Jon Navarro Garitano

Predicción de la actividad ligando-proteína basada en la representación de grafos

TRABAJO DE FIN DE GRADO

Dirigido por Francesc Serratosa

Doble Grado de Biotecnología e Ingeniería Informática

UNIVERSITAT ROVIRA I VIRGILI

Tarragona 2023

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

ÍNDICE DE FIGURAS

1. Resumen

La proteína M-pro del COVID-19 es esencial para el proceso de infección del virus. Interrumpir el centro activo de esta enzima conduce a la pérdida de la función de esta y a la incapacidad del virus para infectar células. Por lo tanto, la búsqueda de ligandos que inhiban enzimáticamente esta proteasa es una estrategia importante para el desarrollo de fármacos contra el coronavirus.

Para seleccionar qué moléculas pueden inhibir efectivamente esta proteína, es importante conocer la actividad enzimática de los ligandos sobre la M-pro. Para lograr esto, se utiliza el algoritmo K-Nearest-Neighbor (KNN) junto con la métrica de distancia Graph Edit Distance (GED) para la predicción de la actividad enzimática de los ligandos. El método de representación de grafos permite representar los átomos y enlaces de la proteína M-pro del COVID-19 y los ligandos que interactúan con el sitio activo de la proteína, lo que facilita la comparación y selección de los ligandos más prometedores.

Sin embargo, es importante tener en cuenta que la actividad enzimática de los ligandos no es la única variable a considerar en la predicción de la efectividad de un fármaco contra el COVID-19. Otras variables como el pH y la temperatura también pueden afectar la capacidad de un ligando para inhibir la proteína M-pro. Por lo tanto, es necesario considerar estas variables en el diseño y selección de fármacos efectivos contra el virus.

2. Introducción

La enfermedad por coronavirus 2019 (COVID-19) es un brote infeccioso humano de la família del coronavirus desarrollado en Wuhan, China que posteriormente se propagó ràpidamente a otros países. El COVID-19 es una enfermedad infecciosa causada por el virus del síndrome respiratorio agudo severo coronavirus 2 (SARS-CoV-2). El COVID-19 ha infectado a 659 millones de personas y se ha cobrado la vida de 6.68 [1].

Las proteínas son moléculas grandes formadas por cadenas lineales de aminoácidos. Los aminoácidos, estructura básica de las proteínas, son moléculas orgánicas que contienen un grupo amino (-NH_2), un grupo carboxilo (-COOH), un átomo de hidrógeno y una cadena lateral específica para cada aminoácido (radical).

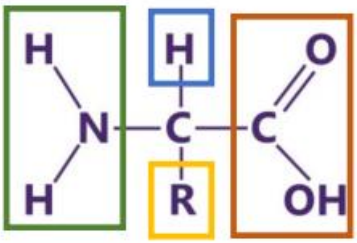

Figura 1. Formula general de un aminoácido

Dos aminoácidos se combinan a través de la reacción del grupo amino de un aminoácido y el grupo carboxilo de otro aminoácido para formar un péptido. Los péptidos se dividen en:

- Oligopeptido: El número de aminoácidos que forman la molécula está en el rango de 2 a 10.
- Polipéptido: El número de aminoácidos que forman la molécula es superior a 10 aminoácidos.
- Proteínas: El número de aminoácidos que forman la molécula es superior a 50 aminoácidos.

La unión entre dos aminoácidos se denomina enlace peptídico. A continuación, se muestra un ejemplo de oligopéptido formado por tres aminoácidos [2].

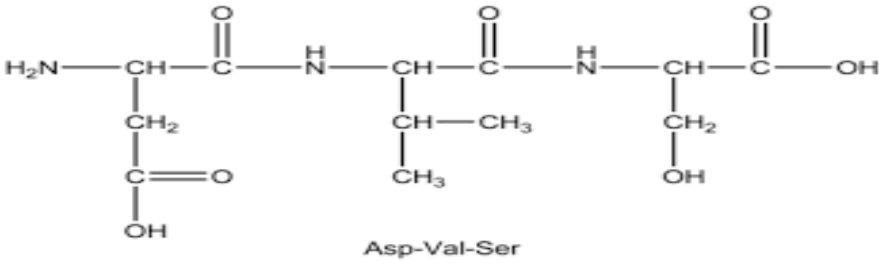

Figura 2. Ejemplo de un péptido formado por tres aminoácidos (tripéptido)

Las proteínas se dividen en cuatro niveles de estructuras: primaria, secundaria, terciaria y cuaternaria:

- La estructura primaria está formada por la secuencia de aminoácidos de la cadena polipeptídica.
- La estructura secundaria se constituye por el plegamiento que realiza la cadena polipeptídica por la formación de puentes de hidrógeno.
- La estructura terciaria o estructura tridimensional se forma cuando hay atracciones entre estructuras secundarias de la proteína. Las uniones que se dan en la estructura terciaria son enlaces covalentes y no covalentes. Los enlaces covalentes se forman cuando dos átomos no metálicos se unen.
- La estructura cuaternaria se constituye por la unión de más de una cadena polipeptídica con estructuras terciarias que quedan autoensambladas.

Las proteínas desempeñan funciones críticas en el cuerpo humano. Realizan la mayor parte del trabajo en las células y son necesarias para la estructura, función y regulación de los tejidos y órganos del cuerpo. Existen una gran variedad de funciones que poseen las proteínas. A continuación se muestran algunas de estas funciones:

Tabla 1. Funciones de las proteínas [3]

Función	Descripción	Ejemplos
Estructural	Brindan estructura y soporte a las células. Constituyen estructuras celulares.	Actina, colágeno e histonas.
Defensiva	Se unen a partículas extrañas al cuerpo para eliminarlas. Dentro de este grupo se encuentran los anticuerpos.	Inmunoglobulinas, fibrinógeno y trombina.
Hormonal	Transmiten señales entre las células.	Insulina, glucagón y calcitonina.
Transporte	Transportan átomos y moléculas pequeñas dentro de las células y por todo el cuerpo.	Hemoglobina y mioglobina.
Enzimática	Actúan como biocatalizadores de las reacciones químicas del metabolismo celular.	Lactasa y fosfotriosa isomerasa.

En cuanto al último tipo de proteínas comentadas en la tabla anterior, las enzimas son un tipo de proteínas que catalizan reacciones químicas, acelerando la velocidad de la reacción. En estas reacciones, las enzimas actúan sobre otras moléculas denominadas sustratos para convertirlos en un tipo diferente de moléculas llamadas productos. Las enzimas contienen un centro activo que es una zona específica para reconocer al sustrato que va a ser catalizado. Los inhibidores enzimáticos son moléculas capaces de unirse al centro activo de la enzima. De esa manera, el sustrato no es capaz de ensamblarse a la enzima y esta pierde su actividad. Algunos de los fármacos actuales en el mercado como la aspirina, funcionan como inhibidores enzimáticos uniéndose de manera covalente (irreversible) a la enzima objetivo.

Figura 3. Proceso de inhibición enzimática

Las proteasas son enzimas proteolíticas, es decir, rompen los enlaces peptídicos de las proteínas. Por consiguiente, la proteína objetivo se descompone en péptidos y aminoácidos. Las proteasas son importantes en muchos procesos biológicos, incluida la digestión, la respuesta inmunitaria y la regulación de la función de las proteínas. Algunos virus dependen de las proteasas para cumplir con sus ciclos reproductivos y poder sobrevivir. Por esta razón, es interesante estudiar los inhibidores de proteasas ya que parecen ser una apuesta prometedora para combatir contra patógenos infecciosos [4].

La proteína M-pro del COVID-19 es una proteasa que juega un papel fundamental en el inicio de la infección y su función está estrechamente relacionada la replicación y transcripción viral. Estas funciones son de vital importancia para la supervivencia del virus. La interrupción de la actividad catalítica de la M-pro parece ser una estrategia relevante para el desarrollo de fármacos contra el coronavirus [5].

La búsqueda de nuevas moléculas que puedan unirse a una proteína objetivo es uno de los retos esenciales en el descubrimiento de fármacos. Encontrar las interacciones entre proteínas y moléculas pequeñas es un proceso fundamental en la biología. Sin embargo el diseño de estos medicamentos es costoso en dinero y en tiempo. Para acabar con estas limitaciones se han comenzado a utilizar algunos métodos *in-sillico.* Existen nuevos métodos computacionales para la búsqueda de fármacos a través del acoplamiento proteína-ligando [1].

Los ligandos son moléculas pequeñas que se unen a proteínas para formar complejos proteicos. Algunos ejemplos de ligandos incluyen hormonas, neurotransmisores y medicamentos. La actividad de un ligando se refiere a como este interactúa con una proteína y como afecta a su función. Al unirse al ligando, la proteína puede cambiar su forma o conformación, lo que a su vez puede modificar su actividad enzimática y por lo tanto modificar su función.

Las herramientas computacionales predicen si un ligando en particular puede unirse a una proteína concreta. De esta manera, se pueden encontrar ligandos que se unen al centro activo de la M-pro e inhiban su actividad enzimática. Una manera de abordar la predicción de la actividad de los ligandos sobre las proteínas es mediante el uso del método de representación de grafos. En este enfoque, se representa cada ligando y proteína como un nodo de un grafo y se muestran las interacciones entre ellos como enlaces entre los nodos.

A partir de la representación de las proteínas y ligandos en grafos podemos predecir la actividad de los diferentes ligandos a través de algoritmos de aprendizaje automático. [6]

En el actual trabajo, se propone la predicción de la actividad enzimática de ligandos sobre estructuras tridimensionales de la proteína M-pro del COVID-19. Para ello, se ha utilizado el método de representación de grafos para mapear las proteínas y los ligandos en componentes. Más adelante, se calcula la distancia entre componentes a través de la medida "Graph Edit Distance" (GED). El algoritmo "K-Nearest-Neighbors" (KNN) utiliza esta distancia para predecir la actividad enzimática de los ligandos sobre la proteína M-pro.

Finalmente, se utilizó nuevamente el método de representación de grafos y GED para observar si existía una relación entre las actividades y las distancias entre componentes. De esta manera, se examina que los parámetros que ha empleado el algoritmo KNN sean válidos.

3. Descripción general del trabajo

La **motivación** de este trabajo viene dada por la búsqueda de nuevos fármacos para luchar contra la enfermedad del COVID-19 que nos ha afectado en los últimos años. Además, el proyecto realizado será de utilidad para el grupo de investigación de la universidad para futuros estudios relacionados con complejos ligandos-proteínas.

