Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Maestría en Ciencias de la Computación

Una herramienta de propósito general para el plegado de proteínas mediante técnicas probabilisticas

Presenta: Luis Josué Calva Rosales

Asesor: Dr. Abraham Sánchez López

Puebla, Puebla

Dedicado a mi familia y amigos

II

Agradecimientos

A mi madre Dulce Rocío Rosales Gómez que siempre ha estado ahí para mí y con la cual he tenido las mejores charlas sobre cualquier tema en mi vida, y que cuando nadie confiaba en mí, siempre me motivaba con una sonrisa para salir adelante.

A mi padre Luis Antonio Calva Diez que me ha enseñado la importancia de ser fuerte y proteger a las personas que me rodean, que me ha motivado a seguir teniendo metas y que siempre existen las oportunidad solo hay que saberlas buscar.

Agradezco a mis hermanas Delia Denice Calva Rosales y Dulce Valeria Calva Rosales que me han motivado a ser una mejor persona, que me han enseñado lo importante que es el respeto a las mujeres y con las cuales me divertí durante mi infancia entre peleas y risas.

Agradezco a mi tía Rosa Ana Rosales Gómez que aun con su enfermedad me sigue recibiendo con una sonrisa y que es un vivo ejemplo de lucha y amor a la vida.

A mis abuelos por enseñarme el valor de vivir y la experiencia que da la vejez. A mi abuela Delia Gómez Lara que siempre me ha querido y a la cual le estoy muy agradecido por sus consejos.

Agradezco a toda mi familia que me ha acompañado y motivado en todos estos años de lucha para hacerme lo que soy en especial a María Del Rosario Calva Diez, Guadalupe Carmona Calva, Andrés Arturo Calva López, Roxana López Rosales, Teresa Ylenia Rosales Ramírez, Ximena Rosales Ramírez, Pedro Ramírez Calva, Mario Ramírez Calva, en fin atodos gracias por haberme permitido sentir que tengo un hogar y más que una familia.

A mis profesores que me han ayudado a crecer y explotar mis habilidades para todos ellos mi respeto y admiración. A mi asesor Dr. Abraham Sánchez López que me ha permitido crecer como estudiante y que confió en mí en el desarrollo de este trabajo y que considero un gran amigo.

Agradezco a mi novia Victoria Niño Pineda que me ha acompañado en estos últimos 4 años a seguir luchando y que me ha enseñado la importancia de no darse por vencido.

A mis amigos que siempre han estado hay y que se han chutado mis buenos y malos momentos a todos ustedes que saben quiénes son todo mi cariño.

Agradezco al CONACYT por que sin la beca que me otorgaron durante estos dos años, este esfuerzo y trabajo no hubieran podido ser posibles.

Resumen

La importancia de este trabajo de tesis radica en el hecho de que el estudio del plegado de las proteínas a nivel molecular tiene el fin de prevenir y entender mutaciones que ponen en peligro la vida de los seres vivos. El plegado molecular de proteínas es de vital importancia para entender los factores fisiológicos externos e internos que provocan que una proteína monomérica pase de una conformación a otra.

En este trabajo de tesis se describe el desarrollo de una herramienta que calcula y simula el plegado de proteínas, utilizando técnicas probabilistas de la robótica como son PRM (Probabilistica Roadmap Methods) y RRT (Rapidly-exploring Random Trees).

Actualmente la cantidad de proteínas analizadas es escasa dado que las herramientas existentes son caras y necesitan de costosos equipos para realizar esta tarea. Por lo cual se desarrollo esta herramienta utilizando C++ como lenguaje de programación, junto con OpenGL y se consume el servicio Web DSSP para la asignación de estructuras secundarias en bancos de datos de proteínas.

Esta herramienta permite a los usuarios finales, una vez encontrado el proceso de plegado, reproducir una animación en donde se pueda observar el comportamiento energético, y así poder realizar estudios sobre sus causas y efectos sobre los seres vivos.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Estado del arte

1.1. Proteínas y aminoácidos

El término "protein" originario del griego proteios, significa "primario" o "de primer orden". El nombre fue adoptado por Jöns Berzelius en 1838 para enfatizar la importancia de esta clase de molécula. En efecto, las proteínas juegan un papel crucial, es decir sustentan la vida. Estas moléculas constituyen los factores que desencadenan los procesos fisiológicos de los seres vivos. Por ejemplo, las proteínas proveen la arquitectura de los tejidos que dan soporte a los músculos, ligamentos, tendones, huesos, pies, cabello, órganos y glándulas de los seres vivos. Sus estructuras hacen posible coordinar funciones (movimiento, regulación, etc).

Las proteínas también proporcionan los servicios fundamentales de transporte y de almacenamiento en los seres vivos, tales como: oxígeno, hierro en los músculos y células sanguíneas. Un ejemplo de proteínas estructuradas son la hemoglobina y mioglobina, que sirven para portar oxígeno en los vertebrados. La hemoglobina se encuentra en las células rojas de la sangre y es el principal portador de oxígeno en la sangre (también transporta el dióxido de carbono y los iones de hidrógeno). La mioglobina se encuentra en las células musculares, donde se almacena oxígeno y facilita el movimiento de oxígeno en el tejido muscular. El esperma de ballena depende de la mioglobina para almacenar grandes cantidades de oxígeno durante los viajes largos bajo el agua.

Las proteínas desempeñan funciones reguladoras cruciales en muchos procesos básicos fundamentales de la vida, tales como: la catálisis de reacción

(por ejemplo, la digestión), las funciones inmunológicas, hormonales, la coordinación de la actividad neuronal, células, el crecimiento de huesos, y la diferenciación celular.

1.1.1. Funciones, tamaño y variabilidad

Las moléculas de proteína vienen en una amplia gama de tamaños y se desenvuelven en muchas funciones. Las principales clases de proteínas incluyen globular, fibrosas, y proteínas de membrana. Las proteínas globulares se encuentra entre el grupo más estudiado. Recientemente se descubrió que las proteínas ribosomales forman una clase de proteínas que se pueden ordenar como proteínas globulares, con extensiones desordenadas.

Para adaptarse a su medio ambiente y función, las proteínas fibrosas (por ejemplo, la molécula de colágeno en la piel y los huesos), son generalmente insolubles en medios acuosos, se extienden en formas distintas, mientras que las proteínas globulares tienden a ser compactas.

El colágeno es una proteína con estructura de hélice hecha de fibras ordenadas en una disposición de super-hélice paralelo. Ver [29] para observar la estructura cristalina de un péptido similar al colágeno con una secuencia biológicamente relevante y un resumen de las estructuras de colágeno dilucidado hasta la fecha. La proteína globular mioglobina es altamente compacta, se organiza en un 75 % de hélices. Similarmente, la hemoglobina es un tetrámero compuesto de cuatro cadenas polipéptidas en poder de las interacciones no covalentes; cada sub-unidad de hemoglobina en humanos es muy similar a la mioglobina. En ambas proteínas se unen las moléculas de oxígeno a través de un grupo central.

Ciertamente hay algunas proteínas muy grandes, tales como la proteína titina muscular de alrededor de 27 000 aminoácidos (y la masa de 3.000 kDa), pero la proteína promedio contiene varios cientos de residuos. El tamaño de los polipéptidos se puede determinar a partir de experimentos de electroforesis en gel: la tasa de migración de la molécula es inversamente proporcional al logaritmo de su longitud. La masa de un polipéptido o proteína puede estimarse a partir de las relaciones de la movilidad de la masa establecidos para las proteínas de referencia y por mediciones de espectrometría de masas.

1.1.2. Aminoácidos y su importancia

Las proteínas y polipéptidos son compuestos de enlaces de aminoácidos. Esa composición de aminoácidos es conocida como la estructura primaria o secuencia para abreviar.

Aminoácidos

Un aminoácido es una molécula orgánica simple que consiste de un amino básico (receptor de hidrógeno), unido a un ácido (donante de hidrógeno) a través de un único átomo de carbono intermedio.

Cada aminoácido consiste de un átomo de carbono tetrahédrico central conocido como la alpha (α) del carbono (C^{α}) el cual tiene cuatro enlaces: un átomo de hidrógeno, un grupo amino receptor de átomo (NH_3^+), un grupo ácido pierde un átomo (COO^-), y una cadena lateral distintiva, o grupo R (Ver Figura 1.1).

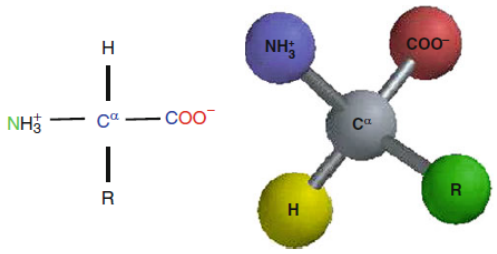


Figura 1.1: La forma general para un aminoácido (izquierda) , y la disposición tetraédrica espacial de un aminoácido (derecha).

Unión de los aminoácidos

Un polipéptido se forma cuando los aminoácidos se unen. Es decir, el carbono del grupo ácido de un aminoácido se une al nitrógeno del grupo amino de otro aminoácido para formar el péptido (C-N) vinculado con la liberación de una molécula de agua (Figura 1.2).

La repetición general de esta fórmula para un polipéptido se muestra en la Figura 1.3. Cuando el residuo aminoácido es prolina, su C^{α} está relacionado con el nitrógeno de la cadena principal peptídica a través del enlace de prolina.

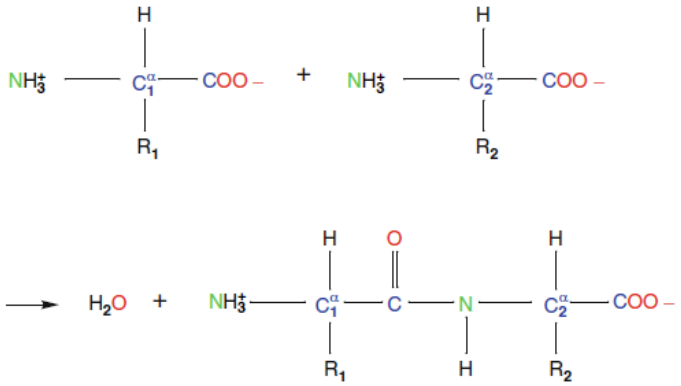


Figura 1.2: La formación de un péptido por la unión de dos aminoácidos.

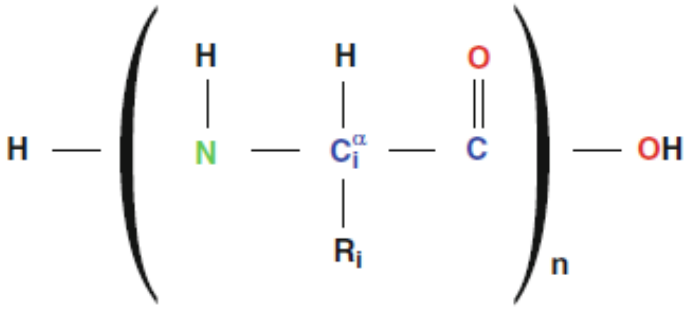


Figura 1.3: La fórmula repetida para un polipéptido.

Un modelo de aspartame, un dipéptido de ácido aspártico y fenilalanina, se muestra en la Figura 1.4. Fue descubierto por el químico James M. Schlatter accidentalmente en 1965, el cual es utilizado como endulzante artificial. En la Figura 1.4 se puede ver la la estructura primaria y secundaria del bipéptido aspartame.

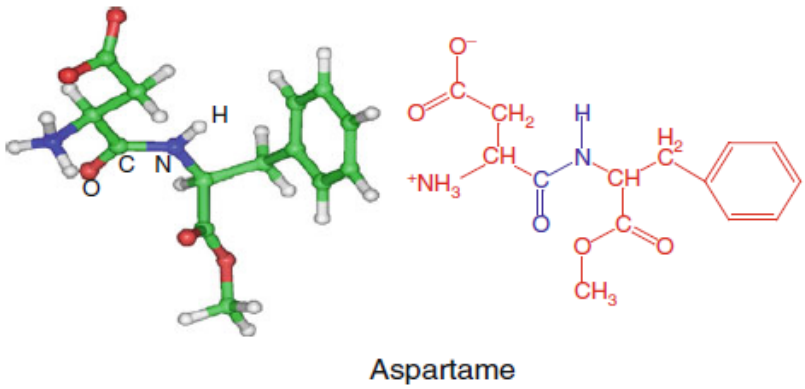


Figura 1.4: El bipéptido aspartame.

El repertorio de aminoácidos

Las formulas químicas de los 20 L-aminoácidos se muestran en la Figura 1.5, con los modelos llenos correspondientes que se muestran en la Figura 1.6. Se ilustra la abreviatura más comúnmente utilizada de tres letras para cada aminoácido, así como una agrupación en sub-familias de aminoácidos. Un mnemotécnico de una letra también se utiliza para identificar secuencias de aminoácidos, como se muestra en la Tabla 3.1.

Observando la figura 1.5, se ven las siguientes tres clasificaciones:

- **NPo**: aminoácidos con cadenas laterales estrictamente no polares (hidrófobos o insoluble en agua):
  - Ala, Val, Leu, Ile, Phe, Pro, Met, Gly, Trp, Tyr;
- **CPo**: aminoácidos con residuos polares cargados:
  - Asp, Glu, His, Lys, Arg;

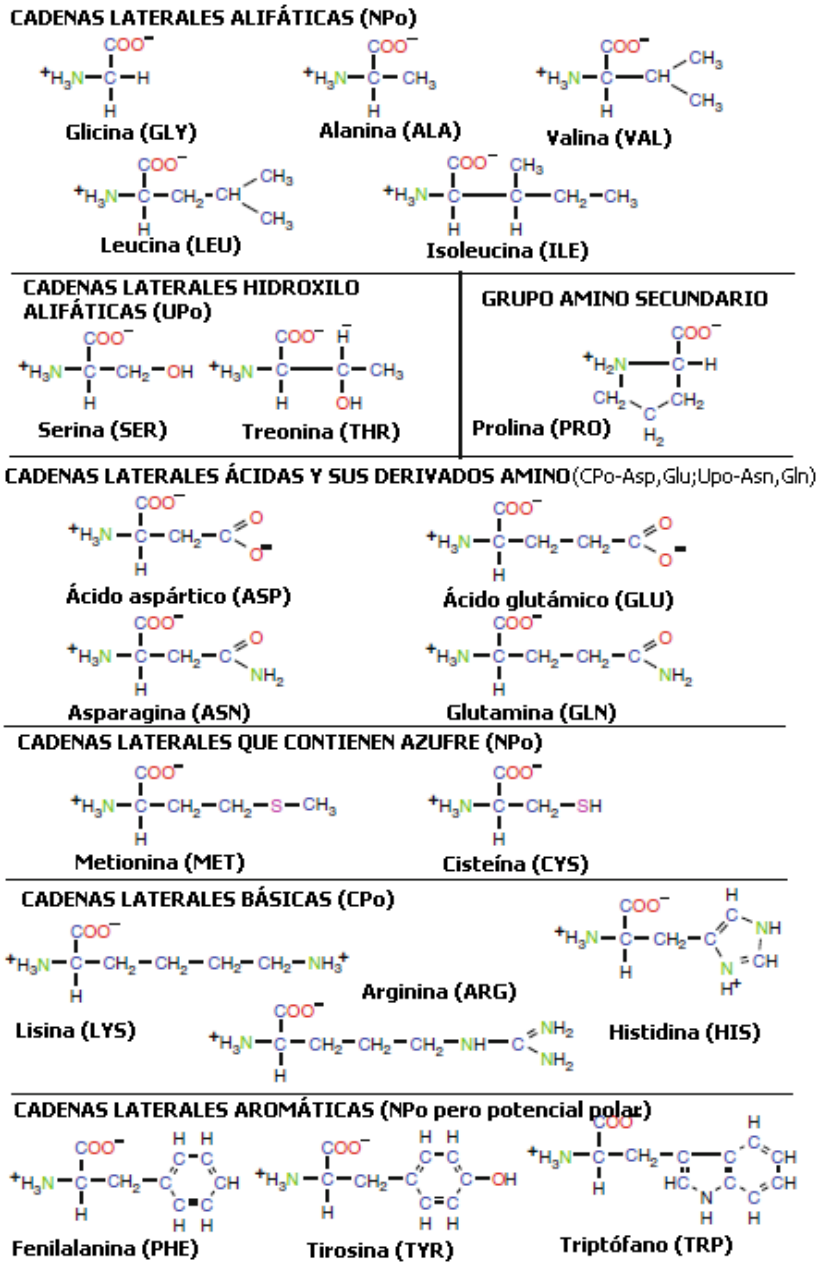


Figura 1.5: Las fórmulas químicas de los 20 aminoácidos naturales que se encuentra en pH neutro (pH de 7). Las siglas NPO, UPO, CPO denotan, respectivamente, no polares, polares sin carga y aminoácidos polares.

- **UPo**: aminoácidos con cadenas laterales polares no cargadas:
  - Ser, Thr, Cys, Asn, Gln.

Cada aminoácido tiene una combinación única de propiedades de tamaño, polaridad, los componentes cíclicos, los componentes de azufre, etc., que afectan críticamente las interacciones no covalentes y covalentes (es decir, enlaces de sulfuró) que le da a la proteína la arquitectura tridimensional (3D). Estas interacciones se originan a partir de electrostáticas, de van der Waals, hidrofóbicas, o fuerzas de enlace de hidrógeno.

Aminoácido	**Frecuencia** [%]
Alanine (Ala, A)	8.1
Arginina (Arg, R)	5.1
Asparagina (Asn, D)	5.2
Ácido aspártico (Asp, N)	4.0
Cisteína (Cys, C)	1.2
Glutamina (Gln, Q)	3.8
Ácido glutámico (Glu, E)	6.5
Glicina (Gly, G)	7.2
Histidina (His, H)	2.2
Isoleucina (Ile, I)	6.8
Leucina (Leu, L)	10.3
Lisina (Lys, K)	5.9
Metionina (Met, M)	2.5
Fenilalanina (Phe, F)	4.2
Prolina (Pro, P)	4.3
Serina (Ser, S)	6.2
Treonina (Thr, T)	5.1
Triptófano (Trp, W)	1.1
Tirosina (Tyr, Y)	3.2
Valina (Val, V)	6.9

Cuadro 1.1: Las frecuencias de aminoácidos en las proteínas sobre la base de los datos de [22] que se analizaron las proteínas 45137 de 15 taxones. Se utilizan tipos negrita y cursiva, respectivamente, para el más alto (8 %) y más baja ($= 2,5\,\%$) frecuencias.

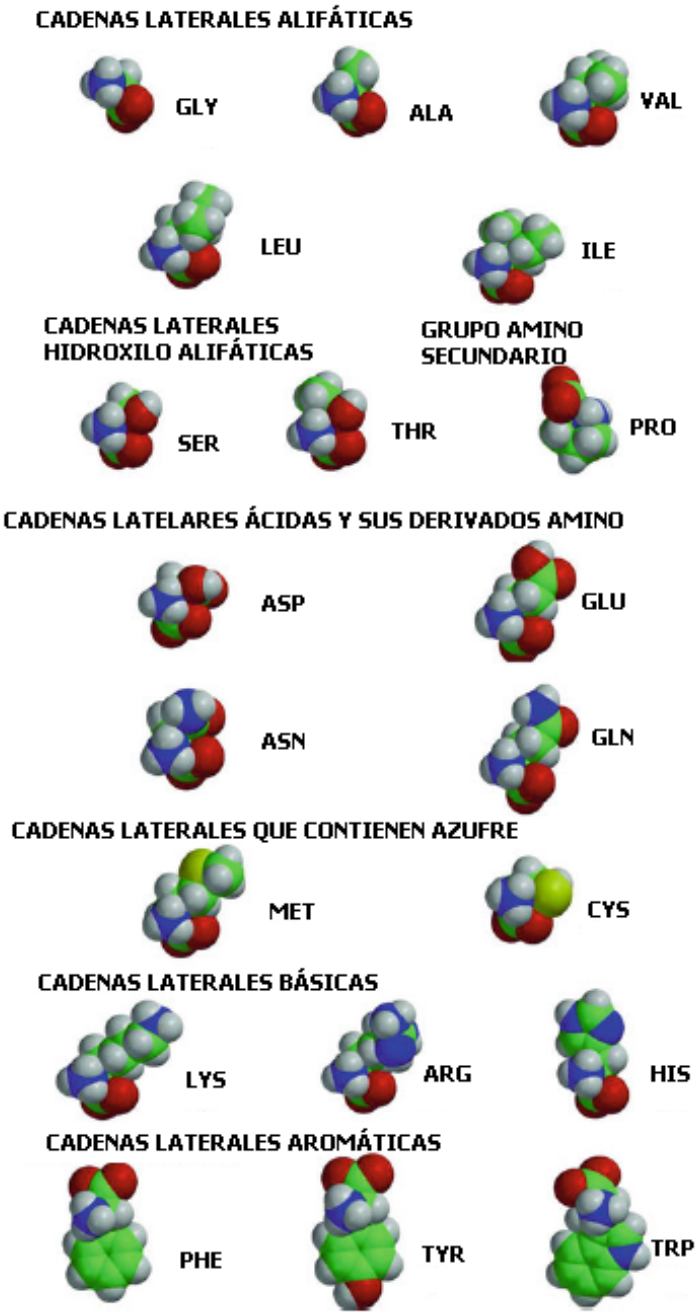


Figura 1.6: Los modelos de estructura llena de los 20 aminoácidos naturales con pH neutro (pH de 7).

