Universidad del Bío-Bío, Chile

Facultad de Ciencias Empresariales

Departamento de Sistemas de Información

Ludificación de docking proteína-proteína enfocado en incrementar set de complejos de proteínas

Anteproyecto de título presentado por Natalia Sepúlveda Mora
de la Carrera Ingeniería Civil Informática
Dirigida por Dra. Tatiana Gutiérrez Bunster

2021

Resumen

El estudio de las interacciones proteína-proteína es una campo de mucho interés para la comprensión de los procesos biológicos existentes. Existen tanto métodos experimentales como computacionales para determinar la estructura tridimensional de los complejos proteicos. El primero, a pesar de ser el más preciso, es también el que presenta mayores dificultades técnicas y costos asociados. Es por esta razón, que nace la necesidad del desarrollo de programas computacionales, que generan predicciones acerca de los posibles modos de unión entre las proteínas involucradas en una interacción, con un menor costo asociado en comparación a los métodos empíricos. A este tipo de programas, se les conoce como docking proteína-proteína. Sin embargo, estos algoritmos aún no se encuentran en una etapa madura, por lo cual día a día se siguen desarrollando nuevos algoritmos para mejorar su rendimiento. Dentro de los nuevos enfoques computacionales, se encuentra la ludificación del proceso de docking a través de plataformas interactivas, donde los usuarios intervienen en la búsqueda del espacio conformacional de unión entre las moléculas que forman la interacción proteica. En este trabajo, se pretende desarrollar un sistema con este tipo de enfoque, a través de un sistema de docking proteico interactivo.

Palabras Clave — Docking proteína-proteína, interacciones proteína-proteína, complejos proteicos

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

Las interacciones proteína-proteína representan un importante rol en la mayoría de los procesos celulares y extracelulares, donde la formación de un complejo tiene una consecuencia funcional. Un complejo proteico corresponde a un conjunto de dos o más proteínas que interactúan entre sí en un mismo tiempo y lugar. El análisis de sus estructuras es un tema de gran interés, ya que su estudio entre otras cosas facilita el descubrimiento de nuevos medicamentos. Lo anterior, debido a que generalmente un fármaco tiene como objetivo la inhibición de la función de una proteína mediante su unión a ella.

Cada complejo posee una forma tridimensional única determinada por la secuencia de aminoácidos que lo componen. Su estructura puede ser descrita en cuatro niveles; primaria, secundaria, terciaria y cuaternaria, es única e influye de manera importante en la función del complejo.

Existen diversas técnicas para determinar de forma experimental la estructura tridimensional, como por ejemplo; cristalografía de rayos X y espectroscopia mediante resonancia magnética nuclear, información que se encuentra almacenada principalmente en el Protein Data Bank (PDB). Sin embargo, comparada al avance sobre proteínas individuales, la determinación de complejos proteína-proteína ha sido mucho menor, principalmente debido a las dificultades técnicas y costos involucrados en los métodos empíricos. Debido a esto, surgió la necesidad de utilizar herramientas computacionales, como el docking proteína-proteína, para predecir la estructura de complejos proteicos sin utilizar métodos experimentales.

En general, los programas de docking proteína-proteína existentes predicen la estructura de los complejos a partir de las estructuras individuales de las moléculas involucradas, a través de la generación de una gran cantidad de modos de unión candidatos y una posterior discriminación de las mejores soluciones a través de un proceso de ranking.

Debido a que los set de datos de complejos de proteínas aún no son lo suficientemente grandes como se esperaría para poder realizar estudios acerca de las interacciones entre proteínas y para predecir éstas con herramientas computacionales, resulta útil desarrollar de una forma

novedosa, a través de un juego, una herramienta que permita aumentar estos conjuntos de datos. Además de esto, existen conjuntos de datos disponibles actualmente para ser utilizados como entradas en el sistema.

1.1. Hipótesis

Es posible, a través de un sistema de docking proteína-proteína interactivo, generar complejos proteicos a partir de las estructuras individuales de las moléculas involucradas, utilizando la capacidad cognitiva de las personas para agilizar el proceso de acoplamiento.

1.2. Objetivos

Para dar respuesta a la hipótesis definida es necesario determinar de qué forma se llegará a ella. Para ello se definen los objetivos de este proyecto.

El objetivo principal de este trabajo es desarrollar un sistema de docking proteína-proteína interactivo, que utilice como entradas estructuras individuales de proteínas contenidas en el PDB. La plataforma permitirá al usuario visualizar tridimensionalmente las estructuras de ambas proteínas. Este deberá tomar las proteínas y ensamblarlas intentando generar el complejo proteico con el mayor puntaje posible (de acuerdo a las funciones del modelo de docking utilizado). De esta forma se espera generar un set de datos con los complejos de mayor puntaje obtenido por los jugadores. De esta manera se buscar aumentar los conjuntos existentes de estructuras de complejos de proteína.

Para lograr este objetivo se pretende: Revisar literatura acerca de los diferentes modelos existentes de docking proteína-proteína para la elección del modelo base del juego; Revisar literatura de sistemas de docking interactivos, existentes hasta el momento sobre docking molecular; Revisar literatura de los conjuntos de datos que existen disponibles actualmente de complejos de proteínas y proteínas individuales que servirán como entradas en el juego; Diseñar e implemedentar un juego en base al modelo de docking y set de proteínas individuales seleccionados

1.3. Metodología de trabajo

En la siguiente sección se describe la metodología que se utilizará para realizar la investigación.

- Para revisar la literatura se estudiarán de forma sistemática aquellos trabajos que existan acerca de modelos de docking proteína- proteína y de ludificación del proceso de docking molecular.
- Luego de revisar la literatura, se seleccionará el modelo de docking más adecuado según lo estudiado para utilizar como base en el desarrollo del juego.
- También se estudiará de forma sistemática la literatura existente acerca de los conjuntos de datos que hay disponibles acerca de complejos de proteínas y de proteínas individuales, seleccionando aquellos que sean más adecuados para el desarrollo del juego.

- Se determinará de acuerdo a los artículos estudiados sobre juegos de docking existentes el más adecuado para desarrollar docking proteína-proteína.
- Luego, se realizará el diseño de la aplicación y su implementación.
- Por último, después de la implementación del juego, se realizarán pruebas para testear el correcto funcionamiento de la aplicación.

1.4. Composición del informe

El presente trabajo se encuentra dividido en seis capítulos. A continuación se describe brevemente el contenido de cada uno de ellos.

(a) **Introducción:** Presentación del tema a tratar, hipótesis, objetivos y metodología de trabajo.
(b) **Marco Teórico:** Descripción de principales conceptos y teoría asociada al tema de investigación.
(c) **Estado del Arte:** Evaluación de las principales diferencias entre los programas de docking proteína-proteína más utilizados en la actualidad y breve descripción de sistemas interactivos similares al desarrollado en este proyecto.
(d) **Predicción de PPI a través de Interfaces Interactivas:** Presentación del problema abordado,la solución propuesta y el método utilizado para llevar a cabo dicha propuesta.
(e) **Resultados** Exposición de la aplicación interactiva obtenida a partir de la metodología utilizada, así como una breve discusión acerca de los resultados obtenidos de su desarrollo.
(f) **Conclusiones:** Principales conclusiones a partir de los resultados obtenidos, como también el trabajo a futuro para mejorar estos resultados.

Además, al final del informe se adjuntan las referencias con los artículos utilizados en el proceso de investigación.

Capítulo 2

Marco Teórico

El presente capítulo tiene por objetivo describir los principales conceptos asociados a la formación de complejos proteicos y a su predicción, facilitando al lector la comprensión de las secciones posteriores.

2.1. Proteínas

Las proteínas tienen un importante rol en los organismos vivos, prácticamente todos los procesos biológicos dependen de sus funciones.

Los bloques de construcción o componentes básicos de una proteína son los aminoácidos. Un aminoácido es una molécula orgánica que se compone de un grupo ácido (carboxilo) y uno básico (amino), ambos unidos a un carbono (Blanco y Blanco, 2017).

A partir de veinte tipos diferentes de aminoácidos se construyen todas las proteínas existentes y estos se unen entre sí a través enlaces no ramificados. A la unión entre aminoácidos se le conoce como enlace peptídico y a una molécula formada por varios aminoácidos se le llama polipéptido.

Por tanto, una proteína puede ser definida como una molécula formada por una o más cadenas de aminoácidos (polipéptidos), plegadas y enrolladas con una estructura tridimensional específica. La estructura adquirida por una proteína define en gran parte la función que desempeña. Dos proteínas con una estructura tridimensional similar se asume, generalmente, cumplirán funciones similares (Blanco y Blanco, 2017).

La estructura que adquiere una proteína no depende únicamente de la cantidad de aminoácidos que la componen. Existen diferentes niveles de organización en una proteína:

- **Estructura Primaria:** Relacionada con la secuencia lineal en que están distribuidos los aminoácidos en las cadenas (Figura 2.1.a). La secuencia primaria de aminoácidos especifica

una estructura tridimensional que se forma para cada proteína.

- **Estructura secundaria:** Define la distribución espacial de los plegamientos de la secuencia de aminoácidos primaria tales como hélices α, hojas β y bobinas o bucles, los que contribuyen a la forma general de la proteína (Figura 2.1.b). Esto sucede cuando la secuencia de aminoácidos se une a través de enlaces de hidrógenos.

- **Estructura Terciaria:** Determina la estructura tridimensional completa de la proteína y se origina cuando están presentes ciertas atracciones entre hélices α y hojas β (Figura 2.1.c).

- **Estructura Cuaternaria:** Sólo se da en proteínas formadas por más de una cadena polipetídica. Cada cadena conforma una subnidad y la unión entre ellas da origen a la estructura cuaternaria de la proteína (Figura 2.1.d).

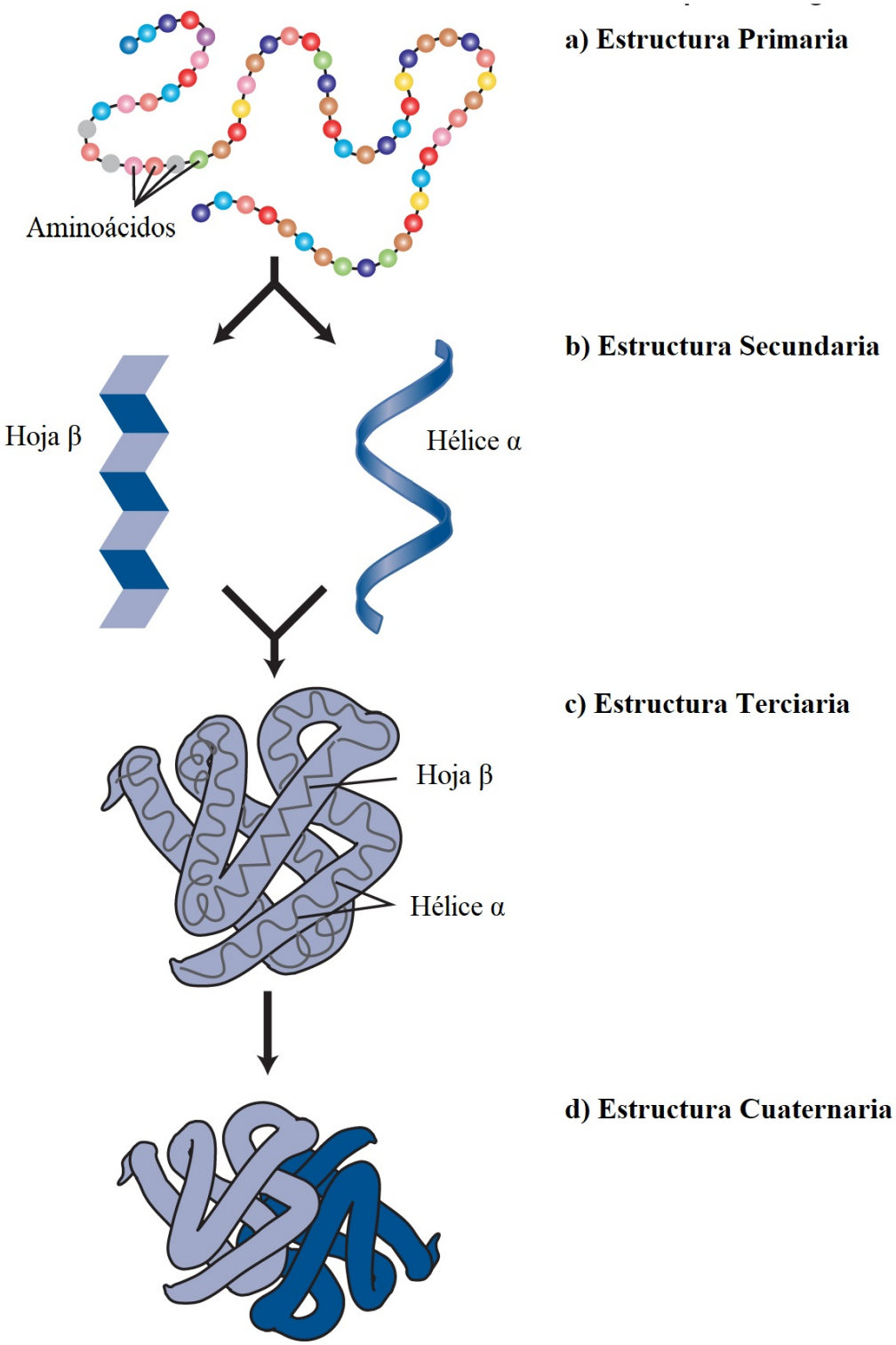


Figura 2.1: Órdenes de la estructura de una proteína. a) Estructura primaria, b) estructura secundaria, c) estructura terciaria y d) estructura cuaternaria. Créditos Imagen: National Human Genome Research Institute (NIH).

2.1.1. Interacciones proteína-proteína (PPI)

Las proteínas se unen así mismas como también junto a otras biomoléculas (moléculas encontradas en organismos vivos): ácidos núcleicos, cofactores orgánicos e inorgánicos, entre otros. Estas interacciones son la causa de muchas de las reacciones bioquímias en los sistemas biológicos (Bagchi, 2018).

Cuando la asociación se produce entre proteínas se le conoce como interacción proteína-proteína (PPI, por sus siglas en inglés) y es posible encontrar diferentes tipos entre ellas. Uno de los factores más utilizados a la hora de determinar el tipo de interacción entre proteínas corresponde a la interfase formada. A continuación se describe brevemente junto a las principales fuerzas que influyen en la formación y estabilidad de los complejos de proteínas.

(a) **Interfase PPI**

Es el área entre las dos cadenas de proteínas que forman el complejo. Si la composición de aminoácidos de las dos cadenas es la misma, se le llama interfase homomérica y si son distintas, interfase heteromérica.

La interfase PPI tiene las siguientes características:

- **Área superficial:** En el caso de proteínas heterodiméricas, la superficie generalmente es del orden de 600 Å^2 (1 Å^2 equivale a 10^{-20} m^2), mientras que para proteínas homodiméricas puede ser incluso mayor.
- **Forma de la interfase:** Se caracteriza por ser casi plana y estar separada en dos zonas; el núcleo que se encuentra enterrado en la interfase y el borde, que es accesible al solvente.
- **Composición de los aminoácidos:** Generalmente se encuentra una gran cantidad de aminoácidos aromáticos (fenilalanina, triptófano, tirosina e histidina) y arginina. Mientras que la cisteína comúnmente no es encontrada en este sitio.
- **Distribución estructural secundaria:** La interfase se encuentra compuestas de regiones con plegamiento de tipo hoja β.

(b) **Clasificación de las interfase PPI**

Existen varias formas de clasificar las PPI entre las que se encuentran aquellas basadas en la naturaleza de las proteínas involucradas, la estabilidad de los complejos PPI, la duración de vida de la interacción entre las proteínas y la naturaleza de la interfase PPI entre

las proteínas involucradas.

- **Naturaleza de las proteínas que interactúan:** Si las proteínas involucradas en la interacción tienen la misma composición de aminoácidos forman homo-oligómeros, con simetría. De lo contrario forman hetero-oligómeros.

- **Estabilidad de los complejos de proteínas que interactúan:** Cuando las proteínas que interactúan no pueden existir en estado libre y son estables sólo en asociación multimérica, son llamados oligómeros obligados (homo-obligómeros y/o hetero-obligómeros). Por otra parte, cuando las proteínas involucradas en la interacción si pueden existir en estados libres individualmente se les llama no-obligados.

- **Vida de la PPI:** Cuando la interacción es altamente estable y necesita de una influencia externa para romperla, son llamados complejos permanentes, en caso contrario, son llamados complejos transitorios.

- **Naturaleza de la interfase de interacción:** Cuando las proteínas que interactúan utilizan la misma interfase para unirse entre ambas, son llamado complejos isólogos. Mientras que en los complejos heterólogos, las proteínas individuales utilizan diferentes interfases para formar PPI sin ninguna simetría cerrada.

(c) **Fuerzas que intervienen:**

- **Interacciones electrostáticas**:
  Es un tipo de fuerza de largo alcance y se genera debido a que las moléculas se encuentran altamente cargadas, pudiendo ser atraídas o repelidas por otras. Corresponde a uno de los principales factores que influyen en la formación de complejos.

- **Enlaces hidrógeno:** Corresponden a un tipo de atracción débil entre moléculas que poseen carga eléctrica. Esta fuerza es originada por la atracción electróstica siendo capaz de modificar las propiedades químicas de una molécula.

- **Fuerzas de Van der Waals:** Es una fuerza eléctrica de atracción, débil y generalmente transitoria, de un átomo por otro. Es generada debido a la nube de electrones que posee cada átomo y que puede fluctuar. La formación de un dipolo en un átomo es capaz de inducir un dipolo complementario, de corta duración, con otro átomo a una distancia suficientemente cerca.

Para poder estudiar la interacción entre proteínas y determinar sus características, como la mencionada interfase de interacción, es necesario conocer la estructura del complejo proteico formado. La importancia de la estructura como se mencionó en secciones anteriores, es la gran

influencia sobre la actividad o función que desempeña el complejo.

Al igual que una proteína individual un complejo posee cuatro niveles de organización estructural: primaria, secundaria, terciaria y cuaternaria, e idealmente se busca realizar los estudios utilizando estructuras determinadas de manera experimental.

Sin embargo, como se indicó en el **Capítulo** 1 la rapidez y costos de este tipo de métodos no es la deseada. Para complementar esta falencia se han desarrollado métodos computacionales como una alternativa que cada día mejora en los resultados obtenidos pero que áun no se encuentra en una etapa madura.

A continuación, se describen las principales características de este tipo de programas, conocidos como docking proteína-proteína.

2.2. Docking proteína-proteína

El problema de docking proteína-proteína puede ser definido como la predicción correcta de la estructura de un complejo proteico dadas las estructuras individuales de las proteínas involucradas (Vakser, 2014). En el caso más general, no existe otra información más que la estructura de cada proteína individual, mientras que en algunos casos hay conocimiento sobre el sitio de unión, lo cual simplifica bastante el proceso.

Existen tres componentes principales que componen el proceso de docking; la representación del sistema, el proceso de búsqueda del espacio conformacional y el ranking de las soluciones candidatas.
En primer lugar, se selecciona la representación del sistema. Este paso influirá directamente en los dos componentes restantes.
Luego se realiza el proceso de búsqueda cuyo objetivo es obtener las posibles conformaciones o modos de unión candidatos de las proteínas que forman el complejo. En cambio, la tarea de la función de puntuación es discriminar de forma efectiva las conformaciones correctas de los falsos positivos obtenidos en la etapa de muestreo. Ambos procesos; búsqueda y puntuación, pueden ser realizados de forma conjunta o en etapas separadas (caso de enfoques post-docking).

En la Figura 2.3 se resumen las características mencionadas en cada componente del método de docking proteína-proteína. Mientras que en la Figura 2.2 se ejemplifica gráficamente el proceso de docking realizado a partir de las estructuras individuales de dos proteínas para obtener el complejo 1UDI (código PDB), luego de realizar la búsqueda y puntuación de las poses obtenidas.