La **hipótesis** del trabajo es que el método de representación de grafos y el algoritmo KNN a través de la distancia calculada entre componentes por la medida GED permiten predecir la actividad enzimática de ligandos sobre estructuras tridimensionales de la proteína M-pro.

Para predecir esta actividad enzimática se utiliza el método de representación de grafos aplicado sobre la base de datos "M-pro database". Esta información nos permite representar en componentes las proteínas y ligandos correspondientes para realizar el cálculo de la distancia entre estas a través de la medida GED. Esta distancia nos permite predecir la actividad enzimática a través del algoritmo KNN.

Posteriormente, se compara las actividades predichas por el algoritmo KNN con las actividades que se encuentran en la base de datos para ver el margen de error. Por último, se utiliza el método de representación de grafos y GED para comprobar que se han utilizado parámetros válidos.

El **objetivo general** de este trabajo es la predicción de la actividad enzimática de los ligandos sobre la proteína M-pro. Estos valores nos proporcionan información sobre la capacidad inhibitoria de estos ligandos, por lo que posteriormente se puede realizar un estudio de selección de fármacos contra el COVID-19.

Dentro de los objetivos específicos del trabajo:

O1: Aplicación del algoritmo KNN, el método de representación de grafos y la medida GED para solucionar un problema biotecnológico.

O2: Aprendizaje y utilización de la plataforma de programación y cálculo numérico MATLAB para el desarrollo de los algoritmos necesarios para llevar a cabo el trabajo.

O3: Adquisición de experiencia en un entorno de trabajo. En concreto, mejorar habilidades de trabajo en equipo y capacidad resolutiva de problemas.

4. Metodología

A continuación, se van a explicar detalladamente los recursos obtenidos y las herramientas utilizadas para llevar a cabo el actual proyecto.

4.1 Base de datos M-pro

La base de datos M-pro fue obtenida del material suplementario del estudio "A review of the current landscape of SARS-CoV-2 main protease inhibitors: Have we hit the bullseye yet?"[6] realizado entre otros por los investigadores Gerard Pujadas y Santiago García de la Facultad de Bioquímica y biología molecular de nuestra universidad.

Esta base de datos contiene información sobre los ligandos con actividad enzimática sobre la proteína M-pro del COVID-19 y sobre los diferentes complejos de esta proteína. En concreto, la base de datos cuenta con un ligando para cada estructura de la proteína, es decir, se encuentran emparejados. Esta base contiene datos sobre los átomos y enlaces que forman las moléculas que nos permiten crear los grafos necesarios para llevar a cabo el cálculo de la distancia entre estos a través de la medida GED.

Los ficheros de la base de datos se encuentran divididos en dos partes diferenciadas. La primera parte contiene los ficheros en formato Mol2 de la proteína. La segunda parte contiene los ficheros en formato Mol2 de los ligandos y los ficheros de las actividades de los ligandos en formado SDF. Posteriormente se van a explicar los aspectos más relevantes de estos dos formatos.

Mol2 es un formato tabular de texto que contiene información sobre compuestos químicos, coordenadas atómicas, enlaces químicos y metadatos de una molécula [7]. Mol2 contiene múltiples secciones y cada una de estas proporciona una parte de la información de la molécula. Seguidamente se van a comentar algunas de estas secciones:

4.1.1 @<Tripos>MOLECULE

Cada registro de datos asociado con esta sección está formado por seis líneas de datos:

1. La primera línea se refiere al nombre de la molécula.
2. La segunda línea de datos contiene el número de átomos, enlaces, subestructuras, características y conjuntos asociados con la molécula.
3. La tercera línea de datos es el tipo de molécula.
4. La cuarta línea nos proporciona información acerca de las cargas asociadas a la molécula.
5. La quinta línea de datos contiene los bits de estado internos asociados con la molécula.
6. La última línea comprende cualquier comentario que pueda estar relacionado con la molécula.

3.1.2 @<Tripos>ATOM

Cada registro de datos asociado con esta sección está formado por una sola línea de datos. Esta línea de datos contiene toda la información necesaria de un átomo contenido dentro de la molécula. Esta línea está formada por los siguientes campos:

1. Atom_id: Número de identificación del átomo en el momento de creación del archivo.

2. Atom_name: nombre del átomo.
3. X: La coordenada x del átomo.
4. Y: La coordenada y del átomo.
5. Z: La coordenada z del átomo.
6. Atom_type: tipo de átomo.
7. Subst_id: número de identificación de la subestructura que contiene el átomo.
8. Subst_name: nombre de la subestructura que contiene el átomo.
9. Charge: Carga asociada al átomo.
10. Status_bit: Bits de estado internos asociados con el átomo.

3.1.3 @<Tripos>BOND

Similar a la sección anterior, cada registro de datos asociado en esta sección consta de una sola línea de datos que contiene toda la información necesaria de un enlace en la molécula. Esta línea está formada por los siguientes campos:

1. Bond_id: Número de identificación del enlace en el momento de creación del archivo.
2. Origin_atom_id: Número de identificación del átomo en un extremo del enlace.
3. Target_atom_id: Número de identificación del átomo en el otro extremo del enlace.
4. Bond_type: Se refiere al tipo de enlace.
    - 1 = simple.
    - 2 = doble.
    - 3 = triple.
    - am = amida
    - ar = aromático
    - un = desconocido
    - nc = no conectado.
5. Status_bits: Bits de estado internos asociados al enlace.

SDF es un formato que contiene información acerca de la estructura química de moléculas [8]. Almacena información sobre los átomos, los enlaces, la conectividad y las coordenadas de los átomos. Existe un campo concreto en este formato denominado “r_user_piC50” que nos indica la actividad enzimática real que ejerce el ligando sobre la proteína M-pro. IC50 es una medida que nos indica la concentración a la que un fármaco es capaz de inhibir un proceso biológico en un 50%. PIC50 es el logaritmo negativo del valor de IC50 [9].

A continuación, se muestra la visualización de un fichero del ligando en formato mol2 de la base de datos a través de la función *molviewer* que proporciona el Matlab:

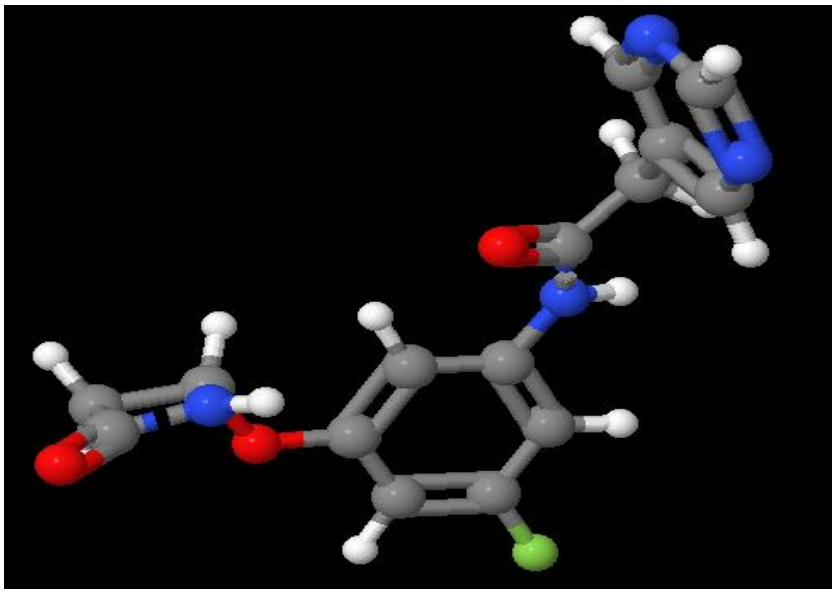

Figura 4. Representación de la molécula Mpro-x10387_ligand

4.2 Método de representación de grafos

De acuerdo con la definición de grafo, este se representa como G = (V, E) donde V es el conjunto de nodos, $V_i \in V$ denota un nodo en V y E es el conjunto de aristas, $E_{i,j}$ denota una arista entre el nodo V_i y V_j. Por último, N representa el número total de nodos del grafo y A es la matriz de adyacencia. [10]

El método de representación de generación de grafos permite calcular la distancia mínima entre grafos a través de la medida GED con la finalidad de predecir la actividad enzimática de los ligandos a través del algoritmo KNN. Para ello, este método permite representar los átomos y los enlaces de ligandos y proteínas en grafos:

- Grafo de ligando: Complejo formado por los átomos del ligando y sus enlaces entre ellos.
- Grafo de proteína. Complejo formado por los átomos de la proteína y sus enlaces entre ellos.
- Grafo de proteína-ligando: Complejo formado por los átomos del ligando, los enlaces entre átomos del ligando, algunos átomos seleccionados de la proteína, los enlaces entre los átomos seleccionados de la proteína y los enlaces entre átomos de la proteína y el ligando. Los átomos de la proteína que forman el complejo se seleccionan en base a una distancia límite entre el ligando y la proteína. Por último, los enlaces entre átomos de la proteína y el ligando son seleccionados en base a la distancia límite de un enlace no covalente. Los enlaces no covalentes normalmente se producen entre átomos que se encuentran a una distancia de entre 2 a 5 Angstroms.

Es interesante la generación de grafos únicamente formados por ligandos para observar como los posibles fármacos interactúan entre ellos. Además, este hecho proporciona una referencia para comparar con los resultados de los grafos formados por proteínas y ligandos. Finalmente, se propone la generación de grafos formados por el ligando al completo y parte de la proteína para probar diferentes partes del centro activo de esta.

A continuación se propone la representación de un grafo formado por el complejo de una proteína y el de un ligando. Los nodos azules denotan los átomos de la proteína mientras

que los nodos rojos denotan los átomos del ligando. A es la matriz de adyacencia que representa los enlaces del grafo.