1.1.3. Estructura de la proteína

Durante la creación de un gen a una proteína, la proteína es formada por la unión secuencial de aminoácidos de extremo a extremo para formar una larga cadena molecular, o **polímero**. Un polímero de aminoácidos se denomina como un **polipéptido**. Se han codificado 20 diferentes aminoácidos cuyas propiedades químicas dependen de la composición de sus **cadenas laterales**. Así, para una primera aproximación, una proteína, no es nada más que una secuencia de estos aminoácidos (o, más apropiadamente, residuos de aminoácidos, porque ambos grupos amino y ácidos pierden su acidez y base propios cuando estos son parte del polipéptido). Esta secuencia se denomina la **estructura primaria** de una proteína.

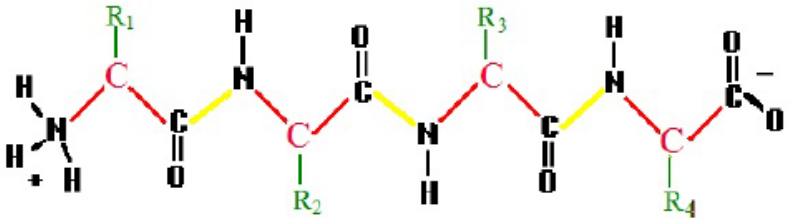


Figura 1.7: Una cadena polipéptida genérica. Los enlaces que se muestran en amarillo, conectan aminoácidos separados y se denominan **enlaces péptidos**.

La estructura primaria de una proteína es fácil de obtener de su correspondiente secuencia, así como por manipulación experimental. Desafortunadamente, la estructura primaria se relaciona de manera indirecta a la función de la proteína. Con el fin de que funcione correctamente, una proteína debe plegarse para formar una forma tridimensional específica, llamada **estructura nativa** o **conformación nativa**. La estructura tridimensional de una proteína generalmente se entiende de una manera jerárquica. La **estructura secundaria** se refiere al plegado en una pequeña parte de la proteína que forma una estructura característica. Las estructuras secundarias más comunes son la $\alpha - helices$ que se muestra en la Figura 1.8 y $\beta - sheets$ que se muestra en la Figura 1.9, una o ambas de estas pueden presentarse en casi todas las proteínas naturales.

La estructura ternaria describe elementos estructurales formados por aportar más de una estructura en distintas partes de una cadena del **dominio**. La disposición espacial de estos dominios con respecto a los otros

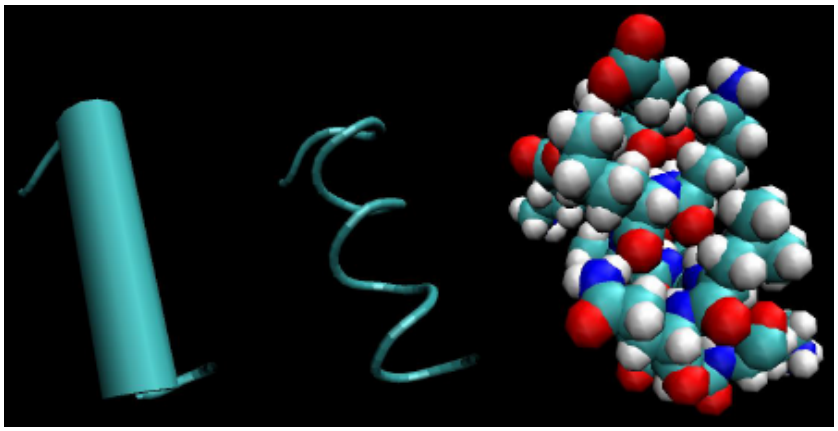


Figura 1.8: La estructura $\alpha - helice$, presenta tres diferentes representaciones. La de la izquierda es una representación típica de cartoon, en la cual la hélice se presenta como un cilindro. La representación del centro muestra enlaces guiados de la proteína. La presentación de la derecha muestra un modelo de espacio lleno, este solo modela todos los átomos (incluyendo los que se encuentran dentro de la cadena).

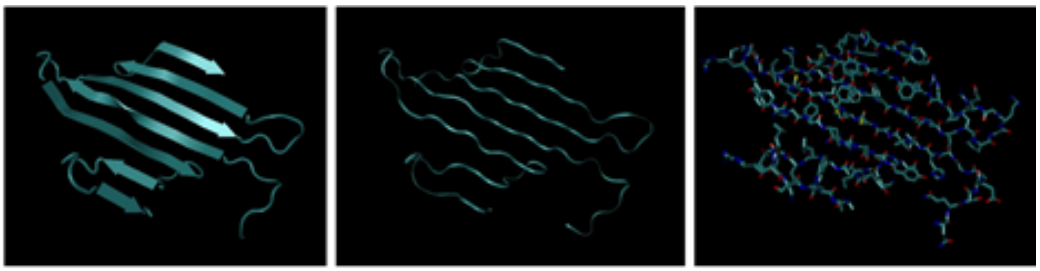


Figura 1.9: La estructura $\beta - sheet$, presenta tres diferentes representaciones. La izquierda es una representación carton, es una representación antiparalela, los segmentos adyacentes de la proteína corren en direcciones opuestas. En la parte del centro se encuentra la representación en cinta o láminas plegadas ß, debido a su forma de hebras en zig-zag. La representación de la derecha a diferencia de las otras dos representaciones, ilustra las cadenas laterales. Se toma en cuenta la alíneación de los átomos de oxígeno (rojo) y nitrógeno (azul).

es también considerada parte de la estructura terciaria. Finalmente, muchas proteínas consisten de más de un polipéptido plegado junto, y la relación espacial entre estos separa el polipéptido en cadenas, esto se denomina la estructura cuaternaria. Es importante notar que la forma nativa de una proteína es una consecuencia directa de la secuencia primaria y el ambiente químico, lo cual para la mayoría de las proteínas es una solución acuosa o bien una biológica determinada por el pH (casi neutral), o el interior aceitoso de una membrana celular. Sin embargo, no existe método computacional fiable para predecir la estructura nativa de la secuencia de aminoácidos, y este es un tema de investigación abierto. Por lo tanto, con el fin de encontrar la estructura nativa de una proteína se han propuesto diferentes métodos experimentales.

1.1.4. Métodos para determinar la estructura de las proteínas

Una estructura de una proteína es una disposición tridimensional de los átomos de tal manera que se preserve la integridad de la molécula (su conectividad). El objetivo de determinar la estructura de una proteína es encontrar un conjunto de coordenadas (x, y, z) para cada átomo de la molécula en su estado natural. Es de particular interés la estructura natural, es decir, la estructura asumida por la proteína bajo sus condiciones biológicas, así como las estructuras asumidas por la proteína cuando el proceso de interacción con otras moléculas se efectúa. A continuación se muestran los principales métodos de determinación de la estructura proteica.

Cristalografía de Rayos X

El método más comúnmente usado y usualmente con más resolución en la determinación estructural de proteínas. Para obtener una estructura por este método, los laboratorios bioquímicos necesitan obtener, una muestra cristalina muy pura de una proteína. Los rayos X son pasados por la muestra, en la que son dirigidos por los electrones de cada átomo de la proteína. La dirección del patrón es registrado, y se puede utilizar para reconstruir el patrón tridimensional de la densidad de los electrones, y por lo tanto, dentro de algún error, en la ubicación de cada átomo. Una estructura cristalina de alta resolución, con una resolución del orden de 1 a 2 Angstroms (A). Un

Angstrom es el diámetro de un átomo de hidrógeno 10^{-10} metros, o un cien millonésimo de un centímetro).

A diferencia de otros métodos de determinación de la estructura, con la cristalografía de rayos X, no hay límite fundamental en el tamaño de la molécula o complejidad para ser estudiado. Sin embargo, con el fin de que el método funcione, se debe obtener una muestra pura cristalina de una proteína. Para muchas proteínas, incluyendo muchos receptores unidos a la membrana, esto no es posible. Además, un solo experimento de cristalografía de rayos x proporciona solo información estática que es, información sobre la estructura nativa de la proteína en las condiciones experimentales particularmente usadas. Las proteínas son a menudo objetos flexibles cuando están en su estado natural, por lo que una sola estructura, si bien es útil, no da mucha información.

Resonancia Magnética Nuclear

La espectroscopia por Resonancia Magnética Nuclear (NMR) se ha utilizado recientemente como un método de determinación de la estructura de proteínas. En un experimento de NMR, un fuerte campo magnético es aplicado a una muestra de proteína a ser estudiada, forzando a los núcleos atómicos a alínearse al campo magnético. La señal se emite por un núcleo a medida que regresa a un estado no alíneado en la característica de un ambiente químico. La información sobre los átomos de dos enlaces químicos de la resonancia del núcleo puede proporcionar información muy importante acerca de la cercanía espacial de los átomos. Esta información conduce a un gran sistema de restricciones de distancias entre átomos de la proteína, que puede ser resuelto para encontrar una estructura tridimensional. La estructura obtenida por un experimento de NMR es variable y depende fuertemente de la flexibilidad de la proteína. Debido a la naturaleza del método y la flexibilidad de la proteína que puede estar siendo estudiada, puede haber muchas estructuras detectadas para una misma muestra.

La estructura determinada por NMR es generalmente limitada a proteínas más pequeñas que 25-30 kilodaltons (kDa), porque las señales de diferentes átomos empiezan a traslaparse y se vuelven más difíciles de resolver. Adicionalmente, la proteína debe estar en una concentración soluble de 0.2-0.5 mM sin agregaciones o precipitación.

Difracción de electrones

La difracción de electrones trabaja bajo el mismo principio de la cristalografía de rayos X, pero en lugar de rayos X, los electrones se utilizan para probar la estructura. La dificultad es obtener e interpretar los datos de difracción del electrón, esta es raramente usada para determinar la estructura de una proteína.

Predicción de estructuras

Las grandes macromoléculas complejas y máquinas moleculares presentan un desafío particular en la determinación de estructuras. Por lo general son demasiado largas para ser cristalizadas, y demasiado complejas para ser resueltas por NMR, la determinación de la estructura requiere combinar los microscopios de gran resolución, con el refinamiento computacional y el análisis. Las principales técnicas utilizadas son microscopia crioelectrónica (Cryo-EM) 11 y la microscopia de la luz estándar.

1.1.5. Repositorio de estructuras de proteínas

La mayoría de estructuras de proteínas descubiertas a la fecha pueden ser encontradas en un gran repositorio de proteínas llamado RCSB Protein DataBank (PDB) [6]. El Protein Data Bank (PDB) es un repositorio de dominio público que contiene estructuras tridimensionales de proteínas por métodos experimentales. La mayoría de las proteínas en PDB han sido determinadas por cristalografía de rayos X, pero el número de proteínas usando NMR ha ido en incremento gracias a las técnicas computacionales para manejar los datos desarrollados en este método. En la Figura 1.10 se muestra la página actual de PDB donde se observa en la parte superior derecha el número actual de proteínas.

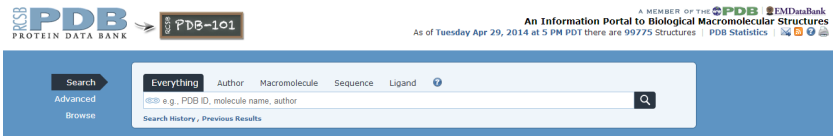


Figura 1.10: Página principal de Protein Data Bank.

1.2. Representación computacional

Para construir un programa eficiente, que sea fácil de mantener, y que otorgue la capacidad de manipular las estructuras de una proteína, se tiene que contar con las estructuras de datos correctas. Dependiendo de la aplicación del programa, las diferentes presentaciones pueden representar una ventaja o una desventaja para el usuario final. Por ejemplo, cuando se diseña un programa de visualización simple, las coordenadas (x, y, z) de cada átomo se utilizan y se muestran en la pantalla. Sin embargo, si el programa es para manipular ángulos de enlace y largo de enlaces por ejemplo, una representación basada en un grado de libertad interno puede ser más apropiado. Algunas aplicaciones pueden entonces necesitar guardar más que una representación al mismo tiempo; por ejemplo un programa que simule la representación de la energía potencial de una proteína.

La estructura de una proteína es el conjunto de átomos contenidos, y enlaces que los conecta, es decir su conectividad inherente. Una forma geométrica especifica de una proteína (es la disposición espacial de los átomos en la molécula) se llama conformación. Esta, da una estructura a la proteína que puede tener muchas conformaciones diferentes. A continuación se explicaran las dos principales maneras para modelar la estructura de una proteína y su conformación para utilizar en un programa de computo: Cartesiana y Diedral.

1.2.1. Conformación estructural cartesiana

La información esencial para modelar la estructura de una proteína es mediante la posición relativa de cada átomo, dando una coordenada Cartesiana (x, y, z). Los métodos populares basados en imágenes como Cristalografía de Rayos X, Resonancia Magnética Nuclear (NMR) y Criogénico Microscópico Electrónico (Cryo-EM) se usan para obtener posiciones relativas de proteínas cristalizadas o en solución. Esta información es provista por Protein Data Bank (PDB) en un archivo con coordenadas. A continuación se muestran las primeras 19 coordenadas de la proteína glucagon de PDB en la Figura 1.11.

Cada línea que inicie con el tipo ATOM indica que es un átomo a ser utilizado para construir la molécula. El número de serie del átomo es el siguiente elemento a ser considerado.

El nombre del átomo es el tercer elemento. Es importante notar que el primero o los dos primeros caracteres del nombre del átomo consisten del

```
ATOM      1  N   HIS A   1      49.668  24.248  10.436  1.00 25.00           N
ATOM      2  CA  HIS A   1      50.197  25.578  10.784  1.00 16.00           C
ATOM      3  C   HIS A   1      49.169  26.701  10.917  1.00 16.00           C
ATOM      4  O   HIS A   1      48.241  26.524  11.749  1.00 16.00           O
ATOM      5  CB  HIS A   1      51.312  26.048   9.843  1.00 16.00           C
ATOM      6  CG  HIS A   1      50.958  26.068   8.340  1.00 16.00           C
ATOM      7  ND1 HIS A   1      49.636  26.144   7.860  1.00 16.00           N
ATOM      8  CD2 HIS A   1      51.797  26.043   7.286  1.00 16.00           C
ATOM      9  CE1 HIS A   1      49.691  26.152   6.454  1.00 17.00           C
ATOM     10  NE2 HIS A   1      51.046  26.090   6.098  1.00 17.00           N
ATOM     11  N   SER A   2      49.788  27.850  10.784  1.00 16.00           N
ATOM     12  CA  SER A   2      49.138  29.147  10.620  1.00 15.00           C
ATOM     13  C   SER A   2      47.713  29.006  10.110  1.00 15.00           C
ATOM     14  O   SER A   2      46.740  29.251  10.864  1.00 15.00           O
ATOM     15  CB  SER A   2      49.875  29.930   9.569  1.00 16.00           C
ATOM     16  OG  SER A   2      49.145  31.057   9.176  1.00 19.00           O
ATOM     17  N   GLN A   3      47.620  28.367   8.973  1.00 15.00           N
ATOM     18  CA  GLN A   3      46.287  28.193   8.308  1.00 14.00           C
ATOM     19  C   GLN A   3      45.406  27.172   8.963  1.00 14.00           C
```

Figura 1.11: Primeros 19 elementos de la proteína glucagon.

elemento químico para el tipo de átomo. El nombre del átomo empieza con C si es un átomo de carbono; N indica un átomo de nitrógeno y O indica un átomo de oxígeno. En un residuo de aminoácido, el siguiente carácter es el indicador de lejanía, el cual es traducido de acuerdo a:

- $\alpha - A$
- $\beta - B$
- $\gamma - G$
- $\delta - D$
- $\varepsilon - E$
- $\zeta - Z$
- $\hbar - H$

El siguiente carácter del nombre del átomo es el indicador de rama, si es requerido.

El siguiente campo es el tipo de residuo. Es importante notar que cada elemento contiene su tipo de residuo. En el ejemplo el primer residuo en la cadena es HIS (histidina) y el segundo residuo es SER(serina).

El siguiente campo contiene el identificador de la cadena, en este caso es A.

El siguiente campo contiene el número de secuencia del residuo. Es importante tomar en cuenta que a medida que el tipo de residuo cambia de histidina a serina, el número de residuo cambia de 1 a 2. Dos residuos pueden ser adyacentes entre sí, por lo que el número de residuo es importante para distinguir entre ellos.

Los tres siguientes campos contienen los valores de las coordenadas X, Y y Z respectivamente. Los tres últimos campos muestran la ocupación, el factor de temperatura (Factor-B), y el elemento químico.

Los espacios entre los campos son importantes. Si los campos no aplican estos deben ser dejados en blanco.

1.2.2. Grados de libertad en una proteína

Los **grados de libertad** de un sistema son un conjunto de parámetros que pueden ser variados, independientemente del estado del sistema. Por ejemplo, la localización de un punto en el espacio cartesiano 2D puede ser definido como un desplazamiento a lo largo del eje X y un desplazamiento del eje Y, dado como un par (X, Y). Puede ser también dado como una rotación a partir del origen por un grado θ y una distancia r del origen, dándonos un par (r, θ). En otro caso, un punto se mueve libremente en un plano teniendo exactamente dos grados de libertad.

Como se menciono anteriormente, la disposición espacial de los átomos en una proteína constituye una estructura. En los archivos de coordenadas de PDB, se puede observar que una manera obvia de definir la conformación de una proteína es dando las coordenadas x, y, z para cada átomo, estos datos son dados a un origen específico. Estos sin embargo, no son grados de libertad independientes, no obstante, los átomos dentro de una molécula no pueden salir de los alrededores de sus átomos vecinos (si no existe una reacción química). Los átomos conectados entre si, por ejemplo, se ven obligados a permanecer cerca, por lo que mover un átomo hace que los que se encuentran conectados a él, se muevan de una manera dependiente. En terminología cinemática, esto quiere decir que, el número de efectos o de grados de libertad independientes es mucho menor que el espacio de entradas de parámetros (x, y, z) para cada átomo.

Enlaces y largo de enlace

Un átomo en una proteína es conectado a otra a través de un enlace covalente. Cada par de átomos unidos tienen una distancia de separación preferente llamada **largo de enlace**. El largo de enlace puede variar ligeramente con las vibraciones como un resorte, y es un grado de libertad, pero las variaciones reales en longitud de enlace son tan pequeñas que la mayoría de las simulaciones suponen que son fijos para cada par de átomos. Esta es una suposición muy común en la literatura y reduce el efecto en los grados de libertad de una proteína. Para el resto de este proyecto y su aplicación en el software se realiza esta suposición.

Aunque el largo de la articulación no variara para realizar este trabajo, la presencia de enlaces es importante, ya que nos permite representar la conectividad de la proteína como un grafo no dirigido, donde los átomos son los nodos y los enlaces entre estos son aristas unidireccionales. En algunos casos, es de ayuda para romper de manera artificial cualquier ciclo en el grafo, elegir un átomo del interior como un átomo de anclaje. El grafo puede entonces ser tratado como una estructura de árbol, con un átomo de anclaje como raíz. A continuación se muestra en la Figura 1.12 una proteína como una estructura de datos de árbol.

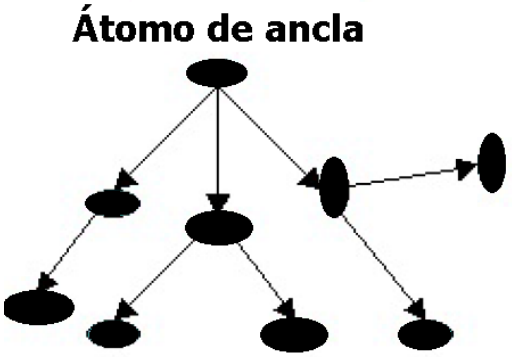


Figura 1.12: Una representación como árbol de la conectividad de una proteína, para una molécula muy pequeña . Los ciclos son rotos para ignorar un enlace en cada uno.

