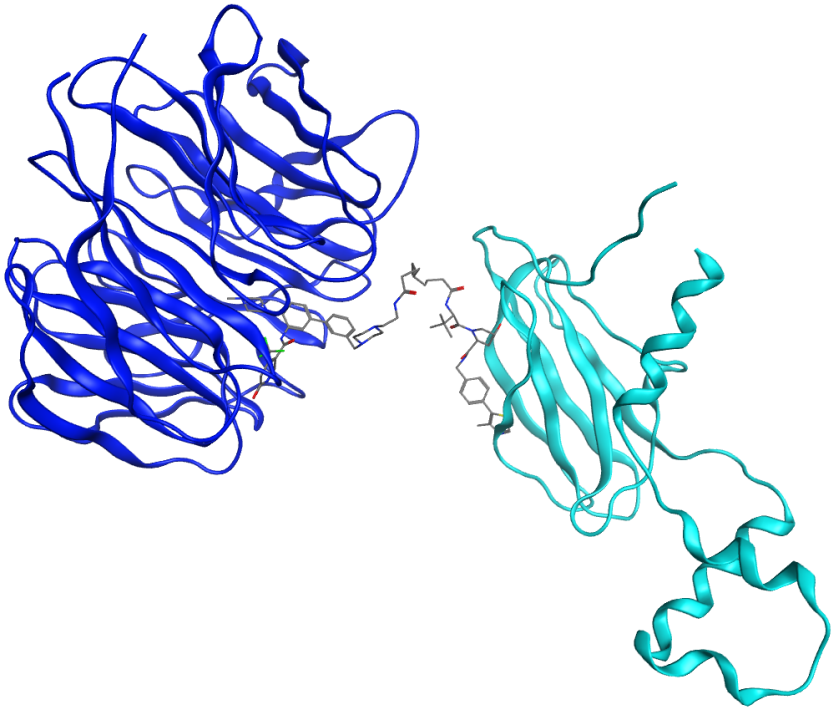
CHALMERS
UNIVERSITY OF TECHNOLOGY



Machine Learning for Structural Predictions of PROTACs

Predicting protein and molecular structures with AlphaFold and graph neural networks

Master's thesis in Biotechnology

ANDERS KÄLLBERG

DEPARTMENT OF LIFE SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

Machine Learning for Structural Predictions of PROTACs

Predicting protein and molecular structures with AlphaFold and graph neural networks

Anders Källberg

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Life Sciences
Division of Data Science and AI
AI Laboratory for Biomolecular Engineering
Chalmers University of Technology
Gothenburg, Sweden 2024

Machine Learning for Structural Predictions of PROTACs
Predicting protein and molecular structures with AlphaFold and graph neural networks

Anders Källberg

Supervisor: Rocío Mercado, CSE Department at Chalmers
Supervisor: Eva Nittinger, AstraZeneca
Supervisor: Christian Tyrchan, AstraZeneca
Examiner: Pernilla Wittung Stafshede, LIFE Department at Chalmers

Master's Thesis 2024
Department of Life Sciences
Division of Data Science and AI
AI Laboratory for Biomolecular Engineering
Chalmers University of Technology
SE-412 96 Göteborg
Telephone: +46 31 772 1000

Cover: Crystallized PROTAC ternary structure (PDB: 7JTO) visualized in Molecular Operating Environment.

Typeset in LaTeX
Gothenburg, Sweden 2024

Machine Learning for Structural Predictions of PROTACs
Predicting protein and molecular structures with AlphaFold and graph neural networks

Anders Källberg

Department of Life Sciences
Chalmers University of Technology

Abstract

PROteolysis TArgeting Chimeras (PROTACs) are molecules that induce the degradation of targeted proteins by hijacking the ubiquitin–proteasome system in the cell. A PROTAC binds simultaneously to an E3 ligase and a protein of interest (POI), forming a ternary complex. The ubiquitin–proteasome system tags the POI with ubiquitin, marking it for degradation by the proteasome. The formation of a good ternary complex is essential for the ubiquitination and subsequent degradation of the POI.

Being able to accurately model ternary complexes thus provides critical advantages in the development of PROTACs; however, data on PROTACs and their crystallized ternary complexes are limited. Accurate predictions of these structures are desirable, but current computational methods struggle to simulate the interactions between the PROTAC and both proteins simultaneously.

AlphaFold, a machine learning tool, has been shown to accurately predict protein complexes. Yet, research on applying AlphaFold to predict ternary complexes is scarce. In the first part of this thesis, the ternary complex was modeled using AlphaFold by utilizing the sequences of both natural and artificially linked POIs and E3 ligase. Nevertheless, it was determined that AlphaFold was unable to accurately predict these complexes, reasonably because it was not able to take the PROTAC into account in the predictions.

The second part of this thesis focused on generating data on PROTAC substructures, essential for the development of these molecules. Despite the availability of such data, obtaining high-quality data on substructures of specific PROTACs can be challenging and time-consuming. To address this, the PROTAC Splitter, a novel machine learning tool based on graph neural networks, was developed to predict these substructures. The PROTAC Splitter predicts 99.7% of PROTACs, with known substructures, to a maximal error of 6 atoms wrong between the boundaries of the ligands and linker. It generalizes to PROTACs with three unknown substructures, where 23.1% of these predictions satisfy the same criteria. The code for the PROTAC splitter is available at https://github.com/AndersKallberg/PROTAC_splitter. Although accurate predictions of ternary complexes remain challenging, the PROTAC Splitter makes the substructures easily accessible to anyone in this field of research.

In summary, the work presented in this thesis answers scientific questions in two complementary areas of PROTAC development: (1) ternary (protein) structure prediction, and (2) PROTAC component prediction. This information is limited and valuable, and accurate predictions of these could accelerate the discovery of effective PROTACs and help in the fight against disease.

Keywords: PROTAC, Ternary Structure, Substructures, AlphaFold, Protein Structure Prediction, Graph Neural Networks, Node Prediction, Link Prediction, Machine Learning, AI.

Acknowledgements

I am very thankful for being given this thesis. Its been incredibly enriching to work at AstraZeneca and in a research group, as it gave me valuable insights into how both the pharma industry and academia functions. I am thankful for being given the trust to complete this thesis, despite that I had no practical experience of machine learning at the start. The thesis has been challenging, but also very rewarding and developmental, and it opened my eyes to what I want to work with.

- Thank you -

Eva Nittinger for introducing me to the team, showing me around the site, making me feel welcomed. You are always cheerful and its been great having you as my supervisor. You helped me whenever I asked about the planning of experiments, analysis of results, AZ administration, and teaching me about the pharma industry and its methods. I'm really grateful for all this.

Rocío Mercado, your constant feedback and expertise in machine learning have been very useful, and I am thankful for all that you have taught me. It really shows that you really care about your team and want us to succeed. You are a great supervisor, and you have gone beyond my expectations, partly as you have been alike a mentor as well. Thank you Rocío.

Christian Tyrchan for your guidance, feedback and ideas in this project.

Stefano Ribes for your help in explaining your work and code, for bouncing ideas and giving feedback, helping me with git, and teaching me about machine learning and GPUs. Its been fun working with you.

Yossra Gharbi for our discussions on PROTACs, they helped me understand a little more of all their complexities. I wish you the best of luck with your PhD!

Leonardo De Maria for providing and explaining the code for analyzing protein-protein interfaces and the dataset of antibodies and antigens. You are a cool character who often brings me a smile and keeps me alert.

Gustav Olanders for introducing me to bash and helping me get started in using AlphaFold right away. Your quick jokes always made me laugh or crack a smile.

Hongtao Zhao and Janosch Menke for your feedback on our regular Monday meetings. I'll always remember those meetings.

Another thank you to everyone in my teams at Chalmers and AstraZeneca for being welcoming, kind and helpful. I am grateful for having worked with you and all pleasant lunches together.

Anders Källberg, Gothenburg, June 2024.

Contents

1

Introduction

PROteolysis TArgeting Chimeras (PROTACs) are molecules that induce the degradation of targeted proteins and show potential as a treatment for a range of diseases including neurodegenerative diseases, inflammatory diseases, cancers, and viral infections [1, 2]. Designing effective PROTACs is challenging, and current methods are mostly empirical [3]. Although computational methods for optimizing PROTAC degradation capacity hold promise, they face multiple limitations due to the complex mechanism involving many interactions and their initial design for small molecules rather than PROTACs [4].

A PROTAC consists of three substructures with distinct roles in its mechanism of action, making them crucial for designing and optimizing PROTACs. Yet, no publicly available tool currently predicts the substructures of a PROTAC, complicating their effective utilization in various projects.

Research using machine learning to predict PROTAC degradation capacity is scarce. Only one study has utilized the substructures and structural information of PROTAC-binding pockets to predict its degradation capacity [5]. Despite that the full structure of the PROTAC-protein complex is critical for its effectiveness [3, 6], no literature was found that predicted the degradation capacity using this information.

Predicting the degradation capacity of PROTACs is challenging, partly due to the limited amount and diversity of data. This thesis explores novel machine learning methods to expand the publicly available data, to allow for better predictions of degradation capacity. Specifically, it focuses on predicting PROTAC-protein complexes with AlphaFold and developing a novel machine learning tool for predicting PROTAC substructures. Accurate predictions of protein complexes and substructures could significantly aid in the design and development of new PROTACs, potentially reducing costs and shortening the time to discovery in early drug development, and thereby saving lives.

2

Background

2.1 The Ubiquitin–Proteasome System

PROTACs facilitate protein degradation by hijacking the cell's ubiquitin–proteasome system (UPS) [2]. A PROTAC targets a protein for degradation by directing the UPS to degrade it. The UPS naturally degrades proteins by tagging them with ubiquitin, a small protein of 76 amino acids [7]. This tagging directs the marked protein to the proteasome for degradation.

The process involves three critical types of proteins: E3 ubiquitin ligase (E3 ligase), E2 ubiquitin-conjugation enzyme (E2 enzyme), and E1 ubiquitin-activating enzyme (E1 enzyme) [7]. The E1 enzyme first binds to ubiquitin and facilitates the activation of ubiquitin by ATP. After activation, the E1 enzyme transfers the ubiquitin to the E2 enzyme and it forms a thioester bond at a cysteine residue. The E2 enzyme then interacts with the E3 ligase and the substrate protein to transfer ubiquitin to a lysine residue on the substrate protein, forming an iso-peptide bond.

The mechanism by which ubiquitin is transferred to the substrate varies by the class of E3 ligase. It can occur directly from the E2 enzyme to the substrate or indirectly through the E3 ligase [2]. A protein can be tagged with one or several ubiquitins; additional ubiquitins can attach either to other lysine residues on the substrate or to previously attached ubiquitins [8]. Once tagged, the protein is degraded by the proteasome.

2.2 PROTAC Structure and Mechanism of Action

PROTACs are engineered to bind simultaneously to a protein of interest (POI) implicated in a specific disease pathway and an E3 ligase [2]. By binding to both the POI and E3 ligase, PROTACs facilitate the formation of a ternary complex, bringing these proteins into proximity. The ternary complex interacts with the UPS similarly to how a natural substrate protein would, leading to the ubiquitination and subsequent degradation of the POI. This process is illustrated in Figure 2.1.

PROTACs reversibly bind to the POI, enabling them to be reused in subsequent cycles to recruit additional POI and E3 ligase pairs. This allows PROTACs to act as catalysts, facilitating degradation of proteins that normally do not occur in the cell and can be reused after each reaction [3].

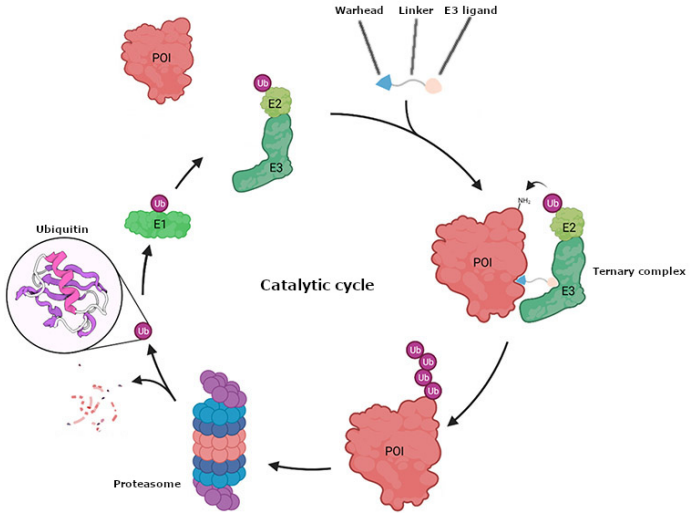


Figure 2.1: An overview over the mechanism of a PROTAC and the ubiquitin-proteasome system. An image was retrieved from [2] and adapted to this work.

A PROTAC consists of two ligands joined together by a molecular linker [1]. One ligand binds to the POI targeted for degradation, and the other to an E3 ligase. These ligands are known as the *warhead* and the *E3 ligand*, respectively. The linker not only facilitates the formation of the ternary complex but also significantly influences the degradation capacity, selectivity, and binding strength between the POI and E3 ligase [9, 2]. Additionally, it affects the PROTAC's solubility and biological properties, such as membrane permeability, metabolic stability, and biodistribution. The design of linkers varies, with no universally applicable method; the optimal length, composition, and rigidity depend on the specific E3 ligand and warhead. Figure 2.2 presents an example of a PROTAC, highlighting its substructures and artificial attachment points, which illustrate how the components are connected within the PROTAC.

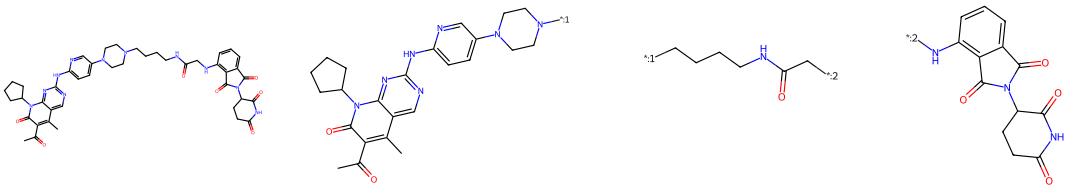


Figure 2.2: Example of a PROTAC (left) and its warhead (center left), linker (center right) and E3 ligand (right).

Generally, a the complex a PROTAC induces includes a POI, PROTAC, E3 ligase, but also adaptor proteins, scaffolding proteins, and an E2 enzyme [7]. Adaptor proteins bind the E3 ligase, connecting it to the scaffolding proteins, which in turn bind the E2 enzyme. Figure 2.3 shows an example of a PROTAC complex.

An "E3 ligase" is not an specific protein, but rather a family of >600 proteins in the human proteome [11]. Combined with the fact that different E3 ligases can recruit different adaptor and scaffolding proteins it further increases the complexity of this field of study. Another layer

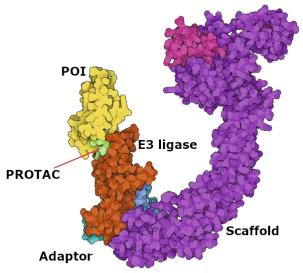


Figure 2.3: Example of a PROTAC complex. The E2 Enzyme and Ubiquitin are not displayed. An image was retrieved from [10] and adapted to this work.

of complexity is that ubiquitination of one POI and a specific E3 ligase have been shown to increase stability of the POI [12].

Despite the complexity of the field, *Crowe 2024 et. al.* have proposed a mechanistic theory explaining the likelihood of ubiquitination of a POI. According to this theory, lysines on the POI that are closer and properly oriented towards the E2-ubiquitin thioester bond are more likely to be ubiquitinated. This ubiquitination depends on the proximity and orientation of the POI, particularly whether proximate lysines are positioned towards or away from the E2-ubiquitin bond. These factors are critical in determining the selectivity of PROTAC-mediated ubiquitination. Additionally, the flexibility of the complex enables it to ubiquitinate multiple lysines on the POI and to target already ubiquitinated lysines for further tagging. Thus, it is not merely the formation of a ternary complex that dictates the degradation capability of a PROTAC, but also its structural arrangement that ensures lysines are accessible to the E2-ubiquitin thioester bond.

2.3 PROTAC Degradation Metrics

The efficacy of a PROTAC is quantified by two key metrics: the maximal degradation percentage of the POI, D_{Max}, and the concentration of PROTAC required to degrade 50% of the maximum amount of the POI, designated DC_{50} [2]. These metrics are depicted in Figure 2.4. Also illustrated in this Figure is the Hook effect, which represents a decrease in degradation efficiency at higher PROTAC concentrations. This effect arises when the concentrations of PROTAC are relatively high compared to those of the POI and E3 ligase, increasing the likelihood that the POI and E3 ligase bind to separate PROTAC molecules, thus inhibiting the formation of effective ternary complexes. Figure 2.4 serves an educational purpose, demonstrating the typical characteristics of a degradation profile, but it does not perfectly capture all nuances of degradation profiles.

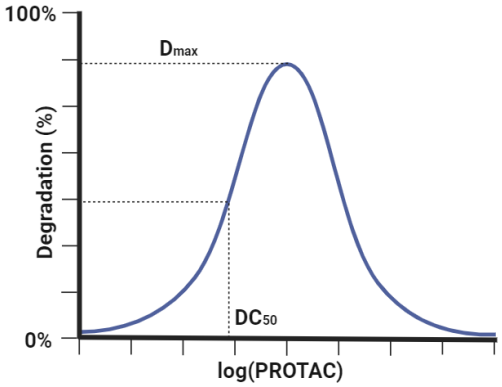


Figure 2.4: A degradation profile for a PROTAC. The figure was created using Biorender.

2.4 PROTACs from a Pharmaceutical Perspective

Historically, the pharmaceutical industry has relied on small molecules to target proteins for disease treatment. Yet, many proteins lack targetable or functional binding pockets, categorizing them as "undruggable" [13]. PROTACs have demonstrated the capability to degrade several of these undruggable proteins [14, 15], promising to broaden the spectrum of targetable proteins. Unlike small molecules, PROTACs do not require binding to functional pockets, due to their distinct mechanism of action [9].

Compared to small molecule inhibitors, which modulate protein function by binding, PROTACs eliminate the protein entirely. This often allows for lower dosages because PROTACs utilize a catalytic mechanism rather than an occupancy-driven one, which small molecule inhibitors does. For instance, an exceptionally effective PROTAC have achieved a D_{Max} of 99% and a DC_{50} of 0.01 nM for the androgen receptor [16], which highlights their potential for reduced side effects and toxicity due to lower required concentrations [17]. However, designing PROTACs with optimal properties remains challenging due to their complex structure [18] and the fact that they do not conform to "Lipinski's Rule of 5" for designing drugs [19]. Additionally, PROTACs generally exhibit lower oral bioavailability, solubility, and permeability compared to small molecules. The prediction of these properties in PROTACs is particularly difficult due to their complexity and the limited data available [4].

Small molecules can be repurposed as warheads for PROTACs, enhancing selectivity beyond that of the small molecule alone. This increased selectivity stems from the specific interactions within the ternary complex formed between the POI, PROTAC, and E3 ligase [20], as supported by the proposed theory by *Crowe 2024 et. al.*.

As of 2022, 23 PROTACs have entered clinical trials, 20 of which target various cancers [21]. PROTACs have also been developed against other age related diseases, such as Alzheimer's disease, where they target the Tau protein [22, 23] (not in clinical trials).

PROTACs belong to class of molecules named targeted protein degraders, which also includes 'molecular glues' [10]. Although generally smaller, molecular glues operate similarly by forming a ternary complex with a POI and E3 ligase to ubiquitinate and degrade the POI. Figure 2.5 provides an example of such a complex, although the E2 enzyme and ubiquitin are not displayed
.

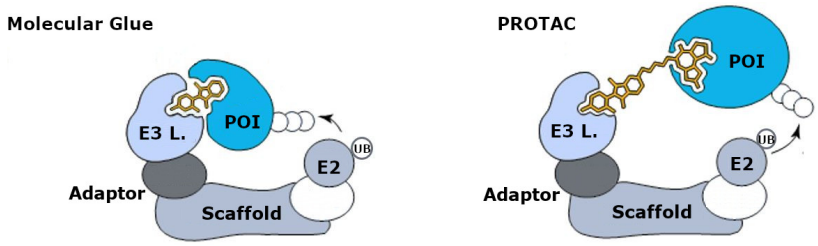


Figure 2.5: Comparison of complex structures mediated by a molecular glue and PROTAC. An image was retrieved from [24] and adapted to this work.

2.5 Available Data

The two primary public databases for PROTACs, PROTAC DB [25] and PROTAC Pedia [26], catalog information on PROTAC structure and substructures (SMILES), UniProt ID of the POI and E3 ligase, PDB ID, DMax, DC50, and physicochemical properties. Together, these databases cover approximately 5000 PROTACs. However, specific data types, such as PDB IDs for ternary complex crystal structures, are limited; PROTAC DB contains only 18 entries with such PDB IDs.

2.6 Protein Crystal Structures

X-ray crystallography is a pivotal method in structural biology, offering a detailed three-dimensional map of atomic positions within proteins [27]. This map is crucial for understanding the mechanistic aspects of protein functions and interactions. However, crystallizing proteins poses significant challenges. The process involves purifying the protein, experimenting with various crystallization conditions to promote crystal growth, and ultimately growing crystals that are suitable for high-resolution diffraction. Each step can be time-consuming and labor-intensive, often requiring a trial-and-error approach to identify optimal conditions.

As of 2024, there exists 190 000 structures in the Protein Data Bank (PDB) [28]. This has allowed the development protein structure prediction tools, such as AlphaFold [29], which are based on machine learning (see Section 2.8 for details).

2.7 Applying Conventional Computational Tools for PROTACs

Once the protein structure is obtained, various computational tools can be employed, such as Molecular dynamics (MD) and Docking simulations. MD simulations is a type of method for simulating the motion of atoms and molecules by calculating force fields and applying Newtons laws of motion to calculate the evolution of the system [30]. In the context of PROTACs, MD is useful for identifying binding sites, predicting binding affinities, [4], generating a conformal ensemble of a protein to utilize in docking [31]. MD have been used to analyse the protein-protein interfaces of PROTAC ternary complexes, as well as to evaluate PROTAC binding affinity and cooperativity [32], and scoring plausible ternary complexes

[33]. MD have also been successfully utilized to predict PROTAC dissociation constants and the a parameter which characterizes the hook-effect [34]. However, MD is challenging because there are many interacting components that need to be simulated with short time steps (1∼2 fs per step), which requires large computational resources for longer simulations. Also, if a binding event is rare, as the binding strength is weak, it may require a long time.

Docking simulates molecular interactions between molecules and proteins [35], where these may be rigid or flexible bodies, in contrast to MD which simulates movements down to the level of atoms. This makes docking faster than MD. Docking can be used to predict binding modes and binding affinities of molecules to proteins, which can be used to screen for molecules that bind to a target protein.

Custom made tools and protocols have been developed to predict experimental PROTAC ternary structures, but the success of these have so far been limited [34]. The difficulty of predicting the ternary structure can partly be attributed to that the task combines the challenge of protein-protein docking with protein-ligand docking simultaneously, and current docking tools do not sample poses of small molecules and proteins simultaneously [4].

Quantitative Structure-Activity Relationship (QSAR) models, which are widely used in drug discovery, utilize molecular descriptors to predict properties of molecules using a variety of mapping functions [36]. These functions can predict either numerical values or categories. The range of mapping functions includes multiple linear regression, nonlinear regression, partial least squares, and linear discriminant analysis, as well as more complex models like Bayesian classifiers, decision trees, k-nearest neighbors, support vector machines, and neural networks. Generally, it is difficult to build effective QSAR models with limited data, as is the case for PROTACs.

2.8 AlphaFold

AlphaFold2 is a deep learning tool which represents a significant advancement in the field of computational biology by addressing the protein folding problem [29]. This problem, which has challenged researchers for over half a century, involves predicting a protein's three-dimensional structure from its amino acid sequence.

AlphaFold2 predicts protein structures by utilizing publicly available protein crystal structures from the Protein Data Bank (PDB). It uses the amino acid sequence of a protein to search databases, creating multiple sequence alignments (MSA) and identifying similar sequences used as structural templates. These are encoded and processed in a unique neural network architecture, where the MSA representation and structural representation iteratively update each other. A structure model then uses these representations to formulate a 3D structure adhering to physiochemical constraints such as bond angles and distances. To refine this predicted structure further, the predicted structure undergoes three additional iterations through AlphaFold2, followed by energy minimization to correct any stereochemical inaccuracies.

AlphaFold2 have been shown to be able to predict multimer interactions by joining proteins with a flexible linker, despite it being trained on single chain proteins [37]. However, it remained a challenge in many cases. To this end, AlphaFold-Multimer was developed and it is a model that is specifically trained on multimeric protein complexes and it show an improved accuracy at predicting multimers than AlphaFold2.

Originally publicized in July 2021, AlphaFold2 enhanced the foundational model by improving accuracy and expanding functionality. Building on this, AlphaFold3 was released in May 2024, showcasing superior accuracy in predicting protein-protein interactions, and is now also capable of predicting the structure of proteins interacting with ions, small molecules and nucleic acids, as well as to predict protein structures with modified residues [38]. AlphaFold3 predicts protein-ligand, protein-nucleic acid, and antibody-antigen interactions more accurately than state of the art docking tools, nucleic-acid-specific predictors and AlphaFold-Multimer respectively.

2.9 Molecular Representations

2.9.1 SMILES and Molecular Graphs

The Simplified Molecular Input Line Entry System (SMILES) [39] is a method for representing molecular structures and is extensively utilized in chemoinformatics. SMILES notations are human-readable; for instance, the SMILES representation for ethanol is 'CCO'. In this format, hydrogens and bond orders are implicitly understood. Despite being a linear text sequence, SMILES can encode more complex 2D and 3D features, such as rings and chiral centers respectively. Furthermore, multiple SMILES notations can describe the same molecule (e.g., 'OCC' and 'C(O)C' for ethanol), it is possible to standardize these variations to a so called canonical SMILES (in this case 'CCO'), which is crucial for data processing. Not all SMILES strings are valid; for instance, 'C1CO' suggests a ring structure beginning at the first carbon without a designated endpoint, representing an example of invalid grammar. Meanwhile, 'CC#O' implies a triple bond to an oxygen atom, which, although grammatically correct, is chemically invalid. These examples underscore the importance of using SMILES notation that accurately corresponds to chemically valid molecules.

Molecular graphs use graph theory to represent molecules, where atoms are represented as nodes and bonds as edges [40]. This forms a two-dimensional representation of nodes and edges with chemical and structural information that can be embedded one-to-one from atoms and bonds.

By drawing further upon graph theory, graph centralities can be calculated and embedded into the nodes [41]. A graph centrality quantifies the importance of nodes within the network, using a given formula. Essentially it encodes information which is purely related to the structure of the graph. Common centrality measures include *Betweenness*, *Closeness*, and *Eigenvector centrality*.

Betweenness centrality measures the extent to which a node appears on the shortest paths between other nodes. This is useful for identifying nodes that act as bridges between different clusters within the graph.

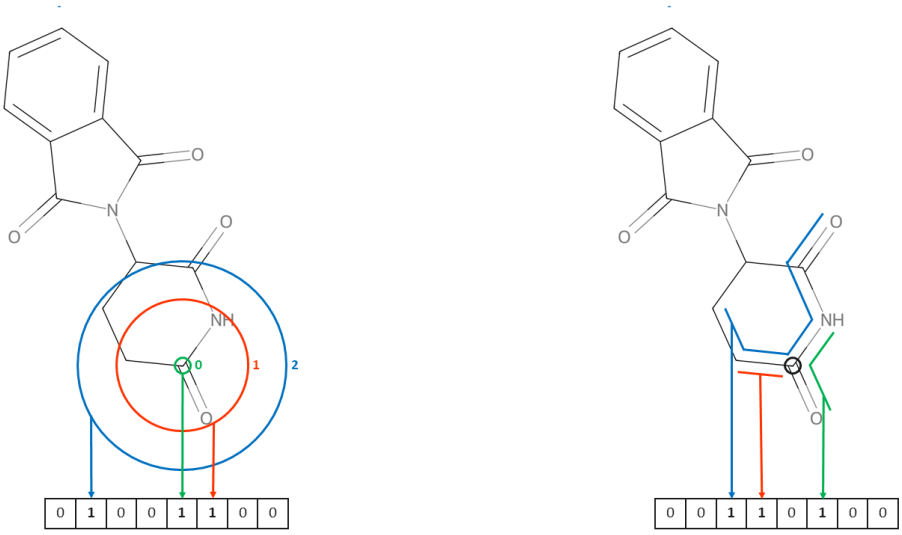
Closeness centrality reflects how close a node is to all other nodes in the graph, calculated as the inverse of the sum of the shortest path distances from the node to all others. This measure highlights nodes that can quickly connect to all other nodes.

Eigenvector centrality evaluates how influential a node is, considering both the number and the significance of its connections. A node linked to other highly connected nodes will have a high eigenvector centrality, indicating its influence within the network.

2.9.2 Fingerprints

Molecular fingerprints are vectors that capture structural and molecular properties of a molecule [42]. These vectors can be binary, indicating the presence or absence of various substructures, or count-based, recording the frequency of each substructure. Although most fingerprints only capture 2D information from the molecular graph, it is also possible to include information from a molecule's 3D structure.

In this work, the count-based Extended-Connectivity Fingerprint (ECFP) and Topological fingerprint (RDKitFP) in RDKit were utilized [43]. The ECFP, commonly referred to as ECFP4 when using a standard radius of 2, is the most commonly used fingerprint and it captures structures within a circular neighborhood around each atom up to a diameter of four atoms [42]. Figure 2.6 presents simplified examples of these fingerprints, highlighting structures associated with specific atoms, although each fingerprint aggregates information from every atom in the molecule.



(a) Morgan fingerprint **(b)** Path fingerprint

Figure 2.6: Examples of molecular fingerprints, inspired by [44]. The illustration focuses on one atom, where as the full fingerprints uses information from all atoms.

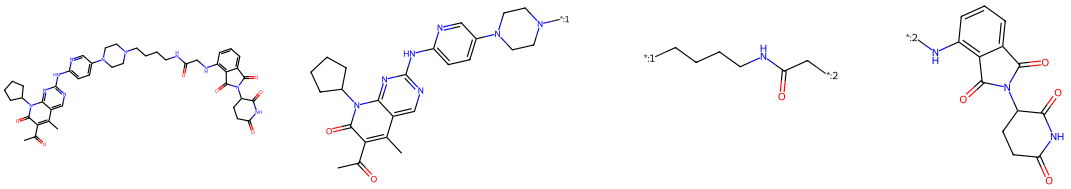
2.9.3 Murcko Scaffolds and Frameworks

A Murcko Framework is a molecular representation consisting of all ring systems and linker atoms that connect them [45]. This representation is useful for grouping similar molecules and emphasizing the core structure of a molecule. Unlike Murcko Frameworks, the Python library RDKit generates Murcko Scaffolds, which includes the framework plus all attached non-rotatable systems [43]. However, RDKit's documentation is unclear, as it defines Murcko Scaffolds to be Murcko Frameworks, yet it is not exactly true, so the exact definition RDKit uses for a Murcko Scaffold is unknown.

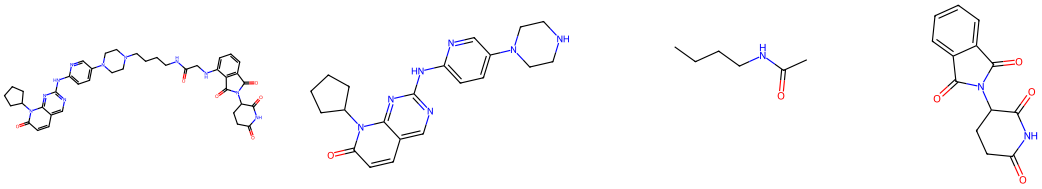
Further, Murcko Frameworks can be abstracted to what is sometimes called a Graph Framework, where atoms and bonds are represented generically [45, 43]. Figure 2.7 shows examples of a Murcko Scaffold and a Graph Framework for a PROTAC and its substructures. Note that substructures in the Figure are marked with artificial attachment points indicated by a star "*" and a number which indicates how the substructures connect to each other. In

this work **:1* will always refer to the warhead-linker boundary and **:2* to the E3 ligase-linker boundary.

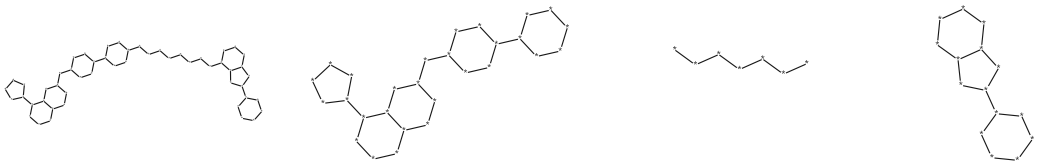
It is important to note that generating a Murcko Scaffold or Graph Framework for a linker typically returns no atoms, as linkers generally lack rings. For this study, the definition for linkers has been adapted: first, a ring is attached at each attachment point; then, the Murcko Scaffold or Graph Framework is derived; finally, these rings are removed.



(a) Example of a PROTAC and its substructures with attachment points (A copy of Figure 2.2).



(b) Murcko scaffolds of the PROTAC and its substructures.



(c) Graph frameworks of the PROTAC and its substructures.

Figure 2.7: Example of a PROTAC and its substructures, represented as molecules with attachment points, as Murcko Scaffolds and Graph Frameworks.

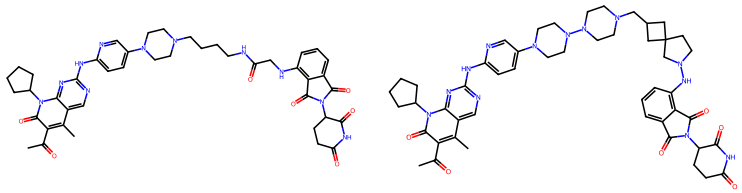
2.10 Molecular Similarity

Molecular similarity is commonly quantified using Tanimoto similarity, a standard method that uses a pair of molecular fingerprints [42]. This similarity metric is calculated as:

$$\frac{|A \cap B|}{|A \cup B|}$$

where A and B represent the sets of structural properties in molecules A and B. For clarity, the Tanimoto similarity is the fraction over the amount of structural properties that the molecules share (intersection) to the amount of structural properties they have combined (union). While other measures exist in literature, this thesis only utilizes the Tanimoto similarity.

As an example, Figure 2.8 compares two PROTACs with identical warheads and E3 ligands but have different linkers. The molecular similarity between the complete PROTAC structures is approximately 0.7, significantly higher than the similarity of just their linkers, which is about 0.03. These similarities were calculated using both Morgan and Path fingerprints.



(a) PROTAC similarity: *0.67, **0.74

(b) Linker similarity: *0.02, **0.04

Figure 2.8: Tanimoto similarity between the PROTACs and their respective linkers, using a *Morgan fingerprint and the **Path fingerprint. The warheads and E3 ligands are identical.

2.11 Machine Learning

Machine learning encompasses a variety of methods aimed at clustering data, reducing dimensions, and visualizing high-dimensional data, as well as techniques for making numerical and categorical predictions [46]. A specific subset of machine learning includes neural networks, such as feedforward neural networks and graph neural networks, which process one-dimensional data and graph-based data, respectively.

Elementary concepts of machine learning will be further detailed in Appendix A. This section will include definitions of commonly used evaluation metrics and essential concepts such as regression, classification, data leakage, data imbalance, learning rate, batch size, epoch, overfitting, regularization, hyperparameter optimization, and cross-validation.