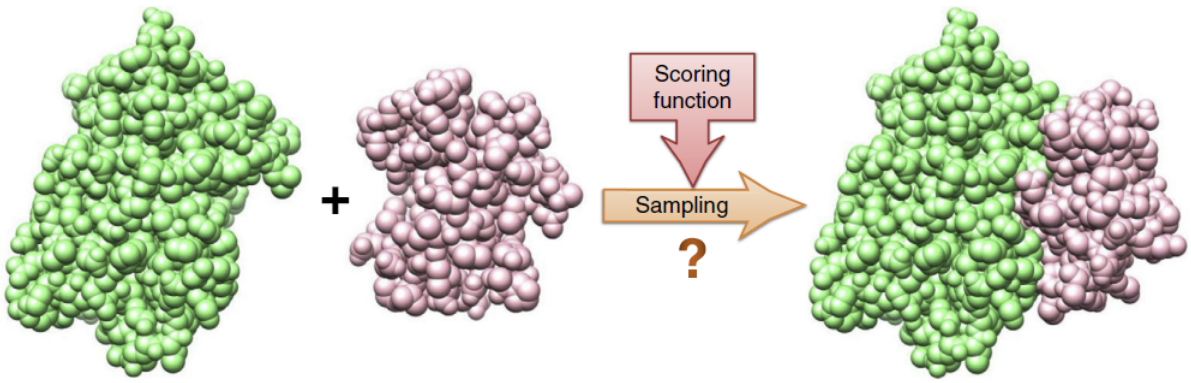


Figura 2.2: Ejemplo de un proceso de docking proteína-proteína donde el complejo de dos proteínas individuales (código PDB 1UDI) se construye a través del muestreo de conformaciones de unión y posterior evaluación y ranking. Imagen extraída desde (Huang, 2014).

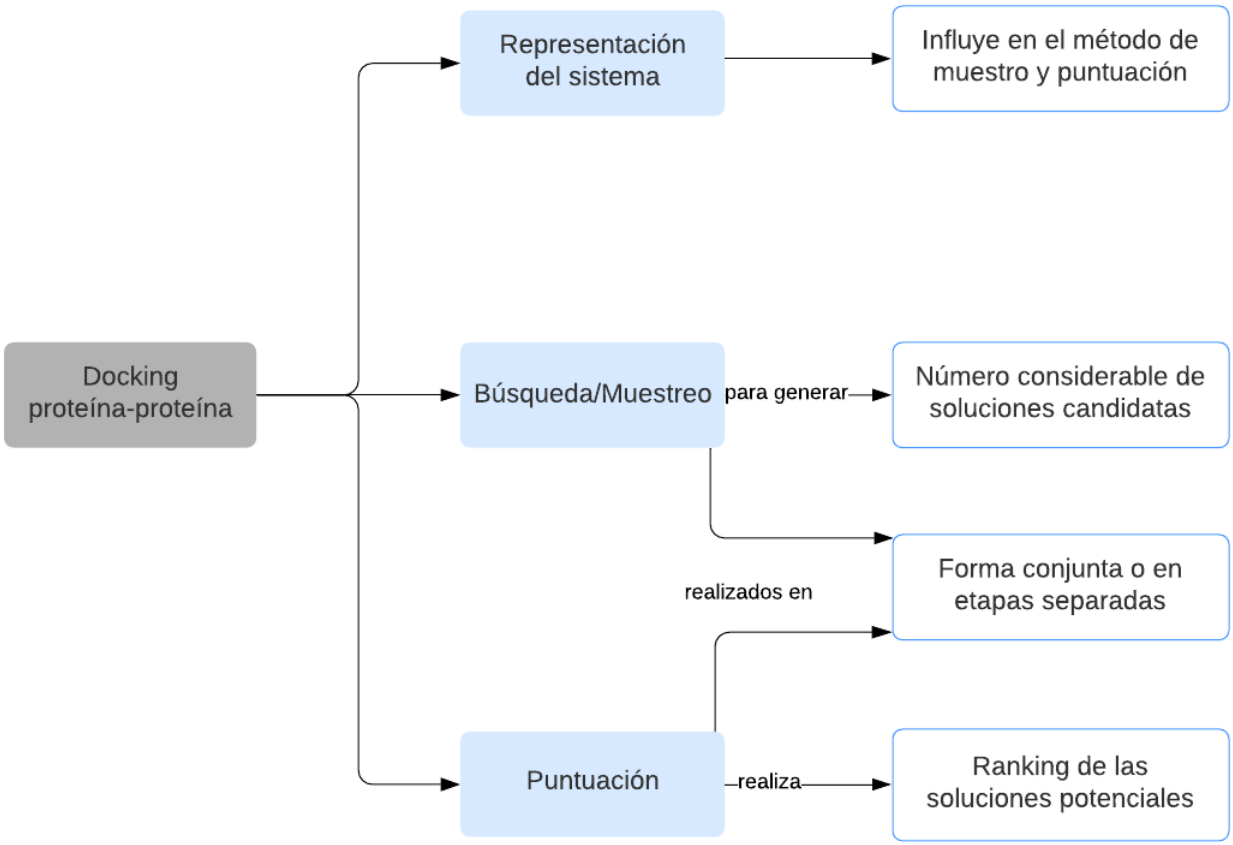


Figura 2.3: Componentes principales del proceso de docking proteína-proteína; represetación del sistema, búsqueda y puntuación.

A continuación se describen de forma más detallada las características más importantes de los procesos de búsqueda y puntuación.

2.3. Proceso de búsqueda

Las estrategias de búsqueda utilizadas actualmente pueden ser divididas en tres grandes grupos; búsqueda global exhaustiva, coincidencia local de características de forma y búsqueda aleatorizada, además de una categoría adicional de enfoques post-dockings.

2.3.1. Búsqueda global exhaustiva

Debido a la falta de información acerca de los sitios de unión (lugar en que se unen las proteínas que forman el complejo) generalmente se requiere una búsqueda global para encontrar las posibles orientaciones en las que se ensamblan las proteínas de la interacción a analizar. Esta búsqueda es realizada sobre seis grados de libertad, tres asociados a la traslación y tres a la rotación de una proteína con respecto a la otra.

De manera general en este tipo de algoritmos, una de las proteínas se deja fija (molécula estática) mientras que la otra (molécula en movimiento) es desplazada alrededor de la primera. A menudo se realiza primero la búsqueda en el espacio rotacional, girando la molécula en movimiento en un ángulo de Euler (coordenada angular tridimensional) en el espacio rotacional 3D. Luego de aplicar la rotación, se realiza una búsqueda exhaustiva en el espacio traslacional de tres dimensiones para la proteína móvil con respecto a la estática. Este proceso se repite hasta completar la búsqueda en el espacio rotacional 3D completo.

A causa de la gran cantidad de traslaciones y rotaciones en el espacio de seis dimensiones, el costo computacional continúa siendo un desafío aún cuando las proteínas son tratadas generalmente como cuerpos rígidos (no se considera cambios conformacionales en las proteínas durante la búsqueda). Por lo tanto, la eficiencia es un factor crítico a la hora de desarrollar este tipo de algoritmos.
Dentro de los algoritmos de búsqueda global exhaustiva más utilizados se encuentran:

- **Algoritmos basados en la transformada de Fourier (FFT)**

  Una de las principales características de este tipo de algoritmos es su capacidad de realizar la búsqueda sobre el espacio traslacional 3D completo de una sola vez a través del espacio imaginario, realizando varios cálculos de FFT.

  De esta manera aceleran el proceso de búsqueda en tres grados de libertad, reduciendo el costo computacional de la búsqueda de $O[N^6]$ a $O[N^3 \log(N^3)]$ mientras es capaz de cubrir cada posición de la grilla del espacio 6D.

El principio general en que se basan estos programas es el mismo, variando la forma en que son mapeados sobre la grilla los potenciales de los átomos de las proteínas y la función de puntuación y/o potenciales utilizados (Huang, 2014).

En la Figura 2.4 se muestra un ejemplo en formato 2D utilizando un esquema de puntación de complementariedad de la forma básico.

Se denomina a la proteína estática como **R**, y a la en movimiento **L**. Ambas proteínas son representadas como grillas sobre un espacio cartesiano de dimensiones $N \times N \times N$ como en observa en la parte **a)** de la imagen.

Los valores que toma cada grilla son calculados al aplicarle tanto a **R** como **L** una función que representa los términos energéticos de sus átomos.

Luego de esto, la parte **b)**, de forma simultánea son calculadas las correlaciones o puntajes de coincidencia para todas las traslaciones relativas entre dos grillas a través de dos cálculos de transformadas de Fourier.

En este ejemplo un valor mayor de $C(x, y)$ significa una mejor correlación o puntaje de coincidencia entre ambas proteínas.

Por último, en la **c)** de la Figura 2.4el complejo final es construido aplicando las traslaciones relativas a las grillas de la proteína **L**.

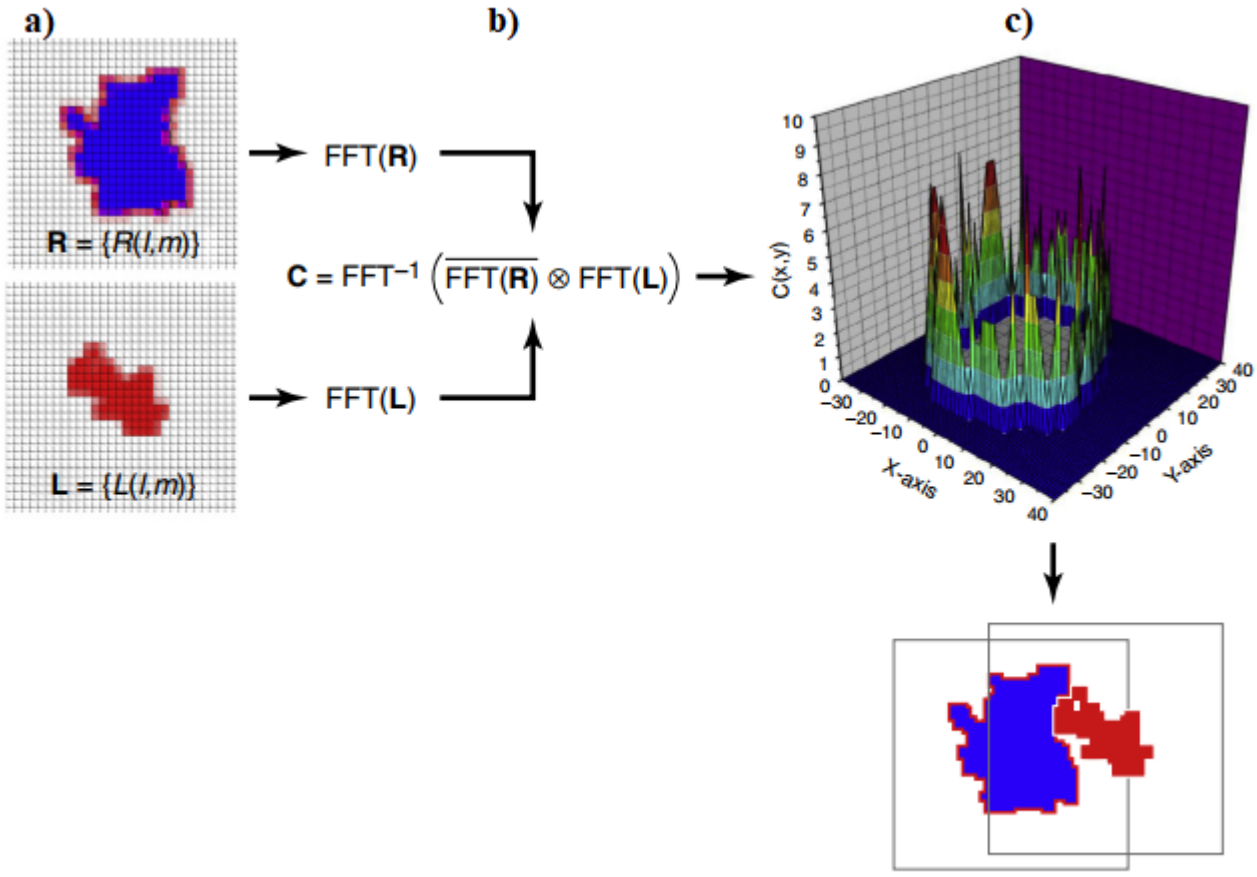


Figura 2.4: Ilustración de un algoritmo de búsqueda basado en la FFT. a) Representación de las proteínas en un espacio cartesiano 2D, b) Cálculo de puntajes de coincidencia a través del cálculo de FFT y c) Construcción de la pose del complejo proteico. Créditos Imagen: (Huang, 2014).

- **Algoritmos basados en búsqueda directa**

  Este tipo de algoritmo busca coincidencias entre ambas proteínas en un espacio cartesiano 3D grillado con ayuda de algunos métodos de aceleración entre los que destacan, la aplicación de operadores booleanos y reglas heurísticas. Tomando como ejemplo el programa BIGGER, uno de los programas que utiliza este método de búsqueda, primero se mapea la forma molecular de las dos proteínas en la grilla 3D. Luego, a cada punto de la grilla se le asigna un valor simple, como "1" si está ocupada por la proteína o "0" si no lo está. Este sistema de representación es similar al utilizado en FFT, excepto que en la búsqueda directa los valores utilizados son más simples.
  La eficiencia de este enfoque es menor a los algoritmos basados en FFT. Sin embargo, debido a que opera directamente sobre el plano cartesiano es más controlable, resultando más fácil la incorporación de flexibilidad en las proteínas e información biológica (Huang, 2014).

En la Figura 2.5 se resumen los principales programas de docking de acuerdo al algoritmo de búsqueda exhaustiva que utilizan.

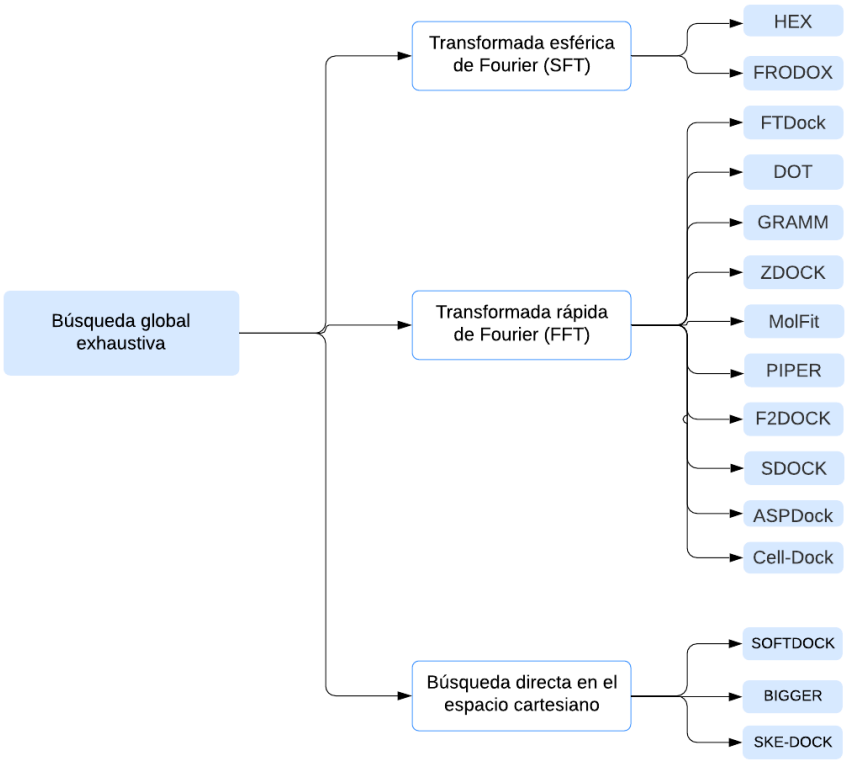


Figura 2.5: Principales tipos de búsqueda global exhaustiva y programas de docking que las utilizan (Huang, 2014).

2.3.2. Coincidencia local de características de forma

En este tipo de algoritmos las proteínas son representadas por formas moleculares. Luego, se aplican métodos para encontrar coincidencias que muestren una buena complementariedad de forma local entre ambas proteínas. Este tipo de algoritmo, caracterizado por la rapidez del proceso de búsqueda, normalmente genera una gran de cantidad (decenas de miles) de posibles orientaciones de unión en minutos, resultando práctico desde el punto de vista computacional. Una de las diferencias que presenta con los algoritmos de búsqueda global exhaustiva, es que la búsqueda en los seis grados de libertad no es presentada de forma tan explícita. En cambio, son incluidas implícimante en una matriz de transformación para una coincidencia y sólo son calculadas cuando una orientación de unión o pose es construida a través de la coincidencia (match).

- **Algoritmo de geometría de la distancia**

  En la Figura 2.6 se ilustra un algoritmo basado en este tipo de búsqueda utilizado por el programa DOCK.

  En primer lugar, se determina la superficie molecular de una de las proteínas (**R**) (generalmente la de mayor tamaño) y a continuación se generan puntos de esfera que representan la forma del sitio de unión en esta proteína (se debe tener información acerca de su ubicación).

  Luego, se construyen posibles complejos proteicos haciendo coincidir los átomos de la proteína restante (**L**) con los puntos esféricos utilizando para ello un algoritmo de geometría de la distancia.

  Una coincidencia (match) se considera exitosa cuando los bordes de todos los puntos esféricos definidos en la proteína **R** coinciden con un conjunto de átomos de la proteína **L** dado una tolerancia de distancia. A partir de una coincidencia exitosa es construida una posible orientación de unión.

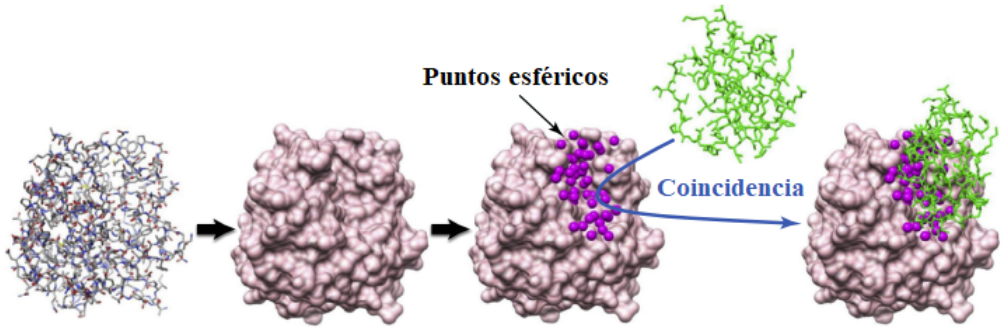


Figura 2.6: Ilustración del algoritmo de búsqueda basado en geometría de la distancia y que es utilizado por DOCK. Créditos Imagen: (Huang, 2014).

- **Hashing Geométrico**

  Este tipo de algoritmos realizan una búsqueda global mediante encontrando coincidencias (matches) locales de descriptores de forma.

  En la Figura 2.7 se ilustra un algoritmo basado en este tipo de búsqueda utilizado en el programa PatchDock.

  En primer lugar, se calculan las superficies moleculares para ambas proteínas. Luego de esto se aplica un algoritmo de segmentación para detectar tres tipos de parches geométricos (descriptores de forma) sobre la superficie molecular: piezas superficales cóncavas, convexas o planas.

  La generación de posibles complejos es llevada a cabo mediante la concidencia entre parches superficiales siguiendo la regla de Hashing Geométrico. Esta regla indica que parches convexos coinciden con parches cónvacos, mientras que los de tipo plano pueden coincidir con cualquier tipo.

