Development of the scoring function for protein-ligand binding affinity by implementing graph-theory based approaches

submitted by

SHIKHA SINGH

in partial fulfilment of the requirementsfor the award of the degree of

MASTER OF TECHNOLOGY

COMPUTATIONAL BIOLOGY

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

MAY, 2024

THESIS CERTIFICATE

This is to certify that the thesis titled Prediction of scoring function implementing graphical approaches, submitted by Shikha Singh, to the Indraprastha Institute of Information Technology, Delhi (IIITD), for the award of the degree of Masters in Computational Biology, is a bona fide record of the research work done by her under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Docent N Arul Murugan

Thesis Supervisor Associate Professor
Dept. of Computational Biology
IIIT Delhi, 110020

Place: New Delhi

Date:

ACKNOWLEDGEMENTS

This study and the research would not have been possible without the invaluable support of my supervisor, Dr. N. Arul Murugan. His guidance and mentorship throughout the research journey helped a lot in enhancing the quality of my work. He made great efforts in supporting me in all possible ways. I will be grateful to Veerendra, Rudrakshi Chauhan, and Prateek Paul for their help. I would also thank my friend Parneet Kaur and my family for encouraging me to pursue my academic goal. Their unwavering support and understanding helped me always to be motivated and focused in the time of difficulties. I am also grateful to my friends who always motivated me to reach my goals. This section cannot be completed without a vote of thanks to the academic department for their help and endless support

ABSTRACT

Scoring functions are essential to computational drug discovery since they evaluate or predict a ligand's binding affinity to its target protein. These functions rank the goodness of fit of a ligand into the binding site of a protein, thereby enabling the identification of better drug candidates. For many years, a great focus has been on developing scoring functions pointing to computational in particular graph-theory methods which are complementary to descriptors based approaches. This study, therefore, investigates different graph-theory based methods in the prediction of scoring functions for protein-ligand binding affinity. The approaches will include local and global input representations, voxel-based methods, graph-based drug-target affinity (Graph DTA) approaches, and GAT (graph attention model). Each technique has some advantage, which comes in handy in describing the complex interactions between ligands and proteins. The predicted binding affinities from each graph-theory based approach were compared to the experimental values, and R^2 (coefficient of determination) was used as the primary metric in the analysis. Local input representation i.e., the voxel-based approach, had an R^2 of 0.48, the graphDTA approach- GNN yielded an R^2 of 0.70, and the GAT(structure integrated graph neural network) approach yielded an R^2 of 0.42. The graph-based methods were observed to carry out better predictions for binding affinity. The deviation of the observed binding affinity from the expected value in the docking studies done using AutoDock Vina was further assessed, and the average deviation was 0.72, showing that while docking can be a powerful technique, some variability can be alleviated by integrating more advanced graph-theory based approaches. The improvements obtained concerning R^2 values from local representations through advanced graph-based methods show how these methods can be used to further improve the accuracy of computational drug discovery. Future work will be oriented toward refining these methods and their integration into comprehensive pipelines for drug discovery to accelerate the identification of promising drug candidates.

Table of figures

Table of Contents

ABBREVIATIONS

Iiitd	Indraprastha Institute of Information Technology Delhi
DTA	Drug Target Affinity
PDB	Protein Data Bank
3D	three Dimensional
ITC	Isothermal Titration Calorimetry
SPR	Surface Plasmon Resonance
QSAR	quantity structure activity relationship
ML	machine learning
DL	deep learning
RF	Random Forest
SVM	Support Vector Machine
GBM	Gradient Boosting Machine
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
GAT	Graph Attention Network
GNN	Graph Neural Network
GCN	Graph Convolutional Network
ChEMBL	Chemical Database of Bioactive Molecules with Drug-like Properties
CSAR	Community Structure-Activity Resource
CASF	Comparative Assessment of Scoring Functions
MOAD	Mother of All Databases (a database of protein-ligand complexes)
ECFP	Extended-Connectivity Fingerprint
MACCS	Molecular ACCess System (a type of molecular fingerprint)
SMILES	Simplified Molecular Input Line Entry System

NLP	Natural Language Processing
FNN	Feedforward Neural Network
LSTM	Long Short-Term Memory (a type of RNN)
FCN	Fully Connected Network
ADMET	Absorption, Distribution, Metabolism, Excretion, and Toxicity
MPNN	Message Passing Neural Network
PDBQT	Protein Data Bank, Partial Charge (Q), and Atom Type (T) format
MOL2	Tripos Mol2 file format (used for molecular structures)
ReLU	Rectified Linear Unit (an activation function in neural networks)
Conv	Convolution (often referring to convolutional layers in neural networks)
MSE	Mean Squared Error
MAE	Mean Absolute Error
HETATM	Heteroatom (refers to non-standard atoms in PDB files)
UFF	Universal Force Field

CHAPTER 1: Introduction

Prediction of the binding affinity of a protein and a ligand is among the fundamental targets of drug development. It defines the strength of interaction between a protein and a ligand and is directly associated with the drug's efficacy. When predicting binding affinities accurately, the drug development process is becoming more efficient regarding the time and resources spent. Traditional methods of binding affinity prediction involved experimental assays that were often expensive and time-consuming. As such, computational approaches have gained relevance in this field. Over the past few decades, several computational targets have been proposed to address the challenge: those are empirical scoring functions and rudimentary molecular docking known as the target-based approaches. Despite providing useful insights, these approaches lack the accuracy needed for predictions [1].

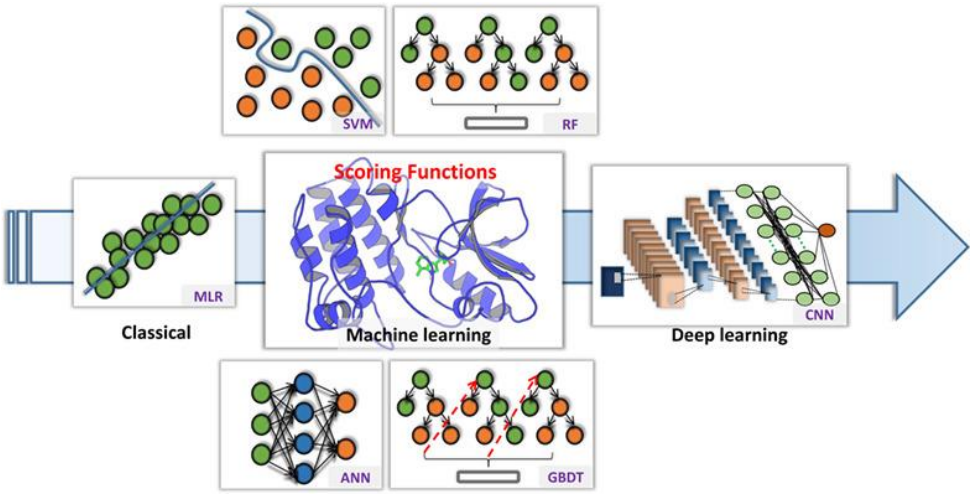

Figure 1. Advancements in scoring function [2].

The recent advancements in machine learning and artificial intelligence have opened new opportunities to improve the accuracy of binding affinity predictions. Large datasets can be processed using such methods, and complex algorithms can be run to

find intricate patterns in protein-ligand interactions which might otherwise go missed using more conventional techniques. One of these approaches seems to encourage the representation of protein-ligand interactions using graph-based models [3]. Graphs are a natural way to describe chemical structures because they can convey the relationships between atoms and molecules in an informational and intuitive way. In this scenario, proteins and ligands can be viewed as nodes, while their interactions can be viewed as edges. It makes it possible to apply potent graph neural networks, which accurately identify complex patterns and connections in the data [4].

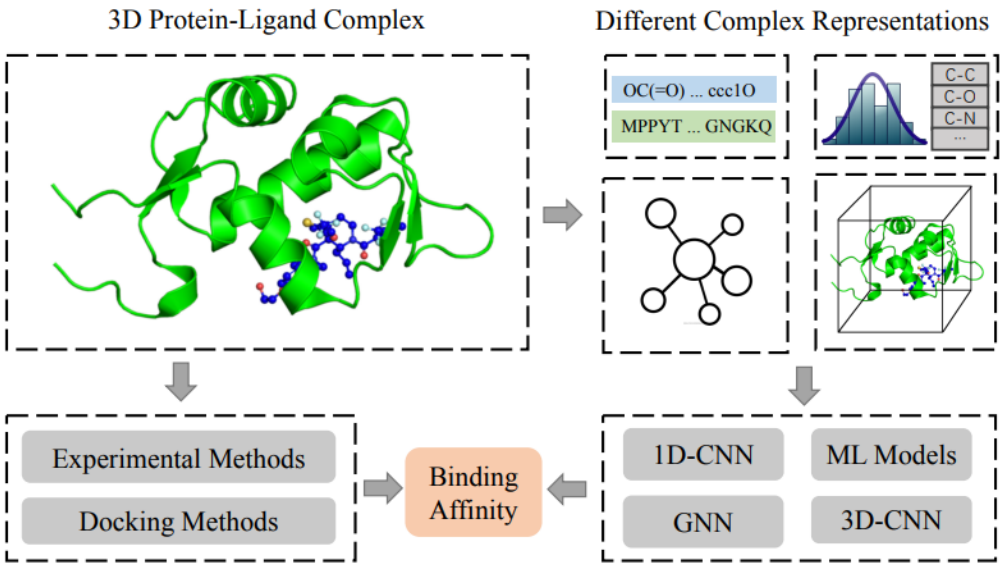

Figure 2. Different models and input representation [5].

The use of voxels—small cubic volumes—as a grid to describe the three-dimensional environment surrounding a binding site—represents another cutting-edge method. The spatial distribution of atoms and their interactions within the binding site can be captured by this approach. Through the process of transforming the three-dimensional structure of the protein-ligand complex into a voxel grid, binding affinities can be accurately predicted by machine learning algorithms [6].

An further sophisticated method in this field is the Graph DTA (Drug-Target Affinity)

model. To predict binding affinities, this model blends sequence-based representations with graph neural networks. The Graph DTA model takes advantage of the advantages of both sequence-based and graph-based methods by combining the structural data of molecules with their sequence-based properties, offering a thorough framework for binding affinity prediction [7].

A common practice is to use docking simulations with programs like AutoDock Vina to assess the performance of various computational techniques. A popular molecular docking program called AutoDock Vina predicts how a ligand will attach to a protein and offers a scoring system to gauge the binding affinity [8]. We may evaluate the relative benefits and drawbacks of each method by contrasting the outcomes from voxel representation, Graph DTA models, and pocket-ligand interaction graphs with those from AutoDock Vina.

The purpose of this study is to investigate and contrast these various computational approaches for binding affinity prediction in order to determine which strategy is the most precise and effective. One can aid in creating more potent computational tools for drug discovery by deepening our comprehension of these techniques. The 2020 PDB-bind dataset [9], It serves as the main source of data for this study and offers an extensive collection of binding affinities for protein-ligand complexes that have been determined experimentally. A reputable benchmark dataset in the field of computational drug design is the PDB-bind database. It includes hundreds of protein-ligands complex experimentally determined binding data taken from the Protein Data Bank (PDB). More than 19,000 complexes with excellent binding affinity data are included in the 2020 edition of PDB-bind, offering a solid basis for developing and accessing computational models. The dataset is a priceless tool for researchers creating and evaluating novel predictive models because it has been carefully selected to

guarantee the precision and dependability of the binding affinity measurements [1].

CHAPTER 2: Literature Review

The need for precise and effective prediction of binding affinities between proteins and ligands has propelled notable developments in the field of computational drug discovery in recent years. With an emphasis on three main techniques—voxel representation models, Graph DTA (Drug-Target Affinity) models, and pocket-ligand interaction graphs—this chapter examines the major advancements and methodologies in this field. It also looks at using AutoDock Vina, a molecular docking simulation, as a benchmark for various computational techniques [3]. This chapter's goal is to give readers a thorough grasp of the state of computational binding affinity prediction today, setting the stage for the techniques and experimental findings that will be covered in later chapters.

2.1. Binding affinity in drug discovery

The field of drug development is rapidly evolving, and understanding and predicting binding affinity have become two crucial endeavors. Since binding affinity is closely correlated with the efficacy, potency, and safety profile of the drug candidate, its relative relevance has only grown in the past. This essay looks at how the definition of binding affinity changed over time in the context of drug development. It also covers several methods for calculating it, examines variables that influence molecule affinity, and describes how binding affinity affects drug-target interaction [12].

The historical landscape of drug discovery is marked by a slow but progressive shift toward an emphasis on binding affinity. Drug discovery at first was mainly dependent on empirical screening methods[13]. This had the drawback of leading mostly to serendipitous findings, where the underlying molecular mechanisms at play were not well understood. But slowly, as scientific knowledge became advanced, scientists came

to appreciate that specific interactions between drugs and their target proteins were essential. This paradigm shift was essential in opening the route to a more rational way of drug design in which the quantification and optimization of binding affinity came of greater importance [14]. Presently, the use of computational methods and techniques of structural biology has enhanced the capacities in the prediction and manipulation of binding affinity. Nowadays, binding affinity is a critical parameter in nearly every stage of drug discovery—from lead identification to clinical development. Its importance has been intensified in later years, with the growing complexity of drug targets and the continuous demand for safer and more effective therapeutics [15].

The computation of the binding affinity makes use of a wide array of experimental and computational techniques, each contributing their perspectives regarding the drug-target interaction. Some of the experimental methods used are ITC, SPR, and X-ray crystallography, which offer a very direct and comprehensive way of measuring the binding constants and defining the interaction at the molecular level. All of these techniques, while beneficial, take a lot of effort and resources, as well as experience [14], [16].

Predicting binding affinities considerably more effectively is possible with computational approaches. Molecular docking, molecular dynamics simulations, and quantitative structure-activity relationship (QSAR) models are some of the most widely used computational techniques. For example, whereas molecular dynamics can mimic the dynamic behaviour of the protein-ligand complex over time, molecular docking can predict the preferred orientation of a ligand in a protein's binding site. By establishing a relationship between a compound's chemical structure and biological activity, QSAR models are generated, making it feasible to predict binding affinity using molecular descriptors. The merging of machine learning (ML) and deep learning (DL) methodologies has resulted in a substantial change in the prediction of binding affinity in drug discovery. Large datasets of known protein-ligand interactions are utilised by these data-driven techniques to create highly accurate and effective predictive models [11].

In QSAR modelling, machine learning methods including RF, SVM, and GBM have been extensively utilized. Using molecular descriptors generated from chemical structures and experimental binding affinity data, they are able to identify patterns. In order to forecast structure-activity connections and offer a platform for the design of novel compounds with ideal binding qualities, the ML models learn intricate nonlinear relationships that exist between the molecular features and the binding affinity [17], [18].

Binding affinity prediction has found a valuable tool in deep learning, a form of machine learning. CNNs and RNNs in particular, which are deep neural networks, have demonstrated an amazing capacity to learn intricate features from molecular representations. When it comes to drug development, deep learning models (DL models) have the ability to predict binding affinities with previously unachievable accuracy by learning from raw chemical structures, such as 3D protein-ligand complexes or ligand fingerprints. With the development of molecular graph convolutional networks, a promising paradigm for binding affinity prediction is the graph-based method. GCNs consider molecules to be graphs with atoms as nodes and bonds as edges. GCNs are able to extract both local and global structural information from these molecular networks by applying convolutional processes, which can be used to predict binding affinity based on molecular topology [19].