2.11.1 Feed-forward Neural Networks

Feed-forward neural networks are a foundational type of artificial neural network [47]. This architecture channels information straight through from input to output, with data moving in one direction across multiple layers, each of which performs different transformations on its inputs. Feed-forward networks consists of an input, hidden, and output layers, which are trained to minimize the error between a predicted and true value, using a loss function. With the Universal approximation theorem, it is shown that feed-forward networks can approximate a wide range of functions, making them applicable for a broad range of tasks.

2.11.2 Graph Neural Networks

Graph neural networks (GNNs) are a specialized type of neural network designed to directly operate on the graph structure, making them well-suited for processing data represented in graphs [48]. These networks efficiently capture the dependency relationships within the data by leveraging the structural information of the graph. This capability allows them to learn from not just the features of individual nodes and edges, but also from the connections between nodes.

GNNs process graph-structured data by aggregating information from neighboring nodes and edges through a technique known as message passing [48]. In each layer of a GNN, nodes update their states by processing and combining features from their own attributes and those of adjacent nodes. These updates are facilitated by an update function, which merges these features to create a new state for each node. There are various types of layers, each with different update functions, but all possess a set of trainable parameters that enable learning from the data. The output of a GNN varies according to the specific task: it can produce node-level predictions (e.g., classifying types of users in a social network), edge-level predictions (e.g., predicting interactions between proteins), or graph-level outputs (e.g., classifying entire molecular graphs as toxic or non-toxic).

2.12 Related Work

This thesis project builds upon a previous thesis [44], with the aim of creating a deep learning model which predicts if a PROTAC is is an effective degrader or not. The previous project used various deep learning models (Feed-forward, Transformer, Graph Neural Network, and XGBoost) multiple molecular representations (fingerprints, graphs, SMILES) and used various protein descriptors (amino acid sequence, cell type and E3 ligase type) to classify if a degrader is active or not. The best model in this study used XGBoost and the author reasons that one possible reason for this could be that the other deep learning models suffer more from the lack of data. It was highlighted that the 3D information of ternary complexes could enhance the accuracy and explainability of the model, however, this data is even more limited.

In March 2024, a preprint was published examining AlphaFold-Multimer's capability to predict protein-protein interfaces within PROTAC ternary structures [49]. While AlphaFold-Multimer accurately predicted most large and small interfaces in this study, including interfaces mediated by ligands, it struggled with PROTAC-mediated interfaces. The authors suggest this limitation arises because AlphaFold-Multimer was trained on naturally occurring complexes, whereas ternary structure interfaces are artificially induced by PROTACs and do not occur naturally.

In the context of machine learning applications to PROTAC degradation, only one study, DeepPROTAC, has been identified that utilizes 3D structural information of the POI and E3 ligase, likewise its the only found study which utilized substructures for predicting if a PROTAC is active (which they define as $DC_{50} < 100$ nM and $D_{Max} > 80\%$) [5]. This study used graph neural networks to encode the ligands and binding pockets, with the linker represented as SMILES and encoded using a Long Short-Term Memory network.

Research on the use of machine learning to predict DC50 remains scarce. Besides the previous thesis and DeepPROTAC, only two other studies have been identified that use machine learning to score DC50 within a generative model for PROTAC design [50] [51]. To date, no published machine learning model predicts DC_{50} by fully utilizing the ternary structure, despite its crucial role in PROTAC efficacy.

No literature was found on using machine learning to predict PROTAC substructures. However, unpublished work by Stefano Ribes explored using transformers to predict the SMILES of the three substructures, framing it as a text-to-text translation task. However, the model sometimes generated invalid SMILES, incorrectly predicted the number of substructures, and produced substructures with incorrect numbers of atoms compared to the original PROTACs. Only 11 % of the PROTACs in the validation set were successfully split with no mistakes. The limited performance was explained by the limited amount of available data, as transformers tend to require much data to be trained.

3

Predicting Ternary Complex Structure with AlphaFold

AlphaFold2 [29] and AlphaFold-Multimer [37] were used for the purpose of generating the structure of ternary complexes and including 3D structural information in the model. The rationale for seeking 3D structural information is that the formation of the complex is essential for the degradation of the POI. Hence, there should be information about the proteins, PROTAC, and their interactions, which would indicate their ability to form a complex leading to the degradation of the POI. AlphaFold2 and AlphaFold-Multimer were chosen because they has been shown to work well for protein folding and were readily available for use at the start of this master's project.

3.1 Method

3.1.1 Data Preparation

The PDB IDs were obtained from PROTAC-DB [52], and the corresponding protein sequences for the POI and E3 ligase were retrieved from the PDB entries. The referenced UniProt IDs within each PDB-page were also retrived and are presented in Appendix B.3 in Table B.1. Some sequences from the PDB included point mutations or artificial sequences, such as 6xHis tags, which are left overs from the crystallization procedure. To simplify the analysis and reduce the number of simulations, the Longest Common Sequence (LCS) was identified and used for proteins that appeared in multiple crystal structures. An example of an LCS is provided in Appendix B.1.

As there are crystal structures with similar amino acid sequences between the POI and E3 ligase, these were grouped into "complex groups." PDB IDs were assigned a complex group based on its similarity to the amino acid sequences in other PDB IDs. BRD4 which has two distinct PROTAC binding domains that have been crystallized separately with VHL, so despite originating from the same protein, the crystallized sequences of the two binding domains differ, which is why it is listed twice in Table 3.1. A representative crystal structure from each group was selected based on the highest resolution available in the PDB. It was verified that the chosen reference structure did not significantly differ structurally from other structures within the same complex group. The representative crystal structures selected are listed in Table 3.1, along with the remaining PDB IDs in that group. This selection process aimed to minimize the number of comparisons required for each analysis. It's important to note that the resolution pertains to the entire complex, which may include additional proteins beyond the POI and E3 ligase. Additionally, the PDB structures within the same complex group might contain different PROTACs. Further details on these PROTACs are available in Appendix B.6.

Table 3.1: Summary of POI and E3 ligase pairs with the chosen reference crystal structure and its resolution. *6BNB is visually different to other PDB structures within the same complex group.

Complex group	POI	E3	Reference PDB ID	Resolution [Å]	Other PDB IDs
1	BTK	CIAP1	6W7O	2.17	6W8I
2	BRD4 (BD1)	CRBN	6BOY	3.33	6BN7, 6BN8, 6BN9, 6BNB*
3	BCL-xL	VHL	6ZHC	1.92	
4	BRD4 (BD2)	VHL	5T35	2.70	6SIS
5	BRD4 (BD1)	VHL	7KHH	2.28	
6	FAK	VHL	7PI4	2.24	
7	SMARCA2	VHL	6HAY	2.24	6HAX
8	SMARCA4	VHL	6HR2	1.76	
9	WDR5	VHL	7JTO	1.70	7Q2J, 7JTP

A small database of all full-length sequences from UniProt, of all POI and E3 ligases, of the reference crystal structures was constructed. A BLAST was performed for all sequences in this database against all other sequences, to check if there were any very similar proteins.

3.1.2 Using AlphaFold-Multimer

Most ternary structures only contained a POI, E3 ligase and adaptor proteins. This limits which experiments can be properly evaluated to these proteins. However, to reduce the complexity of the task, only the POI and E3 ligase were simulated simultaneously with AlphaFold2, using the LCS from the PDB sequences and UniProt sequences.

AlphaFold-Multimer uses two inputs to predict the structure of proteins; the amino acid sequence and structural templates. It is optional to use custom templates as AlphaFold-Multimer creates templates automatically. Using a custom template was deemed a non-trivial task and beyond the time scope of this project. Therefore, only the amino acid sequence was used as input to AlphaFold-Multimer.

The predicted structures were evaluated by calculating the root mean square distance (RMSD of α-carbons of the amino acids) using the reference crystal structure. This was done for the ternary structure as a whole and separately for the POI and the E3 ligase. As the sequence of the predicted structure could differ from the crystal structure (due to point mutations, protein tags, etc.) the longest common sequence was used for calculating the RMSD.

To evaluate why a predicted structure is disorganized, either due to AlphaFold2 or AlphaFold-Mulimer failing to predict the structure or if that protein is disorganised, the amino acid sequence of disorganised predictions were analyzed with the intrinsically disorganised structure predictor IUPRED3 [53].

3.1.3 Using AlphaFold-Multimer on Chimeric Proteins

It had been shown that AlphaFold2 (not AlphaFold-Multimer) could predict complexes if the proteins were linked end-to-end [37], so this strategy was employed. To do this, each reference complex was visualized in the Molecular Operating Environment (MOE) and a linker of 50 glycines was decided to be long and flexible enough to minimally disturb the folding by AlphaFold2. The PDB-LCS of the POI and E3 ligase were linked with a 50 glycine linker and inputted to AlphaFold2. The resulting protein from linking two proteins will here forth be refereed to a chimeric protein.

Another strategy was to join the POI and E3 ligase with short linkers at their interfaces as in the PDB, as to enforce them to contact at that point. To place a linker at the protein-protein interface, a new N and C terminus needs to be introduced, so the POI and E3 were split (into nPOI & cPOI and nE3 & cE3) and joined together with two linkers as each split created two new ends, resulting in two new chimeras: nPOI-linker-cE3 and nE3-linker-cPOI. The POI and E3 ligase were cut and linked at amino acids in flexible loops between β sheets or α helices. The lengths of these linkers varied between 0 and 4 amino acids, as to enforce close contact. Some pairs of linkers were set to different lengths, as to explore if this could affect the rotation between the proteins. The linkers were made either of glycine, for its flexibility and hypothetical minimal disruption to protein-protein interactions, or of proline, for its inherent rotation. Likewise, proline was tested as to explore potential to influence the rotation between the linked POI and E3 ligase. It was ensured that each selected linker would not majorly disrupt the secondary structures after minimizing the energy of the structure in MOE. With the final sequence of both chimeric proteins, they were inputted into AlphaFold-Multimer.

Complex groups 1, 2, 3, 7, and 9 were selected for experiments involving linking at the interface. This selection encompassed all three E3 ligases present in the crystal structures and all POIs except for FAK as well as BRD4 (BD2) and SMARCA4, which are structurally similar to BRD4 (BD1) and SMARCA2, respectively. Complex group 4 and 8 contained BRD4 (BD2) and SMARCA4 which are similar to BRD4 (BD1) and SMARCA2 respectivly, so they was excluded. Complex group 5 was excluded since it contained BRD4 (BD1), which complex group 2 has. Complex group 5 with FAK was not included, as the number of planned experiments with chosen complex groups was quite large already.

Combining both linking strategies was also tested, by linking a the POI and E3 ligase at the interface using the same methodology as before and linking end-to-end with a 50 glycine linker.

Using MOE, a PROTAC was forcibly into a structure to calculate the strain on the bonds of the PROTAC. A high strain would indicate that it would be very unlikely that the PROTAC would not be in this conformation, and if that conformation corresponds to the binding pose of a PROTAC to a predicted complex, then that complex would very unlikely to have that structure. This was done by using the reference structure as a starting point: The linker was removed and the ligands were fixed to their proteins and then aligned to the predicted POI and E3 ligase individually. The linker was the built up manually and joined to the ligands, followed by an energy minimization of only the linker. Using a built in function in MOE, TorAnalyzer, the torsional strain was calculated for the bonds in MOE.

The RMSD between the reference and the chimeric proteins was calculated by aligning the reference POI with the nPOI and cPOI, and likewise for the E3 ligase.

3.1.4 Analysis of the Protein-Protein Interface

PyMol was used to calculate the solvent accessible surface area (SASA) for each amino acid for the proteins in isolation and as a complex. The SASA of each amino acid for the isolated proteins and complex was summed to get their total surface areas. The interface area is defined as the difference of the SASA of the isolated proteins and the SASA of their complex, which is illustrated in figure 3.1. As this interface includes the area from both the POI and E3 ligase it was divided by two to get the average (or "half interface"), as to represent the area *between* the POI and E3 ligase.

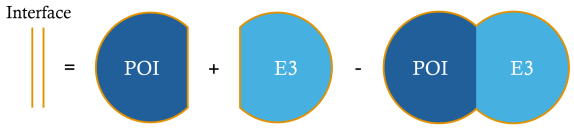


Figure 3.1: Calculations for interface area of a complex.

The SASA of the individual amino acids between the isolated proteins and complex was compared, and if the SASA differed the amino acid was defined to be part of the interface. From this, the SASA of all amino acids in the interface (from both the POI and E3 ligase) was calculated. To get an overview, the amino acids were grouped by their types: Positive, negative, polar, aromatic, aliphatic, and special. Special were defined as Glycine, Proline Cystine and Selenocystine. The SASA for each amino acid type was then calculated.

As a reference, the half interface area was calculated for 314 antibodies, from a dataset provided by Leonardo De Maria. The PDB IDs of these antibodies are provided in Appendix B.2. Likewise, the half interface area and amino acid composition of the interface was calculated for 14 molecular glues, whose PDB IDs are found in the same appendix. The PDB IDs of the molecular glues was obtained by searching the PDB.

3.2 Results

3.2.1 PDB Reference Structures

Nine crystal structures were selected as representatives for each complex group and are presented in Table 3.1. The POI (blue) and E3 ligase (cyan), along with the crystallized PROTAC for each representative PDB structure, are shown in Figure 3.2. Any crystallized adaptor proteins are not displayed in the Figure but are listed in Table B.1 in Appendix B.3.

Additional crystal structures of PROTAC ternary complexes were later identified in the PDB but were not listed in PROTAC-DB. Due to the project plan was determined and already included many experiments, these crystal structures were not examined. The corresponding PDB IDs for these structures are provided in Appendix B.2.

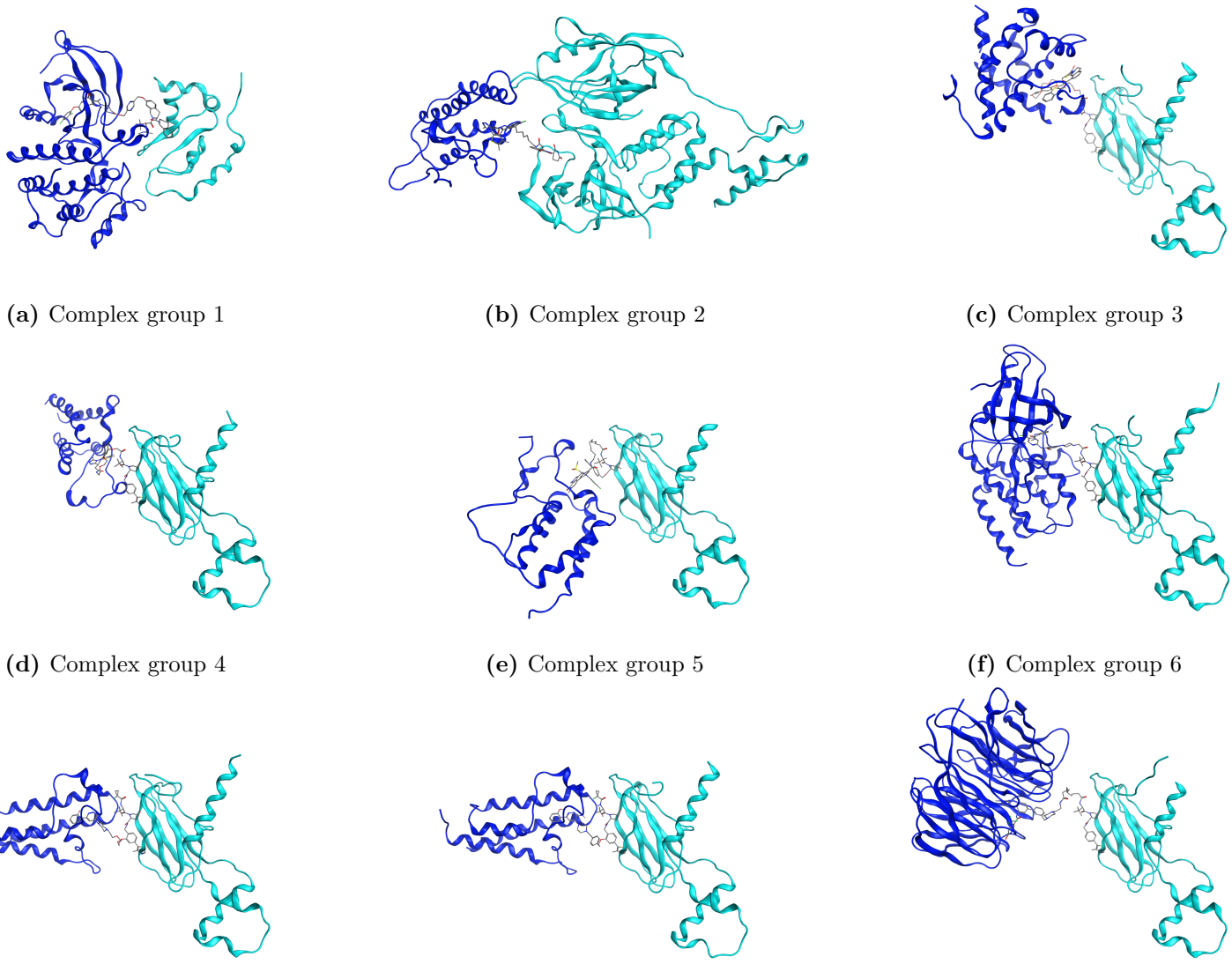
3.2.2 Predicted Structures from Full-length Sequence

AlphaFold-Multimer was unsuccessful in predicting the ternary structures primarily due to instances where the POI and E3 ligase either did not make contact or were misoriented when in contact. All predicted UniProt structures can be seen in Figure 3.3. However, AlphaFold-Multimer effectively folded the individual POI and E3 ligase, as evidenced by the low RMSD values in Table 3.2. This Table presents the median RMSD values for the POI and E3 ligase individually, reflecting the five predictions generated by AlphaFold-Multimer for each input sequence. The RSMD of the 5 predicted structures of all predicted complexes are available in Appendix B.4 Table B.3.

Table 3.2: *Median* RMSD of the 5 predictions from AlphaFold-Multimer, for the UniProt POI and E3 ligase.

Complex group	**POI RMSD [Å]**	**E3 RMSD [Å]**
1	0.9 BTK	0.5 CIAP1
2	0.9 BRD4 (BD1)	0.8 CRBN
3	2.1 BCL-xL	0.5 VHL
4	1.2 BRD4 (BD2)	0.6 VHL
5	0.9 BRD4 (BD1)	0.4 VHL
6	2.4 FAK	0.7 VHL
7	1.0 SMARCA2	0.8 VHL
8	0.9 SMARCA4	0.5 VHL
9	0.5 WDR5	1.4 VHL

In many predicted structures, the POI and E3 ligase are not in contact. This often occurred when the predicted structure contained a large number of disordered amino acids. The the full length sequences were often significantly longer than their corresponding reference structures sequence, and the sequence extending beyond that which is found in the PDB were generally more disordered compared to the PDB structures. Note that the RMSD can only calculated with the amino acids in the reference structure, implying that most disorganised amino acids in the predicted structure is not reflected in the RMSD. Also, the reference POI and E3 ligase were superimposed *individually* onto the predicted structure before the RMSD was calculated. The RMSD values between the full and crystallized predicted complexes were high and found not indicate anything else than they do not match well, which Figure 3.3 clearly displays.



(a) Complex group 1 **(b)** Complex group 2 **(c)** Complex group 3

(d) Complex group 4 **(e)** Complex group 5 **(f)** Complex group 6

(g) Complex group 7 **(h)** Complex group 8 **(i)** Complex group 9

Figure 3.2: All reference PDB structures for PROTAC ternary complexes as displayed in Table 3.1. The POI is shown in blue and E3 ligase in cyan. The adaptor proteins are not shown.

3.2.3 Predicting Disorganized-structure-score

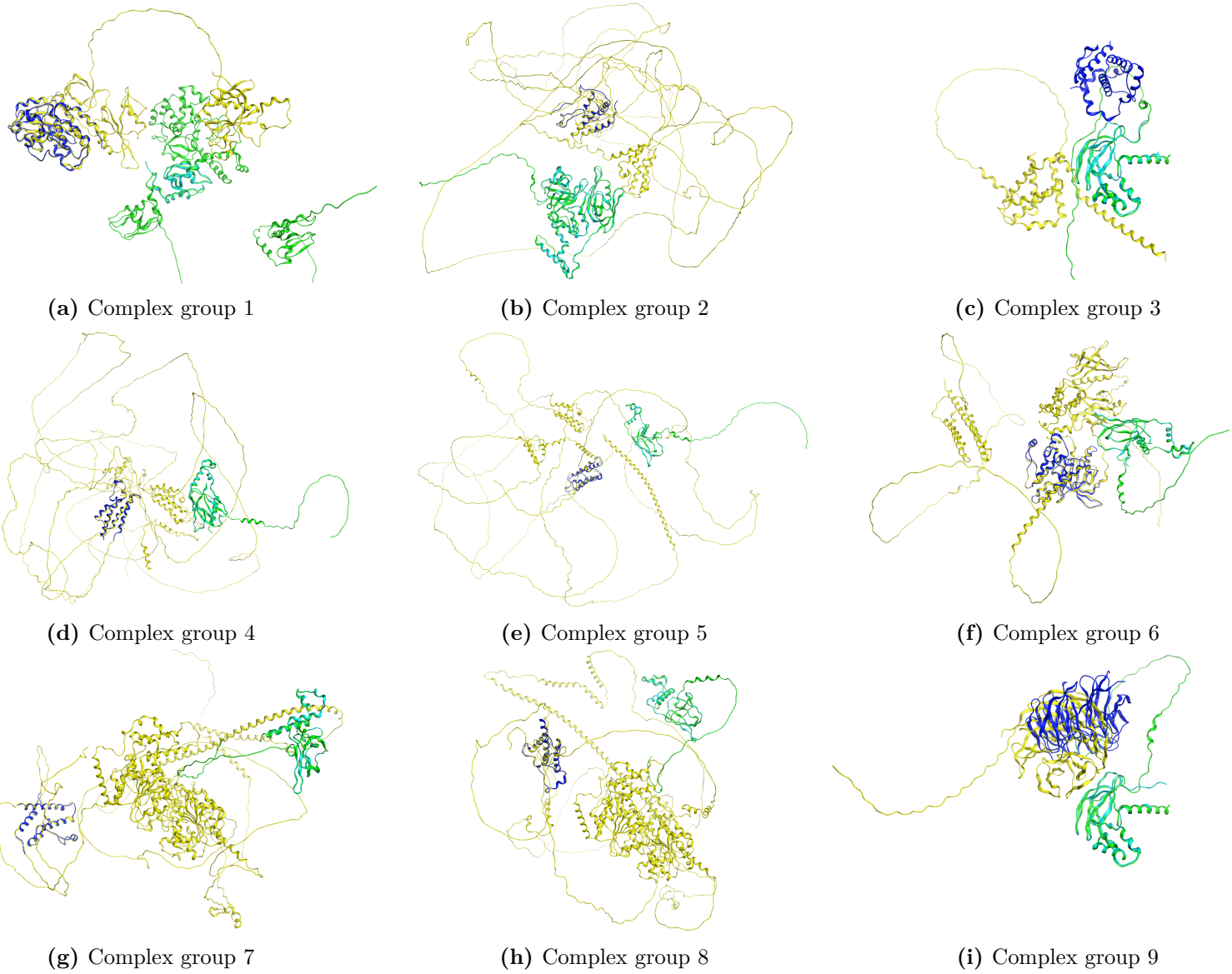
Several predicted structures for the full length sequences were disorganised. To examine if these proteins have a large proportion of disorganised amino acids or if AlphaFold-Multimer failed to predict this structure, the median IUPRED3 score of amino acids are presented in Table 3.3. For clarity, half of the amino acids have a predicted probability of being disorganised which is equal to or greater than the value specified in Table 3.3. Comparing the Table to Figure 3.3, there seems to be a correlation between the how disorganised the predicted structure is to how great the median IUPRED score is.

Table 3.3: Median IUPRED score of the full length amino acid sequence.

Complex group	POI, % disorganised	E3, % disorganised
1	0.21 BTK	0.19 CIAP1
2	0.88 BRD4	0.17 CRBN
3	0.24 BCL-xL	0.43 VHL
4	0.88 BRD4	0.43 VHL
5	0.88 BRD4	0.43 VHL
6	0.30 FAK	0.43 VHL
7	0.51 SMARCA2	0.43 VHL
8	0.53 SMARCA4	0.43 VHL
9	0.16 WDR5	0.43 VHL

3.2.4 Predicted PDB Structures

The predicted PDB structures did not correspond well with their respective references. Figure 3.4 showcases characteristic differences between them. As shown in Figure 3.4a, the relative placements of the POI and E3 ligase often differed from the reference crystal structure. While Figure 3.4b demonstrates relatively good placement, it's important to note that the orientation wasn't accurate. Conversely, Figure 3.4c illustrates poor placement and orientation. Using MOE and applying energy minimization on the predicted structures did not tend to make the position or orientation more alike the reference structure. However, the individual structures of the POI and E3 ligase was predicted accuracy, as indicated by the low RMSD values as shown in Table 3.4. The RMSD values between the full and crystallized predicted complexes were high and found not indicate anything else than they do not match well. Figure 3.5 displays the predicted structure from the full length sequence, with the reference POI and E3 ligase superimposed *individually*. It visually demonstrates how well AlphaFold-Multimer is at predicting the structure of the individual proteins. A key difference between these structures and the predicted structures from the full length sequence is that the POI and E3 ligase were always in contact for the predicted PDB structures, which seems to be due to (another key difference) that there are no large disorganised sequences among the predicted structures.

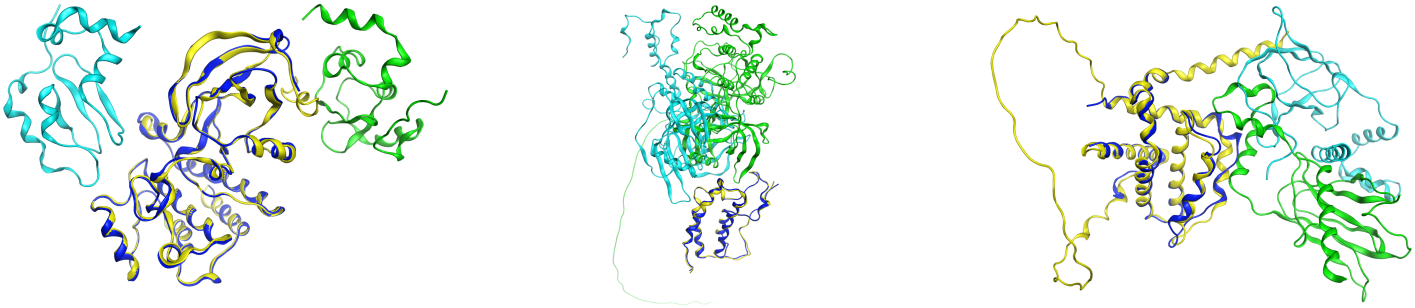


(a) Complex group 1 **(b)** Complex group 2 **(c)** Complex group 3

(d) Complex group 4 **(e)** Complex group 5 **(f)** Complex group 6

(g) Complex group 7 **(h)** Complex group 8 **(i)** Complex group 9

Figure 3.3: The reference POI and E3 ligase were superimposed individually onto the predicted full-length complex of POI and E3 ligase, except for complex group 3 and 9 which the complexes were aligned on the E3 ligase.



(a) Complex group 1 **(b)** Complex group 2 **(c)** Complex group 3

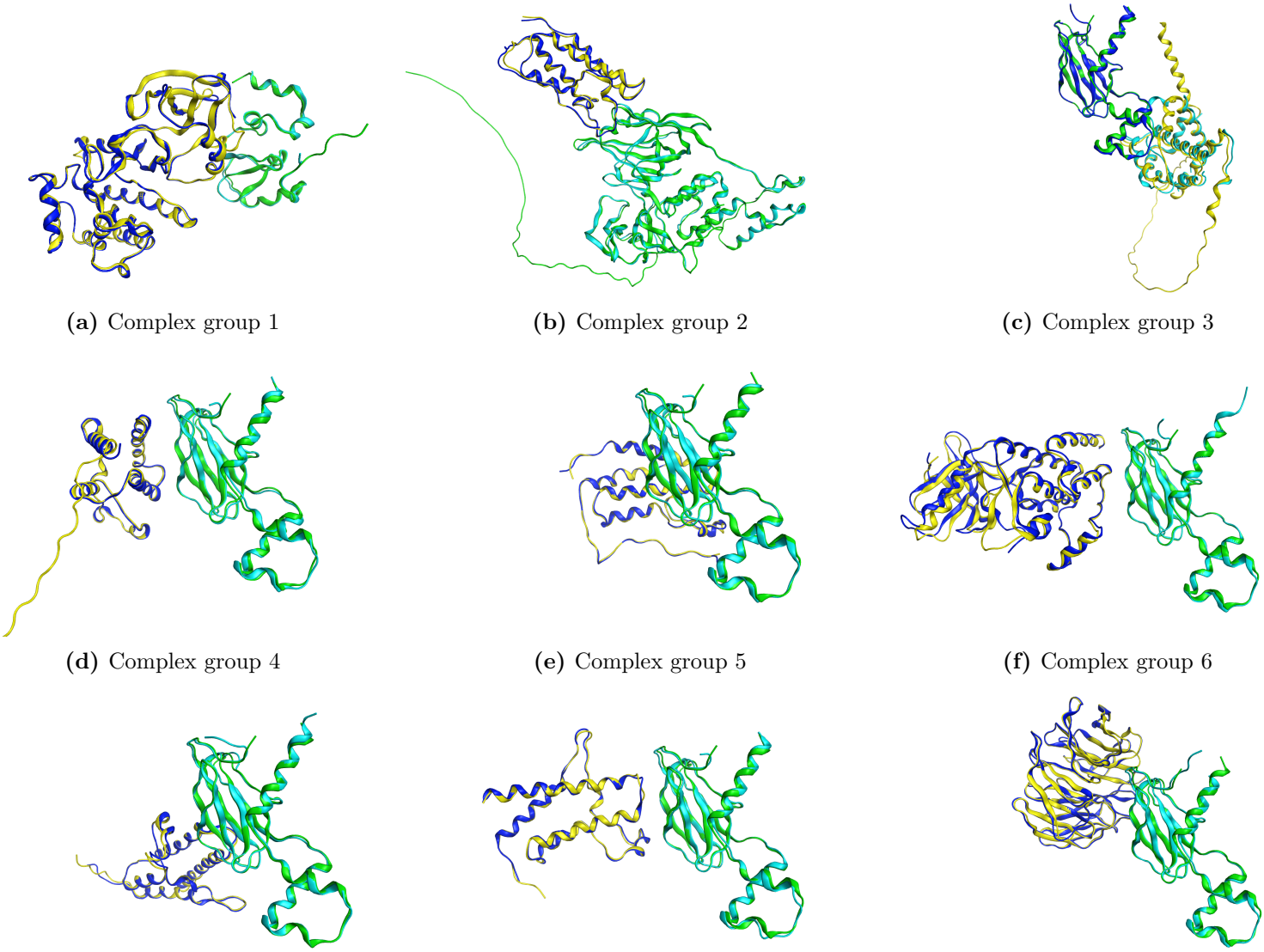
Figure 3.4: Reference crystal structure was superimposed onto the predicted PDB complex, by aligning their POI.

Table 3.4: *Median* RMSD of the 5 predictions from AlphaFold-Multimer, for the longest common sequence of the POI and E3 ligase.

Complex group	RMSD of POI [Å]	RMSD of E3 [Å]
1	0.8 BTK	1.0 CIAP1
2	0.7 BRD4 (BD1)	0.9 CRBN
3	2.0 BCL-xL	0.7 VHL
4	0.3 BRD4 (BD2)	0.8 VHL
5	0.6 BRD4 (BD1)	0.7 VHL
6	3.4 FAK	0.7 VHL
7	1.3 SMARCA2	0.8 VHL
8	0.5 SMARCA4	0.8 VHL
9	0.5 WDR5	0.7 VHL

The similarity between the reference ternary structure of complex group 7 and 8 in Figure 3.5 is not unexpected, as a BLAST of their full-length sequences displayed that they were quite similar. A matrix of all pairings of POI and E3 ligases and their respective sequence identity from BLAST is displayed in in Figure B.1 in Appendix B.3.

When visualizing the protein surfaces of the reference crystal structures, a notable 'gap' between the POI and E3 ligase was observed among a couple of them. This gap was typically occupied by the PROTAC, but not always. By applying energy minimization to the reference structures, it tended to slightly reduce this gap. The reference crystal structure with the largest observed gap is displayed in Figure 3.6.

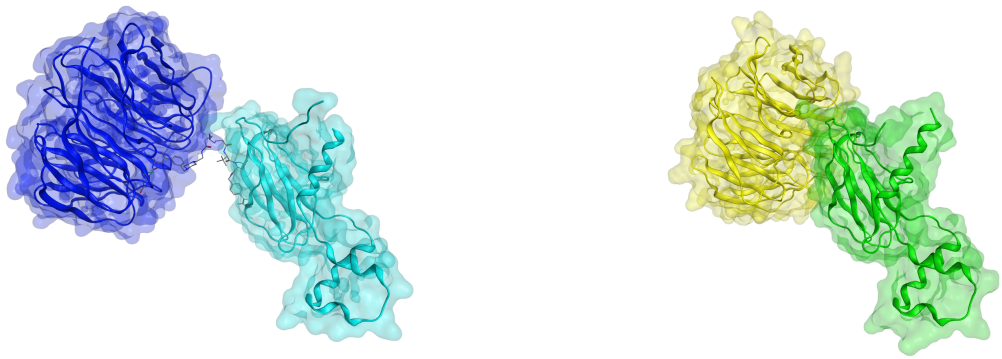


(a) Complex group 1 **(b)** Complex group 2 **(c)** Complex group 3

(d) Complex group 4 **(e)** Complex group 5 **(f)** Complex group 6

(g) Complex group 7 **(h)** Complex group 8 **(i)** Complex group 9

Figure 3.5: Reference POI and E3 ligase were superimposed individually onto the predicted PDB complex of POI and E3 ligase.



(a) Complex group 9, PDB reference **(b)** Complex group 9, AlphaFold-Multimer

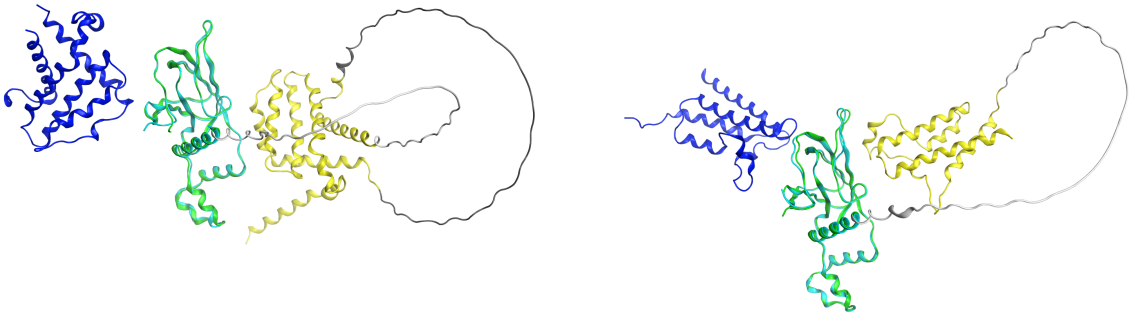
Figure 3.6: Highlighted surface of the reference and predicted complex, from complex group 9.

3.2.5 Docking Experiment

Due to AlphaFold-Multimer's inability to generate accurate PROTAC ternary structures from both full-length and PDB sequences, alternative methods were explored to generate 3D structural information. A docking experiment was conducted using the MOE software on the crystal structure 5T35 from complex group 3, which included its PROTAC, warhead, and E3 ligand. The docking attempts were successful for the warhead but unsuccessful for both the PROTAC and the E3 ligand, likely due to the shallow binding pocket of the E3 ligase (VHL). Given that VHL was present in 7 out of the 9 reference structures, it was determined to be outside the scope of this thesis to develop methods to accurately identify binding pockets and poses for the ternary complex as an alternative source of 3D structural information.