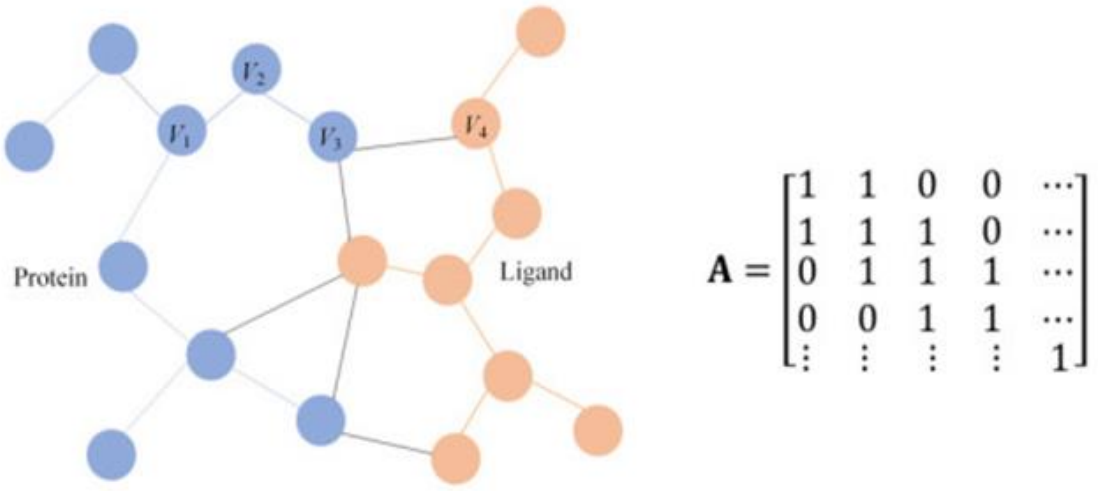

Figura 5. Representación de un complejo formado por una proteína y un ligando a través de un grafo

4.3 Graph Edit Distance (GED)

GED es una medida que permite calcular la distancia entre dos grafos con atributos a través del mínimo de modificaciones que se requieren para transformar un grafo en otro. Para ello, se definen estas modificaciones, que se denominan operaciones de edición. Las operaciones de edición son las siguientes: inserción, eliminación, sustitución de nodos y enlaces. De esta manera, para cada par de grafos (G_p y G_q) existe una ruta de edición *EditPath* (G_p, G_q) = ($\varepsilon_1,\ldots,\varepsilon_k$) (donde cada ε_i denota una operación de edición) que convierte un grafo en otro. Por cada par de grafos, existe un conjunto de rutas de edición, que se denomina $u_{,}$, que cada una de ellas transforma un grafo en otro [11].

La siguiente figura muestra una ruta de edición para transformar un grafo G_p en otro grafo G_q:

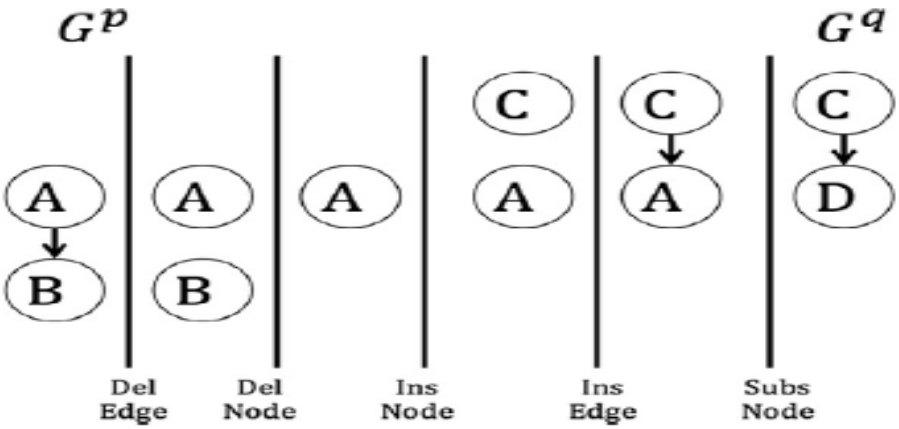

Figura 6. Ejemplo de ruta de edición que transforma Gp en Gq

Esta ruta de edición está compuesta por los siguientes pasos: eliminar enlace, eliminar nodo, insertar nodo, insertar enlace y sustituir nodo.

Dado estos dos grafos se define una función biyectiva [12] entre grafos $f_{p,q}$ que se relaciona con $EditPath\ (G_p, G_q) \in u$, con la finalidad de transformar un grafo en otro directamente. Para ello, se establece que la operación de sustitución representa una asignación de un nodo a otro. Además, las operación de deleción e inserción son transformadas en mapeos de nodos nulos en ambos grafos. Usando esta transformación, dado dos grafos G_p Y G_q y una función biyectiva entre sus nodos $f_{p,q}$ se define el coste de edición de un grafo como:

$$EditCost\ (G_p, G_q, f_{p,q}) = \sum Cvs + \sum Cvd + \sum Cvi + \sum Ces + \sum Ced + \sum Cei \quad (1)$$

Ecuación 1. Coste de edición de transformación de un grafo en otro

donde los diferentes costes son:

- Cvs: coste de sustituir un nodo de G_p por G_q.
- Cvd: coste de eliminar un nodo en G_p.
- Cvi: coste de insertar un nodo en G_q.
- Ces: coste de sustituir un enlace de G_p por G_q.
- Ced: coste de eliminar un enlace en G_p.
- Cei: coste de insertar un enlace en G_q.

Siguiendo con el ejemplo de la Fig. 5, el coste de edición de transformar un grafo en otro es: $editCost\ (G_p, G_q, f_{p,q}) = Ced + Cvd + Cvi + Cei + Cvs$.

Por último, el Graph Edit Distance (GED) es definido como el mínimo coste bajo cualquier posible biyección $f_{p,q}$:

$$EditDistance\ (G_p, G_q, f_{p,q}) = \min Edit\ Cost(G_p, G_q, f_{p,q}) \quad (2)$$

Ecuación 2. Graph Edit Distance (GED)

GED nos permite calcular la distancia que existe entre dos grafos cualquiera generados con la base de datos M-pro. De esta manera, se puede utilizar estas distancias para predecir la actividad enzimática de los ligandos utilizando el algoritmo KNN.

4.4 K-Nearest-Neighbors (KNN)

KNN es un método de clasificación supervisado que utiliza la proximidad para hacer clasificaciones o predicciones sobre nuevas muestras. Este algoritmo forma parte de la familia de modelos de "aprendizaje perezoso" ya que solo almacena el conjunto de entrenamiento en lugar de pasar por una fase de entrenamiento. El valor de k define la cantidad de puntos vecinos que se tendrán en cuenta para la clasificación o predicción de un punto de consulta específico. Para determinar el vecino más cercano se utilizan métricas de distancia, la más popular es la distancia euclidiana. [13]

Este algoritmo ha sido utilizado con la finalidad de solucionar un problema de regresión. Este fue identificado como un problema de regresión debido a que se intenta predecir valores de tipo continuo. En concreto, este problema se base en predecir la actividad enzimática de los ligandos de la base de datos M-pro. Para ello, se calcula la distancia entre todos los grafos a través de GED y se establece un valor de k para predecir el valor de la actividad enzimática del ligando escogido.

5. Diseño e Implementación

El proceso de implementación está formado por nueve pasos. Los tres primeros pasos tienen como objetivo extraer la información necesaria de la base de datos M-pro para generar los grafos. Los tres siguientes pasos sirven para comprobar la relación de los parámetros utilizados por el algoritmo KNN y para generar los grafos formados por complejos de proteínas y ligandos. Los tres últimos pasos tienen la finalidad de aplicar el algoritmo KNN sobre los datos extraídos de la base de datos.

Estos pasos fueron desarrollados a través del lenguaje de programación Matlab. Se desarrollaron los scripts y funciones necesarias para cumplir con los objetivos propuestos y obtener los resultados correspondientes. A continuación, se explican las decisiones de diseño y la implementación de los pasos (*Véase Anexos*) realizados detalladamente:

5.1 Step1_Activity_M_Pro_DB

Este script permite la lectura de las actividades enzimáticas de los ligandos. Este programa abre el directorio de la base de datos M-pro que contiene esta información a través de la función *dir*. Los ficheros de este directorio se encuentran en formato SDF.

Posteriormente, se van abriendo y tratando todos los ficheros a través de las funciones proporcionadas por el Matlab de gestión de ficheros (*fopen* y *fscanf*). Cada fichero con este formato contiene el campo "r_user_piC50" que es el que nos informa de la actividad enzimática de cada ligando. Por lo que, el algoritmo salta todos los campos hasta encontrar el campo de interés "r_user_piC50" para almacenarlo en una matriz de actividades denominada Activity.

Esta matriz posee 1 columna y tantas filas como el número de ligandos de la base de datos (107 ligandos). Cada posición de la matriz contiene la actividad enzimática asociada al ligando correspondiente.

Por último, se almacena la matriz Activity en un archivo denominado Activity.mat a través de la función *save* con la finalidad de poder utilizar esta matriz en los siguientes pasos del proceso.

5.2 Step2_ligan2graph

Este programa permite la generación de todos los grafos de los ligandos que contiene la base de datos M-pro. Para ello, se recorre el directorio que contiene los ficheros con la información de los ligandos en formato mol2 y se realiza la llamada a la función *ligand2graph*.

Esta función recibe por parámetro un fichero con la información de un ligando y devuelve una estructura denominada *Graph*. Esta estructura hace referencia a un grafo formado por los átomos y enlaces de un ligando. *Graph* está formada por cuatro matrices:

- Numnodes (1x1): número de átomos del ligando (nodos del grafo).
- Numedeges (1x1): número de enlaces entre los átomos del ligando (enlaces del grafo).
- Nodes (numatomos x 4): contiene en cada fila de la matriz cuatro campos. Los tres primeros campos son las coordenadas del átomo (x,y,z) y el último campo es el número atómico de este.

- Edges (numatomos x numatomos): hace referencia a la matriz de adyacencia del grafo. Los valores que puede contener esta matriz son del tipo de campo *bond_type* del formato mol2 comentado en el apartado anterior.

Para conseguir estas matrices, la función *ligand2graph* utiliza las funciones de gestión de ficheros para tratar los ficheros de la base de datos. Se recorren las secciones *@<Tripos>ATOM* y *@<Tripos>BOND* para extraer los valores de interés. Para generar el número atómico de la matriz Nodes, se desarrolló la función *Atomic_number*. Esta función recibe por parámetro el átomo en tipo carácter y devuelve el número atómico.

Finalmente, este script utiliza la función *save* para almacenar en el directorio correspondiente los grafos generados por la función *ligand2graph* en formato .mat.

5.3 Step3_protein2graph

Este script permite la generación de los grafos de las proteínas de la base de datos M-pro. La lógica del algoritmo es similar a la del paso anterior. El principal cambio de este algoritmo es que utiliza la función creada *protein2graph* para generar la estructura Graph. Por consiguiente, la estructura está formada por tantos nodos como átomos tenga cada proteína y tantos enlaces como enlaces exista entre los átomos de esta.

5.4 Step4_Generate_graphs_proteins_ligands

Este script permite la generación de todos los grafos formados por las parejas de proteínas y ligandos de la base de datos M-pro. Estos grafos son complejos formados por parte de la proteína y el ligando y permiten realizar el estudio de la predicción de la actividad enzimática del ligando. De esta manera, se pueden probar diferentes partes del centro activo de la proteína.

