Universidad Zaragoza
1542

Trabajo de Fin de Grado

Inclusión de interacciones electrostáticas en el modelo WSME del desplegamiento de proteínas, y aplicación a proteínas relevantes

Autora

Noelia Ferrer Luzón

Directores

Dr. Pierpaolo Bruscolini

David Luna Cerralbo

Facultad de Ciencias - Universidad de Zaragoza

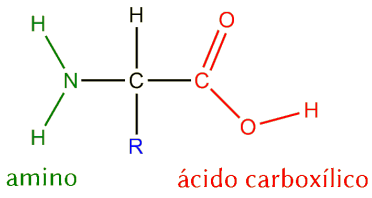
2023

Índice

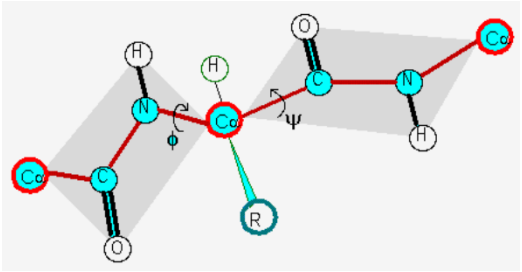
1. Introducción

Las proteínas son uno de los biopolímeros que permiten el funcionamiento de sistemas biológicos complejos. Se pueden modelar como una sucesión lineal de los 20 posibles aminoácidos (figura 1.a) existentes en todo ser vivo.

Los aminoácidos se unen entre sí por enlaces peptídicos, formando la estructura de la figura 1.b, más agua. Es decir, pierden un hidrógeno del grupo amino y un OH del grupo carboxílico. Llamamos residuos a estos aminoácidos, cuando forman un polipéptido. La estructura del enlace peptídico se debe a la distribución de cargas, y es fija: no permite que los átomos del grupo O=C-N-H roten en torno al enlace, por lo que permanecen fijos en el llamado plano peptídico (sombreado en gris en la figura 1.b). De esta forma, dos grados de libertad por residuo son suficientes para especificar la estructura polimérica global: el ángulo ϕ asociado a la rotación del plano peptídico en torno al enlace N-C_α, y el ángulo Ψ asociado a la rotación del plano peptídico en torno al enlace C-C_α. Además, dada la rigidez de todos los enlaces covalentes, la posición de cada átomo queda completamente descrita si se añade la información sobre los ángulos de rotación que determinan la configuración de las cadenas laterales (indicadas con R en la figura 1). Así, asumiendo que los enlaces covalentes sean completamente rígidos, podemos decir que una conformación de una proteína se puede especificar como un punto en un volumen $V = 2\pi^{2N} \prod_{i=1}^{N} V_i^{cl}$, con N el número de aminoácidos de la proteína, y V_i^{cl} el volumen de las configuraciones de la cadena lateral del residuo i.



(a)



(b)

Figura 1: (a) Estructura general de un aminoácido. Todo aminoácido está compuesto por un grupo amino, un grupo carboxílico, y una cadena lateral R, específica del aminoácido. (b) Ilustración de un enlace peptídico, caracterizado por el plano peptídico (área en gris), y los ángulos de rotación ϕ, Ψ, conocidos como el ángulo diedro.

Estas consideraciones sobre los grados de libertad que determinan la estructura de una proteína son relevantes porque una característica muy importante de las proteínas, que las diferencia de otros biopolímeros, es que, desde el punto de vista termodinámico, existe un macroestado de equilibrio, el estado "nativo", biológicamente

funcional, representado por un conjunto de conformaciones tan parecidas que se pueden cristalizar. Esto justifica el concepto de "estructura nativa" (o "plegada"), como la conformación en la que la proteína es funcional. El Protein Data Bank (PDB) contiene las estructuras nativas de un gran número de proteínas [1]. Por contra, en el estado desplegado (también llamado no nativo o desnaturalizado) los ángulos diedro no pueden considerarse fijos, dando lugar a diferentes conformaciones.

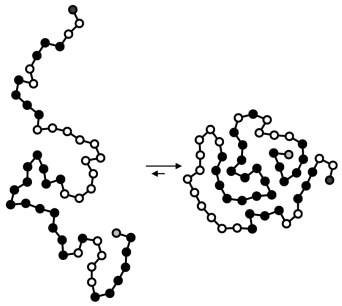


Figura 2: Representación 2D del plegamiento de una proteína (compuesta por dos tipos de residuos diferentes, en negro los apolares, y en blanco los polares) desde uno de sus posibles estados desplegados hasta su único estado nativo.

El plegamiento de proteínas es un proceso termodinámico espontáneo que consiste en el paso de un conjunto aleatorio del ensamble desnaturalizado al único conjunto, perfectamente definido, del ensamble nativo (figura 2), en un tremendo colapso del volumen del espacio de configuraciones efectivamente visitado. Es importante su estudio porque la mayoría de las proteínas únicamente tienen funcionalidad biológica en estado nativo, de forma que un mal plegamiento puede dar lugar a enfermedades como Alzheimer o Parkinson.

Generalmente, el estado nativo es el más estable a temperaturas fisiológicas, aunque su energía se diferencia en unos pocos $k_B T$ de la de las configuraciones desnaturalizadas. La estabilidad de una proteína se mide como la diferencia de energía libre entre el estado nativo y el desnaturalizado, y se puede entender como una lucha entre términos entálpicos y entrópicos, que favorecen uno u otro estado.

Por una parte, debido a que el número de conformaciones en estado desnaturalizado es mayor, la entropía comformacional de este estado es mayor. Por otra, al plegarse una proteína, lo hace de forma que los residuos apolares queden protegidos del solvente (agua), como se puede comprobar en la figura 2. Esto provoca que las moléculas de agua puedan repartirse sin limitaciones por la superficie del biopolímero, y por tanto, supone una ganancia de entropía del agua. Entonces, este efecto de corto alcance, conocido como efecto hidrofóbico, va a favorecer el estado nativo, en lo que aparece como el efecto de una interacción atractiva entre grupos apolares, y en cambio es principalmente la expresión de la entropía del agua, que domina sobre la entropía estructural del biopolímero.

En cuanto a la entalpía, fuerzas de interacción (de corto alcance) como la fuerza

de dispersión de London (en este caso trabajamos con fuerzas de Van der Waals), fuerzas electrostáticas (interacción de largo alcance entre cadenas laterales cargadas de residuos), y puentes de hidrógeno, van a aportar en estado nativo una contribución favorable.

Debido a la naturaleza estocástica del plegamiento de proteínas, es coherente tratar el problema mediante modelos de física estadística. Entre ellos se encuentra el WSME, desarrollado por Wako y Saito (WS) [2], y posteriormente, de forma independiente, por Muñoz y Eaton (ME). Es un modelo muy sencillo, de tipo Go (también conocidos como modelos centrados en el estado nativo), lo que significa que sólo importan las interacciones entre residuos que existen también en el estado nativo. El agua no aparece en el modelo, y sus efectos entálpicos y entrópicos están resumidos en la elección de los parámetros de interacción. El modelo permite, partiendo de la estructura nativa de la proteína, obtener una solución exacta para los parámetros termodinámicos del sistema.

Inicialmente, en el WSME sólo se tuvieron en cuenta interacciones de tipo Van der Waals. Posteriormente, Bruscolini y Naganathan introdujeron de forma más explicita la solvatación en el modelo, creando el WSME-S [3], donde los parámetros de interacción dependen de la temperatura de acuerdo a los efectos de la solvatación. Además, en [4], Naganathan simplifica ese modelo y añade también términos de interacción electrostática entre residuos cargados: nos referiremos a ese modelo como "modelo WSME con electrostática".

Este trabajo tiene varios objetivos:

1. En primer lugar (apartado 3.1.1), reproducir los resultados de Naganathan: Para ello, modificaremos el hamiltoniano del WSME, añadiendo términos de solvatación y electrostáticos de acuerdo a [4]. Ajustaremos los parámetros del modelo, comparando el calor específico obtenido mediante el modelo al DSC experimental de la proteína HEWL. Obtendremos una solución exacta para el calor específico para cada temperatura, a diferencia de Naganathan, que lo calcula de forma numérica. Obtener los mismos resultados que Naganathan sirve de comprobante para asegurar que el código desarrollado es correcto. Además, en el apartado 3.1.2 discutiremos si la solución encontrada es única.

2. En segundo lugar (apartado 3.1.3), averiguar hasta que punto los resultados obtenidos por Naganathan son sensibles a algunas asunciones (en principio, algo arbitrarias) del modelo, como por ejemplo, la elección de un umbral de distancia interatómica para afirmar que existe un contacto entre dos residuos en la estructura nativa.

3. A continuación, en el apartado 3.2, estudiaremos si la adopción de una definición diferente del mapa de contacto (que especifica las interacciones entre residuos), introducida en [6] y basada en el cálculo de la superficie expuesta al disolvente (ASA, "accessible surface area"), permite mejorar los resultados de Naganathan.

4. Finalmente, en el apartado 3.3, analizaremos si existe una mejora en los resultados al optimizar las curvas de ambas proteínas de forma simultánea. Éste sería un primer test para averiguar el alcance de las parametrizaciones del modelo, ya que encontrar una parametrización válida para dos proteínas homólogas es una condición necesaria para poder abordar el caso de más proteínas. Además, en el apartado 3.3.1 realizaremos un análisis del perfil de energías libres y probabilidades de plegamiento de las soluciones obtenidas.

El propósito global es identificar una estrategia óptima para poder crear un modelo que se pueda adaptar a diferentes proteínas, y que se pueda en un futuro presentar a los investigadores biofísicos, bioquímicos y biotecnólogos, como una herramienta rápida para predecir el comportamiento de las proteínas de su interés.

Para alcanzar estos objetivos, tenemos como punto de partida la literatura previa y los códigos en Fortran desarrollados para el WSME-S, sin contribuciones electrostáticas. La adaptación del código para el caso electrostático para reproducir los resultados de Naganathan, el cálculo del mapa de contactos con diferentes distancias de corte, el cálculo de la energía electrostática, los códigos de optimización de los parámetros y el análisis de los resultados, constituyen la contribución original de esta memoria.

2. Métodos

2.1. Modelo WSME

Se trata de un modelo muy estudiado porque, a pesar de su sencillez, permite obtener una solución exacta de la termodinámica del sistema en pocos segundos de cálculo computacional. Sea N el número total de residuos, este modelo, de forma muy similar al modelo Ising, asigna a cada residuo $i \in [1, N]$ dos estados posibles: plegado (nativo) o desplegado, representados por las variables $m_i = 1$, y $m_i = 0$, respectivamente. Los residuos son independientes entre sí, por lo que hay un total de 2^N conformaciones posibles. Además, sólo se permite la interacción entre residuos si tanto ellos, como los residuos que están entre ellos a lo largo de la cadena, están en estado nativo. Entonces,

de forma general, el hamiltoniano será

$$H = \sum_{i=1}^{N} \sum_{j=i}^{N} h_{ij} \prod_{k=i}^{j} m_k. \tag{1}$$

En concreto, en el modelo WSME original se propone

$$H = \sum_{i=1}^{N} \sum_{j=i}^{N} \Big(\xi_{ij} \Delta_{ij} - T \Delta S \delta_{ij} \Big) \prod_{k=i}^{j} m_k. \tag{2}$$

El primer término, con $\xi_{ij} < 0$, tiene en cuenta una energía de interacción de corto alcance (por ejemplo, interacciones de Van der Waals entre residuos, pero también las interacciones efectivas "hidrofóbicas"), únicamente si están en contacto en el estado nativo. Introducimos entonces el mapa de contactos Δ, donde $\Delta_{ij} = 0$ si los residuos i y j no se encuentran en contacto en estado nativo. En caso contrario, es el número de contactos entre los residuos i y j en estado nativo, pudiéndose calcular por varios métodos. El segundo término asigna un coste entrópico, $\Delta S < 0$, a estar el residuo en su estado nativo. Esto se puede entender considerando que cuando el residuo i se encuentra en su conformación nativa, $m_i = 1$, los ángulos (ϕ_i, ψ_i) de la cadena principal, y también los ángulos de la cadena lateral, se ven limitados a asumir valores muy concretos, y esta disminución del volumen accesible implica una disminución de la entropía conformacional.