Ángulos de enlace

El largo de un enlace es un grado de libertad independiente a dos átomos conectados. Un conjunto de tres átomos enlazados en secuencia definen otro grado de libertad: el ángulo entre los dos enlaces adyacentes. Esto es, apropiadamente, definido como ángulo de enlace. El ángulo de enlace puede ser calculado como el ángulo entre los dos vectores correspondiente a los enlaces del átomo central a cada uno de sus vecinos. Como el largo de los enlaces, los ángulos de enlace tienden a ser característicos de los átomos que intervienen, con algunas excepciones, varían muy poco.

1.2.3. Ángulos diedros

En la mayoría de las moléculas orgánicas, incluyendo las proteínas, el grado de libertad más importante es la rotación alrededor de un **ángulo diedro (torsional)**. Un ángulo diedro es definido por cuatro consecutivos enlaces de átomos. Teniendo cuatro átomos consecutivos $A_{i-2}, A_{i-1}, A_i, A_{i+1}$, el ángulo diedro es definido como el mas pequeño ángulo entre el plano Π_1 y Π_2, como se muestra en la Figura 1.13. La variación de los ángulos es consecuencia de la rotación de los dos enlaces externos alrededor del enlace central.

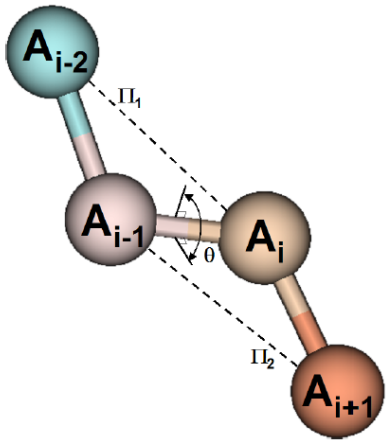


Figura 1.13: Π_1 es el ángulo definido de forma única por el plano de los primeros tres átomos A_{i-2}, A_{i-1}, A_i. Similarmente, Π_2 es el plano definido por los tres últimos átomos A_{i-1}, A_i, A_{i+q}, El ángulo diedro, θ es definido como el ángulo más pequeño entre estos dos planos.

En este modelo, debido a que el largo del enlace y el ángulo de enlace no se toman en cuenta como grados de libertad de una proteína, los grados de libertad utilizados son los ángulos diedros. La conformación de la representación de la proteína con ángulos diedros como único grado de libertad es conocida como modelos ideales o modelos de geometría rígida. Ignorando el largo del enlace y ángulo de enlace se reduce en gran medida el número de grados de libertad y por tanto, se reduce la complejidad de representación y manipulación de la estructura de una proteína. Aún existen más representaciones eficientes las cuales reducen el número de grados de libertad [58].

Representación diedra de la proteína

Todos los aminoácidos comparten el mismo núcleo de un nitrógeno, dos carbonos, y un átomo de oxígeno. Este núcleo en común constituye el esqueleto de la proteína. Hay dos enlaces libres de rotar por residuo de aminoácido en una cadena de proteína: el primero, designado Φ, es consecuencia de la rotación a través de los enlaces N y C_a y el otro, Ψ el cual es consecuencia de la rotación de los enlaces C_α y C. El enlace péptido entre C de un residuo y N del residuo adyacente no se puede girar.

El número de enlaces diedros por aminoácido es 2, pero el número de cadenas laterales diedras varía con el largo de las cadenas laterales. Su valor se encuentra entre 0, en el caso de la glicina, que no tiene cadenas laterales, a 5 en el caso de la arginina que se observa en la Figura 1.14.

Se pueden generar diferentes estructuras tridimensionales de la misma proteína variando los ángulos diedros. Hay 2N grados de libertad de esqueletos diedros para una proteína con N aminoácidos, y llega a 4N cadenas laterales diedras que pueden variar para generar nuevas conformaciones de la proteína.

1.3. Enfoques para el plegado de proteínas

El proceso de plegado es importante por varias razones, ya que puede proporcionar la idea de cómo es que se pliegan las proteínas y puede ayudar a entender los factores que controlan este proceso en algunas proteínas (mecanismos). Esta área de investigación es de gran importancia práctica ya que existen algunas enfermedades devastadoras provocadas por plegamientos

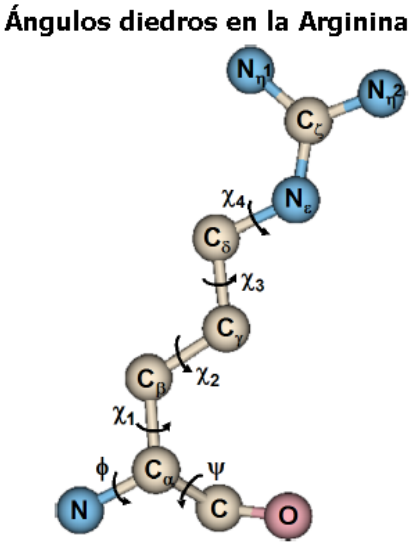


Figura 1.14: Los átomos del esqueleto aparecen en la parte inferior de la ilustración (el enlace péptido no es giratorio). Las cadenas laterales diedras se designan por χ y un subíndice.

no nativos de la proteína, como la *encefalopatía espongiforme bovina*[1] , en estos casos es importante entender por qué ocurren estos plegados y cómo se podrían prevenir.

En este trabajo se realizó una herramienta para el estudio del plegado de proteínas. Esta herramienta es capaz de aproximar el mapa potencial de la proteína y sus energías libres para encontrar caminos de plegado con potencial energético. En particular con las técnicas probabilistas se podrán encontrar múltiples caminos en un simple roadmap.

Existen diferentes trabajos relacionados al plegado de proteínas que han logrado simular el campo potencial pero que cuentan con diferentes ventajas. A continuación se describen estas técnicas en comparación a la técnica utilizada en este trabajo.

Las simulaciones de Monte Carlo [11, 28] y la simulación de dinámica molecular [35, 13, 15] proveen solo una trayectoria de plegado, y cada ejecución es computacionalmente intensiva porque estas intentan simular una cinemática compleja con termodinámica en cada punto visitado en el espacio de conformaciones. Los modelos mecánicos estadísticos [41, 1], por otra

[1] La enfermedad de las vacas locas.

parte, es extremadamente simplificado para interacciones moleculares y se limita al estudio de los promedios globales del plegado de la cinemática. Los modelos de Lattice [8] han sido bien estudiados y poseen fuerte valor teórico pero no pueden ser aplicados a proteínas reales. Por lo tanto, las técnicas que se utilizan en nuestro trabajo que son basadas en PRM [50, 3, 49] y RRT[42], para la construcción de un roadmap para la aproximación de plegado, son las única técnicas capaces de calcular de manera eficiente múltiples caminos plegables en una única prueba.

En el Cuadro 1.2 y el Cuadro 1.3 se proporciona un resumen comparativo de los diversos modelos de plegamiento de proteínas.

Técnica	Pasajes	# Caminos	Calidad camino	Dependencia de tiempo
Dinámica molecular[35]	No	1	Bueno	Si
Monte Carlo [11, 28]	No	1	Bueno	Si
Modelo estadístico [41, 1]	Si	0	N/A	No
PRM [50, 3, 49]	Si	Muchos	Aproximado	No
RRT[42]	Si	1	Aproximado	No
Modelos de Lattice	No usado	en	proteínas	reales.

Cuadro 1.2: Comparación de modelos de plegado de proteínas I.

1.4. Conclusión

El proceso de plegado de proteínas permanece en constante investigación. En este trabajo desarrollamos una herramienta basada en técnicas probabilísticas que permite simular el proceso de plegado, utilizando interfaces gráficas con OpenGL para proveer al usuario final la capacidad de modificar la estructura de la proteína y utilizar distintas técnicas de visualización de moléculas para facilitar el estudio e interpretación de los procesos de plegado.

Este trabajo esta limitado a proteínas monoméricas [2], estas proteínas pue-

[2]Proteínas que cuentan con una única cadena polipeptídica.

Técnica	Tiempo de compensación	Cinemática de plegado	Nativo necesario
Dinámica molecular[35]	Largo	No	No
Monte Carlo [11, 28]	Largo	No	No
Modelo estadístico [41, 1]	Corto	Promedio	Si
PRM [50, 3, 49]	Corto	Múltiple	Si
RRT[42]	Corto	Múltiple	Si
Modelos de Lattice	No usado	en proteínas	reales.

Cuadro 1.3: Comparación de modelos de plegado de proteínas II.

den ser encontradas en la base de datos de proteínas PDB para ser utilizadas en nuestra herramienta. Esta herramienta de plegado una vez generados los diferentes caminos, permite al usuario observar el proceso de plegado en una animación generada con OpenGL, con lo cual cuentan pocas herramientas en la actualidad. A continuación se hace una breve descripción del contenido de cada capítulo en este trabajo de tesis.

1. **Capítulo 2 Desarrollo de una herramienta de simulación para el plegado de proteínas**: En este capítulo se explicara el contexto de la cinemática aplicada a las proteínas (directa e inversa). Se describirá que es un detector de colisiones, y la aplicación de BioCD el detector de colisiones biológicas.

2. **Capítulo 3 Plegado de proteínas con algoritmos probabilistas**: Se explicara los algoritmos de planificación de movimientos basados en probabilidad (PRM y RRT) y como estos se aplican en el problema de plegado de proteínas. Así como los resultados comparativos entre las dos diferentes propuestas.

3. **Capítulo 4 Conclusiones y trabajo futuro**: hablaremos de los resultados obtenidos y lo que podemos concluir sobre el problema de plegado de proteínas y los posibles trabajos que podrían aplicarse para una futura versión.

Capítulo 2

Herramienta de simulación propuesta

En este capitulo se describen las técnicas necesarias para modelar el comportamiento de las proteína, modificar su conformación y verificar sus restricciones biológicas, para ello se utilizaron los algoritmos de cinemática aplicados en la robótica para modelar las conformaciones y los algoritmos de detección de colisiones para validarlas.

2.1. Cinemática para las proteínas

La **cinemática** es una rama de la mecánica que se encarga de estudiar el movimiento de los objetos en ausencia de una masa (inercia) y de la fuerza. Al realizar variaciones de los ángulos diedros se moverán átomos de una proteína relacionados con otros átomos en el mismo espacio. El problema de calcular las nuevas ubicaciones espaciales de los átomos, dado un conjunto de rotaciones diedras se conoce como el problema de **la cinemática directa**.

La importancia de este problema para el modelado y simulación de proteínas es claro: como se mencionó, los grados de libertad internos generalmente considerados para una proteína son sus ángulos diedros. Así, los movimientos de una proteína pueden ser logrados mediante el establecimiento de nuevos valores para sus ángulos diedros. Algunas aplicaciones para esto son, la representación de una imagen de la proteína y el cálculo de su energía, sin embargo, las coordenadas cartesianas (x, y, z) para cada átomo siguen siendo necesarias. Estas se obtienen mediante cinemática directa.

2.1.1. Cinemática directa

Como se dijo anteriormente, una operación común en la manipulación de proteínas en silicio, es para recuperar las coordenadas cartesianas de cada átomo en la proteína a partir de nuestro conocimiento de sus ángulos diedros y las rotaciones aplicadas a ellos. Por simplicidad, se supone que se tiene un átomo de ancla y se está modelando el esqueleto de la proteína únicamente, es decir, la proteína consiste en un enlace en serie compuesto de átomos de cadena principal consecutivos, como se muestra en la Figura 2.1.

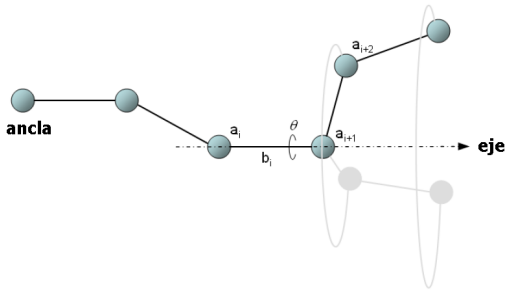


Figura 2.1: Un esqueleto de la proteína como un enlace encadenado.

Enfoque simple

La manera más simple para representar una cadena de proteína es almacenar las coordenadas cartesianas (x, y, z) de cada átomo en todo momento. Estas coordenadas son relativas a algún sistema de coordenadas global, de hecho no se considera importante, por ejemplo el hecho de que las posiciones atómicas fueron obtenidas por cristalografía de rayos X y que por lo general se leen desde el archivo PDB. Estas coordenadas se pueden cambiar si se desea. Los cambios comunes son para eliminar el centro de masa.

Pero se dijo anteriormente que los grados *naturales* de libertad de manipulaciones cinemática son generalmente los ángulos diedros únicamente. Esto significa que los algoritmos que operan en ángulos diedros para lograr sus objetivos normalmente requieren una forma de modificar las coordenadas cuando se realizan rotaciones diedras, para reflejar las nuevas posiciones atómicas. Esto se puede hacer fácilmente con las matrices de rotación de la siguiente manera.

Cuando se efectúa una rotación de θ grados alrededor de un enlace i, se puede pensar en todas las posiciones de los átomos a partir de $i+2$ rotaciones

alrededor del eje definido como enlace i, y todos los otros átomos (de anclaje al átomo $i+1$ inclusive) permanecen fijos. Por lo tanto, en una rotación tal, las coordenadas cartesianas de los átomos después de la unión necesitan ser actualizados, y sus nuevos valores se dan por:

$$[x', y', z', 1]^T = R(i, \theta) * [x, y, z, 1]^T \tag{2.1}$$

Donde $[x, y, z, 1]$ es la posición del átomo genérico n en forma homogénea, $[x', y', z', 1]$ es esta posición después de la rotación (T es la operación translación), y $R(i, \theta)$ es una matriz 4×4 que codifica una rotación de θ grados alrededor del eje que coincide con el vínculo i que pasa a través de un átomo a_i. Y es dado en una forma homogénea como:

$$R(i, \theta) = T(a_i) * R_0(eje, \theta) * T(-a_i) \tag{2.2}$$

En la formula anterior, $T(x)$ es una translación por el vector x, $R_0(eje, \theta)$ es una rotación alrededor de un eje que pasa por el origen del sistema de coordenadas especificado. Como puede verse, esta rotación alrededor de un punto arbitrario se realiza mediante la traducción del punto al origen. Al girar el átomo de objetivo alrededor del eje que pasa por el origen, y luego trasladar lo de nuevo (la composición de estas 3 transformaciones produce una única matriz homogénea 4x4 que realice el mismo efecto).

El eje de rotación puede ser fácilmente calculada a partir de las posiciones de los átomos i e $i+1$ debe tener una unidad de norma. Para realizar rotaciones sucesivas sobre los enlaces, este procedimiento puede ser repetitivo para actualizar las coordenadas para cada rotación. Teniendo en cuenta que la convención usada para la multiplicación de matriz-vector es para multiplicar vectores columna por matrices de la izquierda, por lo que la transformación de más a la derecha queda aplicada en primer lugar, y así sucesivamente. Esta es la convención utilizada en la mayor parte de la literatura, pero una convención alternativa es posible (multiplicar vectores fila con las matrices de la derecha, estas matrices son la transposición de la convención vector columna) por cuestiones de facilidad.

Alternativamente, si muchas rotaciones necesitan ser realizadas al mismo tiempo (y no son necesarias las coordenadas cartesianas intermedias), estas rotaciones pueden ser ordenadas por número de enlaces y se aplican simultáneamente, señalando que las rotaciones se pueden realizar de una manera acumulativa al átomo final. La capacidad de rotación de la cadena

alrededor de vectores arbitrarios en el espacio (es decir, no a través del origen) es uno de los principales beneficios de las transformaciones homogéneas. Por ejemplo, si se deben aplicar al mismo tiempo, uno alrededor del enlace 3 de 30 grados y otro alrededor del enlace 7 de 15 grados dos rotaciones, los átomos entre los enlaces 3 y 7 se actualizan a través de :

$$[x', y', z', 1]^T = R(enlace_3, 30) * [x, y, z, 1]^T \tag{2.3}$$

Para los átomos después del enlace 7 son actualizados como:

$$[x', y', z', 1]^T = R(enlace_7, 15) * R(enlace_3, 30) * [x, y, z, 1]^T \tag{2.4}$$

En lo anterior, el enlace n es el vector unitario definido a lo largo del enlace n, fácilmente calculado restando las coordenadas de los átomos de $n+1$ y n, y luego dividiendo por su norma. El encadenamiento de transformaciones como se explicó anteriormente es muy útil para lograr rotaciones arbitrarias de enlaces dentro de una proteína. Las secciones de la proteína (es decir, átomos que pertenecen a determinados residuos) pueden ser actualizadas cuando una rotación diedra se lleva a cabo simplemente mediante la construcción de la matriz general que debe afectarlos.

Convención Denavit-Hartenberg

La aproximación anterior, es simple e intuitiva, pero tiene algunas deficiencias:

- La acumulación de las operaciones matemáticas en las matrices de rotación es propensa a la inestabilidad numérica. Después de solo un par de cientos de rotaciones de un punto, la acumulación en otro, la posición final del punto puede comenzar difiriendo significativamente de su posición actual, prevista. Como consecuencia de ello, la posición relativa y la orientación de los átomos en la cadena de la proteína ya no están de acuerdo con la estructura de la proteína. En particular, las longitudes y ángulos de enlace comenzara a estriarse y desviarse de sus valores físicamente aceptables.

- Los valores reales de las coordenadas cartesianas se almacenan siempre en una referencia particular arbitrada. Por ejemplo, si se quiere trasladar la proteína, se tendría que modificar las coordenadas cartesianas almacenadas.

- Una vez que se aplica una rotación, el método *olvida* los valores actuales de los ángulos diedros, que tendrían que ser recalculados si es necesario. Lo que se almacena es una instantánea de las coordenadas cartesianas actuales de cada átomo.

La definición original de la cinemática directa, sin embargo, es un método para obtener las coordenadas cartesianas de cada átomo de los valores actuales de los grados de libertad internos (ángulos diedros en nuestro caso) en cualquier momento. En este enfoque, las coordenadas cartesianas no necesitan ser calculadas de nuevo después de cada cambio en los ángulos diedros, sino más bien, la idea es almacenar los valores actuales de los ángulos diedros, y tener un procedimiento para reconstruir las posiciones atómicas cuando sea necesario. Las ventajas de este enfoque son:

- Una representación más compacta de las variables del problema, dado que los ángulos diedros requieren menos espacio que el (x, y, z) de cada átomo (la topología de la proteína requiere los valores de las longitudes de enlace y ángulos de todos modos, por lo que la cantidad total de números para almacenar es importante).

- No es propenso a la inestabilidad numérica considerando que el número de rotaciones realizadas para posicionar un átomo es siempre un número de secuencias en la cadena. (En realidad, si la cadena es de miles de residuos largos, podría surgir cierta incertidumbre en la posición de los átomos de lejos a lo largo de la cadena, pero la posición relativa de átomos consecutivos, se pueden mantener bajo control evitando la unión de estiramiento).

- La realización de una rotación diedra consiste simplemente en sumar o restar el ángulo de giro en el valor almacenado para cada ángulo. En particular, las rotaciones simultaneas (es decir, girar más de un ángulo diedro a la vez), que consiste en multiplicar muchas matrices 4×4 en el método global, reduce la modificación de los valores de ángulos.

- No hay un marco global de coordenadas explícito para la proteína. Se puede colocar arbitrariamente anteponiendo una matriz de posición/orientación para el cálculo de la cinemática directa.

El único paso de procesamiento previo que es necesario para empezar a trabajar con este método, es para a extraer los valores iniciales de los ángulos diedros. En este modelo, la longitud y ángulo de enlace son considerados

constantes. A partir de las coordenadas de cada átomo disponibles en el archivo PDB, se puede calcular la conformación nativa de la proteína. La longitud de enlace es calculada como la distancia existente entre dos átomos conectados, el ángulo de enlace es el ángulo formado entre dos vectores consecutivos (el producto escalar de dos vectores se obtiene del producto de sus longitudes por dos veces el coseno del ángulo entre ellos). A continuación se presentan las transformaciones necesarias para poder aplicar el marco de referencia Denavit-Hartenberg.

Consideremos tres enlaces consecutivos, como en la siguiente figura. Se considera que un marco de coordenadas local se adjunta al comienzo de cada enlace. Por ejemplo, el sistema de coordenadas local $x_{i-1}, y_{i-1}, z_{i-1}$ se centra en el átomo de A_{i-1}. Por lo tanto, imaginemos que la posición de cada átomo en un espacio tridimensional se especifica en términos de un marco que está anclado en el átomo anterior.