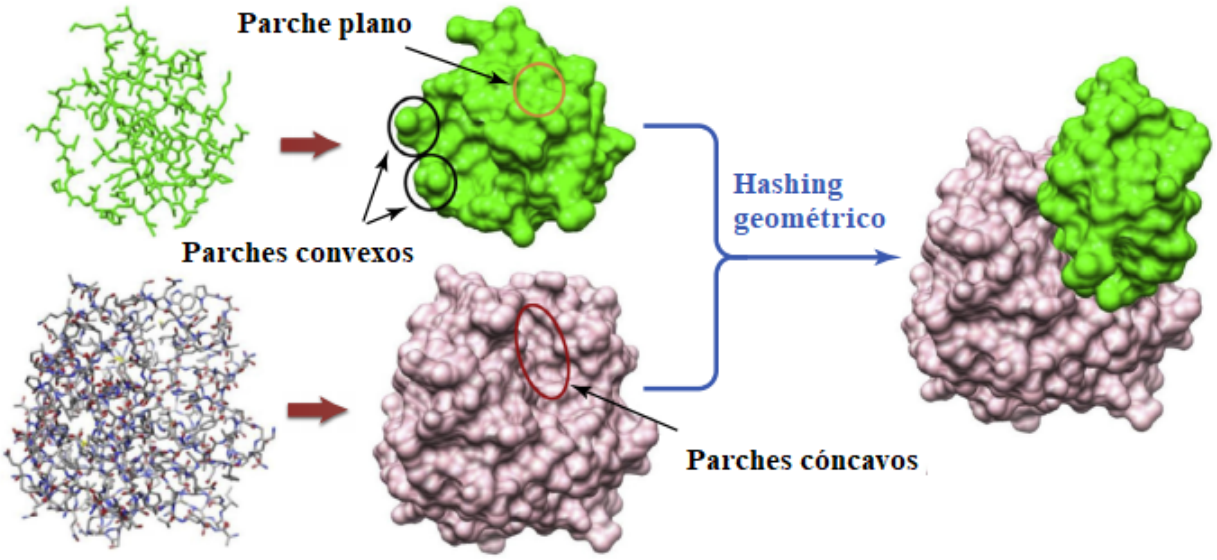


Figura 2.7: Ilustración del algoritmo de búsqueda utilizado en PatchDock basado en Hashing geométrico. Créditos Imagen: (Huang, 2014).

Dentro de las desventajas de este tipo de búsqueda es que muchas de las orientaciones de unión generadas incluyen choques atómicos. Por tanto, generalmente se efectúa un filtro para eliminar aquellas soluciones que presentan demasiados choques.

Además, tienden a generan más orientaciones de unión hacia aquellos sitios con mejor complementariedad de forma, siendo necesario a menudo, incluir un paso de post-agrupamiento para eliminar la redundancia en las soluciones finales.

Por último, en la Figura 2.8 se detallan ejemplos de programas que utilizan diferentes programas que utilizan algoritmos de búsqueda basados en coincidencia local de características de forma.

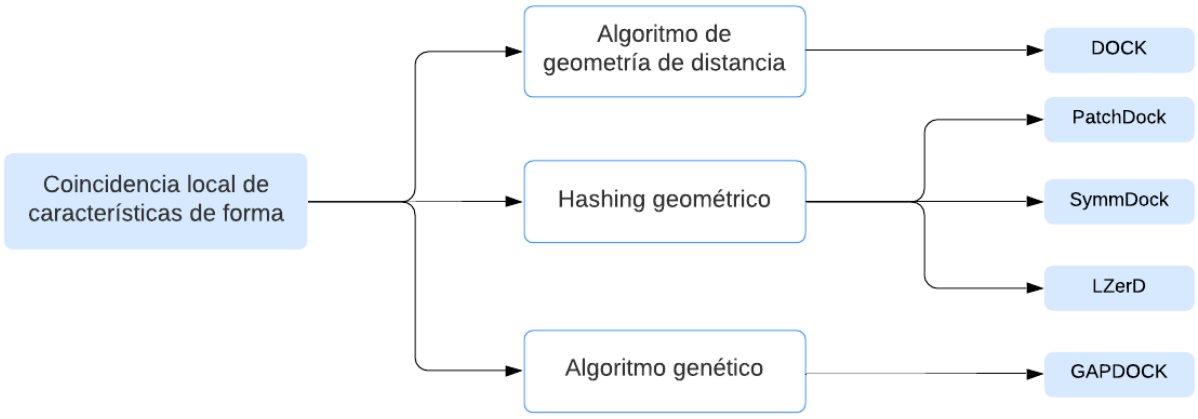


Figura 2.8: Principales tipos de algoritmos de coincidencia local de características de forma y programas de docking que los utilizan (Huang, 2014).

2.3.3. Búsqueda aleatoria

Este tipo de algoritmos es utilizado tanto para búsqueda global como para local, sin existir una representación molecular específica, a diferencia de los casos anteriores. Sin embargo, se suele realizar una búsqueda a nivel atómico, a pesar de ser posible utilizar una representación de grilla o modelos reducidos de proteínas para acelerar el proceso de búsqueda.

En la búsquea aleatoria una de las proteínas se mantiene fija, representada como átomos o grilla, dependiendo del algoritmo. Luego, la otra proteína se posiciona de forma aleatoria alrededor del sitio de unión (búsqueda local) o alrededor de toda la molécula de la proteína estática (búsqueda global) utilizando un cierto número de reglas.

Es posible usar algoritmos con el fin de optimizar el proceso de ubicación con información como la forma molecular y/o superficie molecular, generando orientaciones iniciales de unión más adecuadas.
A continuación, a partir de las posiciones iniciales, cada orientación de unión generada es optimizada y/o refinada por medio de un muestreo de etapas múltiples y/o mediante un enfoque de modelamiento multiescala utilizando algoritmos estocásticos como; algoritmos genéticos o métodos de Monte Carlo.

Este tipo de búsqueda similar a los algoritmos de coincidencia local de características de forma no realiza una búsqueda exhaustiva del espacio 6D completo. Los parámetros de cada orientación de unión correspondientes a los seis grados de libertad se obtienen a partir de su ubicación inicial y se ajustan mediante el proceso de optimización posterior.

En la Figura 2.9 se presentan algunos ejemplos de programas de docking que utilizan este tipo de algoritmos de búsqueda, entre ellos se encuentran RosettaDock, ICM-DISCO, ATTRACT, HADDOCK, SwamDock y AutoDock.

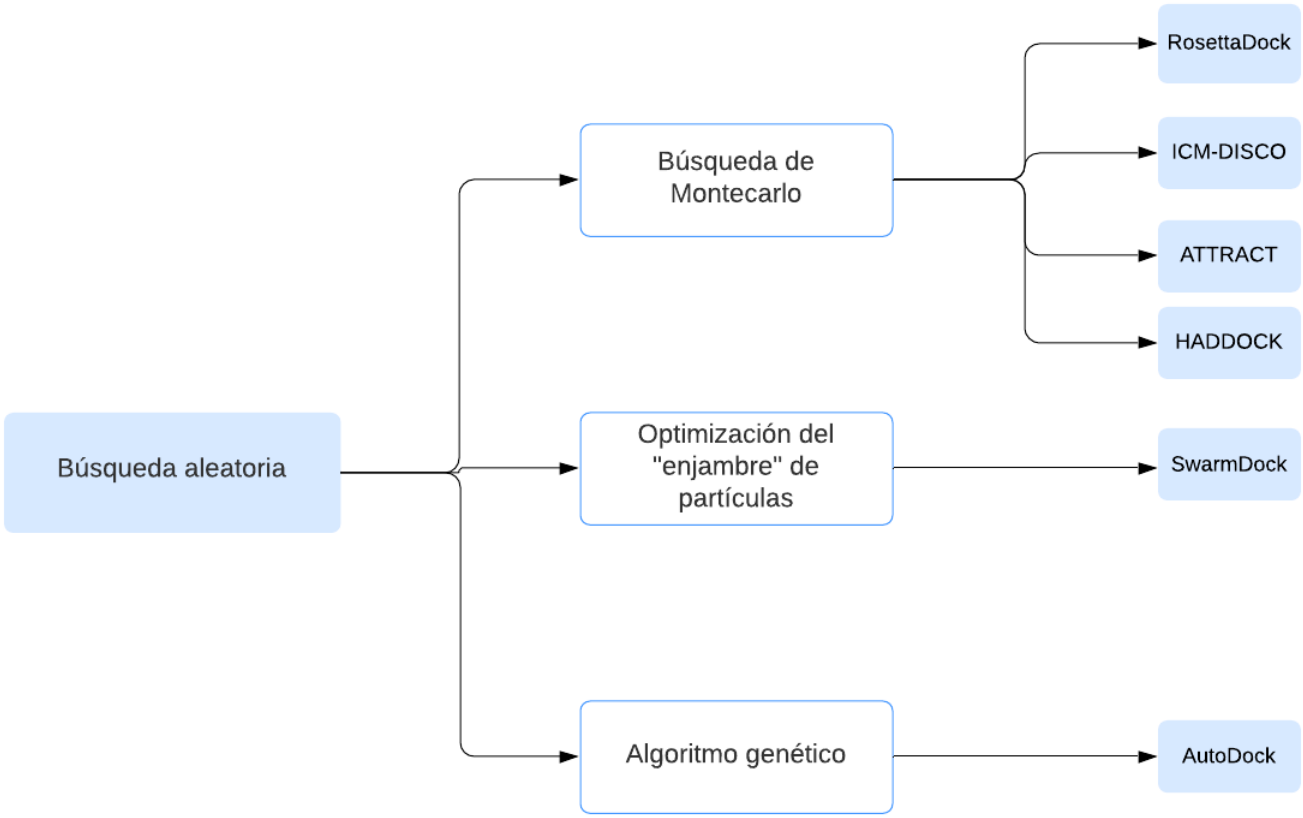


Figura 2.9: Principales tipos de algoritmos de búsqueda aleatoria y programas de docking que los utilizan.

2.3.4. Enfoques post-docking

En este caso, este tipo de aproximaciones no forman una categoría de algoritmos de búsqueda propiamente tal, sino que son un tipo de de refinamiento de docking que no necesita de un algoritmo de búsqueda propio.

Estos enfoques por lo general tienen una estructura jerárquica y presentan por lo menos dos etapas procedimentales.
En primer lugar, se realiza un muestreo de las posibles orientaciones y/o conformaciones de unión utilizando un programa de docking inicial, que puede ser cualquiera de los descritos anteriormente.

Luego, una cierta cantidad de soluciones candidatas con los mejores puntajes obtenidos en el primer paso, pudiendo variar entre cientos a decenas de miles, son optimizadas y re-clasificadas utilizando una técnica de puntuación más sofisticada, en donde puede incorporarse flexibilidad en las proteínas e información biológica.
Esta de separación entre búsqueda y puntuación simplifica de manera importante el proceso computacional.

La principal razón del uso del enfoque post-docking es que los programas iniciales de docking proteína-proteína normalmente generan al menos un modo de unión casi nativo ("hits") en un cierto número de orientaciones y/o conformaciones de unión.

Dado el éxito razonable de los actuales programas de docking proteína-proteína en la generación de hits dentro de las soluciones candidatas con las mejores puntuaciones, los algoritmos de post-docking han experimentado un progreso significativo.

Por último, en la Figura 2.10 se muestran los principales programas de docking que utilizan este tipo de enfoque. Entre ellos se encuentran: RPScore, ZRANK, PyDock, Empire, DARS, DECK, entre otros.

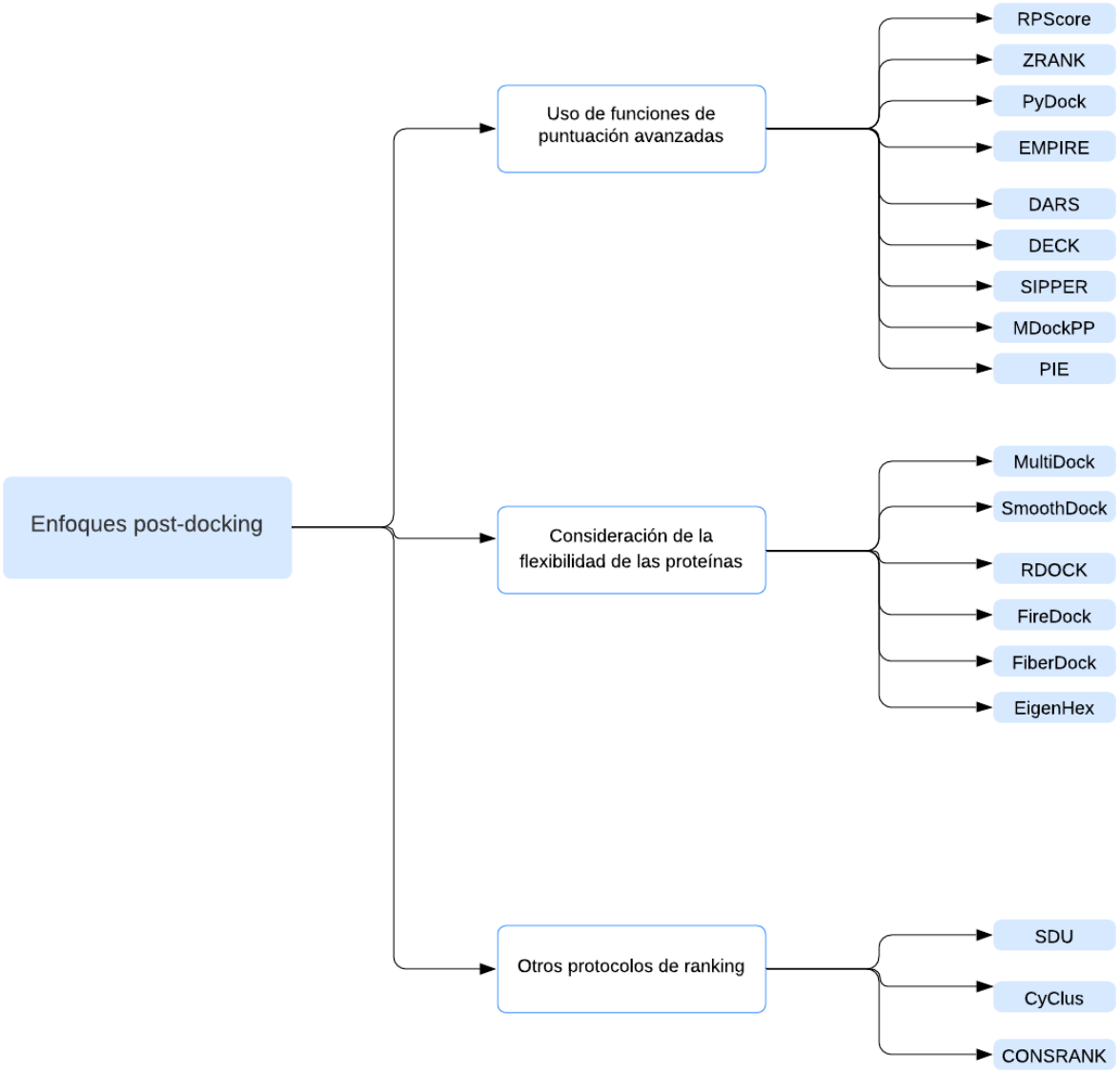


Figura 2.10: Principales tipos de enfoques post-docking y programas que los utilizan (Huang, 2014).

A modo de resumen, en la Tabla 2.1 es posible observar un cuadro comparativo sobre las principales características de los algoritmos más utilizados en los programas de docking proteína-proteína actuales de acuerdo a la obtenido por (Huang, 2014).
Debido a que los métodos basados en la correlación FFT y SFT brindan un buen equilibrio entre eficiencia computacional y búsqueda global exhaustiva la mayor parte de los programas existentes están basados en este tipo de algoritmos.

Algoritmo	**Búsqueda exhaustiva**	**Búsqueda global**	**Búsqueda local**	**Docking rígido**	**Docking flexible**	**Representación molecular**	**Costo computacional**
Basado en la correlación de FFT	X	X		X		Basada en grilla	Bajo
Búsqueda basada en SFT	X	X		X		Superficie armónica	Bajo
Búsqueda directa	X	X	X	X		Basada en grilla	Medio-Alto
Coincidencia local de forma		X	X	X		Grilla o superficie	Medio
Búsqueda aleatoria		X	X	X	X	Basada en átomos	Alto

Tabla 2.1: Comparación de los principales algoritmos de búsqueda utilizados actualmente. Modificado a partir de (Huang, 2014).

2.4. Funciones de puntuación

El objetivo principal de aplicar funciones de puntuación en los programas de docking es identificar las conformaciones de unión correctas entre las posibles orientaciones generadas en el proceso de búsqueda.
Para esto existen dos características importantes que deben cumplir estas funciones; ser lo suficientemente rápidas para ser aplicadas a un gran número de soluciones candidatas y ser capaces de discriminar de forma efectiva soluciones nativas correctas, especialmente cuando no existe información acerca del sitio de unión.

Los algoritmos de puntuación existentes se dividen en tres grupos principales; funciones basadas en campos de fuerza, funciones empíricas, funciones basadas en conocimiento y funciones de consenso (Figura 2.11).

A continuación se presenta un breve descripción de cada categoría:

Funciones de campos de fuerza: Este tipo de algoritmos se basa principalmente en las interacciones físicas átomicas donde podemos destacar las interacciones de van Der Waals, interacciones electrostáticas y fuerzas de estiramiento y torsión de la unión. Los parámetros son obtenidos generalmente de datos experimentales y cálculos mecánicos cuánticos a partir de principios físicos.

Funciones empíricas: Estiman la afinidad de unión entre las proteínas calculando la sumato-

ria de un conjunto de términos energéticos ponderados. Dentro de los parámetros utilizados pueden encontrarse energía de van Der Waals, electrostática, puentes hidrógeno, desolvatación, entropía, entre otros. Debido a que utilizan términos de energía más simples son más rápidas a la hora de realizar los cálculos, comparadas con las funciones basadas en campos de fuerza.

Funciones basadas en el conocimiento/potencial estadístico: Calculan potenciales energéticos a partir de información estructural átomica obtenida de manera experimental. Los potenciales por pares son obtenidos a partir de la frecuencia de ocurrencia de pares de átomos en una base de datos utilizando la relación de Boltzman inversa. Comparado con las categorías anteriores, provee un buen equilibrio entre precisión y rapidez.

Funciones de consenso: Se basan en la combinación de múltiples funciones de puntuación para así aprovechar las ventajas de cada una y equilibrar sus falencias, mejorando de esta forma la probabilidad de encontrar soluciones correctas.

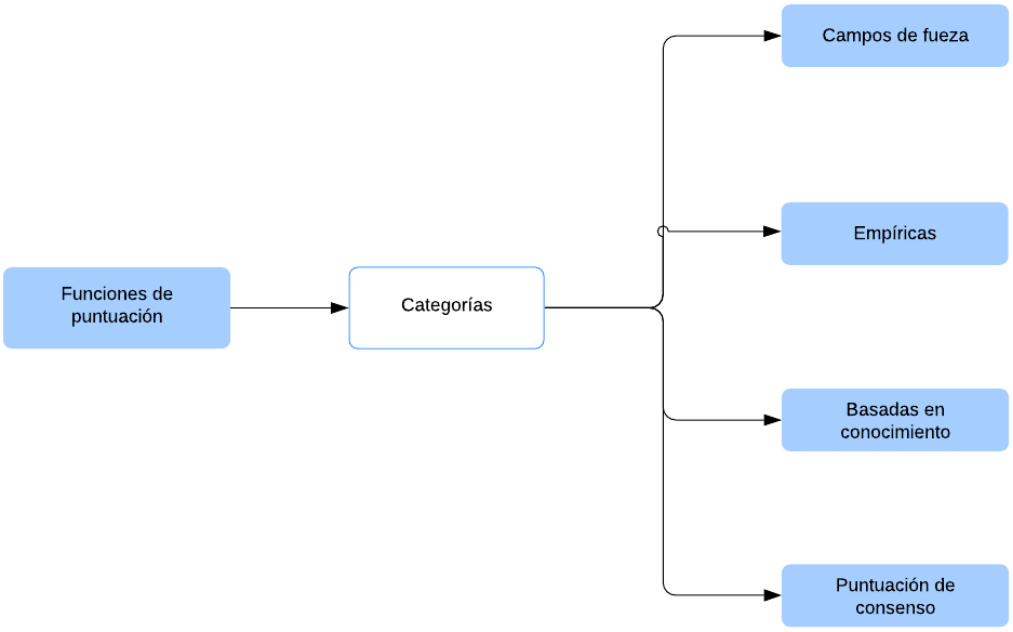


Figura 2.11: Principales categorías de funciones de puntuación para docking proteína-proteína. Modificado a partir de (Huang et al., 2010).

Capítulo 3

Estado del Arte

En esta sección se describe el estado actual de las investigaciones relacionadas con los programas de docking proteína-proteína. En primer lugar, se compara, en relación a diferentes características, el comportamiento de los principales programas de docking proteína-proteína existentes en la actualidad. Luego, se describen brevemente aplicaciones interactivas de docking molecular, similares a la implementada en este trabajo.