Conociendo la energía libre entre residuos nativos se puede calcular la función de partición del sistema. Centrémonos en una cadena de residuos 1 a j. Sean $\{m_1, m_2, \cdots, m_{j-1}, m_j\}$ las conformaciones del sistema. Si $m_j = 0$, el peso estadístico de j será 1 independientemente del resto. Si $m_j = 1, m_{j-1} = 0$, será $z = \exp(\Delta S / k_B)$. Finalmente, si $m_j = 1, m_{j-1} = 1, \cdots, m_{j-k} = 1, m_{j-k-1} = 0$, el peso estadístico será $E_k z$, con $E_k = \exp(-\beta \sum_{i=j-k}^{j} \xi_{ij} \Delta_{ij})$.

A partir de Z_{j-1} se puede calcular Z_j, sumando las contribuciones debidas a $m_j = 1$. Por tanto, este método iterativo permite calcular la función de partición total del sistema.

2.2. Modelo WSME "con electrostática"

En [4], Naganathan propone un hamiltoniano, ecuación 3, que sigue la forma general, ecuación 1, del modelo WSME, para asegurar la existencia de una solución exacta. Introduce términos derivados de las principales interacciones

involucradas en el plegamiento de proteínas: interacciones de Van der Waals (E^{VdW}), electrostáticas (E^{elec}), y de solvatación (ΔG^{solv}),

$$H = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Big(E_{i,j}^{\mathrm{VdW}} + E_{i,j}^{\mathrm{elec}}(T) + \Delta G_{i,j}^{\mathrm{solv}}(T) - T\Delta S \delta_{ij} \Big) \prod_{k=i}^{j} m_k. \tag{3}$$

El primer término, ecuación 4, es equivalente al del modelo WSME. Es el producto de la energía de interacción, que se asume uniforme ($\xi_{ij} = \xi_{ji} \equiv \xi \quad \forall \{i, j\}$), por el mapa de contactos $\Delta_{ij}^{\mathrm{VdW}}$,

$$E_{i,j}^{\mathrm{VdW}} = \xi \Delta_{ij}^{\mathrm{VdW}} \mathbb{1}_{j \geq i+2}. \tag{4}$$

Siguiendo el procedimiento de Naganathan, eliminamos las contribuciones entre residuos muy próximos (i e $i+1$), ya que tienen una gran probabilidad de existir en estado nativo y estado desplegado, aportando una contribución fija. Esta aproximación se considera porque para el cálculo del calor específico nos interesa la diferencia de contactos entre estado nativo y estado desplegado.

Debido a que el agua, al ser polar, apantalla la carga de los átomos cargados, el término de las interacciones electrostáticas, ecuación 5, puede describirse mediante el modelo de Debye-Hückel,

$$E_{i,j}^{\mathrm{elec}} = \sum_{\substack{a_i \in i \\ a_j \in j}} \frac{q_{a_i} q_{a_j}}{\epsilon_{eff} \epsilon_0 r_{a_i a_j}} e^{-\kappa(T) r_{a_i a_j}} \quad (\text{con } j \geq i+1). \tag{5}$$

Se trata de un sumatorio entre los pares de átomos cargados (a_i, a_j) de los pares de residuos (i, j). q_{a_i} es la carga asociada al átomo a_i, r_{ij} la distancia entre los centros de las cargas a_i y a_j, y κ, definida como

$$\kappa^2 = \frac{8\pi e^2 I}{\epsilon_{eff} k_B T},$$

es el inverso de la longitud de Debye, que depende de la fuerza iónica del solvente, I, de la temperatura, T, de la constante dieléctrica efectiva, ϵ_{eff}, de la carga elemental, e, y de la constante de Boltzmann, k_B.

Nótese que para las interacciones electrostáticas, ecuación 5, sólo excluimos las interacciones dentro de un mismo residuo y no quitamos ningún vecino, ya que nos interesa estudiar su importancia en el plegamiento de proteínas.

Finalmente, Naganathan propone un término de solvatación, ecuación 6, que depende

de la temperatura de forma no trivial. Utilizamos para el mapa de contactos la misma aproximación utilizada en la energía de Van der Waals (quitar los vecinos i e $i+1$), y viene dado por

$$\Delta G_{i,j}^{\text{solv}} = \Delta_{ij}^{\text{solv}} \Delta C \Big((T - T_{ref}) - T \ln \frac{T}{T_{ref}} \Big) \mathbb{1}_{j \geq i+2}, \tag{6}$$

donde ΔC es el cambio en calor específico, independiente de la temperatura, debido a fijar un contacto nativo, y T_{ref} es la temperatura de referencia, fijada a 385 K.

Naganathan recurre al formalismo de matriz de transferencia del WSME [2], para calcular la función de partición del sistema. A partir de esta, calcula el calor específico de forma numérica,

$$C_p = 2RT \left(\frac{d \ln Z}{dT} \right) + RT^2 \left(\frac{d^2 \ln Z}{dT^2} \right). \tag{7}$$

En WSME se establece la energía libre del estado desplegado como nula para toda temperatura, como se puede ver en la ecuación 2. Esto implica que el calor específico sea nulo tanto a altas como a bajas temperaturas. Entonces, para ajustar el calor específico obtenido por 7 a los datos experimentales, es necesario añadir una línea de base (que se asume lineal en temperatura, fue determinada de forma empírica en [5]) correspondiente al estado desplegado, ya que no está predicha dentro del modelo. Por tanto, el calor específico calculado por Naganathan para ajustar los datos experimentales es

$$C_p^{fit} = C_p + [a + (b/1000)(T - T_{ref})] \cdot M_r, \tag{8}$$

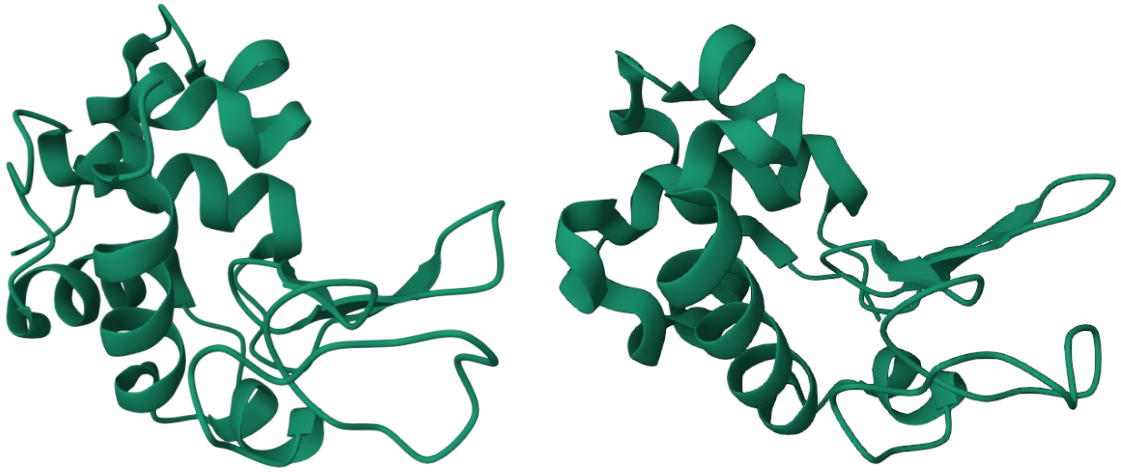
donde M_r es la masa molecular de la proteína en g/mol. a y b son parámetros ajustables que determinan la pendiente, y la ordenada al origen, de la línea de base.

En este trabajo seguimos un camino ligeramente diferente: mediante el formalismo de matriz de transferencia WSME [2], se puede obtener directamente para el calor específico, C_p, una solución exacta a cada temperatura, a partir de la definición de calor específico como derivada en temperatura del valor medio de la entalpía del sistema. Eliminamos por tanto la necesidad de calcular la derivada numérica. De nuevo, para el ajuste, será necesario sumar la línea de base de 8.

2.2.1. Parametrización del modelo

Analizaremos dos proteínas de estructura terciaria (nativa) homóloga; HEWL y apo-BLA. La estructura nativa de las proteínas se obtiene fácilmente en Protein Data

Bank (PDB) [1]. En concreto, utilizamos los modelos 1DPX para HEWL, y 1FR6 para apo-BLA (Figura 3).



(a) (b)

Figura 3: (a) Modelo 1DPX de la estructura terciaria correspondiente a la proteína HEWL [7]. (b) Modelo 1F6R de la estructura terciaria correspondiente a la proteína apo-BLA [8]

De los 20 aminoácidos posibles, sólo unos pocos están cargados. Siguiendo la propuesta de Naganathan, asignamos a los átomos NE, NH1, Y NH2 del residuo Argina una carga de 0,33 cargas electrónicas, al átomo NZ de la lisina una carga de 1, a los átomos OD1, OD2 del Aspartato una carga de -0,5, y a los átomos ND1, ND2 de la Histidina una carga de 0,5.[2] El resto de átomos se consideran no cargados. La fuerza iónica I, y la masa molecular M_r son diferentes para cada proteína, y se obtienen de la literatura (0.05 mol/L, 14400 g/mol para HEWL y 0.025 mol/L, 14200 g/mol para apo-BLA). Entonces, el modelo cuenta con 6 parámetros: ξ, ΔS, ΔC, a, b, y ϵ_{eff}. Todos, excepto ϵ_{eff}, pueden ser obtenidos del ajuste con la curva experimental. La elección de ϵ_{eff} no es trivial, pudiendo tomar valores desde 4 (interior de la proteína) a 78,5 (agua). Mediante el ejercicio sistemático de fijar el resto de parámetros y encontrar la ϵ_{eff} que mejor reproducía los datos experimentales para el sistema HEWL/BLA, Naganathan estableció $\epsilon_{eff} = 29$. Nosotros adoptamos el mismo valor.

[2]Utilizamos aquí la notación del Protein Data Bank, donde "Nxy", "Cxy", "Oxy" identifican nitrógeno, carbono y oxígeno de las cadenas laterales con una etiqueta "xy".

2.2.2. Mapa de contactos

Para el mapa de contactos, en [4] se propone usar una distancia de corte, r_c,

$$\Delta_{ij} = \begin{cases} n_{ij} & \text{si} \quad r_{a_i a_j} < r_c \\ 0 & \text{en otro caso} \end{cases}$$

Es decir, el número de contactos entre los residuos i y j, n_{ij}, es el número de contactos entre las parejas de átomos a_i de i y a_j de j, tal que su distancia, $r_{a_i a_j}$, sea menor del umbral r_c.

Aunque este mapa de contactos, mediante una buena elección de parámetros, permite reproducir los datos experimentales, la elección de la distancia de corte es compleja. Naganathan impone $r_c = 6\,\text{Å}$, y es la distancia que adoptaremos en el apartado 3.3.1, donde buscamos reproducir sus resultados. Estudiaremos también qué resultados se obtienen si modificamos esta distancia de corte a valores de 4,5 Å y 7,5 Å.

Otra forma, propuesta en [6], es considerar el cambio en área expuesta al disolvente entre estado desplegado (Unfolded) y estado nativo (Folded), $\Delta_{ij} = \text{ASA}_{ij}^U - \text{ASA}_{ij}^F$. El cálculo de las superficies se realiza con el programa ALPHASURF. Para las superficies plegadas, ASA_{ij}^F, nos interesa únicamente la superficie expuesta al disolvente en la región i, j. Sin embargo, si calculamos el área expuesta al disolvente considerando la aportación de la cadena i, j, dentro del estado nativo, no es realista, ya que exponemos superficie de los residuos i y j que realmente está protegida en estado nativo. Por tanto se calcula el área expuesta aislando la región $(i-1, j+1)$, minimizando la contribución de residuos no pertenecientes a la cadena i, j, y solucionando el problema de efecto borde. En el caso de las áreas expuestas en el estado desplegado, ASA_{ij}^U, se utiliza el aminoácido más pequeño, la glicina (Gly), para evitar efecto borde. El área expuesta se calcula como la suma de los valores debidos a Gly-X_k-Gly, con X_k el tipo de aminoácido en $k = i, .., j$.

2.2.3. Minimización de parámetros

En este trabajo se han desarrollado dos programas de optimización. El primero sigue los pasos de Naganathan y sirve para la optimización de los parámetros con los datos de una proteína. El segundo optimiza dos proteínas de forma simultánea. En el primer caso, la optimización corresponde a minimizar la distancia d,

$$d = \frac{1}{N_{\text{exp}}} \sqrt{\sum_{i=1}^{N_{\text{exp}}} \left(C_p^{fit}(i) - C_{\text{exp}}(i) \right)^2}, \tag{9}$$

donde N_{exp} es el número de puntos experimentales, y $C_p^{fit}(i)$, $C_{\text{exp}}(i)$ son, respectivamente, el calor específico calculado mediante el modelo (8), y el calor específico experimental, asociados a la temperatura $T(i)$. En el segundo caso la distancia se calcula como

$$d = \sum_{j=1}^{Np=2} \frac{1}{N_{\text{exp}}(j)} \sqrt{\sum_{i=1}^{N_{\text{exp}}(j)} \left(C_p^{fit}(i,j) - C_{\text{exp}}(i,j) \right)^2}, \tag{10}$$

donde el índice j hace referencia a la proteína, y N_p es el número de proteínas.