Teniendo en cuenta los marcos en el átomo de A_{i-2}, y un átomo de A_{i-1}, se puede determinar cómo los marcos en los átomos de A_i y el átomo de A_{i-1} cambiarán en el espacio como consecuencia de una rotación alrededor del enlace que conecta los átomos A_{i-1} y A_i con el ángulo diedro [58].

La transformación correcta puede calcularse en términos de tres operaciones primitivas: dos rotaciones y una translación. Las dos rotaciones son una rotación alrededor del enlace diedro por el ángulo diedro y una rotación alrededor de un eje perpendicular al ángulo de enlace, por el ángulo de enlace. La translación se refiere al hecho de que los orígenes de los marcos están en los respectivos centros de los átomos conectados por el enlace, por lo tanto separadas por longitudes de enlace.

El orden en que se realizan las 3 operaciones para obtener la transformación total que da como resultado la posición del átomo i en términos del marco de referencia $i-1$, es la siguiente:

$$R(x, \alpha_{i-1}) * R(z, \theta) * T(0, 0, d_i) \tag{2.5}$$

Donde las rotaciones en los ejes son los vectores $x(1, 0, 0)$ y $z(0, 0, 1)$, no deben ser confundido con el formalismo DH. El resultado es la transformación homogénea siguiente:

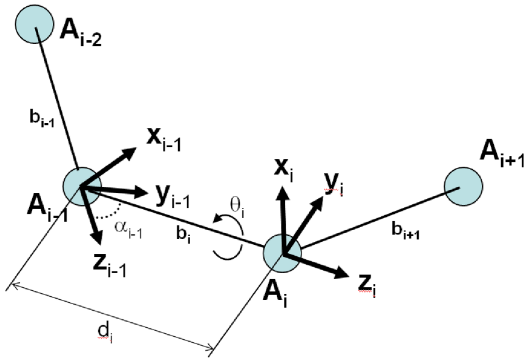


Figura 2.2: Para describir el átomo i en términos del marco de coordenadas centrados en el átomo $i-1$, son necesarias dos rotaciones y una translación.

$$E = \begin{pmatrix} cos(\theta_i) & -sin(\theta_i) & 0 & 0 \\ sin(\theta_i)cos(\alpha_{i-1}) & cos(\theta_i)cos(\alpha_{i-1}) & -sin(\alpha_{i-1}) & -sin(\alpha_{i-1}) * d_i \\ sin(\theta_i)sin(\alpha_{i-1}) & cos(\theta_i)sin(\alpha_{i-1}) & cos(\alpha_{i-1}) & cos(\alpha_{i-1}) * d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.6}$$

La ecuación 2.6 muestra la transformación homogénea para expresar las coordenadas del $i - esimo$ átomo en términos del eje centrado en $i-1$.

Podemos notar que θ_i es el ángulo diedro en el enlace b_i y α_{i-1} es el ángulo de enlace entre en enlace b_{i-1} y $b_i d_i$ es el largo del enlace b_i. En [12] se propone una descripción más detallada de este proceso. La posición de algún átomo en la molécula puede ser determinada por un conjunto de matrices encadenadas de la siguiente manera. Por ejemplo, decimos que $b_i, b_{i-1}, ..., b_1$, representa la secuencia de los enlaces en el camino de un átomo en particular al átomo central a_{cen}. Entonces, para un átomo a, su coordenada cartesiana con respecto a la referencia el átomo central está dado como sigue:

$$[x_i, y_i, z_i, 1]^t = T_1 T_2 ... T_i [0, 0, 0, 1]^t \tag{2.7}$$

La coordenada del átomo a con respecto a la referencia que es $(0, 0, 0)$.

Para completar la descripción, se pueden permitir rotaciones o translaciones a la referencia local, unido al átomo de ancla con respecto a algún marco global. Las rotaciones del átomo de ancla con respecto a un marco global que causa rotaciones rígidas de la cadena polipeptídica entera. Para ello, uno puede definir el marco de la rotación como la matriz de Euler definida por

los ángulos de Euler de la estructura local del átomo de ancla a la estructura global.

Como se mencionó, hay muchas convenciones para definir la matriz de Euler. Una de estas es la convención XYZ, que define la matriz de Euler como el producto de tres matrices de rotación: rotación alrededor del eje z por el ángulo α; rotación alrededor del eje y por el ángulo β; rotación alrededor del eje x por el ángulo γ, El orden de realización de estas tres rotaciones en las convenciones de X, Y, Z es:

Primero la rotación en el eje x, luego alrededor del eje y, y en torno al eje z al final. La matriz de Euler resultante de acuerdo con esta convención se define a continuación.

$$E = \begin{pmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha c\gamma & 0 \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma & 0 \\ s\beta & c\beta s\gamma & c\beta c\gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

α, β, γ son los llamados ángulos de Euler. Los ángulos con respecto a cada uno de los ejes cartesianos. La convención usada aquí es XYZ, $c\alpha$ y $s\alpha$ son las abreviaciones de $cos(\alpha)$ y $sin(\alpha)$ respectivamente.

La matriz de Euler se puede aplicar a las rotaciones diedras de acumulación con el fin de permitir que el átomo de ancla se mueva con respecto a un punto de origen global.

La cinemática directa cumple con resolver el problema de poder manipular la configuración de la proteína modificando sus ángulos diedros para obtener una nueva configuración correcta. La cinemática directa ayuda a manipular la configuración de la proteína pero en el caso de que se quiera obtener una configuración a partir de un punto o un dato de referencia se necesita resolver un nuevo problema llamado cinemática inversa.

2.1.2. Cinemática inversa

La cinemática inversa es el problema de encontrar valores para los grados de libertad de una cadena cinemática, de los ángulos diedros, que satisfacen cierta configuración espacial. Por ejemplo, en algunas aplicaciones, es necesario para encontrar las rotaciones que pueden dirigir ciertos átomos a una localización espacial determinada. El movimiento de los átomos es especialmente importante porque estos han asumido una posición en la localización

espacial. Sin embargo, los átomos deben moverse juntos con el fin de no romper los enlaces durante su movimiento. Es tan fácil dado que los modelos de su movimiento están dados por sus ángulos diedros en el espacio, donde el largo del enlace y el ángulo del enlace son fijos. Estos parámetros del movimiento de la proteína, son llamados **los modelos de geometría rígida**.

Resolver el problema de la cinemática inversa en el contexto de las proteínas, por ejemplo, sirve para encontrar que valores de los ángulos diedros de la proteína en sus cadenas polipéptidas permiten una configuración donde el punto final satisface un cierto criterio, esto es un problema muy importante en la biología estructural. La relevancia de la cinemática inversa para proteínas puede verse en tres principales aplicaciones:

- Encontrar un ciclo perdido (Problema de ciclo cerrado).
- Estudiar las características de un fragmento de la proteína en cierta configuración.
- Generar ensambles de la estructura de una proteína.

En la aplicación de algoritmos de cinemática inversa en proteínas, es posible aprovechar la sorprendente similitud entre las moléculas orgánicas y los manipuladores robóticos (brazos robóticos), en términos de cómo se mueven. Como los robots manipuladores tienen articulaciones, las proteínas tienen átomos. Como los robots manipuladores tienen vínculos que conectan las articulaciones, las proteínas tienen lazos que unen sus átomos. La similitud entre las proteínas y los robots hace posible que se pueda aplicar a las proteínas una gran cantidad de soluciones al problema de la cinemática inversa existentes en la literatura de la robótica

Antes de dar algún ejemplo de cinemática inversa, es importante decir que la cinemática inversa es el problema inverso de la cinemática directa. Una manera inmediata de resolver la cinemática inversa es invirtiendo las ecuaciones de la cinemática directa. Como se puede observar en la Figura 2.3, la solución al problema de la cinemática inversa no es necesariamente único. De hecho, como el número de grados de libertad aumenta, el número máximo de soluciones también aumenta. Un hecho importante es decir que un sistema no tiene solución para un punto dado si este se encuentra fuera del alcance del robot.

La cinemática inversa tienen dos categorías [59]:

- Exacta, clásica o de métodos algebraicos:

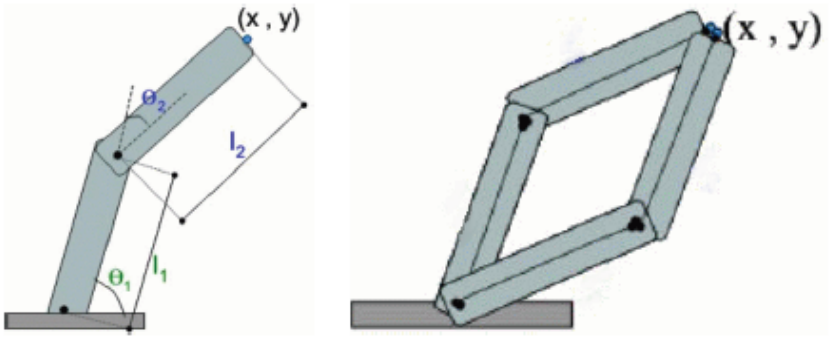


Figura 2.3: Caso con doble solución de la cinemática inversa en un manipulador de dos grados de libertad.

- Son completos.
- Reportan todas las posibles soluciones para una configuración.
- Trabajan bien para configuraciones con menos de 9 grados de libertad.
- Enfoque jerárquico.

- Heurísticos o métodos de optimización:
  - No son completos.
  - No tiene restricciones con los grados de libertad que se pueden trabajar.

En este trabajo de tesis se trabajo con el algoritmo heurístico de Coordenadas Descendientes (CCD) [7] debido a que permite utilizar un amplio número de grados de libertad. Sin embargo, a continuación se mencionara más acerca de las estrategias exactas y heurísticas.

Cinemática inversa clásica

Se sabe que los manipuladores con un máximo de seis grados de libertad, tienen un número finito de soluciones para la cinemática inversa de un problema [12]. Esto quiere decir, sin embargo, que los métodos analíticos no son capaces de encontrar solución para todos los tipos de manipuladores. Para manipuladores con solo articulaciones de giro, como es el caso para las moléculas, el número de soluciones máximas para una configuración es de 16,

cuando el número de grados de libertad no excede de seis [44]. Los trabajos más recientes aumentaron el límite de 6 a 9 grados de libertad haciendo uso de subdivisiones eficientes del manipulador y el espacio de soluciones [61].

Nuestro trabajo de tesis no utilizara algoritmos clásicos de cinemática inversa porque limitaría en gran medida este desarrollo debido a que se podrían manejar un máximo de 6 a 9 grados de libertad y se sabe que las proteínas que cuentan con cientos de aminoácidos tienen 2N grados de libertad por lo cual no tendría ningún sentido utilizarlos.

Cinemática inversa con optimización

Las soluciones basadas en optimización son consideradas como una solución más apropiada para cadenas con un número arbitrario de grados de libertad. Existen dos principales algoritmos utilizados para la cinemática inversa en manipuladores Random Tweak(Ajustes Aleatorios) [16] y Cyclic Coordinate Descent (CCD Coordenadas Descendientes) [7]. Ambos métodos están basados en cambios iterativos de los grados de libertad de algunas articulaciones de la cadena cinemática hacia una posición objetivo.

Random Tweak es computacionalmente caro, numéricamente inestable, no es libre de singularidades matemáticas [16] . CCD es computacionalmente barato y libre de singularidades [54]. En la aplicación del estudio de estructuras biológicas como las proteínas.

En nuestro trabajo de tesis se utilizara CCD por ser el más recomendado en la literatura y por permitir trabajar con un gran número grados de libertad.

Coordenadas Descendientes (CCD)

CCD es un método iterativo que mueve las articulaciones en el orden opuesto a su importancia. Las articulaciones exteriores son rotadas primero y a continuación van iterando con las articulaciones internas de manera consecutiva. A diferencia de los algoritmos basados en el cálculo de la inversa Jacobiana, este algoritmo es más rápido y de convergencia rápida [7]. El algoritmo asigna valores de forma iterativa a todos los ángulos diedros ajustables desde el N-termino al extremo C-termino de la estructura cinemática. CCD es adecuado como un método de predicción de ciclos, ya que genera un gran número de estructuras de prueba.

Como se observa en la Figura 2.4, el algoritmo mide la diferencia entre

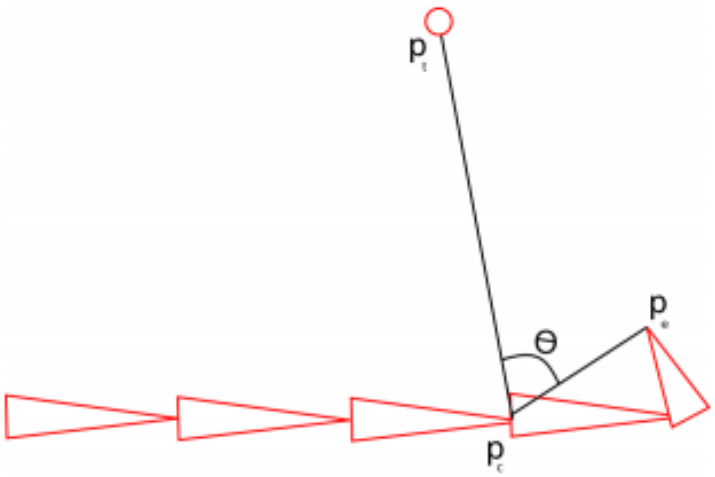


Figura 2.4: Uso de CCD: cada enlace p_c es rotado así que $\theta = 0$.

la posición p_c y la posición final p_e, y entre la posición p_c y la posición objetivo p_t. A continuación, calcula una rotación o un cuaternión para reducir esta diferencia a cero. Se hace esto para cada articulación, la iteración del vector final gira la articulación inmovilizando la raíz de la cadena cinemática. A medida que las articulaciones cercanas al vector final giran más que las articulaciones cercanas a la articulación final, aparecerá la cadena cinemática para rotar sobre si misma [40].

Se necesita resolver un conjunto de ecuaciones para cada articulación de los ángulos en el algoritmo de CCD, se necesitan resolver las siguientes ecuaciones:

$$cos(\theta) = \frac{p_e - p_c}{||p_e - p_c||} * \frac{p_t - p_c}{||p_t - p_c||} \tag{2.8}$$

$$\vec{r} = \frac{p_e - p_c}{||p_e - p_c||} X \frac{p_t - p_c}{||p_t - p_c||} \tag{2.9}$$

Donde p_t es el punto objetivo, p_e es el vector final, y p_c es la articulación actual que esta siendo rotada. El vector $\vec{r}$ es el eje de rotación. Así que cada articulación p_c es rotada por un ángulo θ alrededor de $\vec{r}$.

El trabajo realizado en este proyecto de investigación implementa CCD, basándose en [7], el cual modifica el algoritmo para adaptarlo al ámbito de las proteínas. La Figura 2.5 muestra cómo se realiza la implementación en el campo de las proteínas.

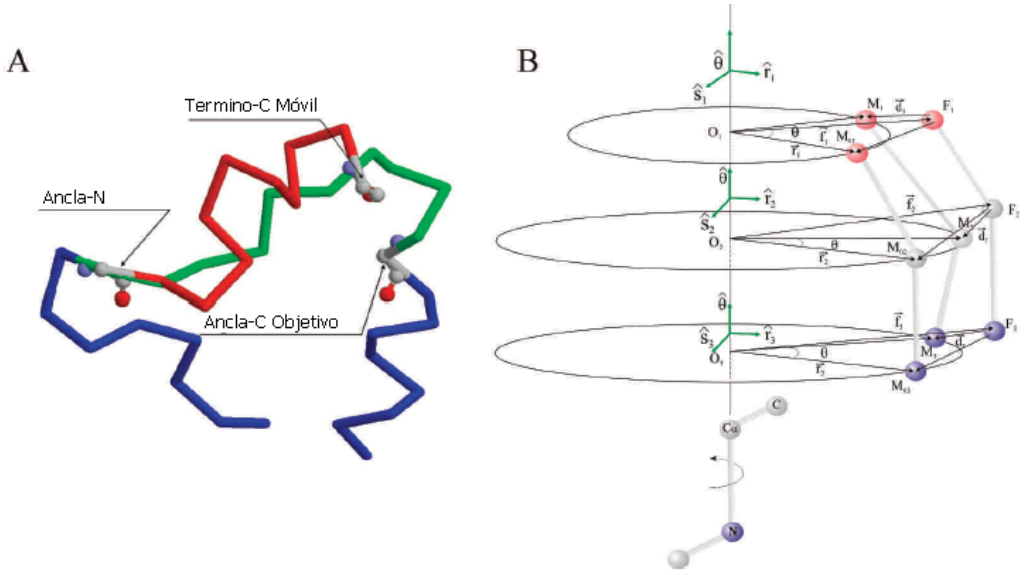


Figura 2.5: CCD aplicado al campo de las proteínas.

2.2. Detección de colisiones

2.2.1. Introducción

La detección de colisiones (CD) es un problema clásico en la robótica y en los gráficos por computadora. Ver [53, 21, 19] para recientes estudios. Esta ha sido ampliamente estudiada durante la década pasada y muchos paquetes de detección de colisiones eficientes están ahora disponibles. Una importante aplicación de los algoritmos CD es la planificación de movimientos robóticos. En particular, los planificadores basados en muestras (e.g. [26, 23]) extensamente usan técnica de CD para checar la validez de las muestras configuradas y del camino local computado entre las muestras. Es conocido que estos planificadores pasan la mayor parte del tiempo de cálculo haciendo estas pruebas de validez. Por lo tanto, su rendimiento global para la exploración es limitado y las configuraciones con altos niveles de dimensionalidad dependen en gran medida de las técnicas CD geométricas más eficientes.

Muchas de las aproximaciones actuales en CD fueron diseñadas para hacer frente a la complejidad geométrica de escenas compuestas por un largo número de obstáculos complicados. La auto-colisión de los robots es generalmente manejada entre cuerpos considerados como un conjunto de objetos

independientes rígidos. Mientras tal aproximación es suficientemente apropiada para robots simples con un limitado número de articulaciones, esto se torna ineficiente cuando los aplicamos a articulaciones más complejas (ejemplo [30]).

Las técnicas de planificación de movimientos son aplicadas hoy en día en diversos dominios como animación [43, 57] y biología computacional [3, 4, 52] que involucran sistemas con muchos grados de libertad (DOF) los cuales requieren una nueva clase de algoritmos de CD. En aplicaciones de biología computacional, la detección de colisiones es un problema muy difícil, puesto que las macromoléculas, como una proteína pueden ser formados con gran cantidad de cadenas articuladas con arriba de 100 DOF. La Figura 2.6 muestra el modelo de una proteína de tamaño mediano y da muestra de la complejidad del correspondiente mecanismo articulado. El requerimiento para la técnica especifica de CD es por lo tanto crucial para evitar el alto costo cuadrático de enumerar todos los pares de átomos no unidos en los modelos con miles de átomos. En particular, el número para ser considerado para autocolisión se puede reducir drásticamente teniendo en cuenta las limitaciones estructurales impuestas por la estructura de la cadena cinemática. Solo pocos trabajos en CD [18, 36] se dirigen a este problema en específico de pruebas de auto-colisión para cadenas cinemáticas complejas.

2.2.2. BioCD

A continuación describe el algoritmo de BioCD[14] propuesto por Juan Cortés, Thierry Siméon y Vicente Ruiz fue desarrollado para hacer más eficientes la detección de colisión y la distancia computacional entre grandes cadenas moleculares articuladas, incluyendo auto-colisión eficiente dentro de cada cadena. BioCD está actualmente integrado dentro de planificadores de caminos en [10] para computar grandes movimientos de flexibilidad molecular.

Agrupación rígida de átomos

El cuerpo de un modelo molecular articulado está formados por grupos de átomos unidos rígidamente. Estos grupos pueden tener tamaños muy diferentes. Tomando provecho de la conocida estructura secundaria (Figura 4), $\alpha - helice$ y $\beta - sheets$ son considerados a menudo elementos rígidos para los métodos de modelización molecular. Los grupos rígidos de átomos pue-

den ser aún más largos, por ejemplo cuando se refieren a todos los elementos de la estructura secundaria en un dominio. En contra parte, en porciones de proteínas flexibles como en los ciclos de proteína, los grupos rígidos son mucho más pequeños y estos solo se preocupan por pocos residuos de átomos internos, como ilustra la Figura 2.6.