3.1. Evaluación de los programas globales de docking

Dentro de las diferentes estrategias de búsquedas descritas anteriormente, como se mencionó, el método de FFT es capaz de alcanzar un buen balance entre eficiencia computacional y búsqueda exhaustiva global. Es por esta razón, que gran parte de los programas de docking proteína-proteína utilizados actualmente se basan en este método de búsqueda. Como ejemplo podemos nombrar; FTDock, GRAMM, MolFit, DOT, ZDOCK1.3, ZDOCK2.1, ZDOCK2.3.2, ZDOCK3.0.2 y PIPER, entre otros .

Por otro lado, las funciones de puntuación han experimentado una evolución más significativa en comparación a las algoritmos de búsqueda.
Entre los programas de docking proteína-proteína más recientes se observan dos tendencias en cuanto a la puntuación (Huang, 2015).
La primera, corresponde a utilizar una algoritmo de puntuación más sofisticado formado por varios parámetros de energía, además de incluir términos enérgeticos adicionales como potenciales basados en conocimiento o modificando algoritmos de puntuación existentes.
La segunda, es hacer uso de paquetes de software como la librería 3D convolution o de las nuevas tecnologías de hardware, como la unidad de procesamiento gráfico (GPU), para acelerar el proceso de búsqueda.

En la Tabla 3.1 se detallan diferentes programas de docking indicando su versión, algoritmo de búsqueda y función de puntuación utilizadas.
Tal como se mencionó, la mayor parte de los algoritmos analizados utiliza un algoritmo de búsqueda exhaustivo basado en la transformada rápida de Fourier. Mientras que en la elección

de la función de puntuación hay mayor divergencia en los tipos utilizados.

Programa	**Versión**	**Algoritmo de búsqueda**	**Función de puntuación**
ATTRACT	-	Búsqueda aletatorizada	Potencial efectivo tipo LJ y electrostática
PatchDock	β 1.3	Coincidencia local de forma	Complementariedad geométrica de forma
FTDock	-	Basado en la correlación FFT	Complementariedad de la forma e interacciones electrostáticas
GRAMM	1.03	Basado en la correlación FFT	Concidencia de forma e hidrofóbica
MolFit	2	Basado en la correlación FFT	Complementariedad geométrica, complementariedad hidrofóbica e interacciones electrostáticas
DOT	2.01	Basado en la correlación FFT	Energías de van der Waals y electrostáticas
ZDOCK 1.3	-	Basado en la correlación FFT	Complementariedad de la forma, desolvatación y electrostática
ZDOCK 2.1	-	Basado en la correlación FFT	Complementariedad de la forma por pares
ZDOCK 2.3.2	-	Basado en la correlación FFT	Complementariedad de la forma por pares, desolvatación y electrostática
ZDOCK 3.0.2	-	Basado en la correlación FFT	Complementariedad de la forma, electrostática y potenciales de pares basado en el conocimiento
PIPER	-	Basado en la correlación FFT	Complementariedad de la forma, interacciones electrostáticas y potenciales de pares basados en el conocimiento
SDOCK	1p0	Basado en la correlación FFT	Potencial de van der Waals, colisión geométrica, potencial electrostático y energía de desolvatación
HEX	6.3	Basado en la correlación SFT	Complementariedad superficial y electrostática
FRODOCK	1.04	Basado en la correlación SFT	van der Waals, electrostática y desolvatación del conocimiento basado en potenciales.

Tabla 3.1: Programas de docking analizados en la revisión de (Huang, 2015).

A continuación se describen las principales conclusiones obtenidas por la revisión realizada en (Huang, 2015), donde se evaluó el desempeño de 14 programas de docking proteína-proteína global.

3.1.1. Desempeño general

En general ZDOCK3.0.2 presenta el mejor desempeño con tasas de éxito del 11,9 %, 30,7 % y 52,3 % cuando se consideraron las primeras 1, 10 y 100 predicciones, seguido por SDOCK, PIPER y FRODOCK. Por otro lado, el programa que mostró el peor desempeño fue FTDock (Tabla 3.2). Todos los programas de docking serían capaces de obtener predicciones correctas para $\sim$ 50 % o más de los objetivos (targets) en aplicaciones realistas como CAPRI si existiera información acerca del sitio de unión o estuviera disponible un enfoque post-docking ideal para procesar algunas miles de predicciones.
Una conclusión importante obtenida a partir de los resultados de la revisión es el que las tasas de éxito de todos los programas bordean o superan el 50 % cuando se considera un rango mayor de predicciones (2000 en este caso).
Esto podría indicar que las diferencias entre las tasas de éxito cuando se consideran menos predicciones esté relacionado principalmente a la función de puntuación más que al algoritmo de muestreo.

Método	Top 1	Top 10	Top 100	Top 1000	Top 2000
ZDOCK3.0.2	11.93	30.68	52.27	78.98	84.09
SDOCK	10.23	22.73	46.02	65.34	73.30
PIPER	8.52	21.02	39.77	60.23	65.34
FRODOCK	5.11	19.32	46.02	75.57	81.82
ATTRACT:LJ	5.11	18.75	47.16	76.70	79.55
ATTRACT	5.11	18.18	42.61	75.57	77.84
ZDOCK1.3	6.82	15.34	41.48	65.91	73.86
ZDOCK2.3.2	6.25	14.21	38.07	67.05	76.70
HEX	3.98	10.79	25.00	39.20	46.59
DOT	1.71	9.66	26.70	48.86	57.95
PatchDock[a]	3.45	7.47	22.99	56.32	63.79
MolFit/GH[a]	1.71	7.39	25.00	53.41	63.07
ZDOCK2.1[a]	1.14	7.39	20.45	52.27	65.91
HEX/G[a]	0.00	3.98	16.48	41.48	48.30
MolFit/G[a]	1.14	2.84	18.75	46.02	52.84
GRAMM[a]	0.00	2.84	10.79	30.68	46.59
FTDock/G[a]	0.57	1.71	11.36	42.61	56.25
FTDock	0.57	1.71	10.79	39.77	56.25

Tabla 3.2: [a] *E*stos métodos no incluyen electrostática en sus funciones de puntuación para docking. Las tasas de éxito (%) sobre 176 objetivos cuando se consideraron las primeras 1 (en azul), 10, 100, 1000 y 2000 predicciones (Huang, 2015).

3.1.2. Efectos de las funciones de puntuación

A partir de la evaluación realizada por (Huang, 2015) se determinó que aquellos métodos de docking/puntuación que incluyen electrostática, como ZDOCK3.0.2, SDOCK, PIPER y FRODOCK, exhiben un mejor desempeño que aquellos que no la consideran, como PatchDock, Molfit y ZDOCK2.1. Esto confirma la importancia de la electrostática en las interacciones entre proteínas. Un análisis más detallado de los algoritmos de puntuación, incluyendo también a la electrostática, demuestra que los programas que consideran los efectos de la desolvatación (ZDOCK3.0.2, SDOCK, y FRODOCK) en general tienen un mejor desempeño que aquellos que no la consideran, como FTDock, HEX y DOT, mostrando también la importancia de la desolvatación en las interacciones proteína-proteína.

Por último, dos de los tres programas que obtuvieron los mejores resultados (ZDOCK3.0.2 y PIPER) consideran potenciales por pares basados en conocimiento, lo que indicaría la eficacia de estos.

3.1.3. Impactos de cambios conformacionales

Para analizar los efectos de los cambios conformacionales se calcularon las tasas de éxito para los programas en estudio, utilizando tres categorías: 123 casos de cuerpo rígido, 29 de dificultad media y 24 díficiles. Tal como se esperaba, en términos generales, todos los programas obtuvieron los mejores resultados en los casos de cuerpo rígido y los peores en aquellos casos difíciles (Figura 3.1).

Los resultados obtenidos por (Huang, 2015) sugieren que un algoritmo de docking de cuerpo rígido bien optimizado tiene un gran potencial para considerar pequeños cambios conformacio-

nales en cuerpo rígido y en algunos casos de mediana dificultad. Sin embargo, la consideración de grandes cambios conformacionales, como en los casos difíciles, representan un gran desafío para todos los programas de docking.

3.1.4. Desempeño dependiendo del objetivo

El análisis del desempeño sobre objetivos individuales resulta más útil cuando se trata de determinar uno o unos pocos casos de complejos proteicos. En este sentido, se encontró que el desempeño del docking tiene una dependencia significativa con respecto al objetivo y ningún método funciona bien en todos los casos. Es decir, los programas con mejor rendimiento general no funcionan necesariamente mejor que los de peor rendimiento cuando se trata de un objetivo determinado. Por lo tanto, puede que sea necesario considerar varios programas de docking adicionales además de alguno de los que muestran un mejor rendimiento general, para así, obtener predicciones alternativas debido a la dependencia que presenta el desempeño de los programas de docking existentes con respecto al objetivo.

Por otra parte, se determinó que aparte de la flexibilidad existen posiblemente otros factores que determinan la dificultad de un objetivo en términos de docking. Al analizar las áreas de superficies accesibles (ASA) de los objetivos, se encontró que generalmente aquellos casos difíciles tienen un pequeño cambio relativo de las áreas áreas accesibles (rΔASA) tras la unión. Por tanto, el rΔASA puede ser un factor común limitante del desempeño del docking de un objetivo. La variación relativa de las áreas superficiales accesibles tras la unión está definida como:

$$r\Delta ASA = \frac{\Delta ASA}{ASA_R + ASA_L} \times 100\,\% \tag{3.1}$$

Donde,

ΔASA es el cambio de áreas superficiales accesibles (ASAs) para la proteína receptora y ligando después de la unión y,
ASA_R y ASA_L son las áreas superficiales accesibles de cada proteína (de forma individual) antes de la unión.

3.1.5. Eficiencia computacional

Otra característica importante de los programas de docking es qué tan rápido es capaz de terminar de realizar un acoplamiento para un par de estructuras proteicas. Este factor resulta muy importante cuando los recursos computacionales son limitados y es necesario realizar docking para un gran número de proteínas. En la Figura 3.2 se muestra el tiempo promedio de ejecución de los 18 métodos de docking/puntuación para el docking de un par de estructuras proteínas sobre el benchmark de 176 objtetivos. El programa con mayor eficiencia computacional resultó ser HEX con un tiempo promedio de 2.3 minutos para una ejecución de docking usando el método de puntuación geométrico y 3.0 minutos utilizando la función de puntuación por defecto (geométrica + electrostática). Luego, le sigue ZDOCK2.3.2 con 5.3 minutos, ZDOCK3.0.2 con 10.0 minutos y ZDOCK2.1 con 14.8 minutos.

La gran eficiencia computacional de HEX y ZDOCK puede atribuirse al hecho de que HEX utiliza correlaciones polares esféricas de Fourier (SPF) para acelerar los cálculos y que ZDOCK implementó recientemente la librería avanzada 3D convolution para acelarar los cálculos en sus nuevas versiones de 2.3.2 y 3.0.2.

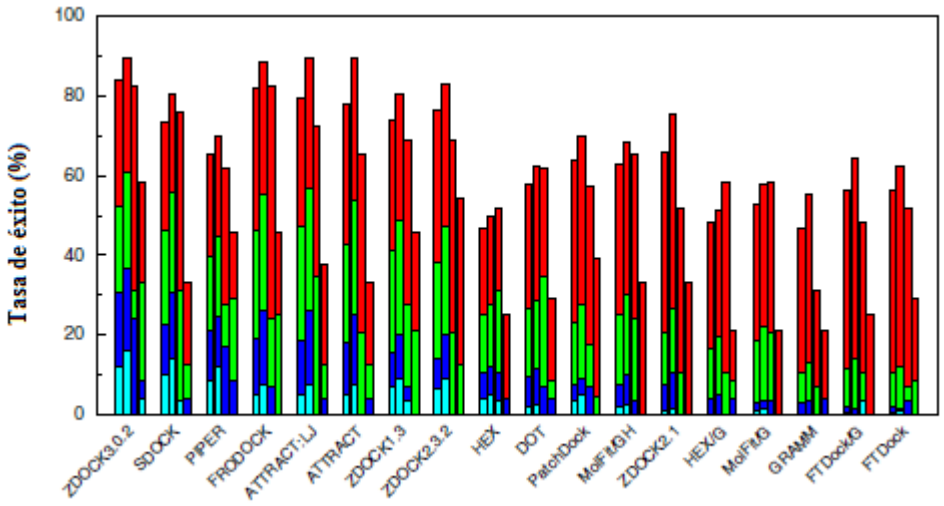


Figura 3.1: Tasas de éxito para 176 objetivos con las primeras 1 (cian), 10 (azul), 100 (verde) y 2000 (rojo) predicciones. Modificado desde (Huang, 2015)

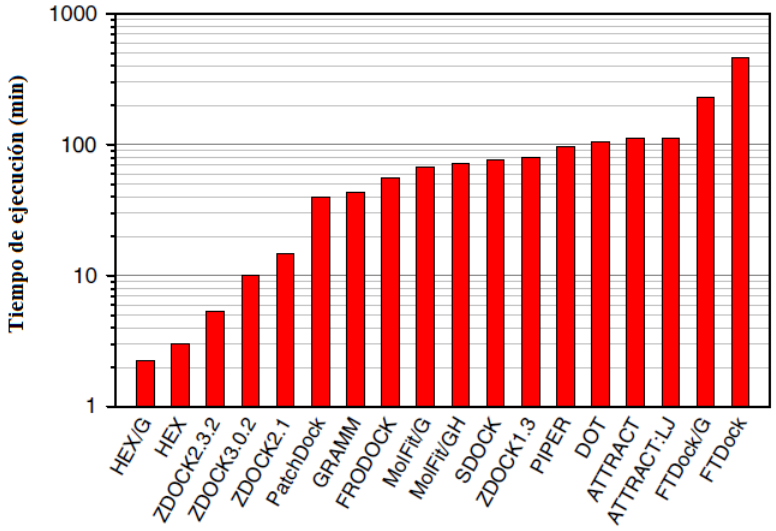


Figura 3.2: Tiempo promedio de ejecución para una predicción de docking proteína-proteína sobre los 176 objetivos del benchark 4.0. Extraído de (Huang, 2015).

3.2. Aplicaciones Interactivas de Docking Molecular

En esta sección se describen algunos de los principales programas interactivos que existen para calcular docking molecular y, que servirán como marco para abordar la problemática a tratar.

- UDock: Desarrollado por National des Arts et Metiers, Francia, por el Centre dÉtudes et de Recherche en Informatique et Communications (CEDRIC) y el Laboratoire de Genomique, Bioinformatique, et Chimie Moleculaire (GBCM). Es un sistema interactivo de docking de proteínas, tanto para entendidos en el tema como para principiantes, que permite guardar el acoplamiento generado en un archivo PDB. A través de representaciones simplificadas de las moléculas, los jugadores tienen la posibilidad de explorar el espacio conformacional con un sistema de puntuación sobre la marcha (Levieux et al., 2014).

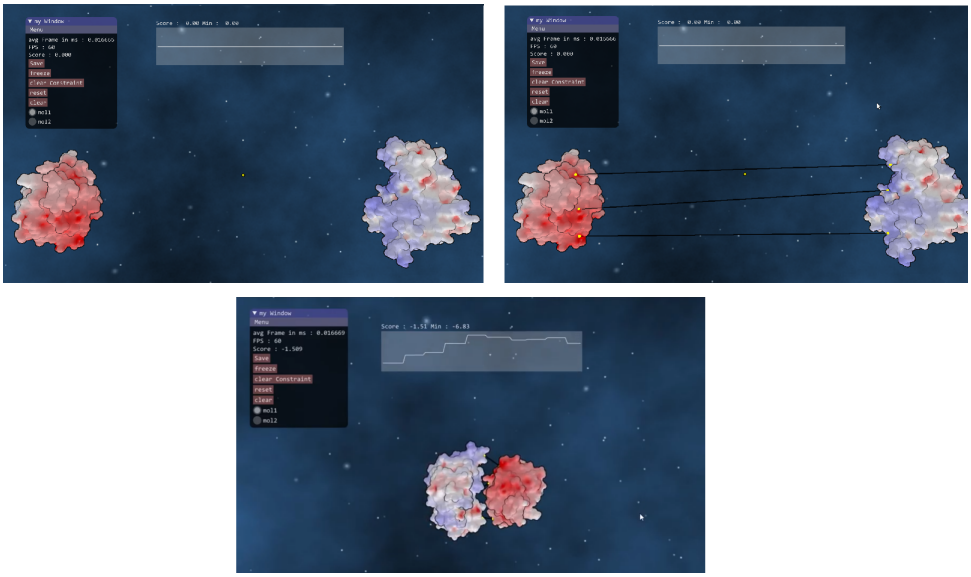


Figura 3.3: Ejemplo de un acoplamiento utilizado UDock.

- BioBlox : Lanzado el año 2017, nace de la colaboración entre investigadores del Imperial College London y Goldsmiths, University of London. Es una interfaz de usuario interactiva e intuitiva que ha sido desarrollada para explorar el docking proteico de estructuras encontradas en la base de datos PDB. La principal característica es que posee diferentes versiones para realizar el acoplamiento utilizando visualizaciones; 3D, 2D y 1D, algunas aún en proceso de desarrollo. Está basada en el programa de docking ATTRACT y su función de puntuación de campos de fuerza utiliza una representación proteica de grano grueso, incluyendo potencial de Lennard-Jones (LJ) y un término coulómbico para las interaccines electrostáticas.

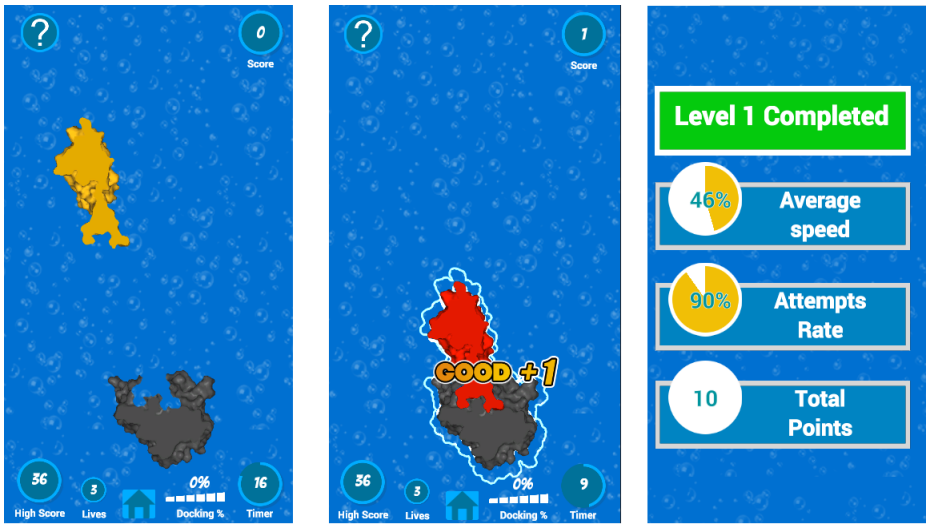


Figura 3.4: Ejemplo de una partida de docking en Bioblox 2½D.

- Juego desarrollado por (Vega Hidalgo et al., 2018) : En esta memoria de título se utilizó y adaptó un método de docking molecular para generar un videjuego de múltiples participantes. Para representar a las proteínas involucradas se transformaron las estructuras 3D, obtenidas de la base de datos PDB, a estructuras bidimensionales, utilizando para ello el programa LigPlot+. Además, se calcula la estabilidad del complejo generado por un jugador a través de una función de puntuación basada en la ley de Coulomb. El acoplamiento de mayor puntaje generado es almacenado en un formato compatible con LigPlot+. De esta forma se espera disminuir los tiempos de simulación gracias a las aproximaciones iniciales generadas por los participantes.