Para la optimización se utiliza el método de búsqueda directa "Nelder–Mead". Este algoritmo busca mínimos calculando la desviación estándar de los n+1 vértices de un politopo, siendo n la dimensión de la función f a minimizar. Para ello, en cada iteración, refleja, extiende o contrae el vértice que peor minimiza la función f, con el objetivo de mejorar esta minimización, y acabar con un politopo de vértices muy cercanos y en un mínimo local. Se considera que encuentra una solución cuando la desviación estándar de los vértices alcanza una cierta tolerancia. Siendo un algoritmo de minimización local, procuraremos efectuar una exploración más exhaustiva a través de un barrido del espacio de parámetros, para cerciorarnos que la solución encontrada sea realmente la mejor, y averiguar si hay mínimos diferentes de magnitud parecida, donde podría considerar una "degeneración"de la solución mejor. En efecto, debido a la no linealidad con la cual los parámetros entran en juego, la solución encontrada es muy sensible a la elección de las condiciones iniciales que se proporcionan al algoritmo.

Para el primer objetivo (apartado 3.1.1), usaremos el programa que minimiza los parámetros para una proteína. En concreto, minimizaremos la proteína HEWL utilizando los propios parámetros de Naganathan como parámetros iniciales para la minimización. Para determinar si la solución encontrada es única, en el apartado 3.1.2, buscaremos más soluciones, y discutiremos si existe degeneración, por el segundo método comentado a continuación, es decir, de forma iterativa.

Para el segundo (apartado 3.1.3) y tercer objetivo (apartado 3.2), ya que buscamos una comparación con los resultados de Naganathan, seguiremos usando el programa de optimización de una única proteína. Sin embargo, cómo modificamos el mapa de contactos, el principal problema será encontrar un conjunto de parámetros iniciales tales que el programa de minimización logre dar con una solución coherente, y cercana a la curva experimental, para ambas proteínas. Para ello, inicialmente, buscamos soluciones que optimicen HEWL. El espacio de parámetros donde pueden existir estas soluciones es muy extenso. Buscamos soluciones de dos formas:

1. Manteniendo a y b fijas al valor de Naganathan, y fijando dos de los tres parámetros ξ, ΔS, ΔC, buscamos un valor del parámetro variable, tal que a $T_m = 351\,\mathrm{K}$(temperatura de plegamiento experimental), los términos entrópicos del hamiltoniano igualen a los entálpicos.

2. Recorrer la caja de soluciones de forma iterativa (3 bucles en los que se van variando ξ, ΔS, ΔC), guardando los parámetros optimizados cuando la distancia a la curva experimental es aceptable.

Mediante estos métodos obtenemos un conjunto de soluciones que describen correctamente el comportamiento de la proteína HEWL. Comenzamos optimizando la proteína HEWL en lugar de la apo-BLA porque esta última cuenta con un pico experimental muy ancho (figura 4), de forma que es difícil determinar las líneas de base nativa y desnaturalizada. En cambio, el pico de la HEWL está mejor definido, y por consiguiente es más fiable el ajuste. Obtenidas soluciones para la HEWL, comprobamos cuales de estas se ajustan de forma coherente a la apo-BLA.

El segundo programa de optimización se utiliza en el cuarto objetivo (apartado 3.3). Este busca soluciones (en el rango creado previamente en los apartados anteriores) de la segunda forma mencionada anteriormente, es decir, iterativamente.

3. Resultados

3.1. Mapa de contactos a partir de una distancia de corte

En este apartado nos encargaremos de:

1. Reproducir los resultados de Naganathan, estableciendo una distancia de corte, $\mathrm{r_c}$, de 6 Å.

2. Determinar si la solución encontrada es única.

3. Estudiar la sensibilidad de los resultados a esta distancia de corte, modificándola tanto a un valor inferior ($\mathrm{r_c} = 4{,}5\,$Å) como superior ($\mathrm{r_c} = 7{,}5\,$Å).

3.1.1. Test del programa con los resultados de Naganathan: Distancia de corte de 6 Å

El objetivo de Naganathan es demostrar la importancia de la electrostática en el plegamiento de las proteínas. Lo lógico, si se busca aislar esta contribución electrostática, es pensar en mutar una proteína, cambiando su distribución superficial

de residuos. De esta forma, nos aseguramos de que esta es la única diferencia. Sin embargo, no es posible, ya que, o bien no sería estable, o bien si consiguiera ser estable, lo haría por medio de un cambio en su estructura, y por tanto del mapa de contactos. Entonces, Naganathan opta por analizar dos proteínas (HEWL, apo-BLA), que cuentan con experimentos de calorimetría, y muy semejantes en estructura, cuya diferencia principal es la distribución superficial de los residuos cargados. Para realizar dicho análisis su estrategia es ajustar los parámetros del modelo con la proteína HEWL, que tiene un pico muy bien definido, y luego averiguar si los parámetros encontrados permiten describir la otra proteína, apo-BLA. Para ello, usa la ecuación 8, y la derivada numérica (ecuación 7).

El primer objetivo de este trabajo es comprobar que el código desarrollado es correcto, verificando que obtenemos los mismos resultados que Naganathan. Entonces, usando el mismo mapa de contactos, utilizamos la ecuación 8, y optimizando el ajuste a la curva experimental que se consigue con los parámetros de Naganathan, obtenemos los parámetros de la figura 4. Con esta optimización logramos resultados ligeramente mejores que los suyos.

ξ (J/mol)	$-52{,}77$
ΔS (J/molK)	$-11{,}70$
ΔC (J/molK)	$-0{,}22$
a (J/g K)	1,57
b (J/g K^2)	5,54
d_{HEWL} (KJ/mol K)	0,095
d_{apoBLA} (KJ/mol K)	1,074

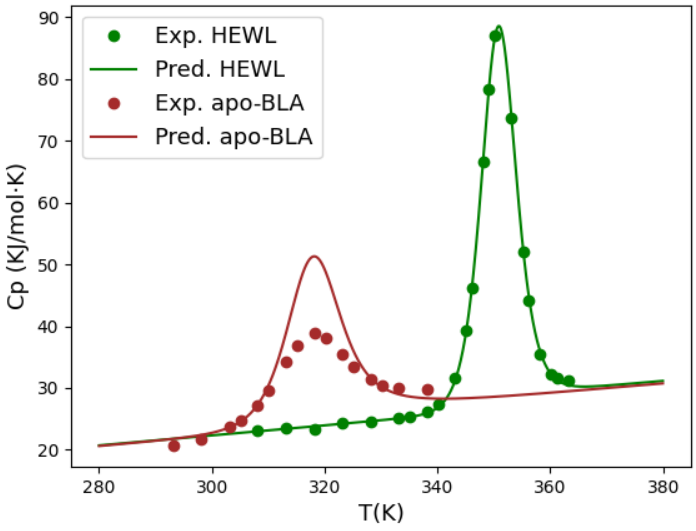


Figura 4: Perfiles de calor específico experimental para ambas proteínas (Exp. HEWL y Exp. apo-BLA) junto a la curva predicha con el modelo WSME-S con electrostática, utilizando los parámetros de la tabla a la izquierda (Pred. HEWL y Pred. apo-BLA), y una distancia de corte $r_c = 6$ Å. $d_{HEWL,\ apoBLA}$ son los valores de la distancia mínima de la ecuación 9 correspondientes.

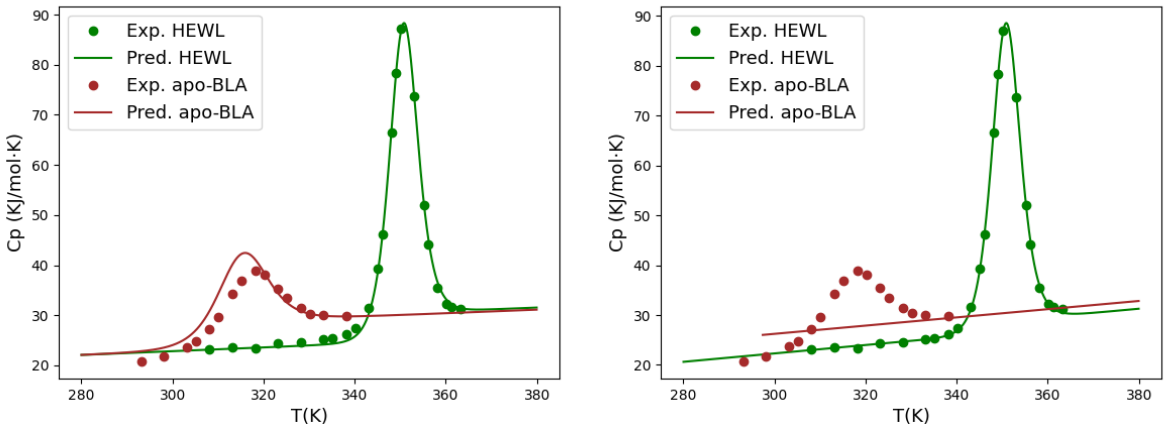
Al utilizar los parámetros obtenidos con el ajuste para la HEWL, no hay ninguna garantía de que las predicciones para la apo-BLA sean buenas. Sin embargo, la curva predicha para apo-BLA cuenta con la forma de la experimental, con una predicción

correcta de la temperatura del pico. La diferencia está en su mayor altura, y también se detecta algún problema con la linea de base del estado nativo, que resulta más alta que la experimental. Se puede ver claramente que, efectivamente, apo-BLA se pliega a una temperatura $T_m \approx 320\,\mathrm{K}$, inferior a la temperatura de plegamiento de HEWL, $T_m \approx 351\,\mathrm{K}$. Al ser proteínas tan parecidas, esto únicamente puede ser consecuencia de una distinta distribución de los residuos cargados superficiales, demostrando su importancia en el plegamiento. Además, la anchura del pico, es decir, lo abrupta que es la transición, es diferente. Naganathan demuestra, desestabilizando la proteína HEWL de forma que ambos picos coincidan, que este efecto también es debido a la electrostática superficial, y no a plegarse en rangos de temperatura distintos.

Los resultados en la figura 4 demuestran que nuestra implementación de código es correcta y valida la continuación del uso del código desarrollado.

3.1.2. Exploración del espacio de parámetros

Tras optimizar los parámetros de Naganathan, nos interesa saber si existe alguna otra solución, o una mejor. En esta búsqueda de parámetros, se han encontrado otras dos soluciones. La pregunta es si estas soluciones son realmente diferentes o son puntos en el borde de la cuenca de una única solución. Las recogemos en la tabla 1, representadas en la figura 5.



(a) (b)

Figura 5: Representación de las soluciones de la tabla 1. (a): Primera solución; (b): Segunda solución.

La primera solución, a simple vista, parece diferente a la expuesta en el apartado anterior (figura 4), tanto gráficamente (figura 5.a), como en cuestión de distancia a la curva experimental (tabla 1). No obstante, al usarla como condición inicial en la

ξ (J/mol)	ΔS (J/molK)	ΔC (J/molK)	a (J/g K)	b (J/gK^2)	d_{HEWL} (KJ/molK)	d_{apoBLA} (KJ/molK)
−62,49	−14,53	−0,52	1,92	2,55	0,163	0,625
−5192,92	−816,49	−20,39	17,07	5,82	0,093	1,708

Tabla 1: Distintas soluciones del modelo con distancia de corte de 6 Å encontradas por minimización de la curva de calor específico de la proteína HEWL.

optimización, acaba convergiendo a la solución del apartado anterior. Es decir, no son dos soluciones diferentes.