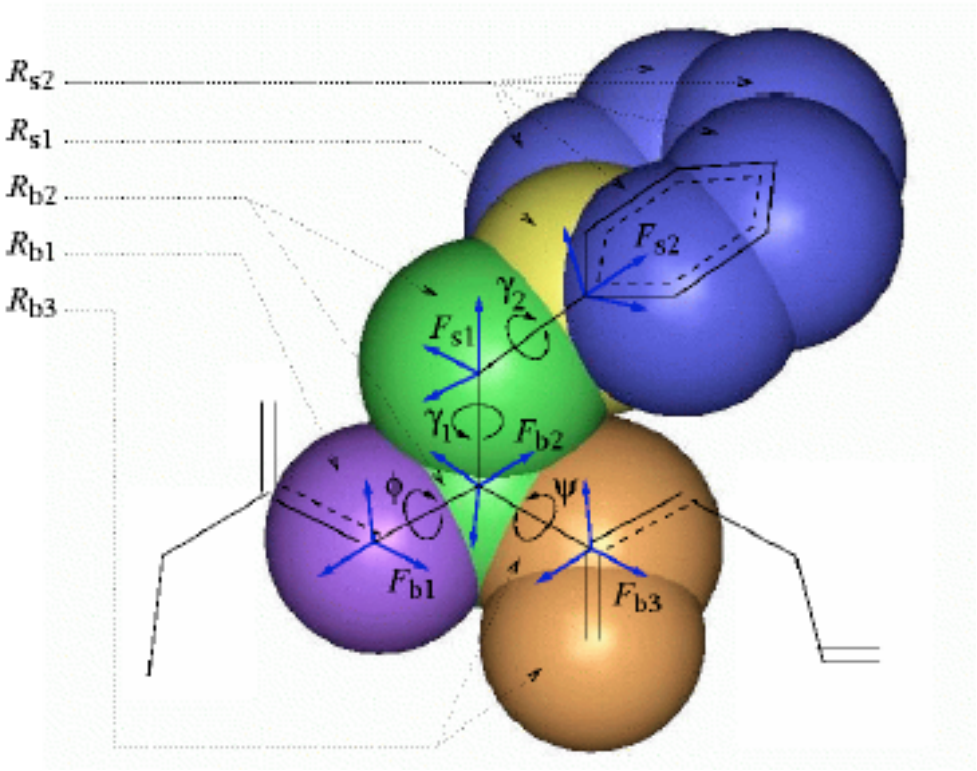


Figura 2.6: Modelo mecánico para un aminoácido flexible de una proteína. Está compuesto de cinco cuerpos rígidos, clasificados en: grupo de esqueletos rígidos $\{R_{b1}, R_{b2}, R_{b3}, R_{s1}, R_{s2}\}$.

Si se usa una aproximación clásica de cinemática de un robot, un sistema de coordenadas cartesianas puede conectar a cada cuerpo rígido. La localización relativa es definida como una matriz de transformación homogénea que es la función de rotación entre estos ángulos. La posición de un átomo en un grupo rígido con relación a su correspondiente cuadro es definido simplemente por un vector. Un método similar de modelado fue propuesto en [60].

Colisiones en cadenas moleculares

La detección de colisión aplicada a modelado molecular actúa como un filtro geométrico que no permiten estructuras con grande restricciones van der Waals (VDW). El filtro necesita ser tan selectivo como sea posible, pero no tanto para rechazar estructuras correctas. A continuación se discutirá como las restricciones energéticas pueden ser traducidas dentro de la distancia geométrica restrictiva entre la posición de los átomos.

La interacción de van der Waals entre átomo i y j depende de su distancia relativa, d, y una distancia equilibrada, d_0, determinada por el tipo de los dos átomos. La fuerza es poco atractiva a distancias medias, nula en $d = d_0$ y exponencialmente repulsiva en distancias pequeñas. Existe un límite de energía van der Waals que no puede ser compensado por cualquier otro componente de energía. Esta energía alcanza una fracción $0 < p < 1$ de la distancia equilibrada ($p \simeq 0{,}8$). Por lo tanto el detector de colisiones debe rechazar alguna estructura para la cual $d < p \times d_0$ para algún par de átomos.

Se necesitan manejar dos restricciones moleculares para el detector de colisiones para evitar rechazar una estructura valida. Primero, la interacción de van der Waals solo con referencia a átomos no enlazados, estos no son relevante entre átomos separados por tres o menos enlaces químicos. Por lo consiguiente, solo los pares de átomos separados por cuatro o más enlaces tienen que ser considerados para la colisión. Esta es una de las particularidades que cualquier CD molecular debe tener en cuenta. La evaluación de la distancia entre átomos topológicamente cercanos es, en consecuencia, inútil, y debe evitarse en la medida de lo posible en lugar de calcular todas las interacciones y seleccionando el que sea relevante en una etapa de post-procesamiento.

Otra restricción especifica de la aplicación molecular es que la distancia de colisión $p \times d$ depende en el tipo de los dos átomos que interactúan. Por ejemplo, es necesario para el modelo la presencia de enlaces de hidrógeno que acortan la distancia equilibrada entre un par de átomos específicos. En resumen, los criterios de colisión pueden ser escritos como:

$$d < M_{(t(i),t(j))} \text{ y } TopDist(i,j) > 3 \tag{2.10}$$

Donde M es una matriz cuadrada simétrica de distancias limites indexadas para los tipos de átomo y $TopDist(i,j)$ es la distancia de topología química entre los átomos i y j. Nota que las condiciones de geometría varían como la condición de la matriz son más general que el usual en la literatura de CD.

Algoritmo

El algoritmo de BioCD se basa en una jerarquía de dos niveles organizados en torno al concepto de grupos rígidos de átomos. Para aprovechar las condiciones SBMP antes mencionadas: Solo un conjunto de k grados de libertad son permitidos cambiar mientras todos los demás son bloqueados. La preselección de los grados de libertad puede cambiar ocasionalmente (cuando se define un nuevo problema de planificación de movimientos en la cadena molecular), pero muchas consultas al detector de colisiones pueden ser realizadas con el mismo conjunto de grados de libertad seleccionados. Si bien los grados de libertad seleccionados no cambian, muchos átomos en la cadena molecular no se someten a ningún desplazamiento relativo con respecto a otros átomos. La jerarquía de dos niveles permite de forma sencilla evitar pruebas inútiles entre dichos pares de átomos.

Un grupo rígido se define como el conjunto máximo de átomos conectados en la que no hay cambio en la distancia interna. Los grupos rígidos pueden tener diferentes tamaños dependiendo del grado de libertad seleccionado. Los grupos rígidos más pequeños corresponden a los cuerpos del resto del modelo articulado. Los grupos más grandes de átomos se crean a lo largo del segmento rígido del esqueleto de la cadena. , ejemplo $\alpha - helices$ o $\beta - sheet$ de la estructura secundaria. También se toman en cuenta los grupos rígidos cercanos cuyas posiciones relativas se mantienen fijas (e.g. parejas de $\beta - sheet$) son reunidos dentro de un grupo. La Figura 2.7 muestra los átomos de una cadena segmentada.

BioCD identifica los grupos rígidos de modelo molecular y construye una jerarquía para cada uno de ellos. Estas son las jerarquías de bajo nivel. Cada uno de ellos organiza los átomos del grupo rígido y la raíz representa el grupo en sí. La jerarquía de nivel superior se encarga de las raíces de bajo nivel. Descartando los pares de átomos cuya interacción no cambia de una iteración a otra (es decir, que tienen lugar dentro de un grupo rígido) se efectúa simplemente mediante no probar el nodo raíz representante del grupo rígido consigo mismo.

La jerarquía de dos niveles se induce mediante el requisito de tener un único nodo que representa a cada conjunto de los grupos de átomos rígidos. Esta es la mejor manera de excluir las pruebas entre los átomos con posiciones relativas fijas. Permite también aislar cerca de las partes de la jerarquía que no debe ser reconstruido, la maximización de su tamaño de manera que una bandera en un nivel de la jerarquía no sería capaz de hacer.

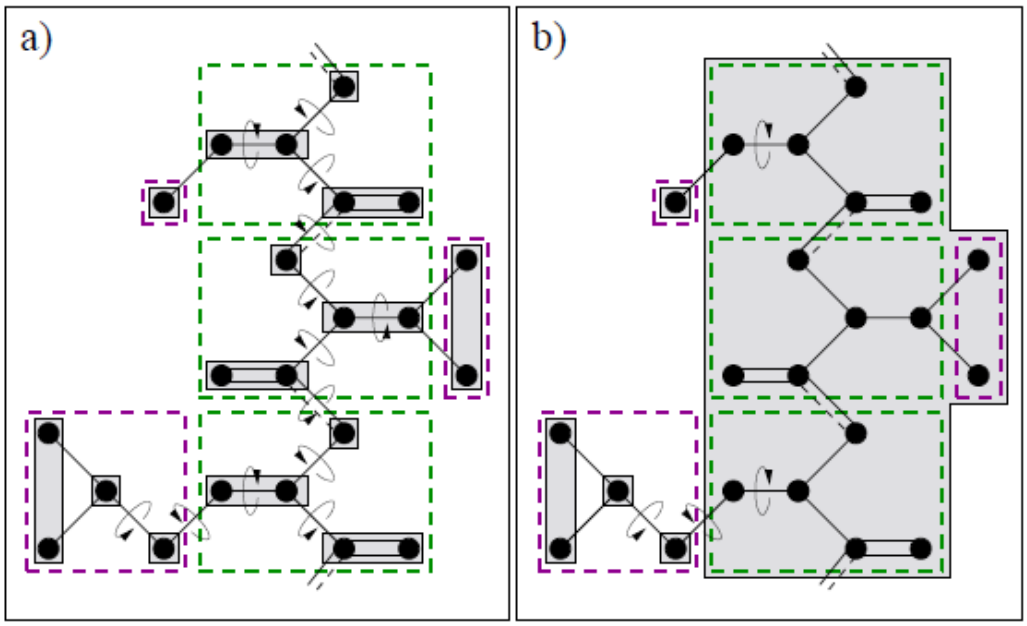


Figura 2.7: Representación de un segmento e para una proteína totalmente articulada (a) y el mismo segmento con solo dos cadenas laterales articuladas (b). Las cajas grises contienen a los grupos rígidos de átomos. Las cajas punteadas corresponden a los grupos de átomos básicos manejados por BioCD para comprobar las interacciones de corto alcance.

El interés de la jerarquía de adaptación espacial es doble. Primero, éstas permite una auto-detección de colisiones rápida, $O(n)$[1] en el peor caso (ver [37]) y también, estas no están limitadas a alguna topología particular. Sin embargo, ellas requieren en el peor caso $O(nlogn)$ de tiempo de construcción.

La jerarquía superior es presentada como un árbol binario de Ejes Alíneados a Cajas Limitadas (Axis Aligned Bounded Boxes (AABBs)), elegidas porque ellas permiten hacer pruebas rápidas de superposición, mientras que son volúmenes delimitadores más estrechos que las esferas. La propuesta es para organizar un conjunto de básicos AABBs de tal manera que la capa de un nodo sea un buen indicador de la aproximación espacial de los elementos límite.

Una vez definida la conformación, la fase de detección de colisiones sigue el algoritmo estándar para probar las colisiones entre los diferentes AABBs. El algoritmo omite la prueba consigo mismo. Para analizar esta etapa del algoritmo se utilizó la librería PQP[17] ya que permite analizar colisiones entre diferentes AABBs. Si se comprueba una auto-colisión en la jerarquía superior se pasa, a verificar los pares de átomos. Para el análisis en la jerarquía inferior únicamente se comprueban las diferencias entre los diferentes tipos de átomos para verificar que se cumplan las interacciones de van der Waals.

Este trabajo se encuentra más detallado en [14]. En él los autores explican más a detalle las funciones, jerarquías y trabajos relacionados a este detector de colisiones biológicas.

2.3. Visualización con la herramienta desarrollada

En esta sección se describirán los módulos con los que cuenta la herramienta desarrollada, se mostraran ejemplos de visualización de proteínas y de la implementación del detector de colisiones BioCD. La herramienta desarrollada se puede observar en la Figura 2.8, la Figura 2.9 y la Figura 2.10. A continuación se describe cada componente que se encuentra en la herramienta:

1. En esta sección se pueden abrir proteínas que hayan sido descargadas de la *Protein Data Bank* (http://www.rcsb.org/pdb/home/home.do), permitiendo al usuario utilizar la proteína *monomérica* que desee.

[1] Donde n es el numero de átomos en la proteína.

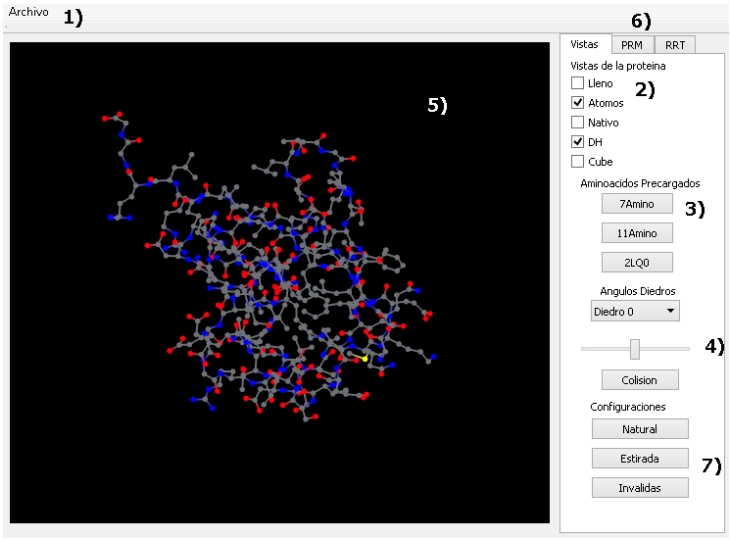


Figura 2.8: En esta interfaz se muestra la configuración de la vista y los ángulos diedros de la proteína. La proteína que se encuentra cargada es la 1UBQ.

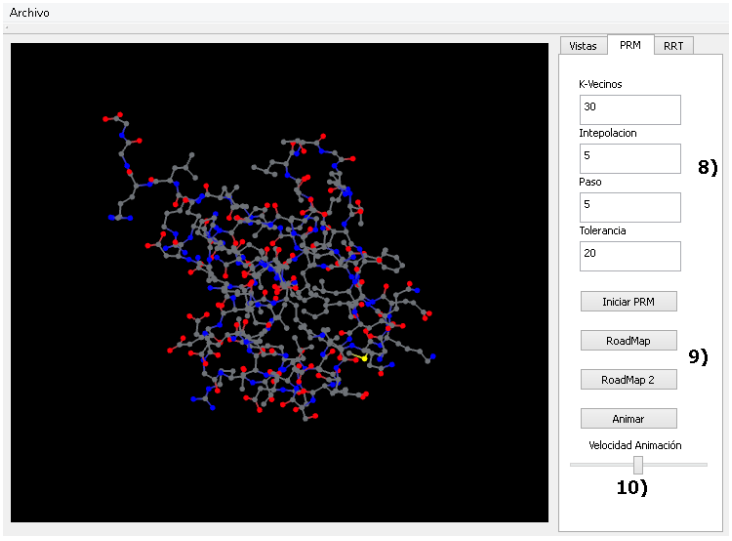


Figura 2.9: En esta interfaz se muestran los elementos necesarios para ejecutar el algoritmo PRM y su animación.

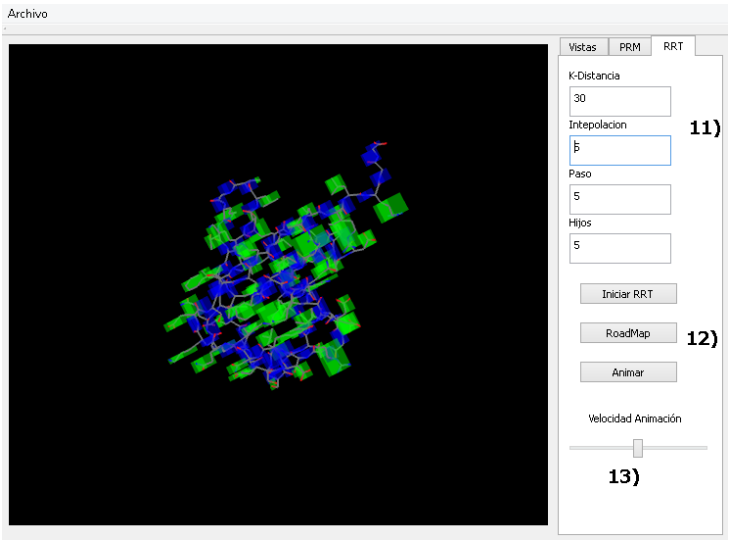


Figura 2.10: Esta interfaz muestra los elementos necesarios para aplicar el algoritmo RRT y su animación. En esta imagen se puede observar los AABBs necesarios para BioCD.

2. Estas casillas de verificación le permiten al usuario modificar la vista actual de la proteína.

3. Estos botones tienen proteínas precargadas que se utilizaron en este trabajo.

4. En esta sección se puede configurar el estado actual de la proteína, modificando los grados de libertad de los distintos ángulos diedros. Con el botón de colisión se puede verificar si la configuración realizada es una configuración correcta.

5. Este lienzo muestra la proteína que está cargada en el sistema dependiendo las casillas que estén verificadas en la sección 2.

6. Estas pestañas nos permite cambiar entre la sección de Vistas, PRM y RRT .

7. Permiten visualizar las configuraciones natural (nativa) y estirada de la proteína, y en el caso del botón invalido encontrar configuraciones no válidas.

8. Son los parámetros básicos necesarios para resolver el problema del plegado de proteínas utilizando PRM.

9. Son los botones que nos permiten inicializar y calcular el *roadmap* con PRM para resolver el problema del plegado de proteínas.

10. Una vez resuelto el *roadmap* se puede visualizar el camino encontrado por PRM presionando en el botón animar y se puede incluso modificar la velocidad de la animación con la barra horizontal.

11. Son los parámetros básicos necesarios para resolver el problema del plegado de proteínas utilizando RRT.

12. Son los botones que nos permiten inicializar y calcular el *roadmap* para RRT y resolver el problema del plegado de proteínas.

13. Una vez resuelto el *roadmap* se puede visualizar el camino encontrado por RRT presionando en el botón animar y modificar la velocidad de la animación con la barra horizontal.

Como se puede observar, se desarrollaron diferentes formatos para visualizar las proteínas. A continuación se describen estos formatos:

- **Lleno**: permite visualizar los átomos con el radio de van der Waals el cual asigna un radio a cada elemento que se encuentra en la proteína.

- **Átomos**: permite visualizar todos los átomos que se encuentran en una proteína, los radios son asignados de manera uniforme lo cual permite localizar la posición espacial del átomo.

- **Nativo**: permite visualizar la conformación nativa de la proteína, esta visa es muestra los enlaces con los que cuenta la proteína.

- **DH**: visualiza las conformaciones obtenidas después de modificar algún ángulo de libertad y aplicar el marco de referencia Denavit-Hartenberg. Al igual que la vista nativa, esta muestra la proteína como enlaces de átomos.

- **Cube**: esta vista permite visualizar los AABBs de la proteína que se está utilizando para la detección de colisiones con el algoritmo BioCD. En la Figura 2.10 se puede observar los AABBs necesarios para este algoritmo.

2.4. Conclusiones

En este capítulo se explicaron detalladamente las diferentes técnicas utilizadas para generar el modelo de la proteína que se aplicó en este trabajo, las cuales son la cinemática y la detección de colisiones.

La cinemática para el modelo molecular que se desarrollo está basada en los ángulos diedros de la cadena polipéptida de la proteína, los ángulos diedros laterales no son considerados en esta aproximación. Se realizó la cinemática inversa de la proteína para el cálculo de ciclos cerrados utilizando el algoritmo de coordenada descendiente.

A continuación se implementó el algoritmo BioCD para la detección de colisiones en la proteína, para poder verificar si la conformación formada por la modificación de algunos ángulos diedros es válida. Este algoritmo esta diseñado para detectar auto colisiones de una cadena cinemática grande, lo que lo hace una buena opción en el campo de las proteínas.

Las técnicas anteriores son importantes porque permiten la implementación de los algoritmos probabilísticos que son los que se desarrollaron en este trabajo. La cinemática directa permite la generación de diferentes conformaciones de la cadena polipéptida que se está trabajando y el analizador de colisiones permite verificar si estas conformaciones son válidas. En el siguiente capítulo explicaremos los algoritmos probabilistas y sus diferencias con las técnicas clásicas.

Capítulo 3

Plegado de proteínas

El primer trabajo sobre la aplicación de algoritmos de planificación de movimiento para el estudio de proteínas fue publicado en 1999 [47]. Desde entonces, muchos métodos fueron inspirados por diferentes algoritmos de planificación de movimientos y han sido aplicados a variados problemas de simulación molecular. La mayoría de los trabajos en esta linea usan PRM o RRT. En esta sección, se describirán las técnicas y como estas se aplican al campo de las proteínas.