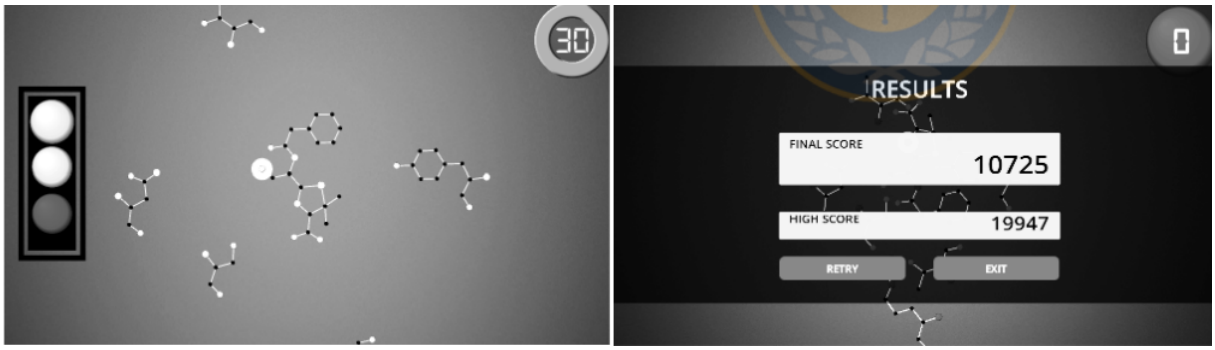


Figura 3.5: Capturas del videojuego de docking molecular desarrollado por (Vega Hidalgo et al., 2018).

A partir de las aplicaciones interactivas revisadas en esta sección se determinó que aquella que resulta más atractiva y útil para el objetivo de este proyecto es el enfoque utilizado en UDock.

En este programa, las proteínas son representadas como superfices moleculares, lo que resulta más atractivo que otros tipos de representaciones.

Además, otro punto importante es la posibilidad que tiene el jugador de revisar cuál sería el puntaje si finaliza su intento con la pose actual. Esto permite orientar al usuario a la hora de realizar el acoplamiento, lo que repercute en obtener mejores resultados y mejorar la experiencia de los jugadores, ya que sin una orientación es posible que se sientan frustrados al no obtener buenos resultados y con esto no vuelvan a utilizar la aplicación.

Por lo tanto, se rescatan estos aspectos para ser tomados en cuenta a la hora de desarrollar la aplicación de este trabajo.

Capítulo 4

Predicción de Interacciones Proteína-Proteína a través de interfaces interactivas

En el presente capítulo se explica el desafío abordado y la solución propuesta en este trabajo. Luego, se describen los algoritmos y programas utilizados como base para obtener la puntuación y ranking de las poses de docking proteína-proteína obtenidas por los usuarios de la aplicación. Por último, se detalla el desarrollo del sistema propuesto.

4.1. Problema a abordar

La rapidez a la cual se determinan de forma experimental nuevas estructuras de complejos proteicos es mucho menor a la requerida para realizar estudios acerca de la función de las proteínas y los procesos de la vida. Por tanto, para complementar y agilizar esto, dentro del campo de investigación de la bionformática, se han desarrollado diversos programas computacionales de docking que predicen la posible estructura de un complejo a partir de las estructuras individuales de las moléculas involucradas. Sin embargo, aún no alcanzan la madurez suficiente y es necesario por tanto, experimentar nuevos enfoques para mejorar la predicción de los modelos computacionales.

4.2. Solución propuesta

Dentro de los enfoques computacionales incipientes, en el marco de docking molecular, se encuentra la ludificación del proceso de acoplamiento a través de sistemas interactivos que permiten la intervención humana en el proceso de búsqueda de conformaciones de unión candidatas. Tomando como referencia los principales sistemas desarrollados con este enfoque, descritos en el Capítulo 3.2, se propone desarrollar una aplicación similar específicamente para docking proteína-proteína. En dicha plataforma se muestran al usuario las proteínas individuales involucradas en

una interacción determinada. El jugador tiene la posibilidad de rotar ambas proteínas para ensamblarlas, con el objetivo de alcanzar el mayor puntaje (de acuerdo al método de puntuación elegido para ello). De esta forma, a partir de la interacción de jugadores, se espera generar un set de datos con los dockings de mayor puntaje para el par de proteínas, con el fin de aumentar los conjuntos existentes de estructuras de complejos de proteína.

Entre los aspectos que se considera es necesario dar mayor énfasis a la hora de implementar este sistema destacan principalmente dos:

- Representación de las estructuras de las proteínas involucradas: para este proyecto se decidió utilizar una representación tridimensional de las proteínas. Esto, pensando en la experiencia de los usuarios del sistema, ya que resulta más dinámica e intuitiva una visualización 3D frente a una 2D.
- Función de puntuación: este es un factor determinante para el éxito del sistema, es necesario utilizar un algoritmo de puntuación que discrimine efectivamente los mejores modos de unión generados por los usuarios de la plataforma.

Por último, se espera obtener como resultado de la implementación y posterior uso del juego, un aumento del conjunto de datos de estructuras de complejos proteicos disponibles, para su posterior uso en estudios acerca de interacciones proteicas y evaluación de algoritmos de docking proteína-proteína.

4.3. Algoritmo utilizado: Método Formas de Contexto

En esta sección, se describe el algoritmo empleado para calcular el puntaje y viabilidad de las poses generadas por los jugadores, tras realizar un acoplamiento entre proteínas en la aplicación desarrollada. El procedimiento empleado se basa en el **método de formas de contexto** (Shentu et al., 2008) y, la principal razón de su elección es su eficiencia a la hora de evaluar la complementariedad de la forma en docking proteína-proteína.

Las características de forma locales de cada proteína se representan a través de **formas de contexto** (CS, por sus siglás en inglés), compuestas por datos booleanos.

Por otra parte, las cantidades energéticas son derivadas del cálculo de la complementariedad de la forma y el **área de superficie enterrada** (BSA), utilizando operaciones booleanas (razón de la eficiencia del algoritmo).

El pseudocódigo del algoritmo puede observarse en Algoritmo 1 y consta de tres pasos principales:

1) Muestreo superficial y representación local de la forma a través de formas de contexto
2) Evaluación de la viabilidad de la pose a través del cálculo del volumen superpuesto y, coincidencia de formas complementarias de los pares de formas de contexto
3) Ranking de las poses basado en los puntajes obtenidos

Algoritmo 1: Algoritmo Formas de Contexto

```
CS_R ← P_R, Formas de contexto de la proteína receptora;
CS_L ← P_L, Formas de contexto de la proteína ligando;
ParesCanditatos ← ∅;
foreach Forma de Contexto CS_R en CS_R do
    foreach Forma de Contexto CS_L en CS_L do
        foreach Pose π de las formas de contexto CS_R y CS_L do
            Calcular la superposición de volumen OV, bajo la pose dada π;
            if OV excede el valor límite then
                Descartar la pose π;
            end
            Calcular el área superficial enterrada BSA, bajo la pose dada π;
        end
        Sólo guardar la mejor pose π con el BSA más grande;
        Insertar la tupla (CS_R,CS_L,π,BSA) en ParesCanditatos;
    end
end
Ordenar los ParesCanditatos basado en BSA (orden decreciente);
```

A continuación se describen en mayor detalle los procedimientos involucrados en este método.

4.3.1. Representación local de la forma

La forma de una proteína se define a través de la superficie excluida al solvente (SES, por sus siglas en inglés). Ésta, puede entenderse como el límite del volumen molecular excluido al solvente y, generalmente se calcula haciendo rodar sobre la superficie de contacto expuesta de cada átomo, una sonda esférica del tamaño del solvente molecular (Figura 4.1). Se encuentra compuesta por tres tipos de curvaturas:

(a) **Cara de contacto:** Superficie átomica accesible al solvente
(b) **Cara toroidal:** Superficie en forma de silla en donde la sonda hace contacto con dos átomos
(c) **Cara re-entrante:** Superficie cóncava, en forma de cuenco, donde la sonda hace contacto con exactamente tres átomos

Para agilizar el proceso, se representa al SES a través de un conjunto disperso de puntos superficiales, compuesto sólo por las caras re-entrantes y cóncavas e ignorando las de tipo toroidal.

Por otro lado, la **estabilidad de una pose** puede ser aproximada a través de la cantidad de área superficial excluida al solvente, conocida como el **área superficial enterrada** (BSA, por sus siglas en inglés). Su valor se obtiene al sumar el BSA de cada proteína, y puede entenderse como el área local SES que se intersecta con la superficie accesible al solvente de la otra proteína (Figura 4.2).

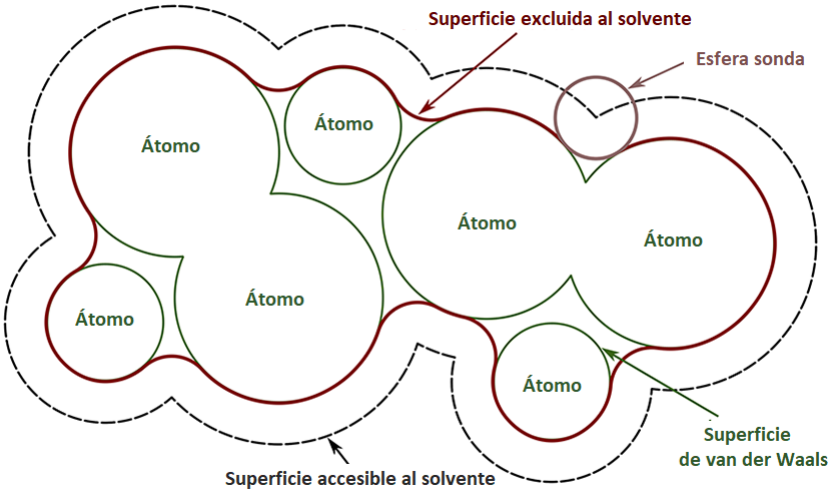


Figura 4.1: Imagen sobre el cálculo de la superficie excluida al solvente. Extraída de la documentación de chimeraX.

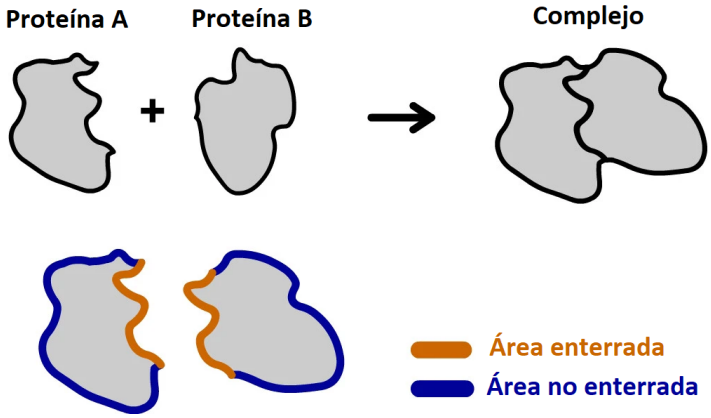


Figura 4.2: Área superficial enterrada (BSA) en un complejo proteína-proteína. Imagen obtenida desde APSDock.

Al colocar dos proteínas rígidas lo más cerca posible una de la otra, sin llegar a atraversarse, existen teóricamente tres puntos de contacto, denominados **puntos de contacto físico** (PCP, por sus siglas en inglés). Sin embargo, en la práctica, pueden existir menos de tres puntos o pueden cruzarse ligeramente entre sí, existiendo por lo tanto puntos superpuestos. A pesar de esto, se espera que las desviaciones del supuesto de PCPs sean pequeñas, proporcionando una aproximación razonable y reduciendo la tarea de encontrar la pose más estable a encontrar puntos de contacto físico.
Es aquí donde se observan las ventajas del uso de **formas de contexto** (CS, por sus siglas en inglés). Una forma de contexto representa la forma local de una proteína, al interior de una esfera centrada en un punto superficial (Figura 4.3). Cada CS es muestreada mediante **rayos de contexto** (CR, por sus siglas en inglés) originados en el centro de la esfera y distribuidos uniformemente sobre ella (Figura 4.4).

Por otra parte, cada rayo de contexto se compone de β bits. Cada bit toma el valor de uno o cero dependiendo si se encuentra dentro o fuera de la superficie o capa superficial.

La superposición de dos formas de contexto, entre dos proteínas, implica la superposición de dos puntos superficiales. El evaluar su complementariedad permite determinar si el par de puntos podría ser un punto de contacto físico (PCP), reduciendo la tarea de encontrar PCPs a evaluar la complementariedad de la forma de todos los pares de formas de contexto.

En cada pose π, se evalúa la complementariedad utilizando operaciones boolenas sobre los rayos de contexto alineados.

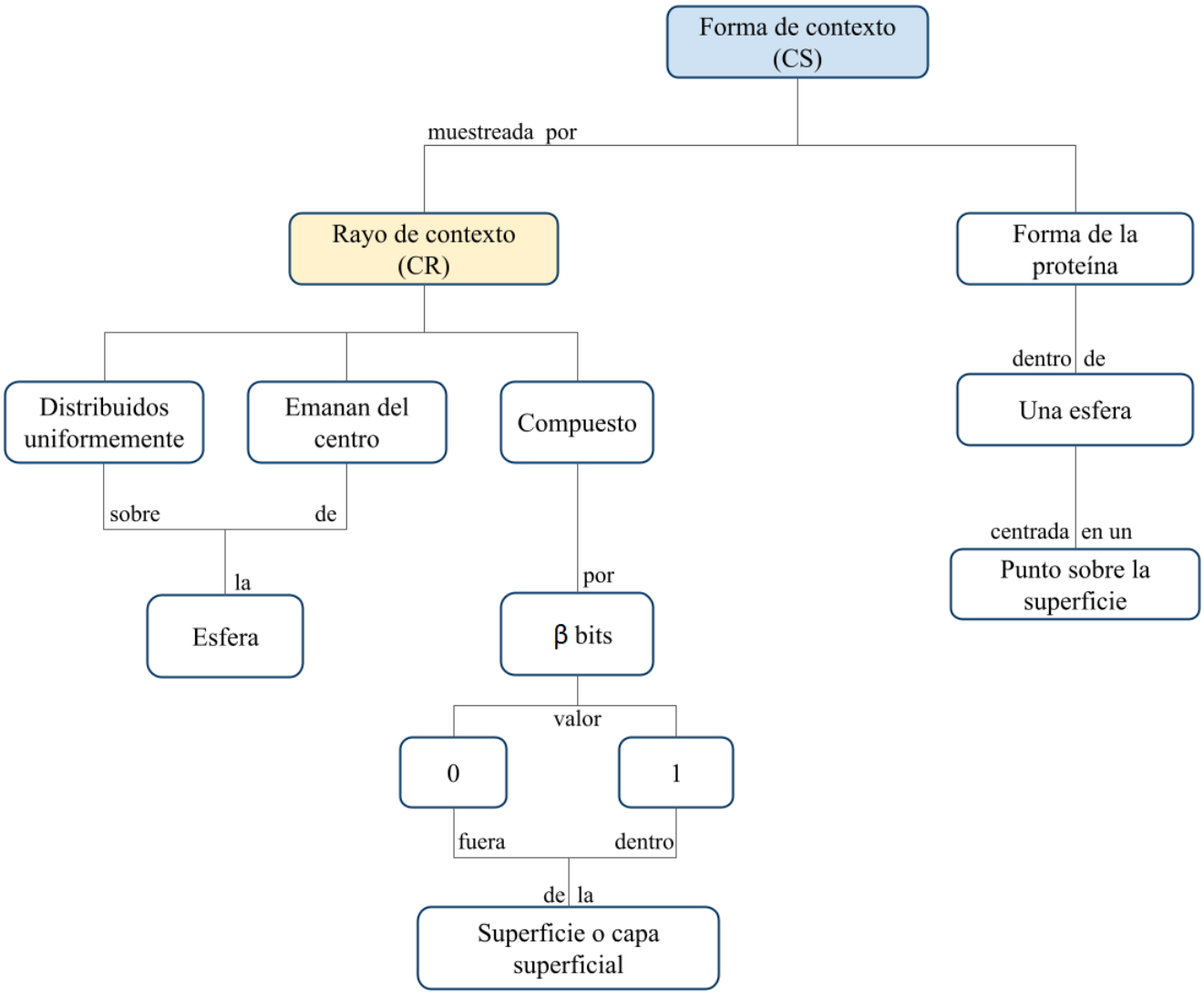


Figura 4.3: Diagrama resumen sobre formas de contexto (CS). Elaboración propia.

En la Figura 4.4 se observa un ejemplo en 2D de la forma de contexto del volumen local en un punto de superficie. El área sombreada en la esfera, representa el volumen local de la proteína en el punto "O". Los rayos de contexto son utilizados para muestrear la forma de contexto (CS). Cada segmento del rayo tiene dos estados posibles; "0" si se encuentra fuera de la capa (líneas punteadas) y "1" si está dentro. Los valores obtenidos son almacenados en un string binario.

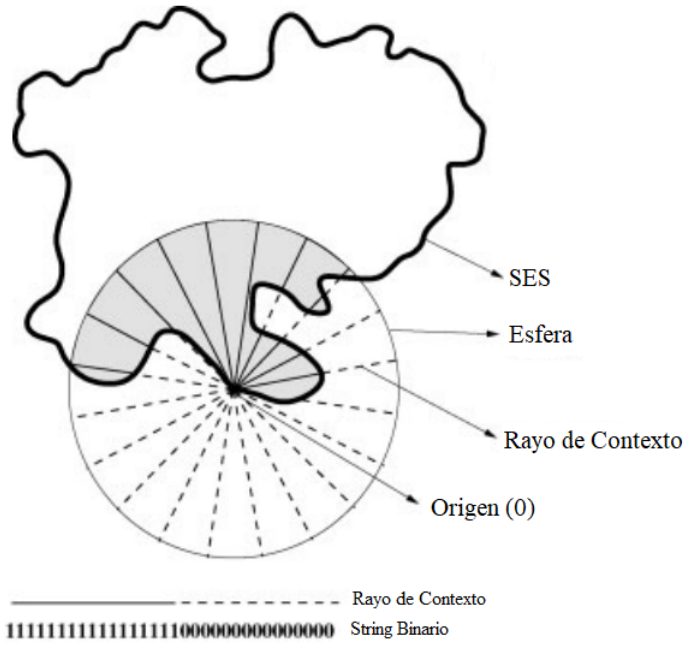


Figura 4.4: Forma de contexto representando el volumen local en un punto de superficie. La proteína, el SES y la esfera son mostrados en 2D por simplicidad. Imagen extraída desde (Shentu et al., 2008).

4.3.2. Capas superficiales

Para evaluar el volumen de superposición y posteriormente, puntuar las poses a través del cálculo del BSA, se definen diferentes capas a una distancia relativa δ del SES. Donde,

$$\delta \in [-r, r] \begin{cases} \delta < 0, & \text{capas internas} \\ \delta > 0, & \text{capas externas} \\ \delta = 0, & \text{SES} \\ \delta = -r, & \text{límite de la esfera dentro del SES} \\ \delta = r, & \text{límite de la esfera fuera del SES} \end{cases} \tag{4.1}$$

Siendo r el radio de la esfera.

