i][j]=datos[i][j];
            centro_c[i][j]=datos_c[i][j];
        }
    }
    if(c>(N_i+N_c))
    {
        for(i=(N_i);i<(N_i+N_c);i++)
        {
            for(j=N_i;j<c;j++)
            {
                {
                    centro[i][j]=datos[i][j];
                    centro_c[i][j]=datos_c[i][j];
                }
            }
        }
    }
}

```

## 5.2. alargar\_bucles

```

#include <stdio.h>
#include <stdlib.h>

#define max 1000
#define max2 200

```

```

int datos[max][max], datos_c[max][max], datos_m[max][max], datos_c_m[max][max],
helice_ini[max2], helice_fin[max2], helice_ini_m[max2],
helice_fin_m[max2];

int leeLineas (FILE *f); //lee las líneas de los ficheros "proteina_modificada_ini.map"
(mapa de contacto de la proteína tomada como original) y "helixlist_modificada_ini.txt"
(estructura de hélices
de la preotína tomada como original)
int estructura (FILE *f, int F); //lee los contactos y su peso del fichero
"proteina_modificada_ini.map" y los copia en vectores
void inicia_vector(int v[max][max]); //inicializa las matrices de datos a 0
void inicia_vector2(int v[max2]); //inicializa los vectores de datos a 0
void leeHelice (FILE *f, int F); //lee la estructura de hélices del fichero
"helixlist_modificada_ini.txt" y la copia en un vector
void copia_vector(int v[max][max], int u[max][max]); //copia los datos de una matriz a otra
void copia_vector2(int v[max2], int u[max2]); //copia los datos de un vector a otro

int main()
{
    int N, m, aux, a, b, i, j, aux2, amino, K;
    FILE *entrada, *salida, *helice_inicial, *helice_final;

    entrada=fopen("proteina_modificada_ini.map","r");
    salida=fopen("proteina_modificada.map","w");

    helice_inicial=fopen("helixlist_modificada_ini.txt","r");
    helice_final=fopen("helixlist_modificada.txt","w");

    inicia_vector (datos);
    inicia_vector (datos_c);
    inicia_vector (datos_m);
    inicia_vector (datos_c_m);
    inicia_vector2 (helice_ini);
    inicia_vector2 (helice_fin);

    N=leeLineas(entrada);
    K=leeLineas(helice_inicial);

    rewind (entrada); //volvemos al principio del fichero
    rewind (helice_inicial);

```

```

amino=estructura(entrada, N);
leeHelice(helice_inicial, K);

aux=amino;
aux2=amino;

printf ("Indique el %cltimo amino%ccido en el que quiere introducir la modificaci%cn
entre 1 y %d, escriba %d si no quiere introducir ninguna
modificaci%cn:\n", 163, 160, 162, aux2, aux2+1, 162);
scanf ("%d",&m);

while (m<(aux2-1))
{
    printf ("Indique si quiere alargar la prote%cna (1) o acortarla (0):\n", 161);
    scanf ("%d", &a);

    if ((a!=1)&&(a!=0))
    {
        printf("La cifra introducida no es v%clida.\n", 160);
        printf ("Indique el %cltimo amino%ccido en el que quiere introducir la
modificaci%cn entre 1 y %d, escriba %d si no quiere introducir
ninguna modificaci%cn:\n", 163, 160, 162, aux, aux2, 162);
        scanf ("%d",&m);
    }

    if(a==1 || a==0)
    {
        //Copiamos los contactos en el vector auxiliar
        for(i=0;i<m;i++)
        {
            for(j=0;j<m;j++)
            {
                datos_m[i][j]=datos[i][j];
                datos_c_m[i][j]=datos_c[i][j];
            }
        }
        //Buscamos la posición del aminoácido
        for(i=0;i<K-1;i++)
        {
            if(helice_ini[i]<m)helice_ini_m[i]=helice_ini[i];

```

```

        if(helice_fin[i]<m)helice_fin_m[i]=helice_fin[i];
    }

if (a==1) //Alargamos la proteína
{
    printf("%cCu%cntos amino%ccidos sin interacci%cn quiere a%cadir?\n",
    168, 160, 160, 162, 164);
    scanf("%d", &b);
    //Se introducen tantos aminoácidos como se ha indicado,
    con contactos con segundos vecinos
    for (i=0;i<K-1;i++)
    {
        if(helice_ini[i]>=m)helice_ini_m[i]=helice_ini[i]+b;
        if(helice_fin[i]>=m)helice_fin_m[i]=helice_fin[i]+b;
    }

    for(i=0;i<m;i++)
    {
        for(j=m;j<aux2;j++)
        {
            datos_m[i][j+b]=datos[i][j];
            datos_c_m[i][j+b]=datos_c[i][j];
        }
    }

    for (i=m;i<aux2;i++)
    {
        for(j=0;j<m;j++)
        {
            datos_m[i+b][j]=datos[i][j];
            datos_c_m[i+b][j]=datos_c[i][j];
        }
        for(j=m;j<aux2;j++)
        {
            datos_m[i+b][j+b]=datos[i][j];
            datos_c_m[i+b][j+b]=datos_c[i][j];
        }
    }

    for(i=m-2;i<(m+b);i++)
    {

```

```

        datos_m[i][i+2]=1;
        datos_m[i+2][i]=1;
    }
    aux2+=b;
}

if (a==0) //Acortamos la proteína
{
    printf("%cCuantos aminoácidos quiere quitar?\n", 168, 160);
    scanf("%d",&b);