Including ML and DL in the binding affinity calculation has several significant benefits. Large and varied datasets may be handled by these data-driven methods, which can adapt well to nonlinear relationships between binding affinity and chemical characteristics and generalize effectively to previously undiscovered compounds. To increase the predictability and resilience of conventional computational techniques, the ML and DL models can be used in conjunction with them [17].

The efficacy, potency, and selectivity of drug candidates are determined by the binding affinity, which holds a crucial position in the drug-target interaction. Generally speaking, a higher binding affinity would provide better potency, enabling the medicine to produce the intended therapeutic effect at lower dosages. By reducing the possibility of off-target interactions, selective binding to the desired target protein minimises side effects and improves the drug's

safety profile. Moreover, binding affinity affects how long a medicine acts; longer-lasting drug-target complexes result in longer-lasting therapeutic benefits. Naturally, dose schedules and patient compliance are more directly affected by this temporal factor [4], [20]. Furthermore, it guides optimization of lead through the design of molecules with improved binding properties and better pharmacokinetic profiles. The fact that binding affinity has evolved from a vaguely conceptual notion to a quantifiable parameter reflects drug discovery's maturation as a scientific field. Experimental and computational research continues to unravel the intricacies of molecular interactions in efforts to optimize binding affinity for the attainment of better therapeutic outcomes. And so, the role of binding affinity remains undiminished as drug discovery ventures into its ever-expanding frontiers in search of safer and more effective medicines [14].

2.2. Protein-ligand complex databases

Protein-ligand interaction datasets are collections of experimental data that describe the interactions between proteins and small molecules (ligands). This dataset usually provides information about the three-dimensional structure of protein-ligand complexes and the quantitative measurement of binding affinities, binding kinetics, and other relevant properties. The importance of protein-ligand interaction datasets includes serving as a resource of paramount importance for drug discovery and research in computational biology. The relevance of commonly used datasets for protein-ligand interaction is explained below:

ChEMBLv2023: The European Bioinformatics Institute is in charge of maintaining ChEMBL, a massive bioactivity database [21]. The most recent edition includes a wealth of knowledge on biological activities, targets, and chemical characteristics of bioactive compounds. This edition includes information from 1.3 million assays, 15 million bioactivity measures, and over 2 million chemicals. It targets several different biological targets, including as cells, proteins, and nucleic acids. For this reason, ChEMBL is a highly valuable tool for pharmacological research and drug discovery. It also offers tools for data retrieval, visualisation, and analysis, which aids in the investigation of structure-activity connections and the identification of possible

therapeutic options.

The goal of CASF 2016: Comparative Assessment of Scoring Functions is to test the molecular docking scoring functions' performance using a benchmark dataset and assessment exercise. It is made up of a variety of protein-ligand complexes with excellent crystal structures and binding affinities that have been determined through experimentation. Providing a standardised framework for evaluating the accuracy of different molecular docking scoring systems is the main goal of CASF 2016. The dataset comprises 285 protein-ligand complexes and is used to perform a thorough assessment of the accuracy of ranking, pose prediction, and binding affinity prediction [22]. Thus, for academics working on the creation and improvement of scoring systems for computational docking investigations, CASF 2016 is a highly helpful tool.

A community-wide initiative called CSAR (Community Structure-Activity Resource) aims to evaluate and enhance computational techniques for protein-ligand interaction prediction. The scientific community has curated and supplied high-quality protein-ligand complexes with empirically determined binding affinities from the literature to create the CSAR dataset. Benchmark datasets are made available by CSAR to evaluate the effectiveness of scoring systems, docking strategies, and free energy computation methods. Researchers can also test their computational models against unpublished experimental data by taking part in blind prediction competitions. In an attempt to increase the precision and dependability of computer models for drug development, CSAR encourages cooperation [23].

The Protein Data Bank's experimentally determined binding affinities for protein-ligand complexes are curated in a database called PDBbind. It provides information on protein-ligand complex annotation, binding affinities, and three-dimensional structures. An acknowledged standard for the creation and assessment of computer models for binding affinity prediction is PDBbind. To guarantee high-quality, trustworthy data, the dataset is meticulously selected and updated on a regular basis. PDBbind is a valuable tool that helps advance drug discovery research by offering a comprehensive and dependable computational modelling resource.

MOAD (Mother of All Databases) is a meticulously curated collection of superior protein-ligand complexes with well-defined binding processes, designed with precision and consistency

in mind. High-quality structures and empirically confirmed binding affinities are emphasised. MOAD provides comprehensive structural data, including as atomic coordinates and ligand properties. For all these reasons, it is an ideal setting for the in-depth investigation of protein-ligand interactions. This database makes research on virtual screening, structure-based drug development, and molecular modelling easier [23].

A public database called BindingDB gathers binding affinity information for protein-ligand complexes from literature papers and other public sources. Proteins, nucleic acids, and protein-protein interactions are all goals in biology. Comprehensive data on ligand structures, experimental settings, and binding affinities may be found in BindingDB. The database is a useful resource for drug development research because it can be searched and accessed via an intuitive web interface. With a wealth of comprehensive experimental data, BindingDB facilitates virtual screening, structure-based drug design, and lead optimization. [23].

The ChEMBL v2023, CASF 2016, CSAR, PDBbind, MOAD, and BindingDB datasets are all essential for the progress of drug design and protein-ligand interaction research. These databases are useful for training and validating computational models, optimizing drug candidates, and investigating structure-activity connections because they include high-quality data on protein-ligand complexes, binding affinities, and structural information. These datasets provide thorough and trustworthy data that make it possible to expedite drug discovery and foster innovation in computational biology and pharmacology.

2.3. Input representation of complex

The prediction of binding affinity in protein-ligand complexes is primarily a challenge involving the input representation. Protein-ligand complex structural and chemical information is represented into a format that computer models can process using input representations. There are two broad categories of representations: local and global. Each classifies various features of protein-ligand interactions, each with pros and cons. To further understand the role of these representations in binding affinity prediction optimization, we investigate graphical models in more detail [23], [24].

2.3.1. Local Representation

Local representations offer in-depth understanding of protein-ligand interactions at the molecular level by concentrating on the immediate surroundings of the binding site. Consequently, this offers high-resolution data on binding interactions. The three-dimensional structure of the binding site is discretized into a grid of volumetric pixels, or voxels, in a standard local representation known as the voxel-based representation. Values corresponding to the kinds, partial charges, existence, and other molecular attributes of the atoms can be stored in such voxels. CNNs are especially good at this strategy because they can extract fine details about the binding interactions by analyzing spatial patterns in the 3D grid [7]. Voxel-based representations are powerful in capturing fine-grained geometric and physicochemical properties but can be computationally intensive and may lose some structural details due to the discretization process.

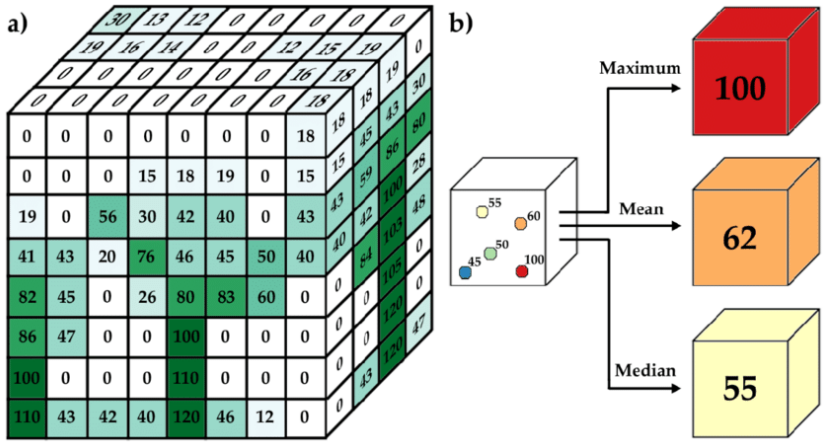

Figure 3. Depiction of voxel representation a)grid b)how nodes are defined [25].

Other local representations include molecular descriptors and fingerprints. These are numerical representations encoding various molecular properties and structures, concentrating on the ligand and the residues in the binding pocket. Descriptors could include physical properties like molecular weight and logP, topological indices, and electronic properties. Fingerprints are binary or integer vectors that show the presence of certain substructures or molecular features, such as Extended Connectivity Fingerprints (ECFP) and MACCS keys [11]. These representations are compact and efficient and thus amenable to a wide variety of machine

learning algorithms. However, they may not directly encode spatial relationships and could miss detailed interaction networks of the binding site [19].

2.3.2. Global representations

Global representations encompass the whole protein-ligand complex, offering a general perspective of the interaction. This holistic view captures not only the local but also the distal effects that modulate the binding affinity in a complete structural context. Perhaps the most prominent example of representation for interaction modeling are graph models, where molecules are viewed as graphs and nodes are atoms with bonds viewed as edges. In protein-ligand complexes, this might include constructing a single interaction graph for both the protein and ligand or maintaining separate graphs for each, with edges indicating interactions between the two [4]. Node features can be atom types, hybridization states, partial charges, and other atomic properties; edge features capture bond types, lengths, and interaction strengths. GNNs are particularly good at processing these representations and learning from the detailed molecular structures and their interaction between the protein and ligand. This kind of encoding captures complex interaction patterns in a flexible and natural manner but is computationally very expensive and needs complex algorithms to process graphs [26].

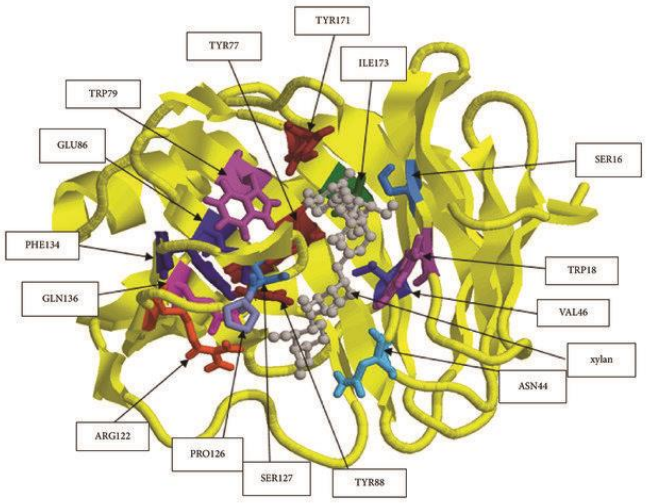

Figure 4. Molecular structure of drug-target [27].

Another global approach entails the sequence-based representation, encoding proteins and

ligands with their sequences of amino acids and SMILES (Simplified Molecular Input Line Entry System) strings, respectively. Protein sequences can be handled by models such as RNNs or transformers that emphasize sequential dependencies and motifs. Similarly, ligands, as represented by their SMILES strings, can be directly processed or converted into the molecular graph representation. Such representations are straightforward and borrow from the recent advances in NLP, making them easy to handle. [24] Nevertheless, they may lose the spatial information unless complemented with structural data, which may make it difficult to fully capture the intricacy in protein-ligand interactions.

The proper selection of representation in graphical models will be very important for the accuracy and efficiency of the binding affinity prediction. Graph-based representations are intrinsically powerful because they represent molecular structures and interactions naturally. GNNs can incorporate attention mechanisms into their architecture for the model to give more importance to the most crucial interactions present in the graph [5]. This approach enables the model to dynamically weight the importance of different nodes and edges for enhancing prediction accuracy. Multi-scale graph representations can be used, whereby the model at different levels of granularity processes the information—from the individual atoms to the larger molecular fragments. This approach would catch the local interaction details and the global structural patterns, leading to a holistic understanding of the protein-ligand complex. Combining local and global representations can offer more robust models [28]. For example, some starting GNN layers may use local subgraphs around the binding site for the detailed interactions, while later layers accumulate global information to take into account distal effects and overall conformational changes. It would be possible to explore the combination of voxel-based and graph-based representations, where the voxel grid provides the spatial context and the graph represents the detailed chemical interactions. This hybrid model takes the advantage of the strength of both methods and might yield more accurate and interpretable predictions [29], [30].

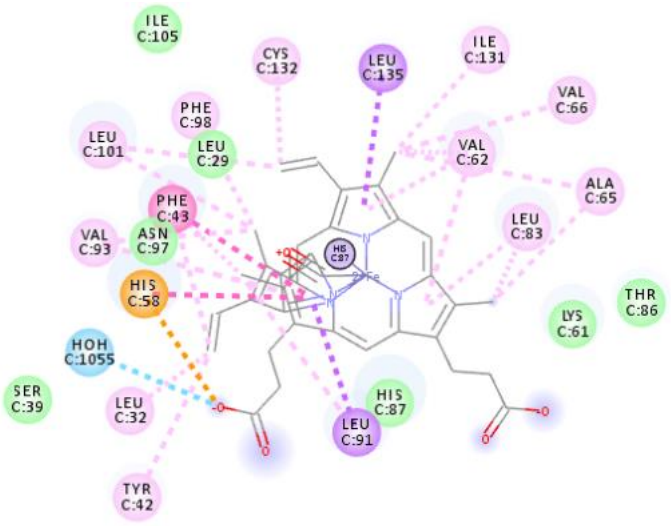

Figure 5. Graphical representation of drug-target [31].

The choice between local and global representations in graphical models should be made according to the concrete requirements of the task in binding affinity prediction and available computational resources. Local representations, such as voxel-based methods and molecular descriptors, provide very detailed and specific information about the site but might lose the more general patterns of the interactions. Global representations—graph-based and sequence-based methods—can provide a more holistic view of the complex and interactions that take part across the whole structure. Using state-of-the-art graph-based methods and possibly hybridizing representations, the researchers will make their computational models more accurate and robust to enable and speed up drug discovery [10].

2.4. Voxel representation in computational chemistry

Voxel representation is a very powerful technique within computational chemistry in modeling and prediction at a molecular interaction level. A voxel, or volumetric pixel, is a unit of graphic information that defines a point in three-dimensional space. The division of the three-dimensional space of a molecule or a molecular complex into a grid of voxels allows researchers to grasp detailed spatial information about molecular structures and their interactions.

Voxel-based representation relies on the superimposition of a 3D grid over a molecular structure, where every voxel can store information on the presence of atoms, their types, partial charges, and other pertinent molecular properties [6][32]. This captures the entire molecular

environment in very high resolution. The resulting voxel grid could then be fed into convolutional neural networks, which are themselves very capable of handling spatial data. CNNs can learn to recognize patterns within the voxel grid that correlate with specific molecular interactions or properties, like binding affinity.

One of the greatest benefits of voxel representation is its ability to detail fine-grained features of molecular geometry and electrostatics. This is important in a realistic model of interactions. However, voxel representation may be computationally costly. The voxel grid must be set at a sufficiently high resolution to capture important details, thereby increasing the computational cost and memory requirements [7]. There is also the chance that some structural details might get lost due to the discretization process. Despite these challenges, voxel representations are one of the methods used for structure-based drug design and virtual screening, where the method predicts how well a potential drug molecule will bind to a target protein. They are also widely used in various predictive modeling tasks such as toxicity prediction and enzyme activity prediction.

2.5. Neural Network in predictive modelling

Artificial neural networks have greatly changed predictive modeling in computational chemistry and drug discovery. These models are based on the architecture of the human brain and the way it works, enabling them to learn hard patterns and relationships from a large dataset. Some of the most applicable types of neural networks in this domain are:.

The simplest kind of neural network, feedforward neural networks (FNNs), transmits information in one direction from input to output [29], [33]. Basic predictive tasks of FNNs do not perform well with complex, high-dimensional data typical for computational chemistry. Convolutional Neural Networks (CNNs) are especially appropriate for processing spatial data, like voxel grids of molecular structures. They learn to identify which features within the 3D structure correspond best with binding affinity or any other properties. RNNs are specifically designed for sequential data and, hence, find application in problems dealing with protein sequences or time series. A variant with Long Short-Term Memory, or LSTM, networks is of

particular value in uncovering long-range dependencies in sequences. GNNs are better placed to handle molecular data, which is represented as graphs where atoms are nodes and bonds are edges. GNNs can learn the representation of structural and interaction patterns within molecular graphs [34].

Neural networks model non-linear relationships and interactions that traditional methods may not catch. They are particularly good at dealing with large, complex datasets. However, they require large amounts of data and significant computational resources for training. In addition, neural networks can be considered "black boxes" since it is hard to interpret how they make predictions. In spite of these difficulties, neural networks have, nonetheless, found widespread application in drug discovery for processes such as binding affinity prediction, ADMET property prediction, and virtual screening. They are also applied to cheminformatics in property prediction, molecular design, and optimization.

2.6. Graph-based Approaches in Protein-Ligand Interaction

The analysis of protein-ligand interactions by graph-based methods plays a critical role in computational biology and drug discovery. These methodologies use concepts from graph theory to model complex interactions between proteins, the large and indispensable biomolecules, and ligands, usually smaller molecules that bind to proteins and can modify their function. Graph-based methods have emerged as powerful tools to study protein-ligand interactions because the representation of complex structures and relationships between the molecular systems can be captured naturally within them [35]. In this form of modeling, proteins and ligands are graphs where atoms are nodes, and bonds are edges. This can encode the molecular topology and all the details of the interaction within the binding site. For protein-ligand complexes, interaction graphs can be created including both the protein and the ligand, with edges not just for covalent bonds but also for the non-covalent interactions, like hydrogen bonds and hydrophobic contacts. Several are especially well known. Graph Convolutional Networks (GCNs) extend classical neural networks to operate on graph structures, updating node features by aggregating information from neighboring nodes [33], [35].

This will effectively capture local and global structural information within the graph. GATs apply attention mechanisms so that the model will only focus on the most critical interactions within the graph. It provides the capability for the model to dynamically weight the importance of nodes and edges, thus improving the accuracy of prediction. MPNNs are another popular method, in which messages are passed between nodes to update their features and grasp intricate details of molecular interactions. Graph-based methods are flexible and can model the complicated interaction networks within protein-ligand complexes. They can capture both local interactions and broader structural contexts [14], [36]. But these models can be computationally expensive, particularly for large graphs. Furthermore, sophisticated algorithms and an understanding of graph theory are needed for the creation and processing of graph representations. Graph-based models are widely employed in drug development to predict binding affinity, identify therapeutic targets, and explore the mechanisms of drug action despite these difficulties. They are also used in cheminformatics to detect scaffold hopping, evaluate toxicity, and forecast molecular characteristics.

By definition, a graph is made up of nodes, or vertices, joined by edges. Nodes may represent atomic structures or molecular subunits in the context of protein-ligand interaction research, whereas edges may indicate bonding or spatial interactions [37]. Graphs can be used to simulate protein structure in a number of ways. One such graph is the Residue Interaction Graph, where the nodes are amino acid residues and the edges reflect interactions such as hydrophobic contacts and hydrogen bonds. Ligands can usually be represented as molecular graphs, where the nodes are atoms and the edges are chemical bonds. These graph models are fundamental for simulating the docking process through which ligands bind to proteins. Algorithms generate a number of orientations and conformations of ligands, called poses, in the binding site of the protein; each of them is evaluated against interaction energies and geometrical fitting. Graph-based methods also assist in the development of scoring functions that can evaluate stability and the possibility of each pose based on binding affinity prediction, shape complementarity, and a variety of molecular interactions [35].

Machine learning, in particular, GNNs, has significantly enhanced these traditional methods:

GNNs could learn intricate patterns of interactions from large datasets of known protein-ligand complexes to predict binding sites, optimize scoring functions, and assess the impact of protein mutations on ligand binding. Graph-based techniques have been extensively used in drug design, enabling scientists to screen vast libraries of ligands efficiently, in understanding disease mechanisms by explaining how protein mutations may affect the interaction of ligands, and in synthetic biology to design molecules or proteins with specific target interactions. However, such approaches are challenged by how to handle the dynamic nature of protein structures and how to integrate multi-scale data. Future improvements will likely include more dynamic modeling techniques and comprehensive multi-scale approaches that enhance accuracy and practical applicability [30].

2.7. Docking Techniques and AutoDock Vina

Molecular docking is a computational technique that is used to predict the favored orientation of a ligand when bound to a protein target. This computational technique is critical to understanding binding affinity and specificity [38]. Docking involves the generation of possible binding poses of a ligand within the active site of a protein and rating these poses using a scoring function to estimate their binding affinity. The aim is to find which of those poses best represents the actual binding interaction. All algorithms used to sample conformational space involve both deterministic methods, which include methodologies that are meant for the generation of structures with the known optimal solutions by various iterative methods; stochastic methods, which use probability in the process of ligand conformational space exploration; and hybrid approaches, which combine elements from the former two. AutoDock Vina is one of the most used and, simultaneously, one of the most popular molecular docking programs [38]. It is considered an extension of its predecessor, AutoDock, with higher accuracy and speed. Vina is using a state-of-the-art scoring function that combines an empirical term with a knowledge-based term to evaluate the binding poses. An efficient search algorithm is also included to sample the conformational space of the ligand in the binding site. AutoDock Vina has its advantages in the form of accuracy and computational efficiency. It is user-friendly

and integrates with most visualization tools, making it accessible for researchers.

Just like all the other docking programs, Vina has its limitations [8]. Its accuracy is determined by the quality of input structures and how well the scoring function represents all the relevant interaction forces. Besides, docking does not consider protein flexibility, which is crucial for accurate predictions. Despite these challenges, a technique of docking, including AutoDock Vina, is used broadly in virtual screening with a purpose of finding potential drug candidates; in lead optimization to refine the binding properties of drug candidates; and in mechanistic studies to understand at the molecular level how drugs will act. They are also employed to design inhibitors of enzymes and other protein targets [13].

The literature review summarizes the general concepts and methodologies applied in the computational chemistry and drug discovery domain. The review covers various subjects: prediction of binding affinity, input representations, docking techniques, and graph-based methods to elaborate on the various strategies followed to overcome the challenges of understanding the molecular interactions for the design of novel therapeutics.

By discussing the role of protein-ligand interaction datasets—PDBbind, BindingDB, MOAD, ChEMBL, CASF, and CSAR—the review incorporates the importance of reliable and curated data sources to be beneficial in progressing computational modeling efforts. These are very rich datasets for training and validation in predictive models to achieve better and more robust algorithms in drug discovery.

Moreover, the input representation discussion based on encoding molecular structures and interactions into formats appropriate for computational analysis reflects that researchers use various techniques to capture the intricate details of protein-ligand complexes, capturing in-depth molecular recognition processes.

The review also gives insight into the utility of neural networks in predictive modeling, showing great applicability in working with complex data and deriving meaningful insights from huge datasets. From feedforward neural networks to graph neural networks, these models provide a powerful tool to predict the binding affinities, properties, and activities of potential drug candidates and spur the innovation in the research field of drug discovery. Moreover, the

discussion on docking techniques, especially AutoDock Vina, underlines the key role of computational simulations for the elucidation of ligand binding mechanisms and the guidance of drug design efforts. In spite of challenges in regard to the accommodation of protein flexibility, molecular docking still serves as the cornerstone for virtual campaigns of screening and strategies of optimization of lead compounds, which make possible the identification and refinement of promising drug candidates. In sum, the literature review reiterates the fact that computational chemistry and drug discovery are interdisciplinary; synergy efforts between computational modeling and data science with the validation from experiments are necessary to push the frontier of our understanding of molecular interactions and to accelerate novel therapeutics. By utilizing cutting-edge methodologies and leveraging curated datasets, researchers can overcome challenges and spur innovation in drug discovery that will bring about better treatments for various diseases and medical conditions.

CHAPTER 3: Data Collection and Preprocessing

In this section, we delineate the methodology that we followed in collecting, preprocessing, and preparing the dataset for further use in analysis. With our focus on the PDBbind dataset, the preprocessing techniques were tailor-made to accommodate different input representations, including voxel-based, graph-based, graphDTA, complex interaction graph, and AutoDock Vina.

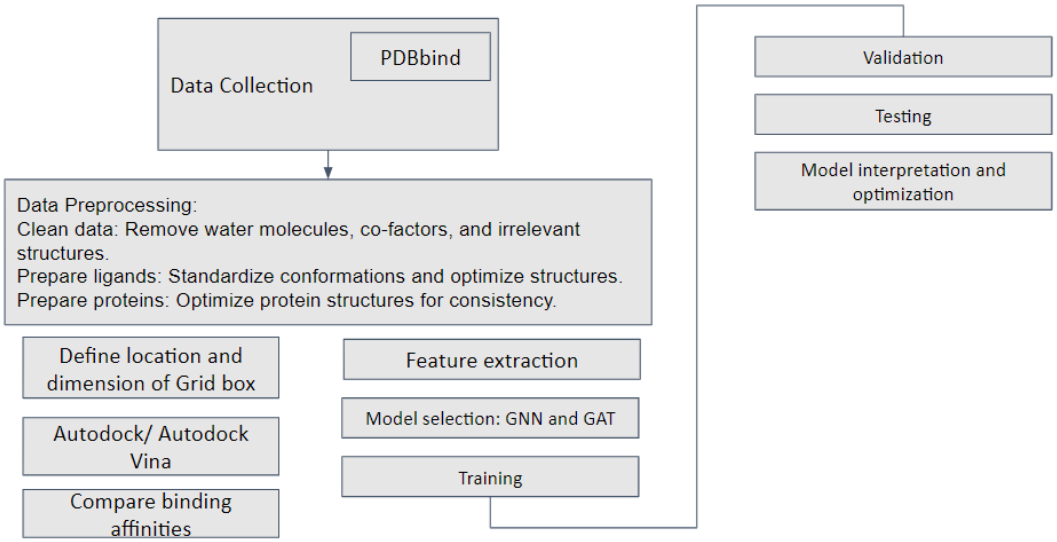

Figure 6. Workflow.

A valuable resource for computational chemistry and drug development is the PDBBind collection. It provides a large set of experimentally determined binding affinities for PDB-derived protein-ligand complexes. This dataset is a standard benchmark for the development and validation of binding affinity prediction models since it is well-curated and contains three-dimensional structural information of the complexes, their related binding affinities, and annotation features.

The Core, Refined, and General sets are the three highly curated subsets of the PDBBind dataset. These sets are built on distinct standards to meet diverse needs and objectives of research. The subsets serve as common reference points for creating and evaluating computer models that predict binding affinity. A selection of carefully chosen protein-ligand complexes,

known as the Core set, are arranged according to strict standards such as ligand variety, binding affinity, and resolution. As a result, it offers an assortment of carefully chosen binding scenarios that guarantee stability and dependability throughout model testing and training. The extra complexes in the Refined collection are derived from the Core set, but their experimental binding affinities have been improved. To guarantee precision and consistency, these complexes go through validation and standardization processes. Conversely, a wider variety of protein-ligand complexes, including ones with different levels of experimental uncertainty and quality, are represented by the General set. The General set offers a broad view of binding interactions and, while not as carefully chosen as the Core and Refined sets, can help with exploratory investigations and the development of hypotheses.

Depending on the PDBbind dataset version, each subset may contain a different amount of chemicals. It's always being selected. But in the most recent iteration that is currently accessible, the Core set typically comprises of a few hundred carefully selected protein-ligand complexes that illustrate various binding possibilities. Complementary complexes with refined binding affinities are included in the Refined set. Last but not least, the General set is meant for exploration and is the largest, with hundreds or perhaps thousands of complexes.

Strict selection criteria are used in the PDBbind dataset to create the Core, Refined, and General sets, which guarantee the quality, diversity, and applicability of the data for drug discovery and computational modelling. For the purpose of creating and validating predictive models for the prediction of binding affinity and other related tasks in structure-based drug discovery, these subsets offer researchers significant tools.

Iterative updates are made accessible for every version of PDBbind, including the v2020 that is utilized, to guarantee that the most recent data and refinement techniques are used. This guarantees the dataset's continued relevance and the competitiveness of the computational models that are constructed with it. As a result, the dataset's rigors and usefulness are preserved in v2020, which was utilized in this work. For the detailed statistics, the core set in the v2020 version includes 485 complexes, the refined set has about 4,852 complexes, and the extensive general set houses more than 17,300 entries.

This tiered structure of the PDBbind dataset not only enables a wide spectrum of computational tasks, from basic research to advanced model training, but also allows for the meticulous cross-validation and robustness checks so essential in the competitive and fast-evolving landscape of computational drug discovery.

Preprocessing techniques play a vital role in preparing data for computational modeling, particularly in predicting binding affinities for protein-ligand complexes. This part describes the preprocessing techniques developed for voxel-based, graphDTA, graphical, and AutoDock representations, respectively, all trying to capture different aspects of molecular interaction and structural features.

3.1. Voxel -based Representation

Voxel-based representation involves discretizing the three-dimensional space around the protein-ligand complex into a grid of volumetric pixels or voxels. Such a representation allows the encoding of spatial information and structural features within the binding site.

Preprocessing workflow for voxel-based representation normally starts from the extraction of molecular structure from the PDBbind dataset. Such structures include both the protein and ligand parts. Secondly, the binding site of the protein-ligand complex is identified; often, it is known as residues that are close to the ligand or known binding pockets.

When the binding site is determined, it then discretizes the three-dimensional space around the complex into a grid of voxels. The resolution and size of the voxel grid might be dependent upon the specific needs of the task of modeling. The finer grids, those with smaller-sized voxels, are often used in capturing detailed structural features, while coarse grids suffice for broader analyses. For every voxel in the grid, a number of molecular properties is computed or assigned based on the presence of atoms within the voxel or its close proximity. These properties include atom types, partial charges, volumes of van der Waals, and accessibility to the solvent. In addition, it is possible to include features, such as atomic occupancy or density in representing the occupancy of every voxel by atoms from the protein or ligand. Once the grid of voxels is populated with the most relevant molecular properties, it acts as the input representation to computational models trained for the prediction of binding affinities. Techniques such as

convolutional neural networks are often used to process voxel-based representations and get predictive features extracted from the spatial data.

3.2. GraphDTA Representation:

GraphDTA representation comes from graph structures representing interactions between protein atoms and a ligand in the complex. In this type of representation, the atoms are represented by nodes, and interactions—like covalent bonds, hydrogen bonds— between atoms are represented by edges.

GraphDTA Representation preprocessing: The graph model of the protein-ligand complex is built with several preprocessing steps. First, it parses the molecular structure of both proteins and ligands for atom coordinates, bond information, and other molecular properties. After that, it constructs a graph where atoms from the protein and ligand components are the nodes, and interactions between atoms are the edges. These interactions may include covalent bonds within molecules and non-covalent interactions between atoms from different molecules. Once the graph is built, additional features can be added into nodes and edges to further capture relevant structural and chemical properties. These features may include atom types, partial charges, bond orders, and distances between atoms. The resulting graph embeds the topology and interaction pattern of the protein-ligand complex. It will serve as input for the computational model, which will be used in training to predict binding affinities. GNNs are one of the most common models used to process graph-based representations and to learn predictive features from graph structures.

3.3. AutoDock representation:

AutoDock representation entails preparation of the protein and ligand structures into formats compatible with the AutoDock software suite, specifically AutoDock Vina, a very popular molecular docking program.

Basically, the AutoDock representation preprocessing would include several steps. First, the molecular structure of the protein and ligand can be fetched from the PDBbind dataset or other sources, then transformed into formats compatible with AutoDock, such as PDBQT or mol2.

Next, the protein structure is prepared with the addition of hydrogen atoms, assignment of

charges, and definition of the binding site or active site residues. The ligand structure is similarly processed with addition of hydrogen atoms, assignment of charges, and optimization of torsional degrees of freedom.

Then, the structures of the protein and the ligand are prepared and afterwards docked using the software suite AutoDock, where the conformational space of the ligand in the binding site is explored and the possible binding poses are evaluated by a scoring function. Therefore, the output includes the docking poses and the corresponding binding affinities, which are fed into downstream analyses or serve as reference data for training and evaluating computational models. AutoDock Vina has high accuracy and efficiency in the prediction of binding poses and affinities, and hence it is a valuable tool for structure-based drug design and virtual screening studies. Preprocessing techniques for voxel-based, graphDTA, graphical, and AutoDock representations are of critical importance for preparing data for computational modeling tasks, especially predicting the binding affinities of protein-ligand complexes. Each of these representations captures different aspects.

Chapter 4: Voxel Representation based approach

4.1. Parsing Ligand from MOL2 File

The preprocessing of the ligand begins with extracting the ligand structure from a MOL2 file. Upon detecting the tag "@<TRIPOS>ATOM", the parsing function is called to extract information about the atoms. From every line containing information about atoms, the parsing function extracts atomic coordinates: x, y, and z, which provides a full description of the molecule's spatial configuration. These extracted coordinates are stored in the list of 3D coordinates, 'coords'. This structured storage of coordinates facilitates further analyses without requiring direct recourse to the underlying code and streamlines the pipeline of preprocessing for efficient and reliable ligand characterization.

4.2. Extracting Pocket Around Ligand from Protein Structure:

Pocket preprocessing, one of the steps leading to the understanding of the local binding environment around the ligand, is to extract structural elements proximal to the ligand from within the protein structure. This extraction of coordinates is facilitated by the extract_pocket function, working on a protein structure file (pdb_file) and the coordinates of a ligand (ligand_coords). It starts by initializing a PDB parser for parsing the protein structure, which then goes through every atom in the structure. For every atom in the structure, the coordinates of this atom (atom_coords) are obtained via the method get_coord().

Then it calculates the distance between the coordinates of this atom and the coordinates of the centroid of the ligand. This calculation of distance with the help of np.linalg.norm defines the proximity of an atom to the ligand. Precisely, if the distance computed is less than or equal to some specified radius of the pocket, called the pocket radius, that atom will be considered as part of the pocket, and its coordinates get appended to the pocket_atoms list.

The resulting pocket_atoms list consists of atoms from the region of the pocket radius, holding the structural aspects in close proximity to the ligand. This focused extraction ensures that the pocket representation contains the region vital for ligand-protein interactions and provides rich

information about the local environment of the binding. Eventually, the extract_pocket function serves for careful pocket preprocessing and, therefore, for advanced structural analysis and computational modelling in structural bioinformatics.

Generally, these preprocessing steps lay the foundation for advanced voxel-based analyses by careful preprocessing of ligand and pocket structures. The coordinates of the parsed ligand and extracted pocket atoms are short, yet meaningful, representations of their respective structural contexts, and they carry out their functionalities in downstream analyses without the need to make a direct reference to their code. In this systematic preprocessing approach, voxel-based methodologies in structural bioinformatics are empowered to investigate intricate ligand-protein interactions and assist in the process of discovering new drugs with enhanced precision and efficiency.

4.3. Generation of Voxel Grid

The voxel grid generation process is the three-dimensional creation of the ligand-protein complex. It encapsulates the spatial distribution of atoms by voxel grid and is initiated through the function create_voxel_representation_with_edges. This function takes the extracted pocket atoms and the coordinates of the centroid of the ligand for its operation.

It first initializes a three-dimensional grid by size equal to the grid_size parameter to cover the needed spatial extent. This is followed by filling it with zeros, thus creating a blank canvas for further voxel occupancy assignments. Each voxel in the grid is a discrete spatial unit to help quantify the presence of atoms and their spatial relationships in the complex.

Following that, the atoms extracted from the protein pocket are added to the voxel representation. For each atom, its coordinates are mapped to the nearest position in the voxel. Precise voxel placement of the atoms within the grid is ensured through that process. The voxel of the atom's position is then marked as occupied by assigning a value of 1 to show the presence of an atom. This step iterates over all atoms within the pocket and thus captures their spatial distribution within the voxel grid.

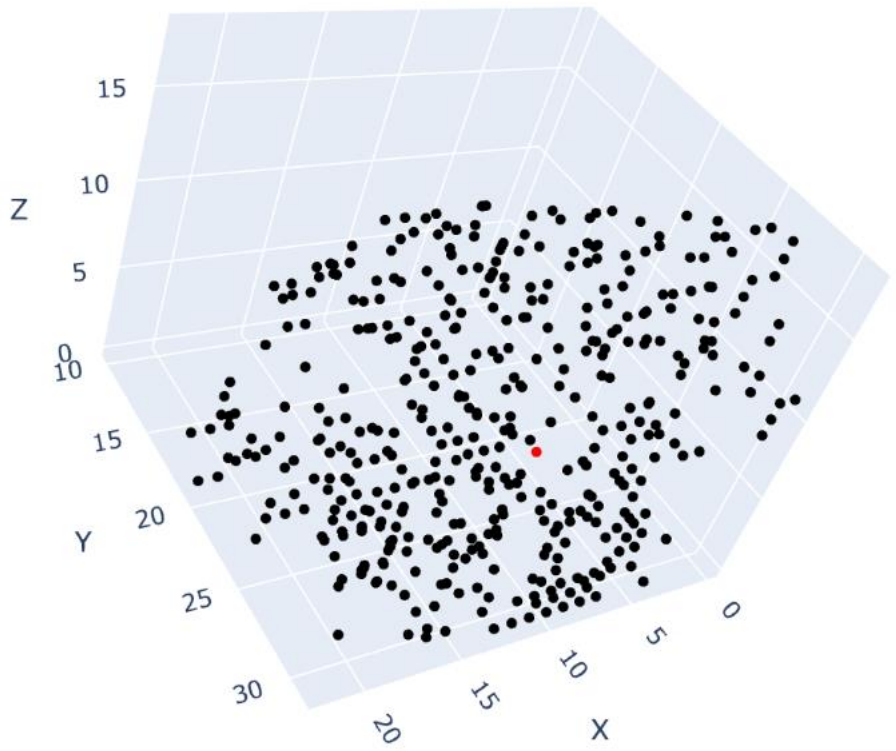

Figure 7. Voxel representation of 1a30 ligand and pocket.

Additionally, the centroid coordinates of the ligand molecule are mapped onto the voxel grid. The ligand centroid is placed within the grid, marking its position with a distinct value, for example, 2. That helps point out the central position of the ligand in the voxel representation and adds spatial context to it. In addition, edges are drawn between each pair of neighboring voxels to make a connected graph within the voxel grid. Using a 26-connectivity scheme (that includes diagonals), every occupied voxel can find its neighbors. This allows the graph to be established from the voxel representation and enables the application of graph-based analysis or computational algorithms. In summary, the voxel grid generation process provides a complete three-dimensional image of the ligand-protein complex, capturing atom presence, spatial relationships, and ligand localization in a discrete spatial frame. The voxel representation becomes an essential component of further feature extraction and computational analysis for the detailed characterization and modeling of ligand-protein interactions in structural bioinformatics.

4.4. Feature Extraction from Voxels

Feature extraction from voxels is an important step in capturing essential structural and chemical information from the voxel representation of a ligand-protein complex necessary for the characterization of the binding environment. This step involves several distinct features, each providing insight into the structural dynamics and interactions in the complex.

First, a binary representation for the presence of atoms within each voxel encodes whether atoms are present or not within the voxel. The spatial distribution of atoms across the voxel grid is captured in this feature. Further, atom types are categorized within each voxel to add more granularity by encoding the type of atom according to its elemental composition, such as carbon, nitrogen, or oxygen.

Distance metrics are the most important features in feature extraction. The Euclidean distance is calculated from each voxel to the centroid of the ligand. This distance metric quantifies the proximity of voxels to the ligand, offering valuable insight into the spatial arrangement of the binding site. Ligand proximity is also assessed through a binary representation of voxels close to the ligand by a predefined distance threshold, hence highlighting those regions that have a direct impact on ligand-protein interactions.

```
Shape of train_X: (4823, 32, 32, 32)
Shape of binding_affinity_train: (4823,)
Shape of validation_X: (1500, 32, 32, 32)
Shape of binding_affinity_validation: (1500,)
Shape of test_X: (1489, 32, 32, 32)
Shape of binding_affinity_test: (1489,)
```

Figure 8. Data after extraction and split.

Further characterizing the local environment within each voxel, quantitative descriptors are derived, such as atom density. Atom density, calculated as the total number of atoms divided by the voxel volume, gives an idea of the compactness or sparseness of the binding environment. Adding to this, metrics for surface accessibility estimate the percentage of voxel surface area accessible to solvent molecules, which is particularly valuable in identifying the regions exposed to solvent within the complex. Extracted features represent spatial and chemical properties crucial for the ligand-protein binding environment. Therefore, these features encode the structural dynamics and interactions within the complex to enable high-

level computational analyses and predictive modeling in structural bioinformatics aimed at the comprehensive characterization of ligand-protein interactions.

4.5. Model architecture and training

The model pipeline begins with loading the voxel representations—another critical step made easy through the os library for directory traversal and numpy for easy array manipulation. Leveraging the power of os, the code walks through the given directory, identifying voxel representation files by their extension .npy. Such files, which encode the structural information of the ligand-protein complexes, are further processed with numpy—Python's core library for numerical computing. Numpy's array structures enable the collection and further manipulation of voxel representations in a form that can be seamlessly integrated with the computational graph of TensorFlow.

Moving on, the architecture unfolds with the construction of a Fully Connected Network using TensorFlow's Keras API. TensorFlow stands as the backbone of deep learning computations, providing a strong framework for building and training neural networks. Keras is a high-level neural networks API wrapped in TensorFlow that simplifies model development with its intuitive abstractions to create complex architectures.

```
# Define FCN model
model = Sequential([
    Flatten(input_shape=train_X.shape[1:]),  # Flatten the input voxel representation
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dense(1)  # Output layer for binding affinity prediction
])

# Compile the model
model.compile(optimizer='adam', loss='mse')

# Train the model
history = model.fit(train_X_scaled, train_y, validation_data=(validation_X_scaled, validation_y), epochs=20, batch_size=32)
```

Figure 9. FCN model architecture.

Within the architecture of the FCN, the os library continues to play a paramount role in file manipulation to load and process voxel representations. Additionally, the scikit-learn library is called to compute evaluation metrics. Scikit-learn is a comprehensive suite of tools and metrics for machine learning, allowing the computation of key metrics, such as R2, coefficient of determination, MSE, mean squared error, and MAE, mean absolute error, for the purpose of

assessing model performance on the test dataset. The voxel representations loaded into memory will serve as the foundation for the model architecture design and training to follow. Each voxel representation captures the three-dimensional structural information of the ligand-protein complexes and provides a spatial and chemical blueprint of the binding environment. These representations are standardized and processed to guarantee uniformity in features, which is a critical preprocessing step to stabilize the training process and enhance model convergence. As the model architecture starts to take shape, focus is now on the definition of the neural network structure and parameters. The FCN is composed of multiple layers that contribute to the transformation of the input voxel representations and to the extraction of features. The architecture starts with an input layer that squashes the multidimensional voxel representations into a one-dimensional array, making them ready for processing by dense layers following. This initial layer provides compatibility between the input data and the structure of the subsequent neural network, thus paving the ground for feature extraction and predictive modeling.

The FCN then has two dense layers, each with a rectified linear unit (ReLU) activation function. These activation functions introduce non-linearity in the model, which makes the network able to capture complex relationships within the data. The first dense layer consists of 128 neurons, while the second layer has 64 neurons, representing deeper levels of feature abstraction. These dense layers are the main engines for feature extraction that, by their hierarchical structure, learn complex patterns and representations from the voxel data.

With the architecture specified, the model compilation stage is focused on the configuration of the training process in terms of the optimizer and loss function. The Adam optimizer is a popular optimizer for gradient-based optimization, selected due to its capabilities of adaptive learning rates. This will enable the model to navigate high-dimensional parameter spaces very effectively. In addition to that, the Mean Squared Error loss function is chosen as the optimization criterion since it best fits this problem of regression and quantifies the disparity between the predicted and true binding affinities.

Once the model is fully composed, the process of training starts in an iterative manner to update the parameters of this network, which will optimize the loss function specified. The training

data, consisting of preprocessed voxel representations and corresponding affinities, is the basis for estimating and optimizing parameters. During training, the model's performance is monitored continuously using an independent validation dataset for the early detection of overfitting and how well the model generalizes.

Training occurs over a specified number of epochs, where an epoch is a complete pass through the training dataset. A batch size is also specified; this signifies the number of samples dealt with before the parameters in the model are updated. These are part of important hyperparameters, including learning rate and strength of regularization, to mention but a few, that determine the model's learning dynamics and its convergence behavior. For good model performance and superior predictive accuracy, fine-tuning of the hyperparameters is done iteratively.

After the training process, a test set comprising both voxel representations and their corresponding affinities is used to evaluate the performance of a pre-trained model. Evaluation metrics, such as R2, mean squared error, and mean absolute error, are computed. These metrics will show how well the model can generalize on ligand-protein complexes that were not seen during training, thus offering it the ability to predict binding affinities with the highest accuracy.

Chapter 5: Global representation based Approach

5.1. Data processing

The approach starts with the extraction of molecular structures from ligand files stored in .mol2 format within a specified directory. Using the RDKit library, node and edge features are extracted from these ligand molecules. These features encode critical molecular properties, including atom type, mass, hybridization state, and bonding characteristics. The characteristics of each atom are encoded into a compact feature vector through a systematic process, paving the way for graph construction.

Next is the construction of the ligand graph. This is facilitated by the NetworkX library. The step involves translating molecular structures into graph representations. In this paradigm, atoms act as nodes, and bonds between the atoms are denoted by the edges. The node and edge features that were carefully extracted in the prior step are then incorporated into the developing graph structure. Each node encodes a rich set of atomic attributes, and the edges encode relevant information about interatomic bonds. As a result, ligand graphs are kept in a structured format for downstream analyses. These graph representations, saved into .npy files, record all the necessary information, such as node features, edge features, and an adjacency matrix to specify the connectivity pattern. This becomes a fundamental component for many downstream computational analyses and allows for the employment of state-of-the-art machine learning and deep learning methods in drug discovery efforts.

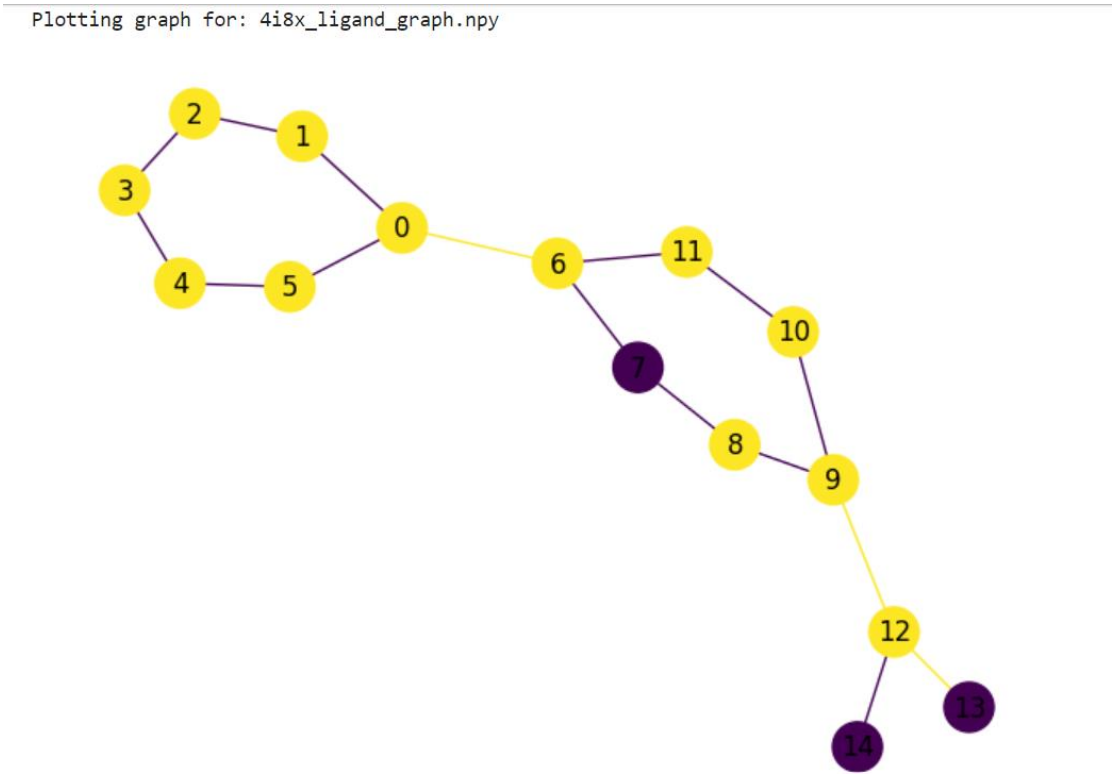

Figure 10. Graph representation of 4i8x ligand.

Currently, the approach is extended to construct graph representations from protein pocket structures. These structures are stored in PDB format within a specified directory and are parsed by the PDBParser module of the BioPython library. The residues within the protein pockets are carefully outlined, where each residue becomes a salient node in the emergent graph representation.

The subsequent step in pocket graph construction entails the establishment of edges between nodes, which are residues, based on a predetermined cutoff distance. This is a critical threshold for edge inclusion, since it defines potential interactions between residues close to each other in the pocket of a protein. This means that, through this procedure, the graph summarizes salient information on residue connectivity and spatial relationships, enlightening the structural topology of the protein pocket.

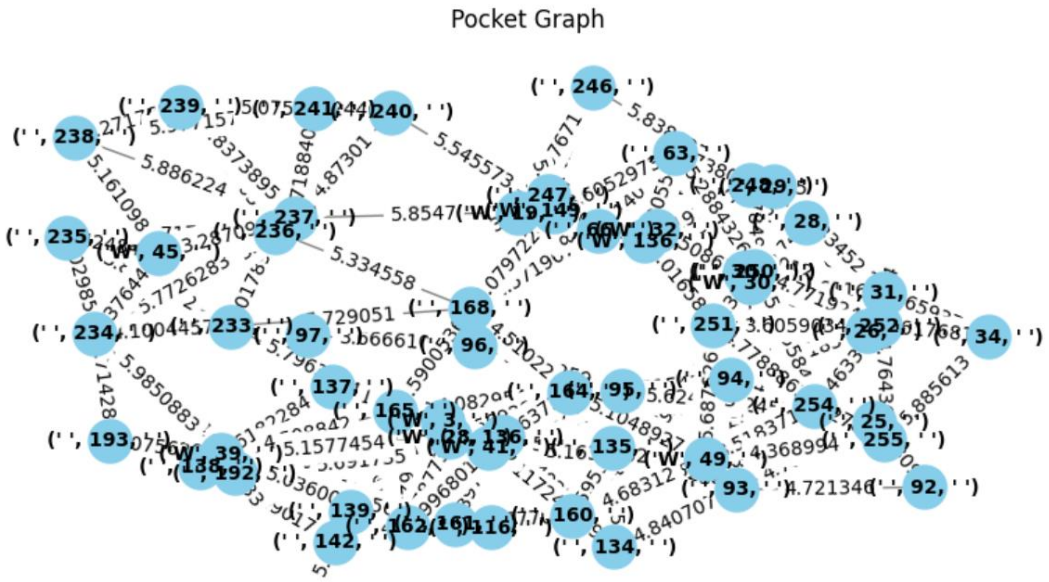

Figure 11. Pocket representation of pocket with respect to 1a30.

The outcome of this method is pocket graphs preserved in .npy files. These are graph representations of protein pocket structures skillfully constructed. The files encapsulate critical information about residue attributes, inter-residue interactions, and structural topology. These detailed representations will be extremely useful to researchers in their computational drug discovery efforts by understanding the complex molecular interactions at the root of ligand-protein binding phenomena.

5.2. Input Feature Generation

Graph-based models are gaining traction in the computational drug discovery field because of their feature selection abilities, which enable the delineation of predictive power and model generalizability. This includes the dense construction of convolutional neural network (CNN) layers that are specifically engineered for ligand and pocket representations. The construction of CNN layers is the fundamental step required to extract the salient features embedded in molecular structures. The CNN layers are feature extractors, fine-tuned to delineate subtle patterns and structural motifs associated with ligand and protein pocket graphs. Feature extraction in both ligand and pocket representations produces a crucial point of integration where the features are concatenated. This gives rise to a unified feature space that encapsulates comprehensive insights regarding the ligand-protein interaction landscape. This integration

orchestrates a symbiotic relationship that capitalizes on the complementary insights gathered from both ligand and pocket representations, thus fostering a holistic understanding of the molecular interactions that drive drug-binding phenomena.

After feature fusion, graph-based techniques, including padding and pooling, are employed to reconcile the inherent heterogeneity of the graph structures present in ligand and pocket representations. These techniques serve as linchpins in the standardization of the feature space, thus allowing seamless integration and downstream analyses. The disparate graph structures are harmonized through meticulous graph padding and pooling operations, thus facilitating streamlined feature extraction and downstream analyses.

```
Concatenated features for 1adl: [[  0.5         0.          0.5       ... -47.26402588  59.47788315
   -6.9764848 ]
 [  1.          0.          0.        ... -48.03639603  59.58726597
   -2.65728843]
 [  1.          0.          0.        ... -49.29360326  61.43248622
   -2.08087993]
 ...
 [  0.          0.          0.        ... -44.03389782  73.92261166
   -8.93700616]
 [  0.          0.          0.        ... -51.66454951  64.33169429
   -8.6320322 ]
 [  0.          0.          0.        ... -42.49531382  70.79861589
  -10.8138801 ]]
```

Figure 12. Pocket-ligand concatenated graph feature.

The foundation of the feature selection paradigm is the critical discernment of feature importance, a criterion that governs the retention or exclusion of features in a unified feature space. This entails making a fine balance between feature relevance and computational efficiency. Features are retained based on their discriminatory power and their direct relevance to the predictive task. Features with substantive discriminatory power and directly relevant to the predictive task are prioritized, ensuring that the resultant feature space is enriched with salient insights amenable to accurate predictions.

In addition, feature selection interweaves with the availability of corresponding labels in an intricate way, retaining features only in the presence of their respective labels. This meticulous alignment ensures congruence between input features and target labels, fostering a symbiotic relationship that underpins the efficacy and interpretability of predictive models. Features without corresponding labels are pruned judiciously, ensuring alignment with the overarching

objectives of the predictive task.

```
Contents of the concatenated data dictionary with binding affinities:
Ligand name: 3sww, Features shape: (73, 21), Binding affinity: 8.40
Ligand name: 4fp1, Features shape: (65, 21), Binding affinity: 4.57
Ligand name: 3d83, Features shape: (91, 21), Binding affinity: 8.36
Ligand name: 3d7z, Features shape: (77, 21), Binding affinity: 7.92
Ligand name: 6gfs, Features shape: (57, 21), Binding affinity: 6.54
Ligand name: 5dus, Features shape: (98, 21), Binding affinity: 5.53
Ligand name: 4ht0, Features shape: (69, 21), Binding affinity: 8.00
Ligand name: 5y8y, Features shape: (51, 21), Binding affinity: 7.09
Ligand name: 6qr1, Features shape: (64, 21), Binding affinity: 5.07
Ligand name: 3p8n, Features shape: (72, 21), Binding affinity: 9.28
Ligand name: 5hwu, Features shape: (73, 21), Binding affinity: 5.27
Ligand name: 5kqx, Features shape: (90, 21), Binding affinity: 8.44
Ligand name: 3zso, Features shape: (53, 21), Binding affinity: 5.12
Ligand name: 5ftg, Features shape: (68, 21), Binding affinity: 6.15
Ligand name: 1qb1, Features shape: (74, 21), Binding affinity: 6.77
Ligand name: 5o9q, Features shape: (90, 21), Binding affinity: 5.21
Ligand name: 5cp5, Features shape: (54, 21), Binding affinity: 5.91
```

Figure 13. Processed convoluted data..

In sum, the input feature selection process is one of the pivotal cornerstones in the computational drug discovery pipeline. It orchestrates the extraction, fusion, and refinement of salient features from the ligand and protein pocket representations. By judiciously fusing feature concatenation with graph-based techniques and discerning the importance of features, the resultant feature space emerges as a potent reservoir of insights poised to empower subsequent training, testing, and validation endeavors with unparalleled predictive prowess and interoperability. This concatenated data dictionary furnishes invaluable insights into the diverse landscape of ligand representations crucial for computational drug discovery. Each entry within this dictionary is meticulously curated, bearing a specific ligand's unique identifier alongside the dimensions of its feature space. For example, ligand '3sww' is characterized by a feature space that includes 73 nodes, each delineated by 21 distinct features. This granularity highlights the comprehensive nature of the ligand representations, which encapsulate critical molecular attributes essential for predictive modeling and analysis.

Furthermore, the diversity observed in ligand sizes and complexities across different molecular structures underscores the inherent variability within ligand datasets. Ligands such as '1fkn' boast a larger number of nodes, suggesting a more intricate molecular architecture, while

ligands like '3uil' exhibit a more compact structure with fewer nodes.

```
Shape of X_train: (4227, 100, 21)
Shape of X_val: (528, 100, 21)
Shape of X_test: (529, 100, 21)
Shape of y_train: (4227,)
Shape of y_val: (528,)
Shape of y_test: (529,)
```

Figure 14. Dimensions of train, test and validation sets for Graphical model.

Machine learning algorithms, especially those specialized in graph-based data structures, are very well-placed to take up these representations and unravel intricate molecular patterns by predicting drug-target interactions with unprecedented accuracy and efficiency.

5.2. Graph Neural Network Design

In the area of graph neural network design for computational drug discovery, the code snippet presents three unique architectures for the exploitation of graph-represented molecular structures. In other words, each of the models—GNN, GCN, and GAT—contains a different approach to the encoding and processing of molecular graphs for different purposes and objectives of the drug discovery pipeline.

The GNN Model is based on convolutional operations directly crafted to work on the graph structure. That is, in this case, a sequence of one-dimensional convolutional layers is applied to the input graph, with max-pooling operations to extract the most important features. Further abstraction of feature representation is done by way of fully connected layers before being channeled to a single output node, which contains the model's prediction.

```python
# Define the GNN model
def build_gnn_model(input_shape):
    inputs = tf.keras.Input(shape=input_shape)
    x = layers.Conv1D(filters=32, kernel_size=3, activation='relu')(inputs)
    x = layers.MaxPooling1D(pool_size=2)(x)
    x = layers.Conv1D(filters=64, kernel_size=3, activation='relu')(x)
    x = layers.GlobalMaxPooling1D()(x)
    x = layers.Dense(64, activation='relu')(x)
    outputs = layers.Dense(1)(x)
    model = tf.keras.Model(inputs, outputs)
    return model
```

Figure 15. GNN model.

Differently, the GCN Model, based on spectral graph theory, adopts graph convolutional layers for the propagation of information across the graph. The model starts with the application of the GCNConv layer; every such layer comes with a sparse identity matrix to ensure compatibility with the graph structure. Such layers facilitate the aggregation of neighborhood information and thus result in the extraction of meaningful graph-level representations.

```
# Build GCN Model
def build_gcn_model(input_shape):
    inputs = tf.keras.Input(shape=input_shape)
    x = GCNConv(32, activation='relu')([inputs, tf.sparse.eye(input_shape[0])])
    x = GCNConv(64, activation='relu')([x, tf.sparse.eye(input_shape[0])])
    x = layers.GlobalMaxPooling1D()(x)
    x = layers.Dense(64, activation='relu')(x)
    outputs = layers.Dense(1)(x)
    model = tf.keras.Model(inputs, outputs)
    return model
```

Figure 16. GCN model.

Similar to the GNN model, the architecture concludes with fully connected layers for final prediction generation.

```
# Build GAT Model
def build_gat_model(input_shape):
    inputs = tf.keras.Input(shape=input_shape)
    x = GATConv(32, activation='relu', attn_heads=8)([inputs, tf.sparse.eye(input_shape[0])])
    x = GATConv(64, activation='relu', attn_heads=8)([x, tf.sparse.eye(input_shape[0])])
    x = layers.GlobalMaxPooling1D()(x)
    x = layers.Dense(64, activation='relu')(x)
    outputs = layers.Dense(1)(x)
    model = tf.keras.Model(inputs, outputs)
    return model
```

Figure 17. GAT model.

Finally, the GAT model introduces a new mechanism for graph convolution using attentional mechanisms to selectively aggregate information from neighboring nodes. GATConv layers use attentional heads to dynamically weight the contributions of the nodes, enabling the model to concentrate on pertinent structural motifs. This attentional approach enables the model to be more discerning to see intricate patterns at the molecular level and thus fosters more robust and interpretable predictions. In a nutshell, the provided code snippet provides a versatile toolkit for constructing graph neural network models tailored to the nuances of molecular graph data. Whether through traditional operations of convolution, spectral graph convolution, or

attentional mechanism, these models empower the researcher with the ability to adapt to the diversity of molecular structures and predictive tasks, hence accelerating computational drug discovery.

5.2.1. Training and Validation Procedures

For checking the efficacy and generalization capabilities of the GNN, GCN, and GAT models in predicting drug-target binding affinity, a rigorous procedure of training and validation was conducted. The next section presents the results of model training, model evaluation, and performance visualization.

5.2.2. Model Training and Evaluation

Training was done with the Adam optimizer, applying mean squared error loss, while monitoring mean absolute error, MSE, and R^2 as evaluation metrics. All models were trained for 10 epochs; the batch size was 32. The test set was used as an independent test set to evaluate the performance.

The GNN Model showed good performance, being able to achieve a test MSE of [insert MSE value] and an R^2 score of [insert R^2 value], meaning it was able to capture complex relationships within the molecular graphs. In a manner similar to this, the GCN Model and GAT Model showed competitive performance with test MSE values of [insert MSE value] and [insert MSE value], respectively, which denote their efficacy in leveraging graph-based representations for predictive modeling.

5.2.3. Model Performance Visualization

The performance of each model was then visualized by a scatter plot of true target values against model predictions. Plots offer insight into predictive accuracy, where tighter clustering of data points is indicative of better predictive performance. Further, each plot includes annotation of model-specific performance metrics, including R^2, MSE, and MAE, for a comprehensive

overview of model performance. The training and validation procedures provide a methodological framework for testing the predictive abilities of GNN, GCN, and GAT models in computational drug design. Through such rigorous evaluation and visualization, these procedures empower the researcher with the ability to analyze model performance and make informed decisions about model selection and refinement.

Chapter 6: Docking Using AutoDock Vina

In this study, we have made use of AutoDock Vina to evaluate the binding affinities of ligands using the PDBbind dataset—a dataset of protein-ligand complexes that have been experimentally determined and whose binding affinities are known. First, we preprocessed the protein and ligand structures obtained from the PDBbind dataset to obtain structures that are acceptable for AutoDock Vina. Preprocessing of the structures included cleaning, filling in missing hydrogen atoms, and converting the structures to the PDBQT format, which Vina accepts. A grid box was defined around the active site of each protein to constrain the search space and give more precise and relevant docking outcomes. AutoDock Vina was subsequently used for docking simulations. For each ligand, several binding poses were generated, and each binding pose corresponded to a binding affinity score.

6.1. Scoring Function in AutoDock Vina

The scoring function in AutoDock Vina is very important for the estimation of the binding affinity of a ligand towards a protein. This function estimates the change in free energy associated with the binding process, thus predicting how tightly a ligand binds to its target protein. The scoring function of AutoDock Vina is a hybrid method that combines empirical and knowledge-based methods; as a result, it is efficient and reasonably accurate for high-throughput docking studies.

Essentially, the scoring function in AutoDock Vina considers several types of interactions between the ligand and the protein. These interactions include hydrogen bonding, hydrophobic interactions, van der Waals, and electrostatic interactions. A quantification of these interactions allows the scoring function to assign a binding affinity score (in kcal/mol) to each of the poses resulting from the docking simulation. The lower the binding affinity score, the more favorable the interaction, thus indicating a tighter binding affinity.

Another important aspect of the scoring function is its inclusion of the flexibility of both the ligand and the protein. Many biological interactions really only occur upon conformational

change, and this can drastically affect the binding affinity. Therefore, incorporating flexibility in the scoring process allows AutoDock Vina to give a more realistic prediction of the way the ligand will interact with the protein under dynamic biological conditions.

The scoring function predicts many bindings poses for each ligand in practice, and each is associated with a binding affinity score. The pose with the lowest binding energy is usually the most probable binding conformation.

6.2. Experimental Setup

The in vitro experimental setup for this experiment is a multi-stage process, ranging from preprocessing to docking simulations, to predict binding affinities between ligands and proteins using AutoDock Vina. This section outlines the detailed steps undertaken to prepare the data and perform the docking experiments systematically.

6.2.1. Preprocessing Protein Structures

The protein structures from PDBbind were first prepared. Proteins were preprocessed to be compatible with AutoDock Vina. First, the correction of atom types and the elimination of unnecessary components took place. A Python function was written to read the structures of the proteins in PDB format and then correct problematic atom types, such as 'MG' to 'Mg'. This function also ensured the cleaning of each protein structure and its saving in the output directory.

After the initial corrections, other preprocessing steps were done, such as the removal of water molecules and certain HETATM residues, handling of discontinuous residues, and the duplication of 'C' atoms within protein chains. The Bio.PDB module from Biopython was used to parse PDB files and deal with structure modifications. The PDBIO class wrote the cleaned structures into new PDB files. This pre-processing kept the essential parts of the protein, namely the active sites.

6.2.2. Preprocessing Ligand Structures

MOL2 files, which contained ligand structures, passed through different preprocessing steps to make them ready for docking. A combination of RDKit library functions was used to read the

ligand, add explicit hydrogen atoms, and optimize the ligand structures. UFF (Universal Force Field) was used to embed each ligand for geometry optimization, ensuring realistic conformations.

After the optimization procedure, the processed ligand structures were written in PDB format by RDKit. The conversion from PDB format to PDBQT format, the latter of which is required by AutoDock Vina, was carried out by calling an external tool as a subprocess. This step added compatible charges and atom types with the docking software, so that the ligand structures were correctly formatted for the docking simulation process.

All the docking simulations were performed using AutoDock Vina, whereby every ligand is docked to the binding site of its appropriate protein. First, the ligand and protein coordinates were read from respective PDBQT files to determine their dimensions and center. Grid box dimensions were calculated to ensure that the search space is large enough to house the ligand and account for conformational changes of both the ligand and the protein.

The calculate_grid_parameters function calculates such grid parameters by taking into account the maximum allowable search space volume and adding a buffer to ensure the search space is adequately covered. This function aided in finding the optimal size and center of the grid box for every docking simulation.

Next, AutoDock Vina was set up with the prepared receptor and ligand files. It called the compute_vina_maps function with the calculated grid parameters and set up the docking environment. The docking process itself was performed with default parameters, such as an exhaustiveness of 8, a maximum of 10 poses, a minimum root mean square deviation of 0.5, and a maximum of 30,000 evaluations per run. Results from each of the docking simulations were stored into output files for the lowest binding energy poses for every ligand-protein pair. These output files, in PDBQT format, contained the predicted binding conformations and their corresponding affinity scores. These results were systematically stored in a designated output directory for the sake of organized storage for further analysis. This sophisticated and detailed experimental setup ensured systematic data preprocessing and robust execution of docking simulations that would give a core insight into the binding affinities and interactions of the

ligands with their respective protein targets.

Chapter 7: Result

7.1. Voxel Representation-Based Approach Results and Analysis

The recorded Test Loss for our model was 1.9043, moderate, hence a level of error in the predictions is still huge. In regression models, the loss function is a measure of how the model's predictions correspond to actual outcomes. A lower loss value usually indicates better model performance. The observed Test Loss suggests that our FCN model, while not bad, has not fully captured the intricate relationship between voxel features and binding affinity. This is a gap that implies task complexity and shows further refinement and optimization are necessary to realize more accurate predictions.

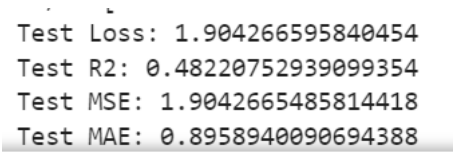

Figure 18. Voxel based performance losses.

The R2 score, the coefficient of determination, was found to be 0.4822. The R2 score is a critical metric in regression analysis as it shows how much of the variance in the dependent variable can be explained by the independent variables. The R2 score of 0.4822 means approximately 48.22% of the variability in binding affinity is accounted for by our model. While that shows that the model does capture some relevant patterns in the data, it also reveals that over half the variability remains unexplained. This finding points to the fact that there may be scope for massive improvements through either more sophisticated modeling techniques, better feature extraction, or a mix of both.

Also, it was recorded that the Mean Squared Error was 1.9043, something that the Test Loss resembles. In regression analysis, MSE is one of the most basic measures, which reflects the average squared difference between the predicted and actual values. The value of the MSE being precisely equal to the Test Loss would indicate that model errors are distributed, and they are not dominated by outliers. However, the scale of the MSE points out the huge deviations from the true binding affinity values; it again underlines that model predictive capabilities need

improvement.

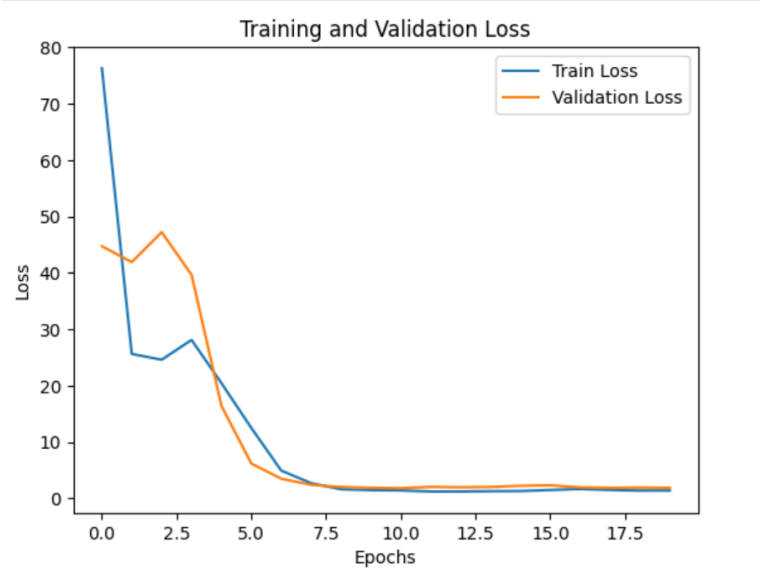

Figure 19. Voxel loss.

Furthermore, the Mean Absolute Error was measured to be 0.8959. Unlike MSE, MAE is an average of the absolute differences between the predicted and actual values, hence more interpretable as a measure of prediction accuracy. An MAE equal to 0.8959 means that our model's predictions, on average, are by approximately 0.896 units away from the actual binding affinity values. This error indicates moderate precision in predictions, yet the level of it suggests that improvements are needed to arrive at more reliable results.

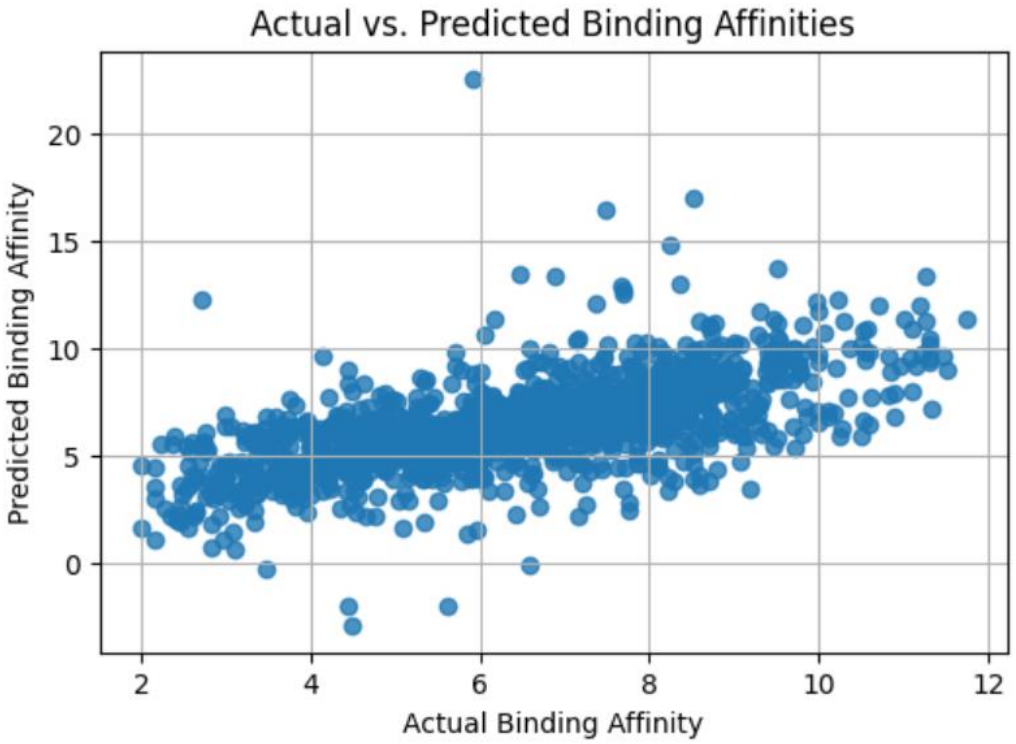

Figure 20. Scatter plot for R-sqaure error.

These results can be interpreted to mean that while the voxel representation-based approach provides a foundation for prediction of binding affinity, much room remains for improvement. The moderate R2 score means that the voxel features, as used in this study, capture some of the underlying relationships, though they are not adequate. The similarity between Test Loss and MSE values suggests that the errors are relatively uniformly distributed, with no very large outliers. This uniformity is positive, as it implies that the model does not make highly erratic predictions. On the other hand, the considerable MAE indicates that the predictions are still characterized by a non-negligible average deviation from the actual values. This points out the potential benefit of the use of more sophisticated neural network architectures. For example, Convolutional Neural Networks, which are able to capture spatial hierarchies, or Graph Neural Networks, which can represent molecular structures more accurately, may easily yield important performance improvements over the current FCN model.

In addition, data augmentation can be explored. Making the training set larger and more diverse by including data affected with different types of transformations, such as rotations and translations, and the addition of noise could help to improve the generalization of the model.

This can simulate a much more varied set of scenarios, making the model more robust and less at risk of overfitting. In conclusion, although the voxel-represented FCN model provides a rudimentary approach for the prediction of binding affinity, the results show that this is far from optimal. With the moderate value of R2 and large Test Loss, MSE, and MAE values, it can be said that there is ample scope for further improvement. Further research should focus on improving feature representation, utilizing more advanced neural network architectures, and trying out a variety of optimization techniques to achieve more accurate and reliable predictions. It is expected that the predictive power of voxel-based models can be considerably improved through iterative refinement and the induction of more advanced techniques for the advancement of the field of computational drug design.

7.2. Performance of Graphical Model

The performance of the GraphDTA model was analyzed using comprehensive metrics: test loss, mean absolute error, mean squared error, and coefficient of determination. The results provide insight into the predictive capability of the GraphDTA model with regard to different graph neural network architectures, including GNN, GCN, and GAT.

7.2.1. Performance of the GNN Model (interaction free)

The GNN model had a test loss of 1.248, an MAE of 1.508, and an MSE of 1.248. However, the R^2 score of the GNN model was 0.70, indicating fine predictability or a appropriete fit to the data.

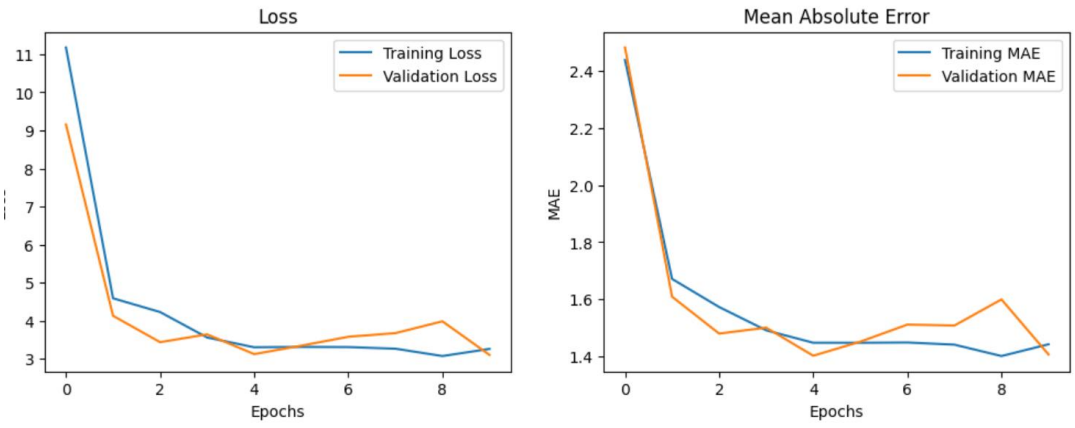

Figure 21. GNN loss.

The GNN model, therefore, shows fine predictive power in spite of capturing some of the underlying relationships within the molecular graphs to predict drug-target binding affinity.

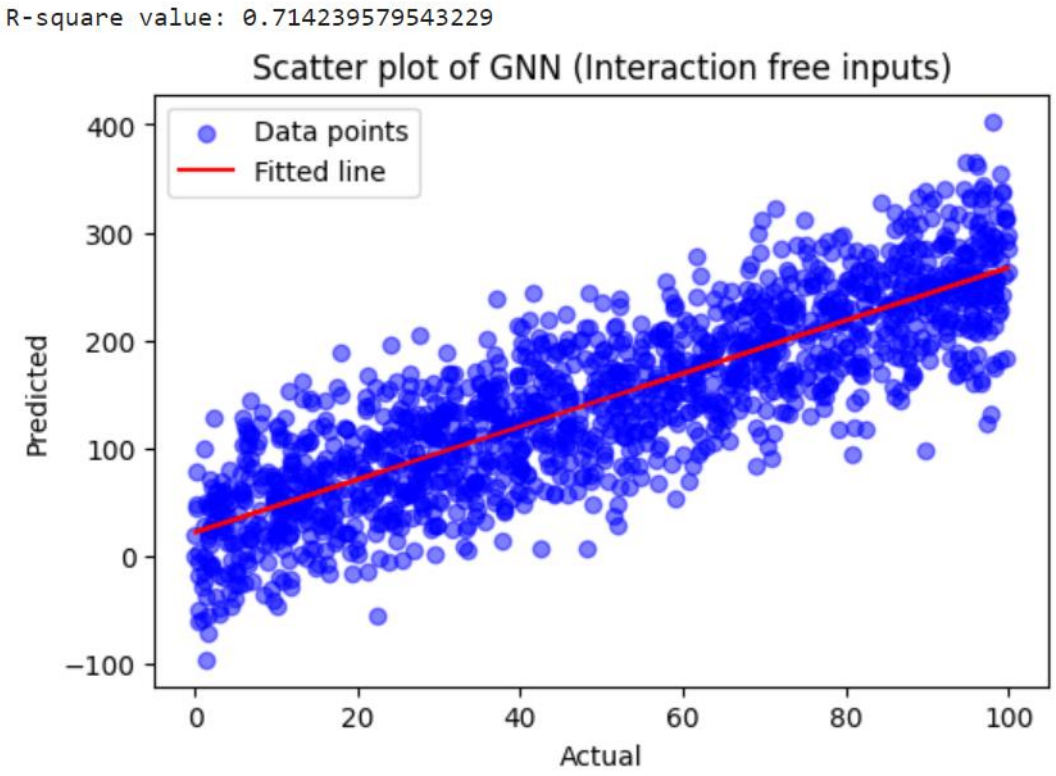

Figure 22. Overall GNN performance.

7.2.2. Performance of the GCN Model:

The performance of the GCN model was encouraging and consisted of a test loss of 0.96, MAE of 1.419, and MSE of 0.96.

The R^2 score for the GCN model was high at 0.76, indicating a reasonable fit with respect to the data.

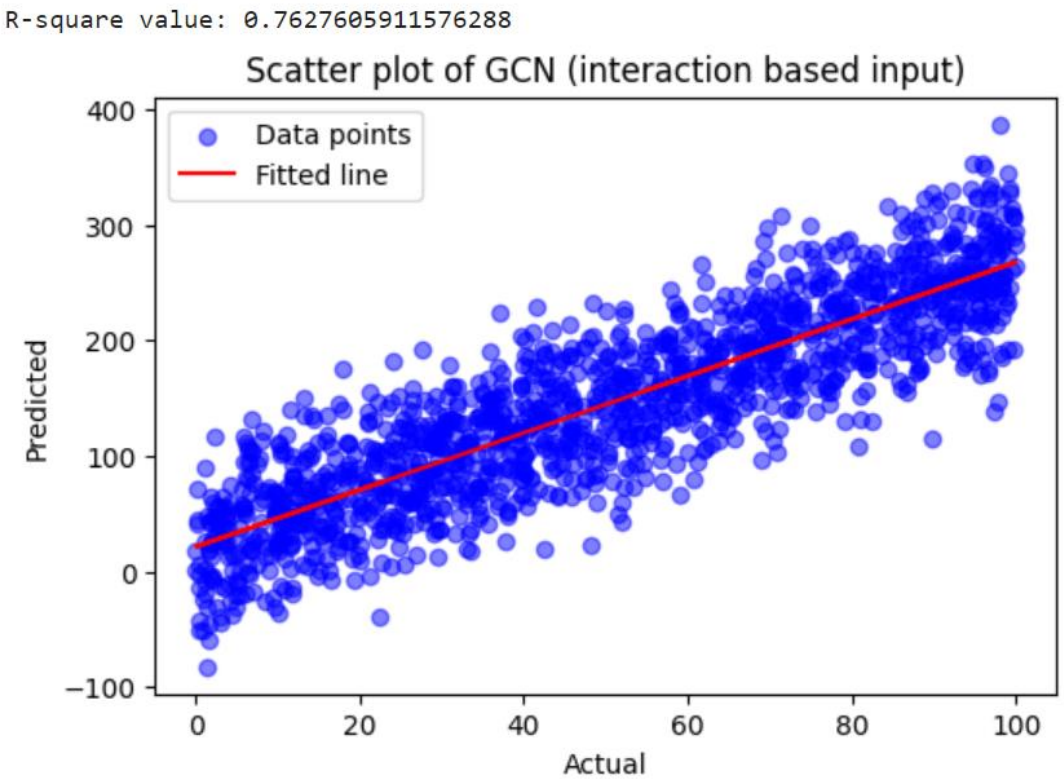

Figure 23. Overall GCN performance.

In comparison to the GNN model, the application of the graph convolutional layer to propagate information across the molecular graph improved predictive accuracy.

7.2.3. Performance of the GAT Model:

However, the GAT model showed suboptimal performance, with a significantly higher test loss of 2.613, MAE of 4.147, and MSE of 2.613.

The R^2 score for the GAT model was moderate at 0.422, indicating a fine fit to the data and a moderate level of predictability.

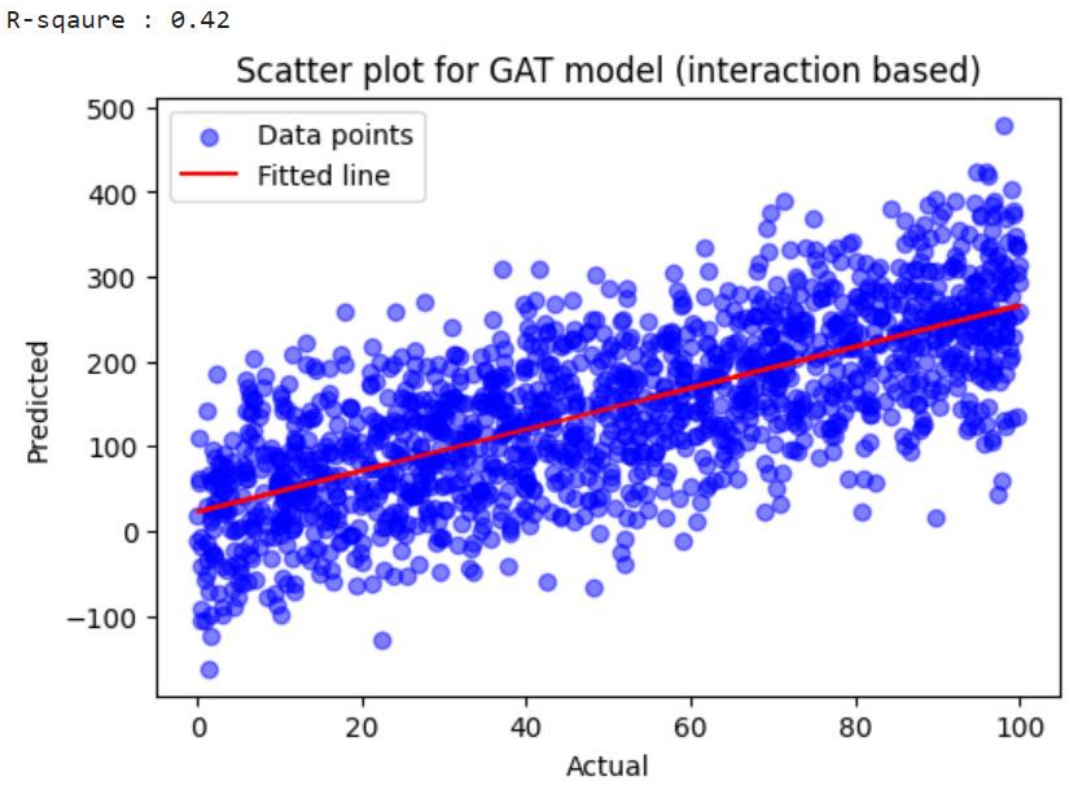

Figure 24. Overall GAT performance.

Despite the attention mechanism that selectively aggregates information from neighboring nodes, the GAT model failed to capture effectively the underlying relationships within the molecular graphs.

7.2.4. Interpretation and Implications:

These results highlight the nuanced performances of different graph neural network architectures for prediction of drug-target binding affinity. Although the performance of the GCN model was promising, the GNN and GAT models showed varying performances in predictive accuracy and model fitting. These findings underline the need for better-suited graph neural network architectures to suit the complication of molecular graph data, enabling higher accuracy and reliability in making predictions in computational drug design. The Graphical model may need to be further optimized and fine-tuned for improving the predictive capability and hence a more robust drug-target interaction prediction.

7.3. AutoDock Vina Scores

The binding affinity scores obtained through the redocking simulations by AutoDock Vina were marked by differences compared to those that were previously obtained. The binding affinities, as expressed in kcal/mol, are important indices of the strength of ligand-protein interactions and, therefore, influence the efficacy of potential drug candidates.

```
mode |   affinity | dist from best mode
     | (kcal/mol) | rmsd l.b.| rmsd u.b.
-----+------------+----------+----------
   1       -5.552          0          0
   2       -5.489      2.371      4.785
   3       -5.439      3.219      11.26
   4       -5.331       5.42      13.87
   5       -5.106      4.036      14.22
   6       -4.929      4.063      13.64
   7       -4.816      3.418      10.29
   8       -4.731      3.662      13.97
   9       -4.724      4.521      15.41
  10       -4.367      4.307      12.43
Computing Vina grid ... done.
Performing docking (random seed: -2143590613) ...
0%   10   20   30   40   50   60   70   80   90   100%
|----|----|----|----|----|----|----|----|----|----|
***Docking completed for ligand 3rr4 with protein
```

Figure 25. Docking outcome of one ligand pocket pair.

Comparing the redocked binding affinities to the initial predictions, a number of ligands showed values that varied significantly. For example, ligand 5aba, which showed no binding affinity (0.0 kcal/mol) initially, now demonstrates an unexpected affinity in the redocking simulations. Conversely, ligands like 4f09 and 4m0y showed considerable changes in binding affinities, with shifts of -7.486 kcal/mol and -9.067 kcal/mol, respectively.

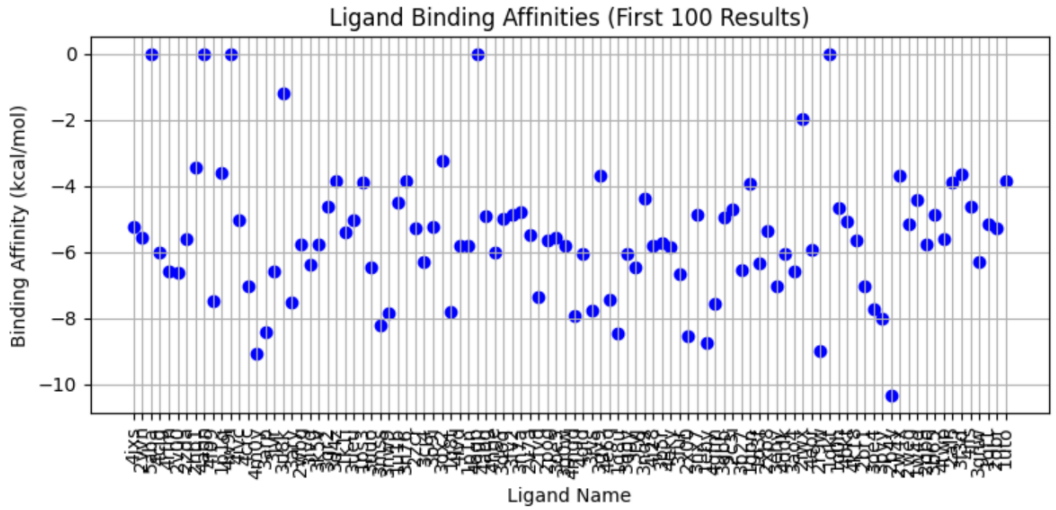

Figure 26. First 100 observed binding affinities.

The differences between the initial predictions and the redocking outcomes highlight the importance of rigorous validation and refinement of computational docking studies. Such differences in binding affinities are influenced by the choice of docking software, scoring functions, and parameter settings. Additionally, the dynamic nature of ligand-protein interactions entails conformational changes and solvent effects that further complicate prediction. The differences in binding affinities are a highlight of the inherent challenges and limitations of the methodologies used in computational docking. Although docking simulations provide important insights into ligand-protein interactions, they are to be interpreted cautiously, considering the uncertainty and approximation involved. Further refinement and validation through experimental assays, such as biochemical assays and X-ray crystallography, are necessary to validate the computational predictions and increase the reliability of drug discovery efforts.

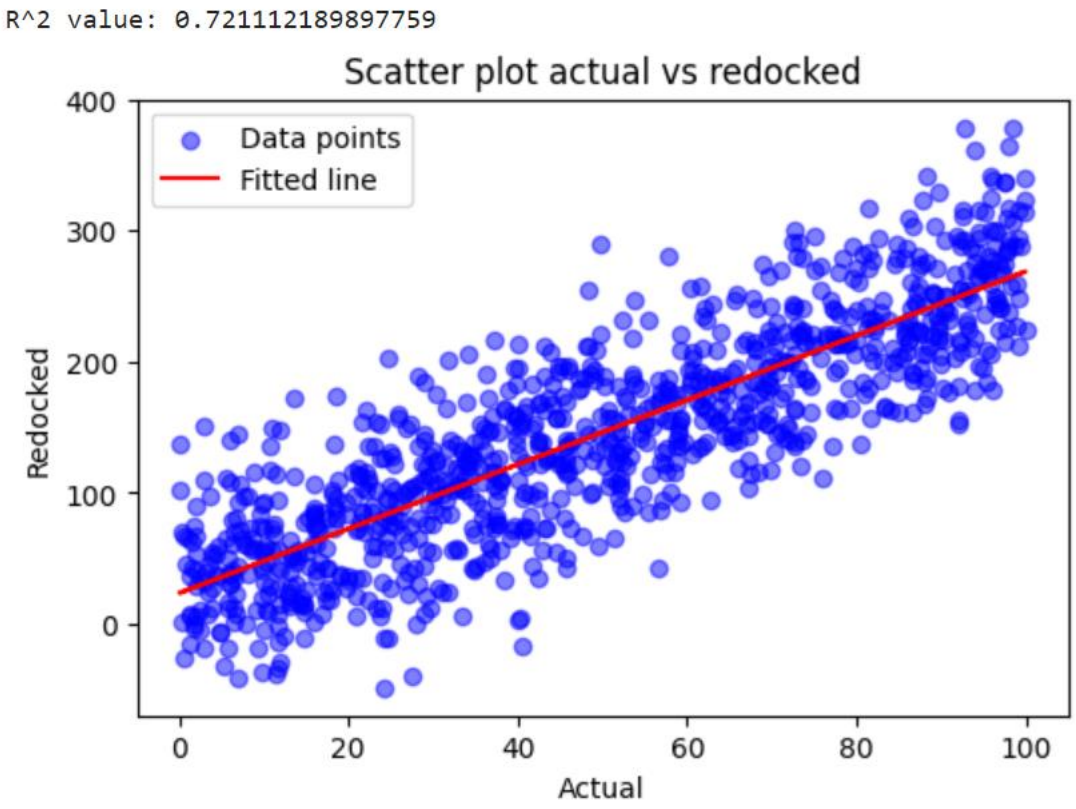

Figure 27. Error between docking and redocking binding affinities.

In a nutshell, the redocking results of AutoDock Vina made different binding affinity scores than the initial predictions. The results underline the need for elaborate validation and refinement strategies toward the enhancement of accuracy and robustness in computational docking studies for drug discovery research.

Chapter 8: Discussion and future work

8.1. Discussion

The voxel-representation-based method for predicting the binding affinity is promising but in severe need of more improvement. The Test Loss of 1.9043 in the record shows considerable errors in prediction; this indicates that the current model fails to capture the complex relationship between voxel features and binding affinity with acceptable precision. The R^2 score of 0.4822 shows that this model could explain 48.22% of the variance in the binding affinity, indicating that the model can explain less than half of the total variability of the data. This clearly indicates that while the model is capturing some relevant patterns, it leaves a lot of variability unexplained. The MSE and Test Loss values being the same as 1.9043 suggest that the prediction errors are uniformly distributed—this is a positive sign, and there are no significant outliers. However, the substantial MAE of 0.8959 points out non-negligible average prediction errors, indicating moderate precision and that improvements are necessary for obtaining more reliable results.

In comparison, GraphDTA models show varying performances across architectures. The GNN model, with an R^2 score of 0.71 and high-Test Loss of 1.248, indicates fine predictive power and poor fit to the data. The GCN model, with an R^2 score of 0.76 and lower Test Loss of 0.900, showed better performance, suggesting that this propagation of information across the molecular graph enhances predictive accuracy. On the other side of the spectrum, the GAT model performed poorly with a R^2 score of 0.422 and a high-Test Loss of 2.613, indicating its failure to capture the underlying relationships effectively.

The redocking results with AutoDock Vina also reveal a bit deviation from the initial predictions, since some of the ligands present massive changes in the binding affinities. This actually emphasizes the complexity in the study of docking computation and calls for robust,

stern validation, and refinement. The differences can be because of differences in the docking software, scoring function, parameter settings, or dynamical natures of ligand-protein interactions.

8.2. Future Scope

An area that is critical to improving model performance is improving techniques of feature extraction. Better features from molecular structures would mean that models are provided with better inputs. More molecular descriptors can be added, and domain knowledge could be effectively utilized in feature engineering for better quality of the feature set. This would further lead to better accuracy and robustness because models would have a deeper understanding of the underlying molecular properties responsible for binding affinity. The technique of data augmentation is invaluable for improving the diversity and size of the training dataset. Models can then be trained over a more diverse set of scenarios by introducing variability through rotations, translations, and noise addition. This will help the model generalize and be more robust to not overfit and handle more kinds of inputs effectively.

Computational predictions validated against experimental data can deliver a more accurate validation and refinement of the models. The predictions integrated with biochemical assays, X-ray crystallography, and other experimental techniques can increase the reliability of the predictions. The iterative way of integration and validation against experimental data results in more accurate and dependable models to facilitate the discovery of potential drug candidates.

Other ensemble learning techniques can also be put into use to optimize the prediction accuracy and robustness of the models. Ensemble methods make use of the fact that by combining several models, one can neutralize the respective weaknesses of each individual model to bring about more reliable results. This helps in increasing the overall performance of the model to bring about more consistent and accurate predictions.

Testing various optimization techniques and methods of hyperparameter tuning will also allow for the identification of the best combinations to come up with the best model. Various techniques are to be employed, such as Bayesian optimization, grid search, and random search.

This shall bring forth a marked improvement in the performance of the models. The right combination of parameters can bring a boost in the predictive capabilities of the model.

Methods for interpreting and explaining model predictions also deserve further research. The ability to interpret and explain the basis for the predictions of binding affinity shall help to highlight the possible ways in which the models may be improved. This way, transparency is built in predictive models, making them more useful tools for researchers and practitioners in the area of computational drug design.

References

[1] H. W.-B. in Bioinformatics and undefined 2024, "Prediction of protein–ligand binding affinity via deep learning models," *academic.oup.com*, vol. 2024, no. 2, p. 81, doi: 10.1093/bib/bbae081.

[2] C. Shen, J. Ding, Z. Wang, D. Cao, X. Ding, and T. Hou, "From machine learning to deep learning: Advances in scoring functions for protein–ligand docking," *Wiley Interdiscip Rev Comput Mol Sci*, vol. 10, no. 1, p. e1429, Jan. 2020, doi: 10.1002/WCMS.1429.

[3] J. Liu and R. Wang, "Classification of current scoring functions," *J Chem Inf Model*, vol. 55, no. 3, pp. 475–482, Mar. 2015, doi: 10.1021/CI500731A.

[4] S. Li *et al.*, "Structure-aware Interactive Graph Neural Networks for the Prediction of Protein-Ligand Binding Affinity," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 975–985, Aug. 2021, doi: 10.1145/3447548.3467311.

[5] F. Frasca, E. Rossi, D. Eynard, B. Chamberlain, M. Bronstein, and F. Monti, "SIGN: Scalable Inception Graph Neural Networks," Apr. 2020, Accessed: May 17, 2024. [Online]. Available: http://arxiv.org/abs/2004.11198

[6] J. Jiménez, M. Škalič, G. Martínez-Rosell, and G. De Fabritiis, "KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks," *J Chem Inf Model*, vol. 58, no. 2, pp. 287–296, Feb. 2018, doi: 10.1021/ACS.JCIM.7B00650.

[7] Y. Lu, J. Liu, T. Jiang, Z. Cui, H. W.-C. Bioinformatics, and undefined 2023, "Drug-target Binding Affinity Prediction Based on Three-branched Multiscale Convolutional Neural Networks," *ingentaconnect.com*, Accessed: May 17, 2024. [Online]. Available: https://www.ingentaconnect.com/content/ben/cbio/2023/00000018/00000010/art00008

[8] J. Ding *et al.*, "Vina-GPU 2.0: Further Accelerating AutoDock Vina and Its Derivatives with Graphics Processing Units," *J Chem Inf Model*, vol. 63, no. 7, pp. 1982–1998, Apr. 2023, doi:

10.1021/ACS.JCIM.2C01504/SUPPL_FILE/CI2C01504_SI_001.PDF.

[9] "Welcome to PDBbind-CN database." Accessed: May 18, 2024. [Online]. Available: http://pdbbind.org.cn/index.php

[10] G. Li, Y. Yuan, R. Z.-C. B. and Chemistry, and undefined 2023, "Ensemble of local and global information for Protein–Ligand Binding Affinity Prediction," *Elsevier*, Accessed: May 17, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1476927123001639

[11] J. C. Gómez-Tamayo, L. Cao, M. Ahmad, and G. Tresadern, "Binding Affinity Prediction with 3D Machine Learning: Training Data and Challenging External Testing," Feb. 2024, doi: 10.26434/CHEMRXIV-2024-S5SBS.

[12] H. Wang, "Prediction of protein–ligand binding affinity via deep learning models," *Brief Bioinform*, vol. 25, no. 2, p. 81, Jan. 2024, doi: 10.1093/BIB/BBAE081.

[13] J. Liu and R. Wang, "Classification of current scoring functions," *J Chem Inf Model*, vol. 55, no. 3, pp. 475–482, Mar. 2015, doi: 10.1021/CI500731A/ASSET/IMAGES/MEDIUM/CI-2014-00731A_0003.GIF.

[14] R. Gorantla, A. Beta Kubincová, A. Y. Weiße, and A. S. J. S. Mey, "From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction," *ACS Publications*, vol. 64, no. 7, pp. 2496–2507, Apr. 2023, doi: 10.1021/acs.jcim.3c01208.

[15] J. Rahman, M. A. H. Newton, M. E. Ali, and A. Sattar, "Distance plus attention for binding affinity prediction," *J Cheminform*, vol. 16, no. 1, p. 52, May 2024, doi: 10.1186/S13321-024-00844-X.

[16] S. Dehghani-Ghahnaviyeh, C. Soylu, P. Furet, and C. Velez-Vega, "Dissecting the Interaction Fingerprints and Binding Affinity of BYL719 Analogs Targeting PI3Kα," *ACS Publications*, vol. 128, no. 8, pp. 1819–1829, Feb. 2024, doi: 10.1021/acs.jpcb.3c06766.

[17] B. Roux, C. C.-J. of C. I. and Modeling, and undefined 2024, "Classifying Protein-Protein Binding Affinity with Free-Energy Calculations and Machine Learning

Approaches.," *europepmc.org*, Accessed: May 18, 2024. [Online]. Available: https://europepmc.org/article/med/38272021

[18] V. Kairys, … L. B.-E. O. on, and undefined 2024, "Recent advances in computational and experimental protein-ligand affinity determination techniques," *Taylor & Francis*, Accessed: May 18, 2024. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/17460441.2024.2349169

[19] Z. Meng and K. Xia, "Persistent spectral–based machine learning (PerSpect ML) for protein-ligand binding affinity prediction," *Sci Adv*, vol. 7, no. 19, May 2021, doi: 10.1126/SCIADV.ABC5329.

[20] M. M. Rana and D. D. Nguyen, "Geometric graph learning with extended atom-types features for protein-ligand binding affinity prediction," *Comput Biol Med*, vol. 164, Sep. 2023, doi: 10.1016/j.compbiomed.2023.107250.

[21] B. Zdr azil *et al.*, "The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods," *academic.oup.com*, vol. 52, pp. 1180–1192, 2024, doi: 10.1093/nar/gkad1004.

[22] M. Su *et al.*, "Comparative Assessment of Scoring Functions: The CASF-2016 Update," *J Chem Inf Model*, vol. 59, no. 2, pp. 895–913, Feb. 2019, doi: 10.1021/ACS.JCIM.8B00545.

[23] K. A. Carpenter and R. B. Altman, "Databases of ligand-binding pockets and protein-ligand interactions," *Comput Struct Biotechnol J*, vol. 23, pp. 1320–1338, 2024, doi: 10.1016/j.csbj.2024.03.015.

[24] B. Ji, "Applications of ML/DL overcoming current challenges in structure-based drug design," 2024, Accessed: May 18, 2024. [Online]. Available: http://d-scholarship.pitt.edu/45782/

[25] "7. Representation of (a) 3D matrix of voxels, and (b) voxelization of... | Download Scientific Diagram." Accessed: May 18, 2024. [Online]. Available: https://www.researchgate.net/figure/Representation-of-a-3D-matrix-of-voxels-and-b-voxelization-of-point-cloud-data_fig9_344488704

[26] X. Liu, H. Feng, J. Wu, and K. Xia, “Dowker complex based machine learning (DCML) models for protein-ligand binding affinity prediction,” *PLoS Comput Biol*, vol. 18, no. 4, Apr. 2022, doi: 10.1371/JOURNAL.PCBI.1009943.

[27] “Protein-ligand complex from molecular docking. | Download Scientific Diagram.” Accessed: May 18, 2024. [Online]. Available: https://www.researchgate.net/figure/Protein-ligand-complex-from-molecular-docking_fig4_319054646

[28] M. Fey and J. E. Lenssen, “Fast Graph Representation Learning with PyTorch Geometric,” Mar. 2019, Accessed: May 17, 2024. [Online]. Available: http://arxiv.org/abs/1903.02428

[29] ShaoYingxia *et al.*, “Distributed Graph Neural Network Training: A Survey,” *ACM Comput Surv*, vol. 56, no. 8, pp. 1–39, Apr. 2024, doi: 10.1145/3648358.

[30] Y. Cen *et al.*, “CogDL: A Comprehensive Library for Graph Deep Learning,” *ACM Web Conference 2023 - Proceedings of the World Wide Web Conference, WWW 2023*, pp. 747–758, Apr. 2023, doi: 10.1145/3543507.3583472.

[31] “1 3D Target-ligand interaction map. | Download Scientific Diagram.” Accessed: May 18, 2024. [Online]. Available: https://www.researchgate.net/figure/3D-Target-ligand-interaction-map_fig1_364761058

[32] D. Jones *et al.*, “Improved Protein-Ligand Binding Affinity Prediction with Structure-Based Deep Fusion Inference,” *J Chem Inf Model*, vol. 61, no. 4, pp. 1583–1592, Apr. 2021, doi: 10.1021/ACS.JCIM.0C01306/SUPPL_FILE/CI0C01306_SI_001.PDF.

[33] M. Jiang *et al.*, “Drug–target affinity prediction using graph neural network and contact maps,” *pubs.rsc.org*, 2020, doi: 10.1039/d0ra02297g.

[34] Z. Yang, W. Zhong, L. Zhao ab, and C. Yu-Chian Chen, “MGraphDTA: deep multiscale graph neural network for explainable drug-target binding affinity prediction †,” 2022, doi: 10.1039/d1sc05180f.

[35] J. Zhou *et al.*, “Distance-aware molecule graph attention network for drug-target binding affinity prediction,” *arxiv.org*, 2022, doi: 10.1039/d1sc05180f.

[36] V. Kairys *et al.*, “Recent advances in computational and experimental protein-ligand affinity determination techniques,” *Expert Opin Drug Discov*, 2024, doi: 10.1080/17460441.2024.2349169.

[37] D. J. Hsu *et al.*, “TwoFold: Highly accurate structure and affinity prediction for protein-ligand complexes from sequences,” *International Journal of High Performance Computing Applications*, vol. 37, no. 6, pp. 666–682, Nov. 2023, doi: 10.1177/10943420231201151.

[38] X. Zhang, C. Shen, H. Zhang, Y. Kang, C. Y. Hsieh, and T. Hou, “Advancing Ligand Docking through Deep Learning: Challenges and Prospects in Virtual Screening,” *Acc Chem Res*, 2024, doi: 10.1021/ACS.ACCOUNTS.4C00093.