Una forma de contexto, corresponde al volumen dentro de la esfera, y es delimitada por dos capas superficiales. Su notación es la siguiente:

$$CS(S_L, S_U)$$

Donde,
S_L denota al límite inferior de la superficie
S_U al límite superior de la superfice

Se definen cuatro tipos de formas de contexto, representadas de forma visual en la Figura 4.5, y denotadas como:

(a) $CS_{vol} = CS(S_{-r}, S_0)$: Forma de contexto del volumen local excluido al solvente
(b) $CS_{SES} = CS(S_0, S_0)$: Forma de contexto del SES local
(c) $CS_{inK} = CS(S_{-k}, S_{-k+1})$: Forma de contexto de volumen de capa interna
(d) $CS_{outK} = CS(S_{k-1}, S_k)$: Forma de contexto de volumen de capa externa

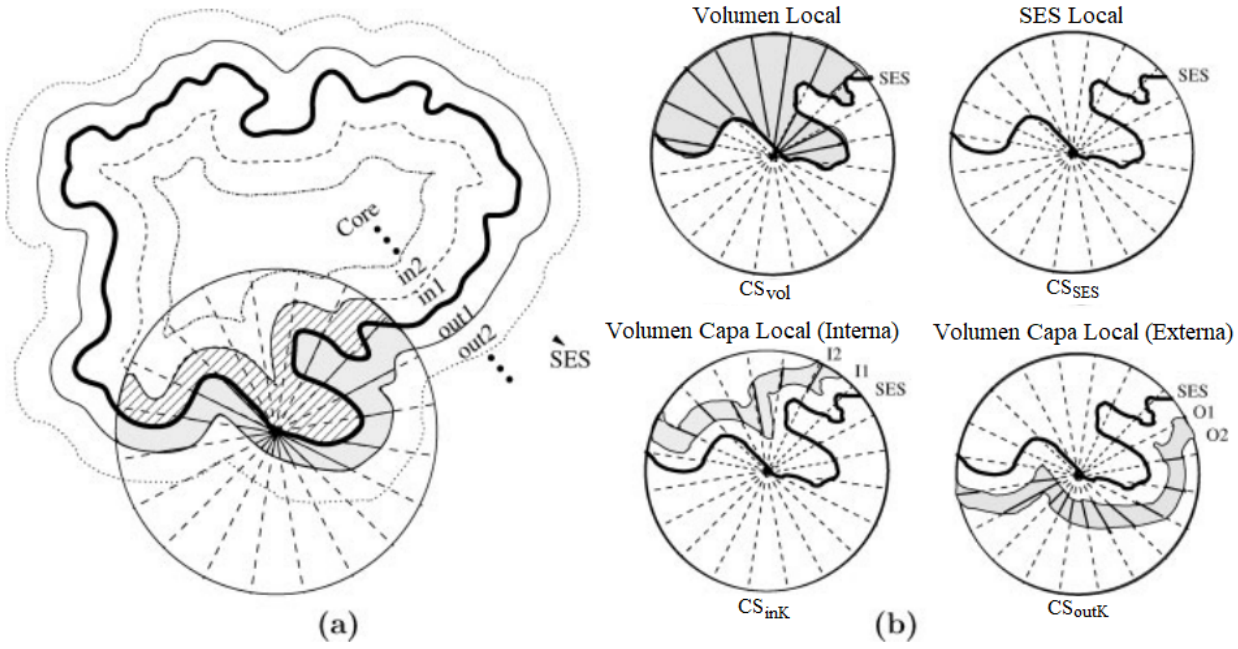


Figura 4.5: **a)** Capas dentro y fuera del SES. Cada capa se encuentra a una distancia relativa de él. **b)** Se muestran cuatro tipos de formas de contexto (región sombreada): *i*) volumen local, *ii*) SES local, *iii*) volumen local capa interna y *iv*) volumen local capa externa. Créditos imagen: (Shentu et al., 2008).

4.3.3. Viabilidad de una pose π

Para determinar si efectivamente una pose π es viable se evalúa el volumen de superposición. Se denomina **volumen de superposición** (OV, por sus siglas en inglés) a la medida en que dos superficies se penetran entre sí en una pose.

Para obtener el valor de la superposición se utilizan las capas descritas previamente, calculando el volumen dentro de dos formas de contexto. De esta manera es posible permitir penetración superficial mas no profunda.

El cálculo de OV es definido como:

$$OV(\ CS^L_{vol}\ ,\ CS^R_X\ ) = \sum_{i=1}^{K} V(\ CR^L_i\ \wedge\ CR^R_{i\pi}\ ) \tag{4.2}$$

Donde $(\ CR^L_i \wedge CR^R_{i\pi}\ )$ representa una operación AND, bit a bit entre dos rayos de contexto (CR).

Además, $V(\ CR^L_i\ \wedge\ CR^R_{i\pi}\ )$ se obtiene calculando:

$$(\ CR^L_i\ \wedge\ CR^R_{i\pi}\ ) = \sum_{j=1}^{\beta} v(j)\ V[j] \tag{4.3}$$

y,

$$v(j) = V(\ CR^L_i\ [j]\ \wedge\ CR^R_{i\pi}\ [j]\ ) \tag{4.4}$$

Con $V[j]$ el volumen actual correspondiente al j-ésimo segmento del rayo de contexto CR.

El volumen de superposición, puede ser dividido en dos tipos dependiendo de la capa X:

(a) **Volumen total de superposición:** La capa X corresponde al volumen local. Representa la superposición de una proteína con respecto a la otra y es una cantidad simétrica (Ecuación 4.5).

(b) **Volumen superpuesto de capas:** La capa X corresponde al volumen local interno (CS^R_{ink}) o externo (CS^R_{outk}). En este caso, parte del volumen de una proteína se superpone a una capa X de la otra proteína y es una cantidad asimétrica (Ecuación 4.6).

$$OV(\ CS^L_{vol}\ ,\ CS^R_{vol},\ \pi\ ) \tag{4.5}$$

$$\begin{array}{c} OV(\ CS^L_{vol}\ ,\ CS^R_X\ ,\ \pi\ ) \\ y \\ OV(\ CS^R_{vol}\ ,\ CS^L_X\ ,\ \pi\ ) \end{array} \tag{4.6}$$

4.3.4. Poses descartadas

Luego de obtener el volumen total de superposición y volumen superpuesto de capas, se consideran inviables todas aquellas poses π que muestren:

(a) **Superposición grande:** La superposición total es mayor a 80Å^3
(b) **Superposición aguda:** El volumen superpuesto de capas es mayor o igual a 5Å^3

4.3.5. Puntuación de la pose π

Si la pose evaluada no presenta superposición grande o aguda, es puntuada empleando para ello el cálculo del BSA. Luego, se clasifica junto al resto de formas de contexto.

Para una pose π dada, el área de superficie enterrada (BSA) de una proteina P^R_L, forma de contexto CS^L_{SES}, con respecto a la forma de contexto CS^R_X para la capa X de la proteína P_R, está dada por:

$$BSA(CS^L_{SES}\ ,\ CS^R_X\ ,\ \pi) = \sum_{i=1}^{K} A(\ CR^L_i\ \wedge\ CR^R_{i\pi}\ ) \tag{4.7}$$

Donde $CR^L_i \in CS^L_{SES}$, y $CR^L_{i\pi} \in CS^R_X$ es un rayo de contexto mapeado con respecto a CR^L_i de acuerdo a la pose π.

El área enterrada es calculada como:

$$A(\ CR^L_i\ \wedge\ CR^R_{i\pi}\ ) = \sum_{j=1}^{\beta} \alpha(j)\ A[j] \tag{4.8}$$

Donde $\alpha(j) = (\ CR^L_i\ [j]\ \wedge\ CR^R_{i\pi}\ [j]\ )$ y $A[j]$ es el área actual correspondiente al punto de superficie representado por el bit j.

El área total de superficie enterrada para la capa X es la suma:

$$BSA(\ CS^L,\ CS^R,\ X,\ \pi\ )\ =\ BSA(\ CS^L_{SES},\ CS^R_X,\ \pi\ )\ +\ BSA(\ CS^R_{SES},\ CS^L_X,\ \pi\ ) \tag{4.9}$$

Mientras la función de puntuación, utilizada para clasificar las diferentes poses, es una suma

ponderada del área enterrada a través de varias capas:

$$P(C\ S^L, CS^R, X, \pi\ ) = w_1 \times BSA(CS^L, CS^R, in1, \pi) + \sum_{K=1}^{3} w_K \times BSA(CS^L, CS^R, outK, \pi) \tag{4.10}$$

Con $w_1 = 4$, $w_2 = 1$ y $w_3 = 0{,}25$ elegidos de forma empírica para optimizar la clasificación, indicando la importancia relativa del área enterrada en cada una de las capas.

La mejor pose π será aquella que obtenga el mayor puntaje:

$$M(CS^L, CS^R) = max_\pi \left\{ P(CS^L, CS^R, \pi) \right\} \tag{4.11}$$

4.3.6. Datos de Entrada

Los datos utilizados para desarrollar el sistema corresponden a archivos PDB. Un archivo PDB (Protein Data Bank) contiene estructuras tridimensionales de moléculas macrobiológicas, determinadas experimentalmente, y son utilizados en todo el mundo por investigadores, estudiantes y educadores.

Dentro de un archivo PDB se encuentran; coordenadas atómicas, factores de estructuras cristalográficas y datos experimentales de espectroscopia de resonancia magnética nuclear (NMR, siglas en inglés).

En la Tabla 4.1 se describen las doce secciones que componen un archivo PDB; título, observación, estructura primaria, heterogéneo, estructura secundaria, anotación de conectividad, características misceláneas, cristalográfica, transformación de coordenadas, coordenadas, conectividad y bookkeeping. Además, se describe brevemente cada sección y los tipos de registros dentro de cada una.

Además de las coordenadas atómicas se incluyen los nombres de las moléculas, información sobre la estructura primaria y secundaria, referencias de bases de datos de secuencias, información de ligandos y ensamblaje biológico, información acerca de la recolección de datos y solución de la estructura y, citaciones bibliográficas.

Sección	**Descripción**	**Tipo de Registro**
Título	Observaciones descriptivas resumidas	HEADER, OBSLTE, TITLE, SPLIT, CAVEAT, COMPND, SOURCE, KEYWDS,EXPDTA, NUMMDL, MDLTYP, AUTHOR, REVDAT, SPRSDE, JRNL
Observación	Varios comentarios sobre anotaciones de entrada con más profundidad que los registros estándar	REMARKS 0-999
Estructura primaria	Secuencia de péptidos y/o nucleótidos y la relación entre la secuencia de PDB y la que se encuentra en las bases de datos de secuencias	DBREF, SEQADV, SEQRES MODRES
Heterogéneo	Descripción de grupos no-estándar	HET, HETNAM, HETSYN, FORMUL
Estructura secundaria	Descripción de estructura secundaria	HELIX, SHEET
Anotación de conectividad	Conectividad química	SSBOND, LINK, CISPEP
Características misceláneas	Características dentro de la macromolécula	SITE
Cristalográfica	Descripción de la celda cristalográfica	CRYST1
Transformación de coordenadas	Operadores de transformación de coordenadas	ORIGXn, SCALEn, MTRIXn
Coordenadas	Datos de coordenadas atómicas	MODEL, ATOM, ANISOU, TER, HETATM, ENDMDL
Conectividad	Conectividad química	CONECT
Bookkeeping	Información resumida, marcador de fin de archivo	MASTER, END

Tabla 4.1: Secciones de un archivo PDB, su descripción y registros dentro de cada sección (Callaway et al., 1996).

4.4. Softwares y lenguajes utilizados

4.4.1. Plataforma de Desarrollo

Para la creación del videojuego, se utilizó el motor de juegos Unity. Esta herramienta es una de las más utilizadas en el área viodejuegos, permitiendo el desarrollo de aplicaciones interactivas de manera más rápida y fácil que otros sistemas similares.
La versión utilizada corresponde a la 2020.1.6f1, última versión disponible en el momento en que se creó la aplicación de docking.

4.4.2. UCSF Chimera

Dentro de las características de este programa se encuentra la posibilidad de visualizar y exportar proteínas en varios formatos. Para este proyecto se exportaron a formato stl las representaciones de las superficies de cada una de las proteínas a utilizar en el juego. Estos archivos fueron utilizados posteriormente tanto para el cálculo de las formas de contexto en Python como para generar los archivos fbx en Blender. La versión utilizada corresponde a Chimera 1.12.

4.4.3. Programa MSMS

El software MSMS es un programa escrito y desarrollado por Michael Sanner en lenguaje C. Permite calcular de manera eficiente para un conjunto de esferas S y una sonda de prueba sp, la superficie reducida y la superficie excluida al solvente (SES) a partir del archivo PDB de la proteína a analizar.

MSMS está compuesto de cuatro algoritmos (Sanner et al., 1996):

(a) El primer algoritmo calcula la superficie reducida de una molécula
(b) El segundo algoritmo construye, a partir de la superficie reducida, una representación analítica de la superficie excluida al solvente que puede auto-intersectarse
(c) El tercer algoritmo descarta todas aquellas partes que se auto-intersectan
(d) El cuarto algoritmo genera una triangulación del SES

4.4.4. Blender

Blender es una suite 3D gratuita y de código abierto. Permite modelado, montaje, animación y renderizado 3D, entre varias otras funcionalidades. Para este trabajo fue utilizado con el objetivo de generar archivos en formato fbx, compatibles con Unity, con las superficies de las proteínas a utilizar en el juego. Otra de las razones de la elección de este software sobre otros similares es la necesidad de aplicar una rotación a los archivos stl de las superficies. Esto ya que Unity utiliza el sistema coordenado de mano izquierda, mientras que Chimera y Blender utilizan el sistema coordenado de mano derecha (Figura 4.6). En Blender existe un plugin que permite realizar de manera sencilla el cambio de sistema para renderizar correctamente las proteínas en Unity. La versión utilizada corresponde a Blender 2.91.2.

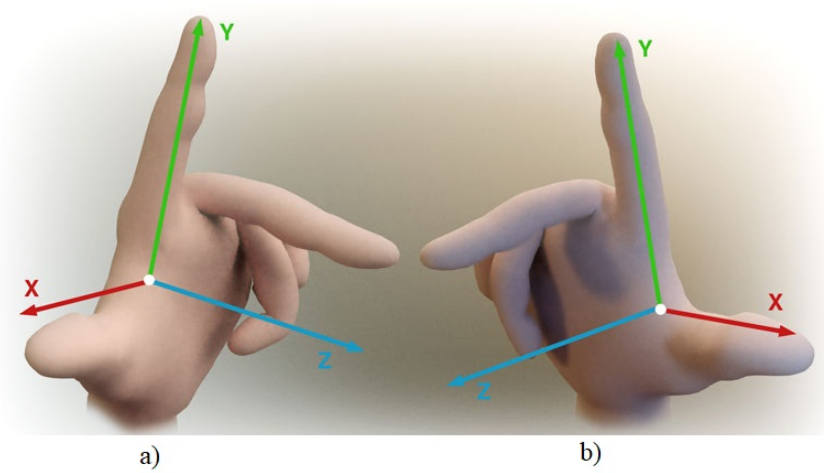


Figura 4.6: a) Sistema de coordenadas utilizado en Unity. b) Sistema de coordenadas utilizado en Blender y Chimera. . Créditos imagen: Primalshell.

4.5. Lenguajes

Para llevar a cabo la evaluación de las poses π generadas por los usuarios de la aplicación, es necesario efectuar los cálculos involucrados en el algoritmo detallado en la Sección 4.3, de manera offline y online. Dependiendo de esto se utilizan dos lenguajes:

(a) **Cálculos offline:** Estos incluyen la generación de las formas de contexto de cada proteína incorporada en el juego, y son implementados en Python. La versión utilizada corresponde a Python 3.8.5.

(b) **Cálculos online:** Incluyen el cálculo del volumen superpuesto, BSA y puntaje final de cada pose π. Estos cálculos deben realizarse mientras el usuario hace uso de la aplicación, por lo que se debe emplear un lenguaje compatible con Unity. El lenguaje permitido, para crear scripts, por defecto en Unity es C# y es el ocupado en este caso. La versión utilizada es C# 8.0.

4.6. Metodología

4.6.1. Procedimiento para el cálculo de formas de contexto

En este apartado se detalla el procedimiento empleado para obtener las formas de contexto de las proteínas que se desea acoplen los usuarios. Los cálculos principales son realizados de forma offline en Python y los archivos generados son ocupados posteriormente en Unity.

La Figura 4.7 resume a través de un diagrama los principales pasos para obtener los archivos que contienen las formas de contexto, necesarias para implementar los cálculos de volumen superpuesto y área superficial enterrrada en el motor de videojuegos.

A continuación se detallan en más profundidad cada una de las etapas:

(a) **Descarga de archivos PDB:** En primer lugar se descargan los archivos PDB de cada proteína perteneciente al par que se desea acoplen los usuarios. De forma preliminar, se utilizan archivos PDB del Benchmark proteína-proteína 5.0 disponibles en `https://zlab.umassmed.edu/benchmark/`.

(b) **Obtención archivo .STL del SES:** Cada proteína es visualizada en el software Chimera, seleccionando la representación de su superficie. Esta representación es exportada en formato .STL para ser utilizada posteriormente en Python y Blender.

(c) **Cálculo de SES empleando MSMS:** Es utilizado el programa MSMS con cada proteína para obtener la triangulación del SES, compuesta por dos archivos; un archivo .vert que contiene las coordenadas de todos los vértices y, un archivo .face que contiene los índices de los vértices de cada triángulo, además de indicar a qué tipo de cara corresponde (de contacto, re-entrante o toroidal).

En la Figura 4.8 se resumen los pasos *a*), *b*) y *c*) para obtener los datos de entrada necesarios en los algoritmos implementados en Python.

(d) **Obtención de posibles puntos de contacto físico:** Una vez conseguidos los archivos .vert y .face se realiza el procesamiento en Python, descartando todas aquellas caras de tipo toroidal, quedando sólo las de contacto y re-entrantes. A estas caras restantes se les calcula su centroide.

(e) **Definición de coordenadas para formas y rayos de contexto:** Cada centroide es utilizado como el origen de una esfera para representar las formas de contexto asociadas a ese punto y de esta forma obtener una representación local de la forma de la proteína. Todas las esferas poseen el mismo radio y son muestreadas de manera uniforme a través de cien rayos ($K = 100$), almacenando en un archivo de texto plano las coordenadas de origen y fin de cada rayo.

(f) **Generación de formas de contexto:** Una vez obtenido el archivo .STL y las coordenadas de los rayos de cada esfera se calculan las formas de contexto asociadas a cada centroide. Cada rayo se divide en β segmentos ($\beta = 15$ en este caso) de igual longitud y se evalúa si el segmento se encuentra dentro o fuera de la capa superficial correspondiente. Si el segmento se encuentra dentro se le asigna el valor "1", y "0" de lo contrario.

En este caso se calcularon ocho formas de contexto para cada centroide:

- Una forma de contexto de volumen local (CS_{vol})
- Una forma de contexto del SES local (CS_{SES})
- Tres formas de contexto de volumen de capa interna (CS_{in1}, CS_{in2} e CS_{in3})
- Tres formas de contexto de volumen de capa externa (CS_{out1}, CS_{out2} e CS_{out3})

(g) **Exportación de las formas de contexto:** Para poder utilizar las formas de contexto calculadas anteriormente, para cada proteína y forma se contexto, se exportan a un archivo de texto plano el índice de la forma de contexto, el índice del rayo, sus coordenadas de origen y fin y, los valores obtenidos para cada segmento del rayo. Además, también se exporta en otro archivo de texto las coordenadas e índice de cada centroide. Estos archivos serán utilizados en el procesamiento online en Unity.

Este mismo procedimiento realizado en Python se muestra de forma sintetizada en la Figura 4.9, del cual se obtienen los archivos de salida necesarios para implementar la plataforma en Unity.

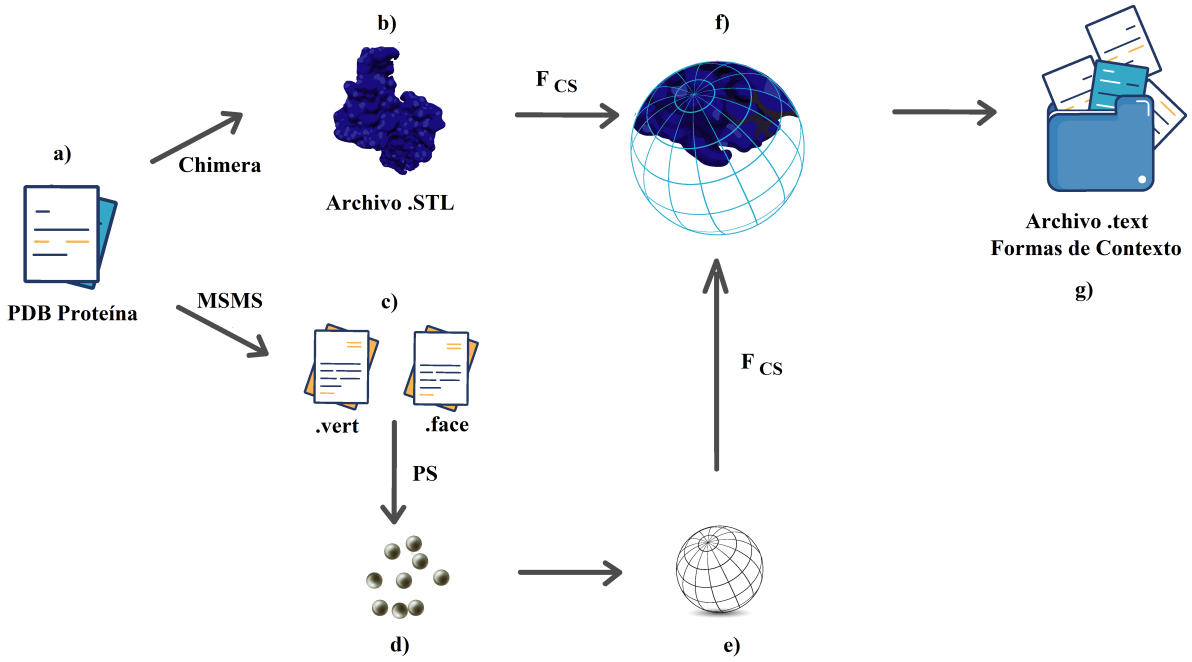


Figura 4.7: Resumen del procesamiento realizado para obtener los archivos con las formas de contexto para una proteína. a) Descarga de archivos PDB, b) obtención archivo .STL del SES, c) cálculo del SES utilizando MSMS, d) obtención de posibles puntos de contacto físico, e) definición coordenadas de los rayos de contexto, f) generación de formas de contexto y g) exportación de formas de contexto.