    //Se eliminan tantos aminoácidos como se ha indicado, añadiendo contactos
    con segundos vecinos de la proteína final los aminoácidos que pierden sus
    contactos con los eliminados
    for (i=0;i<K-1;i++)
    {
        if(helice_ini[i]>=m)helice_ini_m[i]=helice_ini[i]-b;
        if(helice_fin[i]>=m)helice_fin_m[i]=helice_fin[i]-b;
    }

    for(i=0;i<m;i++)
    {
        for(j=m+b;j<aux2;j++)
        {
            datos_m[i][j-b]=datos[i][j];
            datos_c_m[i][j-b]=datos_c[i][j];
        }
    }
    //printf ("m=%d, b=%d, i=%d, j=%d\n", m, b, i, j);

    for (i=m+b;i<aux2;i++)
    {
        for(j=0;j<m;j++)
        {
            datos_m[i-b][j]=datos[i][j];
            datos_c_m[i-b][j]=datos_c[i][j];
        }
        for(j=m+b;j<aux2;j++)
        {

```

```

        datos_m[i-b][j-b]=datos[i][j];
        datos_c_m[i-b][j-b]=datos_c[i][j];
    }
}

for(i=m-2;i<m;i++)
{
    datos_m[i][i+2]=1;
    datos_m[i+2][i]=1;
}

aux2-=b;
}

copia_vector (datos, datos_m);
copia_vector (datos_c, datos_c_m);

copia_vector2 (helice_ini, helice_ini_m);
copia_vector2 (helice_fin, helice_fin_m);

inicia_vector (datos_m);
inicia_vector (datos_c_m);

inicia_vector2 (helice_ini_m);
inicia_vector2 (helice_fin_m);

aux=m;
printf ("Indique el %cltimo amino%ccido en el que quiere introducir la
modificaci%cn entre 1 y %d, escriba %d si no quiere introducir ninguna
modificaci%cn:\n", 163, 160, 162, aux, aux2,162);
scanf ("%d",&m);
}
}

//Copiamos a fichero el nuevo mapa de contactos de la proteina
for(i=0;i<aux2;i++)
{
    for(j=0;j<aux2;j++)
    {
        if (datos[i][j]==1)
            fprintf(salida, "%d \t %d \t %d \t %d \n", i, j, datos[i][j], datos_c[i][j]);
    }
}

```

```

    }
}

//Copiamos a fichero la nueva estructura de hélices de la proteína
fprintf(helice_final, "offset 0\n");

for (i=0;i<K-1;i++)
{
    fprintf (helice_final, "%d %d\n", helice_ini[i], helice_fin[i]);
}

fclose(entrada);
fclose(salida);
fclose(helice_inicial);
fclose(helice_final);
}

int leeLineas (FILE *f)
{
    int i,N=0;
    char c;

    do
    {
        c=getc(f);
        if(c=='\n')
        {
            N++;
        }
    }
    while (c!=EOF);

    return(N);
}

int estructura (FILE *f, int F)
{
    int i, j, contactos, k;
    int m;

    for (m=0;m<F;m++)

```

```

    {
        fscanf(f, "%d %d %d %d", &i,&j, &contactos, &m);

        datos[i][j]=1;
        datos_c[i][j]=contactos;
    }

    return (i);
}

void inicia_vector(int v[max][max])
{
    int i, j;

    for (i=0;i<max;i++)
    {
        for (j=0;j<max;j++)
        {
            v[i][j]=0;
        }
    }
}

void inicia_vector2(int v[max2])
{
    int i;

    for (i=0;i<max2;i++)
    {
        v[i]=0;
    }
}

void leeHelice (FILE *f, int F)
{
    int i, j, m;
    char c;

    do
    {
        c=getc(f);

```

```

    }
    while (c!='\n');

    for (m=0;m<F-1;m++)
    {
        fscanf(f, "%d %d", &i, &j);

        helice_ini[m]=i;
        helice_fin[m]=j;
    }
}

void copia_vector(int v[max][max], int u[max][max])
{
    int i, j;

    for(i=0;i<max;i++)
    {
        for(j=0;j<max;j++)
        {
            v[i][j]=u[i][j];
        }
    }
}

void copia_vector2(int v[max2], int u[max2])
{
    int i;

    for(i=0;i<max2;i++)
    {
        v[i]=u[i];
    }
}

```

### 5.3. pesos\_contactos\_nn