3.1. PRM

La meta de la planificación de movimientos es calcular una secuencia de estados intermedios validos que transformen un estado inicial dado (el inicio) en un estado final seleccionado (meta). Sabemos que el problema del plegado de proteínas es mucho más complicado que la planificación de movimientos tradicional aplicada en robots, sin embargo, los algoritmos de planificación son con frecuencia descritos por utilizar una abstracción llamada espacio de configuraciones (C-space) que es lo suficientemente general para aplicarse a problemas no relacionados a la robótica.

3.1.1. Introducción

En esta sección describiremos el trabajo y aplicación de Probabilistic Roadmap Method (PRM) [27] para la planificación de movimientos en el ámbito del plegado de las proteínas.

Los trabajos anteriores en el área de la planificación de movimientos han aplicado esta metodología para problemas de plegado como es el plegado de cartoon [50] (con aplicaciones en empaquetamiento y ensamble [38]), y papel artesanal (estudiado en geometría computacional [38]), de esta manera se provee evidencia de la factibilidad de esta aproximación para determinar una secuencia en el problema del plegado de proteínas. Se puede ver la similitud entre los enfoques de plegado de cartón Figura 3.1 y plegado de proteínas Figura 3.2.

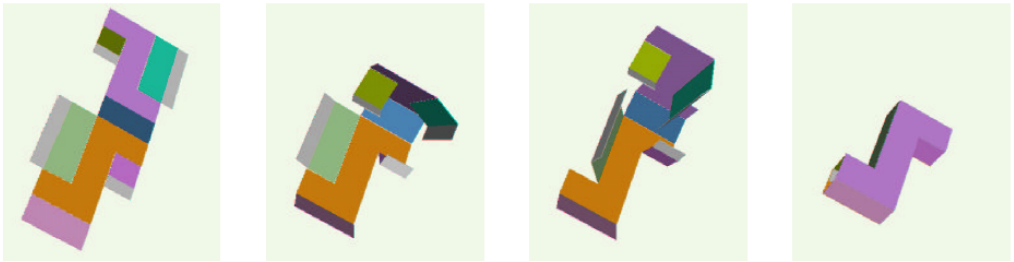


Figura 3.1: PRM aplicado al problema de plegado de cartoon.

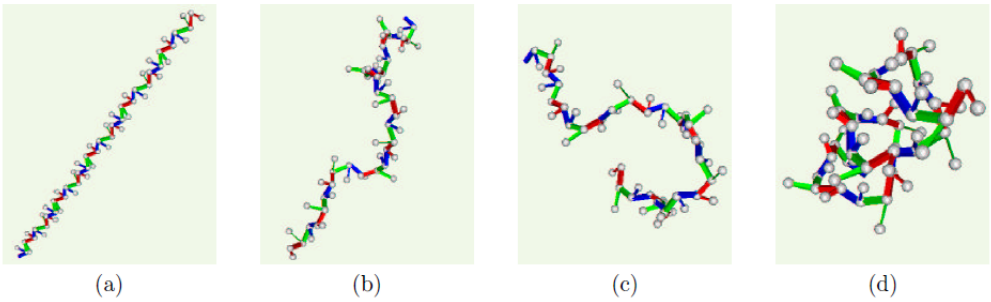


Figura 3.2: PRM aplicado al problema de plegado de proteínas.

Existen muchas investigaciones que tienen como objetivo determinar el estado nativo de una proteína plegada [45, 39]. En este trabajo de tesis asumimos que el estado nativo es conocido, y se enfoca en el proceso de plegado, es decir como de un estado inicial se llega al estado nativo. En este trabajo, el estado nativo de la proteína es obtenido de la base de datos de proteínas **Protein Data Bank**.

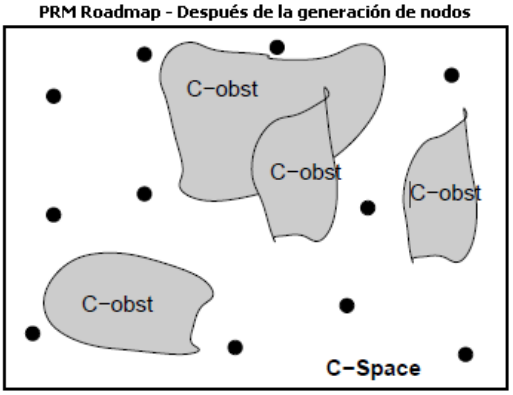
Dado un ambiente descrito y un objeto (el robot), el objetivo de la planificación de movimientos en PRM es encontrar un camino factible que lleve al objeto de un punto dado de inicio a una meta [32].

Este trabajo esta basado en una aproximación PRM para la planificación de movimientos [3]. En términos generales PRM trabaja muestreando puntos **aleatoriamente** en el espacio de configuraciones (C-Space), y se almacenan de los puntos muestreados los que tienen el grado de factibilidad requerido (por ejemplo, una configuración libre de colisión de un objeto móvil, ver Figura 3.3(a). Estas configuraciones se conectan para formar un grafo, o roadmap, utilizando algún método de planificación local para conectar las configuraciones **cercanas** (ver Figura 3.3(b)). Durante el proceso de consulta, son conectados al grafo la configuración inicial y la configuración meta, y después se utiliza un método de búsqueda en grafos como A* o Dijkstra (ver Figura 3.3(c)).

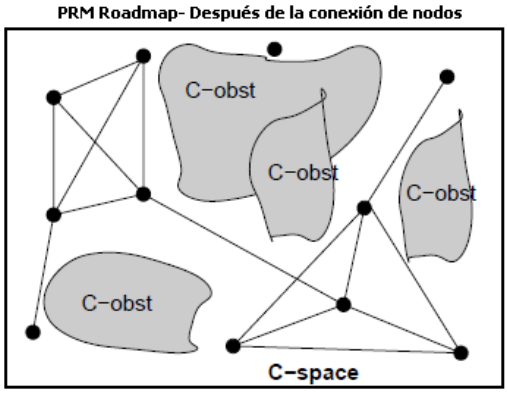
Una de las grandes ventajas de PRM es que se puede aplicar de una manera muy simple, incluso a problemas con gran dimensionalidad en el espacio de configuraciones, requiere solo la habilidad de generar configuraciones aleatorias en el C-space, y entonces probar la factibilidad para estas (la conexión local se efectúa con un método capaz de unir 2 configuraciones factibles y este respeta la cinemática del objeto en cuestión).

El problema del plegado de proteínas tiene una notable diferencia de la aplicación usual de PRM. Primero, la tradicional detección de colisiones es remplazada por una conformación preferente de bajas energías. En general, los potenciales pequeños, son las conformaciones más estables, y el estado nativo de la proteína es el mínimo global. Segundo, en PRM, a menudo se considera suficiente encontrar algún camino factible que conecte la configuración inicial y objetivo. Para el problema del plegado de proteínas, sin embargo, es de vital interés la calidad del camino, y en particular, se encuentran caminos con energías favorables. De esta manera las técnicas tipo PRM se pueden aplicar al problema del plegado de proteínas, en la Figura 3.4 se muestra un ejemplo de esta aplicación.

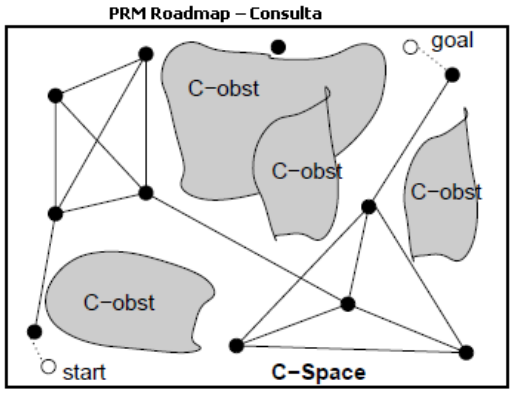
Para aplicar técnicas PRM en el problema de plegado de proteínas, primero se necesita considerar el modelado molecular el cual se explicó en el primer avance de tesis. En la siguiente sección se explicara como se construye el roadmap, la generación de conformaciones, etapa de conexión y etapa de consulta para el problema de plegado de proteínas.



(a)

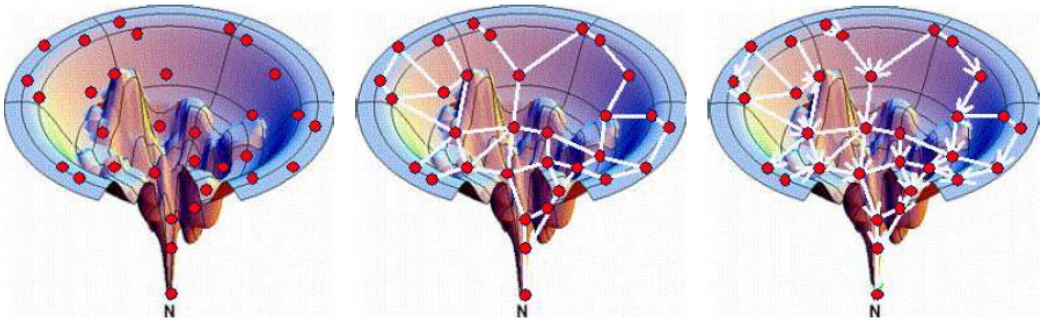


(b)



(c)

Figura 3.3: Un roadmap en el C-space. Un roadmap: (a) después de la generación de nodos, (b) después de la fase de conexión, y (c) utilizado para resolver una consulta.



(a) (b) (c)

Figura 3.4: Un roadmap para el proceso de plegado de proteínas que muestra el potencial energético en el C-space. (a) después de la generación de nodos (el muestreo es más denso alrededor de N, conocida como la configuración nativa), (b) después de la fase de conexión, y (c) utilizando caminos de plegamiento para extraer caminos a la estructura nativa.

3.1.2. Modelo proteico

Ya se ha hablado del modelado molecular, pero es importante retomar algunas ideas con el fin de entender cómo aplicar estas estructuras con las técnicas PRM en el problema del plegado de proteínas y ver su correspondencia con el espacio de configuraciones. Como se mencionó los grados de libertad que se manejaran en este modelo son los ángulos diedros de la proteína Φ y Ψ de cada aminoácido. De esta manera los grados de libertad manejados en cada proteína están relacionados con la cantidad de aminoácidos que tenga.

En este trabajo no se considera la posición absoluta u orientación de la proteína, una conformación de $n+1$ aminoácidos puede ser abstraída como un vector de, $2n$ Φ y Ψ ángulos (el primero y último ángulo de libertad no contribuyen información), cada uno en el rango de $[0, 2\pi)$, con el ángulo 2π igual a 0, lo cual es naturalmente asociado a un círculo en el plano, denotado como S^1. Así, el espacio de conformaciones (C-space) para la proteína con $n+1$ aminoácidos puede ser expresado como:

$$C = \{q : q \in S^1 \times S^1 \times ... \times S^1\} \tag{3.1}$$

Donde son $2n$ copias de S^1.

Esta definición es conocida en el campo de las proteínas como conformación, las conformaciones son relativas a las configuraciones en PRM. De esta

manera es posible generar diferentes conformaciones de una misma proteína, sin embargo, antes de empezar la etapa de muestreo es necesario entender las medidas que serán utilizadas posteriormente en PRM.

3.1.3. Métricas de distancia

Una vez entendido el modelo de la proteína, este se puede definir como un vector de ángulos Φ y Ψ. A continuación se describirá como medir la distancia entre dos conformaciones diferentes. Estas medidas son necesarias para las etapas de conexión y generación de nodos en PRM.

$$d_E(C_a, C_b) = \sqrt{\frac{(\Phi_1^a - \Phi_1^b)^2 + (\Psi_1^a - \Psi_1^b)^2 + ... + \Phi_n^a - \Phi_n^b)^2 + (\Psi_n^a - \Psi_n^b)^2}{2n}} \tag{3.2}$$

La segunda métrica de distancia es la distancia de la raíz media (root mean square distance o RMSD), esta medida no es entre los ángulos diedros sino entre los átomos del mismo tipo entre dos conformaciones. Para dar un ejemplo si la cadena lateral fuera tomada como un único átomo R, se tendría un total de 6 átomos: C, C_α, R, O, N y H. Las coordenadas de los átomos se denotarían como x_1 a x_{6n} y la distancia $d_R(C_a, C_b)$ entre dos conformaciones diferentes $C_a(x_1^a, x_2^a, ..., x_{6n}^a)$ y $C_b(x_1^b, x_2^b, ..., x_{6n}^b)$ esta definida como:

$$d_R(C_a, C_b) = \sqrt{\frac{\|x_1^a - x_1^b\|^2 + \|x_2^a - x_2^b\|^2 + ... + \|x_{6n}^a - x_{6n}^b\|^2}{6n}} \tag{3.3}$$

Debido a que en esta métrica se mide la distancia de los átomos, la posición y orientación de la proteína puede cambiar los valores de la medida. La solución para este problema ha sido propuesta en [24, 20]. El RSMD entre dos conformaciones es definido como el valor mínimo de $d_R(C_a, C_b)$:

$$RMSD = (C_a, C_b) = min d_R(C_a, C_b) \tag{3.4}$$

Una alternativa para disminuir este problema es utilizar únicamente los átomos C_α de cada aminoácido para el cálculo de el RMSD.

En este trabajo se utilizó la métrica euclidiana por trabajar con los ángulos diedros y evitar problemas con la posición u orientación de la proteína y así realizar menos cálculos al comparar conformaciones.

3.1.4. Generación de nodos

Una conformación $q \in C$ puede ser generada asignándole valores a cada ángulo Φ/Ψ en el rango permitido $[0, 2\phi)$. Ya que se conocen todos los ángulos Φ/Ψ de la nueva conformación, las coordenadas de cada átomo en el sistema se calculan, y estos son usados para determinar la energía potencial de la conformación. El nodo q es aceptado y agregado al roadmap basado en su potencial de energía $E(q)$ con la siguiente probabilidad:

$$P(\text{acepta q}) = \begin{cases} 1 & \text{Si } E(q) < E_{min} \\ \frac{E_{max} - E(q)}{E_{max} - E_{min}} & \text{Si } E_{min} \leq E(q) \leq E_{max} \\ 0 & \text{Si } E(q) > E_{max} \end{cases} \tag{3.5}$$

Esta prueba de aceptación, puede ayudar a retener más nodos en regiones de poca energía, estas energías también son usadas cuando se construyen roadmaps para el ligamento [47, 5]. Una configuración con cadenas superpuestas, tiene mayor potencial y es más probable que sea rechazada durante la generación de nodos.

Si se consideraran los ángulos de los diferentes ángulos diedros en **grados** enteros, se tendría que existen un total de 360^{2n} diferentes conformaciones para una proteína que cuenta con n aminoácidos. Debido a que el problema tiene un espacio de configuraciones con gran dimensionalidad, el muestro uniforme simple que se describe a continuación puede ser computacionalmente intenso y tener problemas para muestrear alrededor de la configuración nativa.

Muestreo uniforme:

1. Se genera una conformación c por muestreo aleatorio de los ángulos Φ / Ψ de todos los aminoácidos.
2. Si el potencial(c) satisface el umbral.
3. Se almacena c.
4. Fin Si.
5. Repetir paso 1 a 4 hasta que n nodos sean generados.

Este enfoque es impráctico al utilizar el método de muestreo uniforme, debido a esto se buscaron otros métodos de muestreo, con el fin de muestrear mejor en problemas con gran dimensionalidad. En el problema de plegado de

proteínas, se da por hecho que se conoce el estado nativo de la proteína, por este motivo es posible utilizar diferentes estrategias que permitan sesgar la búsqueda entre el estado inicial y final.

Entre las diferentes técnicas de muestreo existentes, en este trabajo se utilizó la técnica de muestreo **Gaussiano** el cual se realizó alrededor de la configuración nativa. Esta estrategia utiliza la configuración nativa para que el muestreo capture la información alrededor de la configuración nativa. Algunas estrategias similares se han aplicado correctamente en aplicaciones típicas de la robóticas [55, 56, 46, 2], donde el sobre muestreo cerca de pasajes estrechos en el espacio de configuraciones es crucial para algunos problemas. El problema de pasajes estrechos con los robots, es similar al problema de muestreo alrededor de la conformación nativa. A continuación se muestra el pseudocódigo de esta técnica.

Muestreo Gaussiano, entrada: v (vector de desviaciones estándar):

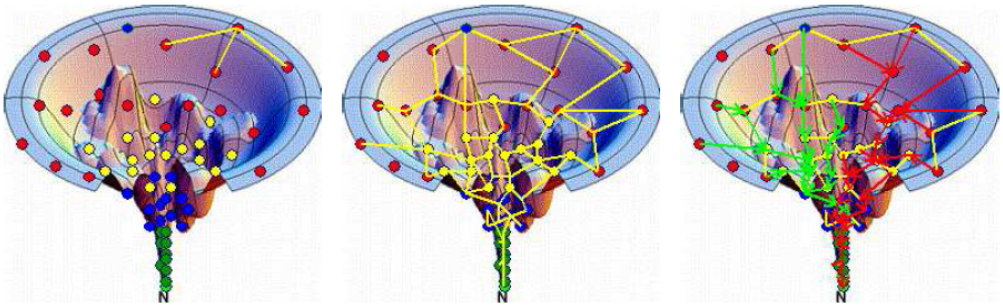
1. Para i = 1 al tamaño de(v).
2. Se genera una conformación c utilizando un muestreo Gaussiano usando la desviación estándar $v[i]$ para todos los ángulos Φ / Ψ, usando los ángulos Φ / Ψ del estado nativo como el centro.
3. Si el potencial(c) satisface el umbral.
4. Se almacena c.
5. Fin Si.
6. Fin Para.
7. Repetir paso 1 a 4 hasta que n nodos sean generados.

La etapa de generación de nodos se tiene que realizar con cuidado ya que si esta no se realiza correctamente no se encontrará un camino por falta de muestras en áreas importantes.

3.1.5. Conexión de nodos

La conexión de nodos es la segunda fase que se debe realizar en la construcción del roadmap. El objetivo es para obtener un roadmap que capture caminos de baja energía. Para cada nodo del roadmap, se buscan los vecinos

más cercanos que estén a una distancia k, y entonces conectar usando un planificador local como se muestra en la Figura 3.5(a). Después, esto es repetido para todos los nodos del roadmap, creando un grafo que captura caminos con baja energía que se puede ver en la Figura 3.5(b). En nuestro trabajo se utilizaron diferentes distancias siendo las mejores k=20 y k=30, la métrica de distancia usada fue la distancia euclidiana en el espacio de conformaciones C. Para la cantidad de vecinos que puede tener una configuración se utilizaron diferentes cantidades, donde se encontró que con cantidades menores a 10 no se conectaba correctamente el roadmap y mayores de 50 el tiempo de cómputo se dispara. Por esta razón el número de vecinos se estableció en 30.



(a) (b) (c)

Figura 3.5: Roadmaps (a) y (b) muestran la etapa de conexión y (c) muestra la captura de los caminos de plegado con el potencial energético, donde N es la estructura nativa.

Para la planificación local se utilizaron fórmulas que se describirán a continuación y el detector de colisiones BioCD [14] que se explicó en el capítulo anterior para evaluar la interpolación.

Cuando dos nodos q_1 y q_2 cumplen con la distancia requerida, se obtienen sus configuraciones c_1 y c_2, la probabilidad de que estos dos nodos se conecten depende de la diferencia entre sus energías potenciales $\Delta E_i = E(c_{i+1} - E(c_i)$.

$$P_i = \begin{cases} e^{\frac{-\Delta E_i}{kT}} & \text{Si } \Delta E_i > 0 \\ 1 & \text{Si } \Delta E_i \leq 0 \end{cases} \tag{3.6}$$

De esta manera se mantiene el ajuste entre dos diferentes configuraciones, evitando que existan grandes diferencias energéticas entre configuraciones

diferentes. Una función similar fue utilizada, con diferentes probabilidades en [47].

3.1.6. Consulta del roadmap

El roadmap resultante puede ser usado para encontrar caminos factibles entre la configuración inicial y meta. Para nuestro problema de plegado de proteínas la configuración meta es siempre parametrada como el estado nativo de la proteína y es agregada al roadmap. Para la configuración inicial en este trabajo la proteína es estirada poniendo el valor de todos los ángulos diedros Φ y Ψ a cero y agregando esta nueva configuración al roadmap. Una vez conectadas la configuración inicial y meta, se utiliza el algoritmo de búsqueda A* para encontrar el camino con la menor varianza energética. En la Figura 3.5(c) se muestran diferentes caminos obtenidos realizando las consultas en el roadmap.

3.2. RRT

Los algoritmos RRT (Rapidly-exploring Random Tree) se diseñaron para búsquedas en espacios con gran dimensionalidad en el cual se puede construir un árbol que explore este espacio. Los árboles en este algoritmo son construidos incrementalmente de muestras aleatorias alrededor de las configuraciones actuales, esto permite explorar en diferentes zonas de un espacio. A continuación describiremos a fondo el algoritmo y la estructura de estas técnicas.