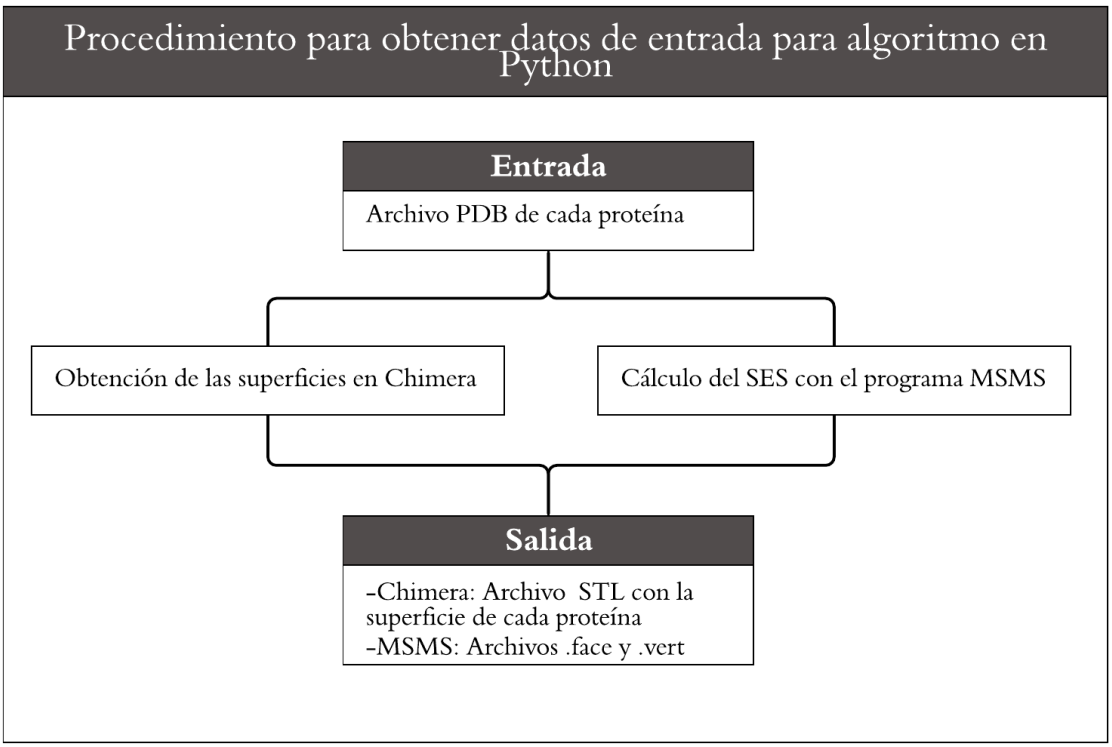


Figura 4.8: Resumen de la metodología utilizada para generar los datos de entrada utilizados en algoritmo en Python.

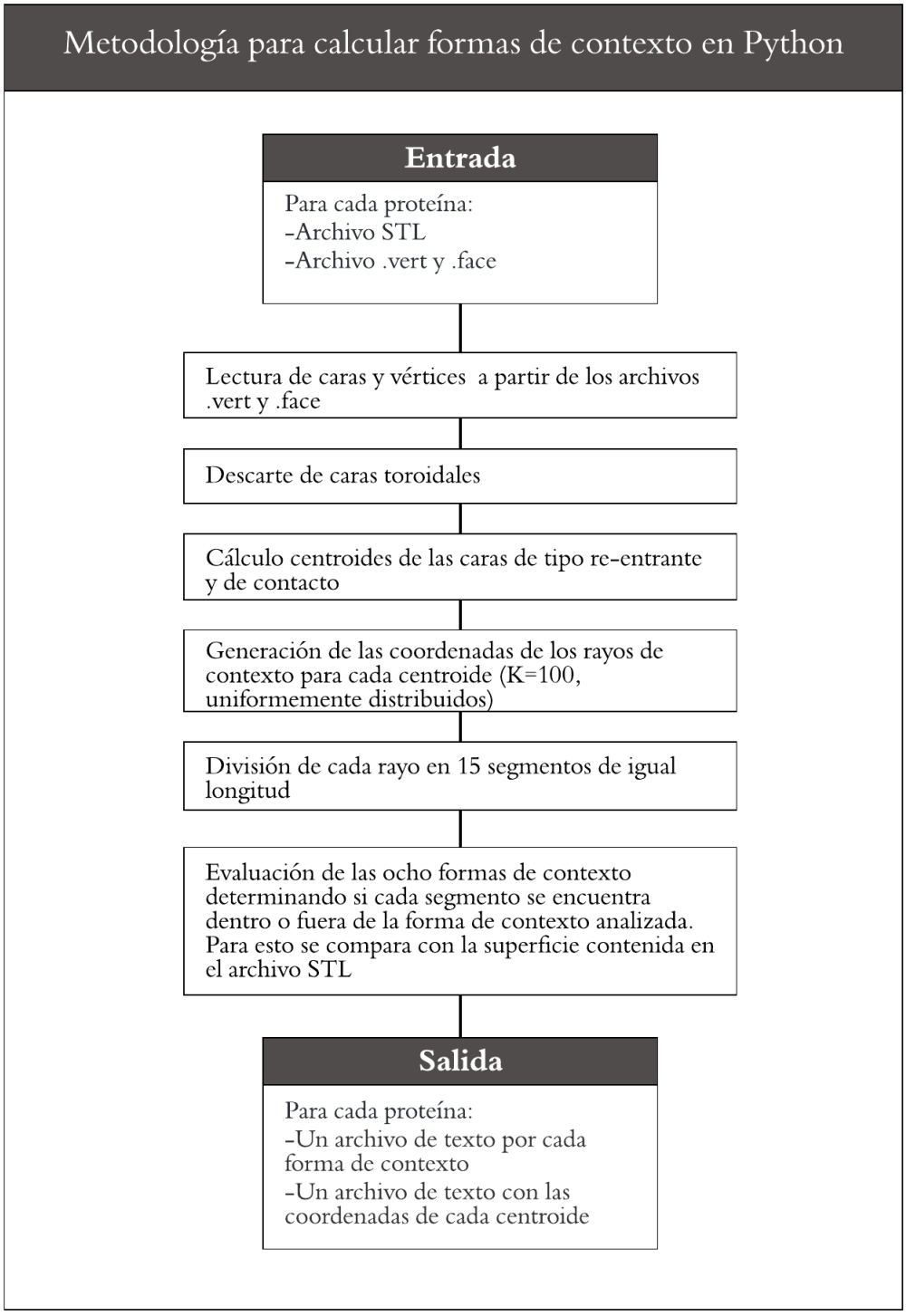


Figura 4.9: Resumen de la metodología utilizada para generar las formas de contexto en Python.

4.6.2. Procedimiento en Unity

En el caso del motor de videojuegos, existe procesamiento tanto offline como online.

En la Figura 4.10 se observa un resumen del desarrollo de la aplicación interactiva. De forma preliminar fue implementada en un entorno local, pero con el objetivo final de subirla a la web, para así tener una mayor cantidad de usuarios utilizando la aplicación. De esta manera se espera mejorar los resultados a la hora de obtener el ranking de las mejores poses acopladas por los jugadores.

La aplicación desarrollada en Unity necesita los siguientes datos como base:

- **Archivos fbx de la superfice cada proteína:** Para poder renderizar las superficies tridimensionales de cada proteína que estarán disponibles para acoplar en el juego, es necesario transformar los archivos .STL obtenidos en la Sección 4.6.1 a un formato compatible con Unity. Para esto, son transformados y rotados en Blender, que permite realizar este proceso de manera sencilla y rápida a través de un plugin.

- **Archivos con formas de contexto:** Para poder evaluar la viabilidad de las poses y el puntaje obtenido por cada jugador se necesitan los archivos que contienen las formas de contexto, calculadas en la Sección 4.6.1, para cada proteína. Además, para ayudar en el proceso del acoplamiento se generan esferas con origen en los centroides calculados previamente, utilizando también los archivos que contienen la información de las coordenadas de estos puntos.

Las etapas llevadas a cabo para implementar la aplicación interactiva en Unity comprenden:

(a) **Transformación y rotación de la superficie de cada proteína:** Como se explicó previamente, es necesario convertir la superficie obtenida en el programa Chimera a un formato compatible y con el mismo sistema de coordenadas que Unity. Para esto se utiliza el plugin Unity FBX.

(b) **Lectura de archivos que contienen las formas de contexto de cada proteína:** Para poder realizar los cálculos online de volumen superpuesto y BSA, es necesario tener disponibles los archivos de CS en la aplicación desarrollada.

(c) **Implementación algoritmos:** Tomando como entrada los datos de *a*) y *b*) se generan los scripts que controlan el comportamiento de la aplicación. A continuación se describen los principales:

   - *Renderizar-proteinas.cs* : Encargado de visualización de cada par de proteínas cuando el usuario inicia una partida en la aplicación

- *Eventos-proteinas.cs* : Implementa los eventos que le permiten al usuario manipular cada par de proteínas (traslación y rotación en una y sólo rotación en la restante). De esta manera los jugadores pueden efectuar el docking de cada par

- *Lectura-centroides.cs* : Encargado de leer y almacenar las coordenadas de los centroides y formas de contexto asociadas a cada uno de ellos. Renderiza estos centroides como esferas para ayudar al usuario en el proceso de acoplamiento

- *Calculo-ov-puntaje.cs* : Detecta la colisión entre dos esferas de diferentes proteínas y calcula el volumen de superposición. De esta manera es posible evaluar la viabilidad de la pose actual. Si la pose es viable, además calcula a partir del BSA el puntaje asociado al docking realizado

- *Actualizar-puntaje.cs* : Se encarga de actualizar los archivos que contienen la información de los mejores acoplamientos para el par de proteínas una vez el jugador haya finaliza su intento

En Unity se trabaja con objetos, compuestos por diferentes componentes. Cada proteína es un objeto y los scripts son agregados como componentes de él, exceptuando el primero y último que son parte del objeto raíz que contiene a las proteínas. Por otro lado, las esferas son creadas como objetos hijos en cada proteína.

Todos los algoritmos descritos previamente, trabajan de forma síncrona con la interacción de los jugadores y son implementados en C#, lenguaje compatible por defecto con Unity.

(d) **Interacción de usuarios**: Para que el algoritmo pueda ser ejecutado totalmente, es necesario que existan jugadores que utilicen la aplicación. Es el usuario quién realiza el docking proteína-proteína, la plataforma se encargará de evaluar la calidad y viabilidad de la pose realizada.

(e) **Viabilidad del acoplamiento:** Mientras el jugador realiza su intento, por pantalla se le muestra un semáforo que le indica si es viable o no la pose actual. Esto, a partir del cálculo del volumen superpuesto para cada esfera que se encuentre colisionando con una esfera de la otra proteína. Se fijó un límite de siete esferas colisionado al mismo tiempo para ser una pose viable, ya que en teoría existen aproximadamente tres puntos de contacto físico. Además, permitir un número muy grande de colisiones disminuye el rendimiento de la aplicación. Este límite es modificable.

(f) **Cálculo puntaje del docking proteína-proteína:** Si la pose es viable y el usuario termina el intento, se calcula el BSA, para obtener con esto el puntaje de la pose acoplada de

acuerdo al algoritmo descrito en el capítulo.

(g) **Ranking del acoplamiento**: Se compara el puntaje obtenido para el docking proteína-proteína realizado por el usuario actual, con los puntajes previos obtenidos por otros usuarios para el mismo par de proteínas.

(h) **Actualización mejores acoplamientos**: Si el puntaje obtenido está dentro de los mejores diez para ese par de proteínas, se actualiza el archivo que contiene esta información para el par de proteínas correspondientes.

Este procedimiento es repetido para cada par de proteínas diponible en la plataforma y puede observarse de manera simplificada en la Figura 4.11.

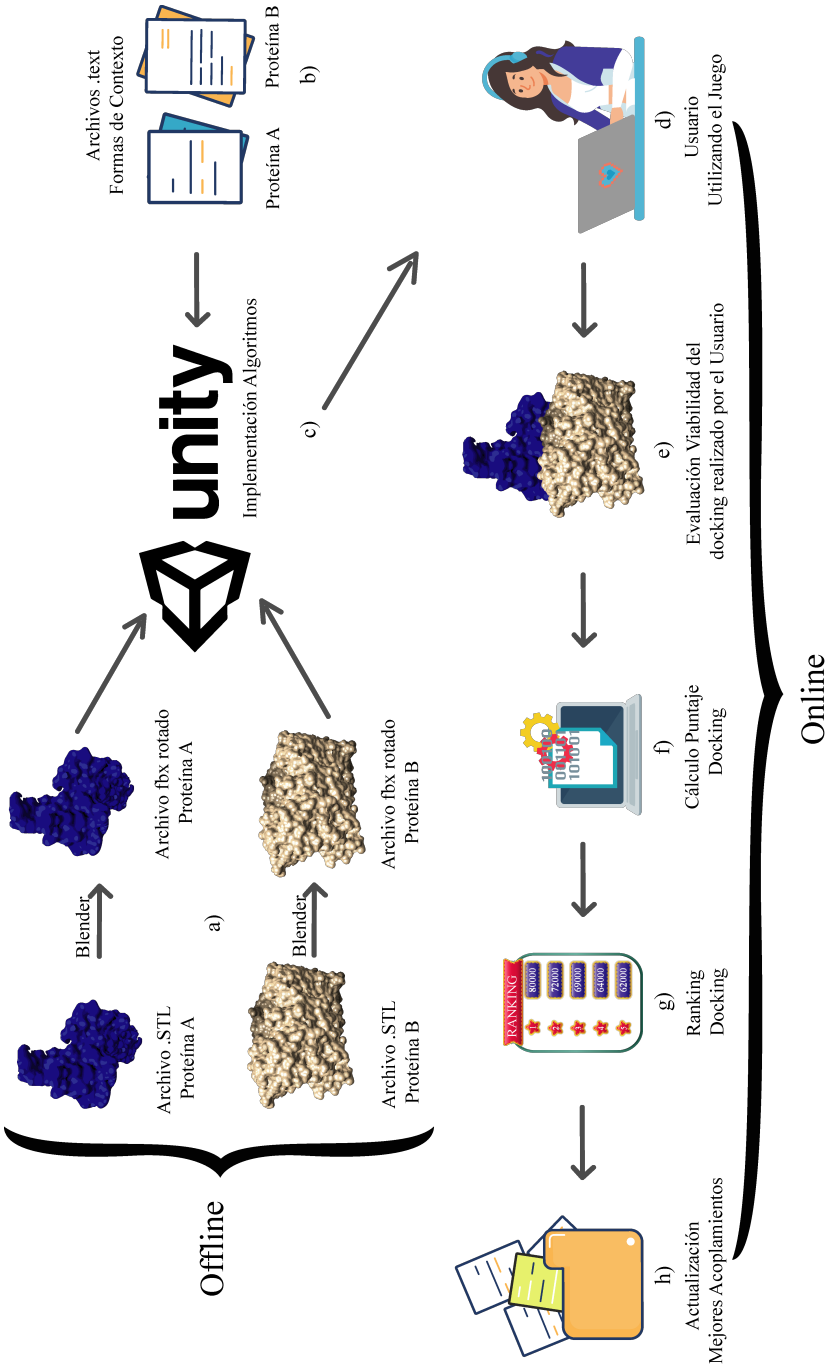


Figura 4.10: Imagen acerca de la aplicación de docking proteína-proteína desarrollada en Unity. a) Transformación de archivos .STL a .fbx, b) Archivos con datos de las formas de contexto, c) los datos obtenidos en a) y b) son cargados en la aplicación, d) es necesario la interacción de un usuario en la plataforma, e) jugador realiza un acoplamiento y se le indica la viabilidad actual, f) se realiza de forma online el cálculo del docking realizado, g) se clasifica de acuerdo al puntaje y h) se actualiza el listado de mejores puntajes de ser necesario.

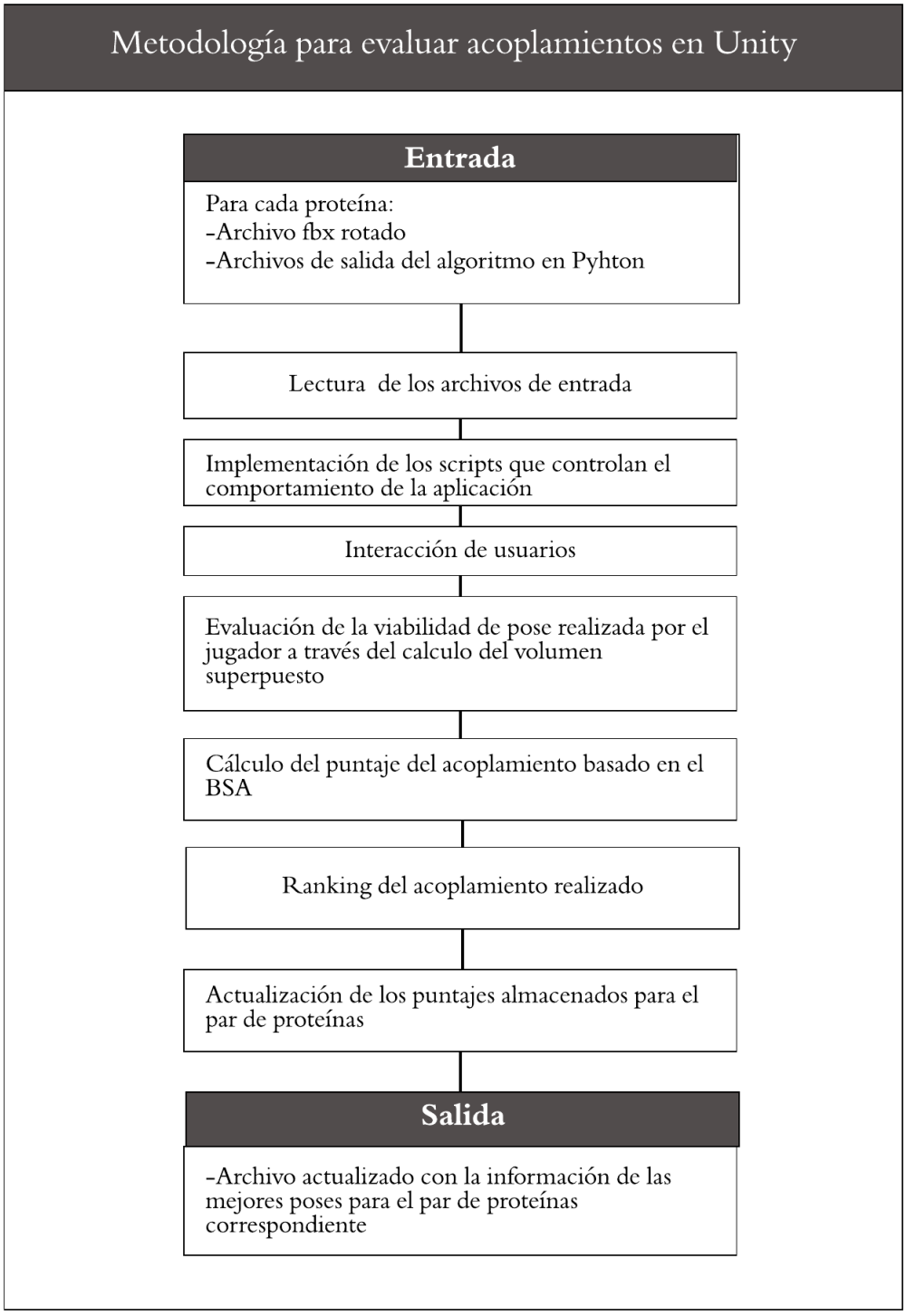


Figura 4.11: Resumen de la metodología utilizada para el desarrollo de la aplicación en Unity.