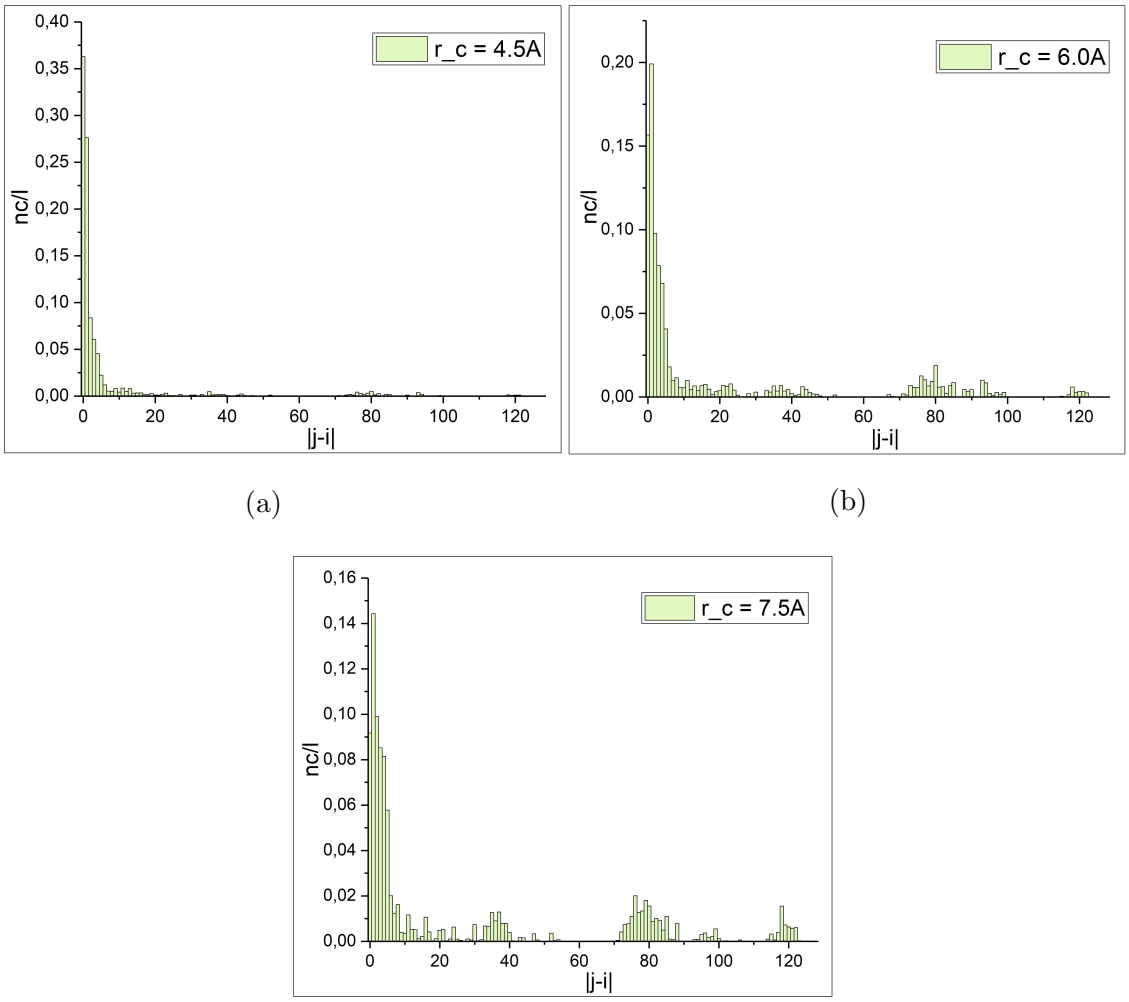
Sin embargo, este no es el caso de la segunda solución, y además, su distancia a la curva de la proteína HEWL es menor (tabla 1). Ahora bien, sus parámetros son demasiado elevados, y es incapaz de reproducir la curva de la proteína apo-BLA en absoluto (figura 5.b).

La posibilidad de diferentes soluciones para el ajuste de parámetros con la calorimetría de una sola proteína, aunque en este caso se pueda obviar observando que los parámetros de la segunda solución no son realistas y descartarla, plantea una importante cuestión de cómo es posible, en general, saber si los parámetros encontrados son los adecuados. Las posibles respuestas pasan o bien por estudiar otras variables termodinámicas relativas a la misma proteína, como mostramos en el apartado 3.3.1, siempre que existan datos experimentales con los cuales compararlas; o bien por utilizar los datos calorimétricos de otra proteína, como en la figura 5. Por tanto, es muy método que conlleva bastante tiempo, pues hay que buscar primero un mínimo para una proteína, determinar si existe degeneración y después comprobar si es capaz de ajustar correctamente por lo menos la posición del pico de otra proteína, lo cual no está garantizado. Es por esto, por lo que en el apartado 3.3 aplicamos una optimización conjunta, en búsqueda de un método que sea capaz de ajustar grupos de proteínas en poco tiempo.

3.1.3. Modificación de la distancia de corte

Otra pregunta que nos hacemos concierne la importancia del parámetro distancia de corte en la definición de mapa de contactos. Normalmente, en los trabajos que involucran la modelización del comportamiento de proteínas, debido al rango de las interacciones de Lennard-Jones, se impone una distancia de 4,5 Å. Esto significa considerar un número menor de vecinos a los que propone Naganathan. Los contactos electrostáticos no cambian, porque no dependen del mapa de contactos, pero los vecinos relevantes en las interacciones de Van der Waals y de solvatación pasan de 11089 a 3041,

una reducción del ∼73 %. Por contra, aumentar la distancia cut-off a 7,5 Å implica considerar un número mayor de vecinos, el cambio es de 11089 a 23949, un cambio del ∼ 54 %.



(a) (b)

(c)

Figura 6: Distribución del número de vecinos normalizada (nc es el número de contactos entre átomos a_i, a_j, pertenecientes a los residuos i y j, que están a una distancia $|j - i|$, l es el número de parejas de residuos a dicha distancia), para los tres casos: (a) $r_c = 4{,}5$ Å, (b) $r_c = 6{,}0$ Å, y (c) $r_c = 7{,}5$ Å para la proteína HEWL.

Sin embargo, no se produce una mera reducción o aumento en contactos, ya que, entonces, bastaría con reescalar los parámetros obtenidos en la figura 4 para obtener las nuevas curvas de calor específico. Realmente, estos cambios en distancia de corte son interesantes porque producen un cambio en la distribución de vecinos. La figura 6 recoge los histogramas para cada caso. Se observa que al reducir la distancia de corte a 4,5 Å, pasan a predominar los contactos cercanos a la diagonal, y al aumentarla a 7,5 Å aparecen nuevos contactos entre residuos lejanos, restando importancia a los contactos cercanos a la diagonal.

Nos preguntamos si el modelo WSME "con electrostática" es sensible a esta modificación en la distribución de los vecinos. Optimizando primero la proteína HEWL, y comprobando que valores se ajustan mejor a la curva experimental de la proteína apo-BLA, recogemos en la tabla 2 el conjunto de valores que mejor ajusta cada curva para estos dos últimos casos. Además, en la figura 7 representamos las curvas, junto a la de la figura 4, y en la tabla de esta figura vienen las distancias correspondientes, de acuerdo a la ecuación 10, a las curvas experimentales para todos los casos.

r_c (Å)	d (KJ/mol K)
6,0	1,17
4,5	1,72
7,5	2,54

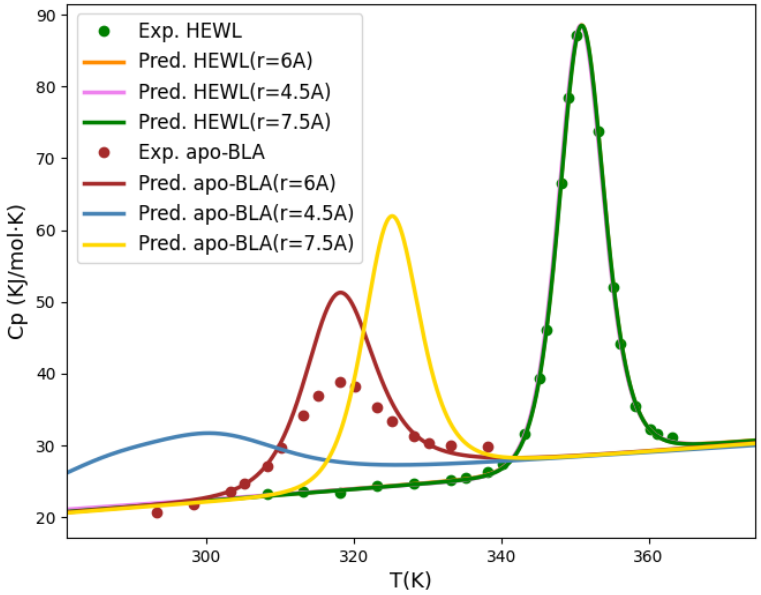


Figura 7: Tabla: Distancias de las curvas simuladas a las curvas experimentales, calculadas a partir de 10. Figura: Perfiles de calor específico experimental para ambas proteínas (Exp. HEWL y Exp. apo-BLA) junto a la curva predicha con el modelo WSME-S con electrostática (Pred. HEWL y Pred. apo-BLA), utilizando los parámetros de la tabla de la figura 4 y de la tabla 2.

r_c (Å)	ξ (J/mol)	ΔS (J/molK)	ΔC (J/molK)	a (J/g K)	b (J/g K^2)
4,5	−199,42	−12,26	−0,83	1,60	4,95
7,5	−24,36	−11,66	−0,10	1,57	5,51

Tabla 2: Parámetros del modelo obtenidos por optimización, utilizando tanto una distancia de corte de 4,5Å como de 7,5Å para crear el mapa de contactos Δ.

De la figura 7 y la tabla 2 se pueden extrapolar una serie de conclusiones:

1. Los tres métodos son capaces de reproducir el perfil de calor específico de la proteína HEWL

2. El modelo no es capaz de trabajar con una distancia de corte de 4,5 Å. Esto

es debido a qué para intentar compensar la pérdida en contactos, aumentan en módulo ξ y ΔC, provocando una reducción en la energía de interacción de Van der Waals, y en la de solvatación, mientras que la electrostática permanece constante. Como consecuencia, el coste entrópico ΔS aumenta (en módulo), y por tanto la entropía, haciendo que la energía total del sistema disminuya (en módulo), y además, que el pico aparezca antes. Ocurre lo contrario en el caso $r_c = 7{,}5$ Å.

3. La distancia de corte que mejor ajusta ambas curvas es 6Å, tanto visualmente (ya que es capaz de posicionar correctamente el pico de la apo-BLA), como numéricamente (la distancia a la curva experimental es la menor).

4. La energía de interacción de Van der Waals entre 2 carbonos es de -46,1 J/mol a una distancia de 6 Å [9]. De nuevo, esto sugiere que la consideración más correcta es la distancia de corte de 6 Å, y refleja por qué Naganathan habría escogido esta distancia.

5. Como resultado de optimizar primero la proteína HEWL, el modelo es capaz de ajustar su curva para todas las distancias, pero no consigue reproducir la curva de la proteína apo-BLA correctamente. Entonces, nos preguntamos si realmente existe un conjunto de parámetros capaz de ajustar ambas. Contestaremos a esta pregunta en el apartado 3.3.

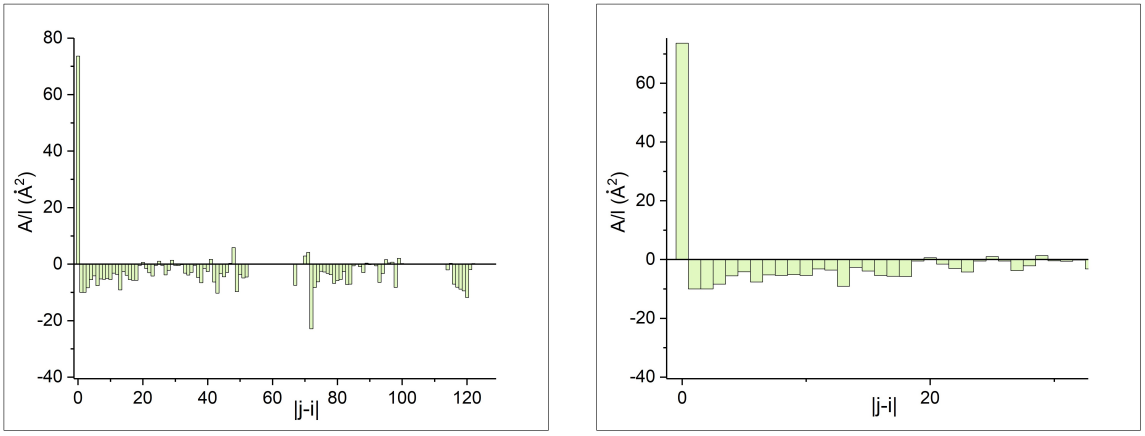
3.2. Mapa de contactos a partir de las ASAs

Como se ha comentado en la introducción, la interacción agua-residuo juega un papel importante en el plegamiento de proteínas, ya que, en estado desnaturalizado, existen más residuos apolares en contacto con el agua, haciendo que deba adoptar un posicionamiento más restrictivo. Sin embargo, en estado nativo, los residuos apolares quedan en el interior de la proteína, protegidos del agua, aumentando la entropía de esta. Esto implica que incluso los vecinos i e $i + 1$, eliminados en el modelo de Naganathan, son importantes.