Para esto, el algoritmo utiliza los ficheros de ligandos y proteínas en formato .mat generados en los pasos anteriores para obtener las estructuras *Graph* a través de la función *load*. Para conseguir estos ficheros, se han de abrir los directorios correspondientes y comprobar que ambos ficheros (proteína y ligando) coinciden debido a que están emparejados.

Una vez obtenido las estructuras *Graph* para el ligando y la proteína, se realiza una llamada a la función *Generate_Graph* para generar el grafo formado por el complejo proteína-ligando. Esta función recibe por parámetro la matriz de nodos del ligando, la matriz de nodos de la proteína, la matriz de enlaces del ligando, la matriz de enlaces de la proteína, la distancia del rango de la proteína (thR) y la distancia límite del enlace no covalente (thC).

La primera distancia hace referencia al rango de distancia que se va a establecer para escoger una parte de la proteína. Por lo que, el nuevo grafo generado tendrá todos los átomos del ligando y parte de los átomos de la proteína. De esta manera, se pueden llevar a cabo diferentes pruebas con distintas partes de la proteína. La segunda distancia hace referencia a la distancia límite para que haya un enlace no covalente y por lo tanto un enlace entre la proteína y el ligando. Por consiguiente, la distancia entre un átomo de proteína y un átomo del ligando debe ser menor que la distancia límite del enlace no covalente para que se forme un enlace ellos.

La función *Generate_Graph* inicializa las matrices de nodos y enlaces del nuevo grafo a través de las matrices de los ligandos respectivas. Para crear la matriz de nodos del nuevo

grafo (nodesC), se calcula la distancia entre cada uno de los átomos de la proteína y el centro del ligando. Para este cálculo se utiliza la distancia euclidiana entre las tres coordenadas de ambos átomos a través de la función *pdist2*. Se incluye el átomo de la proteína inicial en la matriz de nodos del nuevo grafo si esta distancia es menor a la distancia del rango de la proteína (thR). Adicionalmente, en este paso del algoritmo se crea una matriz denominada posCP que almacena las posiciones de los átomos de la proteína seleccionados de la matriz inicial de nodos de la proteína y de nodesC para generar el cuarto cuadrante de la matriz de enlaces del nuevo grafo (edgesC).

Para la generación de la matriz de enlaces del nuevo grafo se crea una matriz de adyacencia formada por cuatro cuadrantes:

1º: se forma por los enlaces entre átomos de los ligandos.

2º: se forma por los enlaces entre los átomos de la proteína y el ligando.

3º: se forma por los enlaces entre los átomos del ligando y la proteína.

4º: se forma por los enlaces entre átomos de la proteína.

Tabla 2. Representación de la matriz de adyacencia del nuevo grafo generado por la función Generate_Graph

1º Ligando - Ligando	**2º** Ligando - Proteína
3º Proteína - Ligando	**4º** Proteína - Proteína

El primer cuadrante es inicializado con la matriz de adyacencia de los ligandos. Para conseguir el segundo y tercer cuadrante se itera sobre todos los átomos del ligando y sobre todos los átomos de la proteína previamente seleccionados. A medida que se va iterando se calcula la distancia euclidiana entre los átomos de ambas moléculas. Se incluye un enlace en este cuadrante si esta distancia es menor que la distancia límite del enlace no covalente (thC). El enlace se guarda como un valor de 3 dentro de la matriz de adyacencia. El cuarto cuadrante se consigue generando parejas entre los átomos de la proteína previamente seleccionados que están almacenados en la matriz posCP. Una vez generadas las parejas de átomos, se comprueba que haya un enlace entre ellos. En caso afirmativo, se utiliza la matriz posCP para conseguir la fila y columna de edgesC donde hay que actualizar el enlace entre átomos de la proteína seleccionados y se actualiza con el valor correspondiente.

Finalmente, se almacenan los nuevos grafos mat en un directorio nuevo llamado *proteins_ligands* a través de la función *save*.

5.5 Step5_Activity_distances

Este script crea una matriz llamada Activities para posteriormente observar gráficamente la relación entre la actividad enzimática de los ligandos y la distancia entre grafos. Esta matriz contiene en la primera columna la distancia entre las actividades de los ligandos y en la segunda columna la distancia entre grafos de ligandos o entre grafos de ligandos-

proteínas. La matriz contiene tantas filas como número de parejas de ligandos o número de parejas de ligandos y proteínas hay en la base de datos.

Para la generación de los valores de la distancia entre actividades se carga el fichero Activity.mat a través de la función *load* para obtener la matriz Activity. Seguidamente, se itera sobre esta matriz para realizar la diferencia entre actividades y se actualiza este valor en la matriz Activities.

Para la obtención de los valores de distancia entre grafos es necesario abrir el fichero correspondiente dependiendo si se quiere calcular la distancia entre grafos formados por ligandos o entre grafos formados por ligandos y proteínas. Posteriormente, se itera sobre el directorio y se van almacenado las estructuras de los grafos de las correspondientes parejas. Por último, se utiliza la función *GED* para calcular la distancia entre grafos. Esta función recibe por parámetro la matriz de nodos del primer grafo, la matriz de nodos del segundo grafo, la matriz de enlaces del primer grafo, la matriz de enlaces del segundo grafo, el coste de delecion e inserción de un nodo y el coste de deleción e inserción de un enlace. *GED* devuelve la distancia entre los dos grafos que se actualiza en la matriz Activities. El último paso del algoritmo almacena la matriz Activities en un fichero Activities.mat para que pueda ser cargado en el siguiente paso del proceso.

5.6 Step6_ ViewPlot

Este script permite representar gráficamente la distancia entre actividades de los ligandos y la distancia entre grafos de ligandos o ligandos-proteínas. Esta representación permite conocer la relación entre estas dos variables. Es de importancia conocer esta relación debido a que la distancia entre grafos se utiliza para predecir la actividad enzimática de los ligandos por el algoritmo KNN.

Para la representación de las dos variables se utiliza la matriz Activities del fichero Activities.mat. Se realiza una llamada a la función *plot* que recibe por parámetro los valores de los ejes de la gráfica. Finalmente, se definen los títulos de los ejes y la gráfica.

5.7 Step7_Generate_matrix_distance

Este script genera una matriz de distancias entre ligandos o entre ligandos y proteínas. Esta matriz es de utilidad en el siguiente paso del proceso para seleccionar las distancias mínimas que existen entre estas moléculas. Estas distancias permiten predecir la actividad de los ligandos. La matriz contiene tantas filas y columnas como ligandos o complejos de proteínas-ligandos hay en la base de datos.

Por ello, se selecciona el directorio correspondiente dependiendo si solo se quieren utilizar ligandos o ligandos y proteínas para calcular la distancia. A medida que se recorre este directorio se van cargando los grafos almacenados gracias a los pasos anteriores. Se utiliza la función *GED* para el cálculo de la distancia entre todos los grafos. Se actualizan estos valores en la matriz de distancias. Si ambos grafos seleccionados son iguales, se actualiza la posición de la matriz con un valor elevado (200) para que no interfiera en el resultado final. Para finalizar se guarda la matriz de distancias en un fichero distances.mat a través de la función *save*.

5.8 Step8_Test_K_Nearest_Neighbor

Este script predice todas las actividades enzimáticas de los ligandos a través de la matriz de distancias. Concretamente el programa genera una matriz denominada

actividades_minimas con tantas filas como ligandos o complejos proteínas-ligandos hay en la base de datos y dos columnas. La primera columna hace referencia a las actividades reales de los ligandos y la segunda a las actividades predichas.

Para conseguir la primera columna se carga la matriz Activity a través del fichero Activity.mat generado en el primer paso del proceso. La segunda columna se obtiene tras la llamada a la función creada *K_nearest_neighbor* para todas las moléculas.

Esta función recibe por parámetro la posición de la molécula y devuelve la actividad enzimática del ligando en base al vecino más cercano (K=1). Para ello, la función carga las matrices distances y Activity de los ficheros correspondientes. Se inicializa una variable con el valor de la primera molécula que indica la distancia mínima y otra variable con la posición de esta molécula. Se recorre la lista de distancias y se va actualizando estas variables respecto al vecino más cercano. Para finalizar se accede a la matriz Activity con la posición del vecino más cercano para conseguir la actividad predicha.

Por último, el algoritmo almacena la matriz actividades_minimas en un fichero para poder ser representada gráficamente.

5.9 Step9_ViewPlot_K_nearest_neighbor

Este script permite representar gráficamente las actividades predichas y las actividades reales para compararlas. El eje x de la gráfica corresponde a las actividades reales y el eje y a las actividades predichas. Ambas actividades se extraen de la matriz de actividades_minimas.

6. Resultados

La predicción de la actividad enzimática se llevó a cabo a través de pruebas sobre diferentes complejos. Para los complejos formados por proteínas y ligandos hay que seleccionar los valores de las distancias límites (thR y thC). Además, estas predicciones se realizaron en base al vecino más cercano (K=1) y se utilizaron valores constantes para el coste de inserción y deleción de nodos y enlaces para la función GED. Para la obtención de los resultados, se ejecutaron los scripts necesarios varias veces. Las pruebas que se realizaron se muestran en la siguiente tabla:

Tabla 3. Pruebas realizadas para la predicción de la actividad enzimática de los ligandos

K = 1	Ligando-Ligando	Proteína-Ligando	Proteína-Ligando	Proteína-Ligando
thC	-	3	4	5
thR	-	5	6	7

A continuación se muestra la relación entre actividades enzimáticas reales y predichas que se consiguieron tras haber ejecutado los scripts sobre los grafos formados por ligandos:

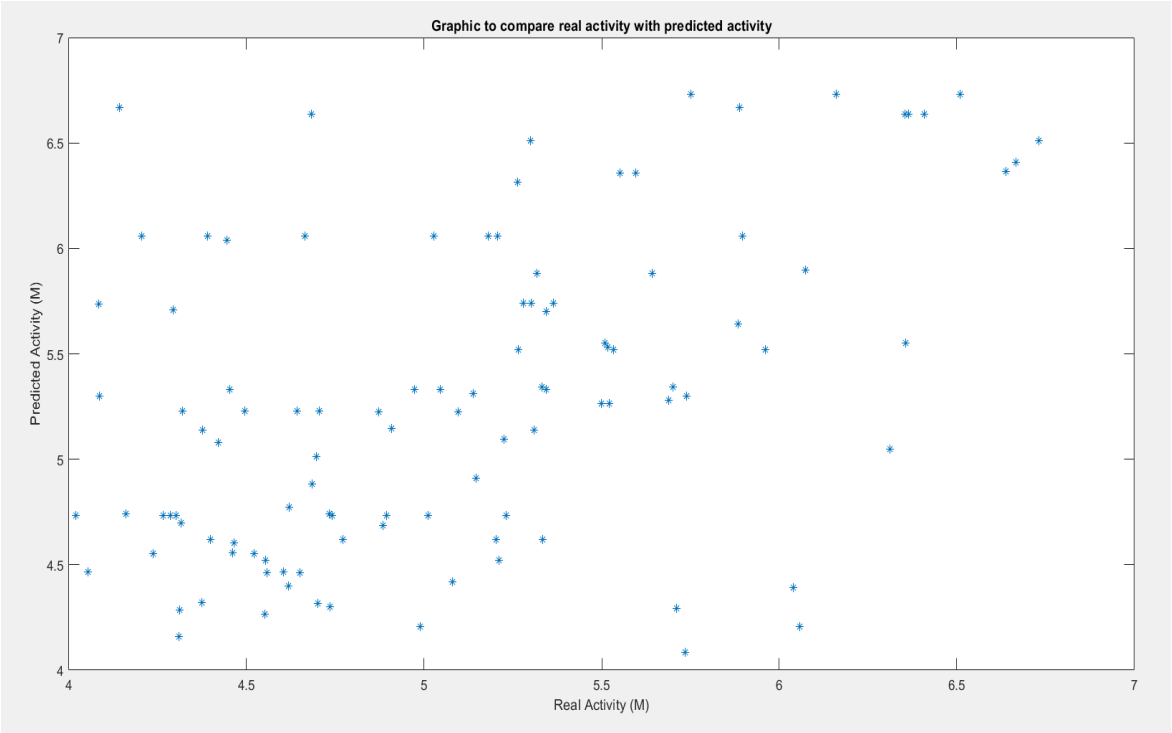

Figura 7. Relación entre actividad enzimática real y predicha por grafos formados por ligandos

Los valores de la gráfica no siguen una línea recta por lo que no existe una similitud entre la actividad real y la actividad predicha. Esto significa que no hay resultados significativos en la predicción de la actividad enzimática en grafos formados por ligandos. Se comprobó la relación entre la distancia mínima entre grafos y la actividad enzimática. En la siguiente gráfica se representa la distancia entre grafos formados por ligandos frente a la diferencia entre actividades.

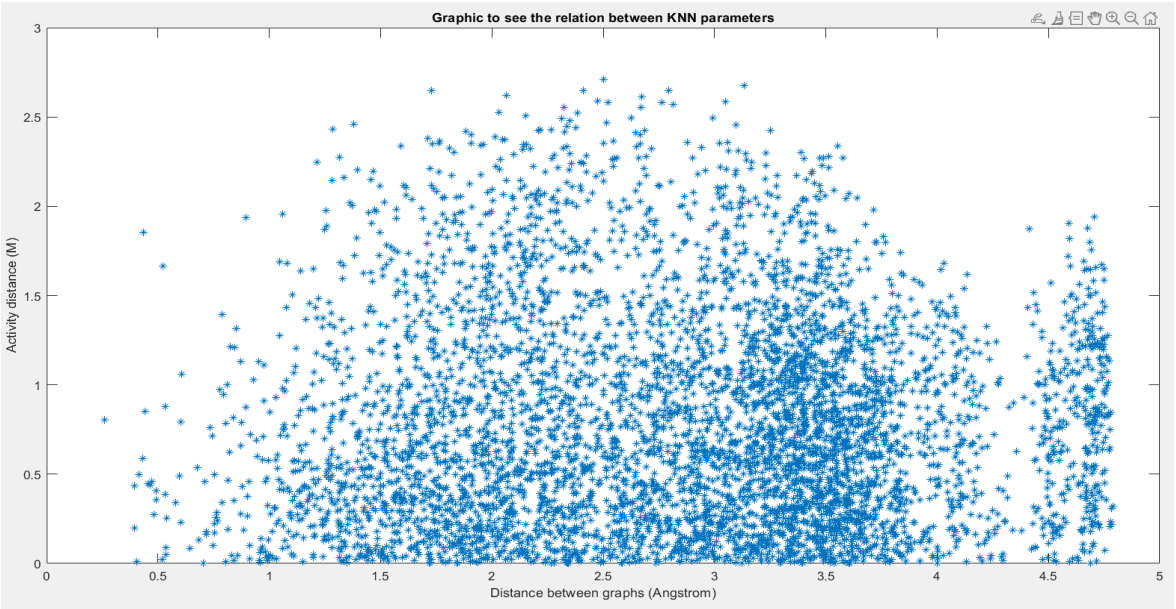

Figura 8. Relación entre distancia de actividades y distancia entre grafos formados por ligandos

La gráfica no sigue ningún patrón en particular por lo que no existe ninguna relación entre las distancias calculadas por GED entre grafos formados por ligandos y las actividades enzimáticas de ligandos. Este hecho es debido a que para la predicción de la actividad

enzimática de los ligandos presuntamente es necesario tomar parte del centro activo de la proteína sobre la que actúan.

Con el objetivo de cumplir esta premisa, se llevó a cabo el mismo estudio tomando parte de la proteína y generando grafos formados por todos los átomos del ligando y algunos átomos de la proteína. Se realizaron pruebas sobre diferentes distancias límites del rango de la proteína (thR) con el objetivo de coger diferente cantidad de átomos de esta para la creación del nuevo grafo. También, se realizaron pruebas sobre diferentes distancia límites de enlace no covalente (thC) con la finalidad de crear distintos enlaces entre átomos del ligando y átomos de la proteína y crear grafos diferentes.

La siguiente gráfica muestra la relación entre las actividades enzimáticas reales y predichas por grafos formados por ligandos y parte de la proteína. Se estableció un thC de 3 y un thR de 5.

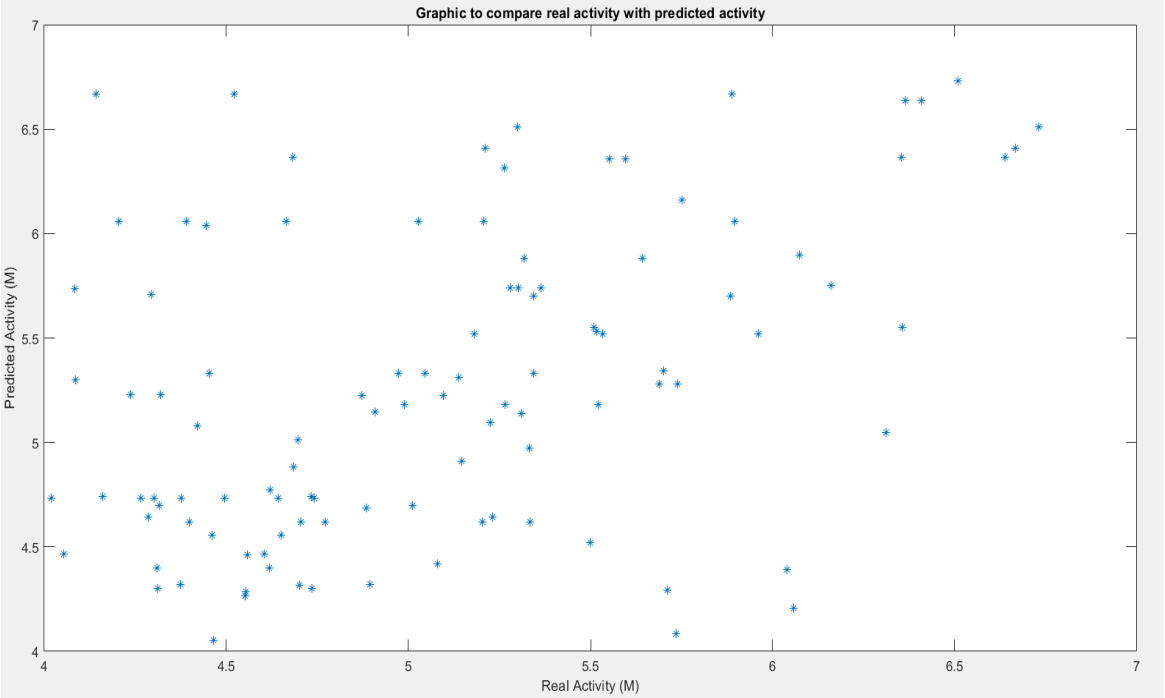

Figura 9. Relación entre actividad enzimática real y predicha por grafos formados por ligandos y parte de la proteína (thC = 3; thR = 5).

Se puede observar que no hay una línea recta por lo que existe una diferencia entre la distancia real y la distancia predicha. Este hecho significa que no hay resultados significativos en la predicción de la actividad enzimática con la distancia a través de grafos formados por ligandos y proteínas con este tipo de *threshold*. De nuevo, se observa la relación entre este parámetro y la actividad enzimática a través de la siguiente gráfica:

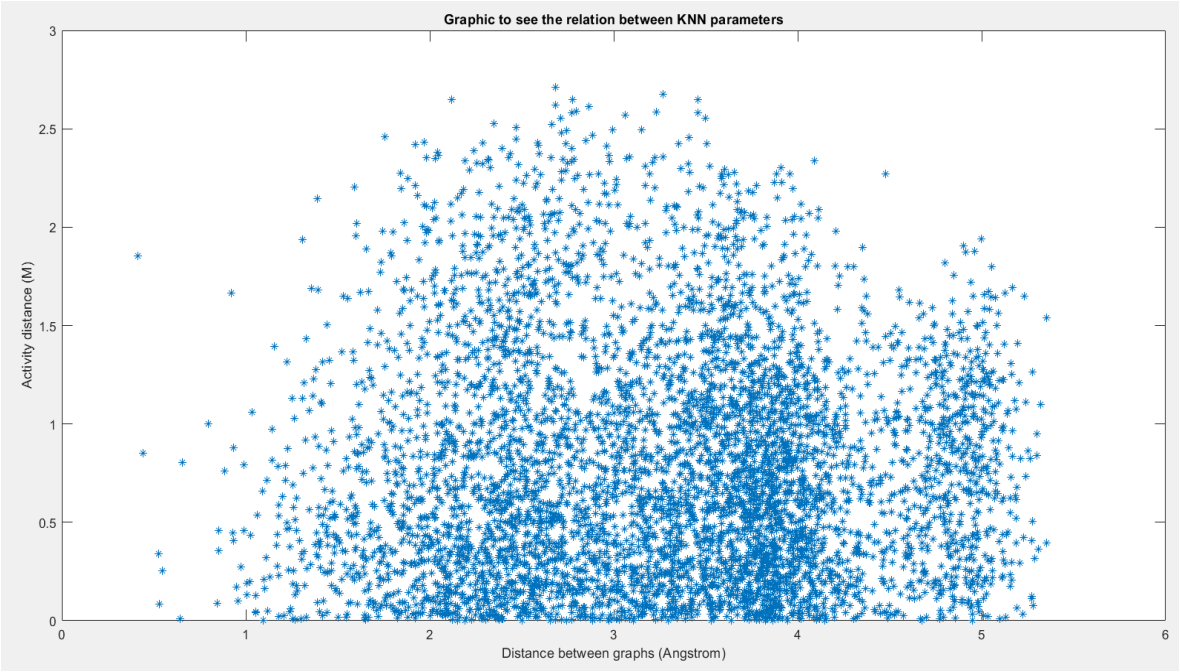

Figura 10. Relación entre distancia de actividades y distancia entre grafos formados por ligandos y proteínas (thC = 3; thR = 5).