Capítulo 5

Resultados

En este capítulo se presenta el software desarrollado luego de llevar a cabo la metodología descrita en la sección 4.6.2 y se realiza una breve discusión sobre lo obtenido. Como se mencionó previamente, para crear el sistema se utilizó el motor de videojuegos Unity 3D.

5.0.1. Aplicación Desarrollada

Pantalla Inicial

En primer lugar, cuando el usuario accede a la plataforma se despliega la pantalla de bienvenida al juego. Ésta contiene el nombre del juego y un botón para iniciar una partida. En la Figura 5.1 se observa la interfaz que ve el usuario al entrar a la aplicación.

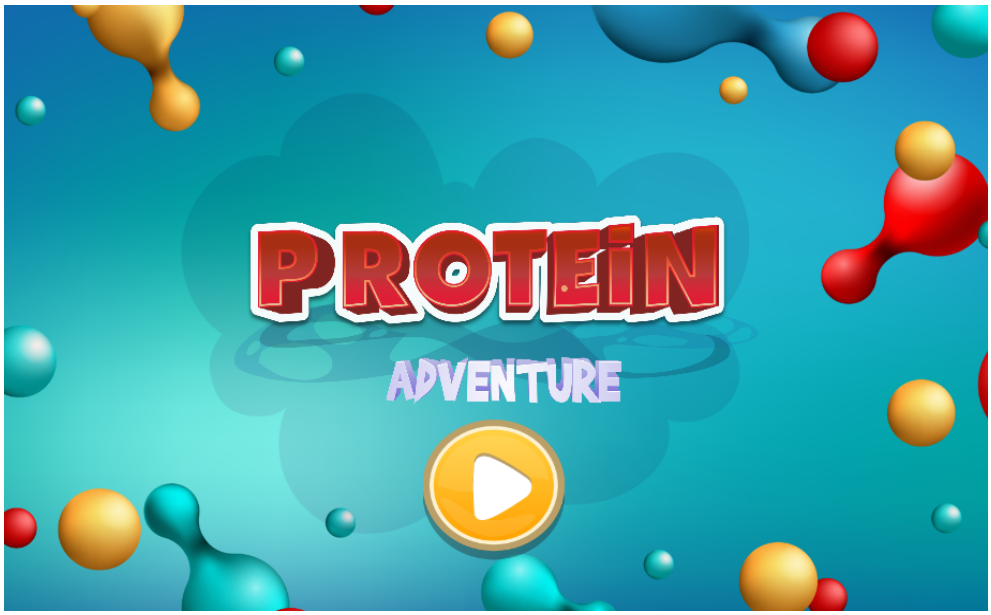


Figura 5.1: Pantalla inicial mostrada al usuario al utilizar el sistema desarrollado.

El jugador debe presionar el botón para iniciar la partida, de lo contrario no podrá acceder a los pares de proteínas disponibles para acoplar.

Pantalla Partida Iniciada

Una vez presionado el botón, desaparece la pantalla de inicio y se despliega el primer par de proteínas a acoplar, como se observa en la Figura 5.2.

Cada proteína es cargada de forma automática y se encuentra almacenada previamente en los archivos de la aplicación. Además también se han almacenado con anterioridad los archivos que contienen la información acerca de los posibles puntos de contacto físico y sus correspondientes formas de contexto de acuerdo al algorimto utilizado y que fueron calculadas de forma offline como se describió en la sección anterior.

Para ayudar al usuario en el proceso de docking se señalan los posibles puntos de contacto a través esferas en ambas proteínas. Una posible pose será aquella donde se superpongan dos o más esferas entre ambas proteínas.

Por otra parte, también se le señala si el acoplamiento que está realizando hasta el momento es viable o no, a través de un semáforo. Si no es viable o no existe contacto entre las proteínas se muestra una mano con el pulgar hacia abajo y con fondo rojo (Figura 5.3.a)). Mientras que si es viable se le muestra un pulgar hacia arriba con fondo verde como se observa en la Figura 5.3.b).

La viabilidad de la pose es evaluada a través del cálculo de volumen superpuesto. De acuerdo al algoritmo utilizado no son viables aquellas poses que tienen superposición grande o aguda. Para ello se calcula constantemente la superposición entre las formas de contexto de los puntos de superficie que estén en colisión. Además, se fijó un límite de puntos de contacto físico que pueden estar colisionando entre ambas proteínas.

Sólo se calcula el puntaje de aquellas poses que son viables, de lo contrario a pesar de que el usuario presione el botón para terminar el intento (botón en el medio del menú inferior) el puntaje será cero.
Si la pose es viable, como se muestra en la Figura 5.3.b) internamente se calcula automáticamente el puntaje del acoplamiento. Para ello se evalúa el BSA de cada par de puntos de contacto físico que se encuentren en colisión en la pose realizada por el usuario. Luego, para obtener el puntaje se determina el valor máximo de BSA obtenido entre los pares en colisión y el puntaje final del acoplamiento es calculado en base a él.

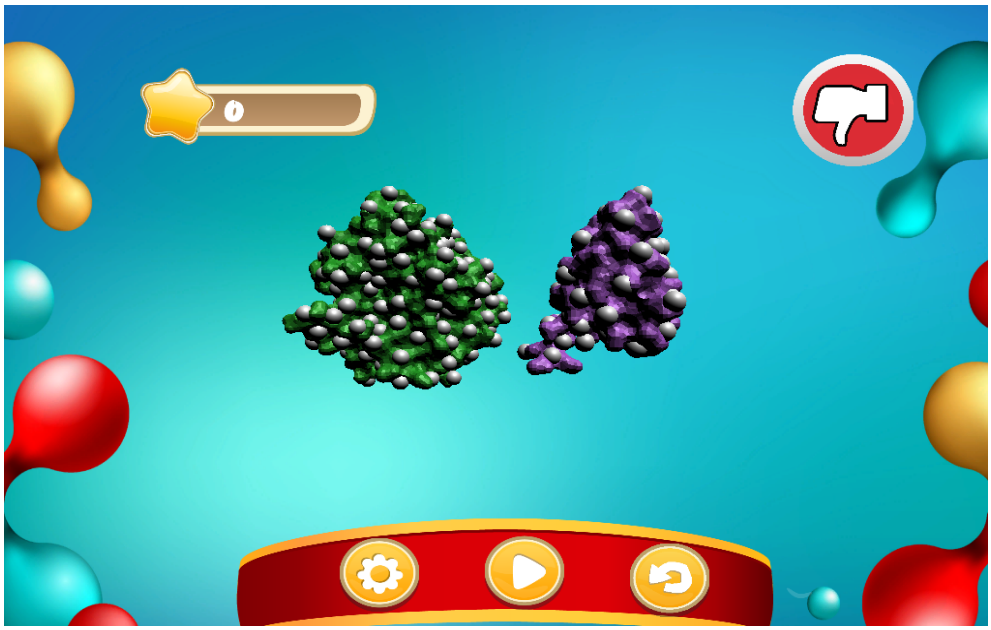
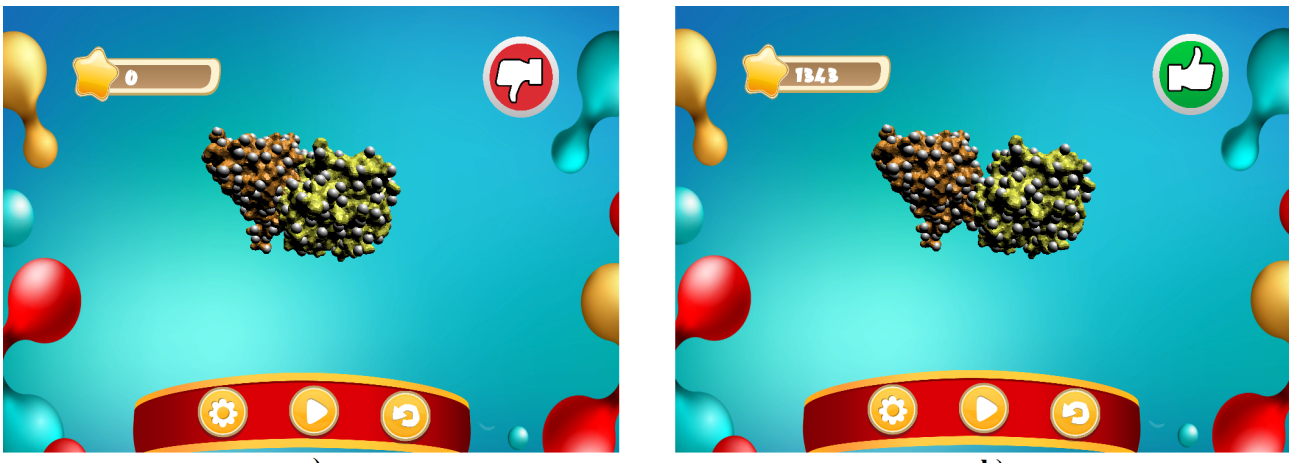


Figura 5.2: Pantalla mostrada al usuario luego de iniciar la partida. Se despliega el primer par de proteínas que debe acoplar el jugador actual. Inicialmente el semáforo está en rojo ya que no hay ningún punto de contacto entre ambas proteínas.



a) b)

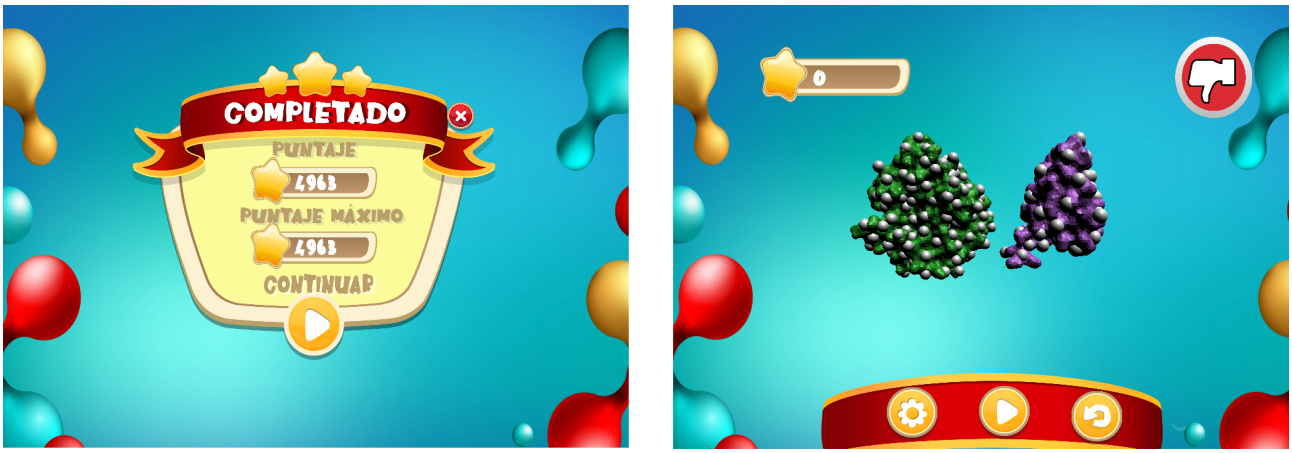
Figura 5.3: a) Pose realizada por el usuario no es viable, el semáforo se muestra en rojo. b) Pose obtenida por el usuario es viable, el semáforo cambia a color verde.

Pantalla Puntaje

En la Figura 5.4.a) se muestra la pantalla desplegada al usuario el terminar su intento de acoplamiento, donde puede ver el puntaje obtenido, el puntaje máximo registrado y, además, tiene la posibilidad de realizar un nuevo acoplamiento entre otro par de proteínas o volver a intentar con el mismo par.

Si el jugador decide continuar con el siguiente par de proteínas, se cargan en pantalla las siguientes proteínas y se repite el mismo procedimiento descrito anteriormente, como se observa en la Figura 5.4.b).

Los puntajes obtenidos en cada docking viable son evaluados de manera interna. Cada par de proteínas tiene un archivo con las poses con mayor puntaje (las primeras diez, pero este número es modificable), y asociado a estos puntajes existe un archivo que contiene la información acerca de los puntos de superficie en contacto entre ambas proteínas que dan origen a la pose correspondiente. El puntaje y pose realizados por el usuario sólo serán almacenados en los archivos si se encuentran dentro de los diez (o el número definido) primeros, de lo contrario la pose se descarta.



a) b)

Figura 5.4: a) Pantalla mostrada al usuario cuando termina el intento de acoplamiento actual, donde se despliega el puntaje obtenido por la pose y el máximo registrado hasta el momento. b) Carga del siguiente par de proteínas a acoplar por el usuario.

5.0.2. Discusión

El resultado principal en este proyecto corresponde a la creación de un sistema interactivo que permite ludificar el proceso de docking proteína-proteína.

Este sistema evalúa las poses generadas por los usuarios utilizando para ello un método basado en complementariedad local de la forma.

Sin embargo, existen varios puntos pendientes que por limitaciones de tiempo no son incluidos en el alcance de este proyecto pero, deben ser llevados a cabo en un futuro para cumplir con el objetivo de aumentar los datos sobre estructuras de complejos de proteínas.

Dentro de estos puntos se encuentra en primer lugar implementar el juego desarrollado de forma online para llegar a la mayor cantidad de usuarios posibles. De esta manera será posible reunir un número de datos suficientes para evaluar el porcentaje de éxito del algoritmo utilizado a la hora de distinguir las poses más probables de acoplamiento entre los pares de proteínas disponibles en el sistema.

Luego de evaluar esto será posible realizar ajustes al algoritmo o corregir posibles errores en la implementación de este.

Por último, se debe evaluar el método de almacenamiento para guardar la información generada por el sistema. Actualmente, debido a que el software se desarrolló en un entorno local no hace uso de ninguna base de datos para ello.

Sin embargo, si el sistema se implementa de forma online será necesario utilizar una base de datos para administrar la información sobre las mejores poses realizadas por los usuarios y los puntajes obtenidos. De esta manera es posible administrar la información de forma segura y eficiente.

Capítulo 6

Conclusiones

A partir de estructuras individuales de proteínas contenidas en archivos PDB y utilizando el método de complementariedad local de la forma conocido como **formas de contexto**, se generó una plataforma interactiva en el motor de videojuegos Unity para el problema de docking proteína-proteína cumpliendo con el objetivo principal de este trabajo.

En este sistema, los usuarios realizan acoplamientos entre pares de proteínas disponibles en la plataforma, visualizando de forma tridimensional la superficie de cada una de ellas. Tienen la posibilidad de manipular cada proteína para realizar el acoplamiento y mientras se encuentran ejecutando el intento son ayudados a través de un semáforo que indica si la pose actual es viable o no. Una vez que finalizan el docking la pose final es puntuada.

Para implementar esta aplicación se separó su desarrollo en dos partes.

(a) **Generación de formas de contexto:** Utilizando Python se implementó el algoritmo asociado al método de formas de contexto, entregando como resultado los datos necesarios para evaluar posteriormente viabilidad y calidad de las poses efectuadas por los usuarios de la aplicación interactiva.

(b) **Evaluación de acoplamientos:** Tomando como entrada las formas de contexto y la superficie tridimensional de cada proteína se desarrolló la ludificación del problema de docking proteína-proteína utilizando el motor Unity.

A las poses realizadas por el usuario se les evalúa el volumen superpuesto para obtener la viabilidad y posteriormente, se efectúa el cálculo del BSA para obtener el puntaje asociado a la pose.

Para cada par de proteínas en el sistema se almacenan los mejores puntajes obtenidos por los usuarios, siendo actualizados cada vez que se obtiene un nuevo acomplamiento con

un puntaje dentro de los diez primeros registrados.

De esta manera fue posible ludificar el problema de docking proteína-proteína. Sin embargo, para que el sistema desarrollado efectivamente permita aumentar el conjunto de datos de estructuras de complejos de proteína-proteína disponibles es necesario realizar trabajo a futuro.

6.0.1. Trabajo Futuro

Dentro de las principales tareas pendientes se encuentran:

(a) **Generar documentación de la aplicación desarrollada**: Es necesario crear la documentación asociada al desarrollo del sistema ya que de esta manera, podrá ser utilizada por otras personas interesadas en un enfoque de docking ludificado, facilitando la continuación del proyecto realizado en este trabajo.

(b) **Implementar la aplicación en la web:** Para que realmente el sistema pueda ser utilizado por la mayor cantidad de usuarios posibles es necesario que esté disponible de forma online. De esta forma, será posible incentivar rápidamente su uso y con esto aumentar la cantidad de acoplamientos con mejor puntaje almacenados por el sistema. Además, resultaría beneficioso dejar disponibles a la comunidad, en el mismo sitio web, los resultados que se generen del uso del juego, para que estén accesibles a otros investigadores interesados en esta área.

(c) **Almacenar la información de los puntajes en una base de datos:** Al buscar una gran interacción de usuarios con la plataforma, se hace necesario almacenar los datos de las poses con mejor puntaje en una base de datos. De lo contrario, el rendimiento de la aplicación se verá afectado y además, el acceso a la información generada no será el más adecuado para el objetivo principal.

(d) **Evaluación de las poses y ranking obtenidos:** Una vez la aplicación esté operativa de forma online, será necesario analizar los resultados obtenidos por los usuarios. Para ello se debe evaluar el éxito del algoritmo implementado a la hora de determinar las poses más probables generadas por los jugadores. A partir de este análisis será posible realizar ajustes y añadir mejoras a los algoritmos implementados, complementarlos o probar con otros similares.

(e) **Recibir retroalimentación de los usuarios:** Esto permitirá realizar mejoras en la interfaz para que la cantidad de personas que utilicen la aplicación aumente con el tiempo.

De esta manera, se espera que la aplicación implementada en este proyecto sea un aporte real en el problema de la predicción de estructuras de complejos proteicos utilizando un enfoque menos convencional como es la ludificación.

Referencias

Angshuman Bagchi. Protein-protein interactions: Basics, characteristics, and predictions. En *Soft Computing for Biological Systems*, págs. 111–120. Springer, 2018.

Antonio Blanco y Gustavo Blanco. Chapter 3 - proteins. En Antonio Blanco y Gustavo Blanco, eds., *Medical Biochemistry*, págs. 21–71. Academic Press, 2017. ISBN 978-0-12-803550-4.

J Callaway, M Cummings, B Deroski, P Esposito, A Forman, P Langdon, M Libeson, J McCarthy, J Sikora, D Xue, et al. Protein data bank contents guide: Atomic coordinate entry format description. *Brookhaven Natl Lab*, 1996.

Sheng-You Huang. Search strategies and evaluation in protein–protein docking: principles, advances and challenges. *Drug discovery today*, 19(8):1081–1096, 2014.

Sheng-You Huang. Exploring the potential of global protein–protein docking: an overview and critical assessment of current programs for automatic ab initio docking. *Drug Discovery Today*, 20(8):969–977, 2015.

Sheng-You Huang, Sam Z Grinter, y Xiaoqin Zou. Scoring functions and their evaluation methods for protein–ligand docking: recent advances and future directions. *Physical Chemistry Chemical Physics*, 12(40):12899–12908, 2010.

Guillaume Levieux, Guillaume Tiger, Stéphanie Mader, Jean-François Zagury, Stéphane Natkin, y Matthieu Montes. Udock, the interactive docking entertainment system. *Faraday discussions*, 169:425–441, 2014.

Michel F Sanner, Arthur J Olson, y Jean-Claude Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, 1996.

Zujun Shentu, Mohammad Al Hasan, Christopher Bystroff, y Mohammed J Zaki. Context shapes: Efficient complementary shape matching for protein–protein docking. *Proteins: Structure, Function, and Bioinformatics*, 70(3):1056–1073, 2008.

Ilya A Vakser. Protein-protein docking: From interaction to interactome. *Biophysical journal*, 107(8):1785–1793, 2014.

Camilo Ignacio Vega Hidalgo et al. *Ludificación de docking molecular para acelerar el diseño de fármacos*. Tesis Doctoral, Universidad de Concepción. Facultad de Ingeniería., 2018.