Por tanto, como en el plegamiento no solo tiene que ver con las interacciones entre átomos, sino que también hay que tener en cuenta la entropía y en especial modo la del disolvente, tiene sentido físico tratar de escribir la energía eficaz de la proteína, ecuación 2, en función del cambio de superficie expuesta al disolvente en el proceso de plegamiento, redefiniendo los mapas de contacto en términos de las contribuciones que cada pareja de residuo proporciona a la diferencia entre áreas expuestas al disolvente en estado desnaturalizado y estado nativo (ASA). Además, de esta forma, conseguimos

liberarnos de un parámetro, la distancia de corte. El objetivo de este apartado es comprobar si utilizando un mapa de contactos confeccionado a partir de las ASAs mejoran los resultados con respecto a los obtenidos mediante una distancia de corte de 6 Å (figura 4).

De nuevo, buscamos conjuntos de valores que optimicen la proteína HEWL, y comprobamos si son buenas soluciones para la proteína apo-BLA. A diferencia del caso de Naganathan, en este modelo no eliminamos los vecinos i e i+1. El número de residuos considerados en contacto pasa de 11089 a 12575,39. Aunque, lo realmente significativo es el cambio en la distribución de vecinos (figura 8). En este caso, la diferencia en áreas es especialmente relevante entre átomos del mismo residuo (dando validez a la consideración de tener en cuenta los primeros vecinos en este modelo), y pasa a ser principalmente negativa para residuos distintos.



(a) (b)

Figura 8: (a) Distribución del número del cambio en áreas en función de la distancia entre los residuos i y j para la proteína HEWL. (b) Zoom de (a).

La figura 9 muestra una comparación con el caso de $r_c = 6$ Å (figura 4). Estudiamos también, el caso mixto: es decir; el mapa de contacto de las interacciones de Van der Waals, Δ^{VdW}, es el correspondiente a una distancia de corte $r_c = 6$ Å, y el de las interacciones de solvatación, Δ^{solv}, es a partir de las ASAs. Aunque de esta forma el modelo gana complejidad (al usar dos mapas de contacto distintos), es interesante su estudio, ya que, en el caso de las interacciones de Van der Waals, al ser de corto alcance, la distancia juega un papel muy importante a la hora de considerar interacción entre átomos, mientras que la interacción de solvatación, depende del área expuesta al disolvente, en este caso el agua. Los principales puntos que se pueden extraer de la figura 9 son:

1. De nuevo, los tres métodos son capaces de reproducir el perfil de calor específico de la proteína HEWL.

2. El método utilizando sólo ASAs no es capaz de predecir la posición del pico de la proteína apo-BLA. Sin embargo, es el que mejor ajusta su línea de base naturalizada, y el que cuenta con un parámetro de coste entrópico es más cercano al obtenido en experimentos de calorimetría de proteínas de dos estados: -16.5 J/mol·K [10].

3. El método de utilizar una distancia de corte de 6 Å, y el de usar ambos mapas de contacto logran localizar el pico de la apo-BLA. Sin embargo, no estiman correctamente su altura. Como muestran tanto la tabla como la figura de la figura 9, es con este último método con el que se obtiene una curva más cercana a la experimental, siendo por consiguiente el más apropiado. Sin embargo, al igual que el método con una distancia de corte, presenta un aumento en complejidad frente al método de las ASAs, ya que requiere el uso de una distancia de corte.

Método	d (KJ/mol K)
$r_c = 6$ Å	1,17
ASA	1,47
mix	1,01

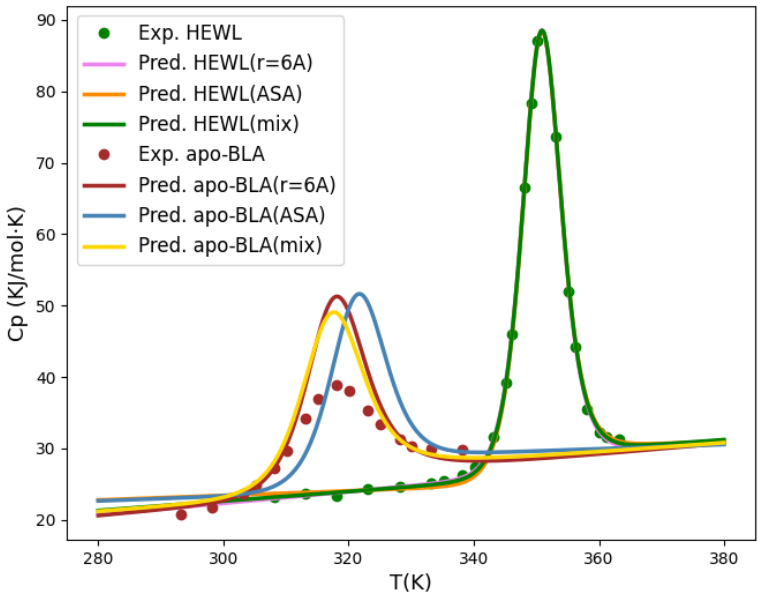


Figura 9: Tabla: Distancias de las curvas simuladas a las curvas experimentales, calculadas a partir de 10. Figura: Perfiles de calor específico experimental para ambas proteínas (Exp. HEWL y Exp. apo-BLA) junto a la curva predicha con el modelo WSME-S con electrostática (Pred. HEWL y Pred. apo-BLA), utilizando los parámetros de la tabla de la figura 4 y de la tabla 3.

Método	ξ (J/mol)	ΔS (J/molK)	ΔC (J/molKÅ^2)	a (J/g K)	b (J/g K^2)
ASA	$-46{,}81$ *	$-13{,}99$	$-0{,}36$	1,92	2,16
mix	$-55{,}42$	$-12{,}49$	$-0{,}23$	1,68	4,54

Tabla 3: Parámetros del modelo obtenidos por optimización, utilizando tanto las ASAs (ASA) como una combinación de $r_c = 6\,\text{Å}$ para VdW, y ASAs para solvatación (mix) para crear el mapa de contactos Δ. *Nótese, que aunque no esté puesto por claridad visual, las unidades de ξ en el método ASA son J/(molÅ^2).

Búsqueda de soluciones alternativas

Al igual que ocurría en el apartado 3.1.1, en la búsqueda de soluciones para el modelo con mapas de contacto a partir de las ASAs, se han encontrado varias soluciones, pero todas convergen a un única, por lo que no existe degeneración.

3.3. Optimización conjunta

En los apartados anteriores hemos comprobado que el ajuste con una única proteína es viable, pero costoso en cuestión temporal. Además, la minimización a una única proteína no permite ajustar los calores específicos de dos proteínas simultáneamente de forma satisfactoria. Por tanto, con el objetivo de demostrar si realmente existe un conjunto de parámetros que las reproduzca, y reducir el tiempo en encontrar dicho conjunto, procedemos a la minimización de ambas de forma simultánea.

Estudiamos los casos de distancia de corte de 6 Å, y mapas de contacto a partir de las ASAs. Los parámetros obtenidos quedan recogidos en la tabla 4, y representados en la figura 10. La figura 10 demuestra:

1. El modelo es capaz de encontrar un conjunto de soluciones que ajusta a la perfección la curva apo-BLA, cosa que no ha sido posible en los apartados anteriores.

2. En el caso de mapas de contacto a partir de las ASAs, localiza correctamente el pico de la proteína HEWL (pero lo infraestima), sobreestima la línea de base desnaturalizada, e infraestima la línea de base naturalizada. En el caso de los mapas de contacto a partir de la distancia de corte de 6 Å, ocurre lo mismo, a excepción de que sobreestima la posición del pico de la proteína HEWL.

3. El fallo por ambos métodos de reproducir las líneas de base puede deberse a la falta de datos experimentales de las líneas de base de la proteína apo-BLA. Puede ocurrir lo mismo en el pico de HEWL, donde no existen puntos suficientes. Esto indica que si necesitáramos mucha precisión tanto en la posición del pico

como en la líneas de base para ambas proteínas, habría que obtener más datos experimentales, y ajustar pico y líneas de base por separado.

4. Como demuestra la tabla de la figura 10, el caso de los mapas de contactos a partir de las ASAs consigue dar un ajuste ligeramente mejor, debido a que estima mejor la linea naturalizada de la HEWL.

5. Si comparamos la figura 10 con las figuras 7 y 9, al igual que las distancia de las tablas de dichas figuras, se comprueba que este método supone una clara mejora frente a los comentados anteriormente.

Método	d (KJ/mol K)
$r_c = 6$ Å	0,55
ASA	0,49

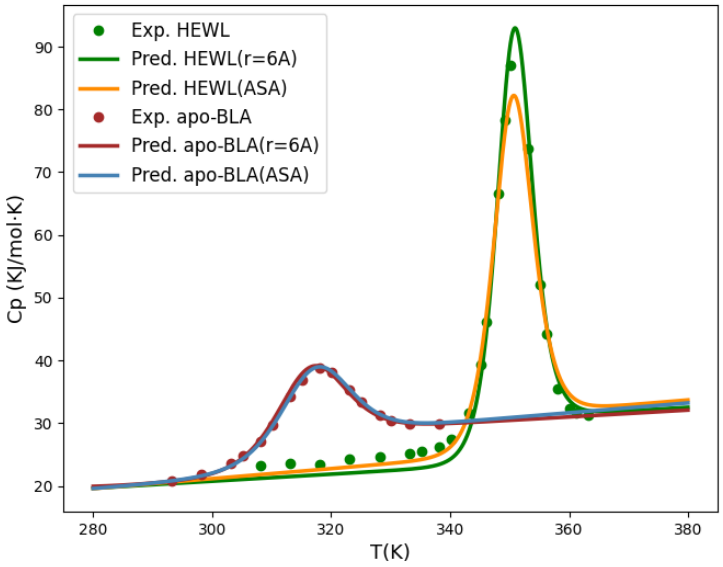


Figura 10: Tabla: Distancias de las curvas simuladas a las curvas experimentales, calculadas a partir de 10. Figura: Perfiles de calor específico experimental para ambas proteínas (Exp. HEWL y Exp. apo-BLA) junto a la curva predicha con el modelo WSME-S con electrostática (Pred. HEWL y Pred. apo-BLA), utilizando los parámetros de la tabla 4.

Método	ξ (J/mol)	ΔS (J/molK)	ΔC (J/molK)	a (J/g K)	b (J/g K^2)
$r_c = 6$ Å	$-69{,}01$	$-16{,}36$	$-0{,}67$	1,85	3,85
ASA	$-47{,}56$ *	$-14{,}24$	$-0{,}44$ *	1,75	5,47

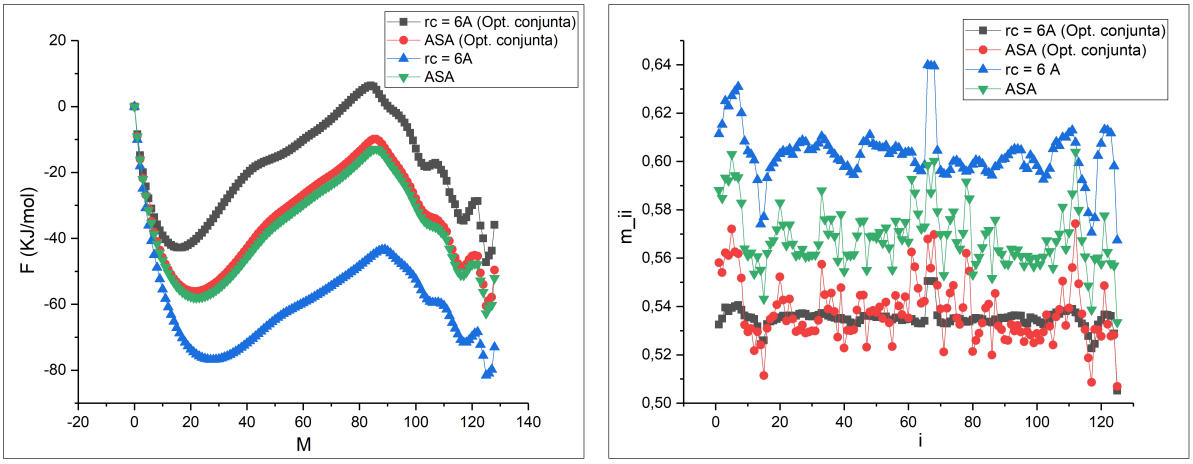
Tabla 4: Parámetros del modelo obtenidos por optimización, utilizando tanto una distancia de corte $r_c = 6$ Å como las ASAs para crear el mapa de contactos Δ. *Nótese, que aunque no esté puesto por claridad visual, las unidades de ξ, y de ΔC en el método ASA son J/(molÅ^2), y J/(molKÅ^2), respectivamente.