```
#include <stdio.h>
```

```

#include <stdlib.h>

#define max 10000

int data[max][max];

void inicia_vector(int v[max][max]); //inicializa las matrices de datos a 0
int leeLineas (FILE *f); //lee las líneas del fichero "proteina_modificada.map"
(mapa de contacto de la proteína)
int estructura (FILE *f, int F); //lee los contactos y su peso del fichero
"proteina_modificada.map" y los copia en vectores

int main()
{
    int k, i, j, a;
    FILE *entrada, *salida;

    entrada=fopen("proteina_modificada.map", "r");
    salida=fopen("proteina_modificada_nw.map", "w");

    inicia_vector(data);
    k=leeLineas(entrada);
    rewind(entrada); //volvemos al principio del fichero
    a=estructura(entrada, k);

    //Copiamos a fichero únicamente los contactos, sin tener en cuenta los pesos
    for (i=0;i<(a+1);i++)
    {
        for (j=0;j<(a+1);j++)
        {
            if(data[i][j]==1) fprintf(salida, "%d \t %d\t %d\n", i, j, data[i][j]);
        }
    }

    fclose(entrada);
    fclose (salida);
}

void inicia_vector(int v[max][max])
{
    int i, j;

```

```

    for (i=0;i<max;i++)
    {
        for (j=0;j<max;j++)
        {
            v[i][j]=0;
        }
    }
}

int leeLineas (FILE *f)
{
    int i,N=0;
    char c;

    do
    {
        c=getc(f);
        if(c=='\n')
        {
            N++;
        }
    }
    while (c!=EOF);

    return(N);
}

int estructura (FILE *f, int F)
{
    int i, j, contactos, y, vecinos;
    int m;

    printf("Con vecinos (1) o sin vecinos (0):\n");
    scanf("%d",&vecinos);

    for (m=0;m<F;m++)
    {
        fscanf(f, "%d %d %d %d", &i,&j, &contactos, &y);
        if(vecinos==1) data[i][j]=1;
        if (vecinos==0)if (fabs(j-i)>1)data[i][j]=1;
    }
}

```

```

    }

    return (i);
}

```

#### 5.4. k\_van't\_hoff

```

#include <stdio.h>
#include <stdlib.h>

#define max 1000
#define kB 0.008314472 //En moles (R)

double T_m, T_min, T_max;
int aminoacidos;

double temperatura[max], calor_especifico[max], energia[max];

void inicia_vector (double v[max]); //inicializa los vectores de datos a 0
void lee1NAO (FILE *f); //del fichero "1NAO.in" lee los datos necesarios
void leeThermo (FILE *f, int j); //lee el fichero "thermo_1NAO_4.5_noNN_nw.out"
(datos termodinámicos de la proteína)

int main()
{
    int i, j, posicion;
    double cpmx, Hcal, K;
    FILE *input, *thermo;

    input=fopen ("1NAO.in", "r");
    thermo=fopen ("thermo_1NAO_4.5_noNN_nw.out", "r");

    inicia_vector(temperatura);
    inicia_vector(calor_especifico);
    inicia_vector(energia);

    lee1NAO(input);

    j=(T_max-T_min)*2+1;//Definimos el número de temperaturas estudiadas

```

```

leeThermo(thermo, j);

cpmax=calor_especifico[0];
posicion=0;

//Buscamos el máximo de calor específico y su temperatura
for(i=0;i<j;i++)
{
    if(calor_especifico[i]>cpmax)
    {
        //printf("He entrado\n");
        cpmax=calor_especifico[i];
        posicion=i;
    }
}

printf("posicion=%d\n", posicion);
printf("cpmax=%f\n", cpmax);

T_m=temperatura[posicion];
printf("Tm=%f\n", T_m);

//Calculamos la Hcal
Hcal=energia[j-1]-energia[0];
printf("Hcal=%f\n", Hcal);

//Calculamos la k de van't Hoff
K=2*T_m*sqrt(kB*cpmax)/Hcal;

printf ("K=%.20lf", K);

fclose(input);
fclose(thermo);
}

void inicia_vector (double v[max])
{
    int i;

    for (i=0;i<max;i++)

```

```

    {
        v[i]=0;
    }
}

void lee1NAO(FILE *f)
{
    double eps, q, aux1;
    fscanf (f, "%d", &aminoacidos);
    fscanf (f, "%lf %lf", &eps, &q);
    fscanf (f, "%lf %lf", &T_min, &T_max);
}

void leeThermo(FILE *f,int j)
{
    int i;
    double T, cp, aux1, aux2, aux3, aux4, ener, aux6;

    for(i=0;i<j;i++)
    {
        fscanf(f, "%lf %lf %lf %lf %lf %lf %lf %lf",
            &T, &aux1, &aux2, &aux3, &aux4, &ener, &aux6, &cp);
        temperatura[i]=T;
        energia[i]=ener;
        calor_especifico[i]=cp;
    }
}

```