No existe una relación entre la actividad y la distancia entre grafos formados por ligandos y proteínas debido a que no hay ningún patrón entre estas dos variables en la figura. Con la finalidad de contrastar estos resultados con grafos con una mayor cantidad de átomos de proteína se llevaron a cabo pruebas aumentando las distancias. La siguiente gráfica muestra la relación entre las actividades enzimáticas reales y predichas por grafos formados por ligandos y parte de la proteína dónde se estableció un thC de 4 y un thR de 6.

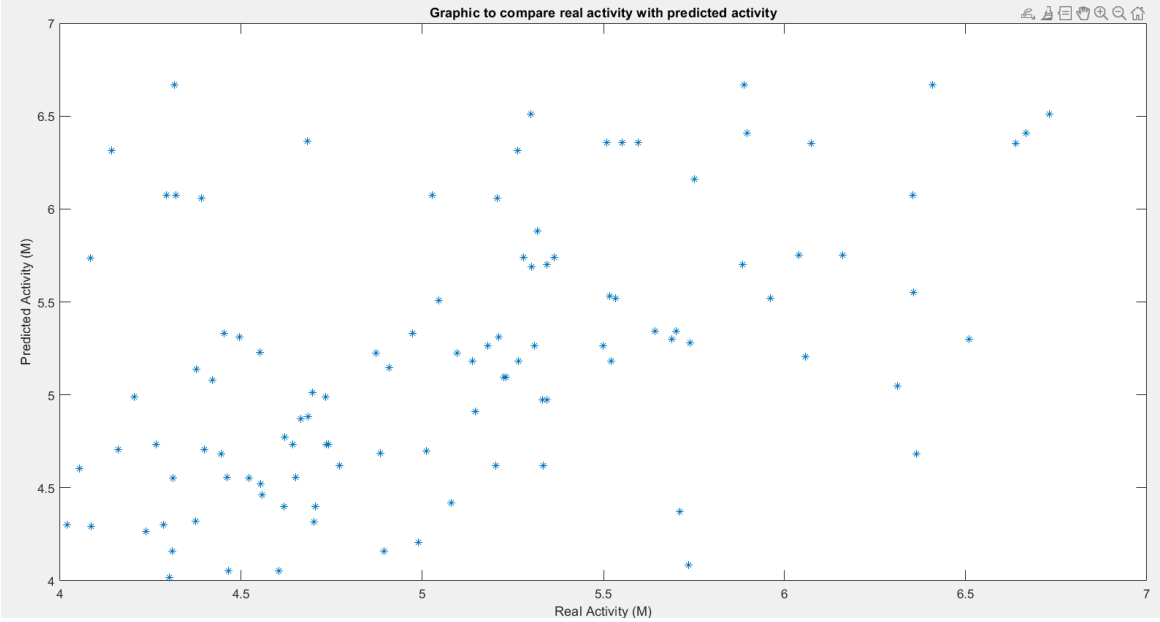

Figura 11. Relación entre actividad enzimática real y predicha por grafos formados por ligandos y parte de la proteína (thC = 4; thR = 6).

Tampoco existe una coincidencia entre las actividades estableciendo este tipo de *threshold*. Seguidamente, se realizó el mismo estudio que en los pasos anteriores para observar la relación entre los parámetros.

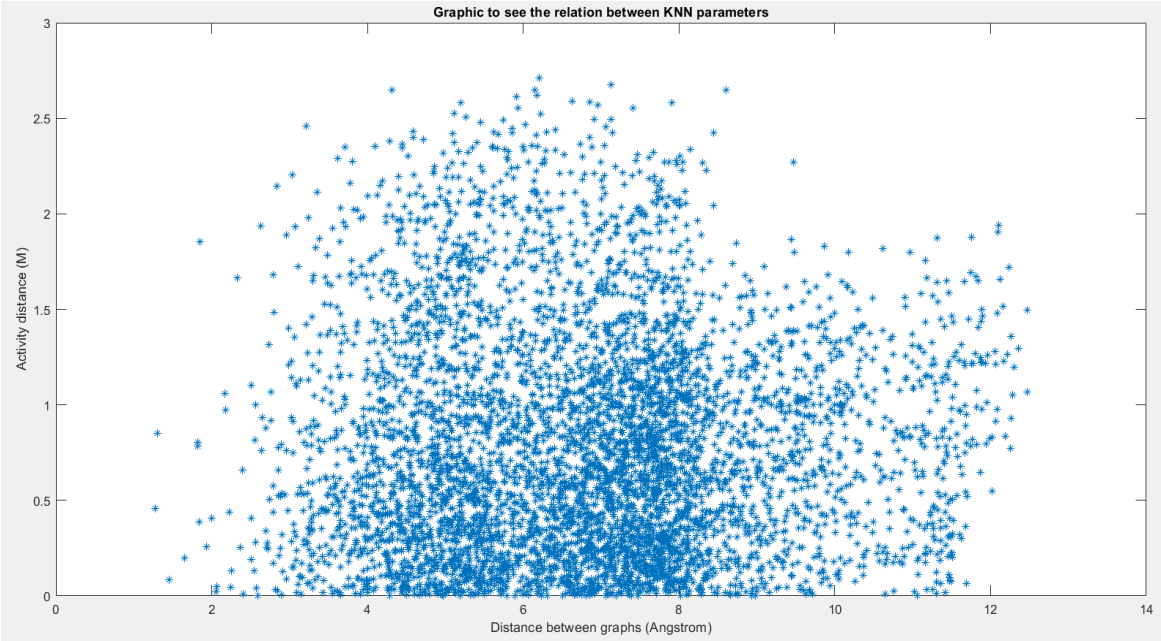

Figura 12. Relación entre distancia de actividades y distancia entre grafos formados por ligandos y proteínas (thC = 4; thR = 6).

Como se puede observar en la figura, en este caso tampoco hay un patrón por lo que no existe una relación clara entre ambos parámetros. Por último, se llevó a cabo otra prueba aumentando el rango de las distancias a thC a 5 y thR a 7.

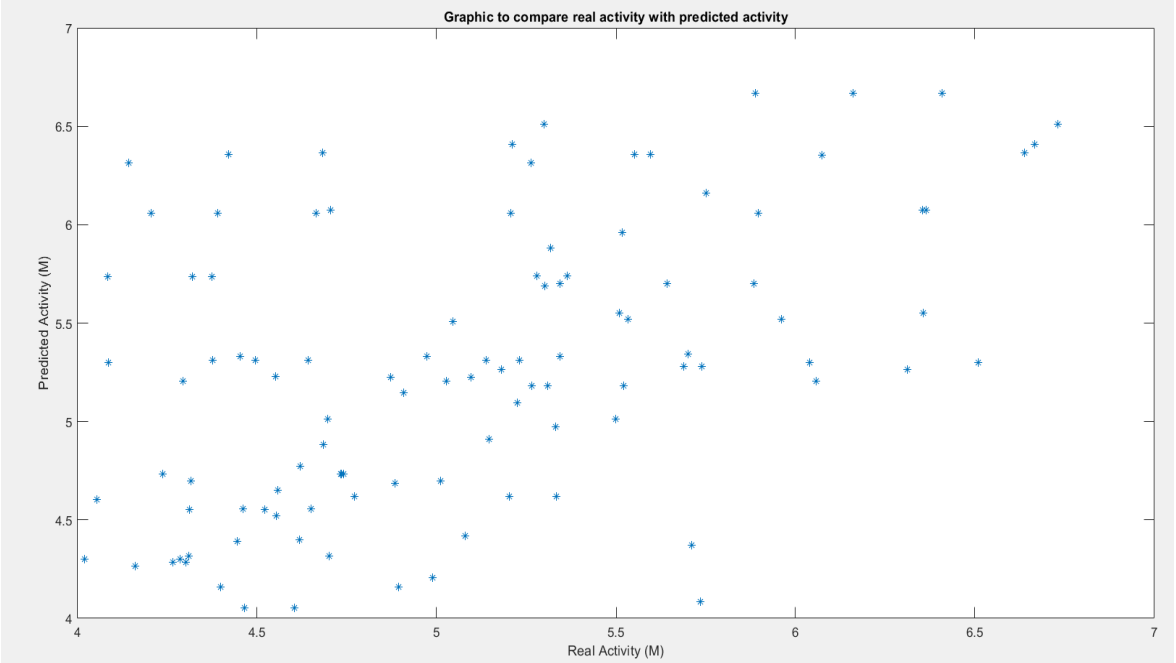

Figura 13. Relación entre actividad enzimática real y predicha por grafos formados por ligandos y parte de la proteína (thC = 5; thR = 7).

Finalmente, no existe tampoco una similitud entre las actividades teniendo en cuenta estos valores para las distancias. La siguiente figura muestra la relación entre los parámetros:

Como se ha observado a través de las figuras, no existen resultados significativos. La distancia entre grafos calculada por GED no es suficiente para la predicción de la actividad

enzimática de los ligandos. Se han realizado pruebas para grafos formados por solo ligandos y para grafos formados por ligandos y proteínas con diferentes tamaños. Ninguna de las pruebas es válida para demostrar que con la distancia entre grafos calculada por GED se puede predecir la actividad enzimática.

Esto se debe a que existen otros factores que afectan al proceso de actividad enzimática. Dentro de estos factores se encuentran la concentración de la enzima y el sustrato, el pH, la temperatura y los moduladores enzimáticos. Estos factores impactan sobre la actividad enzimática debido a que modifican la estructura terciaria de la enzima. Esto desnaturaliza la proteína y modifica la velocidad de la reacción. De esta manera, la enzima no funciona correctamente. Por lo que, para llevar a cabo un estudio sobre la predicción de la actividad enzimática de posibles fármacos habría que tener en cuenta este tipo de factores [14].

7. Conclusión

En conclusión, el presente TFG abordó la predicción de la actividad enzimática de ligandos que interactúan con la proteína M-pro del SARS-CoV-2, utilizando el algoritmo de k-Nearest-Neighbors y la distancia entre grafos calculada por la medida Graph Edit Distance. Los resultados obtenidos no mostraron una correspondencia significativa entre las actividades enzimáticas reales y las predichas por el modelo.

En particular, se demostró que la distancia entre grafos no es suficiente para predecir la actividad enzimática de ligandos, ya que otros factores, como el pH y la temperatura, pueden tener un impacto significativo en la actividad enzimática. Esto destaca la importancia de considerar múltiples variables en futuros estudios para mejorar la precisión de los modelos predictivos.

Cabe destacar que, en el presente estudio, se evaluó el efecto de diferentes tipos de *threshold* de enlaces no covalentes y de rango de la proteína en la actividad enzimática predicha. A pesar de estas exploraciones, no se observó una correspondencia significativa entre las actividades enzimáticas reales y las predichas por el modelo en ningún caso. Esto sugiere que la distancia entre grafos por sí sola no es suficiente para predecir la actividad enzimática de ligandos que interactúan con la proteína M-pro, y que se deben considerar otros factores adicionales para mejorar la precisión del modelo.

A pesar de que los resultados no fueron los esperados, el presente trabajo ofrece una base sólida para futuras investigaciones en esta área. Se han utilizado herramientas computacionales avanzadas para analizar la actividad enzimática y se ha identificado un área de mejora para la predicción de la actividad enzimática. En consecuencia, los resultados de este estudio proporcionan información valiosa para investigaciones futuras en la identificación de los factores que influyen en la actividad enzimática de los ligandos que interactúan con la proteína M-pro.

En última instancia, esta investigación puede tener importantes implicaciones para el desarrollo de terapias efectivas contra enfermedades infecciosas, ya que la proteína M-pro es un objetivo terapéutico potencial en la lucha contra el SARS-CoV-2. Al comprender mejor los factores que afectan la actividad enzimática de los ligandos que interactúan con la proteína M-pro, los científicos pueden desarrollar mejores enfoques para el diseño de fármacos y optimizar los tratamientos para pacientes con enfermedades infecciosas.

8. Perspectiva de futuro

En el futuro, se espera que el aprendizaje automático desempeñe un papel fundamental en la predicción de la actividad enzimática de ligandos que actúan sobre la proteína M-pro del COVID-19. Una posible solución a la conclusión de que los costos de edición del Graph Edit Distance (GED) no permiten predecir la actividad enzimática debido a otros factores como el pH y la temperatura, podría ser la implementación de algoritmos de aprendizaje automático para aprender los costos de edición del GED.

En particular, los algoritmos de aprendizaje automático pueden aprender los costos de edición del GED de forma automática a partir de un conjunto de datos de ligandos y sus respectivas actividades enzimáticas conocidas, lo que permitiría una comparación más precisa de los ligandos y una mejor predicción de su actividad enzimática. Además, estos modelos pueden considerar otros factores para lograr una mayor precisión en la predicción de la actividad enzimática.

Otra perspectiva de futuro prometedora para mejorar la predicción de la actividad enzimática de los ligandos es la utilización de técnicas de “Graph Embedding”. Al pasar los grafos a vectores mediante esta técnica, se pueden utilizar algoritmos de aprendizaje automático tradicionales para predecir la actividad enzimática de los ligandos. Además, al utilizar esta técnica, se puede obtener una mayor precisión en la predicción de la actividad enzimática, ya que la información estructural de los grafos complejos se mantiene en la representación vectorial.

En combinación con los algoritmos de aprendizaje automático que aprenden los costos de edición del GED, el uso de técnicas de “Graph Embedding” puede llevar a una mayor precisión y eficiencia en la predicción de la actividad enzimática de los ligandos que actúan sobre la proteína M-pro del COVID-19

9. Bibliografía

[1] M. Abbasi and H. Sadeghi-Aliabadi, “An In-silico Screening Strategy to the Prediction of New Inhibitors of COVID-19 Mpro Protein,” *Iran. J. Pharm. Res.*, vol. 20, no. 4, pp. 125–136, 2021, doi: 10.22037/ijpr.2021.114997.15146.

[2] Wikipedia La Enciclopedia libre, “Proteína.” https://es.wikipedia.org/wiki/Proteína (accessed Jan. 14, 2023).

[3] Universidad de Murcia, “Funciones de las proteínas.” https://www.um.es/molecula/prot07.htm (accessed Jan. 16, 2023).

[4] A. A. Agbowuro, W. M. Huston, A. B. Gamble, and J. D. A. Tyndall, “Proteases and protease inhibitors in infectious diseases,” *Med. Res. Rev.*, vol. 38, no. 4, pp. 1295–1331, 2018, doi: 10.1002/med.21475.

[5] A. Citarella, A. Scala, A. Piperno, and N. Micale, “Sars-cov-2 mpro: A potential target for peptidomimetics and small-molecule inhibitors,” *Biomolecules*, vol. 11, no. 4, 2021, doi: 10.3390/biom11040607.

[6] G. Macip, P. Garcia-segura, J. Mestres-truyol, B. Saldivar-espinoza, G. Pujadas, and S. Garcia-Vallvé, “A review of the current landscape of SARS-CoV-2 main protease inhibitors: Have we hit the bullseye yet?,” *Int. J. Mol. Sci.*, vol. 23, no. 1, 2022, doi: 10.3390/ijms23010259.

[7] M. Kumar, “Explicación del formato de archivo Mol2.” https://chemicbook.com/2021/02/20/mol2-file-format-explained-for-beginners-part-2.html#:~:text=It is a plain text,and metadata of a molecule. (accessed Jan. 20, 2023).

[8] LifeChemicals, “Cómo trabajar con archivos de datos estructurados (archivos SDF).” https://lifechemicals.com/order-and-supply/how-to-work-with-sd-files (accessed Jan. 20, 2023).

[9] CDDVAULT, “What is pIC50?” https://www.collaborativedrug.com/es/what-is-pic50-2/#:~:text=En pocas palabras%2C pIC50 es,es un pIC50 de 9. (accessed Jan. 20, 2023).

[10] H. Yuan, J. Huang, and J. Li, “Protein-ligand binding affinity prediction model based on graph attention network,” *Math. Biosci. Eng.*, vol. 18, no. 6, pp. 9148–9162, 2021, doi: 10.3934/mbe.2021451.

[11] F. Serratosa, “Fast computation of Bipartite graph matching,” *Pattern Recognit. Lett.*, vol. 45, pp. 244–250, 2014, doi: 10.1016/j.patrec.2014.04.015.

[12] W. L. E. Libre, “Función biyectiva.” https://es.wikipedia.org/wiki/Función_biyectiva (accessed Jan. 24, 2023).

[13] IBM, “¿Qué es el algoritmo de k vecinos más próximos?” https://www.ibm.com/es-es/topics/knn (accessed Dec. 20, 2022).

[14] StudySmarter, “Factores que afectan a la actividad enzimática.”

https://www.studysmarter.es/resumenes/biologia/base-molecular-y-fisicoquimica-de-la-vida/factores-que-afectan-a-la-actividad-enzimatica/#:~:text=Inhibición enzimática no competitiva,sustrato se una a él. (accessed Jan. 15, 2023).

10.Anexos

10.1 Step1_Activity_M_Pro_DB

```
% This scripts allows to generate all of the activities from ligand
% database
clc, clear, close all
pathname='./02-ligands-coordinates/01-SDF-format/';
file=strcat(pathname,'*.sdf');
d=dir(file);
j=1;
for i=1:length(d)
    filename=strcat(pathname,"/",d(i).name);
    fileID=fopen(filename,'r');
    if fileID >= 0
        name=fscanf(fileID,'%s',1);
        nothingtodo=fscanf(fileID,'%s',1);
        while(nothingtodo~="<r_user_piC50>")
            nothingtodo=fscanf(fileID,'%s',1);
        end
        num=fscanf(fileID,'%f',1);
        Activity(j,1)=num;
        j=j+1;
    end
end
save('Activity','Activity')
```

10.2 Step2_ligan2graph

```
%This script generates a matlab file with a graph given a file with mol2
format.
clc;clear all;close all
%% Test Ligand
%Reading
d = uigetdir;
d=dir(strcat(d,'/*.mol2'));
for i=1:length(d)
    name=d(i).name;
    pathname=d(i).folder;
    file=strcat(pathname,'/',name);
    [error,graph]=ligand2graph(file);
    if ~error
        graph_file=file(1:end-5);
        save(graph_file,'graph');
        disp('Reading Done')
    else
        disp('Reading Error')
    end
end
```

10.3 ligand2graph

```
% This function generates a ligand graph through reading a mol2 file
function [error,Graph]=ligand2graph(filename)
   fileID = fopen(filename,'r');
   if fileID >=0
```

```
    nothingtodo=fscanf(fileID,'%s',2);
    Graph.numnodes=fscanf(fileID,'%d',1);
    Graph.numedges=fscanf(fileID,'%d',1);
    nothingtodo=fscanf(fileID,'%d',1);
    nothingtodo=fscanf(fileID,'%s',4);
    Graph.Nodes = zeros(Graph.numnodes,4,'double');
    for i=1:Graph.numnodes
        nothingtodo=fscanf(fileID,'%d',1)
        atom=fscanf(fileID,'%s', 1);
        atomic_number=Atomic_number(atom(1));
        Graph.Nodes(i,4)=atomic_number;
        Graph.Nodes(i,1:3)=fscanf(fileID,'%f',3);
        ignoredLine=fgetl(fileID);
    end
    mat=zeros(Graph.numnodes,Graph.numnodes,'int8');
    for i=1:Graph.numnodes
        nothingtodo=fscanf(fileID,'%d',1);
        colTwo=fscanf(fileID,'%d',1);
        colThree=fscanf(fileID,'%d',1);
        colFour=fscanf(fileID,'%s',1);
        if(colFour=="ar")
            colFour=101;
        elseif(colFour=="am")
            colFour=102;
        else
            colFour=str2num(colFour);
        end
        mat(colTwo,colThree)=colFour;
        mat(colThree,colTwo)=colFour;
    end
    Graph.Edges=mat;
    fclose(fileID);
    end
    error=fileID<0;
end
```

10.4 Atomic_number

```
function atomic_number = Atomic_number(atom)
    switch atom
    case 'C'
        atomic_number = 6;
    case 'H'
       atomic_number = 1;
    case 'O'
       atomic_number = 8;
    case 'N'
       atomic_number = 7;
    case 'S'
       atomic_number = 16;
    case 'F'
        atomic_number = 9;
    case 'I'
        atomic_number = 53;
    case 'B'
        atomic_number = 5;
    otherwise
        disp(atom);
```

```
    end
end
```

10.5 Step3_protein2graph

```
    %This script generates a matlab file with a graph given a file with mol2
    format.
    clc;clear all;close all
    %% Test Protein
    %Reading
    d = uigetdir;
    d=dir(strcat(d,'/*.mol2'));
    for i=1:length(d)
        name=d(i).name;
        pathname=d(i).folder;
        file=strcat(pathname,'/',name);
        [error,graph]=protein2graph(file);
        if ~error
            graph_file=file(1:end-5);
            save(graph_file,'graph');
            disp('Reading Done')
        else
            disp('Reading Error')
        end
    end
```

10.6 protein2graph

```
% This function generates a protein graph through reading a mol2 file
function [error,Graph]=protein2graph(filename)
   fileID = fopen(filename,'r');
   if fileID >=0
   nothingtodo=fscanf(fileID,'%s',2);
   Graph.numnodes=fscanf(fileID,'%d',1);
   Graph.numedges=fscanf(fileID,'%d',1);
   nothingtodo=fscanf(fileID,'%d',1);
   nothingtodo=fscanf(fileID,'%s',7);
   Graph.Nodes = zeros(Graph.numnodes,4,'double');
   for i=1:Graph.numnodes
       nothingtodo=fscanf(fileID,'%d',1)
       atom=fscanf(fileID,'%s', 1);
       atomic_number=Atomic_number(atom(1));
       Graph.Nodes(i,4)=atomic_number;
       Graph.Nodes(i,1:3)=fscanf(fileID,'%f',3);
       ignoredLine=fgetl(fileID);
   end
   mat=zeros(Graph.numnodes,Graph.numnodes,'int8');
   for i=1:Graph.numnodes
       nothingtodo=fscanf(fileID,'%d',1);
       colTwo=fscanf(fileID,'%d',1);
       colThree=fscanf(fileID,'%d',1);
       colFour=fscanf(fileID,'%s',1);
       if(colFour=="ar")
           colFour=101;
       elseif(colFour=="am")
           colFour=102;
       else
           colFour=str2num(colFour);
       end
```

```
        mat(colTwo,colThree)=colFour;
        mat(colThree,colTwo)=colFour;
    end
    Graph.Edges=mat;
    fclose(fileID);
    end
    error=fileID<0;
end
```

10.7 Step4_Generate_graphs_proteins_ligands

```
% This script generates ligands and proteins graphs ( firstly it must
% select ligand directory and secondly it must select protein directory )
d = uigetdir;
d = dir(strcat(d,'/*.mat'));
d2 = uigetdir;
d2 = dir(strcat(d2,'/*.mat'));
for i=1:length(d)
    name=d(i).name;
    pathname=d(i).folder;
    file=strcat(pathname,'/',name);
    load(file);
    graph1 = graph;
    name_comparation1=name(1:end-11);
    j=1;
    name2=d2(j).name;
    pathname2=d2(j).folder;
    name_comparation2=name2(1:end-12);
    while ((j < length(d2)) &&
(strcmp(name_comparation1,name_comparation2)==0))
        j=j+1;
        name2=d2(j).name;
        name_comparation2=name2(1:end-12);
    end
    if (strcmp(name_comparation1,name_comparation2))
        pathname2=d2(j).folder;
        file2=strcat(pathname2,'/',name2);
        load(file2);
        graph2 = graph;

[graph.Nodes,graph.Edges]=Generate_Graph(graph1.Nodes,graph2.Nodes,graph1.Edges
,graph2.Edges,7,5);
        file3=strcat('C:\Users\Usuario\Desktop\TFG_INFO\M_Pro\03-
ligands_proteins','/',name_comparation1,'_proteins_ligands');
        save(file3,'graph');
    end
end
```

10.8 Generate_Graph

```
function [nodesC,edgesC]=Generate_Graph(nodesL,nodesP,edgesL,edgesP,thR,thC)
    %thC is the threshold for non covalend bonds, usually 4 A
    %thR is the threshold to select a wider range with more atoms,
    %neighbour atoms of thR, so thC < thR, for example 7 or 8 A
    %The complex will first have all elements from the ligand and after the
    %protein atoms that fulfill a distance less than ThR A.
    edgesC=edgesL;
```

```
    [rowsedgeC,columnsedgeE]=size(edgesC);
    nodesC = nodesL;
    centerL = mean(nodesL(:,1:3),1);
    rowsP = size(nodesP,1);
    [rowsL,columnsL] = size(nodesL);
    rowsC = size(nodesC,1);
    j=1;
     % Generate nodes of the new graph
    posCP=zeros(rowsL,2);
    %Create matrix to know the atom position of a protein in the Complex
    %and in the actual Protein
    for i=1:rowsP
        distanceL = pdist2(nodesP(i,1:3),centerL); %it is the euclidean
distance between 2 points with 3 coordinates each
        if(distanceL<thR) %nodes near the ligand without non covalent bonds
            nodesC(rowsC+j,:) = nodesP(i,:); %we already have the ligand
            % nodes and we add the protein nodes
            posCP(j,1)=rowsC+j;
            posCP(j,2)=i;
            j=j+1;
        end
    end
    posCP = posCP(any(posCP,2),:);

    % Generate edges between ligand and proteins of the new graph
    rowsC=size(nodesC,1);
    covalentP=[];
    for z=1:rowsL %iterate over the ligans
        for x=(rowsL+1):rowsC %iterate over the protein atoms selected
            distanceLP = pdist2(nodesC(x,1:3),nodesL(z,1:3)); %it is the
            % Euclidean distance between 2 coordinates in the space
                if(distanceLP < thC) %nodes near the ligand with possible
                    % non covalent bonds
                    edgesC(rowsedgeC+(x-rowsL),z) = 3;
                    edgesC(z,columnsedgeE+(x -rowsL)) = 3;
                    if find(covalentP==x)>0
                        nothingtodo=0;
                    else
                        covalentP(end+1)=x;
                    end
                else
                    edgesC(rowsedgeC+(x-rowsL),z) = 0;
                    edgesC(z,columnsedgeE+(x-rowsL)) = 0;
                end
        end
    end
    covalentP=sort(covalentP);
    % Generate edges between protein-protein of the new graph
    [rowsedgeC,columnsedgeE]=size(edgesC);
    % posCP contains every proteins selected
    combPairs=nchoosek(posCP(:,2),2);
    rowscombPairs = height(combPairs);
    for c=1:rowscombPairs %iterate over the protein in the complex combination
pairs
        if edgesP(combPairs(c,1),combPairs(c,2)) > 0
            posPair1=find(posCP(:,2) == combPairs(c,1));
            pair1=posCP(posPair1,1);
            posPair2=find(posCP(:,2) == combPairs(c,2));
            pair2=posCP(posPair2,1);
```

```
            edgesC(pair1,pair2)=edgesP(combPairs(c,1),combPairs(c,2));
            edgesC(pair2,pair1)=edgesP(combPairs(c,1),combPairs(c,2));
        end
    end
end
```

10.9 Step5_Activity_distances

```
%This script reads a matlab file which contains activities and it calcultates
distance between ligands.
clc;clear all;close all
%Reading
load("Activity.mat");
% Calculate distances between activities from ligands
num_nodos = size(Activity,1);
x=1;
for i=1:(num_nodos-1)
    for j=(i+1):num_nodos
        activities(x,1)=abs((Activity(i)-Activity(j)));
        x=x+1;
    end
end
% Calculate the distance between ligands or a couple of ligand and protein
d = uigetdir;
d=dir(strcat(d,'/*.mat'));
x=1;
dd=length(d);
for i=1:dd-1
    name=d(i).name;
    pathname=d(i).folder;
    file=strcat(pathname,'/',name);
    load(file);
    graph1=graph;
    for j=(i+1):dd
        name=d(j).name;
        pathname=d(j).folder;
        file2=strcat(pathname,'/',name);
        load(file2);
        graph2=graph;

activities(x,2)=GED(graph1.Nodes,graph2.Nodes,graph1.Edges,graph2.Edges,1,1);
        x=x+1;
    end
end
% Save results in a mat2 file
save('activities','activities')
```

10.10 Step6_ ViewPlot

```
% Represent graphically (X -> distance between ligands ; Y -> distance
% between activities)
load("activities.mat","activities");
plot(activities(:,2),activities(:,1),'*');
xlabel('Distance between graphs (Angstrom)');
ylabel('Activity distance (M)')
title('Graphic to see the relation between KNN parameters')
```

10.11 Step7_Generate_matrix_distance

```
% This script returns a matrix with the distances between ligands or
% proteins-ligands
d = uigetdir;
d=dir(strcat(d,'/*.mat'));
dd=length(d);
disp(dd);
for i=1:dd
    name=d(i).name;
    pathname=d(i).folder;
    file=strcat(pathname,'/',name);
    load(file);
    graph1=graph;
    for j=1:dd
        if(i==j) distances(i,j)=200;
        else
        name=d(j).name;
        pathname=d(j).folder;
        file2=strcat(pathname,'/',name);
        load(file2);
        graph2=graph;

distances(i,j)=GED(graph1.Nodes,graph2.Nodes,graph1.Edges,graph2.Edges,1,1);
        end
    end
end
save('distances','distances');
```

10.12 Step8_Test_K_Nearest_Neighbor

```
% This program calculates all the minimum activities from ligand directory
load('Activity.mat')
num_ligands = size(Activity,1);
actividades_minimas(:,1)=Activity;
for i=1:num_ligands
actividad_predicha=K_nearest_neighbor(i);
actividades_minimas(i,2) = actividad_predicha;
end
save('actividades_minimas',"actividades_minimas");
```

10.13 K_nearest_neighbor

```
% This function finds the minimum activity from a ligand given a distance
% matrix
function [min_activity] = K_nearest_neighbor(pos)
    load('distances.mat');
    load('Activity.mat');
    num_ligands = size(distances,1);
    distancia_minima = distances(pos,1);
    ligando_minimo = 1;
    for i=2: num_ligands
        distancia = distances(pos,i);
        if(distancia<distancia_minima)
            distancia_minima = distancia;
            ligando_minimo = i;
        end
    end
```

```
    min_activity = Activity(ligando_minimo);
```

10.14 Step9_ViewPlot_K_nearest_neighbor

```
% Represent graphically (X -> activities ; Y -> predict activities)

load("actividades_minimas.mat","actividades_minimas");
plot(actividades_minimas(:,1),actividades_minimas(:,2),'*');
xlabel('Real Activity (M)');
ylabel('Predicted Activity (M)')
title('Graphic to compare real activity with predicted activity')
```