3.3.1. Estudio de otros parámetros termodinámicos

Finalmente, nos preguntamos si mirando a una única proteína es relevante usar un método u otro, o las soluciones obtenidas por ambos métodos son la misma para otras

variables termodinámicas. Con este objetivo estudiamos tanto los perfiles de energía libre (figura 11.a) como la probabilidad de plegamiento de cada residuo a la temperatura de transición $T_m = 350,9\,\mathrm{K}$, para la proteína HEWL (figura 11.b).

El perfil de energía libre utiliza el número de residuos nativos como parámetro de orden (o coordenada de reacción), e indica el valor de la energía libre correspondiente a cada valor de ese parámetro, permitiendo estimar la estabilidad de una proteína como diferencia de energía libre entre los mínimos, y también la barrera entre los mínimos, relacionada con la cooperatividad de la transición y con la cinética. El primer mínimo se corresponde con el número promedio de residuos plegados en estado desnaturalizado (≈20), mientras que el último se corresponde con el número promedio de residuos plegados en estado nativo (≈127). Esto se puede medir en el laboratorio y podría servir a modo de comprobante para determinar que método es más realista. En el caso de distancia de corte, observamos un desplazamiento de las curvas, y aún más importante, la altura de las barreras es diferente. Esto implica que no sirven para describir la misma proteína en el laboratorio. Sin embargo, en el caso de las ASAs, aunque no coincidan las curvas, la altura de las barreras es la misma, así que son parámetros equivalentes en términos de energía libre.



(a) (b)

Figura 11: (a) Perfil unidimensional de energía libre predicho para la proteína HEWL, a $T_m = 350,9\,\mathrm{K}$. (b) Perfil de plegamiento de la proteína HEWL, a $T_m = 350,9\,\mathrm{K}$.

En cuanto a la probabilidad de plegamiento de cada residuo i, a temperatura T_m, debería producirse una sucesión de mínimos y máximos, en torno a una probabilidad $m_{ii} = 0,5$, debido a que hay partes de la proteína plegadas (suelen ser las que tienen una mayor estructura; las hélices), y otras desplegadas. De hecho, los picos de todas las curvas coinciden, y además, se encuentran en las posiciones donde la estructura 1DPX

presenta hélices. Esta variable termodinámica si que demuestra una gran sensibilidad a los parámetros del modelo, y cabe destacar que son los métodos de optimización conjunta los que mejor se acercan al 0,5 de probabilidad, aunque el método de distancia de corte presenta picos menos pronunciados. Es decir, al considerar vecinos y no áreas, hay una mayor homogeneidad en la fracción de residuos en estado nativo a temperatura del plegamiento.

4. Conclusiones

En este trabajo se utiliza el modelo WSME, bastante popular en el estudio del plegamiento de proteínas, debido a su baja complejidad y rápida resolución computacional. Además, como input necesita únicamente la estructura nativa de la proteína, de fácil acceso en el PDB. Sin embargo, no sirve para describir el comportamiento de cualquier proteína. Esto es porque en virtud de mejorar la rapidez computacional, se realizan 2 aproximaciones, que hacen que el modelo pierda generalidad:

1. Primero, es un modelo Go-like, es decir, consideramos que únicamente los contactos que existen en estado nativo contribuyen al plegamiento. Es una aproximación drástica, pero, cómo se ha comprobado, funciona bien como primera aproximación para parametrizar la termodinámica en el plegamiento de proteínas.

2. Segundo, consideramos que existe interacción entre dos residuos únicamente si los residuos entre ellos están plegados. Es decir, tenemos en cuenta interacciones no locales, pero solo en el marco de una región estructurada de la proteína. Por esto, este método no trabaja bien con proteínas dónde su plegamiento se ve influenciado por contactos no locales separados por zonas desplegadas. Es precisamente por esta razón por la que estas dos proteínas (HEWL y apo-BLA) no serían unas buenas candidatas para analizar mediante este método, pues cuentan con puentes de disulfuro, que son enlaces covalentes entre los azufres de las cisteínas que jamás se rompen en el desplegamiento térmico. Sin embargo se escogen estas por dos motivos: queremos reproducir los resultados de Naganathan, y son muy parecidas (incluso los puntes de disulfuro están localizados en la misma zona) de forma que buscamos un análisis comparativo entre ambas.

A pesar de estas simplificaciones, es un método muy bueno para comparar homólogos. Si ajustamos primero una proteína, somos capaces de predecir una cuestión tan relevante en los perfiles de calor específico como lo es la posición del pico de la transición,

de ambas proteínas. En este caso se obtienen los mejores resultados utilizando ambos mapas de contacto (figura 9). Sin embargo, es un método que conlleva mucho tiempo, y no consigue ajustar las líneas de base de la proteína apo-BLA. Por tanto, consideramos también ajustar ambas proteínas a la vez. De esta forma conseguimos mejorar el ajuste (figura 10).

También, se ha realizado un estudio de la sensibilidad del modelo a la distancia de corte utilizada para determinar cuando dos residuos son vecinos, encontrando que es un parámetro clave. Esto se hace evidente en la figura 7. Por esta razón, y además para otorgar más sentido físico al modelo considerando no sólo la energía de interacción, sino también la entropía, se ha modificado el modelo utilizado mapas de contactos creados a partir de las ASAs. Aunque ajustando una única proteína se obtienen peores resultados que con el método de distancia de corte (figura 9), esta nueva definición es especialmente interesante si ajustamos ambas proteínas simultáneamente (figura 10), donde teniendo en cuenta los puntos comentados anteriormente, además de la escasez de datos experimentales (principalmente en los picos y en las líneas de base), es realmente extraordinario que un modelo tan sencillo sea capaz de predecir relativamente bien las líneas de base de ambas proteínas, así como la posición de sus picos, con un mismo conjunto de parámetros. Además, es un método rápido, y menos complejo, al no depender de una distancia de corte. En conclusión, este es un buen método para escoger grupos de proteínas similares, y ser capaces de predecir sus parámetros termodinámicos en el plegamiento de proteínas.

Finalmente, el modelo desarrollado es muy versátil, de forma que permite añadir cambios según el tipo de proteínas a analizar o el ambiente en el que se encuentran. En un futuro, podría mejorarse teniendo en cuenta la cantidad de desnaturalizante presente en el medio, o distintos mapas de contactos. Por ejemplo, se podría estudiar más en detalle el mapa de contactos mixto: donde los contactos de Van der Waals se establecen a través de una distancia de corte, mientras que los de solvatación a través de las ASAs. También, podrían añadirse nuevas interacciones al modelo, relevantes en el plegamiento, como lo son los puentes de disulfuro, o los de hidrógeno.

Bibliografía

[1] Bank, R. P. D. (s.f.). RCSB PDB: Homepage: `https://www.rcsb.org/`

[2] Wako, H.; Saito, N. (1978). Statistical Mechanical Theory of Protein Conformation. I. General Considerations and Application to Homopolymers. Journal of the Physical Society of Japan, 44(6), 1931-1938.

[3] Bruscolini, P., & Naganathan, A. N. (2011). Quantitative Prediction of Protein Folding Behaviors from a Simple Statistical Model. Journal of the American Chemical Society, 133(14), 5372-5379.

[4] Naganathan, A. N. (2012). Predictions from an Ising-like Statistical Mechanical Model on the Dynamic and Thermodynamic Effects of Protein Surface Electrostatics. Journal of Chemical Theory and Computation, 8(11), 4646-4656.

[5] Gómez, J., Hilser, V. J., Xie, D., & Freire, E. (1995). The heat capacity of proteins. Proteins, 22(4), 404-412.

[6] Hutton, R. S., Wilkinson, J. S., Faccin, M., Sivertsson, E. M., Pelizzola, A., Lowe, A. A., Bruscolini, P., & Itzhaki, L. S. (2015). Mapping the Topography of a Protein Energy Landscape. Journal of the American Chemical Society, 137(46), 14610-14625.

[7] Weiss, M. S., Palm, G. J., & Hilgenfeld, R. (2000). Crystallization, structure solution and refinement of hen egg-white lysozyme at pH 8.0 in the presence of MPD. Acta Crystallographica Section D-biological Crystallography, 56(8), 952-958.

[8] Chrysina, E. D., Brew, K., & Acharya, K. R. (2000). Crystal Structures of Apo- and Holo-bovine α-Lactalbumin at 2.2-Å Resolution Reveal an Effect of Calcium on Inter-lobe Interactions. Journal of Biological Chemistry, 275(47), 37021-37029.

[9] Cornell, W. D., Cieplak, P., Bayly, C. I., Gould, I. R., Merz, K. M., Ferguson, D. M., Spellmeyer, D. C., Fox, T., Caldwell, J. W., & Kollman, P. A. (1995). A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. Journal of the American Chemical Society, 117(19), 5179-5197.

[10] Robertson, A. D., & Murphy, K. P. (1997). Protein structure and the energetics of protein stability. Chemical Reviews, 97(5), 1251-1268.

Anexos

A. Código

A.1. Cálculo del mapa de contactos

El código está escrito en Python y tiene por input la estructura nativa de la proteína, descargada del PDB.

Como output obtenemos dos ficheros:

1. **Mapa de contactos a partir de** r_c: Fichero con 3 columnas (i, j, Δ_{ij}). Las dos primeras representan una pareja de residuos, y la última, el número de contactos a una distancia menor a r_{cut} entre ellos.

2. **Mapa de contactos electrostáticos**: Fichero con 5 columnas $(i, j, a_i, a_j, r_{a_i a_j})$. De nuevo, las dos primeras son una pareja de residuos. La 3 y 4 se corresponden con la carga de los átomos no neutros pertenecientes a los residuos, y la última, la distancia entre los centros de las cargas.

```
from Bio.PDB import*
import gemmi
import numpy as np

#PDB de la estructura, número N de residuos, distancia de corte r_cut
f_in="henLyzHmod.pdb"
N = 130
r_cut = 6.0

#Ficheros
file = open("1DPX.map", "w")
file2 = open("1DPX_elec.map","w")

#Inicialización
j=0
i=0
Nat=0
nombre=[0 for i in range(Nat)]
residuo=[0 for i in range(Nat)]
```

```
carga = [0 for i in range(Nat)]
x=[0 for i in range(Nat)]
y=[0 for i in range(Nat)]
z=[0 for i in range(Nat)]
delta = np.zeros((N, N), dtype=int)
dist = np.zeros((N, N))

#Almacenamos la estructura del PDB en forma de lista
st = gemmi.read_structure(f_in)
st.remove_ligands_and_waters()
print(st)

#Cálculo del número de átomos Nat en la estructura
for model in st:
        for chain in model:
            for residue in chain:
                for atom in residue:
                    Nat=Nat+1
Nat = Nat +2

#Bucle en átomos. asignamos a cada átomo su nombre,residuo,posición
for model in st:
    for chain in model:
        for residue in chain:
            for atom in residue:
                match residue.name:
                    case 'ARG':
                        if(atom.name=='NE' or atom.name=='NH1'
                           or atom.name=='NH2'):
                            carga[atom.serial] = 0.33
                    case 'LYS':
                        if(atom.name=='NZ'):
                            carga[atom.serial] = 1.0
                    case 'ASP':
                        if(atom.name=='OD1' or atom.name=='OD2'):
                            carga[atom.serial] = - 0.5
                    case 'GLU':
```

```
                        if(atom.name=='OE1' or atom.name=='OE2'):
                            carga[atom.serial]= - 0.5
                    case 'HIS':
                        if(atom.name=='ND1' or atom.name=='NE2'):
                            carga[atom.serial]= + 0.5
                nombre[atom.serial]=atom.name
                residuo[atom.serial]=int(residue.seqid.num)
                x[atom.serial]=atom.pos.x
                y[atom.serial]=atom.pos.y
                z[atom.serial]=atom.pos.z

#Bucle en átomos. Creamos el mapa elctrostático considerando
#todos los contactos (i=j,i=j+1 incluidos)
for i in range(Nat-1):
    for j in range(i+1,Nat):
        d = np.sqrt((x[i]-x[j])**2+(y[i]-y[j])**2+(z[i]-z[j])**2)
        if(d<r_cut):
            delta[residuo[i], residuo[j]] += 1
        if(carga[i]!=0 and carga[j]!=0):
            print(int(residuo[i]),int(residuo[j]),carga[i],
                  carga[j],d,file = file2)

#Bucle en residuos. Mapa VdW: contactos i>j+2. Mapa solvatación:
#todos los contactos
for i in range(1,N-2):
    for j in range(i+2,N):
        if (delta[i][j]!=0):
            print(i,j,int(delta[i][j]),file = file)
```

A.2. Cálculo de la función de partición y del calor específico

El código está escrito en Fortran y tiene por input:

```
N  !número de residuos
N1 !número de contactos eléctricos
Mw !Masa de la proteína
csi DeltaS eps_eff I DeltaC a b !parámetros optimizables
Tmin, Tmax, deltaT !Temperatura inicial, final, paso de T
cmapfile !mapa de contactos por distancia de corte
```

```
elecmapfile !mapa de contactos eléctricos
solvmapfile !mapa de contactos de acuerdo a las ASAs
MapasContacto !Selección del mapa de contactos
```

Como output obtenemos un fichero de calores específicos a cada temperatura.

Utilizamos un código preexistente del modelo WSME sin electrostática. Cuenta principalmente con 3 subrutinas: read_init (lee los inputs), calc_e_phi (calcula las energías), y calc_thermo (calcula el calor específico mediante el formalismo de matriz de transferencia). Hemos modificado la función calc_e_phi, para introducir los nuevos términos en la energía y calor específico debidos a la electrostática, así como la subrutina read_init, para obtener los nuevos datos requeridos por el modelo electrostático:

```
program WSME_genTden
  use  defreal
  use protdep_par
  use globalpar
  use phys_const
  implicit none
  real (kind=db)::parv(nparmax)
  real(kind=db):: Phi(3),natbase(3),Mavg
  real(kind=db), allocatable :: e(:,:,:),F(:),S(:,:)
  real(kind=db), allocatable :: m(:,:),nu(:,:)
  real(kind=db):: FreeonRT,EonRT,EnthonRT,logZeta
  real(kind=db):: T,fracfold,M0,Minf,ConR
  integer:: i,j,iM

  read (*,*) N
  read (*,*) N1
  allocate(delta(5,1:N,1:N))
  allocate(v(N1,5))
  allocate(e(3,N,N),F(0:N),S(0:N+1,0:N+1))
  allocate(m(0:N+1,0:N+1),nu(1:N,1:N))

  call read_init(&
  !     0:
  &     parv)
  open(20,file='profthermo.dat')
      T=Tmin
      do while (T.le.Tmax)
         FreeonRT=0.
         logZeta=0.
```

```
            EonRT=0.
            ConR=0.
            Mavg=0.
            M0=1.
            Minf=0.
            call calc_e_Phi(&
                  !     I:
               &           T,parv,&
               !      O:
               &           e,Phi,natbase)
            call calc_thermo(e,logZeta,EonRT,ConR,Mavg,fracfold)
            FreeonRT=-logZeta+Phi(1)
            EnthonRT=EonRT+Phi(2)
            ConR=ConR+Phi(3)
            write(20,*) cden,T,Mavg,(Mavg-Minf)/(M0-Minf),fracfold,&
            &         FreeonRT,R*T*EnthonRT,R*(EnthonRT-FreeonRT),&
            &         R*ConR,R*Phi(3),R*natbase(3)
            T=T+deltaT
         enddo
      close (20)
      write(*,*) 'theend'
end program WSME_genTden
```

La subrutina read_init:

```
subroutine read_init(&
     !    O:
     &     parv)
  use defreal
  use phys_const
  use globalpar
  use protdep_par
  implicit none
  real(kind=db),intent(out):: parv(nparmax)
  character(len=80):: cmapfile
  character(len=80):: elecmapfile
  character(len=80):: solvmapfile
  character(len=80):: MapasContacto
  integer::  i,j,l
  real(kind=db):: difftot
  double precision s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12
  prefac_kappa = 0.5*(log(2.)+log(Navo)+2*log(qe)-27*log(10.)
```

```
  prefac_kappa = prefac_kappa -log(vac_eps0)-log(kB))
  prefac_kappa = exp(prefac_kappa)

  read(*,*) Mw
  read(*,*) (parv(i),i=1,nparmax)
  read(*,*) Tmin,Tmax,deltaT,T_ref
  read(*,*) cdenmin,cdenmax,deltacden
  read(*,*) cmapfile
  read(*,*) elecmapfile
  read(*,*) solvmapfile
  read(*,*) MapasContacto

  !VdW contact map
  delta = 0._db
  open(1,file=cmapfile)
1 read(1,*,end=2) i,j,difftot
  if (i.le.j-2) then
   delta(1,i,j) = difftot
  endif
  go to 1
2 close(1)

  !Contribuciones eléctricas
  v = 0._db
   open(31,file=elecmapfile)
   do l=1,N1
      read(31,*) v(l,1),v(l,2),v(l,3),v(l,4),v(l,5)
      !v(k,1)= residuo del primer átomo involucrado en el contatco l
      !v(k,2)=residuo del segundo átomo involucrado en el contatco l
      !v(k,3)=carga del primer átomo involucrado en el contatco l
      !v(k,4)=carga del segundo átomo involucrado en el contatco l
      !v(k,5)= distancia átomo-átomo del contacto l
   enddo
   close(31)

   !Mapa solvatación
   open(12,file=solvmapfile)
 3 read(12,*,end=4) i,j,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12
   if (i.le.j) then
    delta(2,i,j) = s9
   endif
   go to 3
```

```
 4 close(12)

 select case (MapasContacto)
 !casos para los mapas de contacto. ct=ambos cutoff,
 !ASA=ambos ASA, mix=VDW ct, solvatacion ASA
   case ('ct')
      delta(2,:,:)  = delta(1,:,:)
   case ('ASA')
      delta(1,:,:)  = delta(2,:,:)
   case ('mix')
end select

  return
end subroutine read_init
```

La subrutina calc_e_Phi:

```
subroutine calc_e_Phi(&
  !     I:
  &  T,y, &
  !     O:
  &  e,Phi,natbase)
use defreal
use phys_const
use globalpar
use protdep_par

implicit none
real(kind=db),intent(in):: y(nparmax),T
real(kind=db),intent(out):: e(3,N,N),Phi(3),natbase(3)
!     e(1,i,j)= - h(i,j)/RT= eq (37b,c)
!     e(2,i,j)=v(i,j)/RT
!     e(3,i,j)=D(i,j)/R

integer :: i,j,l
real(kind=db)::  ee(3,N,N)
!ee(1,i,j)=(Kco*q(i)q(j)/r(i,j))*exp(-kappa*r(i,j))
!Contribución eléctrica a e(1,i,j)
!ee(2,i,j)=Kco*q(i)q(j)*(1/r(i,j)-kappa/2)*exp(-kappa*r(i,j))
!Contribución eléctrica a e(2,i,j)
!ee(3,i,j)=(Kco*q(i)q(j)*kappa*(3-kappa*r(i,j))/4T)*exp(-kappa*r(i,j)
!Contribución eléctrica a e(3,i,j)
```

```
real(kind=db):: csi_nag,DeltaC,cmapd(N,N),ASAmap(N,N),DeltaS
real(kind=db):: epseff,Ionicforce,aonR,bonR,varCp,kappa,Kco

!Inicialización de arrays a 0
e=0._db
ee = 0._db
Phi=0._db
natbase=0._db

!Mapas de contactos a utilizar
cmapd=delta(1,:,:) !Mapa de dsitancia de corte
ASAmap=delta(2,:,:) !Mapa de ASAs

!Parámetros del modelo
csi_nag=y(1) !eps in J/mol
DeltaS=y(2) !s in J/molK
epseff=y(3) !eps_eff
Ionicforce=y(4) !I_solv in M
DeltaC=y(5) !DeltaC in J/molK
aonR=y(6)*Mw/R  !a in J/(g K), Mw in g/mol
bonR=y(7)*Mw/(1000*R) !b in J/(g K^2)

!Parámetros no dependientes de los átomos, se precalculan
!fuera del bucle en átomos
Kco=Kcoul/epseff
varcp=((T-T_ref)-T*log(T/T_ref))
kappa = prefac_kappa*sqrt(Ionicforce/(T*epseff))
!Contribuciones eléctricas. Bucle en átomos
do l=1,N1
  i=int(v(l,1))
  j=int(v(l,2))
  ee(1,i,j)=ee(1,i,j)+Kco*v(l,3)*v(l,4)*exp(-v(l,5)*kappa)/v(l,5)
  ee(2,i,j)=ee(2,i,j)+
  +Kco*v(l,3)*v(l,4)*exp(-v(l,5)*kappa)*(1/v(l,5)-kappa/2)
  ee(3,i,j)=ee(3,i,j)+
  +Kco*v(l,3)*v(l,4)*exp(-v(l,5)*kappa)*kappa*(3-kappa*v(l,5))/(4*T)
enddo

!Hamiltoniano completo+contribuciones C dependiente de T.
!Bucle en residuos
do i=1,N
  do j=i,N
```

```
      if(i.le.j-1) then
         e(1,i,j)= e(1,i,j)+ ee(1,i,j)
         e(2,i,j)= e(2,i,j) + ee(2,i,j)
         e(3,i,j)= e(3,i,j) + ee(3,i,j)
      endif
      if(i.le.j-2) then
        e(1,i,j)=e(1,i,j)+ csi_nag*cmapd(i,j) +
        + varcp*DeltaC*ASAmap(i,j)
        e(2,i,j)= e(2,i,j) + csi_nag*cmapd(i,j) +
        + DeltaC*(T-T_ref)*ASAmap(i,j)
        e(3,i,j)=e(3,i,j) + DeltaC*ASAmap(i,j)
      endif
      if(j.eq.i) then
         e(1,i,i)=e(1,i,i)-T*DeltaS
      endif
      e(1,i,j)= -e(1,i,j)/(R*T)
      e(2,i,j)= e(2,i,j)/(R*T)
      e(3,i,j)= e(3,i,j)/(R)
      natbase(1)=natbase(1)-e(1,i,j)
      natbase(2)=natbase(2)+e(2,i,j)
      natbase(3)=natbase(3)+e(3,i,j)
  end do
enddo

!Cálculo de las Phis
Phi(1)=aonR*((T-T0C)/T-log(T/T0C)) +
+ bonR*((T0C**2-T**2)/(2*T)+T0C*log(T/T0C))
natbase(1)=natbase(1)+ Phi(1)
Phi(2)=aonR*(T-T0C)/T+bonR*((T-T0C)**2)/(2.*T)
natbase(2)=natbase(2)+ Phi(2)
Phi(3)=aonR+bonR*(T-T0C)
natbase(3)=natbase(3)+ Phi(3)

return
end subroutine calc_e_Phi
```

A.3. Optimización de parámetros

Este código utiliza las mismas subrutinas que el código anterior, modificando únicamente el programa principal, WSME_genTden. Además, se ha introducido una nueva subrutina: dist, que calcula la distancia entre la curva simulada y la experimental.

El input también se modifica, para añadir el número de puntos experimentales, y el fichero donde se encuentran:

```
N  !número de residuos
N1 !número de contactos eléctricos
N2 !número de datos experimentales
Mw !Masa de la proteína
csi DeltaS eps_eff I DeltaC a b !parámetros optimizables
Tmin, Tmax, deltaT !Temperatura inicial, final, paso de T
cmapfile !mapa de contactos por distancia de corte
elecmapfile !mapa de contactos eléctricos
solvmapfile !mapa de contactos de acuerdo a las ASAs
expfile !fichero con los datos experimentales
MapasContacto !Selección del mapa de contactos
```

Como output obtenemos los parámetros iniciales desde los que se ha hecho la optimización, los finales, y la distancia a la curva experimental.

```
program WSME_genTden
  use  defreal
  use protdep_par
  use globalpar
  use phys_const
  include 'nlopt.f'

  external dist
  real (kind=db)::parv(nparmax),y(nparmax)
  real(kind=db):: Phi(3),natbase(3),Mavg
  real(kind=db), allocatable :: e(:,:,:),F(:),S(:,:),m(:,:),nu(:,:)
  real(kind=db):: FreeonRT,EonRT,EnthonRT,logZeta,M0,Minf,ConR
  real(kind=db):: T,cden
  real(kind=db):: fracfold
  double precision d, params3(3), grad3(3),params5(5),grad5(5)
  double precision minf_opt
  integer :: npar, flaggrad, ires, maxeval
  integer*8 opt
  REAL time_begin, time_end

  read (*,*) N !número de residuos
  read (*,*) N1 !número de contactos eléctricos
  read (*,*) N2 !número de datos experimentales
  read (*,*) flagpar !número de parámetros a minimizar
  allocate(delta(5,1:N,1:N))
```

```
allocate(v(N1,5))
allocate(e(3,N,N),F(0:N),S(0:N+1,0:N+1),m(0:N+1,0:N+1),nu(1:N,1:N)

call read_init(parv)

!*******************************
!Inicio Optimizacion
flaggrad = 0 !flaggrad=0 (para no usar gradiente)
grad = 0._db
maxeval = 5000 !número de evaluaciones permitidas
opt = 0

CALL CPU_TIME ( time_begin )!calcula el tiempo de calculo

npar = 5 !número de parámetros a optimizar

!parámetros a optimizar
params5(1) = parv(1) !epsilon
params5(2) = parv(2) !s
params5(3) = parv(5) !deltaCp
params5(4) = parv(6) !a
params5(5) = parv(7) !b

!parámetros que no se optimizan
y(1)=parv(3) !epsilon_eff
y(2)=parv(4) !I_solv
write(*,*) 'Optimizando:eps,s,dC,a,b'

call dist(d,npar,params5,grad5,flaggrad,y) !calcula distancia
!entre curva experimental y teorica

call nlo_create(opt, NLOPT_LN_NELDERMEAD, npar)
call nlo_set_min_objective(ires, opt, dist, y)
call nlo_set_maxeval(ires, opt, maxeval)
call nlo_get_maxeval(maxeval, opt)
call nlo_optimize(ires, opt, params5, minf_opt)

if (ires.lt.0) then
    write(*,*) 'fallo␣en␣la␣optimización'
else
    write(*,*) 'min␣en␣eps,␣s,␣dC=',params5(1), params5(2)
    write(*,*) 'min␣en␣dC=', params5(3)
```

```
        write(*,*) 'a,b=',params5(4),params5(5)
        write(*,*) 'd_min final = ', minf_opt
        write(*,*) 'número de iteraciones = ', eval-1
    endif

    call nlo_destroy(opt)

end program WSME_genTden

subroutine dist(d,npar,params,grad,flaggrad,y)

 use defreal
 use phys_const
 use globalpar
 use protdep_par

 implicit none
 real (kind=db)::parv(nparmax),y(nparmax)
 real(kind=db):: Phi(3),natbase(3),Mavg
 real(kind=db):: e(3,N,N)
 real(kind=db):: EonRT,logZeta,ConR
 real(kind=db):: T,cden
 real(kind=db):: fracfold
 integer :: npar, flaggrad
 double precision d, params(npar), grad(npar)
 integer :: l

 d = 0._db
 e=0._db

 do l=1,N2 !N2=n. de datos exp. Declarado en el modulo protdep_par
  T = T_exp(l) !T_exp declarado en el modulo protdep_par

  FreeonRT=0.
  logZeta=0.
  EonRT=0.
  ConR=0.
  Mavg=0.
  ConR=0.
  fracfold=0.

  call calc_e_Phi(&
```

```
     !      I:
     &            T,cden,parv,&
     !      O:
     &            e,Phi,natbase)

   call calc_thermo(e,logZeta,EonRT,ConR,Mavg,fracfold)
   ConR=ConR+Phi(3)
   d = d + ABS(R*ConR - C_exp(l))*ABS(R*ConR - C_exp(l))
   !C_exp declarado en el modulo protdep_par

   enddo

   d = sqrt(d)/(real(N2))

end subroutine dist
```

A.4. Optimización conjunta de parámetros

Finalmente, modificamos este último programa para poder introducir los datos de proteínas, y optimizar sus curvas simultáneamente. Esto implica introducir cambios en el programa principal WSME_genTden, en la subrutina dist, y en la subrutina read_init. También se ve modificado el input:

```
Np !número de proteínas
N  !número de residuos
N1 !número de contactos eléctricos
N2 !número de datos experimentales
Mw !Masa de la proteína
csi DeltaS eps_eff I DeltaC a b !parámetros optimizables
Tmin, Tmax, deltaT !Temperatura inicial, final, paso de T
cmapfile !mapa de contactos por distancia de corte
elecmapfile !mapa de contactos eléctricos
solvmapfile !mapa de contactos de acuerdo a las ASAs
MapasContacto !Selección del mapa de contactos
```

Como output, de nuevo obtenemos el conjunto de parámetros mínimizado.

```
program WSME_genTden
   use  defreal
   use protdep_par
   use globalpar
   use phys_const
   include 'nlopt.f'
```

```
external dist
real (kind=db)::parv(nparmax),y(nparmax)
real(kind=db):: Phi(3),natbase(3),Mavg
real(kind=db), allocatable :: e(:,:,:),F(:),S(:,:),m(:,:),nu(:,:)
real(kind=db):: FreeonRT,EonRT,EnthonRT,logZeta,M0,Minf,ConR
real(kind=db):: T,cden
real(kind=db):: fracfold,minf_opt,eps,ds,dC
double precision d, params3(3), grad3(3),params5(5),grad5(5)
double precision eps_i,ds_i,dC_i
integer :: npar, flaggrad, ires, maxeval
integer*8 opt,l
REAL time_begin, time_end

read(*,*) Np !número de proteínas

allocate(N(Np),N1(Np),N2(Np),Mw(Np),I_solv(Np))
allocate(T_exp(Np,N2),C_exp(Np,N2))

do i = 1,Np
   read (*,*) N(i) !número de residuos
   read (*,*) N1(i) !número de contactos eléctricos
   read (*,*) N2(i) !número de datos experimentales
enddo
read (*,*) flagpar !número de parámetros a minimizar

allocate(delta(Np,2,1:N(1),1:N(1)))
allocate(v(Np,2000,5))
allocate(e(3,2000,2000),F(0:2000),S(0:2000+1,0:2000+1)

call read_init(parv)
open(20,file='HEWLBLA_min.dat')
 !*******************************
 !Inicio Optimizacion
 flaggrad = 0 !flaggrad=0 (para no usar gradiente)
 grad = 0._db
 maxeval = 5000 !número de evaluaciones permitidas
 opt = 0
 eval=0 !número de evaluaciones

 CALL CPU_TIME ( time_begin )!calcula el tiempo de calculo
```

```
    npar = 5 !número de parámetros a optimizar
    !parámetros a optimizar
    params5(1) = parv(1) !epsilon
    params5(2) = parv(2) !s
    params5(3) = parv(5) !deltaCp
    params5(4) = parv(6) !a
    params5(5) = parv(7) !b

    !parámetros que no se optimizan
    y(1)=parv(3) !epsilon_eff
    y(2)=parv(4) !I_solv
    write(*,*) 'Optimizando␣eps,s,dC,a,b'

    call dist(d,npar,params5,grad5,flaggrad,y)
    !calcula distancia entre curva experimental y teorica

    call nlo_create(opt, NLOPT_LN_NELDERMEAD, npar)
    call nlo_set_min_objective(ires, opt, dist, y)
    call nlo_set_maxeval(ires, opt, maxeval)
    call nlo_get_maxeval(maxeval, opt)
    call nlo_optimize(ires, opt, params5, minf_opt)

    if (ires.lt.0) then
        write(*,*) 'Opt␣fallida!'
    else
        write(*,*) 'min␣en␣eps,s=', params5(1), params5(2)
        write(*,*) 'min␣en␣dC=', params5(3)
        write(*,*) 'a,b=',params5(4),params5(5)
        write(*,*) 'd_min␣final␣=␣', minf_opt
        write(*,*) 'número␣de␣iteraciones␣=␣', eval-1

    call nlo_destroy(opt)
    CALL CPU_TIME ( time_end )
    WRITE (*,*) 'Tiempo␣de␣operación=␣',time_end - time_begin

  close(20)
  write(*,*) 'theend'

end program WSME_genTden
!****************************

subroutine dist(d,npar,params,grad,flaggrad,y)
```

```
    use defreal
    use phys_const
    use globalpar
    use protdep_par

    implicit none
    real(kind=db):: parv(nparmax),y(nparmax)
    real(kind=db):: Phi(3),natbase(3),Mavg
    real(kind=db):: e(3,2000,2000)
    real(kind=db):: EonRT,logZeta,ConR
    real(kind=db):: T,cden
    real(kind=db):: fracfold
    integer :: npar, flaggrad !num parametros a optimizar, flaggrad=0
    double precision d, dA, params(npar), grad(npar),distt(Np)
    !distancia entre curvas, parametros a optimizar, gradiente=(0,0,0)
    integer :: l,k

    eval=eval+1 !número de de iteraciones

       parv(1) = params(1) !eps
       parv(2) = params(2) !s
       parv(3)= y(1) !epsilon_eff
       parv(5)= params(3) !deltaCp
       parv(6)= params(4) !a
       parv(7)= params(5) !b

    d = 0._db
    distt=0._db
    e=0._db

    do k=1,Np
        do l=1,N2(k) !N2 = número de datos exp.
          T = T_exp(k,l)
!          FreeonRT=0.
         logZeta=0.
         EonRT=0.
         ConR=0.
         Mavg=0.
         ConR=0.
         fracfold=0.
```

```
          call calc_e_Phi(&
                !     I:
             &           T,cden,parv,k,&
             !     O:
             &           e,Phi,natbase)

          call calc_thermo(e,k,logZeta,EonRT,ConR,Mavg,fracfold)
          ConR=ConR+Phi(3)
          distt(k) = distt(k) + &
          &+ABS(R*ConR - C_exp(k,l))*ABS(R*ConR - C_exp(k,l))
       enddo
  enddo

  do k=1,Np
        d = d + sqrt(distt(k))/(real(N2(k)))
        !d = sqrt(d+dA)/(real(N2)+real(N2A))
 enddo

end subroutine dist

!*****************************
subroutine read_init(&
     !    O:
     &     parv)

  use defreal
  use phys_const
 ! use phenom_const
  use globalpar
!  use protdep_phenom_par
  use protdep_par
  implicit none
  real(kind=db),intent(out):: parv(nparmax)
  character(len=80):: cmapfile(1:Np), elecmapfile(1:Np),
   character(len=80):: expfile(1:Np), solvmapfile(1:Np)
  integer::  i,j,l,k
  real(kind=db):: difftot
  double precision s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12
  prefac_kappa = 0.5*(log(2.)+log(Navo)+2*log(qe)-27*log(10.)-
  -log(vac_eps0)-log(kB))
  prefac_kappa = exp(prefac_kappa)
```

```
read(*,*) (parv(i),i=1,nparmax)
read(*,*) Tmin,Tmax,deltaT,T_ref
read(*,*) cdenmin,cdenmax,deltacden
read(*,*) (Mw(i),i=1,Np)
read(*,*) (I_solv(i),i=1,Np)
read(*,*) (cmapfile(i),i=1,np)
read(*,*) (elecmapfile(i),i=1,np)
read(*,*) (solvmapfile(i),i=1,np)
read(*,*) (expfile(i),i=1,np)
read (*,*) MapasContacto

v = 0._db
delta = 0._db
do k=1,Np
   !VdW contact map
   open(1,file=cmapfile(k))
1 read(1,*,end=2) i,j,difftot
   if (i.le.j-2) then
      delta(k,1,i,j) = difftot
   endif
   go to 1
2 close(1)

 !Contribuciones eléctricas
   open(31,file=elecmapfile(k))
   do l=1,N1(k)
      read(31,*) v(k,l,1),v(k,l,2),v(k,l,3),v(k,l,4),v(k,l,5)
   enddo
   close(31)

   !Mapa solvatación
   open(12,file=solvmapfile(k))
   3 read(12,*,end=4) i,j,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12
      if (i.le.j) then
         delta(k,2,i,j) = s9
      endif
      go to 3
   4 close(12)

enddo

T_exp=0._db
```

```
  C_exp=0._db

  do i=1,Np
     select case (MapasContacto)
     case ('ct')
       delta(i,2,:,:) = delta(i,1,:,:)
       write(*,*) 'ct␣case'
     case ('ASA')
       delta(i,1,:,:) = delta(i,2,:,:)
       write(*,*) 'ASA␣case'
     case ('mix')
       write(*,*) 'mix␣case'
    end select

    !exp data
     open(10,file=expfile(i))

     do l=1,N2(i)
        read(10,*) T_exp(i,l),C_exp(i,l)
     enddo
     close(10)
     do l=1,N2(i)
        T_exp(i,l)=T_exp(i,l)+ 273.15
     enddo
  enddo
  return

end subroutine read_init
```