3.2.1. Introducción

A continuación se describirán los algoritmos de planificación de caminos basados en árboles aleatorios de exploración rápida (RRT), técnica desarrollada por Steven M. LaValle y su grupo de colaboradores en la universidad de Illinois, EU [33, 9, 34, 31]. La base de estos métodos es la construcción incremental de árboles de búsqueda que intentan explorar rápida y uniformemente el espacio de estados, ofreciendo beneficios similares a los obtenidos por otros métodos exitosos de planificación aleatoria, como los métodos de roadmap probabilísticos (PRM) [27] antes vistos.

El tipo de problemas considerados por el enfoque RRT están formulados en términos de seis componentes:

1. **Espacio de estados (configuraciones)**: Un espacio topológico, C.
2. **Valores límite** : $c_{ini} \in C$ y $C_{meta} \subset C$.
3. **Detector de colisión**: Una función, $C : C \Rightarrow verdadero, falso$, que determina sí las restricciones globales son satisfechas desde el estado c. Esta podría ser una función binaria o real.
4. **Entrada**: Un conjunto U, que especifica el conjunto de controles o acciones que pueden afectar al estado.
5. **Simulador incremental**: Dado el actual estado, $c(t)$, y las entradas aplicadas sobre un intervalo de tiempo, $\{u(t')|t \leq t' \leq t + \Delta t\}$, calcular $c(t + \Delta t)$.
6. **Métrica**: Una función real, $p : C \times C \Rightarrow [0, \infty)$, la cual especifica la distancia entre pares de puntos en C.

La planificación de caminos generalmente es vista como una búsqueda en el espacio de configuraciones, C, para un camino continuo desde un estado inicial, c_{ini} a una región meta C_{meta} o un estado meta c_{meta}. Se asume que se tiene un conjunto complicado de restricciones diferenciales sobre C y cualquier camino solución debe mantener al estado dentro de este conjunto. Un detector de colisión reporta si un estado dado, c, satisface las restricciones del problema. Generalmente se utiliza la notación, C_{libre} para referirse al conjunto de estados que satisfacen las restricciones globales. Se asignan restricciones locales y diferenciales a través de un conjunto de entradas (o controles) y de un simulador incremental. Estos dos componentes especifican los posibles cambios en el estado. El simulador incremental puede definirse por integración numérica de una ecuación de transición de estado. Finalmente, se define una métrica para indicar la cercanía de pares de puntos en el espacio de estados.

Los árboles aleatorios de exploración rápida (RRT, del inglés, Rapidly-Exploring Random Tree) fue presentado en [33] como un algoritmo de planificación para búsqueda rápida en espacios de altas dimensiones que tienen tanto restricciones algebraicas (provenientes de los obstáculos) como restricciones diferenciales. La idea clave es dirigir la exploración hacia regiones no

exploradas del espacio tomando puntos en el espacio de estados e incrementalmente **jalar** el árbol hacia ellos.

El algoritmo básico de construcción de RRT, se muestra en el siguiente código. En cada iteración se intenta extender el árbol agregado un nuevo vértice en dirección a un estado seleccionado aleatoriamente. La función **EXTENDER**, ilustrada en la Figura 3.7, selecciona del árbol el vértice más cercano a un estado dado. Este vértice se elige de acuerdo a una métrica, p. La función **NUEVO_ESTADO** hace un movimiento hacia c aplicando una entrada u para algún incremento Δt. Esta entrada puede seleccionarse aleatoriamente o probando todas las posibles entradas eligiendo aquella que produzca un nuevo estado tan próximo como sea posible a c. **NUEVO_ESTADO** utiliza implícitamente una función de detección de colisiones para determinar si el nuevo estado (y todos los estados intermedios) satisfacen las restricciones globales. Si **NUEVO_ESTADO** se cumple, el nuevo estado junto con la entrada se representa por medio de c_{nuevo} y u_{nuevo}, respectivamente. Pueden ocurrir tres situaciones: *Alcanzado*, el nuevo vértice alcanza al estado muestreado c; *Avanzado*, un nuevo vértice c_{nuevo} /c es agregado al árbol RRT. *Atrapado*, **NUEVO_ESTADO** falla en producir un nuevo estado que se encuentre en C_{libre}.

3.2.2. RRT en plegado molecular

Al igual que como se comentó en el capítulo de PRM la aplicación de RRT en el campo del plegado molecular tiene sutiles diferencias con el problema general. Los grados de libertad en este modelo de proteínas son los ángulos diedros de la cadena péptida Φ y Ψ.

El espacio de conformaciones se reutilizara definiéndose como un vector de $2n$ Φ y Ψ ángulos, cada uno en el rango $[0, 2\pi)$, con el ángulo 2π igual a 0. Así el espacio de conformaciones para la proteína con $n + 1$ aminoácidos en RRT puede ser expresado al igual que en PRM. Con este enfoque se pueden definir diferentes configuraciones que permitan expandir el árbol.

En RRT se reutilizaron las métricas de distancia las cuales permiten limitar la extensión de los árboles que serán generados durante la etapa de extensión. Para verificar que se puede realizar el movimiento entre dos árboles se utilizó el mismo planificador local de PRM con el detector de colisiones BioCD [14] y los cálculos potenciales de las conformaciones.

Para este trabajo la versión básica de RRT no es capaz de obtener una planificación adecuada debido a la amplitud del espacio de conformaciones.

Data: c_{ini}
Result: $\mathcal{T}$
$\mathcal{T}$.ini(c_{ini});
for $k = 1$ *a* K **do**
  $c_{aleat} \leftarrow$ ESTADO_ALEATORIO();
  EXTENDER($\mathcal{T}, c_{aleat}$);
end
Regresar $\mathcal{T}$

Algoritmo 1: CONSTRUIR_RRT

Data: $\mathcal{T}, c$
Result: *Estado*
$c_{prox} \leftarrow$VECINO_MAS_PROXIMO(c, $\mathcal{T}$);
if *NUEVO_ESTADO(*c, c_{prox}, c_{nuevo}, u_{nuevo}*)* **then**
  $\mathcal{T}.AgregarVertice(c_{nuevo})$;
  $\mathcal{T}.AgregarArista(c_{prox}, c_{nuevo}, u_{nuevo})$;
  if $c_{nuevo} = c$ **then**
    Estado = *Alcanzado*;
  end
  else
    Estado = *Avanzado*;
  end
end
Estado = *Atrapado*;

Algoritmo 2: EXTENDER

Figura 3.6: Algoritmo básico de construcción del RRT

Por esta razón en la siguiente sección se explicara el algoritmo RRT bidireccional que mejora el desempeño de RRT.

3.2.3. RRT bidireccional

Esta especialización de RRT está inspirada en las técnicas clásicas de búsqueda bidireccional, lo cual asegura un mejor desempeño al hacer crecer dos árboles de exploración, uno desde c_{ini} y el otro a partir de c_{meta}; se obtiene una solución cuando los dos árboles se encuentran. Para una búsqueda simple, la implementación es directa, sin embargo, la construcción RRT debe guiarse para asegurar que ambos árboles se encuentren antes de cubrir el espacio

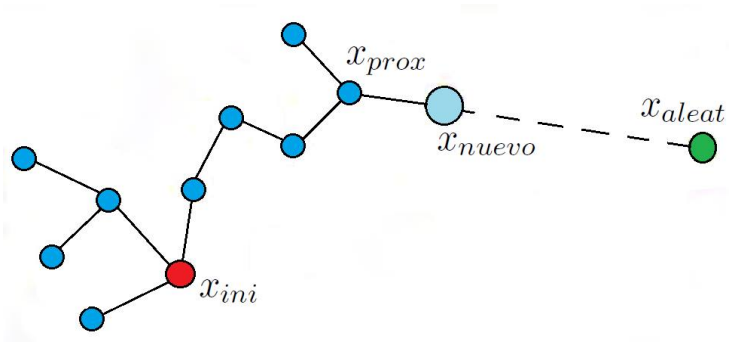


Figura 3.7: Operación de la función EXTENDER.

entero y permitir una unión eficaz.

A continuación se muestra el algoritmo de **RRT_BIDIRECCIONAL**. El **RRT_BIDIRECCIONAL** divide el tiempo de cómputo entre dos procesos: 1) explorar el espacio de estados; 2) intentar crecer los árboles uno hacia el otro. Siempre existen dos árboles T_a y T_b hasta que se enlazan y se encuentra una solución. En cada iteración un árbol crece, y se intenta conectar el nuevo vértice con aquel más cercano en el otro árbol. Entonces se invierten los roles intercambiando los árboles. El algoritmo actual intenta que los árboles crezcan uno hacia el otro en la mitad de tiempo, con lo cual se obtiene un buen rendimiento.

Se consideran algunas variaciones del planificador anterior. Puede remplazar cualquier ocurrencia de EXTENDER con CONECTAR en el **RRT_BIDIRECCIONAL**. Cada reemplazo hace a la operación más agresiva.

La implementación de **RRT_BIDIRECCIONAL** que se utilizara en este trabajo está basada en [42], este trabajo retoma el modelo de [3, 49] en el que está basado este trabajo.

3.3. Resultados comparativos

Para este estudio comparativo entre las dos técnicas implementadas en la herramienta, se realizaron pruebas con 3 diferentes estructuras proteicas. Se diseñaron 2 moléculas con 7 aminoácidos (7 amino), otra con 11 aminoácidos (11Amino) similarmente como se creó la molécula 10-ALA propuesta en [3]. Además se utilizó una molécula más grande descargada de la base de datos de proteínas de nombre 2LQ0 que cuenta con 25 aminoácidos.

Data: c_{ini}, c_{met}
$\mathcal{T}_a$.ini(c_{ini}); $\mathcal{T}_b$.ini(c_{met}); **for** $k = 1$ *a* K **do**
  $c_{aleat} \leftarrow$ ESTADO_ALEATORIO(); **if** *EXTENDER*($\mathcal{T}_a, c_{aleat}$) ¡¿*Atrapado* **then**
    if *EXTENDER*($\mathcal{T}_b, c_{nuevo}$) = *Alcanzado* **then**
      Regresar Camino($\mathcal{T}_a, \mathcal{T}_b$);
    end
  end
  Intercambiar($\mathcal{T}_a, \mathcal{T}_b$);
end
Regresar Fallo;

Algoritmo 3: RRT_BIDIRECCIONAL

Data: $\mathcal{T}, c$
repeat
  EXTENDER($\mathcal{T}, c$);
until $s \neq$ *Avanzado*;
Regresar s;

Algoritmo 4: CONECTAR

Figura 3.8: Algoritmo básico de construcción de RRT_BIDIRECCIONAL

Los parámetros utilizados en el algoritmo de PRM en el problema de plegado de proteína son:

- 30 k-vecinos.
- Una interpolación entre configuraciones de 5.
- Una distancia euclidiana de 20 para el Roadmap [49].
- Un tamaño de paso de 5 para la conexión.

Para probar PRM se realizaron 10 pruebas con cada una de estas moléculas obteniendo los resultados que se muestran en la Tabla 3.1.

Los parámetros utilizados en el algoritmo de RRT-Bidireccional para el problema de plegado de proteína son:

- Distancia euclidiana de 20 [42].
- Interpolación entre configuraciones de 5.

Molécula	Nodos	Aristas	Caminos	Tiempo(Horas)
7Amino	722	3107	6	1.5
11Amino	836	47886	4	3.2
2LQ0	2343	107702	3	4.1

Cuadro 3.1: Resultados obtenidos con la herramienta.

- Un tamaño de paso de 5 para la conexión.
- Número de hijos por árbol de 2.

Para probar la técnica RRT también se realizaron 10 pruebas con las diferentes moléculas. Los resultados se muestran en la Tabla 3.2.

Molécula	Nodos	Tiempo(Horas)
7Amino	29	0.15
11Amino	54	0.37
2LQ0	115	0.66

Cuadro 3.2: Resultados obtenidos con la herramienta.

Como se puede observar la técnica RRT-Bidireccional tiene un tiempo de ejecución menor que PRM para encontrar un camino de plegado y la menor generación de estructuras intermedias. Sin embargo, RRT solo devuelve un camino por prueba y PRM tiene la capacidad de encontrar más de un camino por prueba.

Dependiendo de las necesidades del usuario sería la recomendación que se haría de que técnica utilizar. Si el usuario desea encontrar múltiples configuraciones, PRM es la técnica que se debe utilizar. Por el contrario si únicamente se desea observar un único comportamiento del proceso de plegado RRT es la técnica a utilizar.

3.4. Conclusión

Los algoritmos probabilísticos que se utilizan en la robótica para la planificación de movimientos mostraron ser una buena opción para resolver el problema de plegado de proteínas. Aunque esta tendencia de utilizar estas técnicas en el campo molecular ya tiene varios años, aún le hace falta madurar para realizar mejores modelos de planificación.

Los resultados obtenidos por PRM y RRT fueron correctos ya que se encontraron caminos para los procesos de plegado para las diferentes proteínas estudiadas. El tiempo de ejecución fue menor que el reportado en la literatura para técnicas como Monte Carlo y se obtuvieron caminos con una buena calidad potencial.

La importancia de estas técnicas en el campo de las proteínas, es porque no únicamente se pueden enfocar en el problema de plegado molecular, sino también a la tarea de ligado de proteínas que es útil en la creación de nuevos fármacos y para el diseño de nuevos nano materiales.

Capítulo 4

Conclusiones y trabajo futuro

Este trabajo de tesis integró diferentes técnicas de robótica para construir una herramienta que resuelva el problema del proceso de plegado de proteínas. Esta herramienta es una alternativa para aproximar los caminos de plegado que no requieren muchos detalles para su simulación.

Al desarrollar esta herramienta se implementó el modelo molecular desarrollado en [3] para las proteínas, este modelo es la base fundamental de este trabajo, con este modelo podemos realizar la validación de las conformaciones ayudados del algoritmo de detección de colisiones [14], y generar caminos de plegado utilizando las técnicas probabilísticas [42, 49, 3].

En el Capítulo 3, Sección 3.3 se observan los resultados obtenidos comparando las dos técnicas probabilísticas PRM y RRT. Como se pudo observar las técnicas basadas en RRT mostraron una mayor eficiencia en tiempo en comparación a las basadas en PRM, sin embargo, las técnicas basadas en PRM fueron capaces de encontrar más de un camino de plegamiento.

En trabajos futuros, se planea utilizar esta herramienta con más proteínas, para observar el comportamiento de la herramienta y ver si se pueden buscar mejores opciones de muestreo que ayuden a mejorar la calidad de los roadmaps generados por PRM y RRT. Para la siguiente versión de la herramienta será necesario utilizar las tecnologías GPUs para disminuir el tiempo de procesamiento en los cálculos realizados para la generación del roadmap (operaciones con matrices, cálculo del potencial energético, etc.) , esto permitirá estudiar más proteínas en menos tiempo.

Finalmente se utilizaran estas técnicas siendo asesorados por expertos, con el fin de verificar su aplicación en malformaciones de proteínas como la proteína que genera el problema de *encefalopatía espongiforme bovina* y

así poder desarrollar medicamentos para prevenir estas malformaciones.

Bibliografía

[1] E. Alm and D. Baker. Prediction of protein-folding mechanisms from free-energy landscapes derived from native structures. *Proceedings of the National Academy of Sciences*, 96(20):11305–11310, 1999.

[2] Nancy M. Amato, O. Burchan Bayazit, Lucia K. Dale, Christopher Jones, and Daniel Vallejo. Obprm: An obstacle-based prm for 3d workspaces, 1998.

[3] Nancy M. Amato, Ken A. Dill, and Guang Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *Journal of Computational Biology*, 10(3/4):239–255, 2003.

[4] Mehmet Serkan Apaydin, Douglas L. Brutlag, Carlos Guestrin, David Hsu, and Jean-Claude Latombe. Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion. In *Proceedings of the Sixth Annual International Conference on Computational Biology*, RECOMB '02, pages 12–21, New York, NY, USA, 2002. ACM.

[5] O. Burchan Bayazit, Guang Song, and Nancy M. Amato. Ligand binding with obprm and haptic user input: Enhancing automatic motion planning with virtual touch, 2000.

[6] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Res*, 28:235–242, 2000.

[7] Wouter Boomsma and Thomas Hamelryck. Full cyclic coordinate descent: solving the protein loop closure problem in ca space. *BMC Bioinformatics*, 6(1):1–10, 2005.

[8] Joseph D. Bryngelson, José Nelson Onuchic, Nicholas D. Socci, and Peter G. Wolynes. Funnels, pathways, and the energy landscape of protein folding: A synthesis. *Proteins: Structure, Function, and Bioinformatics*, 21(3):167–195, 1995.

[9] P. Cheng, Z. Shen, and S. M. Lavalle. RRT-Based Trajectory Design for Autonomous Automobiles and Spacecraft. *Archives of Control Sciences*, 11(3-4):167–194, 2001.

[10] J. Cortes, T. Simon, V. Ruiz de Angulo, D. Guieysse, M. Remaud-Simeon, and V. Tran. A path planning approach for computing large-amplitude motions of flexible molecules, 2005.

[11] David G. Covell. Folding protein a-carbon chains into compact forms by monte carlo methods. *Proteins: Structure, Function, and Bioinformatics*, 14(3):409–420, 1992.

[12] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1989.

[13] V Daggett and M Levitt. Realistic simulations of native-protein dynamics in solution and beyond. *Annual Review of Biophysics and Biomolecular Structure*, 22(1):353–380, 1993. PMID: 8347994.

[14] Vicente Ruiz de Angulo, Juan Cortés, and Thierry Siméon. BioCD : An efficient algorithm for self-collision and distance computation between highly articulated molecular models. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[15] Yong Duan and Peter A. Kollman. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science*, 282(5389):740–744, 1998.

[16] R M Fine, H Wang, P S Shenkin, D L Yarmush, and C Levinthal.

[17] S. Gottschalk, M. C. Lin, and D. Manocha. Obbtree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 171–180, 1996.

[18] Leonidas Guibas, An Nguyen, Daniel Russel, and Li Zhang. Collision detection for deforming necklaces. In *Proceedings of the Eighteenth Annual Symposium on Computational Geometry*, SCG '02, pages 33–42, New York, NY, USA, 2002. ACM.

[19] Sunil Hadap, Dave Eberle, Pascal Volino, Ming C. Lin, Stephane Redon, and Christer Ericson. Collision detection and proximity queries. In *ACM SIGGRAPH 2004 Course Notes*, SIGGRAPH '04, New York, NY, USA, 2004. ACM.

[20] Berthold K. P. Horn, H.M. Hilden, and Shariar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOURNAL OF THE OPTICAL SOCIETY AMERICA*, 5(7):1127–1135, 1988.

[21] P. Jiménez, F. Thomas, and C. Torras. 3d collision detection: A survey. *Computers and Graphics*, 25:269–285, 2000.

[22] Adzhubei Ivan A. Wolf Yuri I. Koonin Eugene V. Kondrashov Alexey S. Sunyaev Shamil Jordan I. King, Kondrashov Fyodor A. A universal trend of amino acid gain and loss in protein evolution., 2005.

[23] James J. Kuffner Jr. and Steven M. Lavalle. Rrt-connect: An efficient approach to single-query path planning. In *Proc. IEEE Intĺ Conf. on Robotics and Automation*, pages 995–1001, 2000.

[24] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 34(5):827–828, September 1978.

[25] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, December 1983.

[26] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, Aug 1996.

[27] Lydia E. Kavraki, P. Svestka, Jean-Claude Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[28] A Kolinski and J Skolnick. Monte carlo simulations of protein folding. ii. application to protein a, rop, and crambin. *Proteins*, 18:353–66, 1994 Apr 1994.

[29] Brodsky B Berman HM. Kramer RZ, Bella J. The crystal and molecular structure of a collagen-like peptide with a biologically relevant sequence. *Journal of Molecular Biology*, 2001.

[30] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue. Self-collision detection and prevention for humanoid robots. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3, pages 2265–2270, 2002.

[31] J.J. Kuffner and S.M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 2, pages 995–1001 vol.2, 2000.

[32] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[33] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University, 1998.

[34] Steven M. Lavalle, James J. Kuffner, and Jr. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Directions*, pages 293–308, 2000.

[35] Michael Levitt. Protein folding by restrained energy minimization and molecular dynamics. *Journal of Molecular Biology*, 170(3):723 – 764, 1983.

[36] Itay Lotan, Fabian Schwarzer, Dan Halperin, and Jean-Claude Latombe. Efficient maintenance and self-collision testing for kinematic chains. In *Proceedings of the Eighteenth Annual Symposium on Computational Geometry*, SCG '02, pages 43–52, New York, NY, USA, 2002. ACM.