3.2.6 Predicted Structures for Artificially Linking the POI and E3 Ligase End-to-End

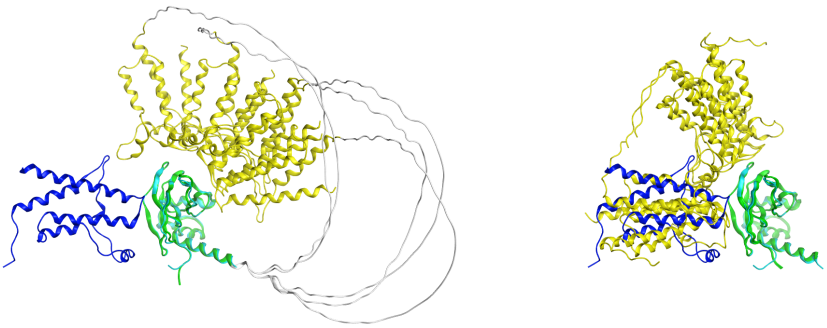
Attempts to link the POI and E3 ligase end-to-end with a 50 G linker did not yield accurate reproductions of the ternary complex in the conducted experiments. Due to time constraints, this strategy was only applied to complex groups 3 and 7. In complex group 3, the POI and E3 ligase exhibited median RMSD values of 1.859 and 0.793, respectively, in comparison to their reference structures. Similarly, in complex group 7, the POI and E3 ligase displayed median RMSD values of 1.859 and 0.917, respectively.



(a) Complex group 3 **(b)** Complex group 7

Figure 3.7: Reference POI and E3 ligase were superimposed onto the predicted complex POI-50G-E3 ligase with a 50G linker, by aligning on their E3 ligase.

An interesting observation was the discernible differences in predictions between AlphaFold2 and AlphaFold-Multimer, as depicted in Figure 3.8.



(a) Complex group 7, AlphaFold2 **(b)** Complex group 7, AlphaFold-Multimer

Figure 3.8: Reference crystal structure was superimposed onto the predicted complex, by aligning their E3 ligase.

3.2.7 Predicted Structures for Artificially Linking the POI and E3 Ligase at their Interface

Linking the POI and E3 ligase at the interface resulted in chimeric proteins (nPOI-linker-cE3 and nE3-linker-cPOI) which were inputted into AlphaFold-Multimer. However, these predictions did not reproduce any ternary structure. Figure 3.9 displays a chimeric complex from complex group 3, linked at the protein-protein interface with two glycine linkers.

Despite testing various linker lengths and types in complex groups 1, 2, 3, 7, and 9, AlphaFold-Multimer failed to accurately predict the ternary complexes. Attempts using two linkers of different lengths also did not yield improved results. The RMSD values for POI and E3 ligase, with linkers of various length and composition, are detailed in Table 3.5, which shows the lowest RMSD values from five predictions for each structure compared to their reference structures.

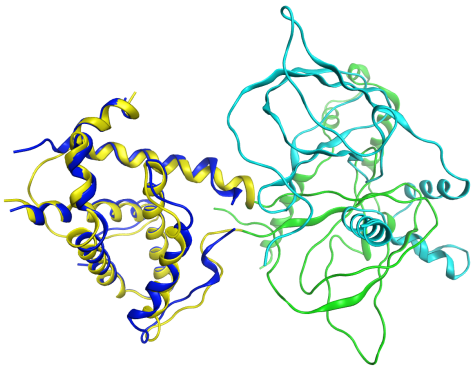


Figure 3.9: Reference POI and E3 ligase were superimposed onto the predicted PDB complex of nPOI and cPOI & nE3- and cE3 ligase, by aligning their POI.

The hypothesis that proline residues could influence the relative rotation between the POI and E3 ligase was tested. However, the results did not follow any obvious pattern between the length of the proline linker and the relative orientation. Instead, proline linkers generally showed less variation in orientation than glycine linkers of the same length among the five predictions of the same complex. Attempts to minimize energy of the predicted chimeric structures in MOE, to correct any discrepancies in relative rotation, were largely ineffective.

Table 3.5: *Lowest* RMSD (Å) of the 5 predictions from AlphaFold-Multimer, for the nPOI and cPOI, & nE3- and cE3 ligase.

	CG1		CG2		CG3		CG7		CG9	
Linker type	POI	E3	POI	E3	POI	E3	POI	E3	POI	E3
50G					1.8	0.8	0.5	0.9		
0/0	1.0	1.9			2.9	1.3				
1G/1G			1.2	0.8	2.9	1.3			1.1	19.9
2G/2G					2.8	1.3				
3G/3G							1.2	0.9		
1P/1P	0.9	2.3	1.3	0.9	16.0	0.9				
2P/2P	1.0	2.8	1.3	0.8	2.9	1.3				
3P/3P					3.0	1.3				
1P/3P					2.9	0.9				
3P/1P					3.0	0.9				

3.2.8 Predicted Structures for Artificially Linking the POI and E3 Ligase, at their Interface and End-to-End

In complex group 2, the chimeric complex was created by linking the POI and E3 ligase end-to-end using a 50 glycine linker, resulting in the sequence nE3-1G-cPOI-50G-nPOI-1G-cE3. While AlphaFold2 successfully folded the POI with an average RMSD of 0.914 Å and the E3 ligase to 1.461 Å, it did not accurately reproduce the ternary complex as can be seen in Figure 3.10.

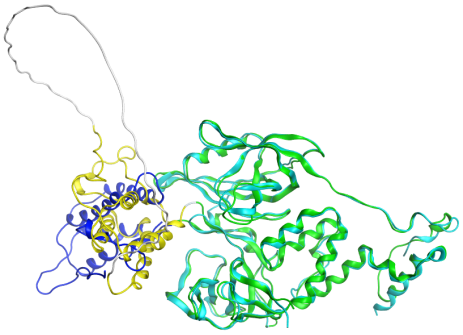


Figure 3.10: Reference POI and E3 ligase for complex group 2 were superimposed onto the predicted chimeric protein (nPOI and cPOI & nE3- and cE3 ligase, joined by a 50G linker). The structures are superimposed via their E3 ligase.

Using MOE, a PROTAC was manually inserted into the structure predicted by AlphaFold2 in Figure 3.10, and the strain on the PROTAC bonds was analyzed using MOE. The results, depicted in Figure 3.11, use color coding to indicate strain levels: green bonds are under minimal strain, whereas red bonds signify significant strain. The red coloring of the bonds indicate that these bonds are under significant strain, and hence would be unlikely to occur naturally. This implies that the conformation of the POI and E3 ligase, which AlphaFold2 predicted, is unlikely to occur naturally, as the binding pose of the PROTAC which mediates this structure is unlikely to occur.

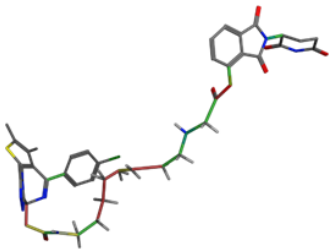


Figure 3.11: A PROTAC forced into binding pockets of the chimeric protein in Figure 3.10, where the POI and E3 ligase are hidden. Green bonds have low strain and red bond are under high strain.

3.2.9 Analysis of Protein Interaction Surface for Predicted and Crystallized Structures

Some interaction surfaces of the complexes generated by AlphaFold-Multimer appeared qualitatively different from those in the PDB, as illustrated in Figure 3.6 and reasoned around in Section 3.2.8. To investigate these discrepancies, two aspects were analyzed for the reference and predicted structures: the area of the interaction surfaces and their amino acid composition.

The half interface area of all complexes from AlphaFold-Multimer and crystal structures from the PDB are displayed in Figure 3.12. For reference, the half interface areas was calculated for a set of molecular glues and antibodies. Comparing the interaction surface are of the dataset of antibodies to the molecular glues and PROTAC ternary structures, it is noticeable that most antibodies have larger interaction surface than most ternary structures. Note that the only PROTAC complex with Cereblon in Figure 3.12 is 6BOY - all molecular glues in this set, except for 8G46, 8OV6, have Cereblon as their E3 ligase, although bound to different POI. Interestingly 6BOY has very similar contact area as the molecular glues, most of which have the same E3 ligase (Cereblon), despite that the size of the POI among molecular glues vary significantly. Figure B.2 in Appendix B.5 displays the number of amino acids of the POI and E3 ligase for 6BOY and the molecular glues.

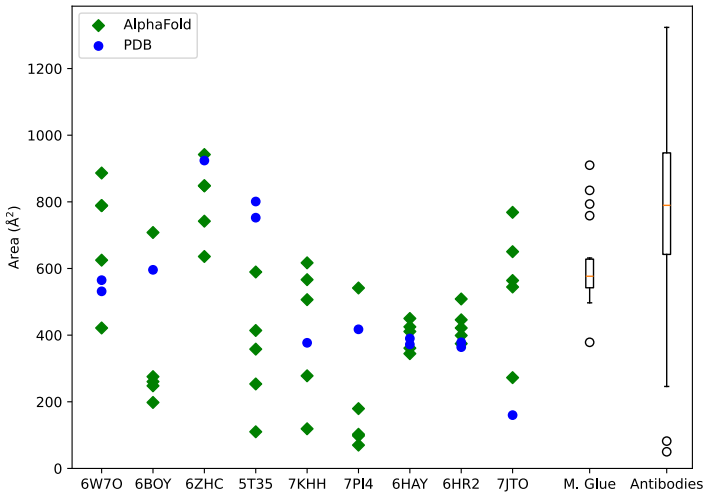


Figure 3.12: Contact areas between POI and E3 ligase. Set of molecular glues and antibodies are used as reference.

Overall AlphaFold-Multimer fails to accuratly predict how large the half interaction surface is, as highlighted in Figure 3.12. AlphaFold-Multimer seems to not be particlarly biased to over or underpredict the size surface area, except for 6HAY and 6HR2, which are the reference structures for complex group 7 and 8, respectively. AlphaFold-Multimer displays no obvious pattern in relation to the predicted structures interaction surface size.

The average amino acid classes of the interface is presented in 3.13. The amino acid classes follow conventional classification, with the 'special' class including cysteine, glycine, and proline.

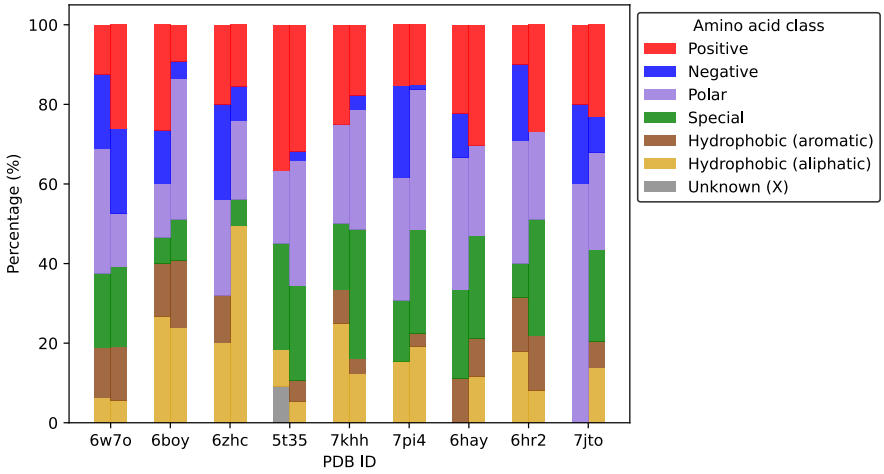


Figure 3.13: Average amino acid classes at the interface between the POI and E3 ligase, for the crystal structure (left) and AlphaFold-Multimer predictions (right).

3.3 Discussion

Linking the complexes at their interfaces serves as a retrospective analysis to assess whether AlphaFold can accurately predict these complexes with additional guidance. This method, particularly when the binding pockets of the POI and E3 ligase are already known, could also be employed prospectively to predict the structure of an unknown complex when the individual structures are known. If the docking of the individual ligands would have yielded reliable results, the two methods could have been combined to predict the ternary complexes.

An unintended consequence of of creating the chimeric proteins (linked at the interface) in MOE was that exported sequences did not include uncrystallized subsequences within the PDB. However, this seems to not affected the overall structure very much. As evidenced by the low RMSD of the POI and E3 ligase in Table 3.5, this seems to not have affected the overall folding. Inspecting the reference structures, it was seen that the start and end points of the uncrystallized subsequence were often spatially close, which explains why it still folded well.

Comparing the predicted structures from the full-length sequences to the IUPRED3 scores, there seemed to be a correlation between how visually disorganised a structure was to how large the IUPRED score was. This indicates that AlphaFold-Multimer did not simply fail to predict these structures but rather that the structures infact have semi-large disorganised regions. However, that is not the reason AlphaFold-Multimer can not predict the ternary structure, as the other experiments did not have large disorganised structures.

It is interesting that despite the different POI with varying sizes, they had quite similar interface areas, and they were similar to the PROTAC crystal structure which shared the same E3 ligase as they did (except for two glues). However, the interface area of the VHL across the reference complexes differed more with a possible plateau at $\sim$400 Å^2. PROTACs are large and flexible, which should allow the POI and E3 ligase larger positional and rotational freedom than what molecular glues allows its POI and E3 ligase. This hypothesis could explain the difference in variation between the interface area of PROTACs and molecular glues, however this is uncertain due to the limited data. Although, if this hypothesis is true it might be easier to predict the ternary structures of molecular glues, however, with the results from this work it seems unlikely as the molecular interactions that stabilize the structure is reasonably a key cause why AlphaFold failed to predict the PROTAC ternary structures.

As the predicted interfaces across the five models in AlphaFold-Multimer often differed, as exemplified by Figure 3.8, it is uncertain how meaningful the average amino acid composition of the interface is for the predicted structures in Figure 3.13. If the interfaces are random, then the average interface amino acid composition should tend towards the average of the surface of the protein. If anything, Figure 3.13 verifies the patternlessness of the predictions from AlphaFold.

Upon visually examining the interaction surfaces of all five predicted structures from the nine complex groups for all described experiments above, and comparing the predicted structures to their reference, no intuitive pattern was found. The implications of this is that the crystallized PROTAC ternary structures deviates from the distribution of structures which AlphaFold2 and AlphaFold-Multimer were trained upon, and reasonably they will struggle to predict new ternary structures as well due to the same reason.

As SMARCA2 and SMARCA4 had a similar crystallized ternary structures, it is not strange that they have similar interface areas in Figure 3.12. However, the variation in the predicted

interaction surface area of these complex groups are much lower and more accurate than any other complex group. This is due to that AlphaFold2 and AlphaFold-Multimer keeps on predicting the same interaction surface of the POI as seen in Figure 3.8, which limits the prediction to a narrow range. However, to explain why a black-box deep learning tool makes a specific prediction, is generally very difficult.

3.4 Conclusions

Although AlphaFold2 and AlphaFold-Multimer excel at predicting individual POI and E3 ligase structures, they struggle to accurately predict PROTAC ternary complexes. AlphaFold-Multimer generates protein-protein interaction surfaces that differ significantly from those observed in crystal structures, as seen when comparing the amino acid composition to that of the reference structure. The variability in predicted interaction surfaces among the five internal models suggests their unreliability in predicting ternary complexes. This limitation may stem from the insufficient number of crystal structure templates in the PDB, which are both used for training and retrieving structural templates for their prediction.

Forcing the PROTAC into the binding pockets of the individual POI and E3 ligase indicated that AlphaFold-Multimer seems to lack implicit knowledge of PROTACs, which validates the results from study [49] where AlphaFold-Multimer successfully predicted ligand mediated interfaces but failed for PROTAC mediated interfaces. The reason why AlphaFold-Multimer predicts ligand mediated interfaces reasonably well may be due to there existing many structures with ligands in the PDB, as the authors of [49] point out, which is another indication that these poor predictions is partly due to a limited number of PROTAC ternary structures in the PDB.

Both AlphaFold2 and AlphaFold-Multimer cannot incorporate molecules in their predictions, posing a significant challenge in accurately predicting ternary structures, which are known to be stabilized by PROTACs. The introduction of AlphaFold3, capable of accounting molecules in the prediction, presents a promising avenue for accurate predictions of complete ternary structures once available.

AlphaFold2 demonstrated a remarkable capability of folding two chimeric proteins, each composed split sequences from two different proteins, into their natural structures. Moreover, AlphaFold2 successfully folded two proteins that were connected from the C-terminus of one to the N-terminus of the other. This suggests that AlphaFold2 is very good at recognizing patterns of individual proteins, within and between different proteins. However, AlphaFold2 did not manage to predict PROTAC ternary complexes using these chimeric proteins. AlphaFold2 did not find the correct relative orientation between the POI and E3 ligase even when it placed the proteins generally in the correct location. Even after minimizing the energy of the chimeric complex, when the POI and E3 ligase were in the correct locations, the correct relative orientation was not achieved. It appears that the local energy landscape in and around the protein-protein interaction surface is too complex for these methods to reproduce respective PDB complex.

Certain PDB structures, particularly from complex group 9, may be too "out-of-distribution" for accurate prediction by both models. This group is characterized by a notably small interface area, less than 200 $Å^2$, suggesting that the observed conformation might be stabilized more by crystal contacts within the lattice rather than intrinsic stability. As AlphaFold should predict intrinsically stable conformation before unstable conformations, it is reasonable that AlphaFold did not predict the reference structure of complex group 9, as the small interface suggests that it is not stable.

Despite these challenges, the reliable predictions of crystallized domains by AlphaFold2 indicate that extracted data could be valuable for machine learning applications. However, while these predictions provide insights into individual protein structures, they fall short of offering direct

information about the entire ternary complex, which would be most beneficial for understanding PROTAC mechanisms and in training machine learning models.

4

PROTAC splitter

The substructures of a PROTAC have distinct roles in the formation of the ternary complex, hence, it is a relevant level of analysis for QSAR when designing PROTACs. With the substructures it is possible to calculate their molecular properties, to cluster PROTACs after individual substructures, and perform a more nuanced data analysis of which PROTACs are active and which properties correlate with the activity.

A common issue in this field is that the molecular representations of a PROTACs substructures are not directly stored alongside the PROTAC. This is problematic when it is the case for public databases, such as PROTAC DB. The only other large public PROTAC database, PROTAC Pedia, fortunately does store the substructures with the PROTAC, however, some annotated substructures seem to be substrates and does not directly match the PROTAC. This makes it more difficult for researchers and the industry to get access to high quality substructure data, which reasonably could slow down the progress of the field. In unpublished work, a transformer based PROTAC splitter was developed by Stefano Ribes, however, it unfortunately had a low validation accuracy and sometimes generated chemically invalid SMILES.

An easy-to-use tool which would split PROTACs into self-consistent substructures and works end-to-end, to allow for integration into existing pipelines, would be broadly applicable for many professionals in this field.

In this second part of the thesis, a novel machine learning tool was developed, which is based on graph neural networks and splits PROTACs into chemically valid substructures that are consistent with the PROTAC. The tool predicts substructure SMILES end-to-end, with a high validation accuracy, and shows the capacity to generalize to new PROTACs.

4.1 Data Preparation

Using the 1813 PROTACs in the curated dataset [44] directly for the training and evaluation of the PROTAC splitter is problematic. The curated dataset contains 1813 PROTACs, which may be insufficient to train a deep learning model. Furthermore, certain types of substructures are more common than others, e.g. the E3 ligands in Figure 4.2a and 4.2b appear in more than 800 PROTACs in the curated dataset, making it heavily unbalanced. Also, it is difficult to make a rigorous test set with these PROTACs, as the risk is that some PROTAC in the test set would share a substructure with some PROTAC in the training set, making it difficult to control for data leakage. So, training a model on the curated dataset could reasonably lead to a model that 1) generally underperforms as it struggles to learn from the limited training data, but 2) may be very accurate for a few overrepresented substructures at the trade of to struggle

for many underrepresented substructures, and 3) simultaneously getting overoptimistic test results, as the same substructures are in the training data.

To solve these three issues, a new set of PROTACs was created. This was done by first splitting the set of warheads, linkers and E3 ligands into a training and test set, where any pair of substructures of the same type has a relatively low tanimoto similarity. Substructures from the training set and test set were then sampled uniformly and recombined into a desired number new PROTACs. The details of how the splitting, sampling and recombination was done is outlined in the sections below.

4.1.1 Preprocessing

A curated dataset [44], based on PROTAC-DB [25] and PROTACpedia [26], was used to create the training set, validation set and the test set for the PROTAC splitter. It contains a set of PROTACs and their respective substructures, which have been processed as follows: Invalid SMILES were removed, and the valid ones had their stereochemistry removed and standardized, which allows the identification of unique and valid molecules. Afterwards, all duplicate PROTAC and substructure entries were removed. It was validated that the warhead and E3 ligand were substructures to respective PROTAC and the linker was inferred as the remaining part, as matching the linker into the PROTAC tended to return multiple substructure matches, especially for shorter linkers. If any of the given substructures did not match into the PROTAC, e.g a warhead, did not match into its PROTAC, it was checked if any other warheads did match into the PROTAC, and if so that other warhead was used instead. It was verified that the substructures matched into the PROTAC with no overlap or gaps, to ensure no double counting or missing of atoms among the substructures. An example of a PROTAC with its annotated substructures are presented in Figure 4.1. It is possible that the annotated substructures rather are substrates for the reaction which creates the PROTAC, and this may not be necessarily a mistake in the database. Regardless, if no warhead or E3 ligand were successfully matched into a PROTAC, that PROTAC was discarded. The final step in the creation of the curated dataset was to add attachment points between the warhead and linker, as well as the E3 ligand and linker. An example of substructures with attachment points are presented in the background in Figure 2.7a.

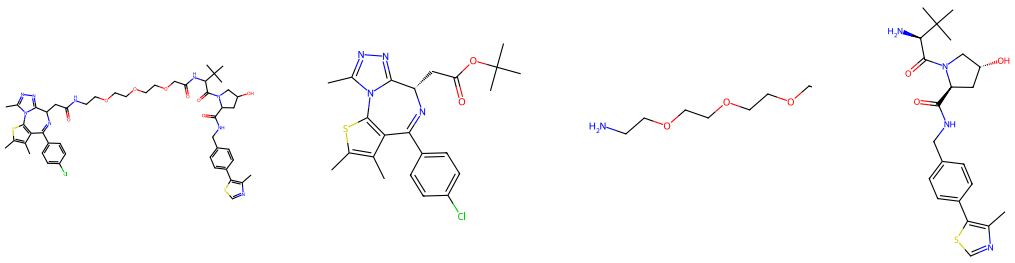
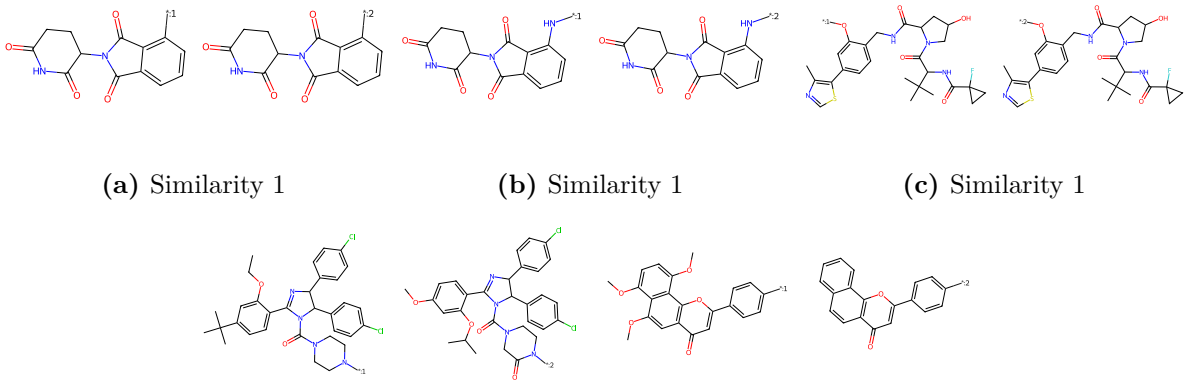


Figure 4.1: A PROTAC (MZ1) and its annotated substructures in PROTAC Pedia. The warhead (center left) does not have a substructure match into the PROTAC (left).

It was identified that some ligands in the curated dataset appeared annotated as a warhead, e.g. thalidomide which is known to bind to an E3 ligase (Cereblon). This occurred for multiple warheads that they can bind to an E3 ligase. This would have the risk of confusing the model, as the model would have been trained to identify the same ligand as both substructure types. It is possible that certain PROTACs are designed to target another E3 ligase as the POI, and hence uses a typical E3 ligand for the task. However, for this project, only warheads that

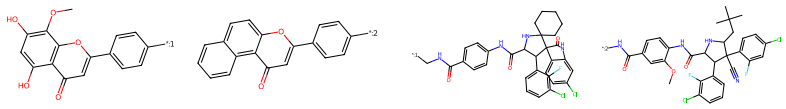
clearly did not target E3 ligases were considered, as to simplify the complexity of the task. Five unique warheads (out of 285 unique warheads, 1.8%) were identified which could bind E3 ligases. These were identified by calculating the tanimoto similarity (based on ECFP4) between all substructures annotated as warheads and E3 ligands, and warheads with a similarity above 0.4 and had an identical graph framework to any annotated E3 ligase was removed. [1]. The warheads that met the criteria are displayed in Figure 4.2. Warheads that almost satisfied this criteria are presented in Figure 4.3.



(a) Similarity 1 **(b)** Similarity 1 **(c)** Similarity 1

(d) Similarity 0.62 **(e)** Similarity 0.46

Figure 4.2: Substructures which were classified as Warheads (left) but were redefined as E3 binders, as the pair have a similarity above 0.4 and have the same graph framework. The best matching E3 binder to this "Warhead" is presented to the right. Ligands annotated as warhead have the attachment point **1* and annotated E3 ligases have **2*.



(a) Similarity 0.45 **(b)** Similarity 0.41

Figure 4.3: Substructures which are classified as Warheads (left), but were not moved to the set of E3 ligands. The substructures and have a similarity above 0.4 to an E3 ligand, but do not share the same graph framework. The best matching E3 ligand to this Warhead is presented to the right. Ligands annotated as warhead have the attachment point **1* and annotated E3 ligases have **2*.

4.1.2 Splitting Dataset of Substructure

The warheads, linkers and E3 ligands were split into a training set and a test set using HDBSCAN. HDBSCAN is a clustering algorithm with a robust mathematical foundation to allow it to identify clusters of any shape and to robustly select clusters despite that may have varying internal density of points (molecules). The downside is that it may be slow for large

[1] These criteria was set through the expert guidance of my supervisors

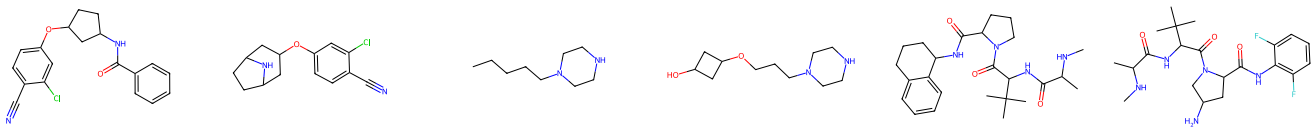
datasets, but that is not an issue in this case. HDBSCAN uses a square matrix of tanimoto similarities between every pair of substructures for each type, to cluster the substructures after specified maximally allowed tanimoto similarity between any pair of molecules between any two clusters.

The Morgan fingerprint was calculated for warheads and E3 ligands as it is the standard fingerprint. The path fingerprint was used for the linkers, with the motivation to use Path fingerprints for linkers being that the Path fingerprint is also a common fingerprint, but may be particularly suited for capturing linear features and to better distinguish between intuitively similar linkers than the Morgan fingerprint.

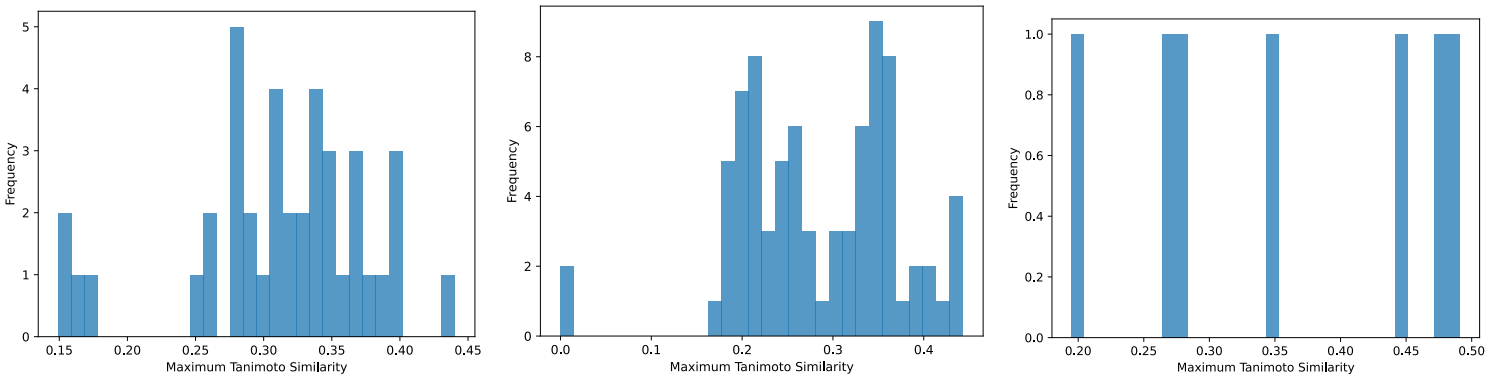
The substructures without attachment points were used, as it was unclear how artificial atoms would be interpreted by the fingerprint, and such they were not used. However, the linkers' attachment point is often part of a ketone group, which would turn into an aldehyde group if the attachment point would be removed. As linkers can be quite small, a change of a single functional group could significantly alter which molecules it is similar to.

The maximally allowed tanimoto similarity between the training and test set was set to 0.45 for Warheads and linkers, and 0.5 for the E3 ligands. These cutoffs are set with consideration to the final number and size of resulting clusters, and a visual confirmation that the maximally similar pair between the training and test set are indeed dissimilar.

HDBSCAN generates a set of clusters that were used to define a test and training set. A test set was built by selecting the smallest clusters first, so that the test substructures will have a high diversity. A higher diversity among the test set would give a better assessment of how good the model is at extrapolating, as a more diverse test set would cover a larger chemical space. By selecting the clusters for each substructure type, 15.3% of all warheads were designated to the test set, as were 8.5 % of the linkers and 13.0 % of the E3 ligands. It is not exactly possible to control the proportion of substructures which are designated to the test set, as this depends on the distribution and sizes of the clusters that can be combined to make a reasonably sized test set. The resulting test set was validated to have no molecule which was similar to any training molecule above the maximally allowed tanimoto similarity. Figure 4.4 displays the maximum tanimoto similarity between the test molecules and any training molecule. The exact count of substructures, murcko scaffolds and graph frameworks in the curated dataset, among the training subtructures and the test substructures is presented in Appendix C.1.



(a) Most similar warheads (sim. 0.44) **(b)** Most similar linkers (sim. 0.44) **(c)** Most similar E3 ligands (sim. 0.49)



(d) Max. sim. test and train warheads **(e)** Max. sim. test and train linkers **(f)** Max. sim. test and train E3 ligands

Figure 4.4: Most similar pair of substructures of each type between training and test sets (a, b, c), and the maximum similarity between all test substructures to any training substructure of the same type (d, e, f).

The test set of warheads, linkers and E3 ligands were split into three parts, where each part had close to equal number of substructures of each type in it. For clarity, the full test set of substructures consists of three splits. Each split has its own set of warheads, linkers and E3 ligands, that is not shared with any other test set or the training set.

4.1.3 Recombining Substructures into PROTACs

New PROTACs were created by recombining substructures into PROTACs. To make a set of PROTACs which consists of a balanced distribution of substructures, Butina clustering was used to cluster the warheads, linkers and E3 ligands for the training and each split of the test set. The Butina clustering algorithm is outlined in Appendix C.2. The distribution of the sizes over each cluster is presented in Figure C.3 and C.4. Each Butina cluster contains a set of substructures which tend to be more similar to molecules within the group than outside the group, however this is not always true for every molecule and an example is presented in Appendix D.1.

Butina clustering was used as it can split clusters that have varying internal densities into multiple clusters which should tend to have a more uniform density than its parent cluster. In contrast, HDBSCAN is built to identify clusters that may have varying internal densities, and varying its cutoff parameter would set the most dissimilar molecules as noise and make the already established clusters smaller. The goal is to "granularize" the clusters and uniformly sample within these sub-clusters, which theoretically Butina clustering should be better suited for than HDBSCAN. Tangentially related, the reason why Butina clustering was not used for splitting the set of substructures is due to that it does not directly control against data leakage, as HDBSCAN does. An example of how Butina clustering can lead to data leakage is presented in Appendix D.1.

Using the attachment points as a guide, the substructures were recombined into PROTACs by sampling one of each substructure type. Firstly, it is made sure that all substructures with attachment points are sampled once (as to use all available information), and if all unique substructures have been sampled once then the substructures will from this point be sampled with uniform probability: A Butina cluster is chosen at random with uniform probability, a substructure (without attachment points) within this cluster is chosen at random with uniform probability, and one attachment point is selected for the substructure with uniform probability. Also, it was verified that all PROTACs in all datasets were unique. This method of selecting the substructures will produce a less imbalanced training and test set. An example of the distribution over how many substructures was sampled from each cluster is displayed in Figure C.5 in Appendix D.

To reassemble the substructures into PROTACs, a bond order must be set between the substructures. The bonds between the substructures were uniformly selected between single, double and triple bonds (up till the order that would not violate the octet rule of electron valency). It would also be reasonable to select bonds based on the bond the sampled substructure had previously, which could improve the models accuracy on PROTACs which have been synthesized. However, as most bonds for all unique substructures are single order bonds, as outlined in Table C.2, the model may associate single bonds with the boundary between the substructures, making it worse at predicting non-single bond boundaries. So, the motivation for selecting a 'random' bond order is that the model may become more robust if it is trained on more diverse data, and may avoid an eventual pitfall of failing to generalize to other bond orders.

The validation set of PROTACs was created by randomly selecting 20 % of the training PROTACs and moving it to the Validation set of PROTACs. As the Validation set is randomly split, it will span the same chemical space as the training set and will gauge the PROTAC splitter's ability to learn overall and ability interpolate between known datapoints in chemical space.

It is also relevant to examine the performance of the PROTAC splitter in the case where one or two substructures are unknown to it. As such, three test sets were created where one substructure type will be unknown for the PROTAC splitter, but using the corresponding test substructure instead, and three test sets were also made where two substructures will be unknown. The naming of the resulting sets and their substructure composition is presented in Table 4.1, where "Text ..." corresponds to which substructures was from the test set.

Table 4.1: Substructure composition of PROTACs in each dataset. Train denotes that the corresponding substructure is part of the training set, where as test substructures are not.

Dataset	**Warhead**	**Linker**	E3 **ligand**
Training	Train	Train	Train
Validation	Train	Train	Train
Test PROTAC	Test	Test	Test
Test Warhead	Test	Train	Train
Test Linker	Train	Test	Train
Test E3	Train	Train	Test
Test Warhead-Linker	Test	Test	Train
Test Warhead-E3	Test	Train	Test
Test E3-Linker	Train	Test	Test

Butina clustering uses a parameter which defines a cutoff value if two molecules are "neighbours" or not based on their tanimoto similarity. With larger cutoff values Butina would create larger clusters, and it is unknown which cutoff is suitable. As such, three datasets of recombined PROTACs were created with a cutoff value of 0.67, 0.33 and 0.00, to examine its effect on the model performance. Note that a cutoff value of 0.00 is equivalent to each molecule being its own cluster.