[37] Itay Lotan, Fabian Schwarzer, and Jean Claude Latombe. Algorithm and data structures for efficient energy maintenance during monte carlo simulation of proteins. *Journal of Computational Biology*, 11:2004, 2004.

[38] Liang Lu and S. Akella. Folding cartons with fixtures: a motion planning approach. *Robotics and Automation, IEEE Transactions on*, 16(4):346–356, Aug 2000.

[39] E. Huang S. Subbiah M. Levitt, M. Gerstein and J. Tsai. Protein folding: the endgame. *Annu. Rev. Biochem.*, pages 66:549–579, 1997.

[40] R. Müller-Cajar and R. Mukundan. Triangulation: A new algorithm for inverse kinematics. *Proceedings of Image and Vision Computing New Zealand 2007*, 2007.

[41] Victor Muñoz and William A. Eaton. A simple model for calculating the kinetics of protein folding from three-dimensional structures. *Proceedings of the National Academy of Sciences*, 96(20):11311–11316, 1999.

[42] Shuvra Kanti Nath, Shawna Thomas, Chinwe Ekenna, and Nancy M. Amato. A multi-directional rapidly exploring random graph (mrrg) for protein folding. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, BCB '12, pages 44–51, New York, NY, USA, 2012. ACM.

[43] Julien Pettré, Jean-Paul Laumond, and Thierry Siméon. A 2-stages locomotion planner for digital actors. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 258–264, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[44] Madhusudan Raghaven and Bernard Roth. Kinematic analysis of the 6r manipulator of general geometry. In *The Fifth International Symposium on Robotics Research*, pages 263–269, Cambridge, MA, USA, 1990. MIT Press.

[45] G N Reeke. Protein folding: Computational approaches to an exponential-time problem. *Annual Review of Computer Science*, 3(1):59–84, 1988.

[46] M. Saha and J.-C. Latombe. Finding narrow passages with probabilistic roadmaps: the small step retraction method. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 622–627, Aug 2005.

[47] Amit Pal Singh, Jean-Claude Latombe, and Douglas L. Brutlag. A motion planning approach to flexible ligand binding. In Thomas Lengauer, Reinhard Schneider, Peer Bork, Douglas L. Brutlag, Janice I. Glasgow, Hans-Werner Mewes, and Ralf Zimmer, editors, *ISMB*, pages 252–261. AAAI, 1999.

[48] Guang Song. *A motion planning approach to protein folding*. PhD thesis.

[49] Guang Song and Nancy M. Amato. Using motion planning to study protein folding pathways. In *Proceedings of the Fifth Annual International Conference on Computational Biology*, RECOMB '01, pages 287–296, New York, NY, USA, 2001. ACM.

[50] Guang Song and N.M. Amato. A motion-planning approach to folding: from paper craft to protein folding. *Robotics and Automation, IEEE Transactions on*, 20(1):60–71, Feb 2004.

[51] M.J.E. Sternberg. *Protein Structure Prediction : A Practical Approach: A Practical Approach*. Oxford University Press, USA, 1996.

[52] Miguel L. Teodoro, George N. Phillips Jr., and Lydia E. Kavraki. A dimensionality reduction approach to modeling protein flexibility. In *Proceedings of the 2002 ACM International Conference on Research in Computational Biology (RECOMB 2002)*, pages 299–308. ACM Press, ACM Press, April 2002.

[53] Gino van den Bergen. *Collision Detection in Interactive 3D Environments*. Number v. 1 in Collision Detection in Interactive 3D Environments. Taylor & Francis, 2004.

[54] L.-C.T. Wang and C.C. Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *Robotics and Automation, IEEE Transactions on*, 7(4):489–499, 1991.

[55] Dawen Xie and N.M. Amato. A kinematics-based probabilistic roadmap method for high dof closed chain systems. In *Robotics and Automation,*

2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, volume 1, pages 473–478 Vol.1, April 2004.

[56] Jeffery H. Yakey, Steven M. LaValle, and Lydia E. Kavraki. Randomized path planning for linkages with closed kinematic chains. *IEEE T. Robotics and Automation*, 17(6):951–958, 2001.

[57] Katsu Yamane, James J. Kuffner, and Jessica K. Hodgins. Synthesizing animations of human manipulation tasks. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 532–539, New York, NY, USA, 2004. ACM.

[58] M. Zhang and Lydia E. Kavraki. Solving molecular inverse kinematics problems for protein folding and drug design. In *Currents in Computational Molecular Biology*, pages 214–215. ACM Press, ACM Press, April 2002. Book includes short papers from The Sixth ACM International Conference on Research in Computational Biology (RECOM 2002), Washington, DC, 2002.

[59] Ming Zhang and Lydia E. Kavraki. A new method for fast and accurate derivation of molecular conformations. *Journal of Chemical Information and Computer Sciences*, 42(1):64–70, 2002.

[60] Ming Zhang, Lydia E. Kavraki, I Lydia, and E. Kavraki. A new method for fast and accurate derivation of molecular conformations. *Journal of Chemical Information and Computer Sciences*, 42:64–70, 2002.

[61] Ming Zhang, R. Allen White, Liqun Wang, Ronald Goldman, Lydia Kavraki, and Brendan Hassett. Improving conformational searches by geometric screening. *Bioinformatics*, 21:2005.

Apéndice A

cálculo de energía potencial

Las proteínas son cadenas de aminoácidos, principalmente consisten de átomos de carbón, oxígeno, nitrógeno, e hidrógeno. Lo átomos dentro de una proteína no solo interactúan con otras proteínas, también interactúan con solventes que se encuentran a su alrededor. Hay interacciones covalentes a través de los enlaces y las interacciones que no son con enlaces como la interacción electrostática y las fuerzas de van der Waals [51, 35]. Éste es el resultado de todas las fuerzas que maneja una proteína para plegarse bajo condiciones de plegado, o las fuerzas para desplegar cuando las condiciones de plegado cambian (por ejemplo, un cambio de temperatura).

La interacción puede ser expresada en términos potenciales. En ese caso, las fuerzas actuales pueden ser deducidas tomando la derivada de la potencia. En muchos casos, una representación potencial llamada función potencial es más conveniente que tratar directamente con fuerzas. La función potencial es un término escalar que resume los principios físicos de la interacción molecular, y es por lo tanto independiente de cualquier proteína especifica. La fórmula general puede ser expresada como [35].

$$\begin{aligned} U_{tot} &= \sum_{enlaces} \frac{1}{2} K_b (b - b_0)^2 + \\ &\quad \sum_{angulos} \frac{1}{2} K_a (\theta - \theta_0)^2 + \\ &\quad \sum_{torsiones} K_\phi [1 + cos(n\phi - \delta)] + \end{aligned}$$

$$\sum_{parejaAtomos} (A/r_{ij}^{12} - B/r_{ij}^{6}) + \sum_{electrostatica} \frac{q_i q_j}{k r_{ij}}$$

El primer término es la asociación potencial con el largo de enlace y son pasados sobre todos los enlaces, el segundo término es el potencial asociado con los ángulos de enlace y son pasados por todos los enlaces angulares, el tercer término es el potencial asociado con los ángulos diedros y pasado sobre todos los ángulos diedros, el cuarto término es el potencial asociado con el potencial de van der Waals el cual es pasado por todos los pares de átomos, y el último término es el potencial asociado con la interacción electrostática. b_0 y θ_0 son los valores ideales para el largo de enlace y el ángulo de enlace, y K_b, K_a y K_ϕ son las fuerzas constantes. A y B son parámetros para la interacción de van der Waals. K es la función del efecto dieléctrico para el medio. Es importante mencionar que los primeros tres términos corresponden a interacciones entre los enlaces, mientras que los últimos dos no.

En general, el potencial se define en términos de todos los átomos de la molécula, y las funciones potenciales que realmente calcula todas las interacciones de todos los pares de átomos. Esta función es lo más precisa disponible para estos modelos. Desafortunadamente, también son muy caros para calcular debido al gran número de átomos, incluso en una pequeña proteína.

La función potencial en la cual se basa este trabajo fue desarrollada por Levitt [35]. Se aproxima el potencial de todos los átomos al ignorar algunos tipos de interacciones, como los términos potenciales asociados al largo de enlace y ángulo de enlace (primer y segundo termino de la ecuación anterior), y también la interacción entre cadenas laterales. En el modelo implementado en este trabajo para el cálculo potencial [3, 49] , se trata la cadena lateral como un simple y largo **átomo** R que es colocado donde se encuentra el átomo C_b en la Figura A.1 se muestra un ejemplo de este modelo para el aminoácido alanina.

De esta manera para el cálculo del potencial cada aminoácido de la proteína a ser estudiado se compone de 6 átomos: un átomo de nitrógeno (N), un hidrógeno (H), un oxígeno(O), dos carbonos (C y C_α), y R. A continuación se describe el potencial que se obtuvo para este nuevo modelo:

$$U_{tot} = \sum_{restricciones} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_0\}$$

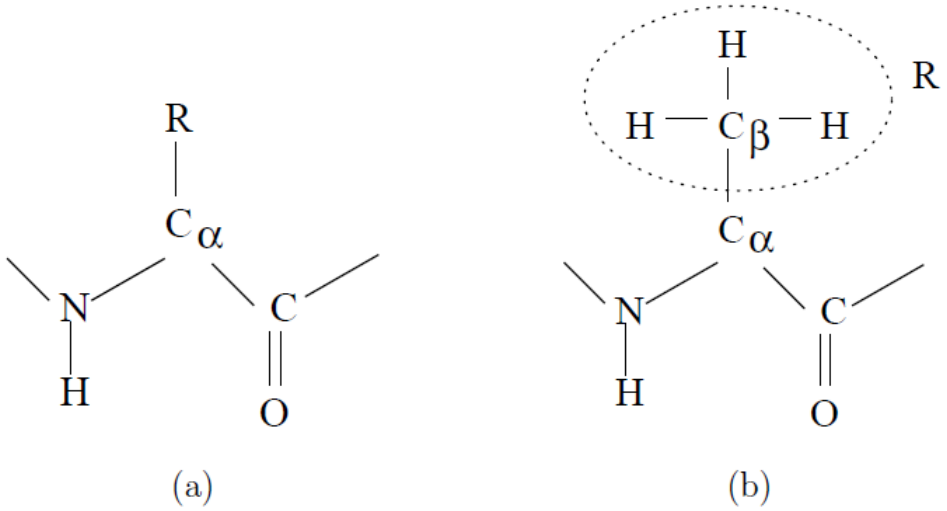


Figura A.1: Nuestro modelo descrito con el aminoácido alanina para la cadena lateral. (a) El modelo de un aminoácido normal, (b) la cadena lateral del aminoácido Alanin es compuesta de un átomo de carbono y tres de hidrógeno, los cuales son modelados como un gran carbón R [3, 49].

$$+ \sum_{parejaAtomos} (A/r_{ij}^{12} - B/r_{ij}^{6})$$

El cual es similar al potencial utilizado en [35]. El primer término representa las restricciones que favorecen la estructura secundaria conocida a través de los enlaces de hidrógeno de la cadena principal y los enlaces de disulfuro. El segundo término son los parámetros de van der Waals que se explicaron con anterioridad.

Las restricciones se pueden obtener a partir de la estructura del estado nativo de la proteína que normalmente se obtiene a partir de los archivos de las proteínas que se pueden obtener de Protein Data Bank (PDB) [6]. El parámetro K_d es ajustado a 100 kj/mol, y las distancias son $d_0 = d_c = 2A$, y d_i es la separación entre los pares de átomos que forman enlaces de hidrógeno o disulfuro en el estado nativo. El segundo término corresponde a la interacción de van der Waals entre los seis átomos que modelan cada aminoácido en el modelo.

Realizar el cálculo de todos los pares para calcular el potencial de van der Waals (el segundo término de la suma) puede ser computacionalmente inten-

so. Para reducir este costo, se utilizó una función que aproxima el potencial de van der Waals y considera únicamente la contribución de las cadenas laterales [48]. Para una configuración dada, se calcula las coordenadas de todos los átomos R (del modelo potencial) para todos los residuos. Si cualquiera dos átomos R están lo suficientemente cerca, se obtiene un potencial alto.

Específicamente, si la distancia mínima entre todos los pares de átomos R (r_{min}) es menor que $1.0A$, se obtiene un valor muy grande. Si r_{min} es más grande que $1.0A$ pero menor que $2.4A$, se regresa un valor más largo que E^{gem}_{max}, pero menor que E^{con}_{max}, donde E^{gen}_{max} y E^{con}_{max} son los umbrales máximos de generación de nodos y conexión de nodos, respectivamente. Los umbrales se establecen de manera que solo se aceptan conformaciones con potencial menor que el umbral. Por último, si r_{min} es más grande que $2.4A$, entonces se procede a usar la siguiente fórmula para calcular el potencial:

$$P_h(c) = \sum_{restricciones} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_0\} + E_{hidrofobico}$$

En resumen, para una conformación C, se calcula su potencial $P(c)$ de la siguiente manera:

$$P(c) = \begin{cases} 2 * E^{con}_{max} & \text{Si } r_{min} < 1{,}0A \\ 2 * E^{gen}_{max} & \text{Si } 1{,}0A \leq r_{min} \leq 2{,}4A \\ P_h(c) & \text{Si } r_{min} \geq E_{max} \end{cases}$$

La información de los enlaces de hidrógeno y enlaces disulfuro pueden ser obtenidos corriendo el programa llamado **DSSP** [25], en este trabajo se utilizó la versión del software como servicio para evitar la introducción manual de los datos de la proteína. El segundo término es el efecto hidrófobo y se considera de la siguiente manera. Se asigna un valor de la hidrofobicidad de 1 a todos los residuos de aminoácidos hidrófobos y 0 al resto. Cuando las cadenas de dos aminoácidos hidrofóbicos (el átomo R del modelo potencial) están a una distancia d_{R_h}, el potencial es reducido por E_h. En este trabajo $d_{R_h} = 6A$ y $E_h = 20kj/Mol$, que son otros dos parámetros en el cálculo potencial [49, 3].

Apéndice B

Arquitectura de la aplicación

La herramienta de plegado de proteínas desarrollada en este trabajo fue implementada en el lenguaje de programación C++, con la plataforma de QT. La arquitectura desarrollada para esta aplicación se divide en 4 módulos principales: **Datos, Vista, Algoritmia, Estructuras de Datos**. En la Figura B.1 se muestra la estructura de la aplicación desarrollada.

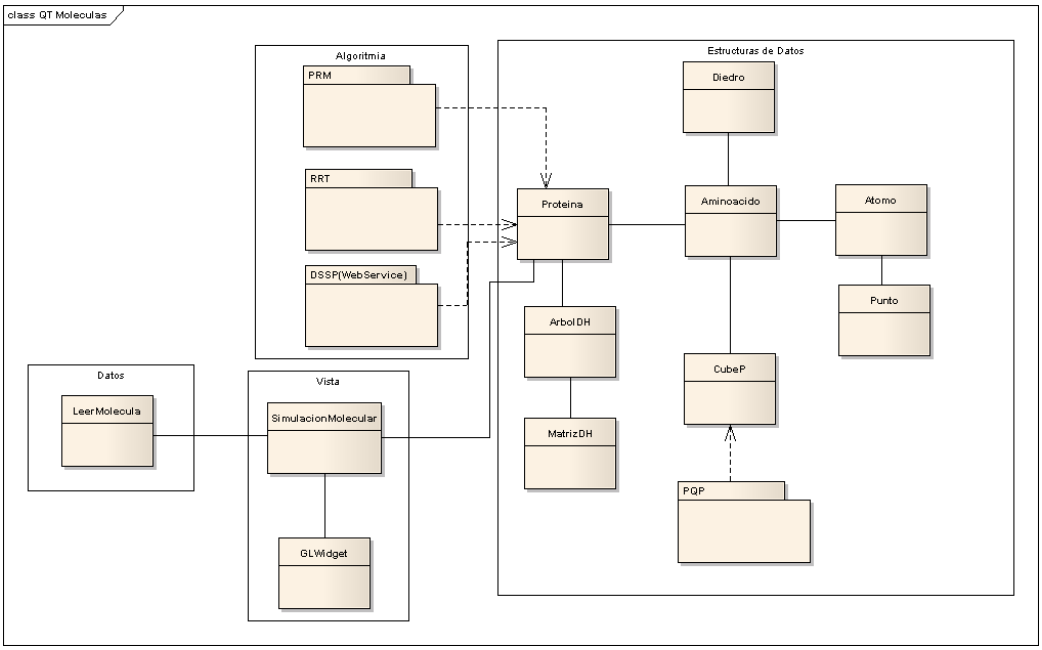


Figura B.1: Arquitectura modular de la aplicación.

A continuación se explicara cada módulo de la arquitectura para describir su funcionalidad en la aplicación.

Módulo de Datos: contiene la clase LeerMolecula, este modulo se encarga de leer los archivos con formato PDB que son descargados de Protein Data Bank para que la aplicación los pueda utilizar. Este modulo contiene un analizador léxico que permite transformar el archivo introducido a la estructura de tipo **Proteina**.

Módulo de Vista: este módulo es el encargado de la interfaz que utilizaran los usuarios finales. Existen dos clases en este módulo la clase SimulacionMolecular que es la interfaz de usuario generada bajo la plataforma de QT, y la clase GLWidget la cual se encarga de realizar todas las tareas de OpenGL que se realizan en la aplicación.

Módulo de Estructuras de Datos: este módulo contiene el núcleo de estructuras de datos necesarias para desarrollar la simulación molecular, entre las estructuras que tiene se puede ver, átomos, aminoácidos, proteínas entre otros. En este módulo se agregó el detector de colisiones PQP como un paquete ya que también es una estructura de datos que de utilidad.

Módulo de Algoritmia: este módulo contiene la algoritmia utilizada en esta aplicación para realizar el plegado molecular. Este módulo fue dividido en tres diferentes paquetes: **PRM, RRT y DSSP**.

El **paquete PRM** contiene todos los algoritmos y estructuras de datos para aplicarlo. Este algoritmo es basado en grafos por lo cual contiene las estructuras de nodos y aristas. En la Figura B.2 se muestra el contenido del paquete PRM.

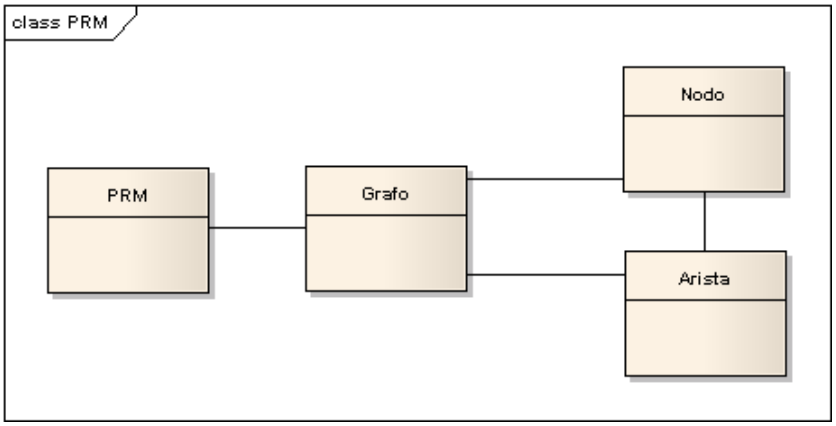


Figura B.2: Diagrama de clases del paquete PRM.

El **paquete RRT** es más pequeño que el paquete PRM, ya que en esta técnica la estructura de datos a utilizar es un árbol únicamente. En la Figura B.3 se muestra la estructura interna de RRT.

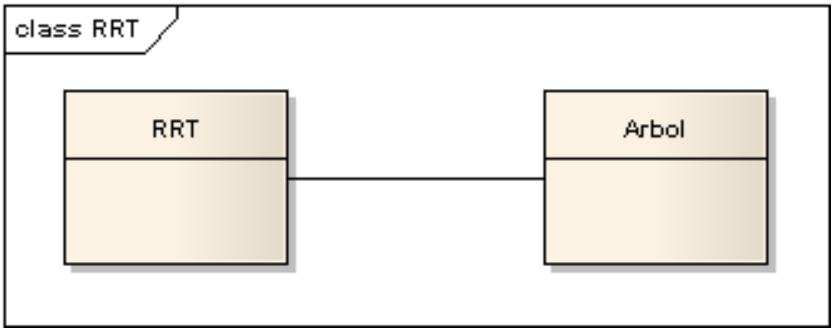


Figura B.3: Diagrama de clases del paquete RRT.

El **paquete DSSP** es el encargado de consumir el servicio Web del programa DSSP diseñado por Wolfgang Kabsch y Chris Sander para estandarizar las estructuras secundarias de las proteínas, este paquete ayuda a calcular el potencial de la conformación de la proteína que se estudia.