Based on the resulting number of substructures in the training set, it is possible to generate 12 million unique training PROTACs by exhaustively generating every possible combination of warhead, linker and E3 ligand (*239 * 952 * 53*). As more training data tends to result in better machine learning models, three datasets of different sizes was generated. This to evaluate if there is an effect of the size of the training data on the model performance. Millions of training PROTACs could have been generated, however, to generate a practically manageable amount of training and test PROTACs, the size of the training sets were adjusted to the number of substructures in the following way: The size of the smallest training set was set to be equal to the size of the set of training substructures with the most substructures, to ensure that all training substructures of each type will be found in the training set. This resulted in a training set of 952 PROTACs, based on the 952 linkers in the training set. Two more training sets with 3 and 10 times more PROTACs were generated, indicating that each linker will be represented in around 3 and 10 different unique PROTACs. As there exists fewer warhead and especially the E3 ligands, the frequency of each unique ligand will be higher than that of the linkers in the datasets of reassembled PROTACs. The exact number of substructures (as well as Murcko scaffolds and graph frameworks) in the training and test sets are displayed in Appendix C.1. Once all substructures have been utilized once in a dataset, the diversity of substructures will not further increase with more combinations of substructures, however the local boundary regions between the substructures will.

The test sets were generated similarly, where the number of generated PROTACs was defined to the maximum number of the test substructures, multiplied by a factor of at least 4. Consideration was taken to the final number of PROTACs and were scaled with a larger factor, to produce test sets around 200 PROTACs. This to ensure that all intended test substructures are used for the given set and that every substructure is likely to be used in 4 different PROTACs, at least. The exception to this was Test E3, whose size is determined by the number of test E3 ligands in that test split, which was scaled up by a factor of 25 to achieve a similarly sized set of PROTACs as the other test sets.

4.2 Method

4.2.1 Model Architectures

Node predictor

The node predictor has a simple architecture which consists of a set of graph neural network (GNN) layers along with the LReLu activation function after the input layer and hidden layers. It is a classification model which classifies the nodes to belong to either the warhead, linker or E3 ligand, and it was trained with the CrossEntropyLoss. An overview of the model is presented in Figure 4.5. The layer type, number of layers and size of each layer was optimized with the hyperparameter optimization library Optuna [54]. These hyperparameters were also optimized for all other model architectures (see Section 4.2.4 for details).

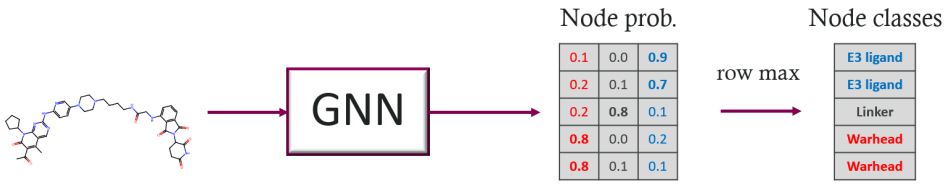


Figure 4.5: Node prediction model. Each row corresponds to a node, and column to a node class.

Boundary node predictor

The minimal necessary information to predict a split is to predict the location of a boundary between each substructure. Using these boundaries, it is possible to infer the rest of the substructures. Hypothetically, it would be a more simple task than node prediction, as node prediction requires every single node to be exactly correct to get a valid prediction, where as the boundary prediction would only need to be confident for two nodes as the inference is relative to the other nodes.

The general architecture of the boundary node predictor is identical to the node predictor, as both only utilizes GNN layers. Although, they differ in how they are trained, how the output is discretized into classes, and the boundary node prediction requires post processing to convert the predicted boundaries to substructures. An illustration of the boundary predictor architecture and determination of classes is illustrated in Figure 4.6.

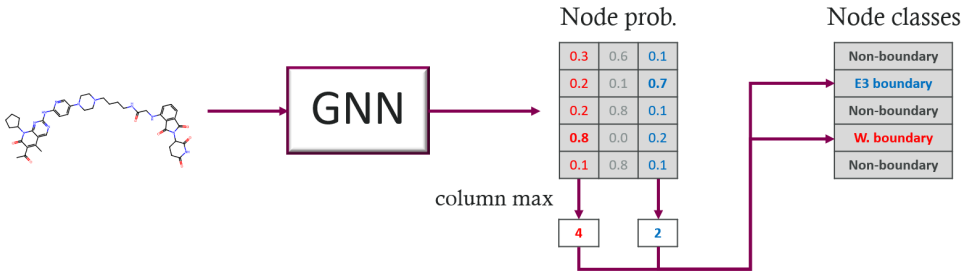


Figure 4.6: Boundary node prediction model. Each row corresponds to a node, and column to a boundary node class.

The boundary node predictor is trained to identify three classes of nodes: Non-boundary nodes, Warhead boundary nodes, and E3 boundary nodes. As only two predicted boundary nodes are desired, the node with the highest probability (over all other nodes) to be the warhead boundary is defined as that boundary, and likewise for the E3 boundary.

When the boundaries have been defined, the linker is identified by the nodes that belong to the shortest path between the boundaries. All nodes that are reachable by these shortest-path-nodes via any path which does not pass through the boundary nodes, are defined as linker nodes as well. The warhead nodes are then defined as all nodes which are reachable by the boundary node via any path which does not pass through the linker, and likewise for the E3 ligand nodes.

Boundary bond predictor - Version 1

The boundary bond predictor predicts the bond between the substructures instead of the boundary nodes. The architectures uses a set of GNN layers as the other Node and Boundary node prediction models, but this time the GNN are used generate a useful hidden representation of the nodes which will be utilized by a Feed-forward network to predict the boundary bonds. It is trained to predict the boundary type (non-boundary bond, warhead boundary bond, or E3 boundary bond) for each bond. During inference, the predicted values for the Warhead bond and E3 bond are summed, and the top 2 bonds with the highest scores are selected as predicted boundary bonds. These are then assigned to the warhead or E3 based on which assignment gives the highest score. An illustration of this model is displayed in Figure 4.7.

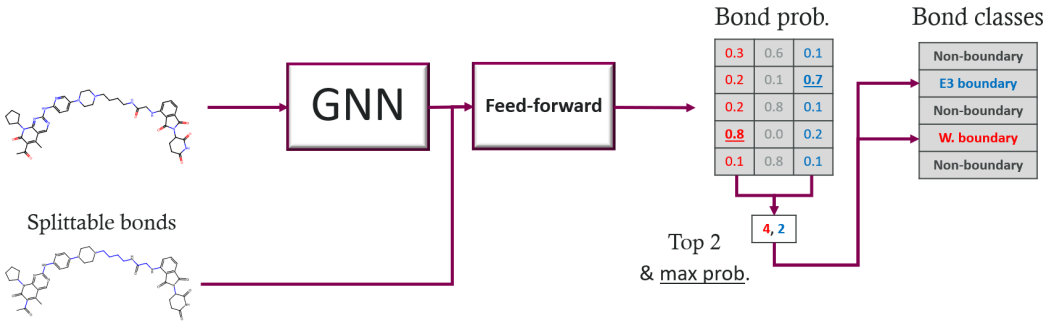


Figure 4.7: Boundary bond prediction model - Version 1. Each row corresponds to a bond, and column to a bond class.

As not every bond would produce a desirable split, the Feed-forward networks are restricted to only make predictions a set of 'splittable bonds'. All bond that are not part of a ring structure, and does not connect a single atom to the rest of the molecule, are defined as splittable bonds. This is because splitting a ring does not split the molecule into two parts. Also, no ligand consists of a single atom, which implies that bonds that connect a single atom are not boundary bonds, hence these are excluded from the splittable bonds.

To prevent the Feed-forward network from learning from the order which the nodes are (artificially) represented in the bond, both orderings (AB & BA) of the node pair is passed through the first layer separately and are summed element-wise, before being passed onto the activation function and the rest of the network. Because addition is commutative, it is ensured that the Feed-forward network is invariant to how the bonds are represented.

The loss was calculated row-wise for the feed-forward network, as to encourage each bond to predict its class as best as possible. Similarly, the loss of the node predictions and boundary predictions was calculated in the same way, by evaluating each individual node. However, the boundary bond prediction also had the loss taken column-wise for the feed-forward network which predicts the bond type, to encourage that only one bond and the correct bond shall be predicted. Mathematically, this is equivalent to the row-wise loss of the transposed output matrix, where the true label is the index of each boundary bond. Both methods were tested separately and indicated to work, so the final loss of the boundary bond prediction was a weighted average of the row-wise loss (weighting of 0.9) and column-wise loss (given a lower weighting of 0.1 as it is more experimental).

With a predicted Warhead and E3 boundary bond it is possible to deduce the substructures. By representing the molecule as a graph, boundary bonds can be easily removed. The remaining graph consists of three disconnected parts which are the predicted substructures, which are

easy to classify into a warhead, linker and E3 ligand, based on which boundary bonds they previously were connected to.

Boundary bond predictor - Version 2

The difference between this version of the boundary bond predictor to the other one, is that it uses two different Feed-forward networks with the same task as the single Feed-forward network: To predict the location and type of each boundary. The first Feed-forward network has the same architecture as in Version 1, and the second Feed-forward network has the task to classify the bonds as boundary bonds or non-boundary bonds. This is outlined in Figure 4.8.

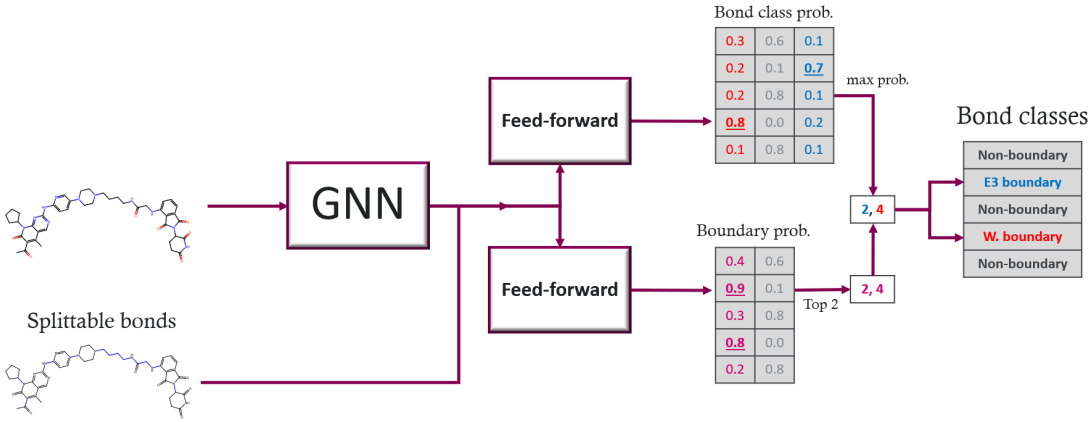


Figure 4.8: Boundary bond prediction model - Version 2. Each row corresponds to a bond, and column to a bond class.

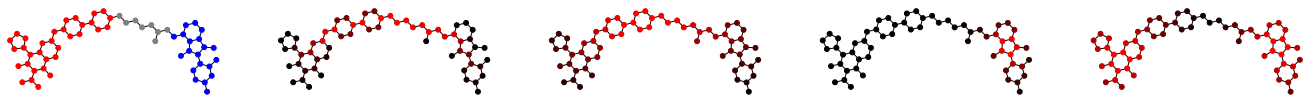
The boundary bond locations are predicted by the second Feed-forward network by getting the top two bonds with the highest probability. The boundary bonds are then given their classes by which ordering (warhead-E3 or E3-warhead) that maximizes their combined average probability, e.g in Figure 4.8 bond 2 and 4 are predicted as boundary bonds, and it is more probable that 2 and 4 are an E3 boundary and warhead boundary respectively than the other way around.

4.2.2 Calculating Descriptors

Chemical descriptors were calculated with the built in function *from_smiles()* in Pytorch Geometric [55], which included atomistic information on the atomic number, how many hydrogens and heavy atoms it is bonded to, chirality, formal charge, number of radical electrons, hybridization, if the atom is part of a ring and if that ring is aromatic. Bond information such as bond type, the stereo configuration of the bond, and if the bond is conjugated or not. These chemical descriptors were encoded into the nodes and bonds of the molecular graph.

Graph centralities can be calculated from the molecular graph be and used as descriptors. Beteweenness and Closeness centralities tend to highlight the linker in PROTACs as seen in Figure 4.9. However, they are calculated with completely different methods and captures different structural information, hence both were used. The Eigenvector centrality tended to highlight just one ligand, and the exact same substructure would not always be highlighted in all PROTACs, as some ligands of the other substructure type would be highlighted instead.

It was reasoned that it would be better with a descriptor which tended to highlighted both ligands. As such, a new descriptor was developed and is named 'Local Eigenvector Centrality'. How the Local Eigenvector Centrality is calculated is defined in Appendix C.6.



(a) True substructures **(b)** Betweenness **(c)** Closeness **(d)** Eigenvector **(e)** Local eigenvector

Figure 4.9: Graph centralities capture structural information of the PROTAC that correlate with substructures.

4.2.3 Evaluation Metrics

To evaluate the PROTAC splitter, the following questions are answered among the Results:

1. Q: How accurately does the model predict the whole PROTAC?
   A: The percentage of correctly labeled atoms.

2. Q: Are the predicted substructures SMILES valid and consistent with the PROTAC?
   A: The percentage of predicted SMILES which RDKit does not return None for & The percentage of predicted substructures that has a substructure match in its PROTAC.

3. Q: What types of mispredictions does the model make for the substructures and how often?
   A: The question is further specified and answered below.

   (a) Q: Are parts of ligands being predicted as linker?
   A: Evaluated with the precision for the warhead and E3 ligand, as well with the recall of the linker.

   (b) Q: Are parts of the linker being predicted as ligands?
   A: Evaluated with the recall for the warhead and E3 ligand, as well with the precision of the linker.

   (c) Q: Is the warhead being predicted as an E3 ligand, or *vice versa*?
   A: Evaluate with the percentage of PROTACs which have any atom mispredicted between the ligands, which will be referred to as a flip.

4. Q: Disregarding any misprediction between the warhead and E3 ligand, how accurately does the model predict which atoms belong to the ligands and linker?
   A: The accuracy over the ligand and linker atoms in the PROTAC & the number of incorrectly predicted atoms.

The definition accuracy, precision and recall are described in Appendix C.5.

4.2.4 Hyperparameter Optimization

To optimize the architecture of each model, the hyperparameter optimization library Optuna [54] was utilized. Default settings was used for Optuna, and it was set to maximize the PROTAC accuracy for the Validation set over 200 trials, for each of the three models. For each trial, the model was trained for 10 epochs, except for the boundary node predictor which was given 15 epochs. The optimization was done with a training dataset of 952 PROTACs, as it would allow the pruining of trials with poor hyperparameters more quickly than the larger training datasets.

The which hyperparameters, and their ranges, that Optuna optimized for each model is presented in Table 4.2. This range for each hyperparameter was set after an initial experimentation and a set of smaller optimization studies, to get an overview of reasonable ranges for a longer optimization. Furthermore, the ranges were partly inspired by typically used values in the field. The ranges was also broadened to include larger and smaller values than was thought to be optimal, to ensure that the optimal combination of parameters would be within these bounds.

Table 4.2: Optimized hyperparameters and the range of values the models were optimized over. *Feed-forward layers are only applicable for boundary bond prediction.

Hyperparameter	**Range**
Num. GNN layers	3, 9 (linear scale)
GNN layer size	100, 250, 500, 750, 1000
GNN layer type	GCNConv, GraphConv, SAGEConv, GATConv, TransformerConv
Learning rate	1e-7 1e-2 (log-scale)
Batch size	1, 8, 64
Dropout	0, 0.5 (linear scale)
Graph Norm.	True, False
Batch Norm.	True, False
Skip connections	True, False
Feed-forward layers*	2, 3, 4

The graph neural network layers were selected from the set of available layers in PyTorch Geometric. GCNConv is the standard layer type to use. GraphConv is designed to be able to process local and higher-order graph structures, which is relevant for this task, as the predictions would likely benefit from taking the whole PROTAC into account. SAGEConv uses an alternative aggregation method than to aggregate from the connected nodes where it samples over a larger neighbourhood instead. As the prediction of PROTAC substructures with graph neural networks, it is interesting to explore various layer types, like SAGEConv. GATConv uses an attention mechanism, which allows it to dynamically weight the importance of each message by the contents of the messages. TransformerConv also uses an attention mechanism, and it is one of the few available layers that can process edge information.

Regularization methods such as Drop out and batch normalization are standard, and testing an available graph normalization layer from Pytorch Geometric was an obvious choice. This is to prevent overfitting and improve generalization to the test sets.

Skip connections are supposed to solve the problems of vanishing gradients and over smoothing in deep neural networks [48]. As the maximum depth was set to 9, which have been understood as quite deep, it was a natural choice to include skip connections in the optimization.

4.2.5 Training

A 3-Crossfold was performed with the test splits (3 different model initialization with the same training and validation data, but with different test data). Each model ran until the lowest validation loss over the last 4 epochs was greater than the lowest validation loss than the 4 epochs before that. With the model stopped, the average and standard deviation of each metric for each epoch was calculated across the different runs. The epoch with the highest average accuracy for the validation PROTACs was defined as the "best epoch". The training curve is plotted up till the best epoch, and all extracted values and plots are at the best epoch.

A set of "dummy" predictions was calculated for each epoch, as to evaluate the accuracy of the model if it had learnt nothing. A 200 PROTACs from the training set was taken at random each epoch, and a simulated output of the model was created by creating a matrix of the same dimensions as the real predictions, but only containing random values between 0 and 1. This was processed by the post-processing and evaluation with the used metrics, as for all other predictions. This is a baseline to compare the training and validation metrics against, to infer how well the model learns. The training and validation sets contain PROTACs from the same chemical space, hence the dummy is a baseline for both. The dummy can be used as a reference for the test sets, however, the test PROTACs span a different chemical space and a dummy prediction on these PROTACs may be slightly different.

4.3 Hyperparameter Optimization Results

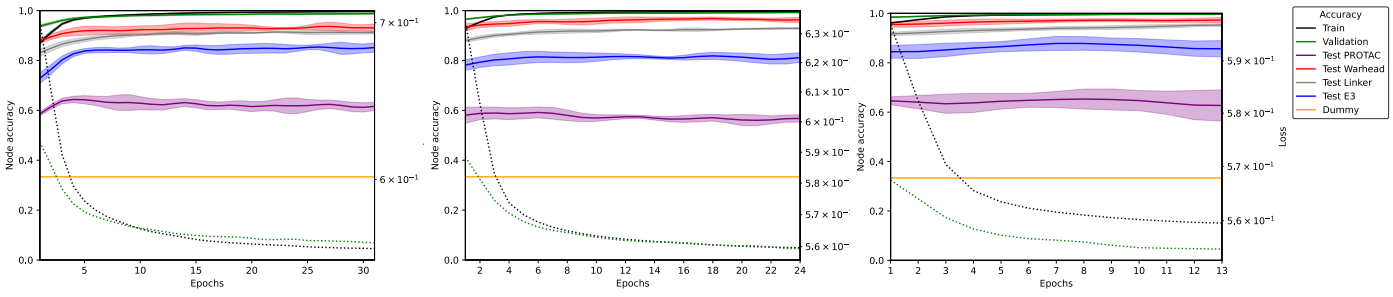
To investigate the effect of training data size, Butina clustering cutoff values, and graph descriptors, the following set of hyperparameters was chosen for the node prediction model based on the results of an initial hyperparameter optimization. From these results, the best training size, cutoff, and set of graph descriptors were chosen, and another hyperparameter optimization was performed for each model using the optimized values for the training size, cutoff, and set of graph descriptors. The node predictor model was chosen over the other models for its simplicity and, hence, lower risk of complex failure modes.

Table 4.3: Chosen hyperparameters to investigate effect of training set size, butina cutoff and graph descriptors.

Hyperparameter	Value
Num. Layers	8
Layer size(s)	[500]*8
Layer type	TransformerConv
Learning rate	1e-5
Batch size	1
Dropout	1e-2

4.3.1 Effect of Training Set Size

The training curves for each test of training set size are displayed in Figure 4.10. Note that test sets with two unknown substructures are not displayed for readability—their general curve is similar to the others and tends towards the values specified in Table 4.4. All models quickly learn from the data and achieve at least a validation accuracy of 90% in the first epoch. The training and validation accuracy tends towards above 99%, and the accuracy of the test sets quickly reaches values close to their final values after around five epochs. The largest training dataset displays slightly more variation than the other two, which have approximately equivalent variation. The average accuracy for all datasets is significantly greater than the dummy accuracy. As expected, the validation accuracy is higher than the test accuracies, with PROTACs having more unknown test substructures showing lower accuracy. Interestingly, Test E3 (blue) is noticeably lower than Test Warhead (red) for each training size. Additionally, the accuracy of the Test Linker for the large training size experiment seems to trend upwards throughout the entire training, even at the best epoch.



(a) 952 Training PROTACs **(b)** 2856 Training PROTACs **(c)** 9520 Training PROTACs

Figure 4.10: Effect of training data size on PROTAC accuracy. Fully drawn lines are arithmetic average accuracy, with a shaded region corresponding to ± 1 standard deviation over the different runs. The geometric mean was calculated for the validation loss (green dotted line) and the training loss (black dotted line). The displayed values were smoothed with a gaussian kernel (σ=1).

The average accuracy and standard deviation at the best epoch are presented in Table 4.4.

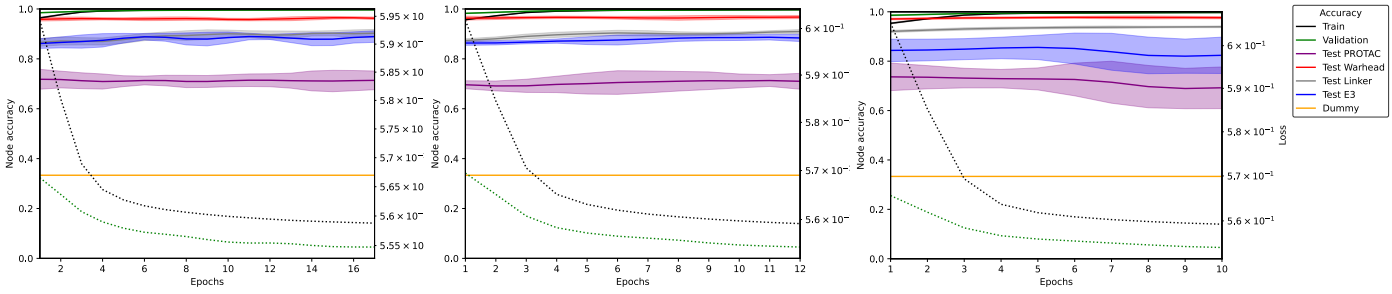
Table 4.4: PROTAC accuracy and standard deviation at the best epoch, for experiments varying the training size of PROTACs.

Dataset	**Accuracy (%), (952 PROTACs)**	**Accuracy (%), (2856 PROTACs)**	**Accuracy (%), (9520 PROTACs)**
Train	99.6 ± 0.1	99.7 ± 0.0	99.7 ± 0.0
Validation	98.9 ± 0.1	99.2 ± 0.1	**99.7 ± 0.0**
Test PROTAC	61.5 ± 2.4	56.5 ± 1.7	63.1 ± 5.9
Test Warhead	92.9 ± 1.4	96.3 ± 0.8	**97.3 ± 1.3**
Test Linker	91.3 ± 0.5	93.0 ± 0.2	**95.3 ± 0.5**
Test E3	85.6 ± 1.6	81.0 ± 2.3	86.1 ± 3.4
Test Warhead-Linker	81.6 ± 1.7	85.4 ± 1.4	**88.8 ± 2.3**
Test Warhead-E3	71.2 ± 1.7	66.0 ± 2.6	73.2 ± 5.0
Test E3-Linker	77.2 ± 2.6	73.5 ± 2.2	**81.0 ± 3.2**
Dummy	33.1 ± 0.4	33.4 ± 0.3	33.4 ± 0.4

The largest training set had some datasets with higher average accuracy than the other two datasets, where the overlap of their ±1 standard deviation was minimal or nonexistent; these are highlighted in bold. Furthermore, observing the bolded accuracies, a clear trend of increasing accuracy with increasing dataset size can be seen. For the datasets where the larger training size did not outperform the smaller training sizes, the larger training size was definitely not worse. As such, the large training size was chosen for the following experiments.

4.3.2 Effect of Balancing Data with Butina Clustering

With the large training size chosen, the effect of the Butina cutoff was investigated, and the training curves of three different cutoffs are presented in Figure 4.11. No significant difference in the training curves was observed between the cutoffs at 0.00 and 0.33. The cutoff at 0.67 showed larger variation in some of the test sets, namely Test PROTAC, Test E3, Test Warhead-E3, and Test E3-linker, all of which contain a test substructure for an E3 ligand. Otherwise, the cutoff at 0.67 resulted in a similar training curve to the other cutoffs.



(a) Butina cutoff 0.0 **(b)** Butina cutoff 0.33 **(c)** Butina cutoff 0.67

Figure 4.11: Effect of Butina cutoff on PROTAC accuracy. Fully drawn lines are arithmetic average accuracy, with a shaded region corresponding to ± 1 standard deviation over the different runs. The geometric mean was calculated for the validation loss (green dotted line) and the training loss (black dotted line). The displayed values were smoothed with a gaussian kernel (σ=1).

The average accuracy and standard deviation at the best epoch is presented in Table 4.5.

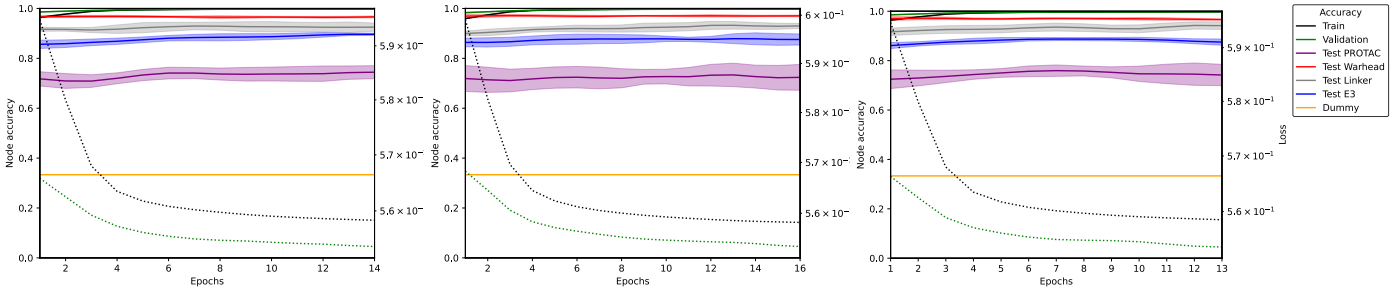
Table 4.5: PROTAC accuracy and standard deviation at the best epoch, for experiments varying the Butina cutoff.

Dataset	**Accuracy (%), (cutoff 0.0)**	**Accuracy (%), (cutoff 0.33)**	**Accuracy (%), (cutoff 0.67)**
Train	99.8 ± 0.0	99.7 ± 0.0	99.7 ± 0.0
Validation	99.7 ± 0.1	99.7 ± 0.1	99.7 ± 0.0
Test PROTAC	72.2 ± 4.5	72.2 ± 3.9	71.6 ± 10.2
Test Warhead	96.2 ± 0.1	96.8 ± 0.8	97.6 ± 0.4
Test Linker	90.4 ± 1.5	91.3 ± 0.8	**93.7 ± 0.3**
Test E3	**90.1 ± 1.8**	**89.3 ± 1.8**	84.3 ± 9.2
Test Warhead-Linker	82.1 ± 2.4	84.8 ± 0.8	**91.6 ± 0.1**
Test Warhead-E3	79.6 ± 2.8	83.8 ± 2.4	79.4 ± 11.1
Test E3-Linker	82.5 ± 3.0	80.8 ± 3.4	77.0 ± 9.3
Dummy	33.3 ± 0.2	33.3 ± 0.2	33.2 ± 0.5

A set of highlighted accuracies are displayed in Table 4.5, which are noticeably larger than their non-highlighted counterparts for the specific dataset. No clear trend of increasing or decreasing accuracy was observed across the three cutoffs, except that the cutoff at 0.67 had larger variation in some of its datasets. As Butina clustering should theoretically help balance the substructure distribution and the results for the cutoff at 0.33 are not significantly worse than 0.00, the cutoff at 0.33 was chosen for the following experiments.

4.3.3 Effect of Graph Descriptors

With the large training size and a Butina cutoff of 0.33 chosen, three sets of graph descriptors were selected, and their training curves are presented in Figure 4.12. The experiment with Betweenness, Closeness, and Eigenvector centrality showed the largest variation over the entire training curve and at the best epoch, most notably for Test PROTAC and Test E3. Otherwise, the training curves for the three experiments are similar.



(a) Betweenness & Closeness centrality.

(b) Betweenness & Closeness & Eigenvector centrality.

(c) Betweenness & Closeness & Local Eigenvector centrality.

Figure 4.12: Effects of graph descriptors on PROTAC accuracy. Fully drawn lines are arithmetic average accuracy, with a shaded region corresponding to ± 1 standard deviation over the different runs. The geometric mean was calculated for the validation loss (green dotted line) and the training loss (black dotted line). The displayed values were smoothed with a gaussian kernel (σ=1).

The average accuracy and standard deviation at the best epoch is presented in Table 4.6. The values of the experiment with Butina cutoff at 0.33, without graph descriptors, is copied into the Table for ease of comparison.

Table 4.6: Node accuracy of PROTACs, for each training, validation and test set, where graph descriptors are added. *No graph descriptors is the experiment with a Butina cutoff at 0.33.

Dataset	**No graph descriptors**	**Closeness & Betweenness**	**Closeness & Betweenness & Eigenvector**	**Closeness & Betweenness & Local Eigenvector**
Train	99.7 ± 0.0	99.8 ± 0.0	99.8 ± 0.0	99.8 ± 0.0
Validation	99.7 ± 0.1	99.8 ± 0.0	99.8 ± 0.1	99.8 ± 0.0
Test PROTAC	72.2 ± 3.9	74.5 ± 1.7	73.0 ± 6.8	73.1 ± 5.3
Test Warhead	96.8 ± 0.8	96.8 ± 0.3	96.9 ± 0.5	96.4 ± 0.0
Test Linker	91.3 ± 0.8	**92.7 ± 1.6**	**93.8 ± 1.2**	**94.6 ± 1.9**
Test E3	**89.3 ± 1.8**	**89.7 ± 0.0**	88.2 ± 2.3	86.7 ± 1.4
Test Warhead-Linker	84.8 ± 0.8	86.0 ± 1.6	**87.4 ± 1.3**	**88.7 ± 2.0**
Test Warhead-E3	**83.8 ± 2.4**	**83.8 ± 0.3**	81.9 ± 3.3	81.9 ± 2.9
Test E3-Linker	80.8 ± 3.4	**82.9 ± 1.3**	**82.7 ± 3.7**	**81.9 ± 3.6**
Dummy	33.3 ± 0.2	33.4 ± 0.4	33.8 ± 0.3	33.2 ± 0.2

A set of highlighted accuracies are displayed in Table 4.6, which are slightly larger than their non-highlighted counterparts for the specific dataset. However, the ranges of their standard deviations partially overlap, so it is uncertain if the differences in accuracies are significant. However, comparing the results of no graph descriptors to all three sets with graph descriptors, there seems to be a benefit to using them. Closeness and Betweenness showed a slight advantage in accuracy for four datasets, whereas the model without graph descriptors showed a slight advantage for two datasets. The results with the eigenvector and local eigenvector were similar in the table. The differences between the models are small, and the choice of highlighting specific accuracies is prone to bias. In any case, the set of Closeness and Betweenness was chosen for the following experiments. Additionally, as there was little difference among all the sets and it was of interest to investigate the Local eigenvector defined in this work, the set of Closeness, Betweenness, and Local eigenvector was also chosen for the following experiments.

4.3.4 Optuna Results

The training size, Butina cutoff, and two sets of graph descriptors were chosen, and the models were hyperparameter optimized for this data, except for Boundary Bond Predictor Version 1, as it was mistakenly believed that the hyperparameters of Version 2 would be the same for Version 1. The optimized hyperparameters for the models are presented in Table 4.7 and Table 4.8. Interestingly, all models set the optimal batch size to 1, and when manually set to larger values, the accuracy suffered. Most optimizations found that these models have better validation accuracy with deeper networks rather than shallower ones. Most optimal models benefited from using graph normalization. No optimal model benefited from using skip connections. The boundary node predictor had a narrower architecture (smaller layer size) than both other models, and the node predictor was narrower than the boundary predictor. There are some differences between the optimal hyperparameters for models with and without the Local eigenvector, but decimal numbers are within an order of magnitude, the integer values often differ by only 1, and only the boundary predictor found two different GNN layer types.

Table 4.7: Optimized hyperparameters for dataset with chemical, Betweenness and Closeness descriptors.

Hyperparameter	Node pred.	Boundary node pred.	Boundary bond pred.
Num. GNN layers	9	8	9
GNN layer size	[500]*9	[1000]*8	[250]*9
GNN layer type	TransformerConv	SAGEConv	TransformerConv
Learning rate	1.2e-5	1.5e-5	8e-5
Batch size	1	1	1
Dropout	2.9e-3	7.0e-4	5.9e-3
Graph Norm.	True	True	True
Batch Norm.	True	False	False
Skip connections	False	False	False
Feed-forward layers	-	-	3

Table 4.8: Optimized hyperparameters for dataset with Chemical, Betweenness, Closeness and Local eigenvector descriptors.

Hyperparameter	Node pred.	Boundary node pred.	Boundary bond pred.
Num. GNN layers	8	5	8
GNN layer size	[750]*8	[1000]*5	[100]*8
GNN layer type	TransformerConv	GraphConv	TransformerConv
Learning rate	2.5e-5	1.8e-5	4e-4
Batch size	1	1	1
Dropout	1.2e-2	3.2e-4	1.1e-2
Graph Norm.	True	False	True
Batch Norm.	True	True	False
Skip connections	False	False	False
Feed-forward layers	-	-	2

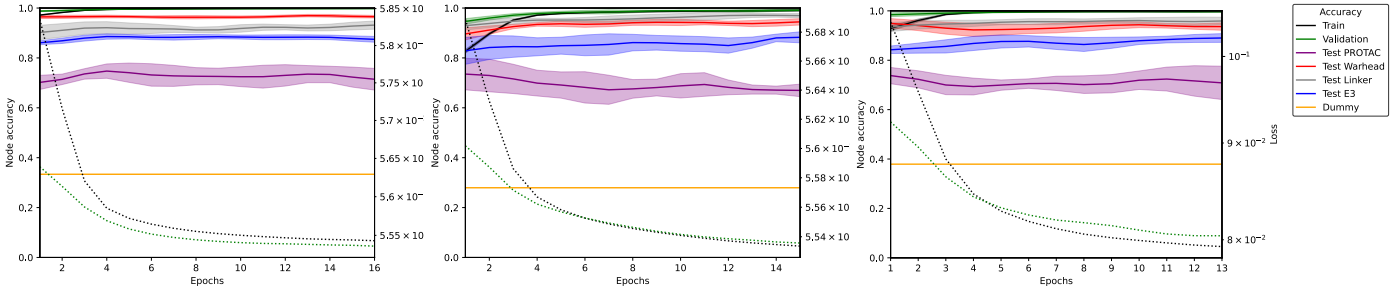
4.4 PROTAC Splitter performance

4.4.1 Training Curves & Accuracy

Using the optimized hyperparameters and their respective data, the models were trained, and their training curves are displayed in Figures 4.13 and 4.14. The training curve of Boundary Predictor Version 1 is presented in Appendix C.7.1 as it was trained using suboptimal hyperparameters. This appendix also contains other metrics calculated for the other models.

The models trained with the Local eigenvector centrality showed greater variation in their accuracies, and their validation loss did not correspond as accurately to the training loss as those without the Local eigenvector. The boundary node predictor with the Local eigenvector was also slower to train than all other models, and it showed signs of overfitting as the training loss continued to decrease beyond the validation loss.

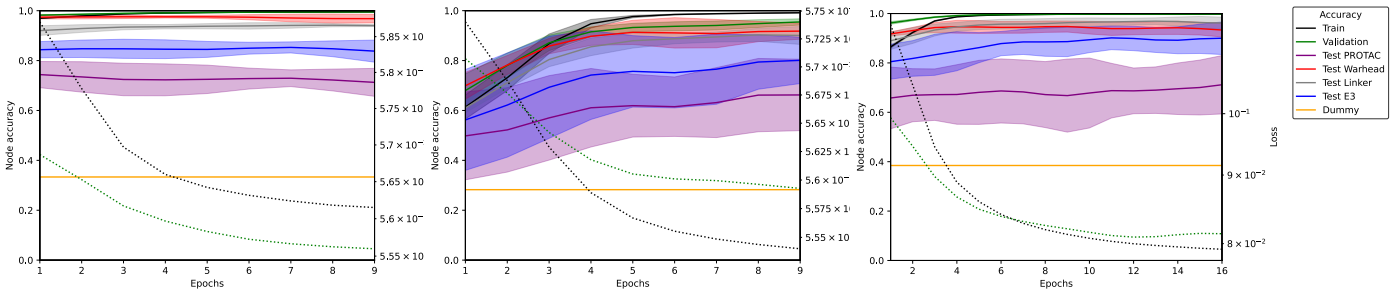
Graph descriptors: Betwenness, Closeness



(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure 4.13: Training curve of models with optimized hyperparameters. Fully drawn lines are arithmetic average accuracy, with a shaded region corresponding to ± 1 standard deviation over the different runs. The geometric mean was calculated for the validation loss (green dotted line) and the training loss (black dotted line). The displayed values were smoothed with a gaussian kernel (σ=1).

Graph descriptors: Betwenness, Closeness, Local Eigenvector



(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure 4.14: Training curve of models with optimized hyperparameters. Fully drawn lines are arithmetic average accuracy, with a shaded region corresponding to ± 1 standard deviation over the different runs. The geometric mean was calculated for the validation loss (green dotted line) and the training loss (black dotted line). The displayed values were smoothed with a gaussian kernel (σ=1).

The average accuracy and standard deviation at the best epoch are presented in Table 4.9 for models without the Local eigenvector and Table 4.10 for models with the Local eigenvector.

Table 4.9: Accuracy of models optimized with Betweenness and Closeness. *Boundary bond pred. version 2.

Training set size	Node pred	Boundary node pred.	*Boundary bond pred.
Training	99.9 ± 0.0	99.1 ± 0.4	99.8 ± 0.1
Validation	99.8 ± 0.0	99.2 ± 0.1	99.8 ± 0.0
Test PROTAC	**70.4 ± 4.6**	66.6 ± 0.5	**73.1 ± 3.5**
Test Warhead	**96.6 ± 0.3**	94.6 ± 1.5	93.6 ± 1.6
Test Linker	93.4 ± 1.9	**96.7 ± 1.2**	**96.1 ± 1.6**
Test E3	86.3 ± 1.5	**89.2 ± 1.0**	**89.1 ± 2.8**
Test Warhead-Linker	86.3 ± 1.7	85.4 ± 0.9	85.5 ± 1.8
Test Warhead-E3	**80.7 ± 1.3**	76.3 ± 4.4	70.6 ± 6.3
Test Linker-E3	80.9 ± 4.2	**83.4 ± 3.7**	**84.4 ± 2.0**
Dummy	33.5 ± 0.9	27.6 ± 0.4	39.7 ± 4.4

All models achieved a training and validation accuracy above 99% and had Test PROTAC accuracy around 65-75%, which is significantly higher than the dummy predictions. The difference between the average accuracies of the models in Table 4.9 is relatively small compared to the standard deviation for most datasets. However, a set of highlighted accuracies is presented in Table 4.9, which are judged to be greater than the other accuracies for the same dataset that are not highlighted.

Table 4.10: Accuracy of models optimized with Betweenness, Closeness and Local Eigenvector. *Boundary bond pred. version 2.

Training set size	Node pred	Boundary node pred.	*Boundary bond pred.
Training	99.6 ± 0.0	99.2 ± 0.0	99.9 ± 0.0
Validation	99.5 ± 0.1	96.2 ± 0.9	99.8 ± 0.1
Test PROTAC	70.5 ± 6.3	66.2 ± 12.3	72.8 ± 12.9
Test Warhead	**96.9 ± 1.3**	92.9 ± 2.9	92.8 ± 3.3
Test Linker	93.8 ± 0.4	90.2 ± 3.2	**96.2 ± 1.8**
Test E3	82.8 ± 5.6	79.5 ± 7.5	**89.7 ± 7.6**
Test Warhead-Linker	**89.0 ± 2.4**	85.1 ± 0.9	**88.5 ± 3.1**
Test Warhead-E3	**78.6 ± 7.2**	61.7 ± 15.3	76.7 ± 19.9
Test Linker-E3	77.0 ± 4.9	72.3 ± 10.4	**85.5 ± 8.1**
Dummy	33.2 ± 0.3	28.6 ± 0.9	38.5 ± 1.3

All models achieved a training and validation accuracy above 99%, except for the boundary node predictor, which is presented in Table 4.10. All models had Test PROTAC accuracy around 65-75%, significantly higher than the dummy predictions. The difference between the average accuracies of the models in the table is relatively small compared to the standard deviation for some datasets. However, a set of highlighted accuracies is presented in Table 4.10, which are judged to be greater than the other accuracies for the same dataset that are not highlighted. The boundary bond predictor performed notably worse with this set of hyperparameters and data than the other models. A key difference between the results of the models that used and did not use the Local eigenvector is that the variation tends to be greater, but the average accuracy is mostly unchanged for most datasets.

4.4.2 Validity

If the predicted substructures cannot be processed into a chemically valid molecule or if more or less than two boundaries are predicted, the prediction is defined as invalid. The node predictor suffers greatly from low validity, as presented in Table 4.11, whereas the boundary node predictor has relatively high validity. The boundary bond predictor has perfect validity due to only being allowed to split bonds that would be valid. Similar validity is presented for the models using the Local eigenvector in Table 4.12. However, the variation in validity with the Local eigenvector is greater for the node predictor and boundary node predictor than without it.

Table 4.11: Validity of models optimized with Betweenness and Closeness. *Boundary bond pred. version 2.

Training set size	Node pred	Boundary node pred.	*Boundary bond pred.
Training	98.6 ± 0.2	99.3 ± 0.1	**100.0 ± 0.0**
Validation	98.2 ± 0.3	99.6 ± 0.1	**100.0 ± 0.0**
Test PROTAC	4.3 ± 2.1	83.0 ± 6.2	**100.0 ± 0.0**
Test Warhead	73.3 ± 4.6	96.7 ± 2.2	**100.0 ± 0.0**
Test Linker	40.0 ± 8.2	96.8 ± 0.4	**100.0 ± 0.0**
Test E3	16.9 ± 4.1	97.8 ± 0.8	**100.0 ± 0.0**
Test Warhead-Linker	24.0 ± 4.8	89.9 ± 2.4	**100.0 ± 0.0**
Test Warhead-E3	4.3 ± 4.9	91.9 ± 3.3	**100.0 ± 0.0**
Test Linker-E3	14.8 ± 5.3	91.6 ± 6.5	**100.0 ± 0.0**
Dummy	0.0 ± 0.0	27.7 ± 2.7	**100.0 ± 0.0**

Table 4.12: Validity of models optimized with Betweenness, Closeness and Local Eigenvector. *Boundary bond pred. version 2.

Training set size	Node pred	Boundary node pred.	*Boundary bond pred.
Training	94.4 ± 0.6	99.4 ± 0.0	**100.0 ± 0.0**
Validation	95.3 ± 1.6	98.5 ± 0.3	**100.0 ± 0.0**
Test PROTAC	4.0 ± 4.2	83.6 ± 6.9	**100.0 ± 0.0**
Test Warhead	70.2 ± 18.8	97.1 ± 1.3	**100.0 ± 0.0**
Test Linker	48.6 ± 10.4	91.1 ± 4.1	**100.0 ± 0.0**
Test E3	12.7 ± 11.0	92.4 ± 6.0	**100.0 ± 0.0**
Test Warhead-Linker	29.6 ± 4.4	86.4 ± 8.6	**100.0 ± 0.0**
Test Warhead-E3	6.6 ± 5.9	92.7 ± 6.1	**100.0 ± 0.0**
Test Linker-E3	7.7 ± 5.7	89.1 ± 6.6	**100.0 ± 0.0**
Dummy	0.0 ± 0.0	31.0 ± 4.1	**100.0 ± 0.0**

4.4.3 Discussion of Worse-Case Predictions

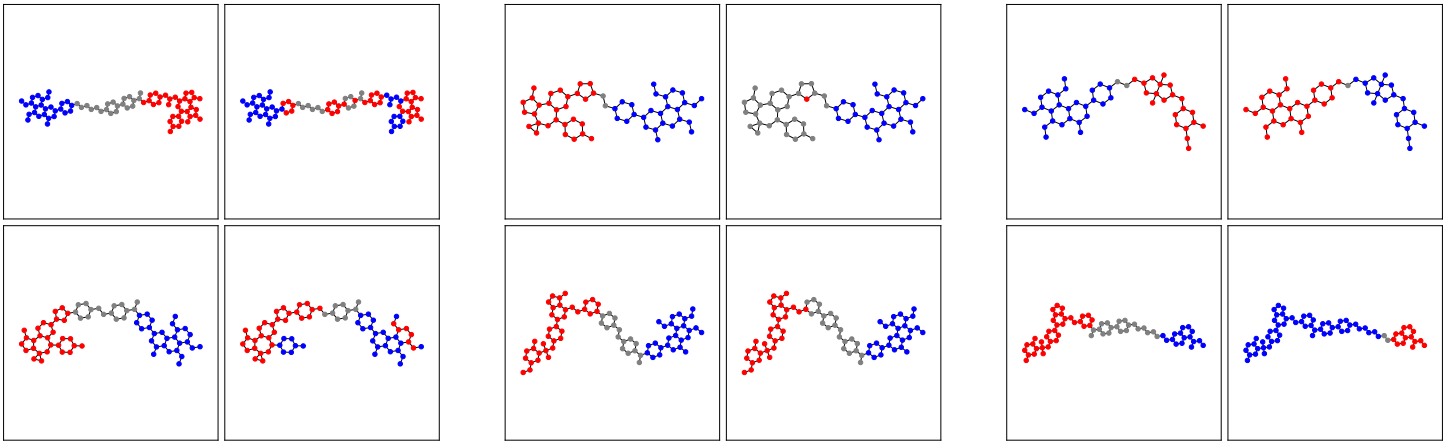
Each model exhibits its own characteristic failure modes/mistakes, and the severity of these mistakes depends mostly on the dataset. The descriptors used may affect the prevalence of these mistakes, but they do not change the nature of the mistakes. A pair of cherry-picked worst-case predictions are highlighted in Figure 4.15 to demonstrate the nature of these mistakes.

The node predictor tends to mispredict nodes within a substructure, which may be a single node or a cluster of nodes. This makes the prediction inconvertible to a valid molecule. Simple

mistakes can be fixed with post-processing, but some are beyond fixing. The node predictor is not forced to predict all classes and sometimes, for short linkers, no nodes are predicted as linker nodes.

Boundary node predictions may completely fail if a boundary node is predicted inside a ring, as it no longer separates the ligand from the linker. Ideally, a set of "separating nodes" corresponding to the "splittable bonds" (but for nodes) would have been developed. However, it was realized that these "separating nodes" do not ensure valid predictions, as exemplified in the figure, where a boundary node prediction separates the linker from the ligand but splits open a ring in the process. Converting those substructures into molecules would not result in valid splits, as all substructures should have only one bond connecting them to the other substructures, never two bonds, as would be the case when a ring is split. However, these single node mistakes that split a ring should be easy to identify and fix with post-processing. The boundary node predictor can hypothetically also predict a linker of zero size; however, this was not observed during the cherry-picking.

The boundary bond predictor can flip the prediction of the ligands, predicting the warhead as an E3 ligand and *vice versa*. This is a failure of the feed-forward network which predicts the bond types incorrectly. The boundary bonds may also be placed on the same side of the PROTAC, which is a failure of the second feed-forward network that predicts the bond locations.



(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure 4.15: Left side displays true substructures, right side displays cherry-picked predictions of the Test PROTAC dataset.

4.5 Evaluating the Best Model

Boundary bond predictor version 2, trained using Betweenness and Closeness, was selected as the best model. Models using the Local eigenvector tended to show greater variation among different test splits, even though the average accuracy was generally the same. This implies that while these models perform better for some substructures and worse for others, their predictions are unreliable overall. Additionally, computing the Local eigenvector for the training set of 9520 PROTACs took approximately 100 minutes, making it less preferable if it does not significantly improve performance. Consequently, these models were not selected as the best model.

When comparing the remaining node predictor, boundary node predictor, and boundary bond predictor version 2, they had similar overall accuracy, with some models performing better on certain datasets. However, boundary bond predictor version 2 consistently produced valid splits, whereas the other two models did not. Valid outputs are highly desired for this tool, especially given the scarcity of PROTAC data. With the current hyperparameters, data, and model architectures, boundary bond predictor version 2 is the best choice. The results for boundary bond predictor version 1 are presented in Appendix C.7.1, as it used suboptimal hyperparameters and generally had worse accuracy than the other models.

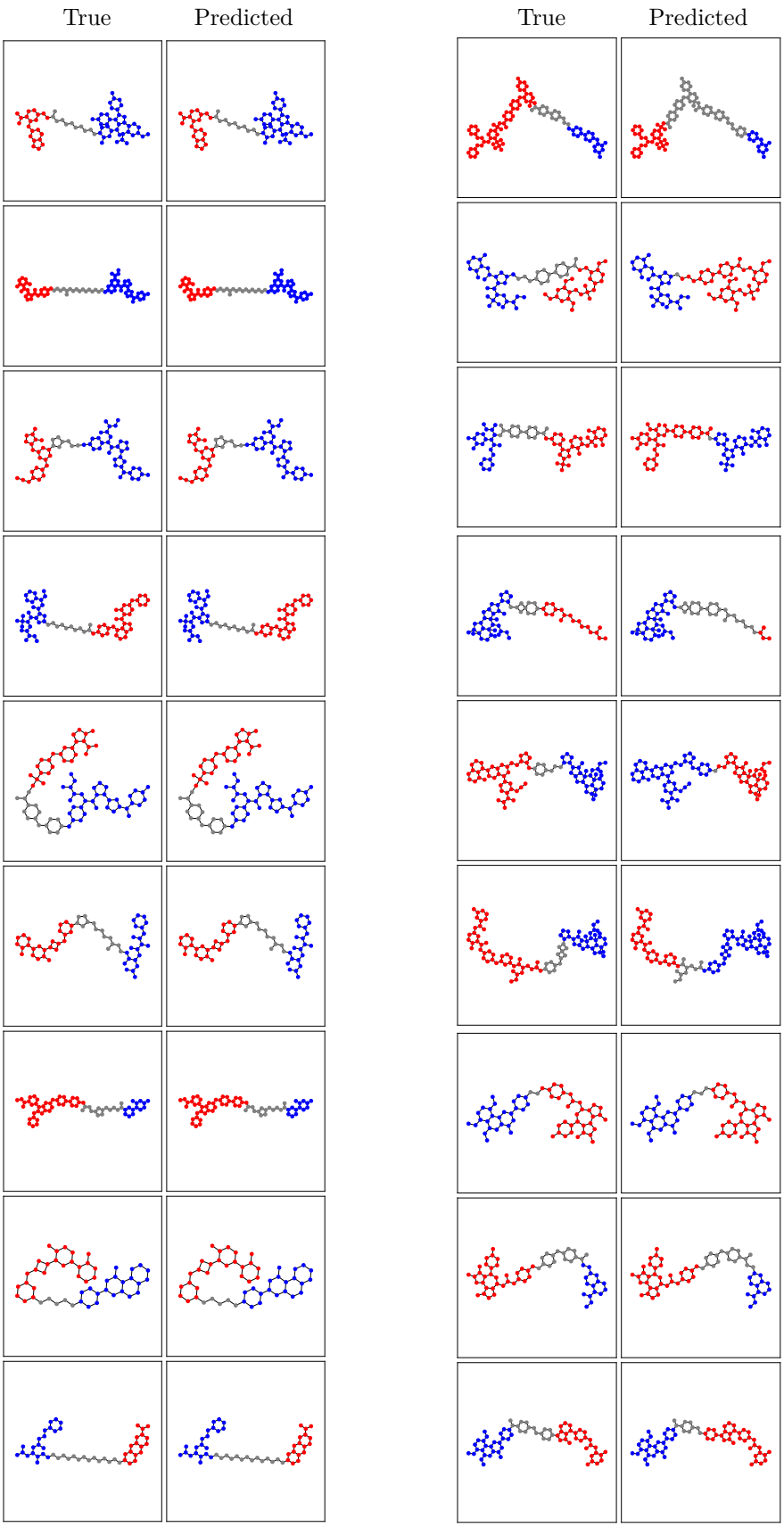
With the best model chosen, it was further evaluated using metrics such as precision, recall, distribution of mislabeled nodes, prevalence of flips, and boundary accuracy with ligand-linker accuracy. All these metrics were calculated for all models and are presented in Appendix C.7. However, the best model will be highlighted here.

A set of randomly selected Validation and Test PROTAC predictions from the best model are displayed in Figure 4.16 to give an intuition of its accuracy for the two sets.

To investigate whether the boundaries are shifted towards the linker or out towards the ligands, the precision and recall of all substructures were calculated. The precision of the linker indicates if the boundaries generally extend into the ligands, as lower precision of the linker can only occur if a true ligand is predicted as a linker. Similarly, a lower recall of the ligands may occur if the boundary is shifted into the ligand or if the PROTAC has been flipped.

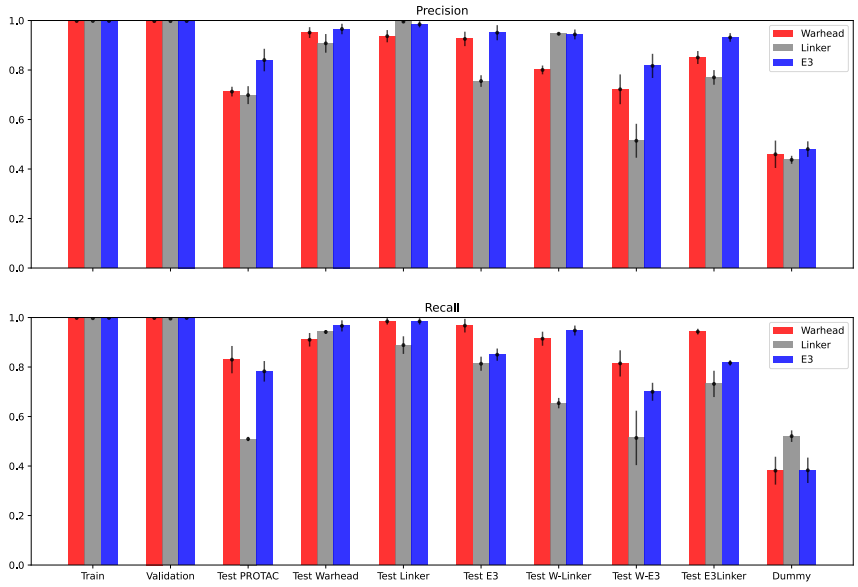
In Figure 4.17, the precision and recall of each substructure are presented. Both recall and precision for the validation set for all three substructures are very high. For the test PROTAC, the recall of the linker is about 0.5, indicating that on average 50% of the true linker is being identified, implying that the boundaries often migrate towards the linker. The dummy recall for the linker is also about 0.5, but this does not mean the boundaries are randomly placed; it implies that the linker tends to be underpredicted in size for the test PROTAC. Additionally, the recall of the warhead and E3 ligand is around 0.7 and 0.8, respectively, indicating that the boundaries also migrate out towards the ligands. These two facts together show that the location of the linker can shift in both directions, but most mistakes occur when the boundaries are placed inside the linker. The precision for the test PROTAC is moderate, with values between 0.7 and 0.8. This means, for instance, that about 70% of the nodes of a predicted linker are correctly labeled as linker. This indicates that the model is fairly reliable in predicting the structures accurately.

The precision and recall for all other test sets are also presented in Figure 4.17, showing that the model is most accurate for PROTACs with fewer test substructures. The case where both the warhead and E3 ligand are unknown is similar to the test PROTAC, indicating a similar conclusion for these predictions. Test warhead-linker suffers most from a low recall of the linker and slightly lower precision of the warhead, indicating that the boundary between the unknown warhead and linker tends to migrate inwards towards the linker. When the E3 ligand and linker are unknown, the recall of the linker and E3 ligand are both moderate, and the precision of the warhead is lower than that of the E3 ligand. As the recall of the warhead is high, any potential flips play a minor role in these results, indicating that the cause is the shifting of the boundaries. An explanation fitting this data is that both boundaries tend to move towards the true E3 ligand, which lowers the precision of the warhead as it starts to predict part of the linker as warhead, thereby lowering the recall of the linker, and when the E3 boundary moves into the E3 ligand, it lowers the recall of the E3 ligand.



(a) Predictions from Validation set. **(b)** Predictions from Test PROTAC.

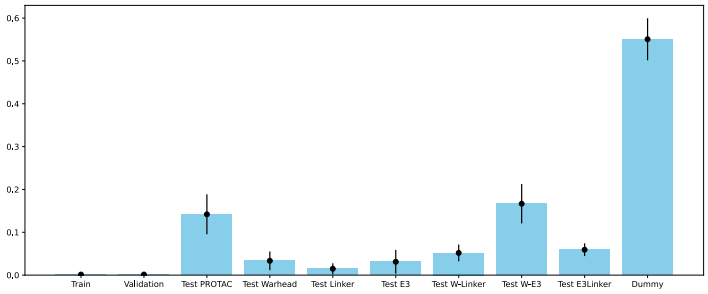
Figure 4.16: Examples of predicted substructures (right) and true substructures (left), for the Validation and Test PROTAC.



(a) Boundary pred. Ver.2

Figure 4.17: Precision and recall of the best model.

As observed among the worst-case predictions, some predictions are flipped. The frequency of flips is presented in Figure 4.18 and Figure 4.16. Very few training and validation PROTACs are flipped. Around 15% of the predictions of Test PROTAC are flipped, similar to Test Warhead-E3 ligand. This indicates that the linker has little effect on inducing a flip in this case. When only the linker is unknown, the frequency of flips is greater than that of the validation set, meaning it does play some role. Test Warhead is similar to Test E3, indicating that the model performs similarly well on both unknown ligand types. Finally, Test Warhead-linker is also similar to Test E3-linker. These results indicate that the model is not particularly more prone to making flipped predictions for one comparable dataset to another. Rather, the risk of a flip correlates well with the number of unknown substructures.



(a) Fraction of predictions that are flipped

Figure 4.18: Flipped predictions

The ligands-linker accuracy is presented alongside the PROTAC accuracy in Figure 4.19. "Ligands & linker" refers to the structure of the PROTAC where the specific identity of the ligands (warhead and E3 ligand) is disregarded, and it is only checked if any mispredictions are made between these two classes. The difference in accuracy in the figure is mainly due to flips, and if the boundary classes were accurately predicted, the PROTAC accuracy would increase to be (nearly) equal to the ligand-linker accuracy. There are rare cases where both

boundaries are placed inside a ligand, which could count as a flip without the boundaries swapping their order. These results quantify the improvements in accuracy that could be made by improving the first feed-forward network (the boundary bond classifier) in the model.

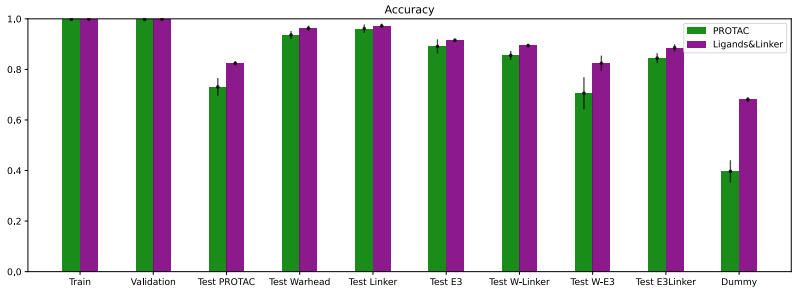
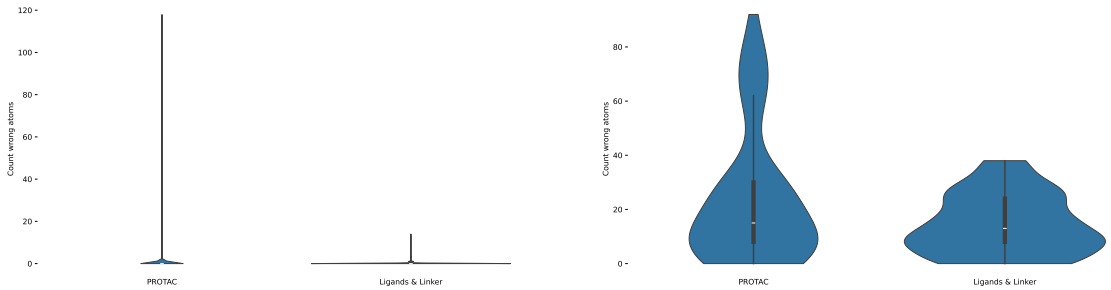


Figure 4.19: PROTAC accuracy relative to Ligands-linker accuracy

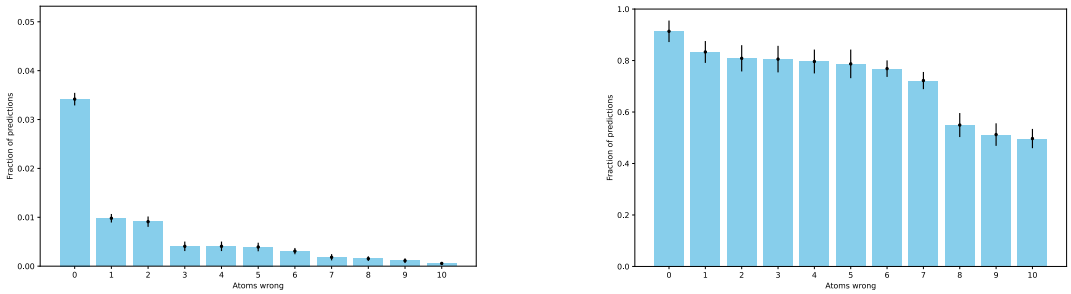
A violin plot showing the distribution of the number of mispredicted nodes is presented for the PROTAC and the ligands linker in Figure 4.20. A small number of PROTACs in the validation set were flipped, evident by the large spike, and the lack of this spike in the corresponding violin plot for the ligands linker. Overall, most predictions have very few mispredictions for the validation set. The Test PROTAC set was prone to flips, as evidenced by the dual bulge in the violin plot for the PROTACs, and the lack of it in the ligands linker violin plot. Most of these PROTACs had an error below 20 nodes, but a significant proportion had a larger error than this in both violin plots.



(a) Validation PROTACs **(b)** Test PROTAC

Figure 4.20: Distribution of mislabeled predictions for the three substructures and between the ligands and linker.

Using the number of mispredictions for the ligands and linker, the fraction of predictions with a greater number of mispredictions for a given number was calculated and is presented in Figure 4.21. For instance, Figure 4.21a shows a column at 2 atoms wrong of an approximate size of 0.01, indicating that 1% of all predictions had more than 2 atoms mispredicted. In other words, 99% of predictions had at most 2 atoms mispredicted for the validation set. This implies that there are very few mispredictions between the ligands and linker, indicating that the location of the boundaries is very precise for the validation set. The boundaries are less precise for the test set, as seen in Figure 4.21b. There is a notable drop in the figure at 8 atoms wrong, indicating a structure that is 8 atoms large tends to be mispredicted. However, it is not possible to identify which structure from these plots, and further investigation is required.



(a) Validation PROTACs **(b)** Test PROTAC

Figure 4.21: Fraction of predictions with more mislabeled atoms than a given number of mislabeled atom. Proportional to the integral of the violin plot of ligands & linker from the given number of atoms + 1.

An appropriate reference for what would signify an accurate prediction of the boundaries is a maximum of 3 atoms mislabeled on each side of the linker.[2] As the data is aggregated, it cannot be seen if the error is just on one side or both sides, but an absolute maximum error that satisfies this condition is a maximum allowed error of 6 atoms for the ligands and linker classes. The fraction of predictions that have at most 6 atoms mislabeled for the ligands and linker is presented in Table 4.13. Nearly all training and validation PROTACs satisfy this criterion, whereas 23% of the test PROTAC predictions do. As shown in the table, the more test substructures a PROTAC contains, the lower the fraction of predictions that satisfy the criteria.

Table 4.13: Fraction of predictions with 6 or fewer mispredicted ligands-linker atoms for the best model.

Dataset	Fraction
Training	99.8 ± 0.0
Validation	99.7 ± 0.0
Test PROTAC	23.1 ± 3.2
Test Warhead	86.7 ± 4.1
Test Linker	80.7 ± 8.4
Test E3	60.9 ± 2.8
Test Warhead-Linker	41.0 ± 3.0
Test Warhead-E3	40.0 ± 7.6
Test Linker-E3	42.5 ± 3.8
Dummy	12.9 ± 1.5

[2]Expert knowledge from my supervisors.

4.6 Discussion

General Discussion

Standard chemoinformatic methods can identify substructures if they are known. If two substructures of a PROTAC are known, the remaining substructure can be deduced; however, if two substructures are unknown, there is no definite method for determining these. The PROTAC Splitter competes against this baseline. Similar to standard chemoinformatic methods, the PROTAC Splitter reliably identifies substructures with near-perfect accuracy. It performs quite well when one substructure is unknown, but accuracy diminishes when two substructures are unknown. With three fully unknown substructures, it struggles but still manages to predict about 23% of PROTACs effectively.

Although the PROTAC Splitter has room for improvement, it remains valuable as an easy-to-use tool, even when the problem could be addressed with custom-made substructure matching code. It can be quite complex when a substructure fits multiple potential locations, and a straightforward tool for splitting PROTACs is still highly desirable.

Discussion of Data

Statistical methods and the calculation of p-values were not performed, which is a limitation of this work. However, a simple statistical measure such as standard deviation was calculated to gain an intuition of how certain a specific metric is. The rule-of-thumb that non-overlapping ranges of ± 1 standard deviation were used to quickly make judgments about what was likely better or worse, but it is recognized that this is not a true measure of statistical significance and should not be interpreted as such.

Node classes are inherently imbalanced due to substructures having slightly different sizes. While not a huge imbalance, the difference between boundary bonds and non-boundary bonds is moderate. It is unknown if this imbalance affects performance, and a potential test would involve training the model only on a randomly sampled set of non-boundary bonds for each PROTAC. The boundary bond predictors may be biased towards scoring all bonds as more non-boundary-like; however, if this bias is uniform, it is irrelevant as the inference of boundary bonds is relative to all bonds.

The standard deviations presented in all tables reflect the best epoch for each model, but given the random fluctuation across epochs and the smoothing of shaded areas in training curves, a more representative measure of model variation might involve averaging the best and preceding epochs. Nevertheless, the training curves were considered during the choice of training size, Butina cutoff, descriptors and the best model.

Discussion of Results

Most models start with a validation accuracy above 90% at the first epoch, which increases with training on larger datasets, as observed in the experiments. This is likely because the model is trained on more PROTACs in larger datasets before being evaluated, thus achieving higher validation accuracy at the first epoch compared to smaller datasets. This suggests that evaluating the model with smaller datasets, or after each iteration within the epoch, would yield a lower accuracy for the first epoch or iteration and would show a clear increase in accuracy. When the models are evaluated before training upon any epoch, the accuracy of all datasets clusters around the dummy accuracy; this was omitted from all training curves. Furthermore,

the high accuracy at the first epoch may explain why the loss does not decrease significantly over the course of training, as it could already be close to a local minimum.

For most training curves, training accuracy lags behind, reasonably so since it is evaluated *during* the training, whereas validation accuracy is assessed *after* the completion of an epoch. When adding the training set as another 'validation set', its accuracy was very similar to that of the validation set.

The trend of increasing accuracy with larger model sizes may be partially due to the fact that these models trained on a larger number of PROTACs before reaching the stopping criteria. The model with the smallest training size trained on at least approximately 13,000 PROTACs (*14x952*), the model with the medium training size on at least approximately 46,000 PROTACs (*16x2856*), and the model with the largest training size on at least approximately 95,000 PROTACs (*10x9520*). The number of PROTACs each model trained on was determined by the stopping criteria, which is set to stop when the lowest values of the most recent losses begin to trend upwards. With more training data, the randomness between epochs may be smoothed out, preventing the stopping criteria from being triggered prematurely before the true loss-minimum is reached. However, this is just a hypothesis; nonetheless, choosing the larger dataset size led to better accuracies.

In the experiments varying the Butina cutoff, it is noteworthy that all tests which included an E3 ligand exhibited greater variation than other datasets. The Butina clusters are created for substructures, and these clusters are sampled to create the reassembled PROTACs. However, each test split contains only two or three E3 ligands. No matter how they are clustered, the lowest frequency that any E3 ligand should appear in the final set of reassembled PROTACs is 25%, instead of the 'worst-case' 33% clustering for 3 E3 ligands, where two are grouped and then equally weighted against the other E3 ligand. Therefore, a slightly altered sampling frequency should not cause such a drastic shift. This suggests that the cause is related to the training set, possibly because some informative E3 ligand is clustered in a large cluster and thus represents a smaller fraction of the sampled E3 ligands.

The training time of the models did not change notably from including the graph descriptors. However, the speed at which the descriptors were calculated for the set of PROTACs varied. Using only chemical descriptors, 3690 PROTACs per minute can be calculated. Including Betweenness and Closeness centralities reduced the speed to 2411 PROTACs per minute (1.5 times slower). Adding Betweenness, Closeness, and Eigenvector centralities further slowed the rate to 1037 PROTACs per minute (3.6 times slower), and incorporating Betweenness, Closeness, and Local Eigenvector centralities further reduced the speed to 93 PROTACs per minute (40 times slower). It is worth considering the trade-off between the speed of calculating descriptors and the performance of the model.

Cross-validation was not performed during hyperparameter optimization, so the chosen best hyperparameters are likely partly selected by chance, as different model initializations and training-validation splits can affect accuracy. However, it was observed that the standard deviation was low (around 0.1 - 0.5) for both training and validation accuracy of PROTACs when the model was given different initial weights. The effect of different training-validation splits on the standard deviation is unknown; however, since the distribution of substructures is relatively uniform and the split between the training and validation sets of PROTACs was done randomly, it is reasonable to assume that the variation would be low and not significantly affect the final best hyperparameters. It may be that hyperparameters which had little influence on the validation accuracy would pose a greater risk of having a larger deviation from the 'real'

best hyperparameters, but since these had little influence on accuracy, it is less critical that they are as close to the 'real' best hyperparameters as the more influential ones. In the end, this was a trade-off between time and accuracy, yet effective hyperparameters were still identified, as demonstrated by the results.

The boundary node predictor with the local eigenvector had a noticeably lower validation accuracy at 96.2%, compared to the other models, which may indicate that Optuna did not find the optimal hyperparameters for it. Training the boundary node predictor was much slower, and it took Optuna more trials to find a satisfactory set of hyperparameters. It was necessary to increase the number of epochs from 10 to 15 per trial, as it was not approaching its potential after only 10 epochs during optimization. In contrast, Optuna quickly identified effective hyperparameters for the node predictor and boundary bond predictor v2, not requiring nearly all 200 trials to achieve a set of hyperparameters similar to those identified at the end of the 200 trials.

In the end, the boundary bond predictor version 2 was argued to be the best model. It was optimized using Betweenness and Closeness as descriptors, but it is unknown whether it would perform better with or without these, as the evaluation was performed using the node predictor. Regardless, calculating these graph centralities does not entail a significant extra computational cost. Ideally, all hyperparameters, training size, graph descriptors, and Butina cutoff would be optimized simultaneously.

The training curves for the models with optimized hyperparameters showed little overfitting, as the training loss did not drop significantly below the validation loss, except for the boundary node predictor with local eigenvectors.

Why the inclusion of the local eigenvector resulted in a larger variation appears to be due to one test split which had significantly worse performance metrics than the other test splits. However, the reason for this poorer performance in the test split remains unknown. This was validated as not being due to random fluctuations by retraining the same model on this split twice; the first few epochs of both runs displayed similarly poor accuracy as the full run of this split. Figure C.17c shows approximately the same average rates of flips with and without the Local eigenvector, yet the standard deviation is high, implying that this split has a higher rate of flips compared to the other two with better accuracies. Furthermore, the poorer accuracy of this split is also attributed to worse predicted locations of the boundary bonds, using the same reasoning as before but applied to Figure C.19c. The reason why the Local eigenvector would cause this effect on just one split remains unknown. A hypothesis is that it could be due to the training data becoming more divergent from the test data, and the model training on a data source that exhibits a different pattern from the test data, reducing its ability to extrapolate to the test sets.

The characteristic mistakes of each model stem from how they make their predictions, and the nature of these mistakes is independent of the data. However, the frequency of various mistakes each model makes depends on the substructures and the datasets used, as clearly observed in the fraction of flipped predictions for each dataset by the boundary bond predictor. Softer mistakes, such as shifting the predicted boundaries, were also shown to depend on the dataset, as suggested by the precision and recall of the substructures.

4.6.1 Future work

A clear point of improvement is to optimize the amount of training data, the Butina cutoff, and the choice of graph descriptors simultaneously with the hyperparameters. However, this was not done as it would require a complex integration of the data processing step with the optimization algorithm. Moreover, it would likely be very slow, as a significant proportion of time would be required to compute the data for each new trial.

Another point of improvement concerns computation speed. The model is surprisingly slow, most likely due to sub-optimal coding practices. Also, it was shown that computing precision and recall, in addition to accuracy, makes the model about four times slower. This is likely because these computations are performed on each individual PROTAC rather than on all PROTACs as an aggregate. However, this difference is intentional; it is preferred to take the macro average (average over all computed precision and recall) over the micro average (calculate the precision and recall from the aggregate of all PROTAC predictions), as the micro average does not consider the size of the substructure in the final metric, whereas the macro average does. The slower speed may also be attributed to the batch size being set to 1, as larger batches allow more data to be processed more efficiently by the GPU, and there are fewer (slow) data transfers between the GPU and CPU with larger batches.

A fundamental issue with these models is that their "field of view" before making a prediction on a specific node or bond is equal to the number of GNN layers the model has. For example, if a model has 9 layers, it can only make predictions using information from nodes up to 9 nodes away. This limitation stems from their reliance on the message passing algorithm. However, since a PROTAC is significantly longer than 9 atoms, predictions on one side of the PROTAC may be completely independent from those on the other side. This is undesirable, as knowing that one ligand is an E3 ligand definitively indicates that the other ligand is a warhead, and *vice versa*. This challenge might be overcome by using graph pooling algorithms, which could reduce the graph's diameter. However, most graph pooling algorithms tried in this thesis (unpresented work), specifically TopK pooling, sample a set of nodes but break the connections between sampled nodes if an unsampled node lies between them. Preliminary tests indicated that these pooling methods did not improve accuracy; however, more rigorous testing would be required to draw definitive conclusions. A promising algorithm is LaPool (Laplacian Pooling) [56], designed to capture molecular substructures in its pooled representation and preserve the original connectivity between node clusters.

Contrastive learning could be performed to first transform the representation of the nodes into one that distinguishes each substructure. This new representation could be used as the input for the PROTAC Splitter, which could simplify the task by embedding the information that all nodes should belong to three separate clusters directly into the graph. Ideally, this would improve the overall performance of all models.

The node prediction suffers from low validity due to chemically nonsensical mispredictions within the PROTAC, such as a substructure consisting of unconnected atoms. The node predictor could be improved by incorporating chemical validity into the loss function or using simpler metrics corresponding to self-consistent substructures without mixing substructure classes. It would also benefit from post-processing the predictions to fix simple mistakes, such as a single node being mispredicted within a substructure. This was performed in the early stages of the project, but it was quickly realized that this post-processing could not fix the worst mistakes. As the node predictor model developed, this code became outdated and is not part of the final model.

The boundary node predictor would likely benefit from being limited to only "separable nodes," as the difference between a perfect split and an unfixable split could be as simple as moving the boundary node by one step. This may be difficult for the model to differentiate, as the node representation may become over-smoothed with many layers, causing neighboring nodes to appear similar. Limiting the boundary node predictor to separable nodes could allow it to achieve 100% validity and rival the current best model, as they already share similar performance metrics.

There are many possible methods for improving the boundary bond predictor, which are outlined below. These methods may also be fully or partially applicable to the other models as well.

An uncertainty measure of a prediction may be able to identify flips, and if it can do so reliably, it could be used as a condition to un-flip the prediction. This was attempted but discontinued due to the complexity of the task—it is non-trivial to define uncertainty. The probability of each individual class for a bond are predicted in isolation and the main issue is determining the probability for the pair of boundary bonds relative to any other possible pair of bonds. This should be done with respect to the bond classes and locations simultaneously. Further analysis could be done on the probability distributions of true boundary bonds, flipped-bonds, and non-boundary bonds, and if they differ significantly, a criteria could be defined which would distinguish between these outcomes, as to identify and correct for flips. Also, the analysis could investigate any eventual changes in the probability distribution for a non-boundary bond based on the distance from the true boundary bond.

To prevent the PROTAC Splitter from predicting the boundary bond on both sides, the prediction could be done in two steps. This would give the second prediction the context of the first and reduce the risk of this mistake. The model would have to be trained on the full PROTAC and then successively on the Warhead-linker and E3-linker structures, which contain only one boundary bond. This could be achieved by either training the same model on both the full PROTAC and the ligand-linker structure, or by training another model on this downstream task.

4.6.2 Applications of the PROTAC splitter

In its current state, the PROTAC splitter is not fit to reliably splitting fully unknown PROTACs in a prospective manner. However, projects which have defined their POI and E3 ligase, as well as their warhead and E3 ligand, would have use of the PROTAC splitter. When the linker is unknown, around 80% of the predictions have 6 or fewer atoms wrong between the ligands and the linker. There is a small risk of flips, but with a known warhead and E3 ligand, this is easily fixable.

The code developed in this thesis allows any researcher or company to annotate their set of substructures and generate their own set of reassembled PROTACs to specifically train the PROTAC Splitter on their data. This task is arduous but manageable and only needs to be done once. This would give the user a performance corresponding to the validation set. Furthermore, if all public substructures were annotated once, it would be possible to reliably split all public PROTACs and expand PROTACpedia with matching substructures. This would be a meaningful contribution to the field. However, the current dataset does not encompass all publicly available substructures, and hence it would give non-perfect results, which may be detrimental if a user would use these substructures without awareness of this.

Since the PROTAC Splitter uses SMILES of a PROTAC and predicts the SMILES of the substructures, it can be integrated into established pipelines. These could include automatic docking protocols to dock just the warhead of a PROTAC, as docking the full PROTAC is more difficult.

It would be possible to cluster PROTACs by their substructure similarity rather than their overall similarity, allowing for better distinguishing between PROTACs that share substructures and greater analysis of the nuances in substructures between PROTACs.

The PROTAC Splitter could also be used to identify the substructures of AI-generated PROTACs, predicting which parts of the PROTAC belong to which class. This could be utilized in reinforcement learning, where the properties of the substructures could be calculated and fed into a reward function to train the PROTAC generator to produce PROTACs with specific substructure properties. Another application could be virtual screening of PROTAC generator outputs, accepting only PROTACs with substructures that have desired properties.

Potentially, better QSAR models could be developed to predict properties such as bioavailability, solubility, and permeability, as the different substructures may have varying effects on these properties, whereas their aggregate properties less informative in the prediction of these properties. Furthermore, it would be possible to train machine learning models on substructures to predict DC50 and DCMax.

5

Conclusions

The work in this thesis has investigated the prediction of PROTAC ternary structure with AlphaFold and prediction of substructures with graph neural networks.

Neither AlphaFold2 nor AlphaFold-Multimer is able to predict PROTAC ternary complexes. While they successfully predict the structures of individual POIs and E3 ligases, they fail to reproduce the complex as a whole. Their inability to predict the complexes likely stems from a combination of factors: the limited number of ternary structures in the PDB, that neither AlphaFold model may have trained upon many structures of this type of complex, the crystal structures having artificial interfaces, and the fact that PROTAC is necessary for the complex to form *in vivo* suggests it is necessary to take it into account when predicting the complex *in silico*.

A novel tool, the PROTAC Splitter, was developed. It accurately predicts the substructures of any PROTAC with known substructures and can partially generalize to chemically distinct PROTACs with unknown substructures. This tool fills a gap in the current field of research, as no published tool capable of this exists. It is a foundational tool with broad applications and addresses the field-specific challenge of obtaining high-quality substructure data, which is both difficult and time-consuming. The PROTAC splitter will hopefully accelerate research in the field and the development of more accurate models that predict the degradation capacity of PROTACs.

As a potential for future work, it could be examined if AlphaFold3 has the potential to predict the ternary structure given its capability to predict protein-ligand structures. The PROTAC splitter could be improved by sequentially predicting the boundaries between the substructures, to give the secondly predicted boundary the context of the first. Hopefully, these strategies could result in accurate structural predictions, aid in the further development of PROTACs and consequently combat debilitating diseases.

Bibliography

[1] Miklós Békés, David R. Langley, and Craig M. Crews. "PROTAC targeted protein degraders: the past is prologue". In: *Nature Reviews Drug Discovery* 21.3 (Mar. 2022), pp. 181–200. ISSN: 1474-1784. DOI: 10 . 1038 / s41573 - 021 - 00371 - 6. URL: https://doi.org/10.1038/s41573-021-00371-6.

[2] Ke Li and Craig M Crews. "PROTACs: past, present and future". en. In: *Chem. Soc. Rev.* 51.12 (June 2022), pp. 5214–5236.

[3] Michael J Bond and Craig M Crews. "Proteolysis targeting chimeras (PROTACs) come of age: entering the third decade of targeted protein degradation". en. In: *RSC Chem. Biol.* 2.3 (June 2021), pp. 725–742.

[4] Barmak Mostofian et al. "Targeted Protein Degradation: Advances, Challenges, and Prospects for Computational Methods". en. In: *J. Chem. Inf. Model.* 63.17 (Sept. 2023), pp. 5408–5432.

[5] Fenglei Li et al. "DeepPROTACs is a deep learning-based targeted degradation predictor for PROTACs". In: *Nature Communications* 13.1 (2022), p. 7133.

[6] Charlotte Crowe et al. "Mechanism of degrader-targeted protein ubiquitinability". In: *bioRxiv* (2024), pp. 2024–02.

[7] Ning Zheng and Nitzan Shabek. "Ubiquitin ligases: structure, function, and regulation". In: *Annual review of biochemistry* 86 (2017), pp. 129–157.

[8] David Komander. "The emerging complexity of protein ubiquitination". In: *Biochemical society transactions* 37.5 (2009), pp. 937–953.

[9] Robert I Troup, Charlene Fallan, and Matthias GJ Baud. "Current strategies for the design of PROTAC linkers: a critical review". In: *Exploration of Targeted Anti-tumor Therapy* 1.5 (2020), p. 273.

[10] Tao Wu et al. "Targeted protein degradation as a powerful research tool in basic biology and drug target discovery". In: *Nature Structural & Molecular Biology* 27.7 (2020), pp. 605–614.

[11] Wei Li et al. "Genome-wide and functional annotation of human E3 ubiquitin ligases identifies MULAN, a mitochondrial E3 that regulates the organelle's dynamics and signaling". In: *PloS one* 3.1 (2008), e1487.

[12] Mingming Liu et al. "Macrophage K63-linked ubiquitination of YAP promotes its nuclear localization and exacerbates atherosclerosis". In: *Cell reports* 32.5 (2020).

[13] Sachin Surade and Tom L Blundell. "Structural biology and drug discovery of difficult targets: the limits of ligandability". In: *Chemistry & biology* 19.1 (2012), pp. 42–50.

[14] John Hines et al. "Posttranslational protein knockdown coupled to receptor tyrosine kinase activation with phosphoPROTACs". In: *Proceedings of the National Academy of Sciences* 110.22 (2013), pp. 8942–8947.

[15] Longchuan Bai et al. "A potent and selective small-molecule degrader of STAT3 achieves complete tumor regression in vivo". In: *Cancer cell* 36.5 (2019), pp. 498–511.

[16] Weiguo Xiang et al. “Discovery of ARD-2585 as an exceptionally potent and orally active PROTAC degrader of androgen receptor for the treatment of advanced prostate cancer”. In: *Journal of medicinal chemistry* 64.18 (2021), pp. 13487–13509.

[17] Michael Zengerle, Kwok-Ho Chan, and Alessio Ciulli. “Selective small molecule induced degradation of the BET bromodomain protein BRD4”. In: *ACS chemical biology* 10.8 (2015), pp. 1770–1777.

[18] Nicolas Guedeney et al. “PROTAC technology: A new drug design for chemical biology with many challenges in drug discovery”. In: *Drug Discovery Today* 28.1 (2023), p. 103395.

[19] Christopher A Lipinski et al. “Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings”. In: *Advanced drug delivery reviews* 23.1-3 (1997), pp. 3–25.

[20] Daniel P Bondeson et al. “Lessons in PROTAC design from selective degradation with a promiscuous warhead”. In: *Cell chemical biology* 25.1 (2018), pp. 78–87.

[21] Xiao Wang et al. “Annual review of PROTAC degraders as anticancer agents in 2022”. In: *European Journal of Medicinal Chemistry* (2024), p. 116166.

[22] Ting-Ting Chu et al. “Specific knockdown of endogenous tau protein by peptide-directed ubiquitin-proteasome degradation”. In: *Cell chemical biology* 23.4 (2016), pp. 453–461.

[23] M Catarina Silva et al. “Targeted degradation of aberrant tau in frontotemporal dementia patient-derived neuronal cell models”. In: *elife* 8 (2019), e45457.

[24] Isabelle Wentzel. *Tragedy to Transformation – Molecular Glue Sparks the Field of Targeted Protein Degradation.* `https://lifesensors.com/tragedy-to-transformation-molecular-glue-sparks-the-field-of-targeted-protein-degradation/` [Accessed: 18-05-2024]. 2023.

[25] Gaoqi Weng et al. “PROTAC-DB 2.0: an updated database of PROTACs”. en. In: *Nucleic Acids Res.* 51.D1 (Jan. 2023), pp. D1367–D1372.

[26] Prilusky. *PROTACpedia.* `https://protacpedia.weizmann.ac.il/ptcb/main`. Accessed: 2023-9-17. 2016.

[27] Alexander McPherson and Jose A Gavira. “Introduction to protein crystallization”. In: *Acta Crystallographica Section F: Structural Biology Communications* 70.1 (2014), pp. 2–20.

[28] Helen M Berman et al. “The protein data bank”. In: *Nucleic acids research* 28.1 (2000), pp. 235–242.

[29] J. Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596 (2021), pp. 583–589. DOI: `10.1038/s41586-021-03819-2`. URL: `https://doi.org/10.1038/s41586-021-03819-2`.

[30] Adam Hospital et al. “Molecular dynamics simulations: advances and applications”. In: *Advances and Applications in Bioinformatics and Chemistry* (2015), pp. 37–47.

[31] Rommie E Amaro et al. “Ensemble docking in drug discovery”. In: *Biophysical journal* 114.10 (2018), pp. 2271–2278.

[32] Ryan P Wurz et al. “Affinity and cooperativity modulate ternary complex formation to drive targeted protein degradation”. In: *Nature Communications* 14.1 (2023), p. 4177.

[33] Junzhuo Liao et al. “In silico modeling and scoring of PROTAC-mediated ternary complex poses”. In: *Journal of Medicinal Chemistry* 65.8 (2022), pp. 6116–6132.

[34] Wenqing Li et al. “Importance of three-body problems and protein–protein interactions in proteolysis-targeting chimera modeling: insights from molecular dynamics simulations”. In: *Journal of Chemical Information and Modeling* 62.3 (2022), pp. 523–532.

[35] Jiyu Fan, Ailing Fu, and Le Zhang. “Progress in molecular docking”. In: *Quantitative Biology* 7 (2019), pp. 83–89.

[36] Arkadiusz Z Dudek, Tomasz Arodz, and Jorge Gálvez. "Computational methods in developing quantitative structure-activity relationships (QSAR): a review". In: *Combinatorial chemistry & high throughput screening* 9.3 (2006), pp. 213–228.

[37] Richard Evans et al. "Protein complex prediction with AlphaFold-Multimer". In: *biorxiv* (2021), pp. 2021–10.

[38] Josh Abramson et al. "Accurate structure prediction of biomolecular interactions with AlphaFold 3". In: *Nature* (2024), pp. 1–3.

[39] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of chemical information and computer sciences* 28.1 (1988), pp. 31–36.

[40] Laurianne David et al. "Molecular representations in AI-driven drug discovery: a review and practical guide". In: *Journal of Cheminformatics* 12.1 (2020), pp. 1–22.

[41] Stephen P Borgatti and Martin G Everett. "A graph-theoretic perspective on centrality". In: *Social networks* 28.4 (2006), pp. 466–484.

[42] Adrià Cereto-Massagué et al. "Molecular fingerprint similarity search in virtual screening". In: *Methods* 71 (2015), pp. 58–63.

[43] *RDKit Documentation*. `https://www.rdkit.org/docs`. Accessed: 2024-04-18.

[44] Stefano Ribes. "Machine Learning for Predicting Targeted Protein Degradation". In: (2023).

[45] Guy W Bemis and Mark A Murcko. "The properties of known drugs. 1. Molecular frameworks". In: *Journal of medicinal chemistry* 39.15 (1996), pp. 2887–2893.

[46] Solveig Badillo et al. "An introduction to machine learning". In: *Clinical pharmacology & therapeutics* 107.4 (2020), pp. 871–885.

[47] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. "Introduction to multi-layer feed-forward neural networks". In: *Chemometrics and intelligent laboratory systems* 39.1 (1997), pp. 43–62.

[48] Jie Zhou et al. "Graph neural networks: A review of methods and applications". In: *AI open* 1 (2020), pp. 57–81.

[49] Gilberto P Pereira et al. "AlphaFold-Multimer struggles in predicting PROTAC-mediated protein-protein interfaces". In: *bioRxiv* (2024), pp. 2024–03.

[50] Shuangjia Zheng et al. "Accelerated rational PROTAC design via deep learning and molecular simulations". In: *Nature Machine Intelligence* 4.9 (2022), pp. 739–748.

[51] Divya Nori, Connor W Coley, and Rocío Mercado. "De novo PROTAC design using graph-based deep generative models". In: *arXiv preprint arXiv:2211.02660* (2022).

[52] *PROTAC-DB*. Accessed: 2023-9-17. URL: `http://cadd.zju.edu.cn/protacdb/about`.

[53] Gábor Erdős, Mátyás Pajkos, and Zsuzsanna Dosztányi. "IUPred3: prediction of protein disorder enhanced with unambiguous experimental annotation and visualization of evolutionary conservation". In: *Nucleic acids research* 49.W1 (2021), W297–W303.

[54] Takuya Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2623–2631.

[55] Matthias Fey and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric". In: *arXiv preprint arXiv:1903.02428* (2019).

[56] Emmanuel Noutahi et al. "Towards interpretable sparse graph representation learning with laplacian pooling". In: *arXiv preprint arXiv:1905.11577* (2019).

[57] Darko Butina. "Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets". In: *Journal of Chemical Information and Computer Sciences* 39.4 (1999), pp. 747–750.

[58] Chenwei Wang et al. "GPS-Uber: a hybrid-learning framework for prediction of general and E3-specific lysine ubiquitination sites". In: *Briefings in Bioinformatics* 23.2 (2022), bbab574.

[59] Arslan Siraj et al. “UbiComb: a hybrid deep learning model for predicting plant-specific protein ubiquitylation sites”. In: *Genes* 12.5 (2021), p. 717.

[60] Weimin Li et al. “Multi-dimensional feature recognition model based on capsule network for ubiquitination site prediction”. In: *PeerJ* 10 (2022), e14427.

[61] Xiaowen Cui et al. “UbiSitePred: A novel method for improving the accuracy of ubiquitination sites prediction by using LASSO to select the optimal Chou’s pseudo components”. In: *Chemometrics and Intelligent Laboratory Systems* 184 (2019), pp. 28–43.

[62] Yushuang Liu et al. “Prediction of protein ubiquitination sites via multi-view features based on eXtreme gradient boosting classifier”. In: *Journal of Molecular Graphics and Modelling* 107 (2021), p. 107962.

[63] Hongli Fu et al. “DeepUbi: a deep learning framework for prediction of ubiquitination sites in proteins”. In: *BMC bioinformatics* 20.1 (2019), pp. 1–10.

[64] Sikandar Rahu et al. “UBI-XGB: Identification of ubiquitin proteins using machine learning model”. In: *Journal of Mountain Area Research* 8 (2022), pp. 14–26.

[65] Jie Chen et al. “Capsulated Graph Neural Network for Ubiquitylation Sites Prediction”. In: *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. 2022, pp. 3797–3799.

[66] Shazia Murad et al. “UbiSites-SRF: ubiquitination sites prediction using statistical moment with random forest approach”. In: (2021).

Appendix

A

Machine Learning Concepts

This appendix provides brief explanations of some elementary concepts in machine learning essential for understanding the methodologies applied in the main content of this document.

Regression models use any input to predict a target variable, which is a continuous value, whereas **classification** models predict discrete values or classes. A classification model can have an identical architecture to a regression model, with the primary difference being the output layer; a function that discretizes the output is used to define the classes. A core difference is also that regression and classification models are trained with different types of loss functions.

The **Loss function** is a mathematical function used in machine learning to quantify the difference between the predicted values by a model and the actual values in the dataset. It serves as a guide for the optimization process, with the objective of minimizing this function during training to improve the model's accuracy. Different types of loss functions are applicable depending on the nature of the problem, such as mean squared error for regression tasks or cross-entropy loss for classification tasks.

Hyperparameter optimization is the process of finding a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is set before the learning process begins.

Learning rate is a hyperparameter that determines the step size at each iteration while moving toward a minimum of a loss function. Choosing the right learning rate is crucial as it affects how quickly a model converges to a local minimum.

Batch size refers to the number of training samples used to train a single iteration. The size of the batch can significantly impact the stability and speed of the training process.

An **Epoch** is when all training PROTACs have been trained upon once. A model is typically trained over multiple epochs and evaluated after each epoch. An epoch is split into a set of iterations, which consists of a batch of PROTACs, and to train upon one epoch implies to train upon all iterations.

. The size of the batch can significantly impact the stability and speed of the training process.

Overfitting occurs when a model learns the detail and noise in the training data to an extent that it negatively impacts the performance of the model on new data [47]. This usually happens when a model is too complex, with too many parameters relative to the number of observations.

Regularization helps prevent overfitting, ensuring models generalize well to new data. **Dropout** is a regularization technique for deep neural networks that randomly deactivates a subset of neurons during training. This prevents neurons from developing strong dependencies on each other's activation patterns, which can degrade model performance on unseen data. By promoting the development of robust, independent features within the network, dropout enhances the model's overall reliability and effectiveness across various datasets.

Data imbalance refers to situations where some classes are significantly more frequent than others in the training dataset. This can lead to models that perform well on the majority class but poorly on less frequent classes, thus compromising overall model performance.

Data leakage occurs when data from the test set or validation set can be found in the training set. This can lead to overly optimistic performance estimates and poor generalization to new data.

Cross-validation is a statistical method used to estimate the generalizability of a machine learning model to new data. It involves dividing the dataset into multiple subsets and systematically using different subsets as the training and validation data.

B

AlphaFold

B.1 Largest Common Sequence

An example of how the largest common sequence (LCS) is generated from a set of sequences:

```
   ABCDEFGHIJK
     CDEFGHIJ
    BCDEGGHIJHHHHHH
     CDAFGH-JK
LCS: CDEFGHIJ
```

The LCS for this set of sequences would be CDEFGHIJ. The principle is to remove variations between sequences to get a sequence that corresponds as much as possible and as well as possible to all sequences it was made from. Some crystal structures contained artificial sequences such as 6xHis tags, and these were not removed if they were present in all PDB sequences in a complex group.

B.2 Other PDB IDs

PDB IDs of PROTAC ternary complexes which was not in the PROTAC-DB: 6NJS, 2EUF, 1HWK, 2ITO, 3U9Y, 5ML8, 4QL1, 6d55, 1ERR, 1Z95, 2OUZ, 3MXF, 3ONI, 4CI3, 4TZ4, 50P4, 5NW2.

PDB IDs of Molecular glue ternary complexes: 8G46, 8OV6, 5FQD, 8G66, 6H0F, 7U8F, 8D80, 8DEY, 7LPS, 8U15, 8U16, 8U17, 6UML, 6XK9, 5HXB.

PDB IDs of antibodies: 1hh9, 1hh6, 3k2u, 3eob, 3eob, 2r56, 2r56, 2xra, 3eo1, 3eo1, 3eo1, 3eo1, 3eyf, 3eyf, 3eys, 3eyu, 1za3, 1za3, 1jps, 1g6v, 3g5v, 1bvk, 1bvk, 2oqj, 2oqj, 2oqj, 2oqj, 1iqd, 4gxu, 4gxu, 4gxu, 4gxu, 4gxu, 4gxu, 1vfb, 3qsk, 2p46, 2p46, 2p44, 2p45, 2p42, 2p42, 2p48, 2p49, 2p4a, 2p4a, 3b2u, 3b2u, 3b2u, 3b2u, 3b2u, 3b2u, 3b2u, 3b2u, 3b2v, 3etb, 3etb, 3etb, 3etb, 1w72, 1w72, 3eoa, 3eoa, 3lqa, 2wub, 2wub, 2wuc, 3lh2, 3lh2, 3lh2, 3lh2, 3lev, 3bky, 3iu3, 3iu3, 3iu3, 3a6c, 1zv5, 2xtj, 1zvy, 3ogo, 3ogo, 3ogo, 3ogo, 3skj, 3skj, 2r0z, 2r0k, 2r0l, 3h3p, 3h3p, 1nsn, 1qfu, 3c09, 3c09, 1ri8, 1mlc, 1mlc, 2hrp, 2hrp, 1wej, 3p0y, 2uzi, 3hi1, 3hi1, 3hi6, 3hi6, 2b2x, 2b2x, 2qr0, 2qr0, 2qr0, 2qr0, 2qr0, 2qr0, 2qr0, 2qr0, 1pz5, 1dqj, 2jix, 2jix, 2jix, 3qwo, 3qwo, 3g5y, 3ifo, 3ifo, 3ifl, 3ifp, 3ifp, 3ifp, 3ifp, 1nca, 3pp4, 2nz9, 2nz9, 3ixt, 3ixt, 1p2c, 1p2c, 2eh8, 3nh7, 3nh7, 3nh7, 3nh7, 1yyl, 1yyl, 3e8u, 1yy9, 1nbz, 1nby, 1uwx, 1uwx, 2cmr, 1bj1, 1bj1, 2h9g, 2h9g, 3uc0, 3uc0, 3q1s, 1f90, 2jel, 2p43, 3l5x, 2xqb, 3h42, 3kr3, 2vwe, 2vwe, 1p4b, 1i8k, 1kxq, 1kxq, 1kxq, 1kxq, 2vis, 2vir, 3ggw, 3ggw, 1n64, 2p47, 3l5w, 3l5w, 1kxt, 1kxt,

1kxt, 1kxv, 1kxv, 2fx8, 2fx8, 2fx8, 2fx8, 2fx7, 2nxz, 1op9, 2a6i, 2fx9, 2fx9, 1sy6, 2nxy, 2zpk, 2zpk, 3p11, 2vxq, 2vxs, 2vxs, 2vxs, 2vxs, 2nyy, 3gjf, 3gjf, 2ny7, 2ny6, 2ny5, 2ny4, 2ny3, 2ny1, 2ny0, 3ngb, 3ngb, 3ngb, 3ngb, 3eba, 3t2n, 3t2n, 2vyr, 2vyr, 2vyr, 2vyr, 2vyr, 3q3g, 3q3g, 3q3g, 3q3g, 3ma9, 3sdy, 2dd8, 3mac, 3o6l, 3o6m, 1cz8, 1cz8, 3fku, 3fku, 3fku, 3fku, 3fku, 3fku, 3nfp, 3nfp, 3be1, 2or9, 2or9, 3l95, 3l95, 3hae, 3hae, 3hae, 3hae, 3ghe, 2w9e, 2aep, 2aeq, 2x89, 2x89, 2x89, 2hfg, 2bdn, 3bdy, 2j4w, 2vxt, 3dvg, 3dvn, 3dvn, 1zmy, 1hez, 1hez, 3a6b, 3g6j, 3g6j, 3a67, 3n85, 2ny2, 1ob1, 1ob1, 2fjh, 2fjh, 2fjg, 2fjg, 3r1g, 1nma, 3bgf, 3bgf, 2qhr, 3mj9, 1jrh, 2j5l, 3l5y, 2hkf, 1tzi, 1tzh, 1tzh, 3rkd, 3rkd, 1e6j, 3ru8, 3lhp, 3lhp, 1eo8, 3mxw, 1tet, 1dzb, 1dzb,

B.3 Proteins in the PROTAC Complexes

Table B.1: The table displays which adaptor proteins each PDB structure within each complex group have co-crystallized alongside the POI and E3 ligase.

Complex group	Uniprot ID of POI	Uniprot ID of E3	Adaptor protein(s)	UniProt ID for adaptor protein(s)
1	Q06187 BTK	Q13490 CIAP1		
2	O60885 BRD4 (BD1)	Q96SW2 CRBN	DNA damage-binding protein 1	Q16531
3	Q07817 BCL-xL	P40337 VHL	Elongin-B, Elongin-C	Q15370, Q15369
4	O60885 BRD4 (BD2)	P40337 VHL	Elongin-B, Elongin-C	Q15370, Q15369
5	O60885 BRD4 (BD1)	P40337 VHL	Elongin-B, Elongin-C	Q15370, Q15369
6	Q05397 FAK	P40337 VHL	Elongin-B, Elongin-C (Isoform 2)	Q15370, Q15369-2
7	P51531 SMARCA2	P40337 VHL	Elongin-B, Elongin-C	Q15370, Q15369
8	P51532 SMARCA4	P40337 VHL	Elongin-B, Elongin-C	Q15370, Q15369
9	P61964 WDR5	P40337 VHL	Elongin-B, Elongin-C	Q15370, Q15369

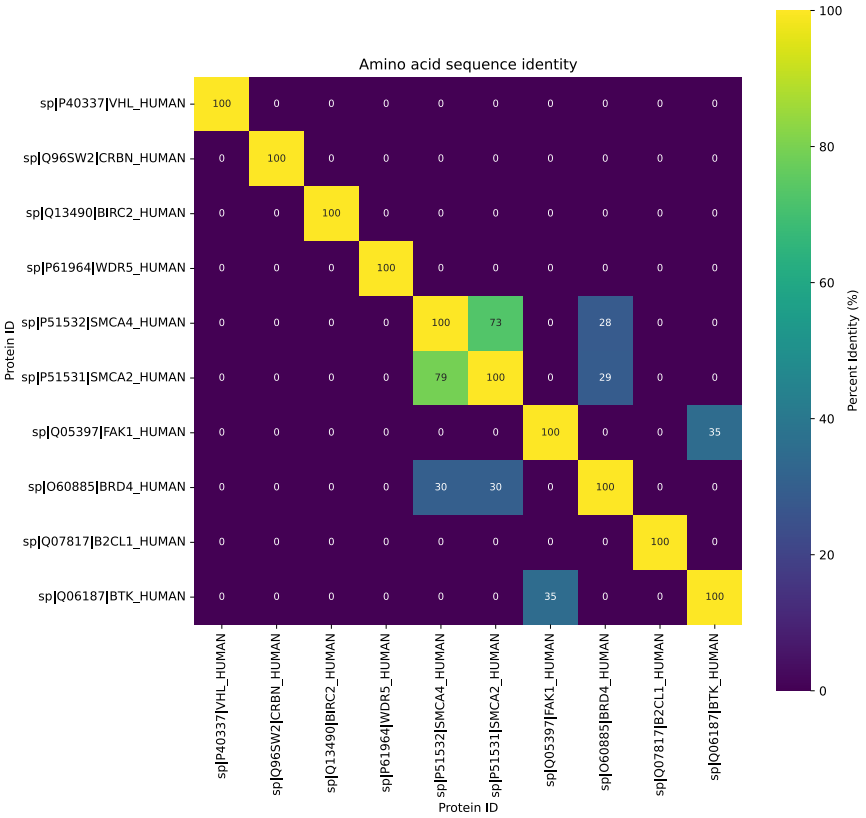


Figure B.1: Sequence identity from BLAST between all full-length sequences from Uniprot of the reference POI and E3 Ligase.

B.4 RMSD of Predicted Structures

Complex ID	POI RMSD (Ångström)	E3 RMSD (Ångström)
1	0.813, 0.809, 0.621, 0.612, 0.877 (BTK)	1.039, 0.923, 0.997, 0.975, 0.961 (CIAP1)
2	0.767, 0.749, 0.735, 0.733, 0. 756 (BRD4 (BD1))	0.888, 0.896, 0.909, 0.928, 0.877 (CRBN)
3	2.083, 1.963, 1.913, 1.957, 1.927 (BCL-xL)	0.704, 0.763, 0.672, 0.721, 0.710 (VHL)
4	1.086, 0.329, 0.342, 0.354, 0.339 (BRD4 (BD2))	0.752, 0.723, 0.780, 0.797, 0.827 (VHL)
5	0.654, 0.668, 0.441, 0.649, 0.460 (BRD4 (BD1))	0.679, 0.726, 0.751, 0.754, 0.706 (VHL)
6	3.382, 3.431, 3.500, 3.369, 3.441 (FAK)	0.657, 0.660, 0.647, 0.658, 0.645 (VHL)
7	1.275, 1.365 1.298, 1.312, 1.293 (SMARCA2)	0.705, 0.784, 0.804, 0.713, 0.764 (VHL)
8	0.484, 0.471, 0.498, 0.471, 0,491 (SMARCA4)	0.807, 0.733, 0.758, 0.726, 0.801 (VHL)
9	0.526, 0.520, 0.520, 0.624, 0.518 (WDR5)	0.738, 0.708, 0.709, 0.652, 0.703 (VHL)

Table B.2: Root mean square distance for the POI and E3 Ligase between the AlphaFold2(PDB largest common PDB sequence) and PDB structures.

Complex ID	POI RMSD [Å]	E3 RMSD [Å]
1	0.877, 0.860, 0.867, 0.872, 0.901 (BTK)	0.498, 0.495, 0.495, 0.481, 0.467 (CIAP1)
2	1.006, 1.013, 0.910, 0.891, 0.901 (BRD4 (BD1))	0.813, 0.861, 0.825, 0.803, 0.801 (CRBN)
3	1.846, 2.086, 2.210, 2.059, 1.934 (BCL-xL)	0.520, 0.480, 0.470, 0.475, 0.461 (VHL)
4	1.222, 1.121, 1.200, 1.189, 1.176 (BRD4 (BD2))	0.576, 0.587, 0.567, 0.732, 0.644 (VHL)
5	0.993, 0.820, 0.799, 1.081, 0.903 (BRD4 (BD1))	0.449, 0.438, 0.425, 0.646, 0.476 (VHL)
6	2.361, 2.426, 2.388, 2.355, 2.326 (FAK)	0.648, 0.697, 0.729, 0.847, 0.648 (VHL)
7	0.971, 0.805, 1.338, 1.317, 0.780 (SMARCA2)	0.830, 0.836, 0.835, 0.859, 0.850 (VHL)
8	0.965, 3.358, 0.645, 0.628, 0.940 (SMARCA4)	0.507, 0.484, 0.504, 0.503, 0.520 (VHL)
9	0.518, 0.516, 0.540, 0.549, 0.537 (WDR5)	1.396, 1.392, 1.407, 1.472, 1.385 (VHL)

Table B.3: Root mean square distance for the POI and E3 Ligase between the UniProt and PDB structures.

B.5 Compare 6BOY with Molecular Glues

Many of the molecular glue structures from the PDB had Cereblon as the E3 ligase. It was observed that the contact area of most retrieved molecular glues were similar to the interface are of the crystal structure 6BOY, as can be seen in Figure 3.12. The amino acid sequence length of each POI and the slightly different cereblon E3 ligase are presented in Figure B.2.

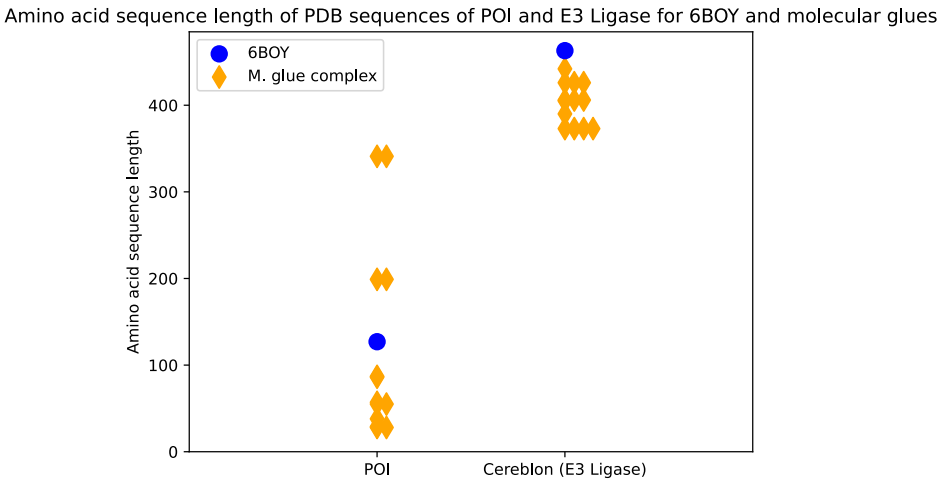
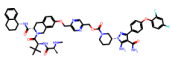
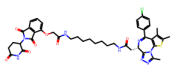


Figure B.2: Amino acids sequence length of POI and E3 Ligase (Cereblon) for 6BOY and molecular glues with same E3 Ligase.

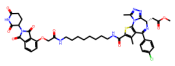
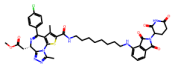
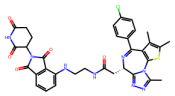
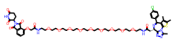
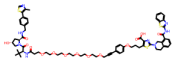
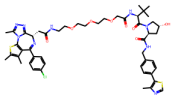
B.6 PROTACs in Crystallized Ternary Complexes

Table B.4: PROTACs in the crystal structures found in PROTAC-DB.

PDB ID	PROTAC	SMILES of PROTAC
6W8I		CN[C@@H](C)C(=O)N[C@H](C(=O)N1CC2=CC(OCCOCCOCCOCCOC COC(=O)N3CCC[C@@H](N4N=C(C5=CC=C(OC6=CC=C(F)C=C6F)C =C5)C(C(N)=O)=C4N)C3)=CC=C2C[C@H]1C(=O)N[C@@H]1CCCC2= CC=CC=C21)C(C)(C)C
6W7O		CN[C@@H](C)C(=O)N[C@H](C(=O)N1CC2=CC(OCC3=CN=C(COC(= O)N4CCC[C@@H](N5N=C(C6=CC=C(OC7=CC=C(F)C=C7F)C=C6)C(C(N)=O)=C5N)C4)C=N3)=CC=C2C[C@H]1C(=O)N[C@@H]1CCCC2=C C=CC=C21)C(C)(C)C
6BOY		CC1=C(C)C2=C(S1)N1C(C)=NN=C1[C@H](CC(=O)NCCCCCCCCNC(=O)COC1=CC=CC3=C1C(=O)N(C1CCC(=O)NC1=O)C3=O)N=C2C1 =CC=C(Cl)C=C1

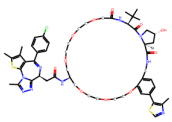
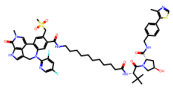
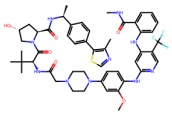
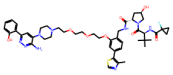
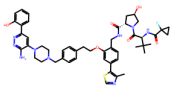
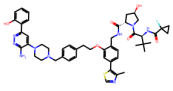
Continued on next page

Table B.4 – continued from previous page

PDB ID	Image	PROTAC (SMILES)
6BN7		COC(=O)C[C@@H]1N=C(C2=CC=C(Cl)C=C2)C2=C(SC(C(=O)NCCCCCCCCNC(=O)COC3=CC=CC4=C3C(=O)N(C3CCC(=O)NC3=O)C4=O)=C2C)N2C(C)=NN=C12
6BN9		COC(=O)C[C@@H]1N=C(C2=CC=C(Cl)C=C2)C2=C(SC(C(=O)NCCCCCCCCNC3=CC=CC4=C3C(=O)N(C3CCC(=O)NC3=O)C4=O)=C2C)N2C(C)=NN=C12
6BNB		CC1=C(C)C2=C(S1)N1C(C)=NN=C1[C@H](CC(=O)NCCNC1=CC=CC3=C1C(=O)N(C1CCC(=O)NC1=O)C3=O)N=C2C1=CC=C(Cl)C=C1
6BN8		CC1=C(C)C2=C(S1)N1C(C)=NN=C1[C@H](CC(=O)NCCOCCOCCOCCOCCOCCOCCOCCOCCOCCOCCOCCOCCOCCOCCOCCNC(=O)COC1=CC=CC3=C1C(=O)N(C1CCC(=O)NC1=O)C3=O)N=C2C1=CC=C(Cl)C=C1
6ZHC		CC1=C(C2=CC=C(CNC(=O)[C@@H]3C[C@@H](O)CN3C(=O)[C@@H](NC(=O)CCOCCOCCOCCOCCOCCOCCC#CC3=CC=C(OCCCC4=C(C(=O)O)N=C(N5CCC6=CC=CC(C(=O)NC7=NC8=CC=CC=C8S7)=C6C5)S4)C=C3)C(C)(C)C)C=C2)SC=N1
5T35		CC1=C(C2=CC=C(CNC(=O)[C@@H]3C[C@@H](O)CN3C(=O)[C@@H](NC(=O)COCCOCCOCCNC(=O)C[C@@H]3N=C(C4=CC=C(Cl)C=C4)C4=C(SC(C)=C4C)N4C(C)=NN=C34)C(C)(C)C)C=C2)SC=N1

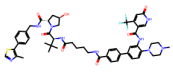
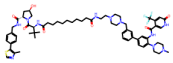
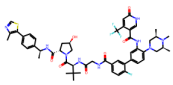
Continued on next page

Table B.4 – continued from previous page

PDB ID	Image	PROTAC (SMILES)
6SIS		CC1=C(C2=CC=C3CNC(=O)[C@@H]4C[C@@H](O)CN4C(=O)[C@H](C(C)(C)C)NC(=O)COCCOCCOC[C@H](NC(=O)C[C@@H]4N=C(C5=CC=C(Cl)C=C5)C5=C(SC(C)=C5C)N5C(C)=NN=C45)COCCOCCOC3=C2)SC=N1
7KHH		CC1=C(C2=CC=C(CNC(=O)[C@@H]3C[C@@H](O)CN3C(=O)[C@@H](NC(=O)CCCCCCCCCCNC(=O)C3=CC4=C(C=C3CS(C)(=O)=O)C3=CN(C)C(=O)C5=C3C(=C[NH]5)CN4C3=NC=C(F)C=C3F)C(C)(C)C)C=C2)SC=N1
7PI4		CNC(=O)C1=CC=CC=C1NC1=CC(NC2=CC=C(N3CCN(CC(=O)N[C@H](C(=O)N4C[C@H](O)C[C@H]4C(=O)N[C@@H](C)C4=CC=C(C5=C(C)N=CS5)C=C4)C(C)(C)C)CC3)C=C2OC)=NC=C1C(F)(F)F
6HAY		CC1=C(C2=CC=C(CNC(=O)[C@@H]3C[C@@H](O)CN3C(=O)[C@@H](NC(=O)C3(F)CC3)C(C)(C)C)C(OCCOCCOCCN3CCN(C4=CC(C5=CC=CC=C5O)=NN=C4N)CC3)=C2)SC=N1
6HAX		CC1=C(C2=CC=C(CNC(=O)[C@@H]3C[C@@H](O)CN3C(=O)[C@@H](NC(=O)C3(F)CC3)C(C)(C)C)C(OCCC3=CC=C(CN4CCN(C5=CC(C6=CC=CC=C6O)=NN=C5N)CC4)C=C3)=C2)SC=N1
6HR2		CC1=C(C2=CC=C(CNC(=O)[C@@H]3C[C@@H](O)CN3C(=O)[C@@H](NC(=O)C3(F)CC3)C(C)(C)C)C(OCCC3=CC=C(CN4CCN(C5=CC(C6=CC=CC=C6O)=NN=C5N)CC4)C=C3)=C2)SC=N1

Continued on next page

Table B.4 – continued from previous page

PDB ID	Image	PROTAC (SMILES)
7Q2J		CC1=C(C2=CC=C(CNC(=O)[C@@H]3C[C@@H](O)CN3C(=O)[C@@H](NC(=O)CCCCNC(=O)C3=CC=C(C4=CC=C(N5CCN(C)CC5)C(NC(=O)C5=C[NH]C(=O)C=C5C(F)(F)F)=C4)C=C3)C(C)(C)C)C=C2)SC=N1
7JTO		CC1=C(C2=CC=C(CNC(=O)[C@@H]3C[C@@H](O)CN3C(=O)[C@@H](NC(=O)CCCCCCCCCC(=O)NCCN3CCN(CC4=CC=CC(C5=CC=C(N6CCN(C)CC6)C(NC(=O)C6=C[NH]C(=O)C=C6C(F)(F)F)=C5)=C4)CC3)C(C)(C)C)C=C2)SC=N1
7JTP		CC1=C(C2=CC=C([C@H](C)NC(=O)[C@@H]3C[C@@H](O)CN3C(=O)[C@@H](NC(=O)CNC(=O)C3=CC=C(F)C(C4=CC=C(N5C[C@H](C)N(C)[C@H](C)C5)C(NC(=O)C5=C[NH]C(=O)C=C5C(F)(F)F)=C4)=C3)C(C)(C)C)C=C2)SC=N1

C

PROTAC splitter

C.1 Data Split

The resulting number of substructures (with attachment points), murcko scaffolds and frameworks, for the training an test sets are presented in table C.1. Note that some few substructures in the training and test set may share a MS or framework, one substructure in the training set of each substructure type have a shared murcko scaffold with the corresponding test set. Six frameworks are shared between the training and test warheads, ten frameworks are shared among the linkers, and one framework is shared among the E3 ligands. Note that the training substructures will be used for the validation set of PROTACs.

Table C.1: Number of public substructures and Murcko scaffolds (MS) of substructures in the training-validation and test sets, from HDBSCAN.

Dataset	Warheads	Linkers	E3 ligands
All SMILES	280	1033	60
Train-validation SMILES	239	952	53
Test SMILES	41	81	7
All MS	207	1015	32
Train/validation MS	173	937	26
Test MS	35	79	7
All frameworks	169	279	26
Train/validation frameworks	143	235	20
Test frameworks	32	54	7

The splitting of the set of substructures was done using the clusters HDBSCAN created. The exact algorithm is complex; instead, an illustrative example of how the clusters could look is presented in figure C.1. Each blue dot represents a molecule, the distances between the dots represent their dissimilarity, and the red circles represent the similarity cutoff. A simplified explanation is that all molecules that are within each others cutoff radii are grouped into the same cluster, and this is done itterativly for all molecules.

The number of bonds of each type, for all unique substructures from the curated dataset, is presented in Table C.2. This includes substructures from the training and test sets. Notably, most bond are single bonds.

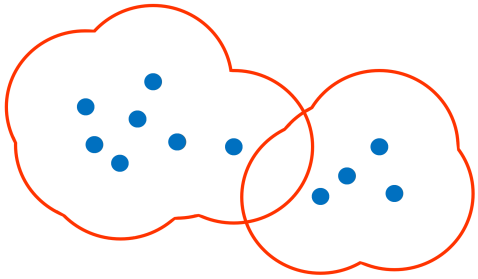


Figure C.1: Example of HDBSCAN clusters.

Table C.2: Sum of all bond orders for unique substructures in the curated dataset [44].

Substructure type	Single	Double	Triple
Warhead	275	1	4
Linker	2061	5	2
E3 Ligand	60	0	0

C.2 Butina Clustering Algorithm

Butina Clustering is a method for clustering molecules using Tanimoto similarity [57].

The clustering algorithm is as follows:

- Calculate the Tanimoto similarity between all pairs of molecules and define a cutoff value
- For each molecule, count the number of neighbouring molecules, defined by if the similarity is above the cutoff.
- Select the molecule with the most neighbours and define it and its neighbours as a cluster. Molecules which have been clustered are exempt from further clustering, as one molecule shall only belong to one cluster.
- Iterate through the set of molecules, in the order of having the most to the least neighbours, until all molecules belongs to a cluster.

An illustrative example of how the Butina clusters looks like is presented in figure C.2. It depicts a set of molecules, represented as blue dots, which have been grouped into three clusters. The numbers indicate the cluster ID, centered at the molecule which had the most neighbours before this cluster was defined.

C.3 Butina Clusters

Butina clustering with a cutoff at 0 is equivalent to each substructure being its own cluster, hence it is not plotted here. Note that the all splits of the test substructures were joined in these plots. In practice, Butina clustering was applied to each split separately.

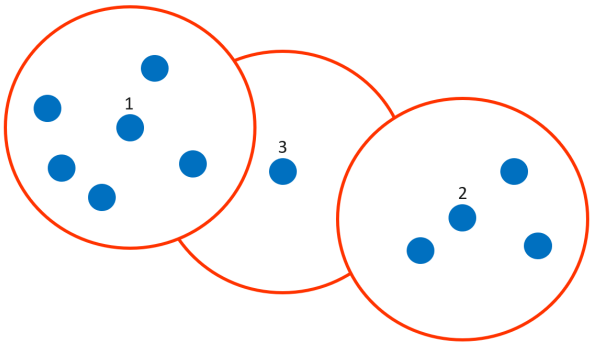
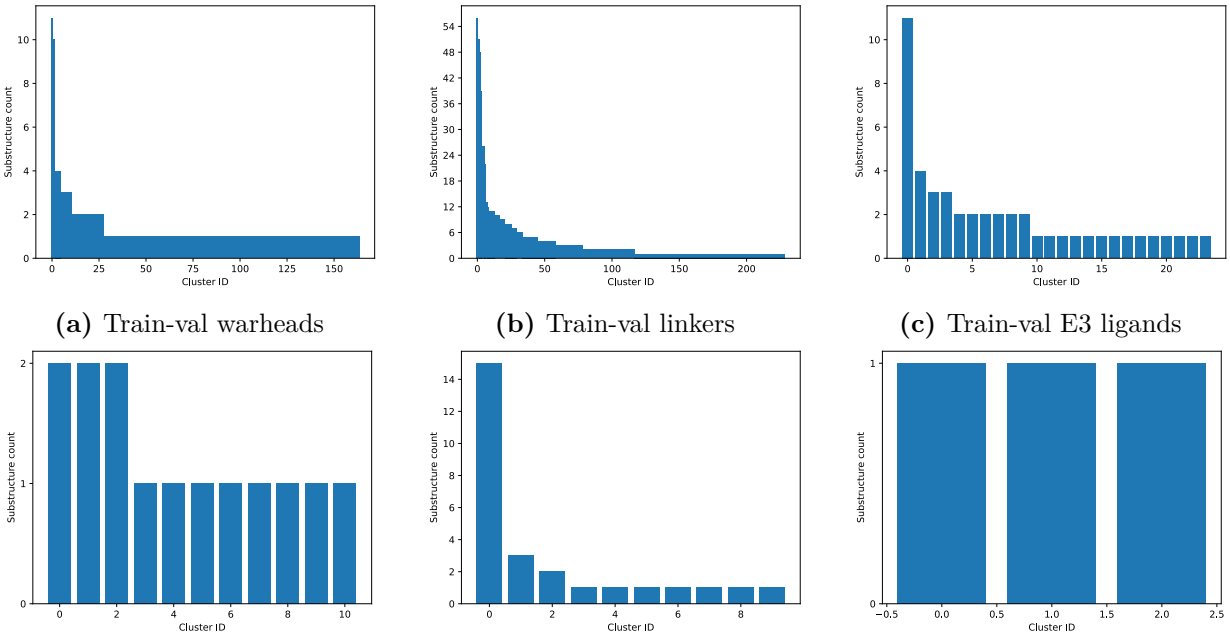


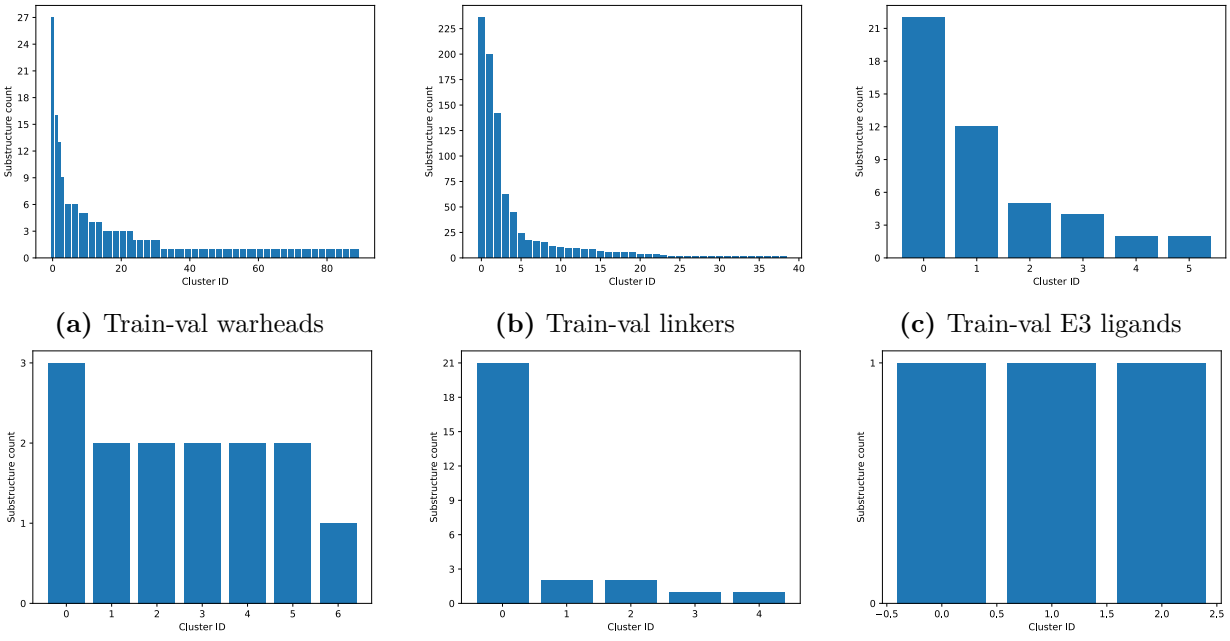
Figure C.2: Example of Butina clustering. Each blue dot represents a molecule, distances represent their dissimilarity, and the red circles represent the similarity cutoff.



(a) Train-val warheads **(b)** Train-val linkers **(c)** Train-val E3 ligands

(d) Test warheads **(e)** Test linkers **(f)** Test E3 ligands

Figure C.3: Butina clusters of substructures for the training set and the *first* test split, with a cutoff value at 0.33



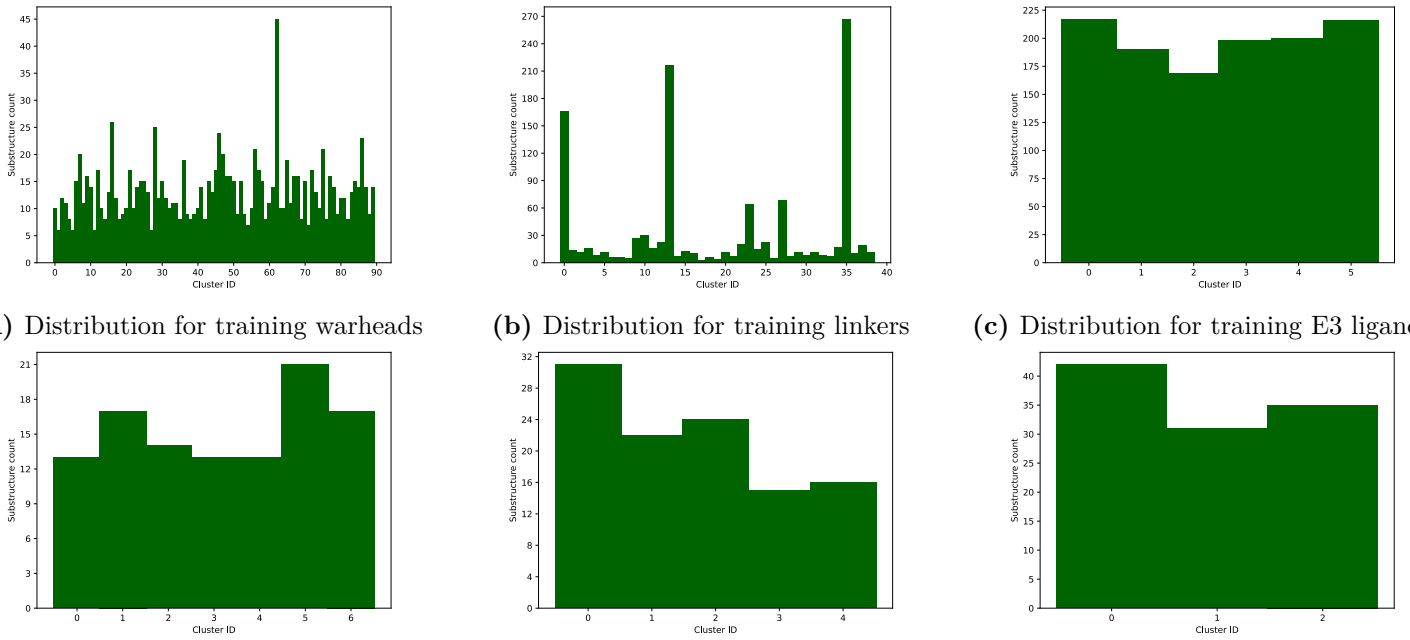
(a) Train-val warheads **(b)** Train-val linkers **(c)** Train-val E3 ligands

(d) Test warheads **(e)** Test linkers **(f)** Test E3 ligands

Figure C.4: Butina clusters of substructures for the training set and the *first* test split, with a cutoff value at 0.67

C.4 Substructure Distribution of Reassembled PROTACs

The sampling distribution over the Butina clusters is presented for the training and first test split of reassembled PROTACs are presented in figure C.5. The cluster IDs in C.5 does not match the cluster IDs in C.4. The distribution is uniform with exception to some outlying clusters, which are clusters with many substructures. These outliers are present due to the enforcement of all substructures must be sampled once, as to not discard any substructure.



(a) Distribution for training warheads **(b)** Distribution for training linkers **(c)** Distribution for training E3 ligands

(d) Distribution for test warheads **(e)** Distribution for test linkers **(f)** Distribution for test E3 ligands

Figure C.5: Cluster sampling distribution for each substructure within training set and the *first* test split of reassembled PROTACs. The Butina clusters were created with a cutoff at 0.67.

C.5 Definitions of Evaluation Metrics

Accuracy is defined as the fraction of the number of correct predictions to the total number of predictions:

$$\frac{\#\text{Correct predictions}}{\#\text{All predictions}} = \frac{\#\text{True positives} + \#\text{True negatives}}{\#\text{True positives} + \#\text{True negatives} + \#\text{False positives} + \#\text{False negatives}}$$

Precision is defined as the fraction of true positives (correctly identified substructure atoms) to all *predicted* positives (all predicted substructure atoms). Intuitively, this is how "pure" the predicted was:

$$\frac{\#\text{True positives}}{\#\text{Predicted positives}} = \frac{\#\text{True positives}}{\#\text{True positive} + \#\text{False positives}}$$

Recall is defined as the fraction of true positives (correctly identified substructure atoms) to *all* positives (all substructure atoms). Intuitively, this is how much of the substructure was "remembered":

$$\frac{\#\text{True positives}}{\#\text{All positives}} = \frac{\#\text{True positives}}{\#\text{True positives} + \#\text{False negatives}}$$

C.6 Calculation of the Local Eigenvector Centrality

The Local eigenvector centrality was calculated by iterating through all nodes in the graph and creating subgraphs with a radius R and centered at each node. The Eigenvector centrality

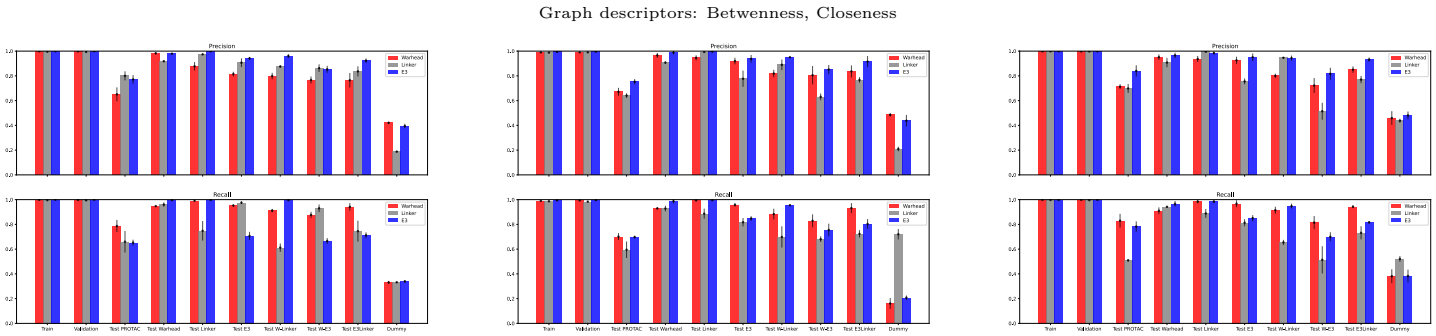
was calculated for each subgraph and then the subgraphs are mapped back to the original molecular graph. The eigenvector values of the nodes in the subgraph which map to the same original node are summed and included into the molecular graph. The summed value of each node in the molecular was then raised to the power of *0.1*. Afterwards, all node values were normalized to between *0* and *1*.

R was chosen to be $D/2.5$, where D is the diameter of the PROTAC (the longest shortest-path in the graph). The value of *2.5* was chosen empirically, as it tended to highlight the ligands well. This can be rationalized that a subgraph centered in the middle of a ligand would approximately capture half of the linker in its subgraph, assuming that the linker and ligands are approximately the same size. This implies that most subgraphs around each ligand would tend to highlight that ligand. The node values were raised to the power of *0.1* to reduce the greatest values in the graph, as without it, after the normalization only a few central nodes in each ligand would tend to be highlighted rather than the whole ligand, and raising it to this power reduced this effect. The values were normalized, as to not make this descriptor dependent on the size of the PROTAC, as larger PROTACs would return larger node values.

C.7 PROTAC Splitter Results

All results for the models, excluding Boundary bond pred. V1, as it likely had suboptimal hyperparameters and hence it is not possible to evaluate its best performance. The results for it is presented separately in Appendix C.7.1.

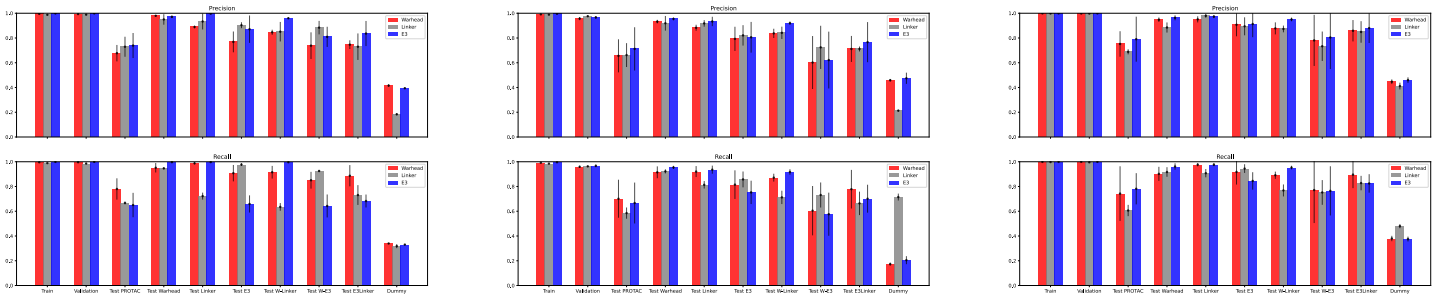
Precision and Recall



(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure C.6: Precision and recall of models with best hyperparameters for datasets with betweenness and closennes.

Graph descriptors: Betwenness, Closeness, Local eigenvector



(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

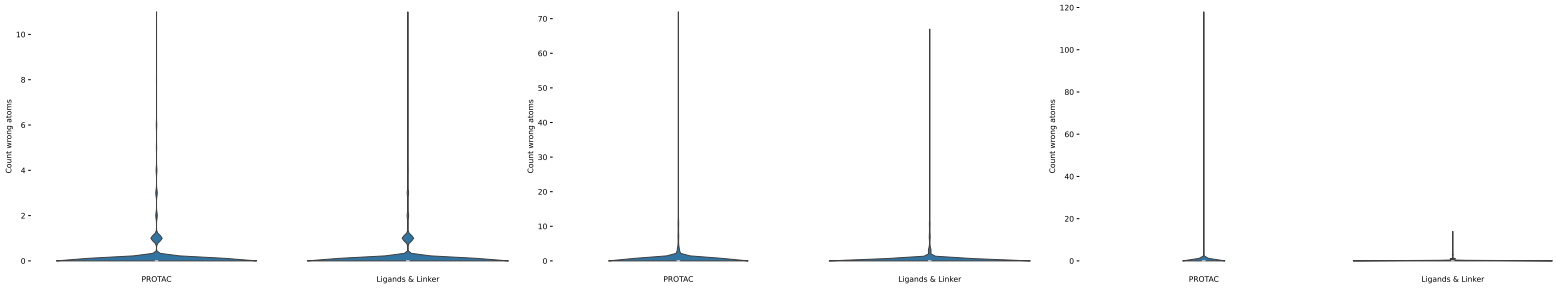
Figure C.7: Precision and recall of models with best hyperparameters for datasets with betweenness and closennes and local eigenvector.

Violin plots

Distribution over predictions of number of atoms mislabeled, for the validation set and Test PROTAC set (all 3 substructures unknown to the model). The large spikes in the validation set, for the P indicates that some PROTACs are

Validation PROTACs

Graph descriptors: Betwenness, Closeness

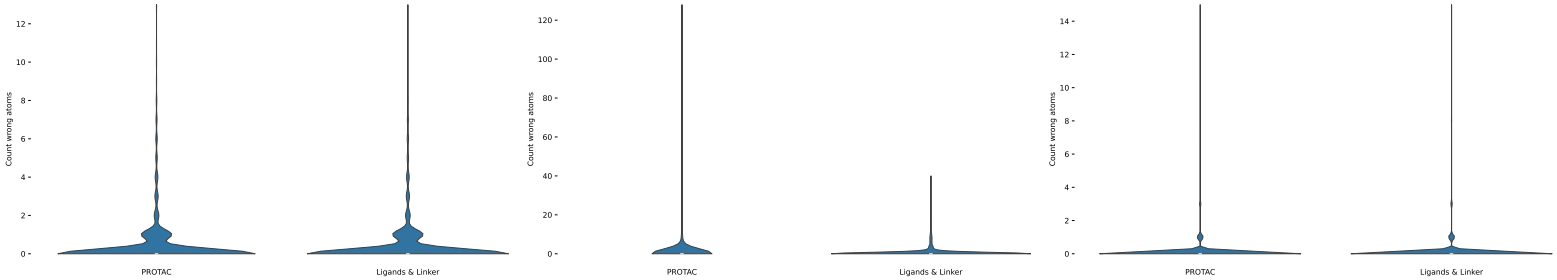


(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure C.8

Validation PROTACs

Graph descriptors: Betwenness, Closeness, Local Eigenvector

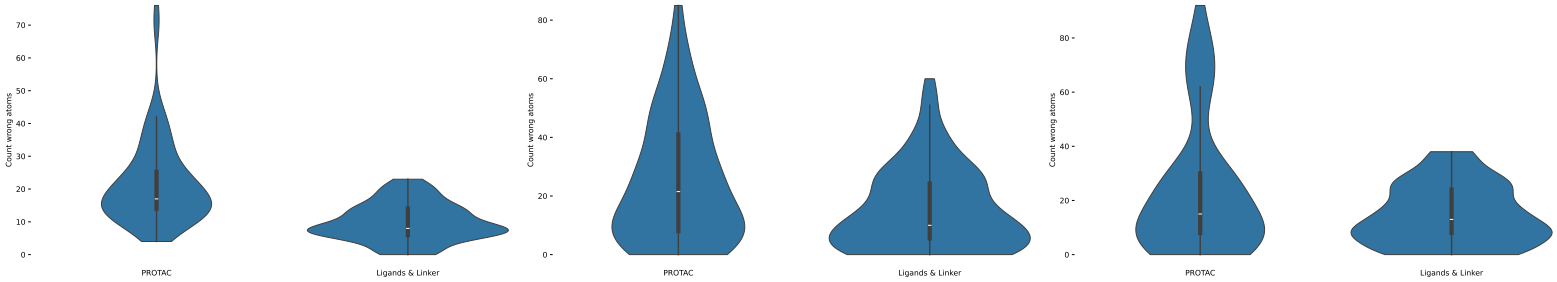


(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure C.9: Violin plots for number of mislabeled atoms for the PROTAC (3 substructure classes) and Ligands & linker (2 substructure classes).

Test PROTAC

Graph descriptors: Betwenness, Closeness

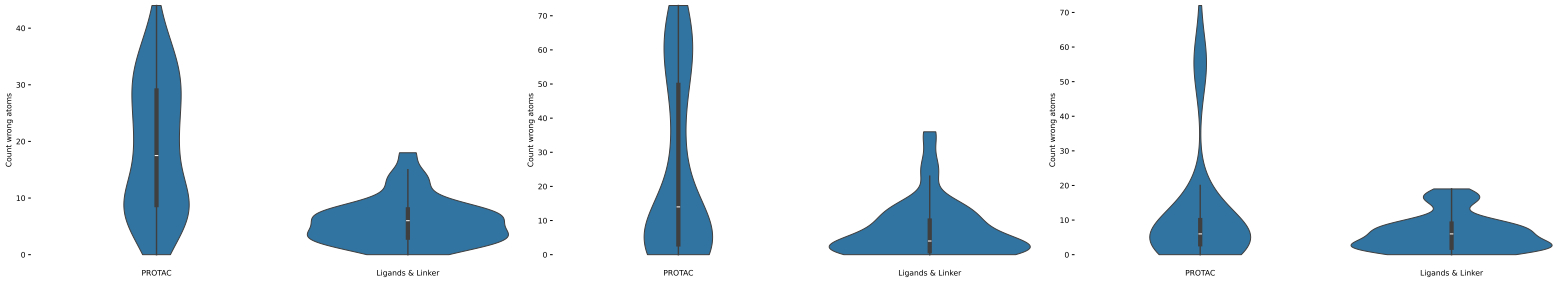


(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure C.10: Violin plots for number of mislabeled atoms for the PROTAC (3 substructure classes) and Ligands & linker (2 substructure classes).

Test PROTAC

Graph descriptors: Betwenness, Closeness, Local Eigenvector



(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

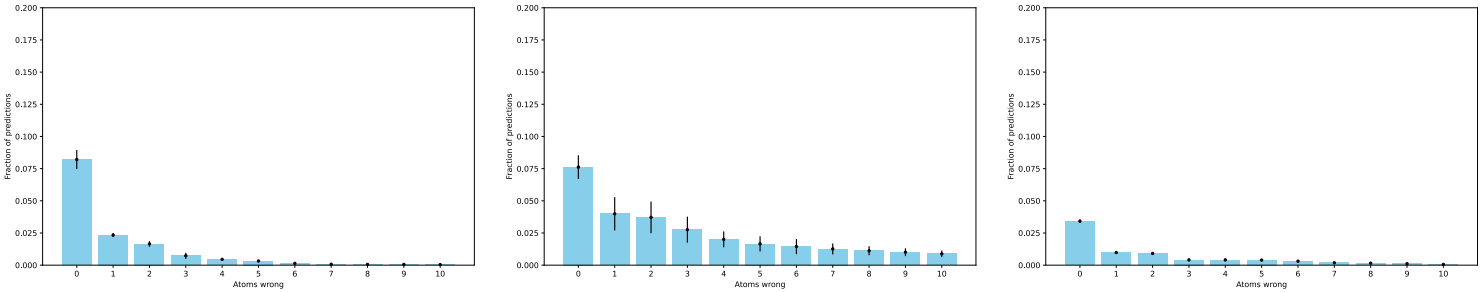
Figure C.11: Violin plots for number of mislabeled atoms for the PROTAC (3 substructure classes) and Ligands & linker (2 substructure classes).

Cumulative error

Fraction of predictions with more than the specified number of atoms mislabeled. For instance in Figure C.12a, less than 10% of the validation predictions have more than 0 atoms mislabeled, indicating that ∼90% of predictions have at most 0 atoms mislabeled.

Validation PROTACs

Graph descriptors: Betwenness, Closeness

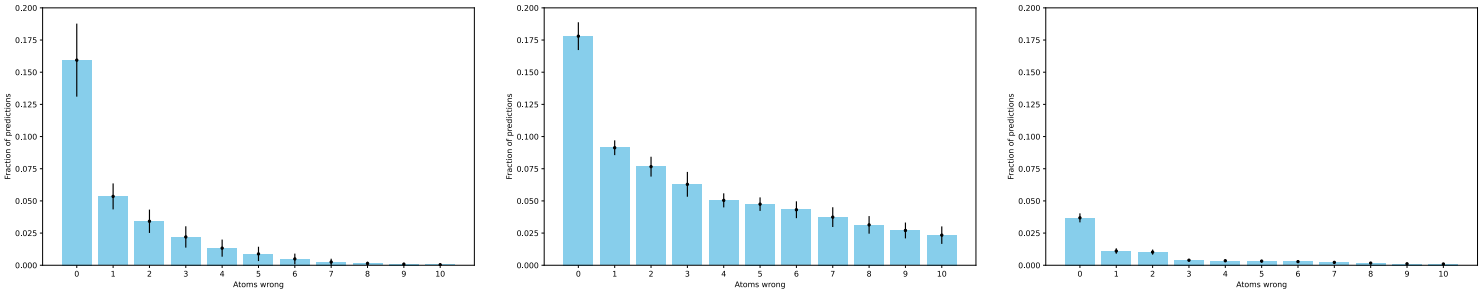


(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure C.12: Fraction of predictions with more than the specified number of atoms mislabeled.

Validation PROTACs

Graph descriptors: Betwenness, Closeness, Local Eigenvector

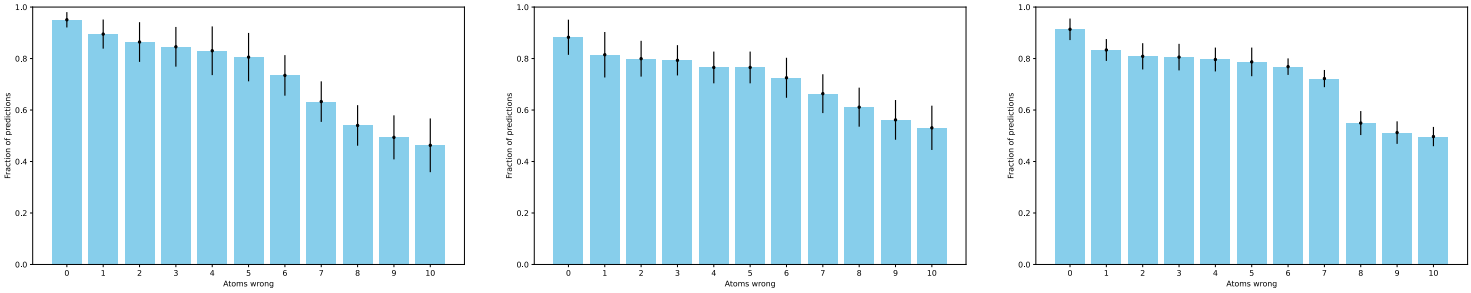


(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure C.13: Fraction of predictions with more than the specified number of atoms mislabeled.

Test PROTAC

Graph descriptors: Betwenness, Closeness

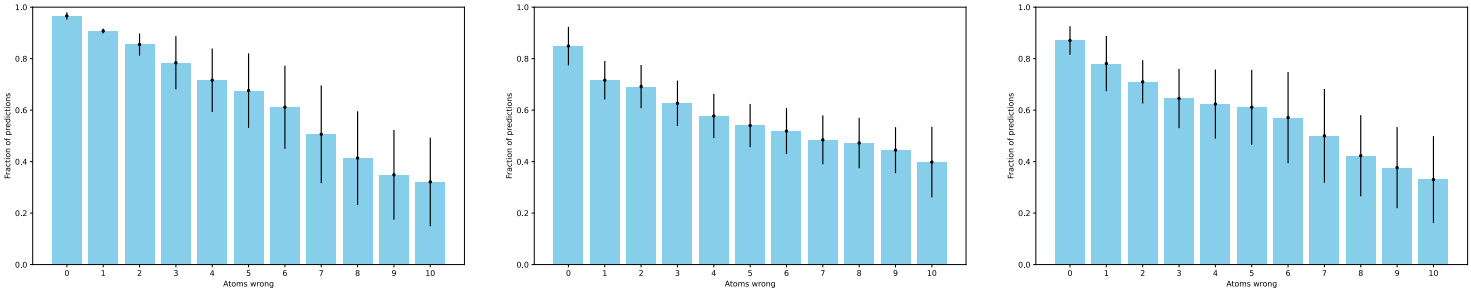


(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure C.14: Fraction of predictions with more than the specified number of atoms mislabeled.

Test PROTAC

Graph descriptors: Betwenness, Closeness, Local Eigenvector

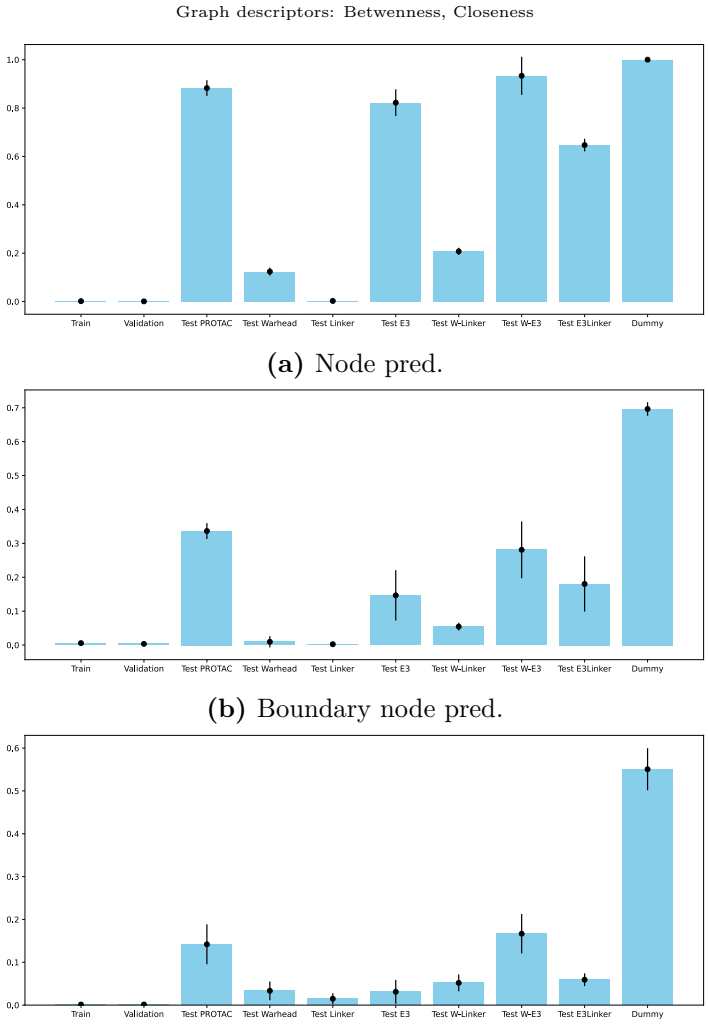


(a) Node pred. **(b)** Boundary node pred. **(c)** Boundary bond pred. Ver.2

Figure C.15: Fraction of predictions with more than the specified number of atoms mislabeled.

Flipped predictions

Flip being defined as if any true warhead node is predicted as a E3 ligand node, or *vice versa*.

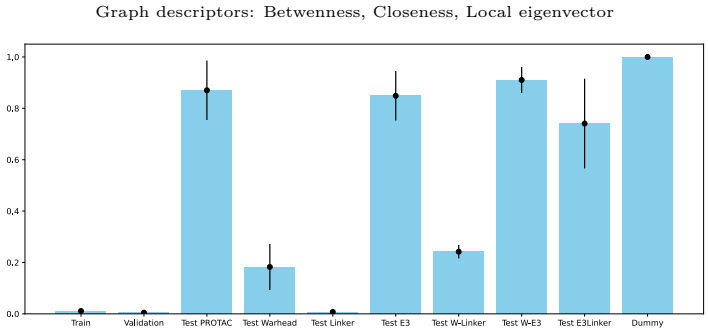


(a) Node pred.

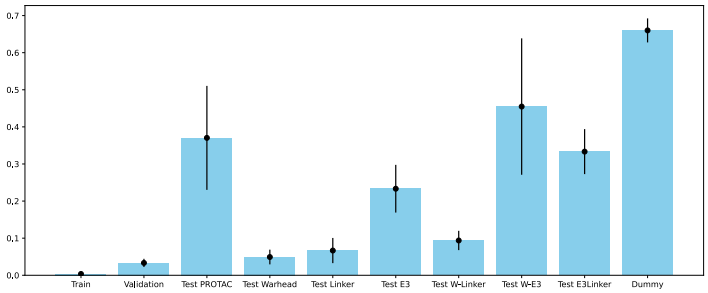
(b) Boundary node pred.

(c) Boundary bond pred. Ver.2

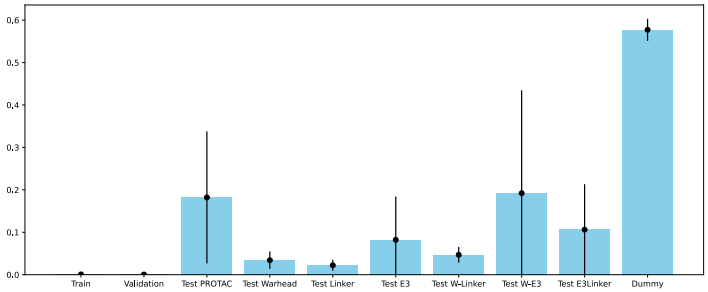
Figure C.16: Fraction of predictions that are flipped, for each dataset.



(a) Node pred.



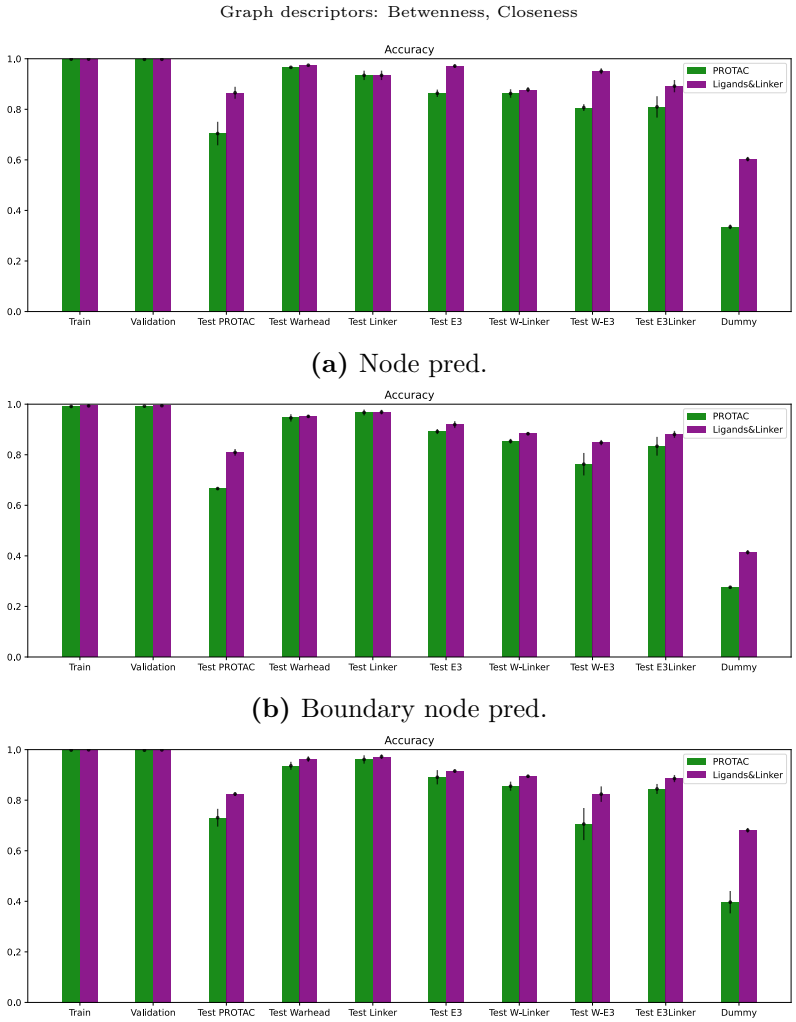
(b) Boundary node pred.



(c) Boundary bond pred. Ver.2

Figure C.17: Fraction of predictions that are flipped, for each dataset.

Ligands-linker accuracy

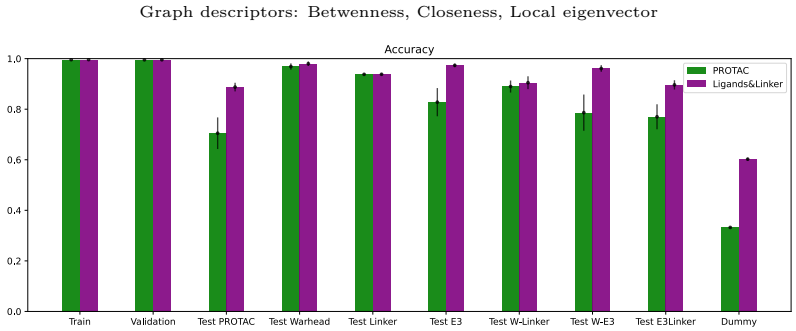


(a) Node pred.

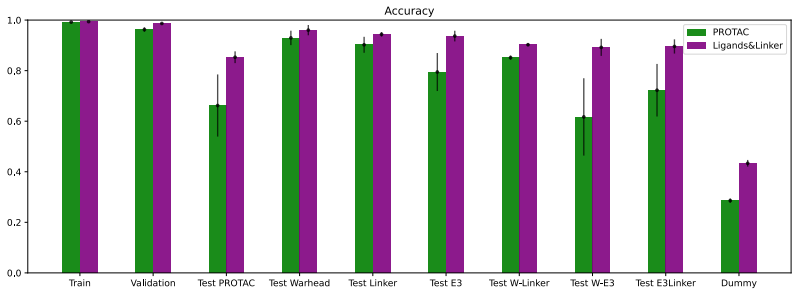
(b) Boundary node pred.

(c) Boundary bond pred. Ver.2

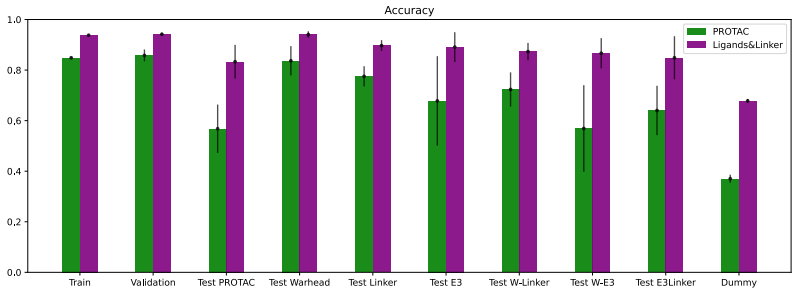
Figure C.18: Accuracy of the PROTAC if represented with two classes, namely ligands & linker.



(a) Node pred.



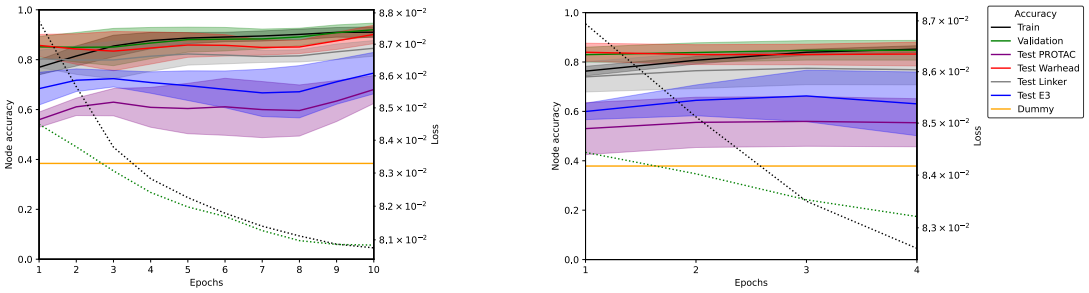
(b) Boundary node pred.



(c) Boundary bond pred. Ver.2

Figure C.19: Accuracy of the PROTAC if represented with two classes, namely ligands & linker.

C.7.1 Results of the Boundary bond predictor, version 1



(a) Without local eigenvector

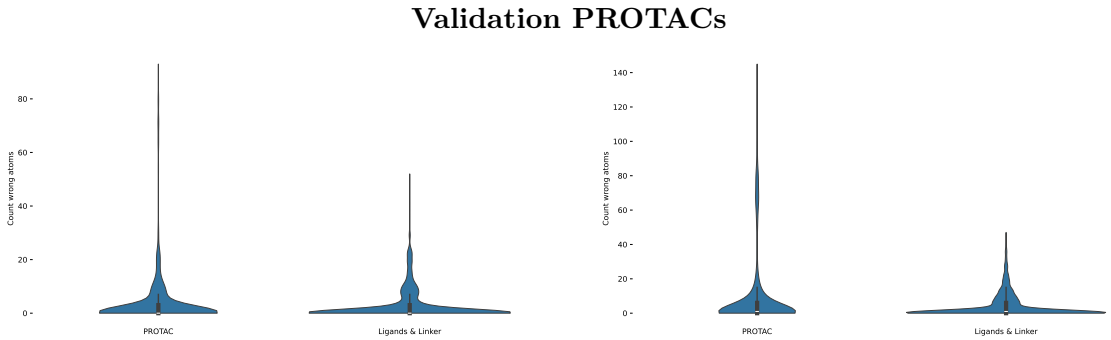
(b) With local eigenvector

Figure C.20: Training curves of boundary pred. Ver1. with hyperparameters optimized for Ver2.

Table C.3: Accuracy and validity for boundary bond pred. Ver1.

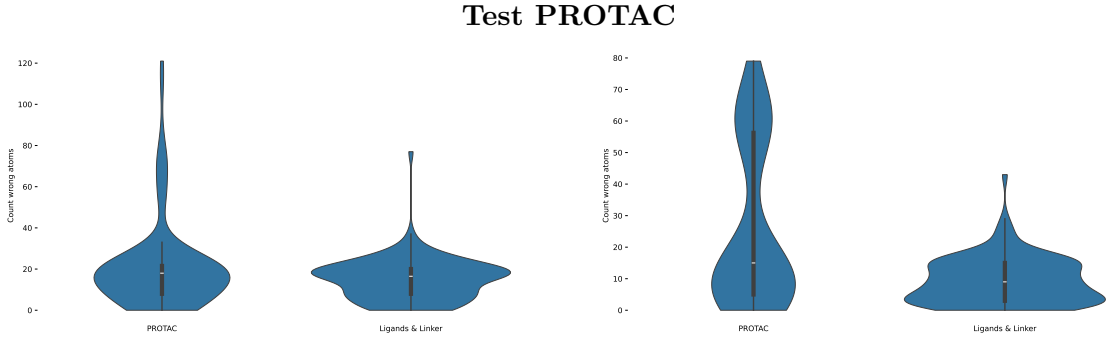
Training set size	Accuracy Without Local Eigenvector	Accuracy With Local Eigenvector	Validity Without Local Eigenvector	Validity With Local Eigenvector
Training	92.0 ± 1.8	84.9 ± 0.4	100.0 ± 0.0	100.0 ± 0.0
Validation	93.8 ± 1.0	85.8 ± 2.3	100.0 ± 0.0	100.0 ± 0.0
Test PROTAC	75.4 ± 1.9	56.8 ± 9.6	100.0 ± 0.0	100.0 ± 0.0
Test Warhead	91.8 ± 3.4	83.7 ± 5.8	100.0 ± 0.0	100.0 ± 0.0
Test Linker	86.9 ± 2.3	77.5 ± 4.0	100.0 ± 0.0	100.0 ± 0.0
Test E3	79.4 ± 2.1	67.8 ± 17.7	100.0 ± 0.0	100.0 ± 0.0
Test Warhead-Linker	85.3 ± 2.2	72.3 ± 6.7	100.0 ± 0.0	100.0 ± 0.0
Test Warhead-E3	75.2 ± 1.4	56.9 ± 17.1	100.0 ± 0.0	100.0 ± 0.0
Test Linker-E3	74.9 ± 1.1	64.0 ± 9.8	100.0 ± 0.0	100.0 ± 0.0
Dummy	36.8 ± 1.5	37.0 ± 1.6	100.0 ± 0.0	100.0 ± 0.0

Precision and recall were not calculated for these runs to shorten training time. However, these should be equal to or lower than for Boundary pred. Ver2. as the accuracies across the datasets are equal or lower. As such, the lack of precision and recall for this model does not change the model selected as the best.



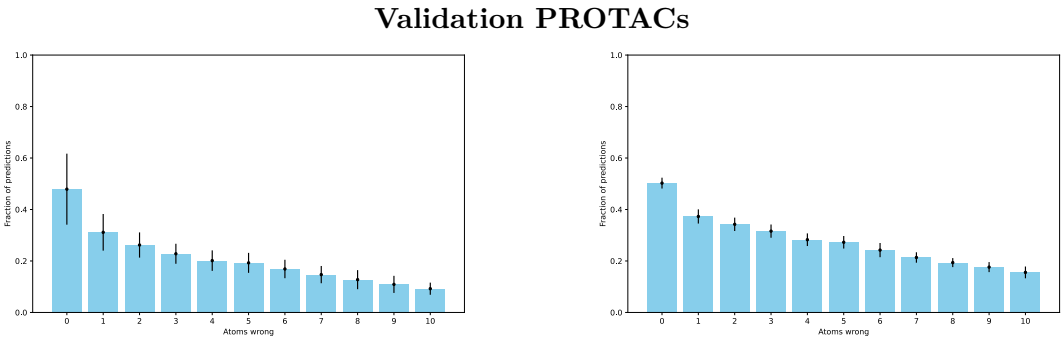
(a) Without local eigenvector **(b)** With local eigenvector

Figure C.21: Violin plots for number of mislabeled atoms for the PROTAC (3 substructure classes) and Ligands & linker (2 substructure classes).



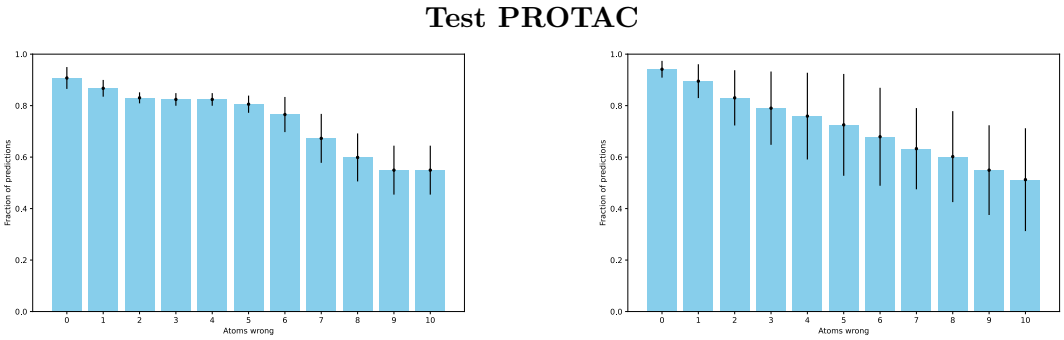
(a) Without local eigenvector **(b)** With local eigenvector

Figure C.22: Violin plots for number of mislabeled atoms for the PROTAC (3 substructure classes) and Ligands & linker (2 substructure classes).



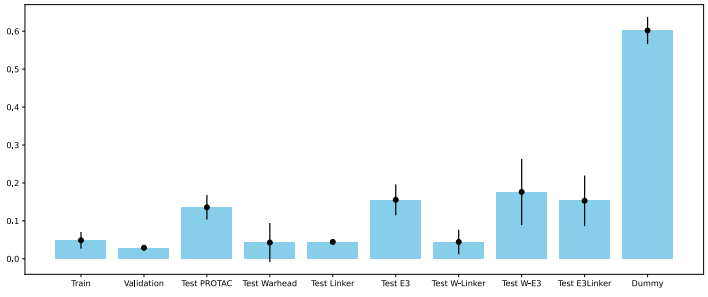
(a) Without local eigenvector **(b)** With local eigenvector

Figure C.23: Fraction of predictions with more than the specified number of atoms mislabeled.

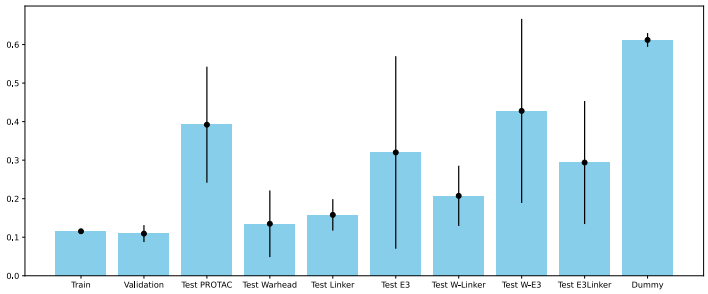


(a) Without local eigenvector **(b)** With local eigenvector

Figure C.24: Fraction of predictions with more than the specified number of atoms mislabeled.

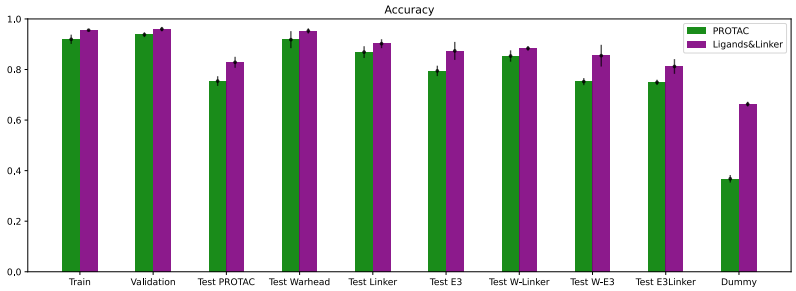


(a) Without local eigenvector

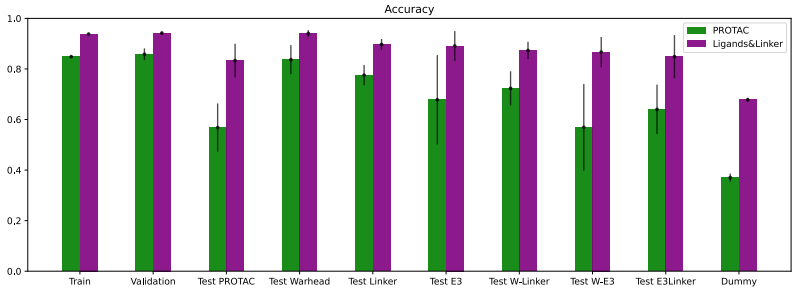


(b) With local eigenvector

Figure C.25: Fraction of predictions that are flipped, for each dataset.



(a) Without local eigenvector



(b) With local eigenvector

Figure C.26: Accuracy of the PROTAC if represented with two classes, namely ligands & linker.

D

Failed projects

D.1 Various datasplits

Datasplit via Butina clustering

Butina clustering was found to lead to slight data leakage due to Butina clustering not being designed to control for similarity between every molecule between clusters. Figure D.1 illustrates a hypothetical clustering where two clusters have a molecule within their cutoff range, but it is only assigned to the first cluster due to it having more neighbors.

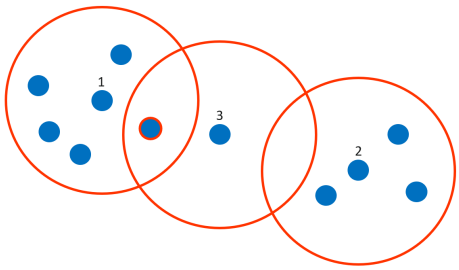
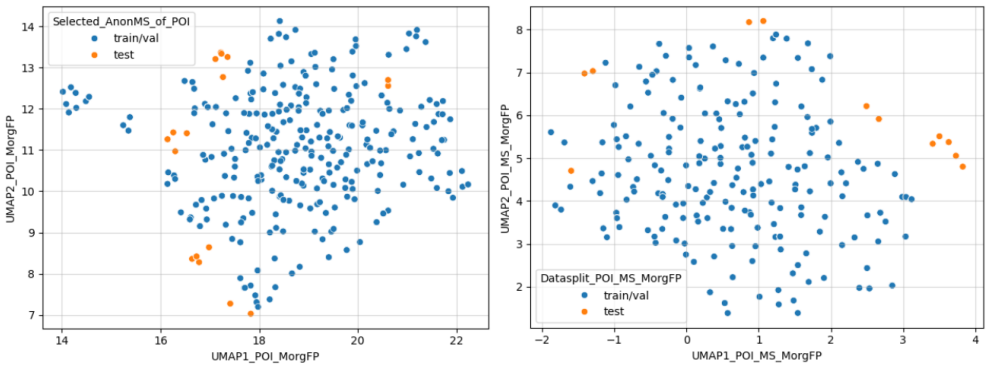


Figure D.1: Three clusters from Butina clustering. The first and third clusters central molecules are similar to the a highlighted molecule.

Datasplit via UMAP and PCA

The Morgan fingerprint was calculated for each substructure, and using the bits as different dimensions, a UMAP was created. Figure D.2 displays an example of how the substructures were selected for the testset using an UMAP. The principle was to select outlying clusters of murcko scaffolds, that were also outlying as substructures and as a frameworks. This was done for the warhead, linker and E3 ligand as to create testsets for the substructures.

However, this always resulted in some substructures that had a high tanimoto similarity to the training set. This is undesired as it would lead to data leakage. Multiple different selections were hand picked, but none of them was satisfactory. It was realized that UMAP does not necessarily preserve distances and apparently distant clusters may actually be close. To this end, PCA was chosen, as it better conserves relative distances on the 2D projection from the higher-dimensional space. The same procedure was tried with PCA, but no selected test set had an acceptable dissimilarity between the training and the test set.



(a) Substructures **(b)** UMAP of Murcko scaffolds

Figure D.2: UMAP of warheads (a), and their Murcko scaffold (b). The selected test set is highlighted in orange.

Custom datasplits

Code was written to produce clusters with similar properties alike HDBSCAN, where no pair of molecules between clusters shared a tanimoto similarity above a set cutoff. It was also set that all clusters which shared a graph framework between any pair of molecules would be joined, as to allow for a training and test set of unique graph frameworks. However, this turned out to join many clusters and it was unfeasable to set a desirable cutoff of the tanimoto similarity, simultaneously to cluster enough substructures to create a test set with. Together, these were too restrictive of a clustering criteria.

It was tried to group substructures with the same graph framework, but this had the effect of grouping very different the small substructures, that e.g. consisted of a single ring. Ideally, all molecules within a cluster should be similar by a tanimoto similarity measure as well. Another issue was that this did not control for data leakage, and a pair of molecules between clusters could have slightly different graph frameworks but have a high tanimoto similarity. As such, these two custom methods of clustering was discontinued.

D.2 Reproducing ubiqutination prediction studies

At the beginning of the thesis it was planned that predicted ternary structures and substructures would be incorperated into a model which predicts PROTAC degradation. Evidently, this was too big of a task for a master's thesis. However, in addition to these incorporating these structures, it was also planned to incorporate an existing ubiqutination prediction tool from the literature, which would predict which amino acid on a protein could get ubiquitinated.

Nine studies were identified and only one was usable [58], but only through their website (http://gpsuber.biocuckoo.cn/) as they did not publish their code. Another study published their code on a website (http://nsclbio.jbnu.ac.kr/tools/UbiComb/) that is now inaccessible [59]. [60] published their code on github, but they did not to upload their getData.py file, which the assessment.py requires. [61] The file MRMD_dimensional_reduction.py calls set_end.csv, yet it is not among the uploaded data or created by any other uploaded file in their github repository. [62] requires R, matlab, and python, and the file Aaindex.py imports data from data AAidx_sll.txt which is part of their local machine, and is not uploaded to the github repository. [63] did not publish their training data. [64] states that the availability of the data

and code is "not applicable". [65] is a conference article, regardless, they did not publish their code. [66] stated an incredible accuracy of 100%, but despite their apparently successful model, the article only has two citations and has not passed peer review since 2021.

DEPARTMENT OF LIFE SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY