THE UNIVERSITY *of* EDINBURGH

THE UNIVERSITY
of EDINBURGH

Machine Learning in Drug Discovery: Advancing Protein-Ligand Binding Affinity Predictions

Rohan Gorantla

A thesis presented for the degree of

Doctor of Philosophy

School of Informatics

University of Edinburgh

United Kingdom

May 7, 2025

Declaration

I hereby declare that I, Rohan Gorantla, have composed this thesis, "*Machine Learning in Drug Discovery: Advancing Protein-Ligand Binding Affinity Predictions*" of my own and that the research presented in this work is of my making. I confirm that:

- the thesis has been composed by me while in candidature for a doctorate degree at the University of Edinburgh.
- the research entities presented in this thesis are my own and that where collaborative efforts were involved this has been clearly stated.
- the work has not been submitted for any other degree or professional qualification except as specified.
- where the published work of peers has been consulted to support the current work this has been clearly stated.
- where I have quoted from the work of others, the source is always given and applicable software licenses are referred to. With the exception of such quotations, this thesis is entirely my own work.

May 7, 2025

....

Rohan Gorantla Date

Abstract

Machine Learning in Drug Discovery: Advancing Protein-Ligand Binding Affinity Predictions

Binding affinity quantifies the strength of the interaction between a protein and a small drug-like molecule. Accurately determining binding affinity helps identify promising drug candidates in the early stage of drug discovery, particularly in hit discovery and lead optimization phases, where screening several millions to even billions of compounds is required. Hit discovery involves identifying potential compounds (known as 'hits') that show initial activity against the choice of disease-causing protein target. Lead optimization focuses on refining these hits to improve their binding affinity and other drug-like properties. Experimental assays are the gold standard for determining binding affinity, but they are not practical for rapidly screening millions of drug-like compounds against potential targets. Accurate *in silico* prediction of protein-ligand binding affinity can significantly expedite drug discovery by streamlining the identification and optimization of viable drug candidates, reducing huge experimental costs and time.

Over the last fifty years, a wide range of *in silico* binding affinity prediction strategies have been developed. They consist of both structure-based and ligand-based approaches. However, these methodologies often fall short in large-scale screenings. So called docking methods, while capable of high-throughput screenings, often lack the desired accuracy for a binding affinity prediction. In contrast, alchemical free energy-based (AFE) techniques, a simulation-based technique, offer improved accuracy but are computationally demanding. The rapid progression of machine learning, coupled with increased accessibility to binding affinity data, opens

avenues to deep learning-based methods for improving the accuracy and speed of binding affinity predictions.

This thesis focuses on exploring and developing machine learning methods for predicting protein-ligand binding affinity. The first part of the thesis investigates how current deep learning models learn from input protein and ligand data to predict binding affinity. Systematic experiments using publicly available kinase datasets are conducted to assess the impact of protein encodings and ligand encodings derived from convolutional and/or graph neural networks by inputting variations of protein and ligand data. The results indicated that protein encodings have minimal impact on binding predictions, while ligand-based features play a more substantial role in model performance.

The second part of the thesis focuses on addressing key challenges at the model, data, and evaluation levels of the deep learning framework for predicting binding affinity. To overcome challenges at the model level, this work introduces a deep learning framework for predicting **b**inding **a**ffinity using pretrained protein and ligand **l**anguage **m**odels, called BALM. Utilizing pretrained language models for proteins and ligands, the BALM method predicts binding affinity by optimizing the distance between protein and ligand encodings using a cosine similarity metric. At the data and evaluation levels, the research demonstrates novel strategies for training and testing these models to ensure they provide meaningful and reliable predictions compared to traditional methods and experimental measurements. While zero-shot prediction on unseen targets may not always be reliable, the few-shot finetuning of the BALM model is shown to be reliable for screening new targets, demonstrating better performance than docking.

The final part of this thesis focuses on integrating machine learning with physics-based simulation methods, such as alchemical free energy calculations. This integration aims to reduce computational costs and time during lead optimization. Specifically, Active Learning (AL) is used to intelligently select compounds for AFE calculations, making the identification of top binders more efficient. AL is an it-

erative process that learns binding affinities from an unlabelled dataset and helps prioritize compounds for detailed evaluation, minimizing the need to compute AFE for all compounds in a large pool. In this approach, machine learning models such as Gaussian process regression and pretrained graph neural network-based models act as surrogate models during each AL iteration. They provide predictions of binding affinity to inform the next set of AFE calculations, allowing for efficient compound selection. Various recommendations on model choice, batch size, and strategies for exploring or exploiting the chemical spaces based on the ligand pool used are provided. Both models show similar recall in identifying top binders on large datasets, but the Gaussian process model performs better than the pretrained graph network model when the training data is limited. Using a larger initial batch size, especially with diverse datasets, improved recall for both models and enhanced overall correlation metrics. However, smaller batch sizes were more effective for later iterations.

Lay Summary

Developing new drugs is an extremely challenging process that usually takes 10-15 years and can cost over $1 billion. Drugs typically work by attaching, or "binding," to proteins in our body. Proteins are molecules that control many important functions, and certain proteins can cause diseases when they malfunction. Drugs can help by sticking tightly to these problematic proteins and stopping or modifying their activity. A key step in discovering new drugs is finding out how strongly a drug molecule attaches to its target protein—a measurement called "binding affinity". It is crucial in identifying promising drug candidates from vast collections of molecules. Knowing this helps promising initial compounds from large libraries containing millions of molecules during the early screening of drug-like molecules and for later optimizing these compounds to improve their binding strength for a given protein. Though accurate, traditional lab methods for measuring binding affinity are slow and expensive, making them impractical for efficiently screening millions or billions of molecules.

Computational methods have been developed over the last forty years to expedite the screening process, ranging from quick but less accurate methods (such as docking) to slower, highly accurate simulations (such as physics-based methods). Recently, machine learning (computational methods that learn patterns from data) has emerged as a promising approach, offering a good balance between speed and accuracy in predicting binding affinity.

My research focuses on exploring and improving machine learning methods to predict better how drugs bind to proteins, particularly for efficiently screening large

compound libraries. Initially, I investigated what current machine learning models focus on when making predictions. I found that these models rely mostly on drug information, with less emphasis on the target protein information. This discovery highlighted important limitations of existing methods. Based on this insight, I developed BALM, a new technique that uses advanced techniques (called large language models) trained on millions of proteins and drug-like compounds data points. BALM quickly and effectively predicts binding affinity, performing better than traditional docking methods and requiring less computational power. Additionally, I showed that BALM can be easily adapted to screening new targets using a small amount of available experimental data. To ensure the reliability of BALM and similar models, I also introduced new evaluation standards, overcoming limitations in previous testing approaches.

Once we have a promising shortlist of drugs against the given target, precise but computationally intensive physics-based methods or lab experiments are essential for optimizing these drug candidates. Here, I explored the "Active Learning" method, a technique to smartly select the most promising molecules for testing with these methods by significantly reducing the resources needed. By intelligently prioritizing compounds, I showed how active learning can speed up the process and reduce costs compared to randomly selecting compounds for testing.

In summary, this thesis demonstrates how machine learning methods can accelerate the initial stages of drug discovery by enhancing the speed and reliability of binding affinity predictions. These advancements could facilitate the quicker and more cost-effective delivery of new medicines to patients, ultimately leading to improved healthcare outcomes.

Acknowledgements

The PhD journey has been an incredible adventure of scientific discovery and personal growth. I am deeply grateful to everyone who has supported me along the way and made the last three years both enriching and memorable.

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Toni, for her exceptional guidance and mentorship. Her passion for science has been truly inspiring, and I have thoroughly enjoyed our brainstorming sessions and discussions. Her support in helping me develop as a scientist, from conducting rigorous research to creating effective scientific figures, has been invaluable. I am equally grateful to my second supervisor, Andrea, for her unwavering support and guidance throughout this journey. Her insights and perspective have significantly enriched my research experience.

Over the past three years, I have had the privilege of working with fantastic collaborators. I have been fortunate to do an internship at Exscientia during my second year, offering me valuable insights into real-world drug discovery. I would like to thank Alzbeta, Ben Cossins, and Ben Suutari at Exscientia for providing an excellent learning experience and mentoring. The development of BALM would not have been possible without the exceptional contributions of Aryo, Ian, Jordi, and Alvaro. I am also grateful to Christian and Frank for our wonderful collaboration, which deepened my understanding of graph neural network theories.

I would like to extend my appreciation to everyone in the Mey group, Andrea's group, and the Biomedical AI CDT group at Informatics for creating such a welcoming environment and engaging in stimulating scientific discussions. You have been an invaluable sounding board for my ideas and an almost infinite source of

knowledge. Thanks to each one of you for all the support and help.

A special note of gratitude goes to my first research mentor, Rajeev, who has been a constant presence in my research journey as a mentor, supporter, and friend. His guidance has been instrumental in shaping my academic path. Special thanks to the CDT Administration team - Ian, Diego, Isabelle and Ekaterina who have always been a constant support in this journey.

I am very grateful for all the friendships I have formed in the beautiful city of Edinburgh, particularly in the university accommodation. You all made me feel at home. Each of you has enriched my experience in unique ways, provided support when needed, and been a family.

Finally, to my parents and grandparents, I cannot imagine my world without your presence. Your unconditional support of my every endeavour and your role in shaping who I am today means more than words can express.

Contents

List of Figures

List of Acronyms

ABFE	Absolute Binding Free Energy
ADME	Absorption, Distribution, Metabolism, Excretion
AFE	Alchemical Free Energy
AL	Active Learning
ATP	Adenosine Triphosphate
BA	Binding Affinity
BERT	Bidirectional Encoder Representations from Transformers
BLAST	Basic Local Alignment Search Tool
BRD2	Bromodomain Containing 2
CADD	Computer-Aided Drug Design
CHEMBL	Chemistry European Molecular Biology Laboratory
CHARMM	Chemistry at Harvard Macromolecular Mechanics
CI	Concordance Index
CNN	Convolutional Neural Network
COVID	Coronavirus Disease
CPU	Central Processing Unit
CRISPR	Clustered Regularly Interspaced Short Palindromic Repeats
CUDA	Compute Unified Device Architecture
DL	Deep Learning
DNA	Deoxyribonucleic Acid
DNN	Deep Neural Network
D2R	Dopamine Receptor D2

DSF Differential Scanning Fluorimetry

ECFP Extended-Connectivity Fingerprint

ESM Evolutionary Scale Modeling

FDA Food and Drug Administration

FBDD Fragment-Based Drug Discovery

FC Fully Connected

FEP Free Energy Perturbation

GDB Generated Database

GLP Good Laboratory Practice

GNN Graph Neural Network

GP Gaussian Process Regression

GPU Graphical Processing Unit

GWAS Genome-Wide Association Studies

HIF2A Hypoxia-Inducible Factor 2 Alpha

HIV Human Immunodeficiency Virus

HMM Hidden Markov Model

HSP Heat Shock Protein

HTS High-Throughput Screening

IC50 Half-Maximal Inhibitory Concentration

IDP Intrinsically Disordered Protein

IND Investigational New Drug

ITC Isothermal Titration Calorimetry

KIBA Kinase Inhibitor Bioactivity

MACCS Molecular ACCess System

MAE Mean Absolute Error

MCC Matthews Correlation Coefficient

MCL1 Myeloid Cell Leukemia Sequence 1

MCS Maximum Common Scaffold

MD Molecular Dynamics

ML	Machine Learning
MLM	Masked Language Modeling
MLP	Multilayer Perceptrons
MLR	Multiple Linear Regression
MM	Molecular Mechanics
MMPBSA	Molecular Mechanics Poisson-Boltzmann Surface Area
MPNN	Message-Passing Neural Network
mAb	Monoclonal Antibody
Mpro	Main Protease
MSE	Mean Squared Error
MSM	Markov State Model
MST	Microscale Thermophoresis
MUE	Mean Unsigned Error
NLP	Natural Language Processing
NMR	Nuclear Magnetic Resonance
NN	Neural Network
PCA	Principal Component Analysis
PC	Principal Component
PCM	Protein Contact Map
PDB	Protein Data Bank
PD	Pharmacodynamic
PK	Pharmacokinetic
PL	Protein-Ligand
PSC	Protein Sequence Composition
PSSM	Position-Specific Scoring Matrix
QM	Quantum Mechanical
QSAR	Quantitative Structure-Activity Relationships
RBFE	Relative Binding Free Energy
ReLU	Rectified Linear Unit

RF	Random Forest
RMSD	Root Mean Square Deviation
RMSE	Root Mean Squared Error
RNAi	RNA Interference
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SAR	Structure-Activity Relationships
SAS	Solvent-Accessible Surface
SB	Structure-Based
SEM	Standard Error of the Mean
SF	Statistical Fluctuation
SGD	Stochastic Gradient Descent
SILAC	Stable Isotope Labeling by Amino Acids in Cell Culture
SMARTS	SMILES Arbitrary Target Specification
SMILES	Simplified Molecular Input Line Entry System
SPR	Surface Plasmon Resonance
SVM	Support Vector Machine
SVR	Support Vector [Machine] Regression
SYK	Spleen Tyrosine Kinase
TDD	Target-Based Drug Discovery
TDC	Therapeutic Data Commons
TYK2	Tyrosine Kinase 2
ULVS	Ultra-Large Virtual Screening
UMAP	Uniform Manifold Approximation and Projection
UCB	Upper Confidence Bound
USP7	Ubiquitin-Specific Protease 7
WHAM	Weighted Histogram Analysis Method

Publications Supporting Thesis

The research presented in this thesis has led to the following publications:

- **From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction**

  Gorantla, R., Kubincova, A., Weiße, A. Y., & Mey, A. S.

  J. Chem. Inf. Model. 2024, 64, 7, 2496–2507

- **Benchmarking active learning protocols for ligand binding affinity prediction**

  Gorantla, R., Kubincova, A., Suutari, B., Cossins, B. P., & Mey, A. S.

  J. Chem. Inf. Model. 2024, 64, 6, 1955–1965

- **Learning Binding Affinities via Fine-tuning of Protein and Ligand Language Models**

  Gorantla, R., Gema, A. P., Yang, I. X., Serrano-Morrás, Á., Suutari, B., Jiménez, J. J., & Mey, A. S.

  bioRxiv 2024.11.01.621495, 2024

Other work done during my PhD but not part of my thesis:

- **Dirac–Bianconi Graph Neural Networks-Enabling long-range graph predictions**

  Nauck, C., Gorantla, R., Linder, M., Schürholt, K., Mey, A. S., & Hellmann. F.

  ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling

Chapter 1

Introduction

The discovery of new medicines is pivotal in advancing human health and global healthcare by combating diseases. Developing a new medicine is a complex and costly process involving multiple stages that require specialized scientific expertise and meticulous methodologies. It is estimated that bringing a new drug to market can cost approximately $2.5 billion and typically spans 12-15 years[1,2]. The success probability after identifying and advancing a new drug through clinical trials is relatively low, with only about 35% of candidates progressing to clinical trials and a mere 9-14% achieving regulatory approval from Phase 1 trials[3]. The average failure rate in clinical trials from 2009 to 2018 reached 84.6%[4]. These high development costs and low success rates in clinical trials slow the process of bringing new drugs to the market[5]. The ten-year rolling average for new drug approvals by the United States Food and Drug Administration (FDA) currently stands at 46 per year[6]. To improve this success rate and expedite the generation of new ideas, computational methods have proven to be instrumental[5,7–10].

Breakthroughs in medical research have paved the way for diverse treatment modalities, ranging from traditional small molecules and biologics, such as monoclonal antibodies (mAbs) and vaccines, to cutting-edge therapies, including cell, gene, and radioligand therapies tailored to various diseases. Biologics, including mAbs, are engineered to bind specifically to target proteins, offering high specificity in treating various diseases[11]. The recent advancements in mRNA vaccines have

revolutionized immunization, enabling rapid development and high efficacy against infectious diseases[12]. Emerging advanced therapies, such as gene therapy, involve introducing, removing, or altering genetic material within a patient's cells to treat or prevent disease, while cell therapy includes transplanting human cells to replace or repair damaged tissues and cells[13]. Similarly, radioligand therapy, which uses radioactive substances linked to ligands specifically targeting cancer cells, shows potential in oncology[14]. Despite this, these advanced modalities face considerable challenges related to manufacturing complexity, administration, and scalability[15]. While advanced therapies continue to evolve, small molecules are expected to retain a central role in the future of pharmacology due to their manufacturability, favourable pharmacological properties, and broad applicability[10,16]. Small molecules, typically defined by their low molecular weight, can be designed for oral administration and can penetrate cell membranes to reach intracellular targets[10]. Small-molecule drugs interact with proteins, often binding to active sites to modulate their function, effectively addressing dysregulation and potentially reversing disease phenotypes[16,17]. Given the continued relevance and potential of small molecules in therapeutic development, this thesis will focus specifically on small-molecule drug discovery.

Historically, many drugs were discovered by observing their effects on normal or diseased physiology. This approach, known as phenotypic drug discovery (PDD), involved empirically observing therapeutic effects directly in humans as part of traditional medicine or disease models[18]. PDD focuses on finding biologically active agents that can directly alter the phenotype of a cell or organism. This approach was particularly advantageous for poorly understood diseases or where the disease-related gene was considered 'undruggable' as it could identify molecules that act on multiple targets, a concept known as polypharmacology[19].

With the advent of the molecular biology revolution in the 1980s and the sequencing of the human genome in 2001, the focus of drug discovery shifted to specific molecular targets, leading to the rise of target-based drug discovery (TDD)[20]. TDD has been the predominant approach for over 30 years, leveraging the detailed

understanding of disease mechanisms at the molecular level[21]. Despite its successes, many drug candidates fail in clinical trials due to efficacy or safety concerns, highlighting the complexity of diseases. While both phenotypic and target-based drug discovery approaches have their strengths, recent efforts are focused on integrating the benefits of both approaches to enhance drug discovery outcomes.

The traditional drug development pipeline faces various challenges, including understanding disease biology, designing and identifying novel drug molecules, and assessing compound quality during pre-clinical and clinical trials[2,8]. Given the immense scale of chemical space estimated to contain over 10^{24} synthesizable, drug-like compounds, the process of finding suitable candidates cannot rely on random experimentation or serendipity[22]. To address these challenges, over the past five decades, the pharmaceutical industry has increasingly integrated computational tools into nearly every stage of the drug discovery pipeline[23]. These tools help narrow down complex objectives by modelling interactions, predicting activity profiles, and optimizing pharmacokinetic properties, thereby reducing the need for extensive empirical screening and experimental iterations[10]. The computational approaches developed to speed up the early stages of small-molecule drug discovery can be broadly categorized into structure-based and ligand-based methods.

Structure-based methods rely on the three-dimensional structures of target proteins to rationally design molecules that modulate protein activity[24]. These approaches predict or simulate how molecules interact with protein targets in three-dimensional space, enabling the testing or designing of compounds that can effectively interact with the target of interest. Key structure-based methods include molecular dynamics (MD) simulations, molecular docking, and de novo design[23,24]. MD simulations use Newtonian mechanics to simulate atomic movements over time, thus allowing the capture of conformational changes in the proteins and identifying potential binding sites and binding events at atomic resolution. The foundations for modern MD simulations were laid in the 1970s. Although early quantum mechanics or molecular mechanics approaches were demonstrated by Warshel and Levitt[25,26],

the first extensive classical MD simulations resembling contemporary approaches, were conducted by McCammon et al. in the late 1970s[27]. Molecular docking predicts the preferred binding orientations of small molecules within protein binding sites. The pioneering docking method, DOCK, was developed by Kuntz et al. in the early 1980s[28], providing a foundation for virtual screening techniques later expanded by software such as AutoDock[29] and GOLD[30]. These methods allow the rapid assessment of large libraries of compounds for their potential interactions with protein targets. De novo design approach aims to create entirely new molecules tailored to specific protein targets by assembling molecular fragments that satisfy the geometric and chemical constraints of the binding site. Emerging prominently in the early 1990s, notable early de novo design tools include LUDI[31] and SPROUT[24,32].

Ligand-based methods, on the other hand, focus on known ligands that bind to the target of interest when the protein's 3D structure is unknown[33]. Quantitative structure-activity relationship (QSAR) studies, first developed in the 1960s, correlate molecular properties with biological activity to guide the optimization of drug compounds[34]. Pharmacophore modelling tools such as DISCO[35] and GASP[36], introduced in the 1990s, have been important in identifying the essential features of ligands responsible for their biological activity[37]. These computational methods have contributed to the development of drugs for diseases, such as HIV protease inhibitors, saquinavir and ritonavir for HIV treatment[38], and targeted cancer therapies such as imatinib[39].

Machine learning (ML) has long played an integral role in drug discovery for developing QSAR models. Several ML techniques, including neural networks[40] (NNs), support vector machines[41] (SVMs), and random forests[42] (RFs), introduced from the 1990s to early 2000s, have addressed various classification and regression tasks in structure-activity relationship analyses and chemical property predictions[43]. In the last decade, deep learning (DL), a subfield of ML, has emerged as a powerful tool in drug discovery with increased availability of experimental data and computing power, offering new capabilities from target discovery to lead optimization

stages[5,44]. A notable success was AlphaFold[45], which revolutionized protein structure prediction, demonstrating the potential of ML in structural biology and target identification. ML methods have also shown promise in predicting Absorption, Distribution, Metabolism, Excretion, and Toxicity (ADMET) properties, which is crucial for assessing drug candidates' potential efficacy and safety[46]. Generative modelling approaches, such as variational autoencoders, generative adversarial networks, and recently flow and diffusion models, have enabled the de novo design of molecules with desired properties[47]. One prominent example of generative modelling in drug discovery is REINVENT[48], a deep learning-based de novo design tool. REINVENT uses reinforcement learning to generate novel molecules with desired properties, iteratively improving its output based on predefined scoring functions[48].

In my thesis, I focus on understanding and developing robust machine learning methods for predicting protein-ligand binding affinity. Binding affinity quantifies the strength of the interactions between target proteins and ligands. Being able to identify molecules that bind well to the desired protein target is an essential first step in finding *hits*. In addition, when improving a *hit* to a *lead* compound, it is crucial not to lose affinity while optimizing other properties. While experimental methods are considered the gold standard, they are time-consuming and costly[9]. Computational methods provide a promising, cost-effective alternative for prioritizing candidates for experimental validation[49–51]. Many approaches exist; from structure-based methods comprising MD simulations, free energy calculations, and docking to more recent machine learning-based approaches for predicting binding affinity. Each of these classes of methods has their own strengths and limitations. For example, MD simulations and free energy calculations provide accurate estimates but are computationally intensive and time-consuming, limiting their applicability to large-scale screening and restricting them to the lead optimization stage[50,52]. Conversely, molecular docking methods, particularly giga-docking methods, are commonly used for screening ultra-large compound libraries during the hit discovery phase by offering faster predictions but often lack sufficient accuracy and are not reliable for ranking these

compound libraries[51,53]. Recent machine learning methods predict binding affinity directly from protein and ligand features, potentially replacing conventional scoring functions. However, these methods face generalizability issues, meaning their predictive accuracy often decreases significantly when applied to chemical structures or biological targets different from those included in their training data, limiting their widespread integration into drug discovery workflows[44,54,55]. These methods are discussed in detail later in the thesis.

My thesis focuses on exploring and developing machine learning methods tailored towards two primary objectives - (a) understanding how recent deep learning models learn and predict protein-ligand binding affinity and how to make these models useful for screening large libraries during hit discovery, and (b) using machine learning to support reliable and computationally-expensive physics-based methods such as alchemical free energy calculations, in order to reduce computational costs and time in identifying top binders during the lead optimization stage.

In **Chapter 2**, I provide a comprehensive background on drug discovery stages, protein-ligand binding affinity measures, and the computational methodologies used to predict protein-ligand binding affinity. The chapter begins by detailing the various stages of the drug discovery pipeline, starting from target discovery and validation and exploring hit discovery and lead optimization stages. It also covers pre-clinical and clinical trials. Next, the **Chapter 3** introduces fundamental concepts related to protein-ligand binding affinity and discusses how these affinities are quantified using experimental and computational techniques. I then delve into the different protein and ligand representations used in computational models. I discuss various computational approaches for binding affinity prediction, including structure-based methods such as alchemical free energy methods based on molecular dynamics simulations and docking and ligand-based methods such as QSAR. Finally, I discuss various machine learning methods for binding affinity prediction, datasets for training these models, and evaluation strategies.

In **Chapter 4** and **Chapter 5**, I explore the use of deep learning models for

understanding and predicting protein-ligand binding affinity. **Chapter 4** focuses on investigating how state-of-the-art deep learning frameworks up to 2022 can directly learn binding interactions from protein and ligand data to predict binding affinity. This chapter motivates the need for better and more robust machine learning models that generalize to unseen targets or drugs, a crucial aspect for accurate prediction during early drug discovery stages. **Chapter 5** takes ideas from large language model approaches to improve on state-of-the-art architectures. The new approaches' utility is then applied to practical screening scenarios, particularly during hit discovery. This chapter addresses three key challenges with the current deep learning models at model, data, and evaluation levels. At the model level, this chapter introduces the Binding Affinity Language Model (BALM) for predicting binding affinity using pretrained protein and ligand language models. At the data and evaluation level, this chapter shows how to train and test these models in a way that truly understands how they compare to traditional methods or experimental affinities. The work in these chapters aligns with the first objective of my thesis — to establish deep learning-based frameworks that enhance the efficiency and accuracy of binding affinity predictions, providing significant value in the context of screening large compound libraries.

In **Chapter 6**, I shift my focus towards the second objective, i.e., using machine learning to support physics-based methods, such as free energy calculations, for binding affinity predictions during the lead optimization stage. This chapter explores how integrating machine learning models can navigate larger ligand libraries to broad to be explored in a cost effective manner using either alchemical free energy methods or even experiments. Finally, **Chapter 7** summarizes the key findings of my thesis and discusses the broader implications of my work. I also outline potential future directions, emphasizing the need for further integration of machine learning and physics-based models to tackle the challenges in drug discovery.

Chapter 2

Navigating the Drug Discovery Landscape

As discussed in Chapter 1, drug discovery is a complex, expensive, and multi-step process aimed at identifying novel therapeutic agents that can effectively target diseases and improve patient outcomes. From identifying a protein target for the disease of interest to preclinical and clinical testing, each stage presents unique challenges and opportunities that shape the development of a successful drug. This section provides an overview of the key phases in the drug discovery pipeline, highlighting the foundational concepts and methodologies that are instrumental in developing effective and safe therapeutic candidates.

2.1 Target Discovery and Validation

Identifying and validating biological targets is an initial step in the drug discovery process, significantly influencing the probability of success at every stage of drug development. Target discovery involves identifying a biological component, most commonly a protein, that a drug can target[56]. However, other molecules such as polysaccharides, lipids, and nucleic acids can also serve as drug targets, depending on the therapeutic approach[56]. In this thesis, I will focus specifically on proteins as the primary targets in drug discovery. I will begin discussing the target discovery phase

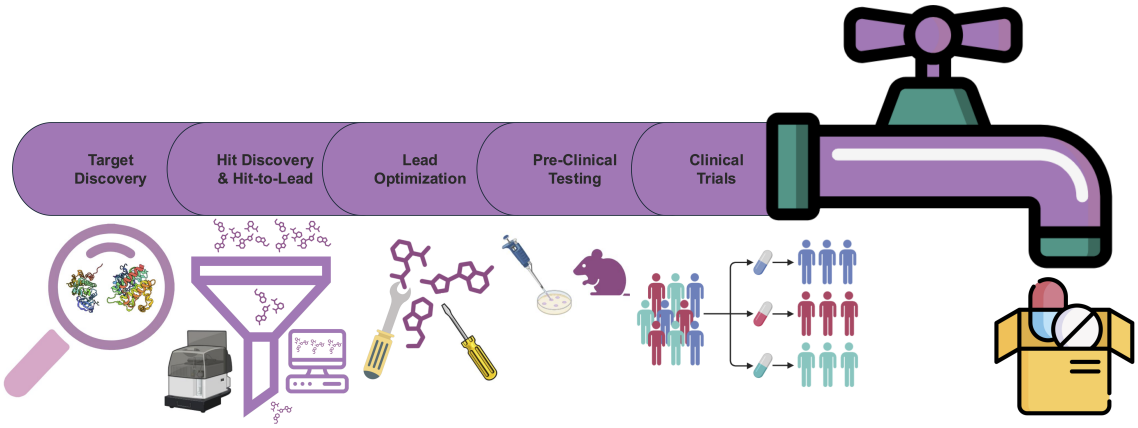


Figure 2.1: **Overview of the drug discovery pipeline, depicting the key stages.** The process initiates with *Target Discovery*, where potential disease-related protein targets are identified and validated. In the *Hit Discovery* and *Hit-to-Lead* stages, compounds with initial activity against the target are identified and refined to enhance binding and drug-like properties. During *Lead Optimization*, these compounds undergo further modifications to optimize their efficacy, selectivity, and other essential properties. Successful candidates proceed to *Preclinical Testing*, involving in vitro and in vivo experiments to evaluate safety and pharmacokinetics, before advancing to *Clinical Trials*. Clinical trials are conducted in phases to assess safety, efficacy, and long-term effects in humans, ultimately leading to regulatory approval and market release if successful.

by providing an overview of proteins and how they are determined experimentally, followed by commonly used approaches for target discovery and validation.

Overview of Proteins – Proteins are fundamental biological molecules that play critical roles in living organisms' biological processes[57]. They are essential for life and have a wide range of functions in cells and organisms. These functions include acting as enzymes to catalyze biochemical reactions, providing structural support and maintaining cell shape, transporting molecules within cells or throughout the body (e.g., haemoglobin for oxygen transport), regulating physiological processes as hormones (e.g., insulin), and defending against foreign substances as antibodies in the immune system[58].

Proteins are made up of one or more linear chains of amino acids called polypeptides[59]. There are 20 standard amino acids used by living organisms to build proteins, each with a unique side chain that determines its chemical properties. Amino acids have a basic structure comprising a central carbon atom (also known as the α carbon), bonded to an amino group (NH_2), a carboxyl group (COOH), and a

hydrogen atom. Every amino acid also has another atom or group of atoms bonded to the central atom, known as the R group, which determines the identity of the amino acid[57,59]. The chemical properties and sequence of the amino acids play a critical role in determining the structure and function of the polypeptide and the protein it ultimately forms a part of. These amino acids are linked together by peptide bonds, forming the primary structure of a protein. A polypeptide chain has directionality, meaning it has two distinct ends. One end called the amino terminus (N-terminus), has a free amino group, while the other end, known as the carboxyl terminus (C-terminus), has a free carboxyl group[57,59]. As shown in Figure. 2.2, the structure of a protein can be described at four different levels as follows.

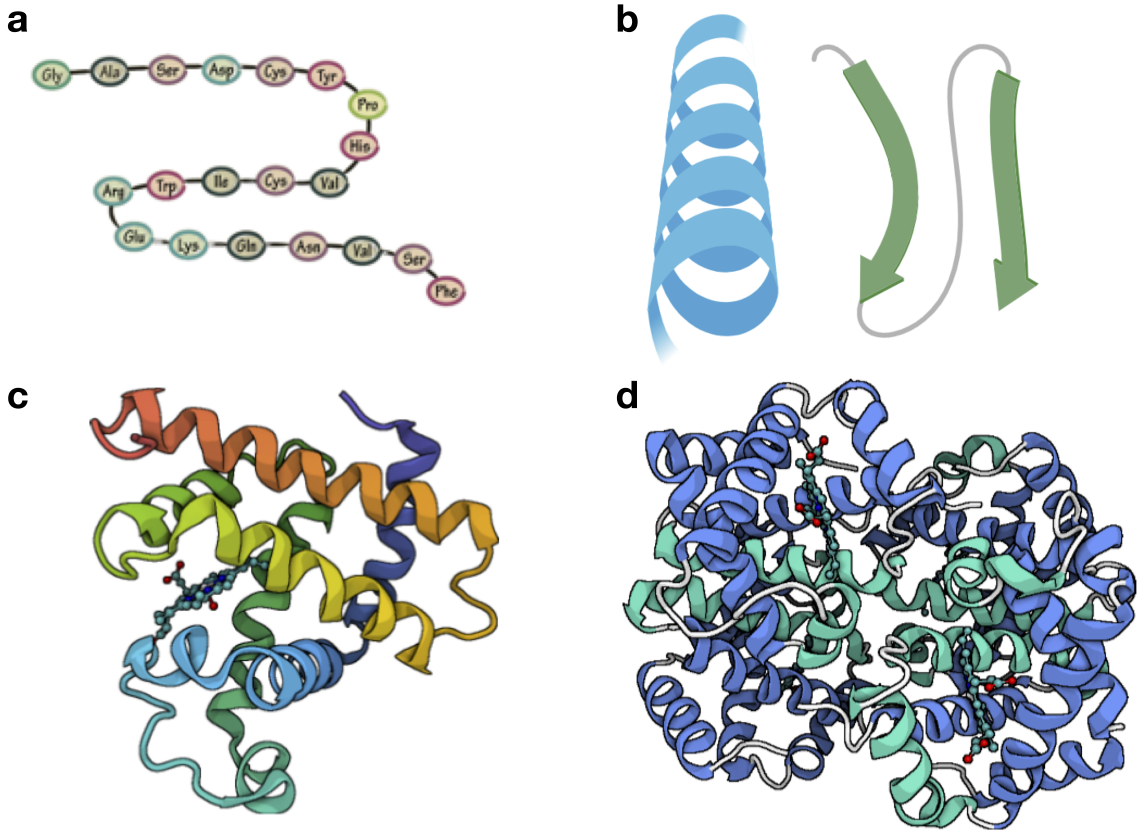


Figure 2.2: **Hierarchical representation of protein structures.** (a) The primary structure represents the linear sequence of amino acids in a polypeptide chain. (b) The secondary structure shows local folded motifs, with the α-helix depicted in blue and the β-sheet in green. These structures are stabilized by hydrogen bonds along the protein backbone. (c) The tertiary structure displays the complete three-dimensional folding of a single polypeptide chain, illustrated here by myoglobin, with interactions between side chains creating a unique, stable conformation. (d) The quaternary structure represents the assembly of multiple polypeptide chains into a functional protein complex, as exemplified by haemoglobin.

Primary structure is the linear sequence of amino acids in a polypeptide chain[57,59]. This sequence is determined by the Deoxyribonucleic Acid (DNA) of the gene encoding the protein and dictates all subsequent levels of protein structure[57,59].

Secondary structure refers to local folded structures within a polypeptide formed due to interactions between atoms of the backbone[57,59]. The two most common types of secondary structures are the α-helix and the β-pleated sheet, which is stabilized by hydrogen bonds between the carbonyl oxygen of one amino acid and the amino hydrogen of another[57,59]. Many proteins contain both α helices and β pleated sheets, though some contain just one type of secondary structure or do not form either type[57,59].

Tertiary structure is the overall three-dimensional shape of a single polypeptide chain, resulting from interactions between the R groups of the amino acids[57,59]. These interactions include hydrogen bonding, ionic bonding, van der Waals interactions, and disulfide bridges. The folding process is complex, with proteins adopting various conformations before settling into their final, energetically favourable form[57,59].

Quaternary structure is formed when multiple polypeptide chains, known as subunits, form a functional protein complex[57,59]. The arrangement and interactions of the subunits determine the protein's final structure and functionality.

It is also important to note that while most proteins adopt a set of stable, natively biologically active folds, there exist *intrinsically disordered proteins* (IDPs) or regions within proteins that are often promiscuous in binding and instead exist as flexible ensembles of conformations[60]. These disordered regions are flexible and can adopt multiple conformations, allowing them to participate in diverse interactions and functions that structured regions cannot[60]. For more details on IDPs, you can refer to this review[60].

Experimental methods for determining protein structures – The study of protein structure and function has been greatly advanced by techniques such as X-ray crystallography[61], nuclear magnetic resonance (NMR) spectroscopy[62], or

cryo-electron microscopy[63] (cryo-EM).

X-ray crystallography[61] is the most widely used technique for determining the 3D structures of proteins. It works by crystallizing a protein and then bombarding the crystal with X-rays. The diffraction pattern produced is used to calculate the electron density map, from which the atomic positions of the protein can be inferred[64]. X-ray crystallography offers high-resolution structures, providing detailed atomic information about proteins, ligands, inhibitors, ions, and other molecules incorporated into the crystal[61]. While it has determined roughly 90% of known protein structures, X-ray crystallography faces limitations[64]. The crystallization process can be difficult and imposes constraints on the types of proteins that can be studied. It works best for rigid proteins that form ordered crystals, while flexible proteins are more challenging as crystallography requires many molecules aligned in exactly the same orientation[61,64].

NMR spectroscopy[62] is a powerful method for determining protein structures in solution, under near-physiological conditions. In this technique, proteins placed in a strong magnetic field are exposed to radio-frequency pulses, and the resultant resonance signals from atomic nuclei (typically hydrogen, carbon, and nitrogen) are recorded[62,65]. NMR spectroscopy provides valuable insights into protein flexibility, dynamics, and interactions through the generation of an ensemble of structures, reflecting various conformations adopted by the protein in solution[62]. However, its applicability is primarily limited to relatively small and medium-sized proteins (generally below 40 kilo Daltons), as larger proteins or complexes tend to produce overlapping signals and experience reduced sensitivity and resolution[65].

Cryo-EM[63] is a relatively recent but rapidly advancing method for structural determination, particularly beneficial for large protein complexes and membrane proteins. Cryo-EM avoids the crystallization step by rapidly freezing protein samples in vitreous ice and imaging them using an electron microscope[63]. The resulting two-dimensional images are computationally reconstructed into 3D structures. While cryo-EM excels at resolving large protein complexes, it struggles to achieve high

resolution for smaller proteins (typically below 50 kilo Daltons) due to low signal-to-noise ratios, making image reconstruction difficult[66]. This is because small molecular weight proteins have low sample contrast, and many lack unique structural features needed for accurate single-particle image alignment[66].

Protein Data Bank (PDB)[67,68] is the primary repository for experimentally determined 3D protein structures. It contains over 225,946 entries (as of October 2024), providing a rich source of structural data for computational studies. PDB files store atomic coordinates, enabling precise modelling of protein structures. Each PDB file contains detailed information about the position of every atom in the protein, the secondary structure elements (such as α-helices and β-sheets), and any ligands or cofactors bound to the protein. This data can be used to generate 3D visualizations and perform in silico experiments, such as molecular docking or molecular dynamics simulations.

Target identification is the first step in drug discovery, where molecular targets responsible for disease mechanisms are identified. Target identification methods can be categorized into three broad approaches - experimental, multiomic, and computational approaches[69–72]. *Experimental methods* involve wet-lab experiments to identify targets based on affinity, genetic modification screening, and comparative profiling. Techniques such as affinity-based biochemical assays, stable isotope labelling by amino acids in cell culture (SILAC), and chemical/genetic screening using RNA interference (RNAi) or CRISPR-Cas9 gene editing have been instrumental[71,72]. CRISPR technology, in particular, has dramatically expanded our understanding of the mechanistic and pharmacological aspects of human diseases, identifying pivotal targets such as Bromodomain-containing protein 2 (BRD2) in the host response to SARS-CoV-2 infection[73]. *Multiomic approaches* provide interconnected molecular information from various perspectives, including genomics, transcriptomics, proteomics, epigenomics, and metabolomics. Genomics, the most established omics discipline, focuses on genetic variants in DNA sequences[70]. Large-scale genome-wide association studies (GWAS) have identified numerous associations between

genetic variants and complex diseases, leading to breakthrough therapies for cystic fibrosis and inflammatory bowel disease[70,74].

Computational techniques for target identification have progressed from early structure-based virtual screens to recent machine learning and deep learning frameworks[75,76]. Structure-based virtual screens, such as reverse docking[77], inverts the conventional docking paradigm by screening a single compound against a database of protein structures to identify potential binding targets. Reverse docking has been instrumental in elucidating the polypharmacology of drugs, identification of off-target effects, and uncovering novel therapeutic targets[77]. ML methods integrate heterogeneous data sources (such as multi-omics profiles, phenotypic screens, and chemical–biological interaction data) and model complex networks to predict novel targets that would be difficult to identify using traditional techniques[75]. For example, deep learning-based methods have identified potential therapeutic targets for diseases such as amyotrophic lateral sclerosis (ALS) by analyzing disease-specific multiomic and text-based data[69,71]. Furthermore, emerging AI-driven pipelines emphasize causal inference—leveraging human genetic and functional genomic data to focus on targets that are true drivers of disease—and consider target reversibility (evidence that modulating the target can reverse disease phenotypes). Equally important is the early prediction of target druggability, to ensure that identified candidates have tractable binding sites for therapeutic modulation[76]. In addition, literature mining is increasingly used to construct knowledge graphs linking genes, diseases, and compounds, thereby providing a rich context for target discovery that complements experimental data[78]. Finally, these data-driven approaches are complemented by breakthroughs in structural biology such as AlphaFold2, which now enables protein structure predictions to inform target druggability assessments and structure-based drug design[45,79].

Over the past decades, advancements in molecular biology have unveiled numerous biomacromolecules that are pivotal in disease progression, thereby presenting promising targets for drug discovery[69,71]. These biomacromolecules, which include

kinases, receptors, and channel proteins, are often closely linked to disease development. Their structures frequently possess specific binding sites for ligands, making them "druggable" [56]. However, some disease-related proteins do not exhibit such characteristics and are thus termed "undruggable" [56].

Target validation is essential to ensure that the target is indeed implicated in the disease mechanism and suitable for therapeutic intervention. Standard methods used for target validation include *genetic validation*, where techniques such as CRISPR-Cas9 and RNAi are employed to knock out or knock down gene expression, helping to assess the biological role of the target. If genetic manipulation correlates with disease progression, the target is considered validated [73]. *Pharmacological validation* involves the use of small-molecule inhibitors or antibodies to modulate the target's activity, and observing the effects of these pharmacological agents helps determine the target's significance in the disease process [21]. *Animal models*, such as genetically modified animals such as knockout mice, provide *in vivo* platforms for validating targets and offer more physiologically relevant data to confirm whether modulating the target affects disease [80]. *Biomarker studies* involve correlating the modulation of the target with clinical outcomes or changes in disease-related biomarkers to validate the target's relevance in human disease [81]. Finally, omics integration, is another approach that combines multi-omics data (e.g., genomics, proteomics, metabolomics), enables understanding of the target's role in cellular networks and disease pathways [74].

2.2 Hit Discovery

Once a protein target has been identified and validated, the next step is to discover molecules that can modulate the protein's activity. Hit discovery is a fundamental step in the early stages of drug discovery, where the goal is identifying compounds, referred to as hits, that demonstrate activity against the target protein. A compound is referred to as a hit if it shows sufficient binding affinity in an experimental or predicted screen [82]. The most straightforward approach is to search public

databases, including patent literature, to find compounds known to bind to a similar target[82–84]. If the target is well established, compounds with activity against it will often have been reported. However, in many cases, there is no existing data, and alternative methods are necessary. Some commonly used techniques for hit discovery are discussed below.

High-throughput screening (HTS) is one of the widely used screening techniques for identifying hits[85,86]. HTS involves screening a large compound library against a therapeutic target *in vitro*. HTS allows for the testing of hundreds of thousands to tens of millions of molecules to establish their bioactivity against the target[85]. Biochemical and cell-based assays are commonly used assay techniques. Biochemical assays are performed using purified target proteins and are designed to measure direct interactions between compounds and their molecular targets. These assays often utilize detection methods such as fluorescence polarization, time-resolved Förster resonance energy transfer, or mass spectrometry to quantify binding affinities or enzymatic activities[85,86]. The controlled environment of biochemical assays enables high sensitivity and specificity, making them particularly suitable for elucidating mechanisms of action at the molecular level[82,87]. However, they may not fully capture the complexity of biological systems, as they lack the cellular context necessary to assess factors such as membrane permeability or metabolic stability[82,86,88]. In contrast, cell-based assays evaluate the effects of compounds within living cells, providing a more physiologically relevant context. These assays can measure a wide range of cellular responses, including viability, proliferation, apoptosis, or specific pathway activation, often using reporter genes or high-content imaging[21,89]. Recent advances in cell-based HTS include the use of three-dimensional cell cultures and patient-derived organoids, which better mimic tissue architecture and function, thereby improving the predictive value of screening results for in vivo efficacy and toxicity[89,90]. While cell-based assays offer greater biological relevance, they are generally more complex to optimize and may have lower throughput compared to biochemical assays due to increased variability and

the need for more sophisticated analysis[82,87,89].

Fragment-based drug discovery (FBDD) has emerged as a valuable approach for identifying hit compounds for challenging targets[91]. FBDD focuses on using smaller, drug-like compounds (molecular weight < 300 Daltons) to identify fragments that may bind weakly to the target but form high-quality interactions. These fragments often exhibit high ligand efficiency, defined as the binding affinity per non-hydrogen atom, allowing for the identification of efficient binders that can be optimized into potent leads[92]. Compared to traditional HTS, fragment screens yield higher hit rates[93], and these fragment hits can be optimized to develop high-affinity leads[94]. Although fragment hits may initially have lower potency, they can be expanded or combined to create high-affinity leads[94].

Virtual screening involves using computational methods to sift through large compound libraries and identify potential hits[83,84]. These methods are broadly categorized into structure-based and ligand-based approaches. Structure-based virtual screening uses three-dimensional protein structures to predict ligand binding through molecular docking and scoring functions, while ligand-based methods utilize known active compounds to identify structurally or pharmacophorically similar molecules, which are molecules that share similar pharmacophoric features and are likely to have similar biological activities[84?] . In Chapter 3.3, a detailed discussion on various structure-based and ligand-based approaches are provided.

Ultra-large virtual screening (ULVS) takes this further by screening over 100 million ligands computationally, offering significant time and cost efficiencies compared to experimental HTS[83,84]. Public, commercial, and proprietary ultra-large ligand libraries enable exploration beyond the compounds currently available commercially. Examples of these libraries include Enamine's REAL Database[95], CHEMriya[96], ZINC Libraries[97], GalaXi[98], and eXplore[99]. Generated Databases (GDBs), such as GDB-11, GDB-13, and GDB-17, provide access to billions of molecules[100–102]. However, the major disadvantage of these ultra-large libraries is that the compounds often cannot be readily purchased in a time and cost-efficient manner, though cus-

tom synthesis may be possible for many[82,83].

2.3 Hit-to-Lead and Lead Optimization

Once hits are identified, they undergo a process of optimization to improve their potency, selectivity, and physicochemical properties[21]. Medicinal chemistry efforts play a key role in this optimization process, as chemists modify the molecular structure of the hits to improve their binding affinity and specificity towards the target protein. The transition from a hit to a lead compound is a critical stage in drug discovery, as it lays the foundation for developing potential therapeutic agents. The hit-to-lead stage involves further investigation of individual hits or series of hit molecules from the discovery stage. Optimized hits with improved properties are selected as lead compounds for further development, involving rigorous testing in cell-based assays and animal models to evaluate their efficacy and safety[103].

At this stage, additional desirable features of a small molecule are considered, such as maintaining oral bioavailability or crossing the blood-brain barrier if the target is located in the brain. An essential aspect of this optimization phase is the evaluation of Absorption, Distribution, Metabolism, and Excretion (ADME) properties[104]. ADME studies are needed to ensure that the lead compounds have favourable pharmacokinetic profiles, meaning they are well-absorbed, appropriately distributed throughout the body, metabolically stable, and efficiently excreted. Compounds with poor ADME properties may be ineffective or cause adverse effects, making these evaluations vital for selecting promising lead candidates. This stage is typically undertaken by a diverse project team, including medicinal chemists, through a sequence of iterative modifications forming a "design, make, test, analyze" cycle[105]. The synthesis of novel chemical matter is challenging, and the feasibility of synthesizing a chemical structure often informs the design process[106].

There are several main paradigms for molecular optimization, including ligand-based, structure-based, and fragment-based approaches. Ligand-based methods optimize molecular features using data from other chemical structures and fundamental

chemical principles. These methods rely on QSAR models and molecular similarity algorithms to predict how modifications might improve drug efficacy[43]. Structure-based design utilizes 3D structural information of protein-ligand binding to rationally introduce functionality to improve existing interactions or find novel ones[107]. The required structural information can be experimentally gathered (e.g., via X-ray crystallography[108]) or computationally generated (e.g., via docking[107]). Fragment-based approaches rely on structural data and involve screening small molecular weight compounds (fragments) to identify low potency but high-quality leads. Techniques for advancing fragment hits are still developing, with fragment library design receiving substantial attention[109,110]. Here challenges revolve around preserving the fragment interaction while elaborating them through linking and merging to drug molecules. Successfully optimized leads are presented as preclinical candidates for further *in vivo* work[111]. Typically, only one or two preclinical candidates are advanced past the lead optimization stage.

2.4 Preclinical Trials

Preclinical trials act as a link between laboratory findings and human clinical trials. These studies are designed to assess the safety, effectiveness, and potential toxicity of new drug candidates before testing them in humans. The duration of preclinical trials can vary significantly depending on the complexity of the studies and the specific requirements of the drug candidate. Generally, preclinical trials can take several months to a few years to complete[112].

The preclinical stage usually begins after a promising compound has been identified through hit discovery and lead optimization processes. This phase involves a series of *in vitro* (test tube or cell culture) and *in vivo* (animal) studies aimed at characterizing the pharmacological properties of the drug candidate and assessing its potential risks[113]. The main objective of preclinical research is to gather enough data to support the start of clinical trials while minimizing risks to humans.

***In Vitro* studies** are the first step in preclinical research. These experiments

are conducted in controlled laboratory conditions using cell cultures, tissue samples, or biochemical assays. They provide initial insights into the drug's mechanism of action, its interaction with the target, and potential off-target effects[114]. *In vitro* studies help assess the drug's stability, solubility, and other physicochemical properties required for its development as a therapeutic agent.

***In Vivo* studies** are the next stage, which involves testing the drug candidate in animal models[115]. These studies are essential for understanding how the drug behaves in a living organism. They provide valuable information on the drug's absorption, distribution, metabolism, and excretion (ADME) properties, as well as its potential therapeutic effects and toxicity profile. Researchers typically use multiple animal species to ensure a comprehensive assessment of the drug's effects across different biological systems.

Pharmacokinetic (PK) studies are a critical component of preclinical research. These studies investigate how the body processes the drug, including its ADME. PK studies help researchers determine appropriate dosing regimens and predict how the drug might behave in humans. They also provide insights into potential drug-drug interactions and the need for dose adjustments in specific patient populations[116].

Pharmacodynamic (PD) studies focus on the drug's effects on the body. These experiments aim to establish the relationship between the drug's concentration at the site of action and its therapeutic effects. PD studies help researchers understand the drug's mechanism of action, its potency, and the duration of its effects[117]. This information helps in optimizing dosing strategies and predicting the drug's efficacy in humans.

Toxicology studies assess the potential adverse effects of the drug candidate, including acute toxicity, chronic toxicity, and specific organ toxicities[118]. Researchers conduct extensive toxicology studies to determine the maximum tolerated dose, identify potential side effects, and establish safety margins. This information is critical for designing safe and ethical clinical trials.

As the preclinical stage progresses, researchers conduct specialized studies to address specific concerns or regulatory requirements[119]. These may include genotoxicity studies to assess the drug's potential to cause genetic mutations, reproductive toxicity studies to evaluate effects on fertility and fetal development, and carcinogenicity studies to determine long-term cancer risks[119]. Throughout the preclinical stage, researchers must adhere to Good Laboratory Practice[120] (GLP) guidelines to ensure the quality and integrity of the data generated. These standards help maintain consistency and reliability across different studies and laboratories, which is needed for regulatory approval. The final step in the preclinical stage involves compiling and analyzing all the data generated from these studies. Researchers prepare comprehensive reports that summarize the drug's pharmacological properties, safety profile, and potential risks. This information forms the basis of the *Investigational New Drug*[121] (IND) application, which is submitted to regulatory authorities to obtain approval for initiating clinical trials in humans.

2.5 Clinical Trials

Clinical trials are the final phase in the drug development process, following preclinical studies. They are designed to evaluate the safety, efficacy, and overall therapeutic value of new drug candidates in humans. Clinical trials are conducted in several phases with distinct objectives and methodologies to ensure a comprehensive assessment of the drug's potential benefits and risks. The entire clinical trial process can take several years to complete. Phase I trials typically last several months, Phase II trials take several months to two years, and Phase III trials span one to four years[122].

Designing a clinical trial involves developing a detailed study plan and a protocol that outlines the research questions, objectives, and methodologies. Key elements of a clinical trial protocol include defining the selection criteria for participants, determining the sample size needed to achieve statistically significant results, establishing the duration of the study and follow-up periods, including control groups (placebo

or standard treatment) for comparison, specifying the dosing regimen, planning assessments and data collection methods, and finally analyzing the data to draw meaningful conclusions[123].

Phase I (Safety and dosage) clinical trials primarily focus on assessing the safety, tolerability, and pharmacokinetics of the drug. Typically, Phase I trials involve a small group of 20 to 100 healthy volunteers or patients with the target condition. The primary objectives are to determine the optimal dosage range, identify any dose-limiting toxicities, and understand how the drug is absorbed, distributed, metabolized, and excreted by the body[124]. During Phase I, researchers closely monitor participants for any adverse effects and gather preliminary data on the drug's pharmacodynamics. This phase helps establish the foundation for subsequent trials by identifying safe dosage levels and potential side effects[124].

Phase II (Efficacy and side effects) trials aim to evaluate the drug's efficacy and further assess its safety profile. These trials involve a larger group of participants, typically ranging from 100 to 300 individuals who have the condition the drug is intended to treat[125]. Phase II is often divided into Phase IIa (exploratory) and Phase IIb (confirmatory) studies. In Phase IIa, researchers explore the drug's efficacy and optimal dosing regimen, while Phase IIb focuses on confirming these findings in a larger cohort. The primary goals are to determine the drug's therapeutic effect, identify any short-term side effects, and refine the dosing strategy[125]. Successful Phase II trials provide critical data that support the transition to larger-scale studies in Phase III.

Phase III (Confirmation and comparison) trials[126] are large-scale studies designed to confirm the drug's efficacy, monitor side effects, and compare it to standard treatments or placebos. These trials typically involve 300 to 3,000 participants and are conducted across multiple sites to ensure diverse and representative data. The primary objectives of Phase III trials are to demonstrate the drug's therapeutic benefits, establish its safety profile in a broader population, and gather comprehensive data to support regulatory approval. Researchers use randomized, controlled

trial designs to minimize bias and ensure robust results[126]. Phase III trials are critical for determining the risk-benefit ratio of the drug and are often the final step before seeking regulatory approval.

Phase IV Post-marketing surveillance - Phase IV clinical trials[127], also known as post-marketing surveillance, are conducted after the drug has been approved and marketed. These trials involve thousands of participants and aim to monitor the drug's long-term safety and effectiveness in the general population. Phase IV studies help identify any rare or long-term adverse effects that may not have been detected in earlier phases and provide valuable data for optimizing treatment guidelines and improving patient outcomes[127].

Chapter 3

Computational Approaches for Estimating Protein-ligand Binding Affinity

3.1 Experiments to Measure Binding Affinity and Datasets

Drugs are designed to bind to a protein target to induce therapeutic effects by modulating the target's function[52,128]. Binding affinity (BA) is a measure of the strength of the binding interaction between a protein target (P) and a ligand molecule (L). BA is measured to screen a large number of potential drugs (i.e., ligands) for their efficacy against an identified disease target (i.e., protein) in the initial phase of drug discovery, guiding researchers towards those drugs that may exhibit better efficacy[52,129].

As shown in Figure. 3.1, the binding of a ligand (L) to a protein (P) can be described as a reaction where the reactants are the protein (P) and the ligand (L) in their free (unbound) forms and the product is the protein-ligand complex (PL)

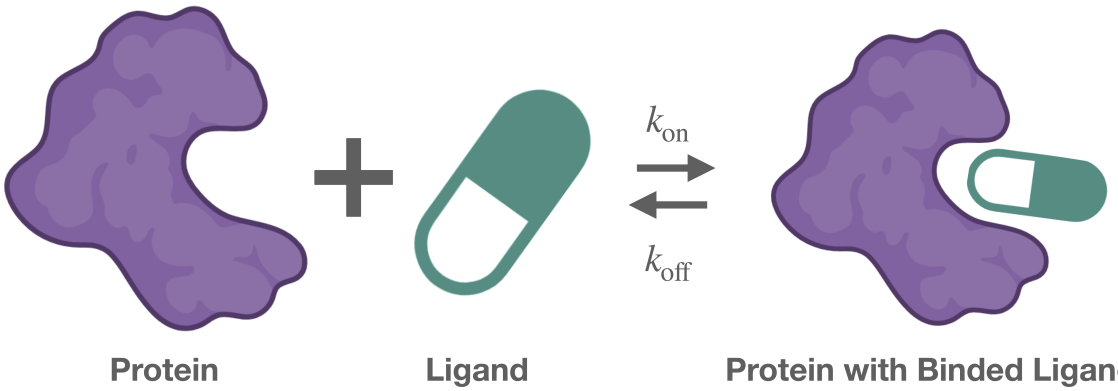


Figure 3.1: **Depicting the protein-ligand binding process.** This schematic demonstrates the binding interaction between a protein (left) and a ligand (middle), leading to the formation of a protein-ligand complex (right). The binding reaction is characterized by two rate constants: k_{on}, the association rate constant, which represents the formation of the protein-ligand complex, and k_{off}, the dissociation rate constant, indicating the disassociation of the complex back into free protein and ligand. The equilibrium dissociation constant K_d, calculated as k_{off}/k_{on}, provides a quantitative measure of binding affinity, where lower values correspond to stronger binding interactions.

formed when the ligand is bound to the protein[129,130],

$$P + L \underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} PL, \tag{3.1}$$

where k_{on} and k_{off} are the forward and reverse rate constants, respectively. The dissociation constant (K_d) at equilibrium is given by,

$$K_d = \frac{k_{off}}{k_{on}} = \frac{[P][L]}{[PL]}, \tag{3.2}$$

where $[P]$ is the concentration of free protein, $[L]$ is the concentration of free ligand, and $[PL]$ is the concentration of the protein-ligand complex at equilibrium[52,128]. K_d expresses the concentration of ligand L required to occupy 50% of the receptor P population at equilibrium. Smaller values of K_d indicate higher affinity, as less ligand is needed to occupy 50% of the receptor population at equilibrium[52,128]. These values are typically measured in micromolar (μM) or nanomolar (nM) concentrations and are often converted to logarithmic scales such as pK_d (where $pK_d = -\log_{10}(K_d)$) for ease of comparison and use in computational algorithms[131].

Binding affinity can be expressed through the standard Gibbs free energy change ($\Delta G^{\circ}_{\text{bind}}$), which describes the thermodynamic favorability of the binding interaction[52,128]. The relationship between $\Delta G^{\circ}_{\text{bind}}$ and the dissociation constant (K_{d}) is given by,

$$\Delta G^{\circ}_{\text{bind}} = -k_{\text{B}} T \ln K^{\circ}_{\text{d}}, \tag{3.3}$$

where k_B is the Boltzmann constant, T is the temperature in Kelvin, and K°_{d} is the dissociation constant divided by the standard state concentration. The standard state concentration is determined by the reference state and is typically defined as 1 mol/L under constant pressure of 1 atm[52]. A negative ΔG° indicates a thermodynamically favourable binding interaction. Gibbs free energy can also be described as,

$$\Delta G^{\circ}_{\text{bind}} = \Delta H^{\circ} - T\Delta S^{\circ}, \tag{3.4}$$

where ΔH° is the change in enthalpy, and ΔS° is the change in standard state entropy. Several non-covalent interactions, including hydrogen bonds, van der Waals forces, and electrostatic interactions, contribute to the enthalpic component (ΔH°) of binding free energy[132]. The entropic component (ΔS°) reflects both conformational flexibility and solvation effects. Although ligand binding generally decreases conformational entropy by increasing the rigidity of the ligand and protein binding site, favourable changes in solvation entropy due to partial desolvation of the ligand can enhance binding affinity[133]. Additionally, the hydrophobic effect, which drives non-polar groups to cluster in aqueous environments, contributes to $\Delta G^{\circ}_{\text{bind}}$ by releasing water molecules from the binding interface, thereby increasing entropy[133].

In addition to K_{d}, other metrics such as the inhibitor constant (K_{i}) and the half-maximal inhibitory concentration (IC_{50}) are commonly used to compare the relative potency of ligands. IC_{50} is the concentration needed to induce half-maximum inhibition. The IC_{50} value depends on the concentration of the target and ligand, and low IC_{50} and K_{i} indicate higher affinity. In the discovery process, as the focus is on determining whether one drug is more effective than another, the ligand typically

competes with either a probe or substrate in these experiments. For competitive inhibitor assays, IC_{50} can be related to the binding affinity of the inhibitor K_i via the Cheng-Prusoff equation[134]

$$K_i = \frac{IC_{50}}{1 + \frac{[S]}{K_m}}, \tag{3.5}$$

where K_m is the Michaelis constant, and $[S]$ is the experimental substrate concentration, and this equation applies specifically to competitive inhibition. We can link $K_i \approx K_d$ if we assume that all protein-ligand complex formations lead to efficient protein inhibition. The substrate concentration used in many experiments is $[S] = K_m$. To estimate IC_{50} values, experimental dose-response data are typically fitted using a non-linear regression model[135]. This approach yields a sigmoidal curve from which the IC_{50} can be directly inferred as the concentration at which 50% inhibition is achieved[135]. Analysis of these dose-response curves yields important insights into inhibitor potency and efficacy. For example, figure 3.2 shows an example of such curves for two hypothetical inhibitors. The inhibitor with a lower IC_{50} value exhibits higher potency, as it requires a lower concentration to elicit the same inhibitory effect.

Several experimental techniques are commonly used to determine binding affinities and kinetic parameters discussed earlier, each with varying degrees of complexity, sensitivity, and throughput. Surface Plasmon Resonance (SPR) stands as the gold standard for detailed binding characterization, enabling direct measurement of k_{on}, k_{off}, and K_d through label-free measurements with high sensitivity across a broad affinity range (pM to mM)[136]. In SPR, changes in refractive index near a sensor surface are measured as analytes flow over immobilized ligands, providing real-time monitoring of binding kinetics and determination of association and dissociation rate constants[136].

For high-throughput screening applications, Microscale Thermophoresis (MST) offers the rapid analysis of molecular interactions and determination of K_d values[137]. While MST requires fluorescent labelling and may be susceptible to artefacts, it

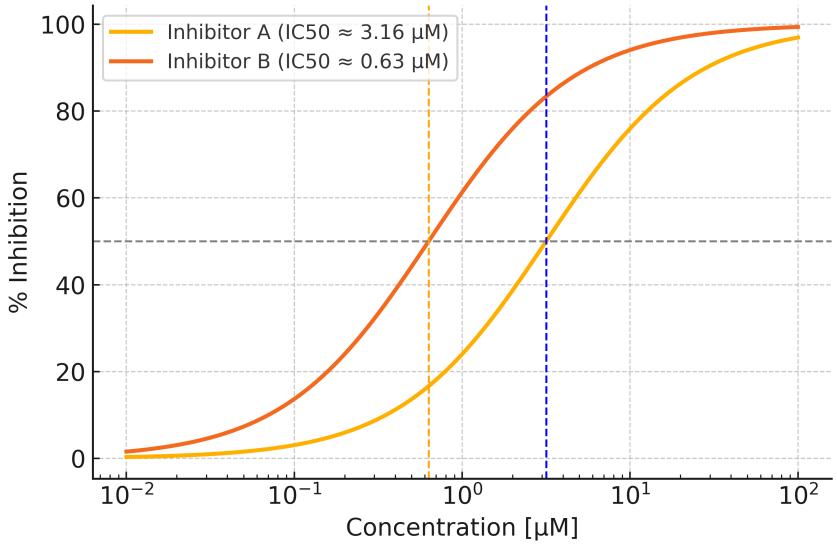


Figure 3.2: **Representative dose-response curves for two inhibitors, illustrating IC_{50} estimation.** The percentage inhibition is plotted against increasing inhibitor concentration on a logarithmic scale. Inhibitor B achieves 50% inhibition at a lower concentration ($IC_{50} \approx 0.63$ µM) compared to Inhibitor A ($IC_{50} \approx 3.16$ µM), indicating that Inhibitor B is more potent. The IC_{50} value corresponds to the concentration at which the response is halfway between the minimum and maximum asymptotes (dashed lines).

enables measurement of hundreds of interactions per day compared to the more detailed but lower-throughput SPR analysis. MST measures the directed movement of molecules along temperature gradients, which is altered upon binding, allowing for quantitative binding assessments in solution. For thermodynamic profiling, Isothermal Titration Calorimetry (ITC) is a robust, label-free, low-throughput method that measures the heat exchanged during binding, yielding binding constants along with enthalpic and entropic contributions, though it has a more limited affinity range (nM to µM) and lacks detailed real-time kinetic analysis[138]. Additional techniques ranging from label-free to fluorescence-based methods are also widely employed for measuring various binding parameters, including kinetic constants, thermodynamic properties, and structural characteristics[138–142].

3.1.1 Datasets with Experimental Binding Affinity

Accurate and diverse datasets are crucial for developing and validating computational approaches such as machine learning. In this section, I focus on datasets that

are publicly available and widely adopted in the literature for training and testing both Structure-based (3D complexes) and Simplified Molecular Input Line Entry System (SMILES) and Sequence-based (1D) methods. SMILES notation is a string-based representation of molecular structures[143], and the details of SMILES notation will be discussed later in this chapter. These selected datasets have been commonly used in machine learning literature for predicting binding affinities as a regression task and are relevant to this thesis. For a broader overview of available datasets and comparative discussions, refer to recent survey articles[44,144–146].

Sequence-based and SMILES (1D) datasets focus on the linear representations of proteins and ligands, such as protein sequences or ligand SMILES strings. These datasets are often used to build ligand-based models such as Quantitative Structure-Activity Relationship (QSAR) models and deep learning models that predict properties such as binding affinity using sequence and/or SMILES-based features.

BindingDB[147] is a comprehensive resource containing experimentally determined binding affinities for protein-ligand interactions, including K_{d}, K_{i}, and IC_{50} experimental values. With over 1.2 million interactions, BindingDB is particularly useful for developing ligand-based predictive models. However, the combination of bioactivity data from different sources (e.g., IC_{50}, K_{d}, K_{i}) introduces significant data noise and inconsistencies, complicating the integration of data for training robust models[148]. *ChEMBL*[149] is another extensive bioactivity database containing more than 20 million bioactivity measurements, making it a valuable resource for building deep learning models.

Kinase inhibitor bioactivity (KIBA)[150] integrates bioactivity data from multiple sources, including IC_{50}, K_{d}, and K_{i}, to create a uniform representation of kinase inhibitor bioactivity (KIBA) score[151]. Lower KIBA scores indicate stronger binding affinities. The KIBA score can be defined based on K_{d} or K_{i}, or the average of them, depending on the availability of the bioactivity types[150] using the equation

below

$$\begin{aligned}
\text{KIBA} &= \frac{\text{IC}_{50}}{1 + H_{\text{i}}\left(\text{IC}_{50}/K_{\text{i}}\right)} && \text{if } K_{\text{i}} \text{ and } IC_{50} \text{ are present} \\
&= \frac{\text{IC}_{50}}{1 + H_{\text{d}}\left(\text{IC}_{50}/K_{\text{d}}\right)} && \text{if } K_{\text{d}} \text{ and } IC_{50} \text{ are present} \\
&= \left(\frac{\text{IC}_{50}}{1 + H_{\text{i}}\left(\text{IC}_{50}/K_{\text{i}}\right)} + \frac{\text{IC}_{50}}{1 + H_{\text{d}}\left(\text{IC}_{50}/K_{\text{d}}\right)}\right)/2 && \text{if } K_{\text{i}}, K_{\text{d}} \text{ and } IC_{50} \text{ are present,}
\end{aligned} \tag{3.6}$$

where H_i and H_d are the parameters that determine the weights of IC_{50} in the model-based adjustments for K_i and K_d. Initially, the dataset included 467 targets and 52,498 small molecules. However, He et al.[152] refined the dataset by retaining only those small molecules and targets with at least 10 observations. As a result, the refined dataset consists of 229 unique proteins and 2,111 unique ligands, making it a more manageable and consistent dataset for training predictive models. Although the KIBA score mitigates inconsistencies between different experimental setups, the integration of diverse data sources introduces uncertainty that impacts the overall accuracy of predictive models, as highlighted by Landrum et al.[148].

Davis[153] dataset provides kinase inhibition profiles for 72 inhibitors tested against 442 kinase targets, comprising 30,746 K_{d} values. This dataset is particularly useful for evaluating ML models in kinase-related drug discovery tasks. Therapeutic Data Commons (TDC)[154] offers a collection of datasets specifically designed for therapeutic applications, including protein-ligand binding affinity datasets, such as BindingDB, KIBA, and Davis datasets.

Complex-based (3D) datasets provide structural information on 3D protein-ligand complexes, which include atomic coordinates of binding modes.

PDBbind[155] is one of the most widely used datasets for structure-based modelling. It provides detailed information on protein-ligand complexes derived from the Protein Data Bank (PDB) and includes experimental binding affinities such as K_{d}, K_{i}, and IC_{50} values. The PDBbind dataset is typically divided into three subsets: the general set (13,285 complexes), the refined set (4,057 complexes), and the core set (290 complexes). The refined set is often used for model training, while the core set serves as a high-quality benchmark for evaluating model performance.

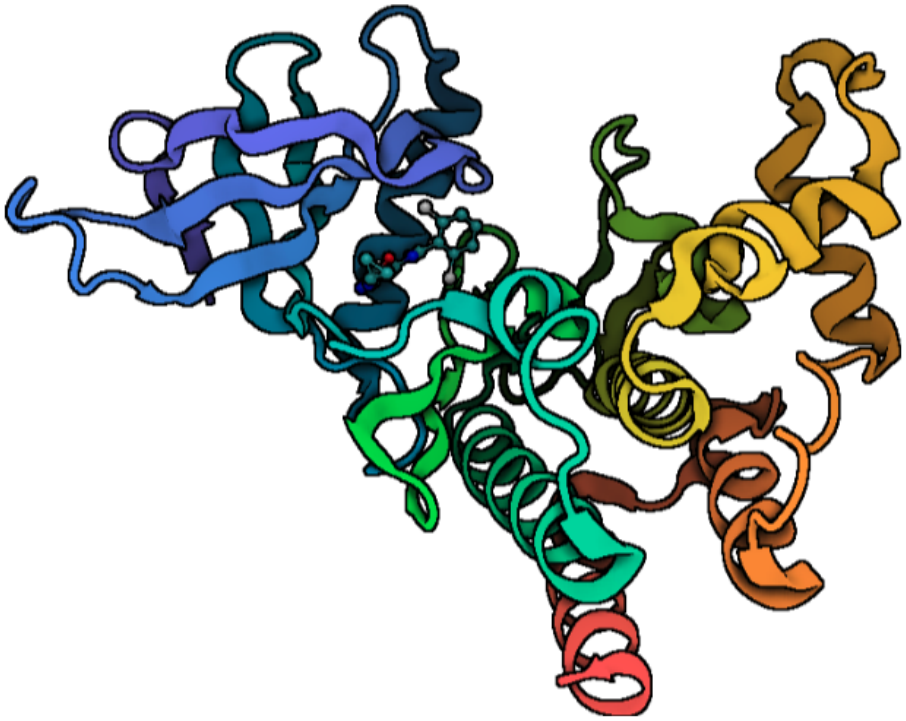


Figure 3.3: **Protein-ligand complex of Tyrosine Kinase 2 with inhibitor.** This structure depicts the Tyrosine Kinase 2 protein bound to its inhibitor (*PDB ID - 4GIH*). The protein is visualized as a ribbon diagram, with secondary structure elements such as α-helices, β-sheets, and loops shown in different colours. The bound inhibitor is represented in stick form within the active site.

Challenges with PDBbind include potential data leakage due to overlapping chemical structures and inconsistencies in data splits, which can impact generalizability.

Leak Proof PDBbind[156] is a reorganized version of the PDBbind[155] that addresses data leakage issues. By restructuring the dataset and implementing similarity-based splitting strategies, leak-proof PDBBind offers a more reliable benchmark for binding affinity prediction models. This version has been reorganized to reduce redundancy in sequence, interaction patterns, and chemical similarity across the training, validation, and test splits. The dataset has been further refined by removing covalently bound ligand-protein complexes, ligands containing rare atomic elements, and entries with structural issues such as steric clashes while ensuring consistency in the reported binding free energies. The dataset has been categorized into three "Clean Levels" based on the quality of the structures - CL1 (14,324 entries), CL2 (7,985 entries), and CL3 (4,404 entries). Li et al.[156] suggested CL1 for training due to its balance of size and quality, whereas CL2, considered more reliable, is recommended for validation and testing.

Binding Mother of All Databases[157,158] *(Binding MOAD)* has been in development for over 20 years and now includes 41,409 protein-ligand complexes, with experimentally determined binding affinity data available for 15,223 complexes (37%). While Binding MOAD and PDBbind share the same primary data source and a similar mission, Binding MOAD distinguishes itself by including only valid protein-ligand complexes with crystal structures at a resolution of 2.5 Å or better. Unlike PDBbind, Binding MOAD also includes complexes without binding data, offering a broader scope but requiring rigorous curation to account for variations in experimental conditions and data quality.

MISATO[159] is a recently introduced dataset that enhances PDBbind by integrating quantum mechanical (QM) properties and molecular dynamics (MD) simulations. Building upon approximately 20,000 experimental protein–ligand complexes, sourced from PDBbind, MISATO applies semi-empirical QM methods to refine ligand geometries and correct common structural issues such as protonation

states and atom assignments. Additionally, it provides over 170 microseconds of MD simulation data in explicit water, capturing the dynamic behavior of protein–ligand interactions. This combination of static and dynamic data (19,443 QM and 16,972 MD simulations) offers a richer representation of molecular interactions, facilitating the development of machine learning models.

3.2 Representations for Proteins and Ligands

Accurate representations of proteins and ligands capturing their features are needed for using computational methods in drug discovery. These representations allow computational methods to analyze, simulate, and predict the behaviour of biological molecules. This section covers the methods used to represent proteins and ligands in computational studies. Figure. 3.4 demonstrates proteins and ligands in their 1D, 2D and 3D forms.

3.2.1 Protein Representations

Proteins can be represented using a variety of formats that capture their structural, sequence, and functional properties. In Chapter 2, I have discussed how protein structures can be described at four different levels, from primary to quaternary structure. These representations are needed for computational methods such as molecular docking, virtual screening, and protein-protein interaction modelling. This subsection explores the common representations used for proteins in computational studies, broadly categorized as sequence-based and structure-based.

Sequence-based Representations Proteins are composed of linear chains of amino acids, and their sequences can be represented as strings where each letter corresponds to a specific amino acid. The primary sequence of a protein is typically written using the one-letter code (e.g., “MKVLY...”), where each character corresponds to one of the 20 naturally occurring amino acids (e.g., “M-K-V” represents methionine, lysine, and valine). This sequence representation serves as the founda-

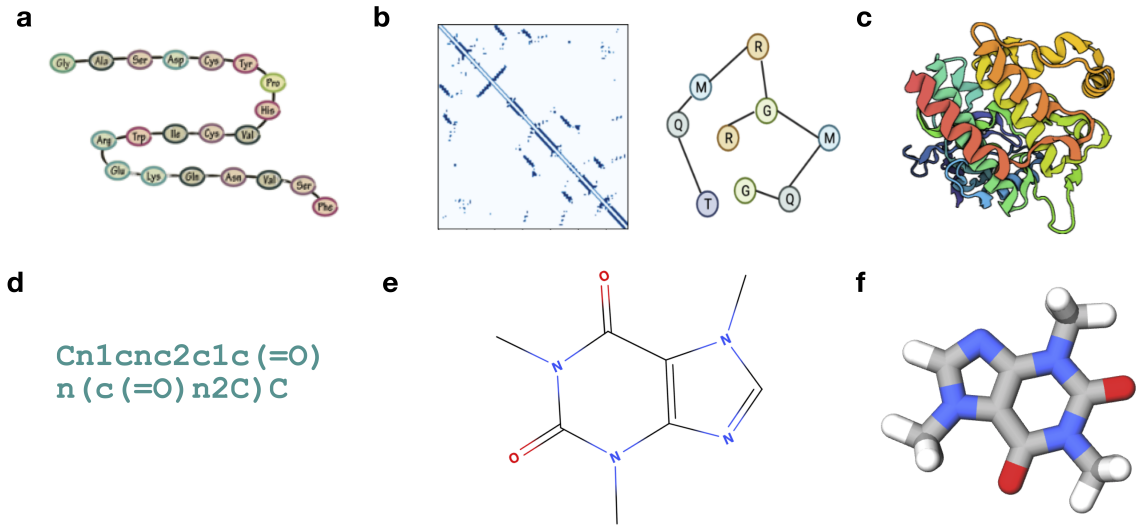


Figure 3.4: **1D, 2D and 3D representations of proteins and ligands.** Row 1 illustrates representations of proteins - (a) 1D sequence-based representation, showing a linear sequence of amino acids; (b) 2D contact map, capturing interactions between residues, useful for understanding protein folding and structure. These contact maps are represented as a graph where nodes are residues and edges represent interactions between a pair of residues. (c) 3D protein structure showing the arrangement of secondary structural elements such as helices and sheets, which can be represented as a grid, graph or surface-based representation. Row 2 provides examples of the molecule caffeine in different forms - (d) 1D SMILES representation encoding the molecular structure as a string, (e) 2D structural representation, displaying atomic connectivity and bond types, and (f) 3D structural representation, showing the spatial arrangement of atoms.

tion for numerous computational analyses and is often used as input for alignment algorithms, sequence-based similarity searches (e.g., BLAST[160]), and machine learning models for predicting secondary structure or other protein properties. Sequence information is needed for understanding evolutionary relationships, performing sequence homology modelling, and annotating functional regions within proteins. In computational models, these sequences are further processed into various numerical or learned representations that capture properties such as evolutionary conservation, physicochemical attributes, or semantic features learned from large-scale protein sequence datasets.

Traditional representations incorporate evolutionary and domain information through methods such as Hidden Markov Models (HMMs)[161,162] and Position-Specific Scoring Matrices (PSSMs), which capture conservation patterns and functional mo-

tifs in protein sequences. Another classic approach is Z-scale descriptors, developed in the 1980s, which utilize principal component analysis (PCA) on physicochemical properties of amino acids to create numerical feature representations[163]. More recent advancements leverage deep learning, particularly transformer-based models that learn protein language directly from sequences. Models such as Evolutionary Scale Modeling (ESM)[164,165] and ProtBERT[166] are trained on large protein datasets such as UniRef, which contains over 200 million sequences[167], to learn rich representations for downstream applications. These models extract complex features for downstream tasks and will be discussed further in Section 3.4.

Structure-based Representations As discussed earlier, structural data are obtained through experimental techniques such as X-ray crystallography[61], nuclear magnetic resonance (NMR) spectroscopy[62], and cryo-electron microscopy (cryo-EM)[63]. The most fundamental representation of a protein structure is through atomic coordinates. In PDB files, the 3D coordinates (x, y, z) of each atom are listed, providing a straightforward description of the molecule's structure[67]. This coordinate data forms the foundation for constructing 3D models, calculating distances between atoms, and analyzing molecular interactions. To facilitate computational analyses, these 3D structures can be transformed into various representations that capture specific features of the protein structure.

Protein contact maps (PCMs) are widely used as simplified representations of the 3D structure of proteins and provide insights into protein folding, dynamics, and interactions[168–172]. A PCM is a 2D matrix that encodes the spatial proximity between amino acid residues, where each element (i, j) in the matrix is set to 1 if the $C\alpha$ atoms of residues i and j are within a predefined cutoff distance d_c, and 0 otherwise. Mathematically, the contact map matrix A_{ij}^{struc} is defined as:

$$A_{ij}^{\text{struc}} = \begin{cases} 1, & \text{if } d_{i,j} \leq d_c \text{ and } i \neq j, \\ 0, & \text{if } d_{i,j} > d_c \text{ or } i = j, \end{cases} \tag{3.7}$$

where $d_{i,j}$ is the Euclidean distance between residues i and j. The cutoff distance

d_c is typically set to 8 Å, but it may vary depending on the study[172]. These matrices are used to generate protein graphs $G_p = (N_p, M_p)$, where N_p represents the set of nodes corresponding to amino acid residues and M_p represents the set of edges, which define connections or 'contacts' between residues. These contact maps could be represented as *graphs* where nodes correspond to residues, and edges represent interactions between a given pair of residues[173,174]. *Distance matrices* extend the concept of contact maps by capturing the actual Euclidean distance between each pair of residues, providing more quantitative spatial information to represent 3D structures.

Grid-based representations use an Euclidean 3D grid (or voxel grid) to discretize the protein structure. Each voxel in the grid represents a small region of 3D space, which may contain information about the atomic presence, density, or physicochemical properties[175–177]. Grids often include empty voxels representing unoccupied space, increasing computational costs, especially at high resolutions[178].

Graph-based representations provide a flexible, non-Euclidean representation of protein structures, where nodes represent atoms, residues, or even secondary structure elements, and edges denote spatial or chemical relationships[178–181]. This format allows for encoding complex topologies, where nodes can have a variable number of neighbours based on spatial distance or chemical bonding. Depending on the level of granularity, nodes may represent only backbone atoms or entire residues, while edges can carry additional properties, such as bond type or distance[178].

Surface-based representations capture the protein's solvent-accessible surface (SAS). SAS are needed in molecular interactions, such as protein-ligand and protein-protein binding[175]. These representations can be visualized in two main forms: *meshes* and *point clouds*. Meshes are constructed from a collection of polygons, typically triangles, that define the boundary between solvent-accessible and inaccessible regions of the protein[182,183]. Meshes can be characterized by their geometric and physicochemical features, making them particularly useful for identifying binding and interaction sites. Point clouds, on the other hand, represent the pro-

tein surface as a set of discrete points, where each point describes a small area of the surface[178]. This approach captures surface topology and can be used to model protein interactions, especially in machine learning applications[178].

3.2.2 Ligand Representations

For ligands, here considering only small drug-like molecules, can be represented in computational studies through two broad categories of methods, string-based notations (1D/2D) and 3D structural representations, each capturing different aspects of their structure and properties.

Simplified Molecular Input Line Entry System (SMILES) is one of the most widely used notations that encodes a molecule's structure as a linear string of characters, as discussed earlier. Developed in the 1980s by David Weininger[143], SMILES captures essential molecular information, including atoms, bonds, rings, aromaticity, branching, stereochemistry, and isotopes. This encoding enables SMILES strings to be converted into 2D graphs, where nodes represent atoms and edges represent bonds, providing a structured view of the molecular architecture[143].

SMILES strings are generated using a depth-first search (DFS) on a molecular graph, represented as $G = (V, E)$, where V is a set of labelled vertices (atoms), and E is a set of labelled edges (bonds)[143]. During DFS traversal, each atom is printed upon the first encounter, while hydrogen atoms are typically omitted for brevity. The DFS algorithm explores each branch of the molecular tree as far as possible before backtracking to the root node (atom)[143]. Branching in a SMILES string is denoted by the symbols '(' and ')', respectively (Figure. 3.5a). For example, the SMILES string for 2-methyl-2-propanol is `CC(C)(C)O`. Single and double bonds are represented by the symbols '-' and '=', respectively, although the symbol for a single bond is typically omitted. For instance, in the SMILES string for propenal, `C=CC=O`, the double bonds are explicitly denoted by '='[143].

Rings in a molecule must be broken at an arbitrary point to create an acyclic tree structure. The broken bond is labelled with a closure number[143]. For instance,

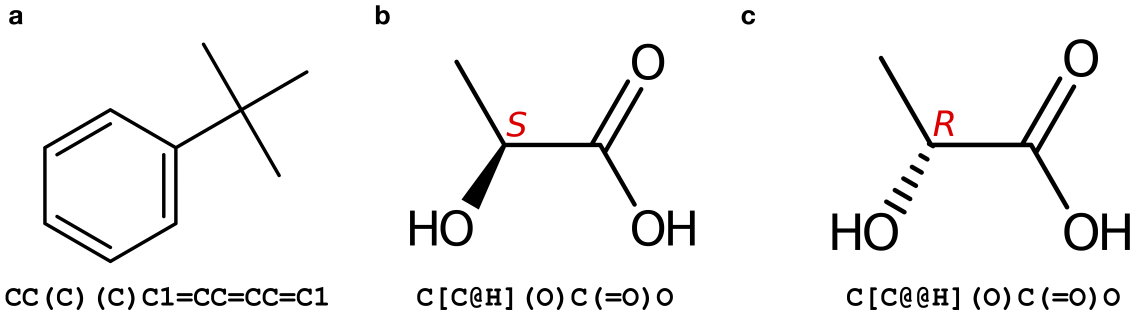


Figure 3.5: **SMILES representation of molecular structures.** (a) This SMILES string represents tert-butylbenzene - `CC(C)(C)C1=CC=CC=C1`. The Depth-first search traversal starts at the main chain (`C`), then branches into `C(C)(C)` before exploring the aromatic ring (`C1=CC=CC=C1`). The branching at the central carbon atom is evident, connecting to two terminal methyl groups (`C(C)(C)`) and a benzene group. The single bonds in the main chain are implied, while the alternating single and double bonds in the aromatic ring represent its aromaticity. The ring closure is indicated by the digit `1`, connecting the start and end of the ring. (b) and (c) show the SMILES strings for the two enantiomers of lactic acid, highlighting stereochemical differences. (b) D-lactic acid is represented as `C[C@H](O)C(=O)O`, where the `@` symbol indicates the clockwise (R) configuration of the chiral carbon. (c) L-lactic acid is represented as `C[C@@H](O)C(=O)O`, where the `@@` symbol indicates the counterclockwise (S) configuration of the chiral carbon.

the SMILES string for a cyclopentane molecule is `C1CCCC1`. For molecules with multiple rings, the closure number increases for each ring, such as in bicyclo heptane `C1CC2CCC1C2`. Aromatic rings are indicated by alternating double bonds. In some cases, atoms belonging to an aromatic ring are represented by lowercase letters to denote aromaticity[143]. In such cases, bonds between aromatic atoms are not explicitly written. Two valid depictions of an aromatic pyridine molecule are `C1=CC=NC=C1` and `c1ccncc1`.

SMILES also supports stereochemistry, allowing the representation of *isomers*, molecules with the same chemical formula but different spatial arrangements[143,184]. A subtype of isomers, called *enantiomers* (or chiral isomers), includes molecules that are non-superimposable mirror images of each other[143,184]. Stereochemistry is represented by appending '@' symbols to indicate the orientation of chiral centres. For instance, lactic acid exists as two enantiomers: the D-isomer is written as `C[C@H](O)C(=O)O`, and the L-isomer as `C[C@@H](O)C(=O)O`, with '@' and '@@' denoting the distinct configurations around the chiral carbon (Figure. 3.5b and Fig-

ure. 3.5c).

SMILES *canonicalization* is the process by which each unique molecular graph is assigned exactly one SMILES string, eliminating the ambiguity that arises in standard SMILES representations[143,184]. In standard SMILES notation, a molecule can be described in multiple ways depending on the order in which atoms and bonds are traversed during string generation. For example, ethanol may be written as `CCO`, `OCC`, or `C(O)C`, all of which are valid but non-unique representations. Canonicalization addresses this ambiguity by applying a series of deterministic rules that assign unique rankings to atoms based on their properties, such as atomic number, connectivity, and formal charge[143]. The canonicalization algorithm typically begins by assigning priority scores to each atom in the molecule using these atomic invariants. The molecule is then traversed, often using a depth-first search approach, starting from the highest-ranked atom, and ties are broken by considering bond order and the ranks of neighbouring atoms[184]. This systematic approach ensures that the same traversal path and, ultimately, the same SMILES string are generated for a given molecular structure regardless of the initial input or atom ordering. For ethanol, this process yields the canonical SMILES string `CCO`, providing a consistent representation across various databases. This process simplifies tasks such as molecule retrieval, comparison, and database management by avoiding redundancy and preventing duplicate entries.

Featurizing SMILES refers to the process of converting SMILES strings into numerical feature vectors that capture molecular information, enabling their use in computational methods such as machine learning and deep learning algorithms. One of the most commonly used approaches is through *fingerprints*, which encode ligands as fixed-length continuous or binary vectors. In binary fingerprints, each bit represents the presence (1) or absence (0) of a specific substructure within the molecule. *Extended-Connectivity Fingerprints (ECFPs)*, also known as *Morgan Fingerprints*, are among the most widely used fingerprints due to their ability to capture detailed molecular substructures. ECFPs are generated by iteratively expanding circular

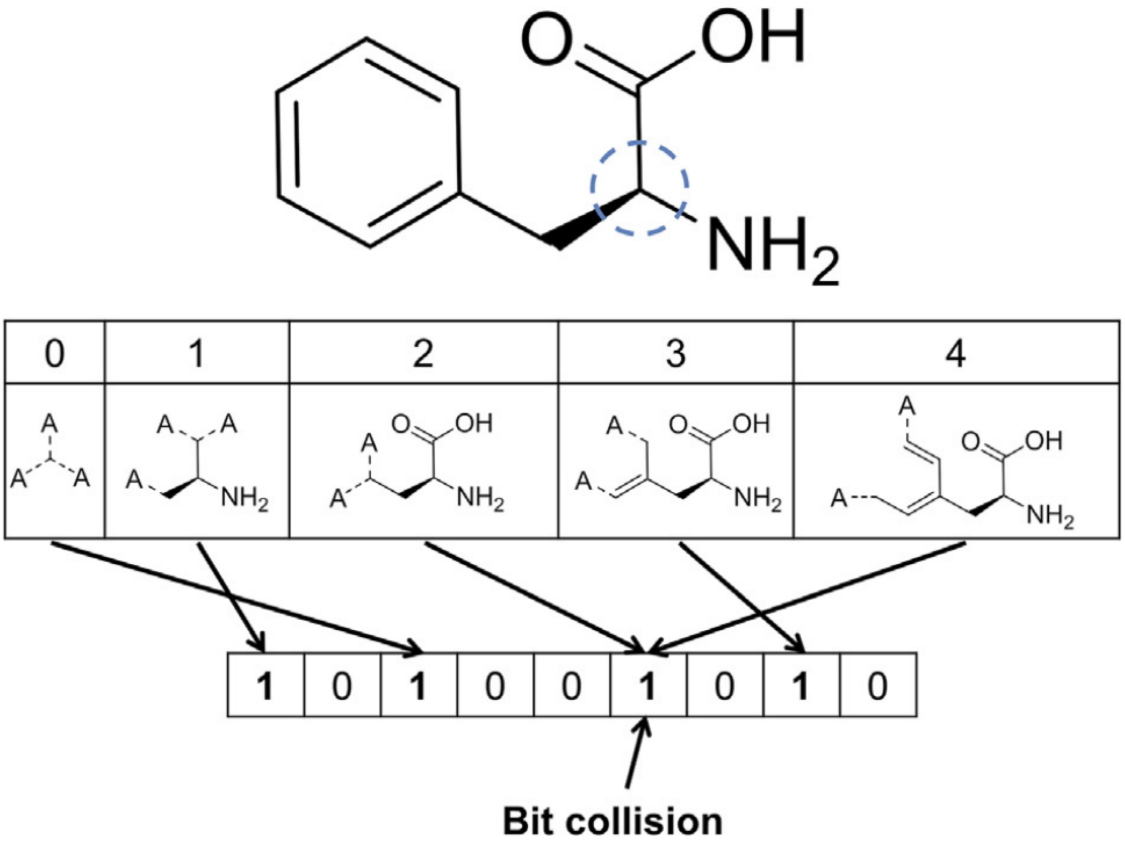


Figure 3.6: **Featurizing SMILES using ECFPs.** Extended Connectivity Fingerprints encode atomic environments by iteratively expanding circular substructures around each heavy atom up to a predefined radius (here, 4 bonds). At each step, the atom's local environment is hashed to generate a unique identifier, with terminal atoms incorporating additional "any atom" (A) bonds that are not part of the fragment. In this example, the central atom is the α-carbon. The hashed identifiers are mapped to bits in a fixed-length binary vector, where collisions may occur when multiple fragments are assigned to the same bit, as illustrated here with the overlap of multiple substructures. Adapted from Bajusz et al.[185]

substructures around each atom up to a specified radius. Each circular substructure is assigned a unique identifier through hashing, resulting in a compact binary vector that captures complex connectivity patterns within the molecule[185,186]. Beyond ECFPs, there are several other molecular fingerprints for featurizing SMILES[187–190].

In addition to these substructure-based fingerprints, *physicochemical descriptors* provide global molecular features that directly quantify chemical properties critical for predicting bioavailability and pharmacokinetic behavior. Molecular weight (< 500 Da) governs passive diffusion through lipid membranes, with excessive values often reducing oral bioavailability as per Lipinski's Rule of Five[191]. The octanol-water partition coefficient (LogP), a measure of lipophilicity, and its pH-dependent counterpart (LogD) predict membrane permeability, with values < 5 favouring solubility and absorption[192]. Hydrogen bonding capacity, quantified through donor (< 5) and acceptor (< 10) counts, influences both solubility and transporter-mediated uptake mechanisms[191,193]. The topological polar surface area (TPSA), calculated from polar atom contributions, correlates inversely with intestinal absorption, where values $< 140\text{Å}^2$ indicate favourable membrane penetration[194].

Another common featurization approach is *graph-based representations*, where SMILES strings are converted into graphs with atoms as nodes and bonds as edges[44,174]. Molecular graphs are typically encoded using a *Feature Matrix* (X), which provides per-atom information such as atomic type and degree with dimensions $N \times D$, where N is the number of atoms and D the number of features[174], and a *Connectivity Matrix*, which describes molecular structure via node connections, represented as an adjacency matrix A of size $N \times N$, or in coordinate (COO) format with dimensions $2 \times E$, where E is the number of edges[174,180]. Recent advancements in machine learning have also introduced *machine-learned featurizers*, which use natural language processing (NLP) techniques to encode ligand features directly from SMILES strings. Models such as *Word2Vec*[195] and *transformers*[196] treat SMILES strings as character sequences, learning embeddings that capture the underlying chemical semantics. Notable models include *SMILES2Vec*[197], *Mol2Vec*[198], *Chem-*

BERTa[199], and the *SMILES Transformer*[200], which have shown promise in tasks such as molecular property prediction and virtual screening by generating feature vectors that capture both local and global chemical properties of the molecule. The machine-learned featurizers are discussed further in Chapter 3.4.

3D Structural Representations of ligands is needed for understanding their interactions with biological targets. Ligands are represented in 3D space using atomic coordinates derived either from experimental data (e.g., X-ray crystallography, NMR spectroscopy) or computational modelling (e.g., molecular dynamics simulations, quantum mechanics calculations). This 3D representation is fundamental for structure-based studies, such as molecular docking, where the fit of a ligand within a protein's active site is evaluated.

Structure data file (SDF) is a widely-used format for storing information about ligands, including detailed atomic coordinates and bonding information that are required for capturing spatial relationships within molecules. Each SDF file typically contains several essential elements - *3D or 2D Atomic Coordinates* (x, y, z) of each atom, which specify the spatial arrangement of atoms in the molecule; *Atom and Bond Information*, including atomic symbols (e.g., C, O, N), bond types (single, double, triple, aromatic), and atomic properties such as charge. *Molecular Properties* in the form of key-value pairs, including calculated descriptors such as molecular weight, hydrogen bond donors and acceptors, and other physical or chemical attributes. This structural data can be transformed into various computational representations - *graph-based* approaches where atoms are nodes and bonds are edges[201], *distance matrices* capturing pairwise atomic distances[202], and *voxel grids* that divide 3D space into regions encoding atomic information[203].

3.3 Conventional Approaches for Estimating Protein-Ligand Binding Affinity

3.3.1 Structure-based Approaches

Structure-based methods make use of 3D structures of proteins and ligands to quantify interaction strengths. They can provide atomistic detail by highlighting favourable interactions between proteins and ligands. There are various methods for calculating binding free energies, and each plays a distinct role depending on the stage of the drug discovery process. These methods can be categorized into docking[204,205], molecular dynamics-based endpoint methods such as MM-PBSA/GBSA[206], and alchemical free energy[52] (AFE) methods. In the following, I will lay out the details of these methods and their role in typical drug discovery pipeline.

Docking

Docking provides two distinct types of information - (a) the preferred binding pose of a ligand with respect to its target protein, and (b) a score that quantifies the interaction strength between the ligand and the protein (Figure. 3.7). This process typically involves generating multiple ligand conformations and orientations (poses) and calculating their associated scores using docking programs. An example of a widely used docking software is AutoDock[207], which employs a genetic algorithm to efficiently explore the conformational space of ligands and identify optimal binding poses, while using a semi-empirical free energy force field that combines theoretical and experimental parameters to estimate the energetics of protein-ligand interactions. A typical scoring function used in docking is denoted as,

$$\text{Score} = \sum_{\text{hbonds}} E_{\text{hbond}} + \sum_{\text{vdw}} E_{\text{vdW}} + \sum_{\text{elec}} E_{\text{elec}}, \tag{3.8}$$

where each energy term captures distinct interactions contributing to the overall affinity[207–209]. *Hydrogen bond energy* (E_{hbond}) quantifies the strength of hydrogen bonds, which are directional interactions formed between a hydrogen atom covalently bound to an electronegative donor atom (such as nitrogen or oxygen) and another electronegative acceptor atom[207,208,210]. This term typically uses geometric criteria (bond distances and angles) combined with empirical parameters derived from quantum mechanical calculations or experimental data[208,210]. *Van der Waals energy* (E_{vdW}) accounts for short-range, non-specific interactions arising from induced dipoles between atoms[211]. *Electrostatic energy* (E_{elec}) represents Coulombic interactions between charged atoms of the ligand and protein[212]. For a detailed review of various scoring functions for docking, please refer to these articles[208,209]

As we have seen in the above scoring example, the scoring functions approximates molecular interactions using simplified empirical or semi-empirical force fields, prioritising computational efficiency over quantum mechanical rigor[207,209,213]. Thus, docking scores frequently correlate poorly with experimentally measured binding affinities (ΔG), mainly due to oversimplifications such as neglecting protein flexibility (rigid-body assumption), inadequate solvent modeling, and ignoring entropy effects[208,209]. Nevertheless, docking still remains a commonly used tool for initial virtual screening efforts.

MM-PBSA/GBSA

In MM-PBSA (Molecular Mechanics Poisson-Boltzmann Surface Area) and MM-GBSA (Molecular Mechanics Generalized Born Surface Area) methods, MD simulations are used to estimate binding free energies by calculating the energy of the bound and unbound states of the protein-ligand complex[206]. Molecular Dynamics (MD) simulations provide a computational method for predicting protein motions and interactions by simulating atomic movements based on Newtonian mechanics[27]. In molecular dynamics (MD) simulations, the force (F) acting on an atom or particle is a quantity that governs its motion[27]. According to Newton's second law, the

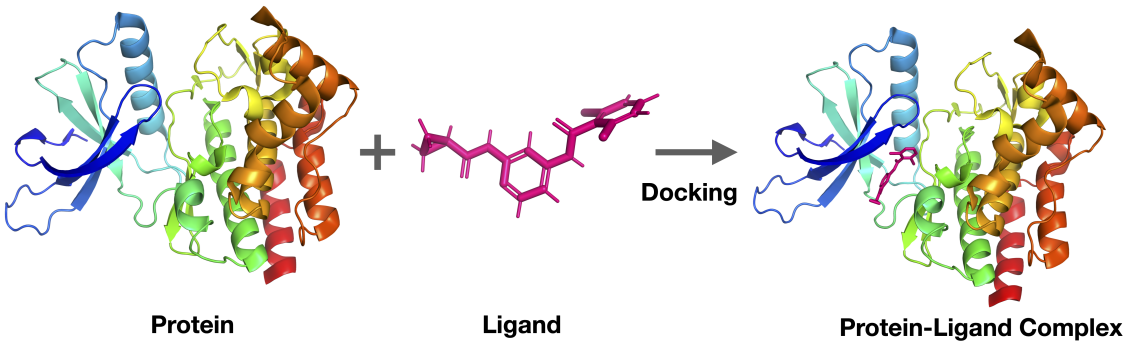


Figure 3.7: **Illustration of the docking process for Tyrosine Kinase 2 (TYK2) target with ligand.** This figure shows the TYK2 protein structure (left) with *(PDB ID: 4GIH)* and the ligand structure (middle) *CHEMBL2387224* before docking, followed by the protein-ligand complex (right) after docking. The docking process predicts the preferred orientation of the ligand within the protein's binding site by generating multiple ligand poses and scoring them based on interactions with the protein. The scoring function typically includes terms for hydrogen bonds, van der Waals interactions, and electrostatic interactions.

force is related to the acceleration of a particle as,

$$F = m\frac{d^2r}{dt^2}, \tag{3.9}$$

where m is the mass of the particle, and r is its position as a function of time (t)[27]. The forces between atoms in MD simulations are derived from the negative gradient of the potential energy (V) with respect to the atomic coordinates,

$$F = -\frac{dV}{dr}. \tag{3.10}$$

Here, V represents the total potential energy of the molecular system, which is approximated using a force field[27,214]. This approach allows to simulation of the dynamic behaviour of atoms and molecules by numerically solving the equations of motion, offering insights into structural flexibility, conformational changes, and intermolecular interactions[27]. The total potential energy (V) of a molecular system

in MD simulations is generally expressed as,

$$V = \sum_{\text{bonds}} K_r(r - r_{\text{eq}})^2 + \sum_{\text{angles}} K_\theta(\theta - \theta_{\text{eq}})^2 + \sum_{\text{dihedrals}} \frac{V_n}{2}\left[1 + \cos(n\phi - \gamma)\right] + \sum_{i<j}\left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}}\right]. \tag{3.11}$$

In this equation, the first three terms represent bonded interactions within covalently connected atoms, and the last term represents non-bonded interactions. The term $K_r(r - r_{\text{eq}})^2$ describes the bond stretching potential, where r is the bond length and r_{eq} is the equilibrium bond length[214]. The term $K_\theta(\theta - \theta_{\text{eq}})^2$ represents angle bending, where θ is the bond angle and θ_{eq} is its equilibrium value. The dihedral term $\frac{V_n}{2}[1+\cos(n\phi-\gamma)]$ captures the energy due to rotations around bonds, which are affected by the eclipsed and staggered conformations; here, ϕ is the dihedral angle[214]. The non-bonded interactions include van der Waals interactions, modelled by the Lennard-Jones potential with constants A_{ij} and B_{ij}, and electrostatic interactions, modelled by Coulomb's law, where q_i and q_j are the partial charges of atoms i and j, ϵ is the dielectric constant, and R_{ij} is the distance between atoms i and j[214].

MM-PBSA and MM-GBSA are both endpoint methods and require simulating only the endpoints (bound and unbound states) of the system of interest[206,215]. These methods estimate binding affinities by combining molecular mechanics (MM) energies with implicit solvation models[206]. The binding free energy ΔG_{bind} is given by,

$$\Delta G_{\text{bind}} = \Delta E_{\text{MM}} + \Delta G_{\text{sol}} - T\Delta S, \tag{3.12}$$

where ΔE_{MM} includes bonded (bond, angle, dihedral) and non-bonded (electrostatic, van der Waals) interactions derived from force field energy differences between the protein-ligand complex and its separated components[206,215,216]. ΔG_{sol} comprises polar ($\Delta G_{\text{PB/GB}}$) and nonpolar (ΔG_{SA}) solvation terms. The polar term is calculated using either the Poisson-Boltzmann equation, which solves the electrostatic potential numerically, or the Generalized Born approximation, a faster but less rigorous alternative[215,217]. The nonpolar contribution is estimated via solvent-accessible surface

area[206]. The entropy term $T\Delta S$ captures the entropy change associated with ligand binding and is often estimated through methods such as normal mode analysis or quasi-harmonic approximation[215]. The workflow involves three steps - (1) explicit-solvent MD simulations of the bound complex, (2) extraction of snapshots with solvent/ions removed, and (3) energy averaging using implicit PBSA/GBSA[216,217]. Two protocols are commonly used, the single-trajectory approach using bound-state snapshots for all components and the separate-trajectory approach using independent simulations for complex, protein, and ligand. The single-trajectory method reduces noise by cancelling internal energy errors but assumes minimal conformational changes upon binding[206,216]. Explicit solvent is critical for MD simulations to preserve water-mediated interactions (e.g., hydrogen bonds), while implicit solvent minimizes energy fluctuations during free energy calculations despite inconsistencies in energy functions[206]. MM-PBSA/GBSA balances speed and accuracy, outperforming docking scores for pose rescoring and lead prioritization[206,213,216]. However, accuracy depends on conformational sampling quality, neglect of explicit solvent polarization, and the assumption of rigid-body separation[216].

Alchemical Free Energy Calculations

Alchemical free energy (AFE) methods involve simulating a series of non-physical (alchemical) transformations, where the ligand is gradually changed from one state to another. This transformation can involve either decoupling the ligand from the protein's binding site or modifying its molecular identity to resemble a different ligand. The alteration process is governed by modifying the potential energy function with a parameter, often denoted as λ, which incrementally adjusts the interaction strength between the ligand and its environment. By carefully tuning λ, AFE methods transition between the two states in controlled alchemical steps, called "windows," to ensure accurate free energy estimation[52]. The free energy difference across these alchemical transformations is then computed using estimators such as the Multistate Bennett Acceptance Ratio (MBAR) or Weighted Histogram Analy-

sis Method (WHAM). AFE methods are particularly powerful due to their ability to compute free energy differences for a wide range of molecular processes, such as the binding of small molecules to receptors, the solvation of compounds across different environments (e.g., aqueous to apolar phases), and assessing the impact of protein mutations on binding affinities or stability[52,218,219]. This alchemical approach is justified because the free energy ΔG is a state function, which means that it depends only on the initial and final states of the system and not on the path connecting them[52]. This path independence allows binding affinities to be computed indirectly through thermodynamic cycles that replace physical binding/unbinding processes (which are computationally prohibitive) with alchemical transformations. By constructing closed thermodynamic cycles (e.g., Figure 3.8), free energy differences between ligands can be derived from alchemical steps alone, circumventing direct simulation of k_{on} and k_{off}[52,219] in Equations 3.1 and 3.2. Different variants of AFE calculations exist. In the following I will focus on Relative Binding Free Energy (RBFE), and for more details on other Alchemical free energy methods, such as Absolute Binding Free Energy (ABFE), please refer to Mey et al.,[52].

Relative Binding Free Energy methods calculate the difference in binding affinity between two related ligands, A and B, bound to the same protein target[52,129,218,220]. Instead of simulating the complete binding and unbinding processes for each ligand, RBFE calculations employ alchemical transformations to gradually mutate ligand A into ligand B within the protein's binding site (Figure 3.8).

The thermodynamic cycle in Figure 3.8 enables this indirect calculation by exploiting the path independence of free energy. The cycle consists of four steps: (i) alchemical transformation of A to B in the bound state ($\Delta G_{A\to B}$), (ii) physical unbinding of B ($\Delta G_{\text{bind},B}$), (iii) reverse alchemical transformation of B to A in solution ($-\Delta G_{A\to B,\text{solv}}$), and (iv) physical binding of A ($-\Delta G_{\text{bind},A}$)[52]. Since the total free energy change around a closed cycle is zero, the energy is denoted as,

$$\Delta G_{A\to B} + \Delta G_{\text{bind},B} - \Delta G_{A\to B,\text{solv}} - \Delta G_{\text{bind},A} = 0. \tag{3.13}$$

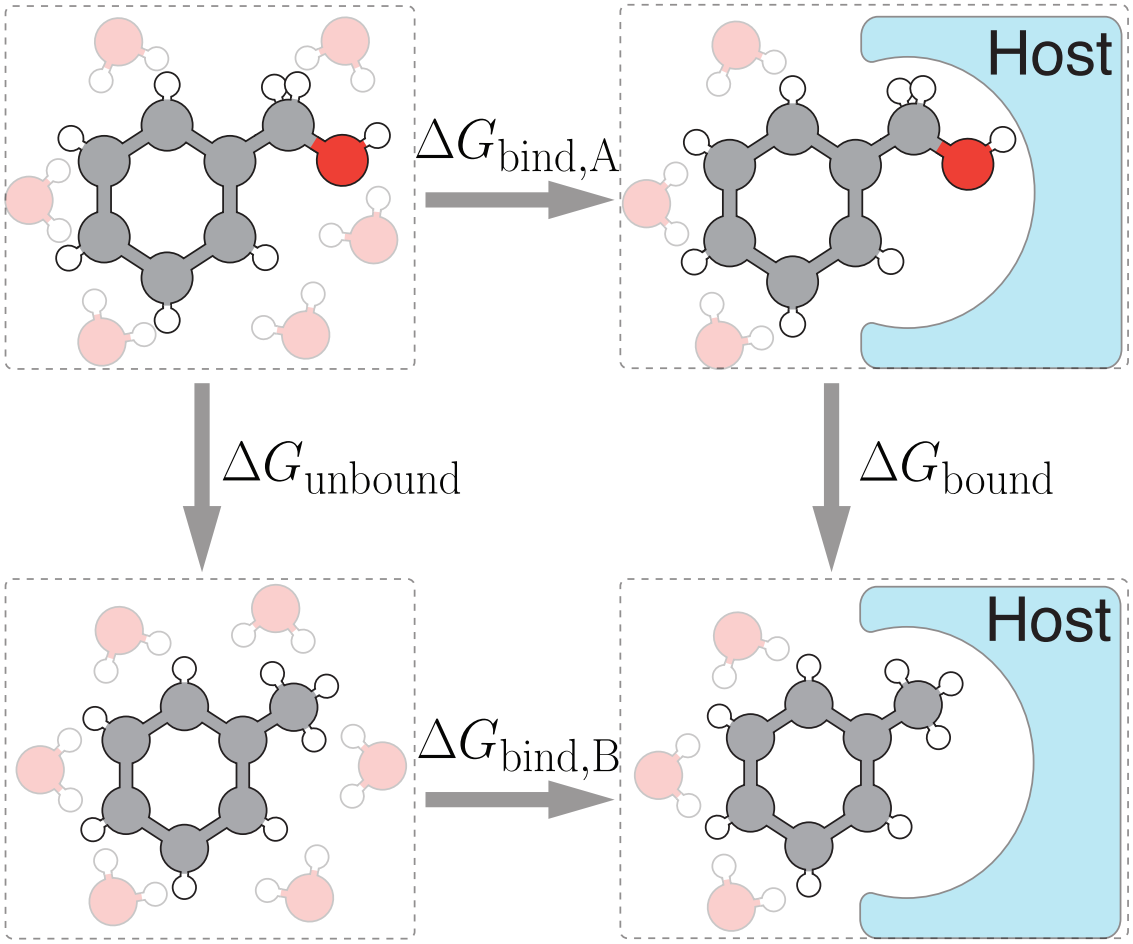


Figure 3.8: **Thermodynamic cycle for Relative Binding Free Energy (RBFE) calculations.** This illustration demonstrates the RBFE approach for evaluating the binding affinity difference ($\Delta\Delta G_{\text{bind}}$) between two structurally similar ligands, A (benzyl alcohol, top) and B (toluene, bottom), bound to a common target using alchemical transformations (horizontal arrows) rather than physical binding/unbinding (vertical arrows). The vertical paths (physical processes) are computationally inaccessible, but their free energy difference is obtained indirectly via the alchemical paths. The alchemical steps mutate A into B in the protein-bound state ($\Delta G_{A\to B}$) and in solution ($\Delta G_{A\to B,\text{solv}}$), leveraging the path independence of free energy. The difference between these alchemical changes ($\Delta\Delta G_{\text{bind}} = \Delta G_{A\to B} - \Delta G_{A\to B,\text{solv}}$) equals the relative binding affinity. Adapted under the CC-BY 4.0 license from Mey et al.[52]

Rearranging yields the relative binding free energy,

$$\Delta\Delta G_{\text{bind}} = \Delta G_{\text{bind},B} - \Delta G_{\text{bind},A} = \Delta G_{A\to B} - \Delta G_{A\to B,\text{solv}}. \tag{3.14}$$

This approach avoids simulating physical binding/unbinding (steps ii and iv), which are computationally intractable, and instead computes the free energy difference using alchemical transformations (steps i and iii)[52].

In practice, this calculation involves two main transformations, one where ligand A is converted into ligand B in the protein binding pocket (yielding $\Delta G_{\text{A}\to\text{B}}$), and another where the same transformation occurs in solvent (yielding $\Delta G_{\text{A}\to\text{B, solv}}$). Thus, the difference in binding affinity between ligands A and B, $\Delta\Delta G_{\text{bind},A\to B}$, is obtained by comparing these two alchemical transformations. This alchemical approach is especially useful in lead optimization, where small structural modifications to ligands are tested to enhance binding affinity. Alchemical free energy methods are among the most accurate techniques available for calculating binding free energies, often achieving root mean square errors within 1-2 kcal/mol when compared to experimental values[52].

3.3.2 Ligand-based Approaches

Ligand-based approaches use SMILES or SMILES-based featuriser to learn the relationship between compounds and their biological activities. The earliest and most widely used method in ligand-based modelling is the *Quantitative Structure-Activity Relationship (QSAR)* model, which dates back to the 1960s[221]. QSAR models are based on the assumption that the biological activity of a molecule can be quantitatively related to its chemical structure through molecular descriptors. These descriptors include physicochemical properties such as hydrophobicity, electronic properties, and steric factors, which are then used to establish a mathematical relationship between structure and activity. Traditional QSAR models are linear models, where the biological activity is expressed as a linear combination of molecular

descriptors,

$$\text{Activity} = \sum_{i=1}^{n} w_i \cdot \texttt{descriptor}_i + b, \tag{3.15}$$

where w_i are the weights assigned to each molecular `descriptor`, and b is the bias term[221–223]. These early models, pioneered by Hansch and Fujita, focused on linear relationships, assuming that the biological response could be adequately explained using a small set of carefully chosen descriptors[221,223]. Despite their simplicity, linear QSAR models were foundational in exploring the effect of molecular properties on biological activity[222]. However, they often fall short when capturing complex, non-linear interactions prevalent in real-world datasets. This limitation led to the adoption of more advanced non-linear machine learning models, which are discussed in Chapter 3.5.

3.4 Machine Learning for Protein-ligand Binding Affinity Prediction

Machine learning (ML) models aim to learn patterns from data and make predictions or decisions based on them. In general, an ML model can be represented as a mapping function $f(\mathbf{x})$, where $\mathbf{x}$ is an input feature vector and $f(\mathbf{x})$ provides the prediction[224,225]. ML models vary in complexity and non-linearity. Simple linear models predict based on linear combinations of features, while more sophisticated models, such as Random Forests (RF) and Support Vector Machines (SVM), capture non-linear relationships. Deep learning (DL), a subset of ML, uses neural networks with multiple layers to model intricate patterns in data[224]. In this section, I discuss ML/DL models focusing on those utilized in this thesis, along with dimensionality reduction methods, evaluation metrics, and statistical tests.

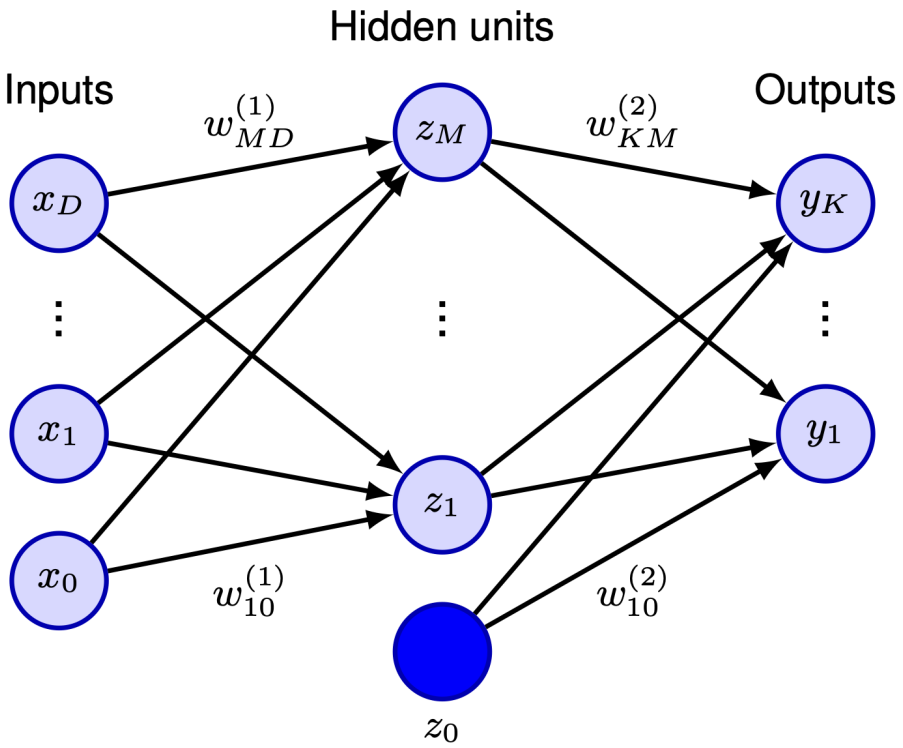


Figure 3.9: **Ilustration of a simple MLP architecture.** The figure illustrates a feedforward neural network with an input layer, one hidden layer, and an output layer. Each layer consists of neurons connected by weights, with biases indicated by separate nodes. The arrows demonstrate the flow of information during forward propagation and the composition of features across layers. The figure was adapted from Bishop et al.,[224] under the CC-BY 4.0 license.

3.4.1 Machine Learning Methods for Learning Affinities

Multilayer Perceptrons (MLPs) are one of the foundational architectures in deep learning and serve as a universal function approximator capable of modelling non-linear relationships across features[224,225]. They are also referred to as fully connected (FC) networks. Unlike linear models, which are limited to capturing linear dependencies, MLPs introduce hidden layers with non-linear activation functions, enabling the modelling of complex relationships in data[224,225]. An MLP consists of multiple layers — an input layer, one or more hidden layers, and an output layer (Figure. 3.9). Each layer is composed of neurons, where the output of each neuron is computed as a weighted sum of its inputs, followed by the application of a non-linear activation function.

For a given hidden layer, the pre-activation values for a neuron j are computed

as,

$$a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \tag{3.16}$$

where $a_j^{(1)}$ is the pre-activation value for the j-th neuron in the first hidden layer, $w_{ji}^{(1)}$ are the weights associated with the connections between the i-th input feature x_i and the j-th neuron, and $w_{j0}^{(1)}$ is the bias term for the j-th neuron[224,225]. The input features x_i represent the raw data passed to the model, while D is the total number of input features[224,225]. These pre-activation values $a_j^{(1)}$ are then passed through a non-linear activation function $h(\cdot)$, to compute the output of the j-th neuron in the hidden layer,

$$z_j^{(1)} = h(a_j^{(1)}), \tag{3.17}$$

where $z_j^{(1)}$ is the activation or output of the j-th neuron in the hidden layer after applying the non-linear transformation[224,225]. The activation function $h(\cdot)$ introduces non-linearity into the model, allowing it to learn complex patterns in the data. For example, the Rectified Linear Unit (ReLU) activation function is defined as $h(x) = \max(0, x)$, which outputs the input value if it is positive and zero otherwise. In this formulation, $a_j^{(1)}$ represents the intermediate computation before applying the activation function, while $z_j^{(1)}$ is the final output of the neuron in the hidden layer that is passed to subsequent layers in the network[224,225]. The outputs of the hidden layer are subsequently combined in a similar fashion to compute the pre-activation values for the next layer, including the final output layer. Adding more hidden layers to an MLP increases the network's complexity, enabling it to model highly intricate and non-linear relationships in the data[224,225]. However, this also increases the number of parameters, making the model more computationally expensive to train and more prone to challenges such as overfitting if not properly regularized[226]. Deep MLPs, with many hidden layers, can capture hierarchical feature representations, where each successive layer learns increasingly abstract features of the input data[224].

Training MLPs involves minimizing a loss function, which quantifies the differ-

ence between the model's predictions and the actual target values. For a regression task, the mean squared error is commonly used as the loss function,

$$\mathtt{L} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - f(\mathbf{x}_i)\right)^2, \tag{3.18}$$

where y_i are the true values, $f(\mathbf{x}_i)$ are the predicted values, and n is the number of data points[226]. The goal of training is to adjust the model's parameters, including weights and biases across all layers, to minimize this loss function[225,226].

Weights are updated iteratively using gradient-based optimization methods, such as stochastic gradient descent (SGD)[224,225]. The updates are guided by the gradients of the loss function with respect to each parameter, which are calculated using backpropagation. Backpropagation efficiently computes these gradients by applying the chain rule, propagating the error from the output layer backwards through the network to earlier layers[224]. For a given weight w, the update rule in gradient descent is,

$$w \leftarrow w - \eta\frac{\partial \mathtt{L}}{\partial w}, \tag{3.19}$$

where η is the learning rate, a hyperparameter that controls the step size of the updates[226]. During training, this process is repeated iteratively over multiple passes through the data (epochs), gradually reducing the loss and improving the model's predictions[226]. The combination of backpropagation and gradient-based optimization allows MLPs to learn complex mappings from inputs to outputs, enabling their application across diverse domains[226]. Below, I discuss various deep learning architectures I use in this thesis.

Convolutional Neural Networks (CNNs) are a class of deep learning models primarily developed for analyzing data with a grid-like topology, such as images[227–231]. The fundamental operation in CNNs is the convolution, which applies a filter (or kernel) to local patches of the input data to produce a feature map[227].

This operation is given by,

$$\mathbf{y}_j = f\left(\sum_i \mathbf{w}_i * \mathbf{x}_i + b_j\right), \tag{3.20}$$

where $\mathbf{x}_i$ represents the input data, $\mathbf{w}_i$ are the convolutional filters, $*$ denotes the convolution operation, b_j is the bias term, and f is a non-linear activation function such as ReLU ($f(z) = \max(0, z)$)[227]. The convolution operation is followed by pooling layers, typically max pooling, which down-sample the spatial dimensions of the feature maps,

$$\mathbf{y}_{\text{pool}} = \max_{\text{region}}(\mathbf{y}_j), \tag{3.21}$$

where the maximum value is taken over a specific region of the feature map, reducing its size and allowing the network to be invariant to small translations in the input[227]. The resulting feature maps are then passed through fully connected layers to perform the final prediction.

Graph Neural Networks (GNNs) are a class of neural networks designed to operate on graph-structured data, where entities are represented as nodes and their relationships as edges[232,233]. GNNs iteratively update node representations by aggregating information from neighbouring nodes and edges[232]. The general framework for GNNs can be described by the message-passing paradigm,

$$\mathbf{h}_v^{(t)} = \sigma\left(\mathbf{W}^{(t)} \cdot \text{AGGREGATE}\left(\{\mathbf{h}_u^{(t-1)}, \forall u \in \mathcal{N}(v)\}\right) + \mathbf{b}^{(t)}\right), \tag{3.22}$$

where $\mathbf{h}_v^{(t)}$ is the hidden state of node v at iteration t, $\mathbf{W}^{(t)}$ is a learnable weight matrix, AGGREGATE is a function that aggregates the hidden states of neighbouring nodes $\mathcal{N}(v)$, and σ is a non-linear activation function[232]. One of the commonly used GNN architectures is Graph Convolutional Networks (GCNs). In GCNs, the aggregation is typically a normalized sum of the neighbours' features,

$$\mathbf{h}_v^{(t)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \frac{1}{\sqrt{d_v d_u}} \mathbf{W}^{(t)} \mathbf{h}_u^{(t-1)} + \mathbf{b}^{(t)}\right), \tag{3.23}$$

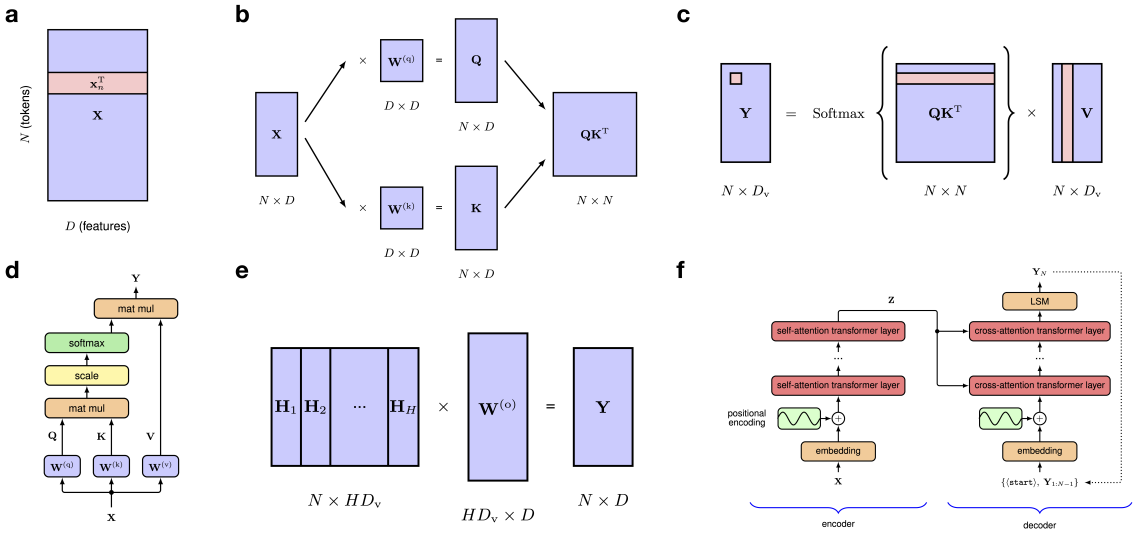


Figure 3.10: **Overview of the transformer architecture.** (a) The input sequence $\mathbf{X}$ consists of N tokens, each with D features. (b) Query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) matrices are computed by projecting $\mathbf{X}$ using learned weight matrices $\mathbf{W}^{(\mathrm{q})}$, $\mathbf{W}^{(\mathrm{k})}$, and $\mathbf{W}^{(\mathrm{v})}$, respectively. (c) Attention scores are obtained by calculating $\mathbf{QK}^{\mathrm{T}}$, scaling, and applying softmax. The result is used to compute a weighted sum of the values $\mathbf{V}$, yielding the output $\mathbf{Y}$. (d) A schematic representation of the self-attention mechanism. (e) Multi-head attention combines outputs from multiple heads $\{\mathbf{H}_1, \ldots, \mathbf{H}_H\}$, followed by a linear transformation. (f) The full transformer architecture consists of an encoder-decoder structure, where the encoder processes the input sequence, and the decoder generates outputs using cross-attention and positional encodings. The figure was adapted from Bishop et al.,[224] under the CC-BY 4.0 license.

where d_v and d_u are the degrees of nodes v and u, respectively. For more details on various GNN architectures, please refer to these works[232–234].

Transformers and language models are a class of models that utilize self-attention mechanisms to process sequential data, capturing dependencies between elements regardless of their distance within the sequence[196,235]. Originally developed for natural language processing (NLP) tasks, transformers have been successfully adapted to other domains, including analyzing protein sequences and SMILES representations for drug discovery, due to their ability to handle complex sequential data effectively[224]. At the core of the transformer architecture is the self-attention mechanism, which allows each element in the input sequence to refer to every other element[196]. This is achieved by calculating attention weights that reflect the importance of each element with respect to others, enabling the model to focus on

relevant parts of the input for better predictions. The input data to a transformer is a set of vectors $\{\mathbf{x}_n\}$ of dimensionality D, where $n = 1, \ldots, N$[224]. These data vectors are referred to as tokens, where a token might, for example, correspond to a word within a sentence, a patch within an image, or an amino acid within a protein. These tokens are stacked together to form a matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, where each row corresponds to a token $\mathbf{x}_n$ (Figure. 3.10a). For each element in the input sequence, represented as a vector $\mathbf{X}$, the model computes three matrices - queries (Q), keys (K), and values (V)[196]. These are obtained by linear transformations using learned weight matrices $\mathbf{W}^{(\mathrm{q})}$, $\mathbf{W}^{(\mathrm{k})}$, and $\mathbf{W}^{(\mathrm{v})}$, respectively[196,225],

$$\begin{aligned} \mathbf{Q} &= \mathbf{X}\mathbf{W}^{(\mathrm{q})}, \\ \mathbf{K} &= \mathbf{X}\mathbf{W}^{(\mathrm{k})}, \\ \mathbf{V} &= \mathbf{X}\mathbf{W}^{(\mathrm{v})}, \end{aligned} \tag{3.24}$$

Here $\mathbf{W}^{(\mathrm{q})}$, $\mathbf{W}^{(\mathrm{k})}$, and $\mathbf{W}^{(\mathrm{v})}$ are weight matrices of dimensionality $D \times D_{\mathrm{k}}$ for $\mathbf{W}^{(\mathrm{q})}$ and $\mathbf{W}^{(\mathrm{k})}$, and $D \times D_{\mathrm{v}}$ for $\mathbf{W}^{(\mathrm{v})}$ [196,225]. Typically, $D_{\mathrm{k}} = D_{\mathrm{v}} = D$ is used to simplify the architecture[224].

The attention mechanism calculates the unnormalized attention weights, denoted ω, by taking the dot product of the query matrix $\mathbf{Q}$ with the transpose of the key matrix $\mathbf{K}^{\mathrm{T}}$ (Figure. 3.10b)[196,225]. These unnormalized weights are scaled by $\frac{1}{\sqrt{D_{\mathrm{k}}}}$, where D_{k} is the dimensionality of the key vectors[196,225]. This scaling ensures that the Euclidean length of the weight vectors remains within a suitable range, preventing the attention weights from becoming too small or too large, which could cause numerical instability or hinder convergence during training[196,225],

$$\omega = \frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{D_{\mathrm{k}}}}. \tag{3.25}$$

The unnormalized weights ω are then passed through a softmax function to obtain the normalized attention weights, α, which determine the contribution of each

element in the sequence to the final output[196,225],

$$\alpha = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\text{T}}}{\sqrt{D_{\text{k}}}}\right). \tag{3.26}$$

Finally, these normalized attention weights are used to compute a weighted sum of the value matrix $\mathbf{V}$, resulting in the output matrix $\mathbf{Y}$ of the self-attention mechanism[196,225] (Figure. 3.10c),

$$\mathbf{Y} = \alpha \cdot \mathbf{V}. \tag{3.27}$$

The transformer architecture leverages multiple layers of self-attention mechanisms (Figure. 3.10d), followed by feed-forward neural networks[224]. Additionally, positional encoding is introduced to the input vectors to retain information about the order of the sequence, as the self-attention mechanism itself does not consider positional information[224]. In practice, transformers employ multi-head self-attention, where multiple self-attention mechanisms (or heads) are run in parallel (Figure. 3.10e)[224]. For a given head h, the output of the attention mechanism is computed as,

$$\mathbf{H}_h = \text{Attention}\left(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h\right), \tag{3.28}$$

where $\mathbf{Q}_h$, $\mathbf{K}_h$, and $\mathbf{V}_h$ are derived using separate weight matrices for each head[224],

$$\begin{aligned} \mathbf{Q}_h &= \mathbf{X}\mathbf{W}_h^{(\text{q})}, \\ \mathbf{K}_h &= \mathbf{X}\mathbf{W}_h^{(\text{k})}, \\ \mathbf{V}_h &= \mathbf{X}\mathbf{W}_h^{(\text{v})}. \end{aligned} \tag{3.29}$$

The outputs of all heads, $\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_H$, are then concatenated and linearly transformed using a weight matrix $\mathbf{W}^{(\circ)}$ to produce the final output[224],

$$\mathbf{Y}(\mathbf{X}) = \text{Concat}\left[\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_H\right]\mathbf{W}^{(\circ)}. \tag{3.30}$$

This multi-head approach enables the model to focus on different aspects of the input sequence simultaneously, enhancing its ability to learn complex relationships

within the data[224]. Multiple transformer layers can be stacked on top of one another, each maintaining the same dimensionality D, to further refine the learned representations[224].

As shown in Figure. 3.10f, the transformer architecture itself is composed of stacks of encoder and decoder units. The input to the encoder model is a tokenized sequence, where the sequence is broken into units called tokens, which can represent characters, words, or other elements depending on the strategy to tokenize the given sequence[196,224]. The transformer model's encoder focuses on identifying a token's relationship to its surrounding tokens, both preceding and following, using self-attention. The decoder, on the other hand, learns to predict the next token based solely on previously processed tokens. The decoder uses masked self-attention, where future tokens are masked for each token in the sequence to prevent the model from looking ahead[196,224,225]. Cross-attention helps in the interaction between the encoder and decoder by computing the relationships between the decoder's predicted tokens and the encoder's contextualized embeddings of the input tokens[224,225]. Cross-attention operates similarly to self-attention but considers both the encoder and decoder contexts[224,225]. For a query vector from the decoder, cross-attention can be formulated as,

$$\alpha = \text{softmax}\left(\frac{Q_{\text{dec}} K_{\text{enc}}^{T}}{\sqrt{D_k}}\right), \tag{3.31}$$

where Q_{dec} is the query matrix from the decoder, and K_{enc} is the key matrix from the encoder[224,225]. The resulting attention weights α are then applied to the value matrix V_{enc} from the encoder to create a context vector that combines information from both the encoder and decoder[224,225]. Also, over the last few years, several transformer-like models (ex., BERT[236] and RoBERTa[237]) have introduced enhancements and new objectives to further optimize the transformer architecture for specific tasks. For a comprehensive review of the advancements and specialized transformer architectures, refer to these works[235,238–240].

In my thesis, I focus on using language models that are particularly trained on

large publicly available protein sequences[167,241] and ligand SMILES[242] datasets for extracting features and for downstream binding affinity prediction. Various transformer architectures have been developed specifically for proteins and ligands[238,239]. Here, I provide an overview of two commonly used models, *ESM-2*[243], a protein language model and ChemBERTa-2[199], a SMILES-based language model I have used in this thesis.

ESM-2 is part of the Evolutionary Scale Modeling (ESM) family[243,244], is a transformer model designed for learning from protein sequences. It follows a BERT-style[236] architecture with an encoder-only transformer setup optimized for masked language modelling (MLM) tasks. The model is trained on UniRef dataset[167], containing 65 million unique sequences allowing the model to learn evolutionary relationships and structural patterns within protein sequences. ESM-2 introduces rotary position embeddings for positional encoding, which improves the model's ability to extrapolate to longer sequences beyond its training context window[243,244]. During training, 15% of tokens are randomly masked, and the model learns to predict these masked tokens based on the surrounding context, with the MLM objective defined as,

$$\mathtt{L}_{\mathrm{MLM}} = -\mathbb{E}_{x\sim X} \sum_{i\in M} \log p(x_i|\hat{x}), \tag{3.32}$$

where X represents the set of all protein sequences, M is the set of masked token positions, x_i is the true token at position i, and $\hat{x}$ is the input sequence with masked tokens[243,244]. ESM-2 is trained using a vocabulary of amino acid tokens with special tokens for the start (`[CLS]`) and end (`[SEP]`) of the sequence[243,244].

ChemBERTa-2 is built upon the RoBERTa architecture to learn SMILES-based molecular representations. The model is trained on a dataset of 77 million unique SMILES strings, sourced from PubChem[242] and preprocessed to ensure canonicalization and shuffling. Similar to ESM-2, ChemBERTa-2 uses the MLM objective, which masks 15% of tokens in each SMILES string and trains the model to correctly predict them. Additionally, ChemBERTa-2 incorporates a multitask regression (MTR) objective to predict multiple molecular properties simultaneously. These proper-

ties are not based on experimental measurements but can each be calculated from SMILES alone using RDKit[245]. The MTR objective aims to capture various molecular features, with the loss function,

$$\mathrm{L}_{\mathrm{MTR}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}(y_{ij} - \hat{y}_{ij})^2, \tag{3.33}$$

where N is the number of molecules, M is the number of properties, y_{ij} is the true value of property j for molecule i, and $\hat{y}_{ij}$ is the predicted value. ChemBERTa-2 uses a token vocabulary of 591 SMILES characters and allows sequences up to 512 tokens, handling diverse SMILES components, including bracketed atoms and special characters. This dual training approach makes ChemBERTa-2 particularly suited for chemical property prediction and feature extraction in cheminformatics applications. ESM-2 and ChemBERTa-2 are both available via the Hugging face transformers library[240].

Gaussian Process Regression (GP) is a non-parametric, probabilistic model used for regression tasks[247]. It models the distribution over possible functions that fit the data, providing both predictions and uncertainty estimates[247]. A Gaussian process is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution[247]. Formally, a Gaussian process is characterized by a mean function $\mu(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$, where $\mathbf{x}$ and $\mathbf{x}'$ are feature vectors in the input space[247].

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{3.34}$$

where the mean function is typically assumed to be zero, $\mu(\mathbf{x}) = 0$, for simplicity, and the covariance function, or kernel, $k(\mathbf{x}, \mathbf{x}')$, defines the relationship between any two points $\mathbf{x}$ and $\mathbf{x}'$ in the input space[247]. The kernel function encodes prior knowledge about the function we wish to learn. For example, the Radial Basis

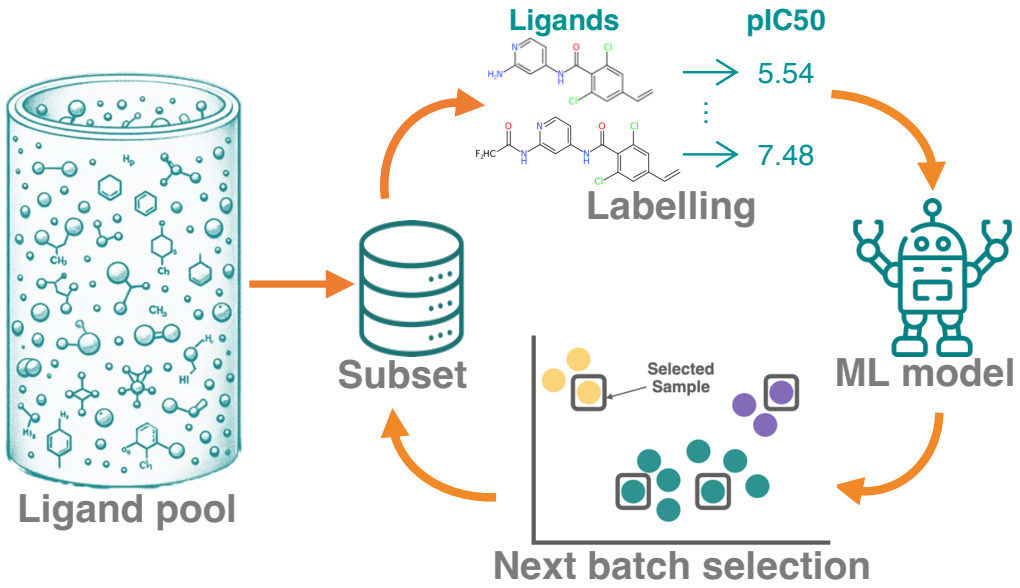


Figure 3.11: **Workflow of the Active Learning (AL) pipeline.** The figure illustrates the sequential steps in an AL cycle, starting with the initialization of a labelled dataset used to train a predictive model. An acquisition function is then employed to select the most informative compounds from an unlabeled pool based on criteria such as uncertainty or similarity to known active compounds. Selected compounds are sent for labelling through experimental validation or computational simulations, with the newly labelled data incorporated back into the labelled dataset. The model is retrained on this updated dataset, improving its predictions over successive iterations. This iterative AL process continues, progressively enhancing the model's predictive accuracy by strategically selecting data points for labelling in each cycle. The figure was adapted from Gorantla et al.,[246] under the CC-BY 4.0 license.

Function (RBF) kernel assumes smoothness and is defined as,

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2l^2}\right), \tag{3.35}$$

where l is the length scale hyperparameter that controls the smoothness of the resulting function[247]. In cheminformatics, where binary fingerprints are used, Tanimoto or Jaccard kernels are often employed, which are particularly suited for measuring similarity between binary vectors[246,247],

$$k_{\text{Tanimoto}}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x} \cdot \mathbf{x}'}{||\mathbf{x}||^2 + ||\mathbf{x}'||^2 - \mathbf{x} \cdot \mathbf{x}'}. \tag{3.36}$$

Given a set of training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the goal is to predict the value $f(\mathbf{x}_*)$ at a new point $\mathbf{x}_*$. The predictive distribution for $f(\mathbf{x}_*)$ is Gaussian with mean and variance given by,

$$\mu(\mathbf{x}_*) = \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, \tag{3.37}$$

$$\sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{k}_*, \tag{3.38}$$

where $\mathbf{K}$ is the covariance matrix of the training data, $\mathbf{I}$ is the identity matrix, $\mathbf{k}_*$ is the covariance vector between $\mathbf{x}_*$ and the training points, σ_n^2 is the noise variance, and $\mathbf{y}$ is the vector of observed values[247].

Active learning (AL) is a semi-supervised learning approach designed to optimize the learning process by selectively querying the most informative data points for labelling[246,248] (Figure. 3.11). In fields where labelled data is limited or costly to obtain, AL provides an efficient strategy to enhance model performance using fewer labelled examples. This is particularly beneficial in drug discovery, where generating labelled data often involves resource-intensive experimental or computational procedures. AL begins with the initialization of dataset D, which is divided into a labelled set D_L containing molecules with known biological activities and an unlabeled set D_U with molecules whose activities are unknown[248] . The labeled set includes instances $(\mathbf{x}^{(i)}, y^{(i)})$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ represents a molecular descriptor vector, and $y^{(i)}$

is the corresponding activity label. The unlabeled set contains instances $(\mathbf{x}^{(i)}, ?)$, where the activity label is unknown[248].

Model training and hypothesis function is the first step in the AL process, where a hypothesis function $h : \mathbb{R}^d \to \mathbb{R}$ is trained on the labelled dataset D_L[248]. This function is then used to predict the activity of molecules in the unlabeled set D_U. As the AL process progresses, the model is iteratively refined by selecting new data points for labelling and incorporating them into D_L[248].

The *acquisition function* is used to determine which molecules from D_U should be selected for labelling. Various strategies can be employed to select molecules, such as *similarity-based selection*, which prioritizes molecules structurally similar to known active compounds, typically using metrics such as the Tanimoto coefficient computed with ECFPs[246,248]. *Exploitative selection* involves selecting molecules that are predicted to have the highest probability of being active, thereby focusing on maximizing the predicted activity. This is formulated as,

$$\mathbf{x}_{\text{exploit}} = \arg\max_{\mathbf{x} \in D_U} P(y|\mathbf{x}, D_L), \tag{3.39}$$

where $P(y|\mathbf{x}, D_L)$ represents the predicted probability of activity for each molecule. *Explorative selection* targets molecules where the model shows the greatest uncertainty in its predictions, often quantified by the entropy of the predicted distribution

$$\mathbf{x}_{\text{explore}} = \arg\max_{\mathbf{x} \in D_U} H(P(y|\mathbf{x}, D_L)), \tag{3.40}$$

where H denotes entropy, providing a measure of the model's uncertainty[248,249].

Upper confidence bound (UCB) is a technique that balances exploration and exploitation by selecting molecules based on the upper confidence bound of their predicted activity[248,249]. UCB considers both the predicted mean $\mu(\mathbf{x})$ and the uncertainty (standard deviation) $\sigma(\mathbf{x})$ of the prediction. Molecules are selected by

maximizing the UCB, typically expressed as

$$\mathbf{x}_{\text{UCB}} = \arg\max_{\mathbf{x} \in D_U} \left[\mu(\mathbf{x}) + \kappa \cdot \sigma(\mathbf{x}) \right], \tag{3.41}$$

where $\mu(\mathbf{x})$ is the predicted mean activity, $\sigma(\mathbf{x})$ is the predicted uncertainty, and κ is a parameter that controls the balance between exploration (higher κ encourages exploring uncertain molecules) and exploitation (lower κ favours molecules with high predicted mean activity)[248,249].

Oracle and experimental validation follows the acquisition step, where selected molecules are sent to an oracle—usually an experimental procedure for biological testing or computational methods such as AFE calculations[250]. The resulting data is labelled and added to D_L, enriching the training set with new, informative examples. Then, the model is updated, which involves retraining the model on the updated labelled set D_L to improve predictive accuracy. This iterative process continues, with each cycle refining the model's ability to identify bioactive compounds, ultimately leading to a more accurate and efficient model[248,250].

3.4.2 Dimensionality Reduction Methods

To effectively visualize the chemical space and understand the relationships between high-dimensional data, dimensionality reduction techniques are used[251]. These methods allow us to project complex data into lower-dimensional spaces while preserving significant structures and patterns. Dimensionality reduction techniques can be broadly categorized into linear and non-linear methods, each with its specific strengths. Below, I describe the commonly used linear method, Principal Component Analysis (PCA)[252], and the non-linear method Uniform Manifold Approximation and Projection (UMAP)[253].

Principal component analysis[252] (PCA) is a widely used linear dimensionality reduction technique that transforms data into a new coordinate system such that the greatest variance by any projection of the data lies on the first coordinate

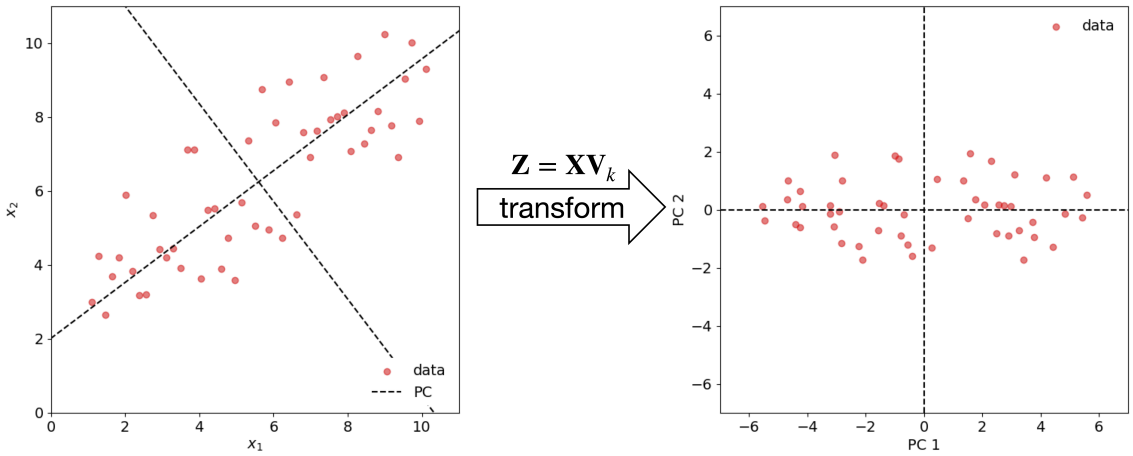


Figure 3.12: **Illustration of PCA transformation.** The left panel shows the original dataset $\mathbf{X}$ consisting of $M = 50$ datapoints in $N = 2$ dimensions. Principal components (PCs) are shown as dashed lines. The right panel shows the transformed dataset $\mathbf{Z}$ after projection onto the first $k = 2$ principal components. The transformation $\mathbf{Z} = \mathbf{X}\mathbf{V}_k$ maps the original data into a new coordinate system defined by the principal components, preserving variance while reducing redundancy.

(called the first principal component), the second greatest variance on the second coordinate, and so on[252]. The idea of PCA is to reduce the dimensionality of the data while retaining as much variance as possible. The transformation is defined by a set of orthogonal vectors (principal components), which are the eigenvectors of the covariance matrix of the data[252]. The corresponding eigenvalues indicate the variance captured by each principal component. Mathematically, PCA is performed by solving the eigenvalue problem,

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}, \tag{3.42}$$

where $\mathbf{C}$ is the covariance matrix of the data, λ is the eigenvalue, and $\mathbf{v}$ is the eigenvector (principal component)[252]. The data is then projected onto the first k principal components to reduce its dimensionality (Figure 3.12),

$$\mathbf{Z} = \mathbf{X}\mathbf{V}_k, \tag{3.43}$$

where $\mathbf{Z}$ is the transformed data, $\mathbf{X}$ is the original data matrix, and $\mathbf{V}_k$ contains the first k eigenvectors[252].

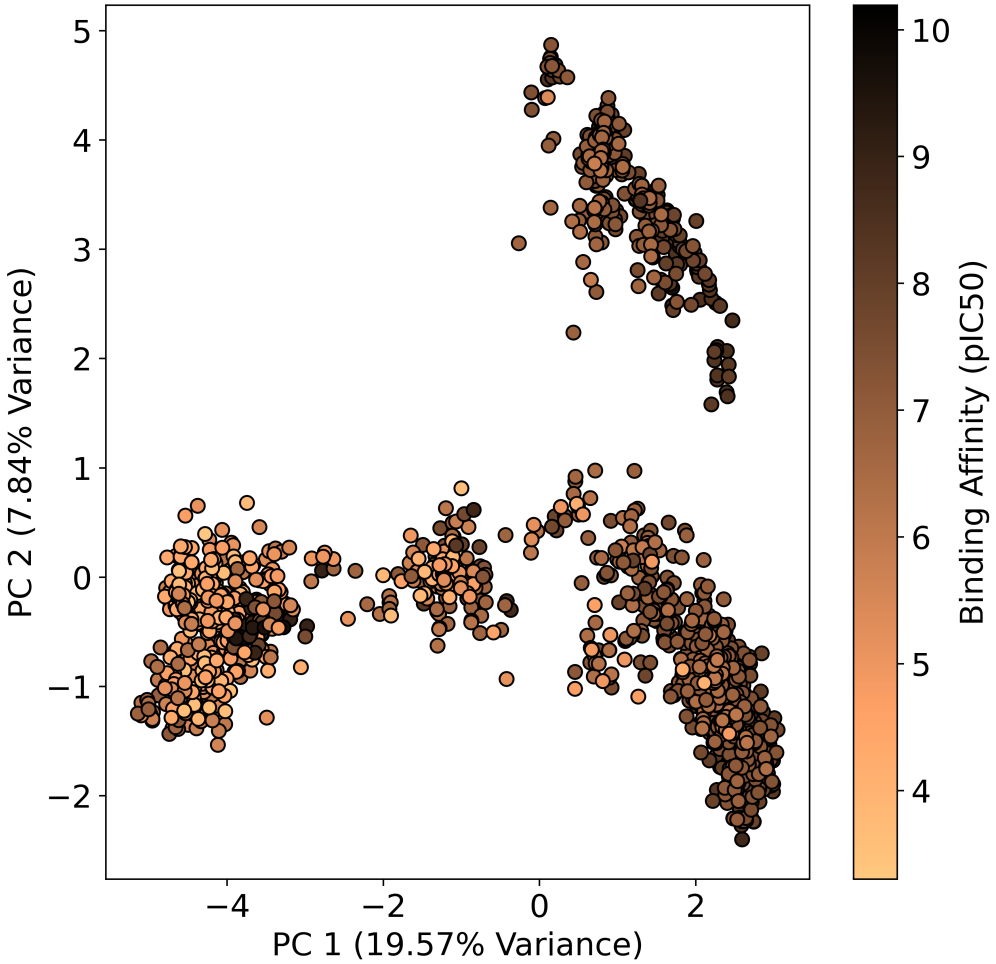


Figure 3.13: **PCA projection of molecular dataset with binding affinity.** The scatter plot illustrates a 2D PCA projection of ECFP fingerprints for the USP7 dataset, with binding affinity values (pIC50) represented by a colour gradient. The x-axis and y-axis represent the first and second principal components (PCs), capturing an explained variance of 19.57% and 7.84% of the total variance in data, respectively. The colour bar indicates the range of binding affinity values, facilitating the interpretation of chemical space and affinity clusters.

PCA is a deterministic method, hence, it does not involve random initialization and produces the same results upon repeated runs with the same data[252]. The primary parameter to set in PCA is the number of principal components (k) to retain. Choosing k involves a trade-off between reducing dimensionality and preserving variance. In Figure 3.13, I show an example of PCA projection with principal components ($k = 2$) on a molecular dataset[254,255] containing inhibitors targeting the ubiquitin-specific protease 7 (USP7). This is often guided by analyzing the explained variance ratio, which indicates how much variance each principal component captures. Typically, components are selected such that a predetermined threshold of the total variance (e.g., 95%) is retained.

Uniform Manifold Approximation and Projection[253] (UMAP) is a non-linear dimensionality reduction technique aimed at preserving both the global structure of data and the local neighbourhood structure within it. Based on manifold learning, UMAP operates on the assumption that the data is uniformly distributed on a Riemannian manifold, which is a mathematical space that locally resembles Euclidean space but may have a more complex, curved structure on a larger scale[253]. UMAP first constructs a weighted k-nearest neighbour graph to approximate this manifold structure and then optimizes a low-dimensional embedding that retains the topological structure of the graph[253].

The method starts by constructing a fuzzy simplicial set in high-dimensional space, representing the local connectivity of the data. The probability of a point $\mathbf{x}_j$ being connected to $\mathbf{x}_i$ is defined as,

$$p_{ij} = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2 - \rho_i}{\sigma_i}\right), \tag{3.44}$$

where ρ_i sets a local connectivity threshold as the distance to the nearest neighbour, and σ_i is a scaling factor chosen to achieve a fixed expected number of neighbours[253]. In the low-dimensional embedding space, UMAP minimizes the cross-entropy between the high-dimensional and low-dimensional representations of this fuzzy sim-

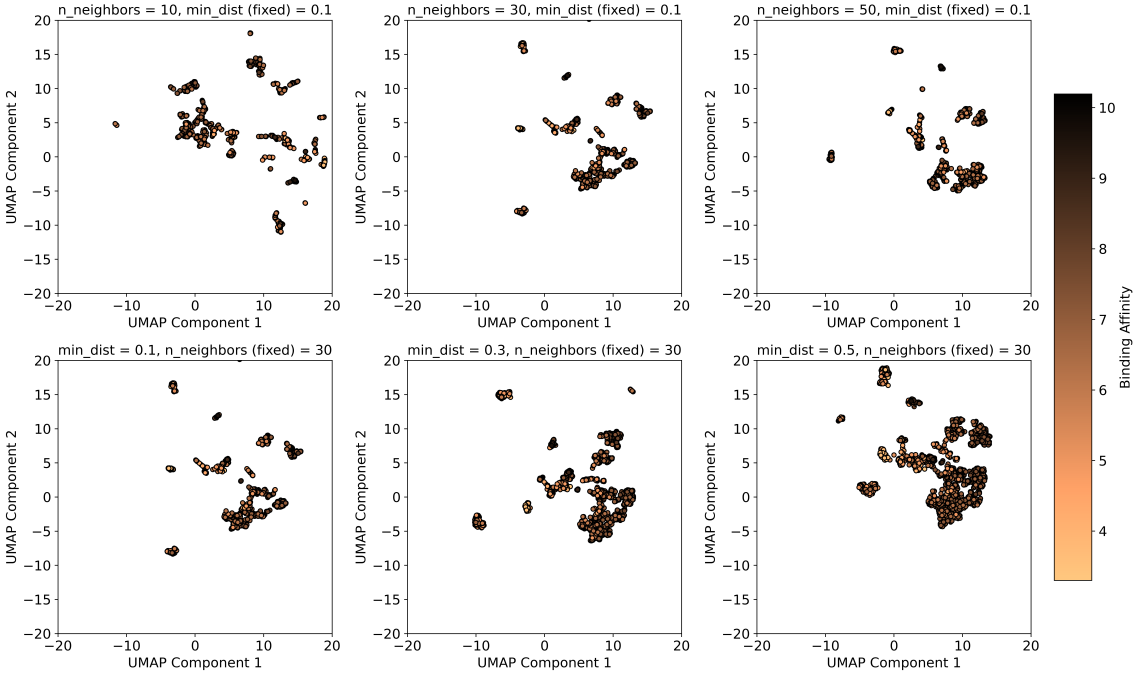


Figure 3.14: **UMAP projections of molecular dataset with varying parameters for local and global structure balance.** The figure illustrates the impact of UMAP parameters on the embedding of USP7 data, using ECFP fingerprints with Jaccard distance metric and binding affinity values represented by a colour gradient. In the first row, the *min_dist* parameter is fixed at 0.1, while the *n_neighbors* parameter is varied (10, 30, and 50) to control the emphasis on local structure. Higher *n_neighbors* values result in more global structure preservation. In the second row, the *n_neighbors* parameter is fixed at 30, while *min_dist* is varied (0.1, 0.3, and 0.5) to adjust the compactness of clusters. Lower *min_dist* values create tighter clusters, while higher values spread the clusters out, providing insights into different levels of neighbourhood preservation. All subplots share the same x-axis and y-axis labels, "UMAP Component 1" and "UMAP Component 2", respectively, ensuring consistency across visualizations. The colour bar on the right indicates the binding affinity across all subplots.

plicial set, with the optimization objective given by

$$C = \sum_{(i,j)} \left[p_{ij} \log q_{ij} + (1 - p_{ij}) \log(1 - q_{ij}) \right], \tag{3.45}$$

where q_{ij} represents connectivity probabilities in the low-dimensional embedding space. This helps to retain both local and global relationships[253].

UMAP's performance and embedding quality are influenced by several parameters as shown in Figure. 3.14 on USP7 data similar to PCA example in Figure. 3.13. The number of neighbours determines the size of the local neighbourhood for mani-

fold approximation, where smaller values emphasize local structures and larger values capture more global patterns. Typical values range from 5 to 50. The minimum distance parameter controls the compactness of points in the low-dimensional embedding; lower values create tighter clusters, while higher values yield more spread-out points. This parameter typically lies between 0 and 1[253]. UMAP also allows customization of the distance metric (Euclidean, Manhattan, and Cosine distances) used in the high-dimensional space, which can significantly influence the resulting embedding. Also, UMAP supports several initialization methods for the embedding process[253]. Spectral initialization, which uses eigenvectors of the graph Laplacian, often leads to faster convergence and better global structure preservation[253]. Alternatively, random initialization is computationally simpler but may require more iterations to reach convergence. These parameters allow UMAP to be tuned according to the specific requirements of the data and analysis task[253].

3.4.3 Evaluation Metrics

In evaluating the performance of binding affinity prediction models, a variety of metrics are used to quantify different aspects of model accuracy and reliability. In this section, I will introduce commonly used metrics to evaluate model performance which are also used in later chapters.

Mean absolute error (MAE) is a common metric for regression tasks that quantifies the average magnitude of the errors in a set of predictions without considering their direction[256]. It is calculated as follows,

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|, \tag{3.46}$$

where $\hat{y}_i$ is the predicted value, y_i is the true value, and n is the number of observations. MAE is useful for understanding the average deviation of predictions from the actual values[256].

Root mean square error (RMSE) measures the square root of the aver-

age squared differences between predicted and actual values, which penalizes larger errors more than MAE[256],

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}. \tag{3.47}$$

RMSE is sensitive to outliers and provides a comprehensive measure of prediction accuracy[256].

Pearson Correlation Coefficient R is a measure of the linear correlation between two variables. It quantifies the degree to which a linear relationship exists between two datasets[256]. The formula for R is given by,

$$R = \frac{\sum_{i=1}^{n}(x_i - \mu(x))(y_i - \mu(y))}{\sqrt{\sum_{i=1}^{n}(x_i - \mu(x))^2 \sum_{i=1}^{n}(y_i - \mu(y))^2}}, \tag{3.48}$$

where x_i and y_i are the paired data points, and $\mu(x)$ and $\mu(y)$ represent the mean values of the x and y data sets, respectively. The Pearson correlation coefficient R ranges from -1 to 1, where a value of 1 indicates a perfect positive linear relationship, -1 is a perfect negative linear relationship, and 0 is no linear correlation between the variables[256].

Matthews correlation coefficient[257] (MCC) is a comprehensive metric used to evaluate the performance of binary classifiers. It accounts for true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) in a single calculation, providing a balanced measure of classification quality even in cases where the classes are imbalanced[257]. The MCC is computed as

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{3.49}$$

where MCC values range from -1 to 1. A value of 1 indicates perfect classification, 0 suggests that the classifier is performing no better than random guessing, and -1 indicates complete misclassification[257]. The strength of MCC lies in its ability to offer a balanced evaluation by considering both the correctly and incorrectly predicted

instances, making it particularly valuable in scenarios with imbalanced datasets[257]. Unlike classification accuracy, which might be misleading in cases of class imbalance, MCC provides a more holistic view by incorporating TPs, TNs, FPs and FNs. This comprehensive approach ensures that the quality of classification is accurately reflected, highlighting cases where accuracy alone might give an incomplete picture of model performance[257].

Next, I discuss *rank correlation* metrics to assess the degree of similarity between the rankings of two variables. These metrics are particularly useful for evaluating the consistency of predicted binding affinities relative to actual values, helping to assess the effectiveness of predictive models.

Spearman rank correlation coefficient (ρ) is a non-parametric measure used to assess the strength and direction of the monotonic relationship between two variables[256]. It evaluates how well the relationship between the variables can be described by a monotonic function. Spearman's ρ is computed as

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}, \tag{3.50}$$

where d_i is the difference between the ranks of corresponding variables[256]. The coefficient ranges from -1 to 1, where 1 indicates perfect positive rank correlation, -1 indicates perfect negative rank correlation, and 0 indicates no correlation[256].

Kendall's τ is another non-parametric statistic that measures the ordinal association between two variables. It is particularly robust to outliers and is often used when data contain tied ranks[256]. Kendall's τ assesses the consistency of orderings by evaluating `concordant` and `discordant` pairs of observations,

$$\tau = \frac{(\texttt{concordant pairs}) - (\texttt{discordant pairs})}{\frac{1}{2}n(n-1)}. \tag{3.51}$$

A pair of observations is considered `Concordant` if the relative ordering of the two variables is consistent between the two observations[256]. Conversely, a pair is `discordant` if the ordering is inconsistent. Kendall's τ ranges from -1, indicating

perfect disagreement, to 1, indicating perfect agreement, with 0 representing no association[256].

Concordance index (CI) measures the proportion of all pairs of samples that are correctly ordered by both the predicted and actual values,

$$CI = \frac{1}{n_c} \sum_{i<j} \mathbf{I}\left((\hat{y}_i > \hat{y}_j) \wedge (y_i > y_j)\right), \tag{3.52}$$

where n_c is the number of comparable pairs, $\hat{y}_i$ and $\hat{y}_j$ are the predicted values, and y_i and y_j are the actual values[131]. The indicator function $\mathbf{I}$ equals 1 if the predicted and actual orderings of the pair are concordant, meaning the prediction accurately reflects the actual ranking[131]. The CI ranges from 0.5 (random prediction) to 1 (perfect prediction), with values below 0.5 indicating a model that performs worse than random[131].

3.4.4 Statistical Methods for Model Comparison

When comparing the performance of different models, statistical methods provide a robust framework to determine whether observed differences are statistically significant or simply due to random variation. Statistical tests help assess performance metrics (such as RMSE, MAE, or correlation coefficients) across datasets or cross-validation folds. The process typically involves formulating a null hypothesis (stating that there is no difference between models) and an alternative hypothesis (indicating a difference). In hypothesis testing, the *p-value* represents the probability of observing results as extreme as the data, assuming the null hypothesis is true. A smaller p-value (often $p < 0.05$) suggests that the observed difference is unlikely to occur by chance, leading us to reject the null hypothesis in favour of the alternative hypothesis[258,259]. In Figure. 3.15, I show a comparison of three model performances using box plots, where the paired t-test[260] has been applied between each model pair. Observed p-values between model pairs indicate whether differences in performance metrics are statistically significant[258,259]. Below I present some commonly

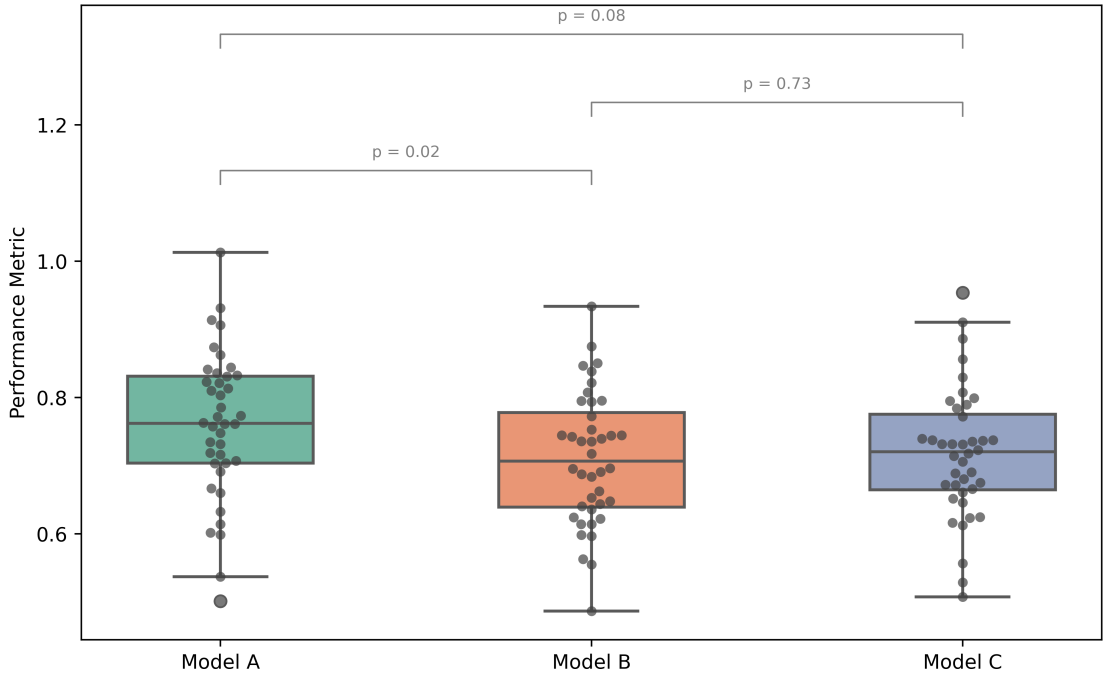


Figure 3.15: **Comparison of model performance metrics across three models using box plots with error bars.** The box plot illustrates the distribution of performance metrics for Model A, Model B, and Model C, with individual data points overlaid for reference. The error bars represent variability within each model's performance. Paired t-tests were conducted to assess the significance of differences between models, with p-values annotated above each comparison. Comparisons between Model A and Model B ($p = 0.02$) and between Model A and Model C ($p = 0.08$) suggest meaningful performance differences, with the Model A vs. Model B comparison reaching statistical significance ($p < 0.05$). However, the comparison between Model B and Model C ($p = 0.73$) indicates no significant difference, suggesting comparable performance levels for these models.

used statistical tests to evaluate and compare model performances, and for more detailed insights, refer to the following works[258–260].

Paired t-test[261] is applied to compare two related samples, such as predictions from two models on the same test set. The test assumes that the differences between pairs are normally distributed and is computed as,

$$t = \frac{\mu(d)}{s_d/\sqrt{n}}, \tag{3.53}$$

where $\mu(d)$ is the mean difference between paired observations, s_d is the standard deviation of these differences, and n is the number of pairs[261]. The test evaluates whether $\mu(d)$ is significantly different from zero[261].

Wilcoxon signed-rank test[258] serves as a non-parametric alternative to the paired t-test, applied when paired differences are not normally distributed. It ranks the absolute differences and sums these ranks for positive and negative differences[258]. The test statistic W is based on the smaller of these sums,

$$W = \min\left(\sum_{i:d_i>0} R_i, \sum_{i:d_i<0} R_i\right), \tag{3.54}$$

where R_i is the rank of the ith difference d_i[258]. The Wilcoxon test checks whether the distribution of differences is symmetric around zero.

McNemar's test[262] is a specific test for comparing two models' predictions on binary classification tasks. It calculates the significance of disagreement counts between models, focusing on cases where one model is correct and the other is incorrect[262]. The McNemar test statistic is given by

$$\chi^2 = \frac{(b-c)^2}{b+c}, \tag{3.55}$$

where b and c denote cases where the models disagree[262]. The test assesses whether these discrepancies are balanced, which could indicate performance differences[262].

For accurate comparisons across models, it is also essential to assess uncertainty around aggregate statistics, especially in limited datasets. *Bootstrapping*—resampling with replacement—or *repeated k-fold cross-validation* offer reliable methods to estimate confidence intervals around performance metrics[263]. By repeating tests on resampled data, these methods help generalize results, ensuring robust model evaluation and meaningful statistical comparisons across datasets and folds.

3.5 From History to State-of-the-art Machine Learning Methods for Binding Affinity Prediction

In this section, I will discuss the progression of machine learning in traditional QSAR models and then present various deep learning-based methods developed in the last

decade for predicting protein-ligand binding.

Advancements in QSAR with Machine Learning – More advanced ML techniques, such as *decision trees*, *random forests*, and *support vector machines (SVMs)*, introduced non-linearity into QSAR models. This enabled them to capture more complex patterns in the data[44,264,265], replacing the earlier regression-based models. Random forests, for example, are an ensemble learning method that aggregates the predictions of multiple decision trees to improve robustness and accuracy. Another significant advancement is the use of *neural networks* for QSAR modelling. Neural networks can learn non-linear mappings between input descriptors and biological activity, making them well-suited for capturing intricate dependencies in large and diverse chemical datasets[266]. Deep learning models, which consist of multiple layers of interconnected neurons, have been particularly effective in learning hierarchical features from molecular representations, leading to improved performance in tasks such as virtual screening and binding affinity prediction[267]. A recent example of a deep learning model such as *ChemProp*[268], uses a message-passing neural network (MPNN) that leverages the molecular graph representation of ligands to predict a wide range of chemical properties. It captures both atom-level and bond-level features, allowing it to model complex interactions within molecules more accurately than traditional QSAR methods. ChemProp has been applied to various QSAR tasks, including toxicity prediction, solubility estimation, and bioactivity prediction, showing improved performance over traditional methods[268]. Graph-based neural networks using D-MPNN (Directed Message Passing Neural Network)[268] and graph convolutional layer[269] architectures have set new benchmarks for predicting molecular properties by capturing the inherent structure-property relationships of ligands.

Overall, recent ligand-based models have benefited from both traditional QSAR and modern machine learning techniques, incorporating domain knowledge with data-driven approaches. Models such as *AutoQSAR* automate the descriptor selection process and model building, reducing the dependency on manual feature engi-

neering and improving reproducibility[43]. Additionally, techniques such as *transfer learning* have been employed to leverage pre-trained models for new tasks, further enhancing the applicability of ligand-based methods to a broader range of chemical and biological systems[270]. The evolution of ligand-based approaches, from classical QSAR models to modern ML-based methods, has significantly expanded the scope of computational drug discovery.

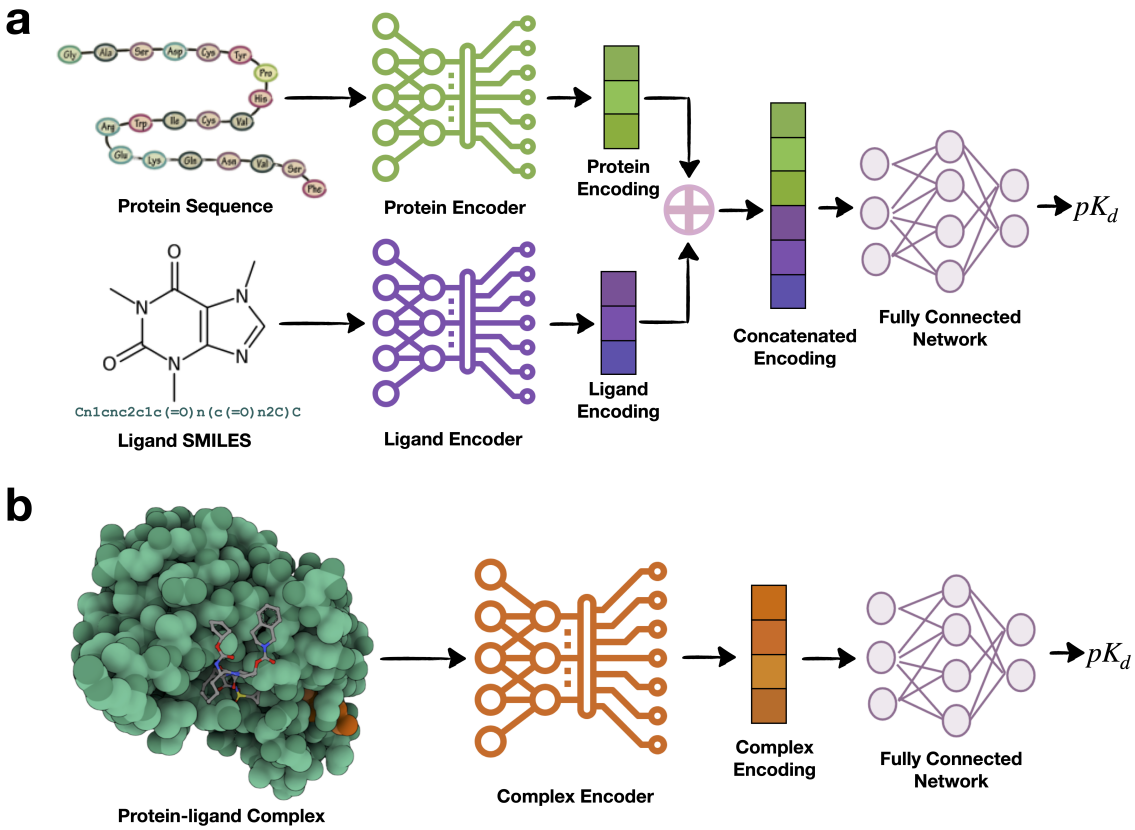


Figure 3.16: **Overview of deep learning architectures for binding affinity prediction based on input molecular representations.** (a) *Sequence and SMILES-based* methods in which the protein is represented by its amino acid sequence, and the ligand is represented using its SMILES notation. These representations are processed by separate encoders, generating respective feature embeddings for the protein and ligand. These embeddings are then concatenated and passed through a fully connected network to predict the binding affinity pK_d. (b) *Complex-based* methods take input as the protein-ligand complex, capturing the spatial interactions within the binding site. This complex is encoded via a dedicated encoder, producing a feature embedding of the protein-ligand interactions, which is subsequently processed through a fully connected network to predict the binding affinity pK_d.

Deep learning methods for protein-ligand affinity predictions can be categorized based on the type of input data into *sequence and SMILES-based* (1D)

models and protein-ligand *complex-based* (3D) models as shown in Figure. 3.16. An overview of these deep learning models providing the information on encodings, model architecture and datasets are summarised in Table. 3.1.

Complex-based models use various molecular representations to capture the intricate interactions between proteins and ligands from the input 3D complex. These representations include interaction fingerprints, atom coordinates, intermolecular contacts, 3D grids, and molecular graphs[44,271]. Interaction fingerprints denote protein-ligand interactions as bit strings, indicating whether specific interactions are present[272]. Another method involves embedding the protein-ligand complex into a 3D cartesian grid centred on the binding site, with each grid point representing the physicochemical properties of the interaction[273]. Atom pair representations focus on modelling the interaction energy based on atomic features and distances between atoms in the protein and ligand[274]. Convolutional neural networks (CNNs) are widely used with 3D grid-based representations. For example, models such as Pafnucy[275], KDEEP[273], and DeepAtom[276] use voxelized grids of the protein-ligand binding site, enabling CNNs to process these grids similarly to image data. Other CNN-based methods include OnionNet[274], which encodes protein-ligand interactions as atom pair contact profiles. These models are particularly effective for capturing spatial patterns in molecular data. Hybrid models such as the one developed by Derek et al.[277] combine 3D-CNNs and graph-based methods to capture complementary features of protein-ligand interactions.

Graph neural networks (GNNs) are utilized with molecular graph representations, where atoms serve as nodes and bonds or other interactions are edges. Models such as PotentialNet[278], GraphBAR[279], and GraphDTI[280] have advanced this approach by using GNNs to process the topological structure of the molecular complex, which naturally captures the relational data between atoms in both the protein and ligand. However, complex-based deep learning models face significant challenges. One issue is the tendency of deep neural networks to memorize patterns in the data, leading to overfitting and poor generalization to new, unseen datasets[54]. Despite

including explicit protein-ligand interactions, these models often need more improvement in prediction accuracy over simpler models that only consider the ligand or protein. Another challenge is data sparsity, particularly in diverse protein-ligand pairings, which hampers model performance[54]. These issues suggest that current complex-based approaches rely more on memorization than on learning the underlying physics of molecular interactions, necessitating model design and dataset diversity improvements to enhance generalization and predictive power[54,271]. Recently, Michael et al.[281] generated kinase complexes data via guided docking to overcome some challenges associated with current complex-based datasets. For a comprehensive review of the complex-based models, refer to Wang et al.[271].

DeepDTA[131], is one of the earliest *Sequence and SMILES-based* models. DeepDTA used CNNs to extract features from protein sequences, and SMILES and concatenated features are fed to fully connected layers to predict the affinity. PADME[282] uses Extended-Connectivity Fingerprints (ECFP) and graphs to represent drugs and Protein Sequence Composition (PSC) descriptors[283] to encode proteins. These descriptors capture information about the protein's sequence by analyzing the types and proportions of amino acids it contains. WideDTA[284] is an extension to DeepDTA. It differed from DeepDTA in the way SMILES and protein sequences were represented since it represented them as words instead of characters corresponding to an eight-character sequence and a three-residual sequence.

DeepAffinity[285] relies only on using the SMILES representation of drugs and the structural property sequence (SPS) representation. A recurrent neural network (RNN) encodes SMILES and protein SPS into embedding representations. GraphDTA[173] introduced graph representation to take advantage of the 2D structural information of the ligand graph. GraphDTA used a three-layer GCN as an alternative for ligand representation while keeping the CNN in the protein branch as in DeepDTA. DeepGS[286] used embedding techniques of Smi2Vec and Prot2Vec to exploit the chemical context within the drug SMILES and sequences, respectively. These embeddings were then combined with graph-derived features for bind-

ing affinity prediction. SAG-DTA[287] is similar to the GraphDTA method, except a more complicated graph representation of the drug molecule is introduced with a self-attention pooling mechanism into the network. DGraphDTA[174] is the first to introduce graph representation for proteins, while the earlier works used graph representations for ligands only. Recently, CAPLA[288] uses protein pocket information and attention mechanism for training these models. For a detailed review of these models, refer to these review articles [44,144,146]. As the focus of my thesis is on sequence and SMILES-based models, in the next chapter, I delve into understanding the information learnt by these models from the input proteins and ligands.

Table 3.1: Overview of deep-learning models for protein–ligand binding-affinity predictions

Model	Encoding	Architecture	Dataset(s)
Complex-based (3D)			
Pafnucy[275]	3D voxel grid (protein + ligand)	3D-CNN	PDBbind v.2016
KDEEP[273]	3D physicochemical grid	3D-CNN	PDBbind v.2016
DeepAtom[276]	Voxelized binding-site grid	3D-CNN	PDBbind v.2016
OnionNet[274]	Atom-pair contact profiles	2D-CNN	PDBbind v.2016
PotentialNet[278]	Molecular graph of complex	GNN	PDBbind v.2016
GraphBAR[279]	Distance-aware complex graph	GNN	PDBbind v.2016
GraphDTI[280]	Complex molecular graph	GNN	PDBbind v.2016
Hybrid 3D-CNN + GNN[277]	3D grid + molecular graph	3D-CNN + GNN	PDBbind v.2016
Sequence & SMILES-based (1D)			
DeepDTA[131]	Protein: sequence; Ligand: SMILES	1D-CNN + 1D-CNN	Davis, KIBA
WideDTA[284]	Protein: seq & domains/motifs; Ligand: SMILES & substructure	1D-CNN + 1D-CNN	Davis, KIBA
PADME[282]	Protein: PSC descriptors; Ligand: ECFP & graphs	FFNN + GNN	Davis, KIBA, Metz
DeepAffinity[285]	Protein: structural-property seq; Ligand: SMILES & graphs	RNN + CNN/GNN	Davis, KIBA, BindingDB
GraphDTA[173]	Protein: sequence; Ligand: molecular graph	GCN + CNN	Davis, KIBA
DeepGS[286]	Protein: Prot2Vec; Ligand: Smi2Vec + graph	Embedding + GNN	Davis, KIBA
SAG-DTA[287]	Protein: sequence; Ligand: graph	GNN + self-attention pooling	Davis, KIBA
DGraphDTA[174] **Davis, KIBA**	Protein: contact-map graph	PSSM; Ligand: graph	GNN
CAPLA[288]	Protein: pocket residues; Ligand: SMILES	1D-CNN + Attention	PDBbind v.2016

Chapter 4

From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction

This chapter is based on the work described in the following publication - Gorantla, R., Kubincova, A., Weiße, A. Y., & Mey, A. S. *J. Chem. Inf. Model.* **64**, 7, 2496–2507 (2024).

In this chapter, I examined the state-of-the-art deep learning-based approaches proposed in the literature up to 2022 and their shortcomings. I specifically analyzed models utilizing protein sequence and SMILES representations, investigating how these models learn from protein and ligand data to predict binding affinity. The research explored different protein and ligand embeddings and their impact on the accuracy of the downstream prediction task.

I conducted systematic experiments using kinase datasets[150,153] to assess various aspects of the commonly used prediction frameworks. Kinases are among the extensively studied targets in drug discovery due to their involvement in numerous diseases, including cancer, inflammatory, and autoimmune disorders[289,290]. Kinases catalyze the transfer of phosphate groups from ATP to the hydroxyl groups of serine, threonine, tyrosine, or histidine residues on themselves or other proteins, thereby modulating signaling pathways that control cell growth, differentiation, and sur-

vival[290,291]. The human genome encodes 518 protein kinases and as all kinases utilize ATP as a substrate, their ATP-binding sites are highly conserved in both sequence and structure, presenting a significant challenge for the development of selective kinase inhibitors[289–291]. As a result of the intense research focus on kinases, the volume of available biochemical data and structural data on kinases continues to grow supporting deep learning model development[290].

The investigation employed multiple approaches for protein representation, including convolutional neural networks for analyzing protein sequences (1D) and graph neural networks that incorporate structural information from contact maps (2D). The ligand component was analyzed using graph neural networks, with various perturbation experiments involving randomization of node and edge properties to better understand the model's learning behaviour. This ablation study highlighted shortcomings in embeddings and choices for combining embeddings and representing data. It also emphasized the need for better approaches to make binding affinity predictions more generalizable.[1]

[1]A correction to Figure 4 (originally published in *J. Chem. Inf. Model.* **64**, 7) has been appended to the manuscript in the thesis. This error does not affect the results, the figure caption reported and conclusions.

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim
Article

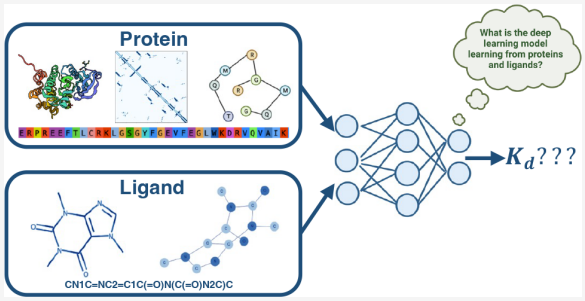
From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction

Rohan Gorantla, Alžbeta Kubincová, Andrea Y. Weiße, and Antonia S. J. S. Mey*

Cite This: J. Chem. Inf. Model. 2024, 64, 2496−2507
Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Accurate in silico prediction of protein−ligand binding affinity is important in the early stages of drug discovery. Deep learning-based methods exist but have yet to overtake more conventional methods such as giga-docking largely due to their lack of generalizability. To improve generalizability, we need to understand what these models learn from input protein and ligand data. We systematically investigated a sequence-based deep learning framework to assess the impact of protein and ligand encodings on predicting binding affinities for commonly used kinase data sets. The role of proteins is studied using convolutional neural network-based encodings obtained from sequences and graph neural network-based encodings enriched with structural information from contact maps. Ligand-based encodings are generated from graph-neural networks. We test different ligand perturbations by randomizing node and edge properties. For proteins, we make use of 3 different protein contact generation methods (AlphaFold2, Pconsc4, and ESM-1b) and compare these with a random control. Our investigation shows that protein encodings do not substantially impact the binding predictions, with no statistically significant difference in binding affinity for KIBA in the investigated metrics (concordance index, Pearson's R Spearman's Rank, and RMSE). Significant differences are seen for ligand encodings with random ligands and random ligand node properties, suggesting a much bigger reliance on ligand data for the learning tasks. Using different ways to combine protein and ligand encodings did not show a significant change in performance.



INTRODUCTION

In computer-aided drug discovery, being able to predict the binding affinity (BA) between a protein and a potential drug candidate is critical to identify new small molecules from large libraries. Accurate experimental screening for good binders is not practical for rapidly testing millions of drug-like compounds against potential protein targets.[1] Over the last four decades, many different approaches to in silico predictions for binding affinities have been developed. This encompasses both structure-based and ligand-based approaches;[2] however, each of them still has certain drawbacks when conducting a large-scale screening of compound libraries against a certain protein target. For example, docking[3,4] methods can be used to screen large libraries, but often the desired accuracy for a BA is not achieved. On the other hand, alchemical free energy-based affinity prediction techniques[5−7] are more accurate, but computationally costly for the discovery of hits in ultralarge libraries.[8] Both through the rapid development of new machine learning methods and better availability of binding affinity data, e.g. through PDBbind,[9] KIBA,[10], and Davis,[11] many different efforts have been explored to generate ML-based methods for BA.[12,13]

In this paper, we will look at some of these machine learning (ML) models for binding affinity predictions more closely to gain insights on how components of these models contribute to the performance of the binding affinity prediction task. Depending on the type of input data used during training, these deep learning (DL) methods can be broadly categorized as sequence- or complex-based methods.[2] Complex-based methods[14−20] are trained on features from 3-dimensional (3D) protein−ligand complexes. Here we focus on sequence-based methods.

Sequence-based approaches try to learn from Simplified Molecular Input Line Entry System (SMILES) strings and one-dimensional (1D) protein sequences. This can either be in the form of language models[21] or converting SMILES and protein sequences to graphs, leveraging 2D connectivity information from these graphs.[22,23] The 1D and 2D-based DL models extract the features from the sequence and SMILES string and the feature vector formed by concatenating encoded protein and ligand features is used to get to the BA prediction (Figure 1). Zhao et al. compiled a comprehensive overview of deep

Special Issue: Machine Learning in Bio-cheminformatics

Received: August 2, 2023
Revised: October 26, 2023
Accepted: October 27, 2023
Published: November 20, 2023

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

Journal of Chemical Information and Modeling pubs.acs.org/jcim Article

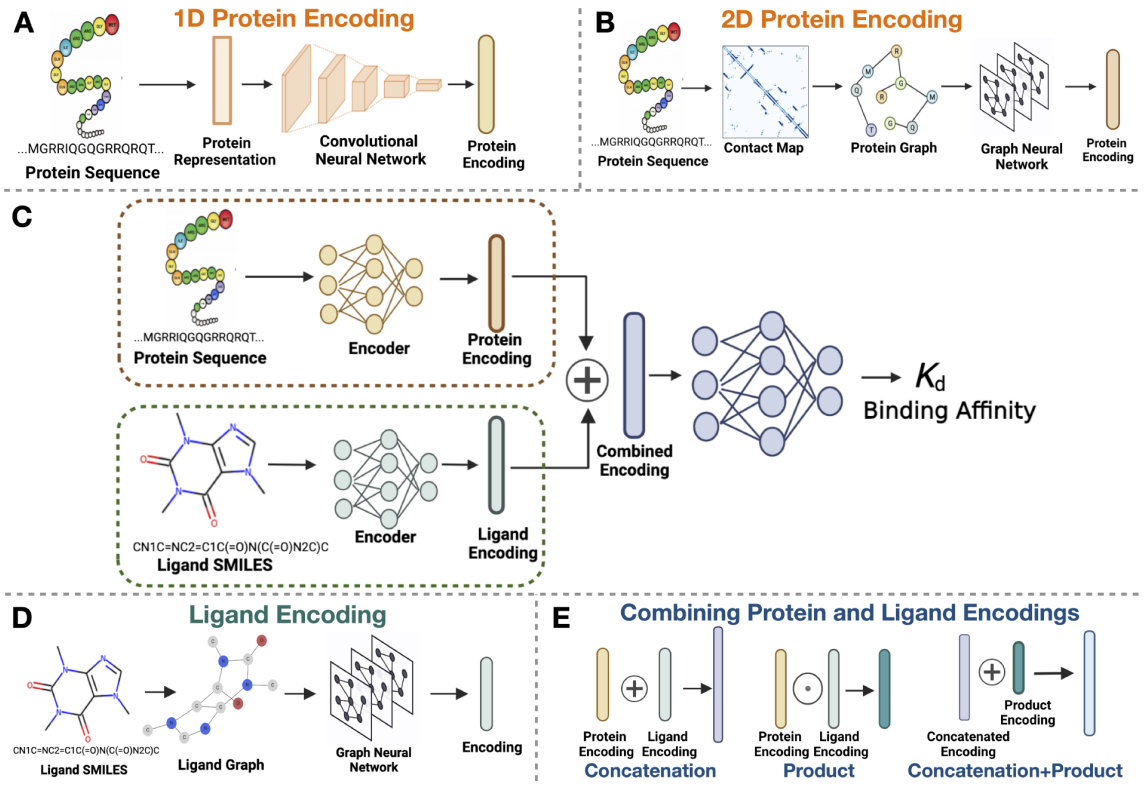


Figure 1. Systematic assessment of protein and ligand encodings on a deep learning framework for protein−ligand binding affinity predictions. A: 1D protein representation is obtained from the input sequence and then passed through a CNN module to obtain the protein encoding. B: 2D protein encoding where an intermediary step of contact map prediction is required for the protein graph generation to obtain structural information from the sequences. The generated graphs are passed to graph neural networks to extract features and obtain the protein encodings. C: Overview structure of the DL framework used for this investigation. The DL framework processes the input sequence and SMILES data using 1D or 2D data structures to form their respective encodings. These encodings are combined and passed to a fully connected neural network for binding affinity prediction. D: The input SMILES string is converted to a 2D graph and processed through the graph neural network to obtain the ligand encoding. E: Combination of protein and ligand encodings, namely concatenation, element-wise product, and concatenating the vectors from protein−ligand encoding concatenation and element-wise product.

learning-based protein−ligand interaction prediction ML-based methods,[13] which provides a useful starting point. We will take a closer look at some of the examples from this review, as our investigations focus on DL architectures from these examples.

Öztürk et al.[24] proposed DeepDTA, one of the earliest sequence-based methods using CNNs to extract 1D sequence information on the protein and ligand SMILES. WideDTA[25] extended DeepDTA by incorporating additional information sources, such as protein domains and motifs, and ligand maximum common substructure words. SMILES strings are a linearized representation of a ligand graph capturing structural, geometric, and topological properties. Jiang et al.[23] introduced a more rational approach to utilize the information from the 2D contact map predicted by a supervised deep learning method, Pconsc4,[26] as the representation of the tertiary structure and have demonstrated improvements in binding affinity performance. These contact maps capture the details of residue−residue interactions and can be naturally modeled as graphs. All of these methods are trained and evaluated using publicly available kinase data sets.[10,11] There are also other sequence-based DL methods[27−31] that have similar architectures to that of Jiang et al.[23]

In this paper, we systematically investigate sequence-based DL models, primarily CNN and G(C)NN-based architectures, to understand how these model architectures learn from information presented to them through different encodings of protein sequences and ligand SMILES string. Specifically, we test ligand and protein encodings in 1D and 2D as summarized in Figure 1. For protein encodings, we look at 1D encodings obtained from sequences (Figure 1A) and 2D protein encodings obtained from contact maps (Figure 1B). For the 1D encodings, we compare the Evolutionary Scale Modeling (ESM-1b) language model[32] to the performance of hand-crafted Kinase−Ligand Interaction Fingerprints and Structures (KLIFS) data using a one-hot encoding of the identified binding sites[33] on the downstream binding affinity prediction task. To test 2D encodings that rely on contact maps we use four different contact map prediction methods: protein sequence,[32] homology information derived from multiple sequence alignment,[26] and 3D structures[34] predicted through AlphaFold2. Lastly, we use a random contact map as a control. To study the impact of ligands on the DL framework, the input SMILES string is transformed into a graph structure and then processed using a GNN to obtain its encodings, as shown in Figure 1D. By looking at various perturbations of the ligand graphs, we can evaluate the effect on the downstream binding

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

affinity prediction task. The last point of investigation is how the ligand and protein encodings are concatenated and how this may affect any binding affinity prediction Figure 1E. All experiments were carried out on the Kinase inhibitor bioactivity (KIBA)[10] and Davis[11] data sets as outlined in the methods section. Overall, we found that current architectures do not make much use of the protein data shown in these typical CNN and GCN architectures as presented in the results and discussion section.

METHODS

Data Sets. We used two kinase data sets, Davis[11] and Kinase inhibitor bioactivity (KIBA),[10] which are common benchmark data sets for the evaluation of how well DL models perform at binding affinity prediction tasks. Davis comprises selectivity assays of 442 kinases and 68 inhibitors, with measurements for the inhibitor's dissociation constants. These values were transformed into logarithmic space, consistent with prior studies.[23,24] Higher pK_d values mean higher affinity. From Figure 2B, it can be seen that there is a skew toward nonbinders in this data set.

The other data set, KIBA, amalgamates various sources of bioactivity data into a single KIBA score, optimizing consistency across different measures (K_i, K_d, and IC_{50}), with lower scores implying a stronger binding affinity. The KIBA data set was originally composed of 467 targets and 52,498 small molecules; however, He et al.[35] filtered it to contain only small molecules and targets with at least 10 observations yielding a total of 229 unique proteins and 2111 unique small molecules. This filtered data set was used to benchmark earlier BA prediction methods.[22−24] We filtered both the Davis and KIBA data sets further, to only include kinases with sequence lengths less than or equal to 1024 residues, as some of the protein-encoding techniques we used are limited to sequence sizes of up to 1024. Figure 2 summarizes the distribution and properties of both the Davis and KIBA data sets. Similarly to previous studies,[22−24] we randomly divided each data set into six roughly equal parts, using 5/6 for training and validation, and the remaining data for testing.

From Features to Encodings for Ligands and Proteins. To assess the performance of the BA learning task, we tested different encodings for ligands and proteins. We used graph-based approaches for ligands, and for proteins we used 1D features and 2D graph-based encodings, which are outlined in detail below.

Protein Representations for Generating 1D Encodings. We tested two protein representations to study the effect of 1D encodings, Kinase−Ligand Interaction Fingerprints and Structures (KLIFS) and Evolutionary Scale Modeling (ESM-1b).

ESM-1b[32] is a protein language model based on a Transformer-34[36] architecture trained on more than 220 million (unaligned) sequences from UniProt[37] through masked language modeling objective. During training, the transformer model is presented with protein sequences where a subset of their residues are masked, either by a random permutation to a different amino acid, by leaving them unmodified, or through a fraction of residues being masked. The objective of the model is to predict the values of the masked residues by considering the context of all unmasked residues in the input. We implemented the ESM-1b[32] model using the fair-esm Python package, and the representations were obtained using the esm1b_t33_650M_UR50S() model.

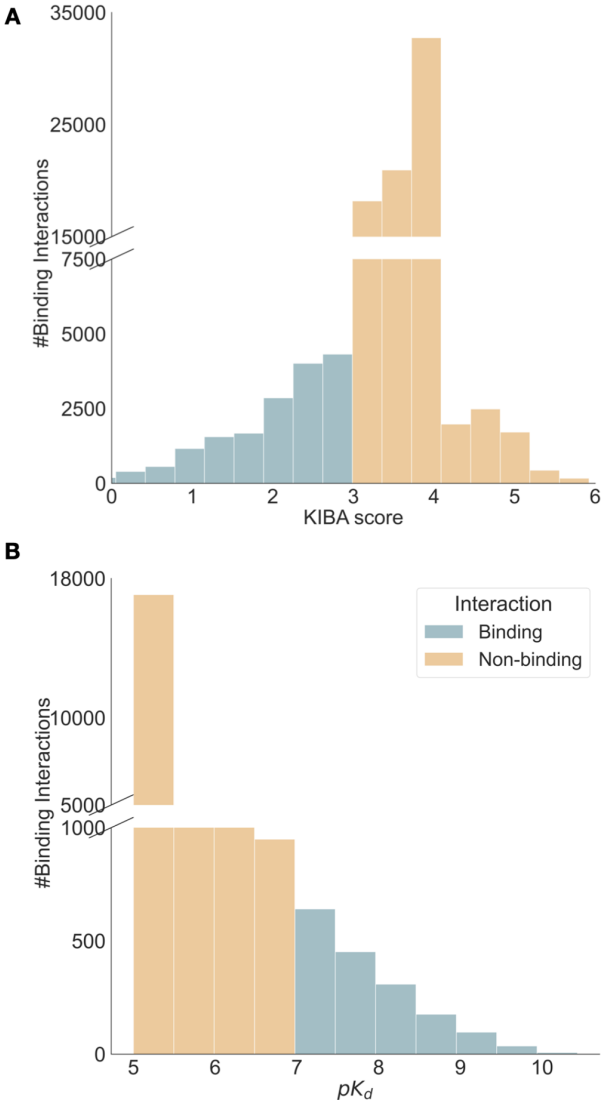


Figure 2. Summary statistics of KIBA[10] and Davis[11] data sets. KIBA has 188 proteins, 2111 ligands, and 95,577 binding interactions, while Davis contains 333 proteins, 68 ligands, and 22,644 binding interactions. A: Distribution of KIBA score across the entire data set. Lower KIBA score denotes higher binding affinity (≤3). B: Distribution of pK_d scores across the Davis data set. Higher pK_d indicates a higher binding affinity with (>7) usually seen as a binder.

KLIFS provides information on how kinase inhibitors interact with their targets.[33] It provides a consistent alignment of 85 kinase ligand binding site residues that enables the identification of family-specific interaction features and the classification of ligands according to their binding modes. We leverage the 85 kinase ligand binding site residues for each kinase in the data sets using either Gene Name or UniprotID query. The 85 residues obtained for each kinase are then one-hot encoded; each residue is encoded to one of the 20 amino acids or a gap. The feature vector obtained from either KLIFS or the ESM-1b model is used as input to the convolutional neural network (CNN) module used for 1D encodings (see Figure 1A).

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

Contact Maps for Protein Graph Generation and 2D Encodings. Protein's 2D encodings are generated by means of a protein contact map. A protein contact map is a graph representation of a protein with an adjacency matrix containing information on which amino acids in the protein chain are in contact or not.[38,39] Protein graphs $G_p = (\mathbf{N}_p, \mathbf{M}_p)$ are generated from the contact maps using input protein sequence with L_p residues. A pair of residues is said to be in contact or linked whenever the Euclidean distance $(d_{i,j})$ between their C_α atoms is less than or equal to a threshold d_c. These connections can be determined from a 3D structure of a protein or predicted by means of other computational methods.

$$M_p^{ij} = \begin{cases} 0, & \text{if } d_{i,j} > d_c \text{ or } i = j \\ 1, & \text{if } d_{i,j} \le d_c \end{cases} \quad (1)$$

Eq 1 summarizes the entries of the adjacency matrix $\mathbf{M}_p$ of the protein. In addition, each node in the adjacency matrix has certain node features which are represented by a matrix $\mathbf{N}_p \in \mathbb{R}^{L_p \times 54}$. The 54-dimensional feature vector for each residue node is computed, for each of the L_p amino acids with a summary of computed features presented in Table S1.

While the node features in the protein graph are kept constant, the contact map, i.e., the underlying adjacency matrix, is computed using four different ways to evaluate this. The first contact map prediction method used to obtain a protein graph is Pconsc4.[26] It is a supervised DL method with a U-net architecture[40] trained on a curated data set with 2791 proteins from PDB and benchmarked on two data sets without homology to the training set.[26] It uses a 72-dimensional feature vector computed from multiple sequence alignment as input. The output of Pconsc4 is the probability of whether there is a contact between two pairs of amino acids, then a threshold of 0.5 is set to obtain the contact map, as proposed in the original paper.[26] The final contact map has a shape of $(L_p \times L_p)$, where L_p is the number of nodes (residues or amino acids). This method was originally used by Jiang et al.[23] in their DL framework for binding affinity prediction.

The next method for obtaining a contact map is using data from an AlphaFold2 structural[34] model. AlphaFold2 predicts the 3D coordinates of all heavy atoms for a given protein using the amino acid sequence and aligned sequences of homologues as inputs. Here, we used the AlphaFold2 protein structure database[41] to get the 3D structures for each of the proteins used in the KIBA and Davis data sets. We downloaded the 3D structures in PDB format and used MDAnalysis version 2.0.0[42] and NetworkX version 2.8.4[43] to compute the contact map from the 3D structure. The pairwise C_α distances were calculated for each given structure, and two residues were said to be in contact if their distance was less than 8 Å.[38,39]

We also used ESM-1b,[32] as discussed earlier, for extracting 1D protein encodings to obtain a contact map. The ESM-1b model predicts the contacts between residue pairs from the input protein sequence only. It learns the tertiary structure of a protein sequence in its attention maps during the unsupervised training on UniProt[37] data. The contact map predictions were made using the esm1b_t33_650M_UR50S() model by calling the model.predict_contacts() method using the default threshold. At the time of our data collection for this study, ESM-1b was the most recent model available. However, it is worth mentioning that it has since been superseded by the release of the ESM-2.[44]

Randomly generated contact maps are used as a control method for studying the effect various contact map methods will have on protein graph encodings and, in turn, the binding affinity prediction. To generate random contact maps, we first generate a random protein sequence with randomly selected amino acid residues of the same length as the input protein sequence. The random sequence string is then used to get residue−residue contacts using the ESM-1b[32] model in a similar way as described above. Using the contact information from each of discussed methods, we then compute the adjacency matrix $\mathbf{M}_p$ to build the protein graph.

Ligand Encodings and Their Perturbations. The ligands are represented as graphs derived from a linearized version of their chemical structure represented as SMILES strings. Ligand encodings are then obtained from these graph representations. Ligand graphs $G_l = (\mathbf{N}_l, \mathbf{M}_l)$ are generated from the input SMILES string with L_l atoms, where $\mathbf{M}_l \in \mathbb{R}^{L_l \times L_l}$ is an adjacency matrix with information about the chemical bonds present between any given pair of atoms. Self-loops are added to the graph construction, i.e., the diagonal of the adjacency matrix is set to one to improve the feature performance of the molecule.[22,23] By adding self-loops, each node can incorporate its own features during the convolution operation, ensuring that its own information is retained and not solely influenced by its neighbors. This is particularly crucial for nodes with fewer connections, ensuring they do not lose their inherent feature information during the convolution process.[22,23] Each ligand node in the graph is denoted by a 78-dimensional feature vector similar to Jiang et al.[23] capturing the one-hot encoding of the atom type, degree of the atom, total number of hydrogens bound to the atom, number of implicit hydrogens bound to the atom, and whether the atom is aromatic or not. We processed the SMILES string with the RDKit version 2020.09.5[45] library using Chem.MolFromSmiles() to get the atom and bond details for building the ligand graph.

To look at the effects of the ligand encodings on the downstream task, we designed three different ways to perturb the ligand graphs. The first randomization technique, which we call *Point randomization*, generates a new SMILES string with minor changes to the original one, thus altering the ligand graph slightly. This involves identifying specific atoms (such as Cl, F, Br, and (=O)) and making selective changes to up to four atoms, such as substituting halogens or eliminating a (=O) atom. If these enumerated atoms are absent, a Cl atom is prefixed. This checks the model's sensitivity to minor ligand structure changes. Detailed insights on point randomization, along with Algorithm 1, are in the SI. The second technique, *Node feature randomization*, assesses the impact of node features in the model's predictions. We randomly permute the node feature values across the graph, thus disrupting the nodes' identities while preserving the graph's structure. The degree of performance change following this randomization will indicate the model's dependency on node features versus the graph structure for its predictions. The third method, *Random sampling*, represents an extreme level of randomization, where the original ligand graph is substituted with a randomly selected ligand graph from the same data set. This approach enables us to evaluate whether our DL model relies on ligand features for binding affinity prediction. By training the model with a randomly selected ligand, we can ensure that we are not generating chemically implausible ligands.

Deep Learning Architecture. To combine information from our ligand encodings and protein encodings in a deep

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

learning architecture, we borrow ideas from the architecture proposed by Jiang et al.[23] We use a module for protein encodings and one for ligand encodings. For the 1D protein encodings, we use a three-layer CNN model (Figure S1). For the 2D graph, approaches used for both ligand and protein a GNN model with three graph convolutional network (GCN) layers similar to Jiang et al.[23] are used (Figure S2). The GCN model learns the representation for a given input graph $G = (\mathbf{N}, \mathbf{M})$, where $\mathbf{N} \in \mathbb{R}^{v \times q}$ is the matrix containing v nodes and each node is represented by a $\mathbf{q}$ dimensional feature vector. $\mathbf{M} \in \mathbb{R}^{v \times v}$ is the adjacency matrix that provides the structural information on the graph. The features are extracted from the graph via GCN layers, where each layer will perform a convolution operation by following the propagation rule[46] defined below

$$\mathbf{H}^0 = \mathbf{N}$$
$$\mathbf{H}^{l+1} = \sigma(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{M}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}^l\mathbf{W}^l) \quad (2)$$

Here, $\mathbf{H}^l$ and $\mathbf{W}^l$ denote the l^{th} GCN layer outputs and its corresponding learnable parameters, respectively. The adjacency matrix, $\tilde{\mathbf{M}}$, with self-loops in each node, i.e., $\tilde{\mathbf{M}} = \mathrm{M} + \mathbf{I}$, where $\mathbf{I}$ is the identity matrix and $\tilde{\mathbf{D}}$ is the diagonal node degree matrix calculated from $\tilde{M}$, $\tilde{\mathbf{D}} = \sum_j \tilde{\mathbf{M}}_{ij}$. The design of the $\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{M}}\tilde{\mathbf{D}}^{-1/2}$ term is intended to add a self-connection to each node and keep the scale of the feature vectors. $\sigma(.)$ represents a nonlinear activation function, Rectified Linear Unit (ReLU).

Experimental Setup for Model Training and Analysis. The detailed DL architecture is outlined in the SI. Figure S1 summarizes the CNN architecture used for the 1D protein and ligand encodings with the 1D protein encoding making use of the highlighted CNN module. Figure S2 contains a summary of the architecture used for the GCN of the 2D protein−ligand encodings. These protein graphs G_p and ligand graph G_l derived encodings are obtained using the GCN module. Both the CNN and GCN modules are implemented with PyTorch and PyTorch geometric. We use the Mean Squared Error (MSE) loss function to train the DL model. Experiments testing combinations of ligand and protein encodings in 1D and 2D are summarized in Table S2. Each experiment trains the DL model for 2000 epochs with batch size 128 and learning rate $\beta = 0.001$ using the Adam optimizer, saving the top-performing model from the validation set. To ensure the robustness of our experiments, we randomly selected three deep learning models trained on three different folds from the training split. These models were then used for bootstrap resampling on a randomly selected sample size of 1500 data points from the test set. The mean predictions of the trained models for each bootstrap iteration are used to compute the evaluation metrics and their associated errors. We use Concordance Index (CI), Root Mean Squared Error (RMSE), Pearson correlation, and Spearman rank correlation to assess the model's performance. All code and models are accessible at https://github.com/meyresearch/DL_protein_ligand_affinity.

RESULTS AND DISCUSSION

Different Protein Contact Map Prediction Methods Provide Different Protein Graphs for Protein Encodings. We computed protein contact maps (PCM) for the KIBA and Davis data sets, as outlined in the methods section using structural data from AlphaFold2, the sequence-based method ESM-1b, and the homology modeling tool Pconsc4. To obtain a baseline idea of how well these methods correlate to experimentally determined structure-derived PCMs, we manually curated 50 protein structures with structural data available in the RCSB protein data bank (PDB) spanning across the kinase data sets KIBA and Davis. These structures were randomly selected based on their sequence length matching the corresponding kinase in the Davis or KIBA data set, ensuring a representative and unbiased sample. We used either Uniprot ID or Gene ID to identify the structures from the PDB. Contact maps from PDB data were computed in the same way as AlphaFold2 PCMs, but used as a reference. To evaluate the performance of contact map prediction methods, we used the F1 score, Matthews' correlation coefficient (MCC),[47] and precision metrics. MCC is a balanced metric that considers the distribution of true positives, false positives, true negatives, and false negatives in a binary classification problem, making it a suitable metric for evaluating models in cases of class imbalance; we provide more details about the metric in the SI. Figure 3A shows an example of a PCM obtained from PTK-6 with Uniprot ID: Q13882 and PDB ID 5D7 V with 8 Å threshold and highlight the true contacts (turquoise squares), falsely predicted contacts (pink circles), and lost contacts (orange crosses), that is, those that were present in the 3D X-ray structure contact map but not present in the predicted one. For more examples, see the SI.

Figure 3B shows violin plots that compare PCM generation methods and their performance according to MCC, F1, and Precision. The contact map predictions from AlphaFold2 structures had the highest mean MCC and standard deviation of 0.54 ± 0.21 and F1-score of 0.55 ± 0.22, while the ESM-1b method had the lowest average MCC (0.07 ± 0.04) and F1-score (0.09 ± 0.04). The contact map prediction results of Pconsc4 (MCC: 0.51 ± 0.16, F1-score: 0.51 ± 0.17) are comparable to that of the AlphaFold2; however, the mean precision of Pconsc4 (0.59 ± 0.25) method is slightly better than AlphaFold2 (0.55 ± 0.22) contact maps on the curated PDB data set. Using the Wilcoxon signed-rank test, we evaluated the performance of the contact map prediction methods on the 50 experimentally determined structures. In the Wilcoxon signed-rank test, the null hypothesis is that there is no significant difference between the performance of the two models. Generally, a p-value less than or equal to the significance level of 0.01 is considered statistically significant, leading to the rejection of the null hypothesis. On comparing the method predictions on MCC, F1-score and precision for each pair of methods we observed a small p-value ($p < 0.01$), indicating strong evidence against the null hypothesis. Thus, there is a significant difference between each pair of the contact map prediction methods used in this study, and the contact map predictions obtained from each method are not the same. ESM-1b appears to be the least reliable and accurate method for predicting contact maps, whereas AlphaFold2 and Pconsc4 exhibit almost biomodal distributions for MCC and F1-score. To understand the reliability of protein structures used to obtain AlphaFold2 contact maps, we computed the average confidence score of AlphaFold2 structures per residue in the Davis and KIBA data sets. The average confidence score per residue for KIBA was 78.2 ± 21.52, while for Davis, the score was slightly higher at 88.82 ± 22.35. Removing low confidence score (<70) AlphaFold2 structures from the set of 50 hand-curated X-ray structures does not remove the bimodal

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

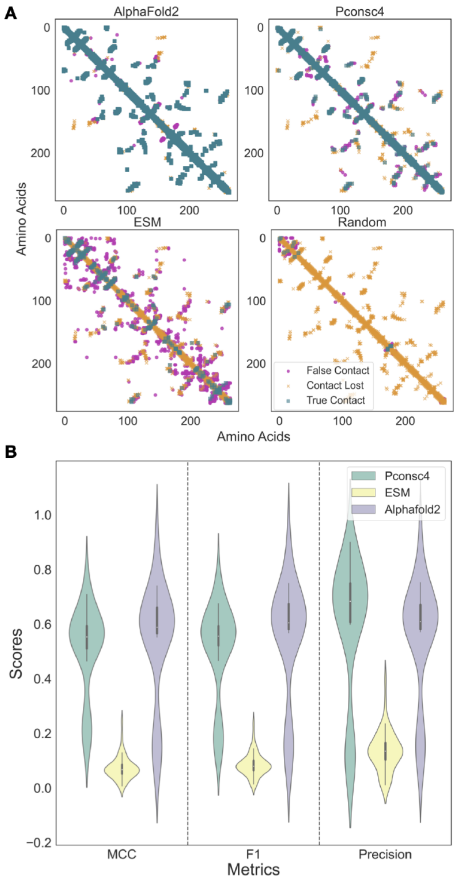


Figure 3. Contact maps obtained from each contact map prediction algorithm (ESM-1b, AlphaFold2, and Pconsc4) are different and provide significantly different protein graphs to the protein encodings. A: Contact map analysis for PTK-6 using PDB ID 5D7 V as a reference, top left AlphFold2, top right Pconsc4, bottom left ESM-1b, and bottom right a random contact map. True contacts are displayed in turquoise squares, lost contacts are shown in orange crosses, and falsely predicted contacts in pink circles. B: Assessing contact map methods on a curated data set of KIBA and Davis protein structures shows that the AlphaFold2 contact maps perform better on MCC, and F1 score metrics, while the Pconsc4 contact map prediction method has higher mean precision. ESM-1b contact predictions are the least reliable.

distribution in MCC, F1, and Precision for AlphaFold2 and Pconsc4 and other underlying factors beyond the scope of this paper give rise to this.

Protein Encodings Based on Significantly Different Protein Graphs Do Not Have Much Effect on Binding Affinity Prediction. Keeping the DL framework fixed, as described in the methods and shown in Figure 1C, we only test the four different contact map generation methods. With four different ways to generate protein graphs established, we want to assess if the protein graph structure has any impact on the downstream binding affinity prediction task when we use the PCM in our protein encoding. We will refer to these as 2D encodings as we are generating graphs with nodes and edges and interpreting them as 2D structures. The ligand encodings are untouched and based on the DL-framework from Jiang et al.[23] The downstream task of estimating binding affinities is evaluated on both the KIBA and Davis data sets.

Figure 4A (KIBA) and 4B (Davis) summarize the findings of changing the PCM generation methods. From Figure 4A and 4B, we can observe that there is not much change in the performance of the DL model with different protein encodings across all four evaluation metrics, i.e., CI, Pearson correlation coefficient, RMSE, and Spearman rank correlation on the test set. On the KIBA data set (Figure 4A), ESM-1b had the lowest RMSE on the test set, with 0.468 ± 0.02, followed by Pconsc4 with 0.475 ± 0.02, and Random and AlphaFold2 with 0.480 ± 0.03. Pearson's correlation between experimental and predicted binding affinity score for ESM-1b and Pconsc4 was 0.82 ± 0.02, while Random and AlphaFold2 had 0.81 ± 0.02. The experiments on the Davis data set in Figure 4B show that the random encoding (CI: 0.86 ± 0.01, Pearson: 0.79 ± 0.02, RMSE: 0.51 ± 0.02) has slightly lower performance than Pconsc4 (CI: 0.89 ± 0.01, Pearson: 0.82 ± 0.01, RMSE: 0.48 ± 0.02), ESM-1b (CI: 0.89 ± 0.01, Pearson: 0.82 ± 0.01, RMSE: 0.47 ± 0.02), and AlphaFold2 (CI: 0.88 ± 0.01, Pearson: 0.82 ± 0.02, RMSE: 0.49 ± 0.02), while there is no change among the rest of the methods. Overall, we saw that the performance of Random encoding appears to be comparable to other PCM methods on the KIBA data set (Figure 4A). However, for the Davis data set, we saw a slight drop in performance with the random encoding.

Using the Wilcoxon signed-rank test, we evaluated the performance of the trained DL models on the bootstrapped test set. The Wilcoxon signed-rank test evaluates whether there's a meaningful difference between two models' performances, and a p-value of 0.01 or less generally suggests this difference is statistically significant, refuting the original assumption of no difference. The KIBA data set shows no significant difference ($p > 0.01$) in the performance of AlphaFold2, ESM-1b, Pconsc4, and Random models in terms of Pearson, Spearman, and RMSE metrics. The overall performance of all metrics is better on the Davis data set, with no significant difference ($p > 0.01$) between AlphaFold2 and Pconsc4. However, ESM-1b has a significantly better performance on all metrics ($p < 0.01$) on the Davis data set. We also observe a significant drop in performance with random encodings on the Davis data set. This performance drop is higher than for KIBA as both these data sets have different proportions of proteins and ligands (Davis: 333 kinases and 68 ligands, KIBA: 188 kinases and 2111 ligands).

Next, we looked at the overall correlation of either KIBA score or pK_d predictions for each molecule in the test set. We arbitrarily picked Pconsc4 as a reference to compare the predictions with various encoding methods (Figure S5). On the KIBA data set, all the PCM methods compared, exhibited an R^2 of 0.94 ± 0.04. Meanwhile, the R^2 values spanned from 0.87 to 0.95 for the Davis data set. From Figure S5, we observed each pair of methods exhibiting a strong correlation in their binding affinity predictions. Figure S6 shows the correlation between experimental and predicted binding affinities along with the kernel density estimate (KDE) of the prediction distributions for all four 2D encoding methods. For the KIBA data set, the R^2 for all the methods is close to 0.71 ± 0.01, while for Davis, the R^2 for ESM-1b, Pconsc4, and AlphaFold2 is 0.72 ± 0.01 and the R^2 for random is 0.69 ± 0.02. We used the Jensen−Shannon (JS) divergence to compare the prediction distributions of each pair of methods for both KIBA and Davis data sets. JS divergence serves as a symmetric measure, quantifying the similarity between two probability distributions. The values of JS divergence are

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

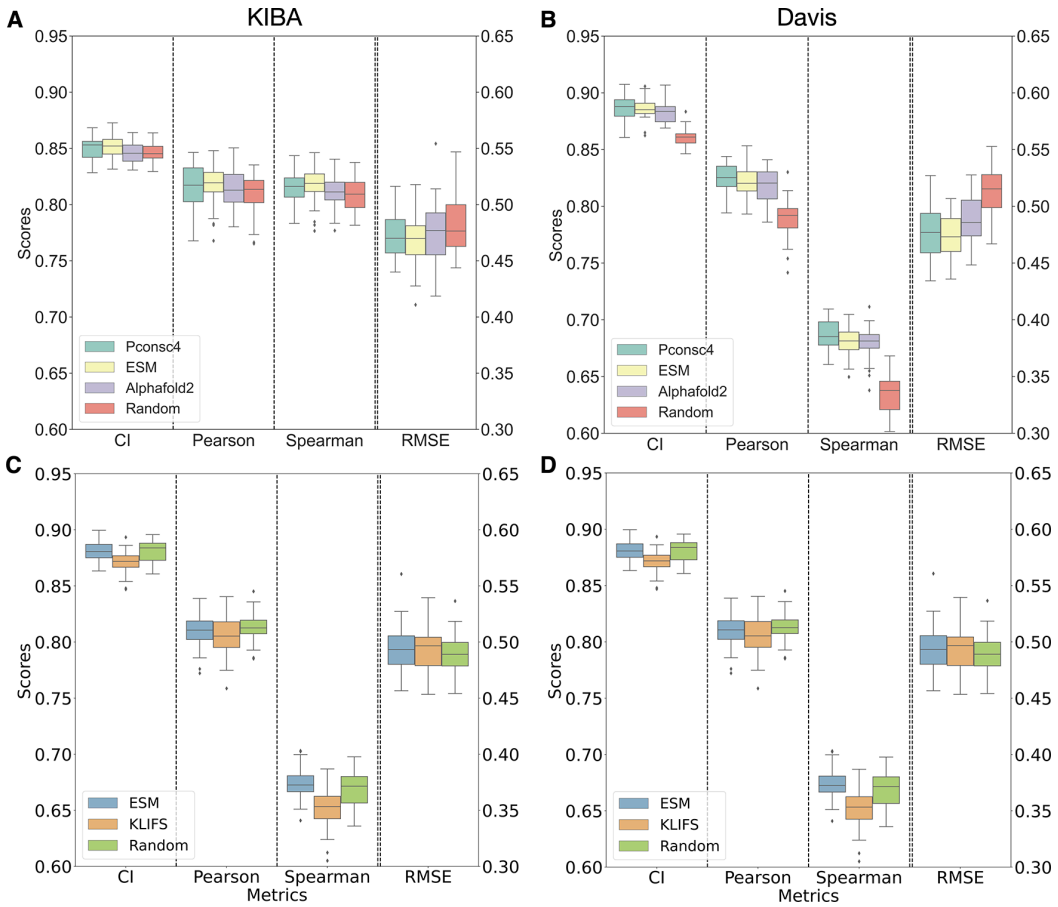


Figure 4. Protein encodings (2D) with structural information from contact maps do not have much effect on binding affinity prediction, and 1D encodings from protein language models (PLM) perform similarly to contact maps enabled encodings. A, B: Boxplots for different performance measures (CI, Pearson correlation, Spearman Rank, and root-mean-square error) of binding affinity predictions for the KIBA data set (A) and Davis data set (B) for four different protein contact map methods. This shows that the structural information from protein contact maps encoded into a graph is not making any significant contribution to DL model performance. C, D: Boxplots for three different 1D encoding methods and their performance metrics (CI, Pearson correlation, Spearman Rank, and root-mean-square error), the PLM encodings of the ESM-1b model perform better than one-hot encodings from KLIFS handcrafted sequences on both the data sets and are comparable to random encodings. Overall, the performance of 1D encodings is comparable to the encodings that include information from protein graphs.

bounded between 0 and 1, where 0 signifies identical distributions and 1 denotes entirely distinct distributions. For the KIBA data set, the mean JS divergence across all method pairs was 0.0008, while for the Davis data set, it was 0.0031. While there is a difference in these values, both are relatively close to 0, indicating that the prediction distributions from each method are notably similar. However, given the scale and nature of our data sets, we consider these values as indicative of comparable prediction distributions. This observation underscores that there is not a substantial variation in model predictions when using encodings generated from markedly different contact maps. In a broader perspective, our results suggest consistent performance across methods such as AlphaFold2, Pconsc4, and ESM-1b on both data sets, while the DL model trained with random contact maps showed a slight drop in performance on the Davis data set.

Encodings from Protein Language Models Outperform Handcrafted Encodings in Predicting Binding Affinity and Perform Similarly to 2D Protein Encodings. Given that different contact maps, including random maps, show little impact on the accuracy of the binding affinity prediction, we next use 1D encoding methods to investigate the importance of structural details in the DL model's prediction capabilities. To this end, we use 1D encodings obtained from ESM-1b, handcrafted sequences that identify binding regions explicitly as contained in KLIFS sequences,[33] and a control encoding generated from a random sequence of the same length. The results obtained from these three different 1D encodings are presented in Figure 4C and 4D. Figure S7 shows a Euclidean distance heatmap of the ESM-1b embeddings capturing variance among the proteins in both KIBA and Davis data sets.

Figure 4C and 4D show that the ESM-1b encodings-based model performs better than the one-hot encoding using KLIFS sequences for both KIBA and Davis data sets. For the random sequence control encoding, the performance is comparable to those of ESM-1b and KLIFS. For KIBA, the ESM-1b-based encoding (CI: 0.84 ± 0.01, Pearson: 0.81 ± 0.01, RMSE: 0.48 ± 0.02) is performing better than KLIFS-based one-hot encodings (CI: 0.81 ± 0.01, Pearson: 0.77 ± 0.01, RMSE: 0.51 ± 0.02). The Wilcoxon signed-rank test on both these encodings on the KIBA data set shows that the change in

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

Journal of Chemical Information and Modeling pubs.acs.org/jcim Article

performance is significant ($p < 0.01$) across all the metrics. From Figure 4D, the performance of both ESM-1b (CI: 0.88 ± 0.01, Pearson: 0.81 ± 0.01, RMSE: 0.49 ± 0.02) and KLIFS (CI: 0.87 ± 0.01, Pearson: 0.81 ± 0.01, RMSE: 0.49 ± 0.02) for the Davis data set is comparable. The Wilcoxon test shows that the change in performance is significant on CI and Spearman metrics with $p < 0.01$, while on Pearson and RMSE, the change is not significant ($p > 0.01$). We can see that both the rank correlation metrics CI and Spearman have seen a significant performance change between PLM encodings and manually curated sequence-based encodings. Further, from our 1D encoding experiments, we can see that 1D ESM-1b encodings perform similarly to the 2D encodings on both KIBA and Davis data sets (Figure 4A and 4B) with no significant change $p > 0.01$ in performance with 2D ESM-1b and Pconsc4 encodings. This shows that adding structural information in the form of a protein graph based on a contact map did not improve the overall performance of the DL model significantly.

The Deep Learning Model Relies on Good Ligand Encodings for Learning Binding Affinities. Now we make changes to the ligand encodings as laid out in the methods section to systematically assess how ligand encodings contribute to the overall learning task. From Figure 5A and 5B, we can see that the model's performance on the test set that the ligand encodings greatly impact the binding predictions on both data sets. On both KIBA and Davis data sets, the point randomized encoding had the lowest drop in performance as compared to random node and random sampling perturbation methods. For point randomization methods, there is less than 1% drop across all metrics on KIBA, whereas on Davis there is 3.65% ± 1% on CI and 8.11% ± 2% on Pearson metrics. However, the Wilcoxon test for point randomization perturbation as compared to the original ligand encoding has $p < 0.01$ on both data sets, denoting the change to be significant. In randomizing node feature perturbation, there is a drastic drop in performance on both data sets. The performance on KIBA dropped by 17.18% ± 2% on CI, 45.82% ± 4% on Pearson, and the RMSE increased by 86.03% ± 0.6%. For Davis, the changes are even more drastic with 34.78% ± 1% on CI, 73.09% ± 3% on Pearson, and the RMSE increased by 90.07% ± 4%. The performance of random sampling perturbation has a similar stark effect as node feature randomization for both KIBA (CI: 27.15% ± 1%, Pearson: 82% ± 2%) and Davis (CI: 37.96% ± 1%, Pearson: 80.81% ± 3%). Both random node and random sample perturbations have a significant change ($p < 0.01$) in binding affinity performance as compared to the original ligand encoding. We can observe that the performance for Davis dropped more than for KIBA; this could be due to the difference in data set distributions, as the number of ligands in Davis (68) is much smaller than for KIBA (2111).

In the SI, we also highlight what effect the changes on ligand encodings have in terms of actually predicting binding affinities with respect to experimentally observed values. The findings are summarized in Figure S8. Original ligand encodings obtained R^2 values on the test set: 0.71 ± 0.01 for KIBA and 0.72 ± 0.01 for Davis. Point randomizations have a more distinct effect for Davis with a drop of R^2 = 0.64 ± 0.01, compared to KIBA R^2 = 0.70 ± 0.01. One possible explanation for this is the smaller ligand data set size for Davis, and another is the KIBA score choice itself, which will be discussed in more detail below. For the random node encodings, R^2 = 0.14 ± 0.00 for KIBA and R^2 = 0.01 ± 0.03 for Davis, and similarly for the randomly sampled ligand encodings, R^2 = 0.01 ± 0.046 for KIBA and R^2 = 0.02 ± 0.13 for Davis. Introducing the randomizations in the ligand encodings, the deep learning model can no longer perform the learning task, and the

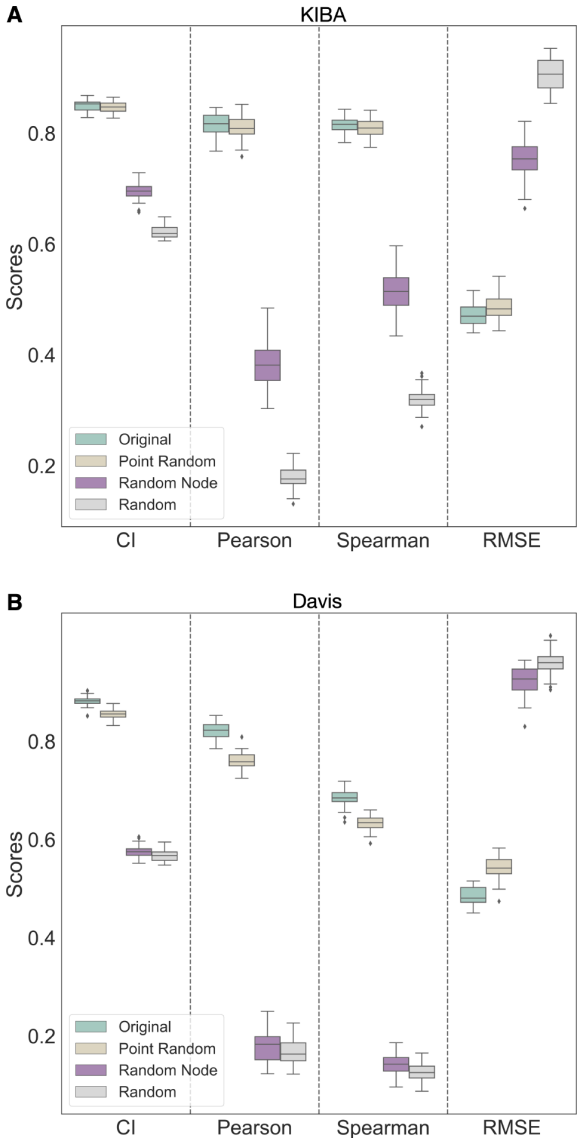


Figure 5. Changes to ligand encodings show a significant change in binding affinity performance. Comparative analysis of the DL model performance in binding prediction testing four different ligand encodings shows that the DL model relies on ligand encodings for both KIBA (A) and Davis (B) data sets. "Original" encoding is the ligand graph generated from the original SMILES string without any changes. "Point Random" encoding is a graph obtained after selectively making changes up to four atoms to the input SMILES string by either substituting one halogen atom with another or removing a (=O) atom, and "Random Node" is the encoding obtained by randomizing only the node features of the input ligand graph. Finally, "Random" refers to the encoding obtained from a graph that is randomly sampled from the data set used in the study. The DL model with randomly sampled and randomized node feature encodings are not learning to estimate BA during training, demonstrating the model's reliance on ligand information.

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

resulting model is unusable as a potential model for binding affinity predictions for kinases. An obvious conclusion is that the presented architecture predominantly learns from ligand encodings, and protein features play hardly any role.

Combining Protein and Ligand Encodings in Different Ways Has No Significant Effect on the Model's Predictability. Lastly, we look at how ligand and protein encodings can be combined in the DL framework. Jiang et al.[23] used concatenation operations to combine protein and ligand encodings with the combined vector being passed along the fully connected layers to predict binding affinity.[22−24] Other combination methods are possible: the element-wise product of protein and ligand encodings and the concatenated vector obtained by combining both element-wise product and concatenation operations. Concatenation allows the DL model to learn complex interactions between the protein and ligand features; here, the model will have access to all the features of the protein and ligand. On the other hand, the element-wise product emphasizes the DL model to learn features important for both protein and ligand. Here, we provide the model with a feature space that is expected to have the most informative aspects of the protein and ligand encodings. When the protein and ligand encoding is concatenated with the product encoding, the model will have access to a feature space that is larger and richer than that in either approach alone.

From Figure 6A (KIBA) and Figure 6B (Davis), we can see no significant improvement in the performance of the DL model on the binding affinity prediction task by both element-wise product of protein and ligand encodings and the concatenated vector obtained by combining both element-wise product and concatenation operations. On the Davis data set, there is a slight drop in performance with the element-wise product (CI: 0.88 ± 0.01, Pearson: 0.81 ± 0.01, Spearman: 0.67 ± 0.01) as compared to both the concatenation (CI: 0.89 ± 0.01, Pearson: 0.82 ± 0.01, Spearman: 0.69 ± 0.01) and fusion encoding of concatenation and element-wise product (CI: 0.89 ± 0.01, Pearson: 0.82 ± 0.01, Spearman: 0.69 ± 0.01). From the Wilcoxon test, the drop by incorporating the element-wise product in the place of concatenation is significant ($p < 0.01$). In contrast, the improvement with fusion concatenation and the element-wise product is not statistically significant ($p > 0.01$). From experiments on the KIBA data set, the element-wise product (CI: 0.85 ± 0.01, Pearson: 0.81 ± 0.02) encoding performed almost the same as the concatenation (CI: 0.85 ± 0.01, Pearson: 0.81 ± 0.02) and fusion encoding (CI: 0.85 ± 0.01, Pearson: 0.82 ± 0.02). The performance for the KIBA data set for both methods shows no statistically significant change ($p > 0.01$). The DL model is not learning anything new from the element-wise product and the fusion of concatenation and product feature spaces.

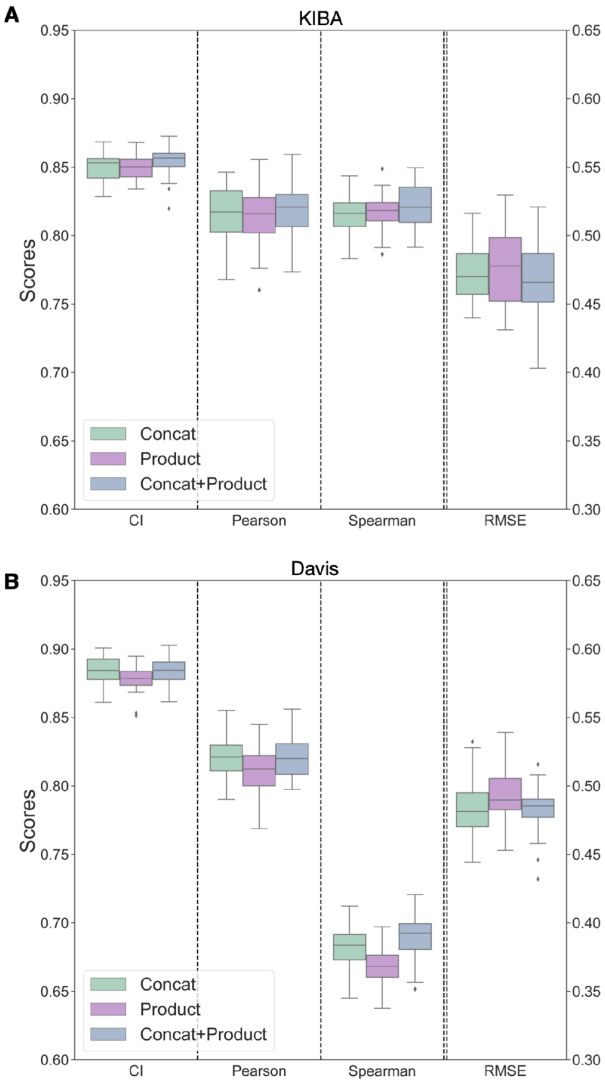


Figure 6. Performance of encoding combining techniques on both KIBA (A) and Davis (B) data sets show minimal change. "Concat" encoding is obtained by concatenating the protein and ligand encodings obtained, while "Product" encoding is from the element-wise product of protein and ligand encoding. "Concat + Product" is formed by concatenating the element-wise product encoding and the concatenated protein and ligand encoding. Binding affinity prediction is minimally affected by the element-wise product and the concatenated vector obtained by combining both element-wise product and concatenation operations.

■ DISCUSSIONS AND CONCLUSIONS

It is often most enticing for a new study to look at binding affinity predictions to introduce a new algorithm or machine learning model, which is then often superficially compared in performance (accuracy in terms of RMSE or correlation) to previous approaches. What is often neglected is looking at good comparison tools for assessing if a new model is statistically actually better than a previous model. What is often forgotten is that the training process is not deterministic, meaning we just pick the best-performing model after training but do not assess its variability. Here, we introduced robust significance testing and error analysis using Wilcoxon's signed rank test to make sure we can make statistically significant statements when comparing our differently trained models on the same deep learning framework. We also included a robust bootstrapping error analysis often neglected when new models are introduced. All of this allowed us to carry out a detailed investigation in terms of how different parts of a deep learning model actually contribute to the overall performance of the final downstream tasks, i.e., the prediction of a binding affinity. In this paper, we systematically investigated the contributions of ligand and protein encodings to the downstream tasks of binding affinity predictions using 1D type of data and 2D type of data. Our 1D data encodings come from protein language

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

models or hand-curated KLIFS data for proteins using convolutional neural network-based DL architectures. For the 2D data, we used graph-based approaches for both ligands, where SMILES strings get converted to ligand graphs and protein sequences get converted to a protein contact map either from a protein structure or through a protein language model. The deep learning CNN and GCN architectures used were not novel; however, we gained new insights into how protein and ligand embeddings contribute systematically to the learning of binding affinities for commonly used data sets used in the literature (KIBA and Davis). We successfully show that protein encodings, as often used in the literature,[48] have little to no contribution to the downstream learning tasks, and all correlation learned between structure and binding affinity is through ligand encodings. Furthermore, augmenting data sets from a 1D language model to a 2D graph model does not make the learning process significantly better. While we highlight the importance of understanding the role of encodings in DL models and provide insights into what the current DL models are learning in predicting protein−ligand binding affinity, we recognize other limitations associated with these DL methods. Most of the current DL methods are trained and tested on small-size kinase data sets with skewed binding interactions data (Figure 2). Testing DL frameworks only on kinase data sets is an obvious choice because of the amount of available data. However, care should be taken with the KIBA data set. It is tempting to augment a data set to include experimental data from multiple experimental assays to include IC_{50}, K_i, and K_d. To broaden the data set, the KIBA score was designed to account for multiple sources of experimental measurements.[10] Unfortunately, by combining information from different assays and measurement types, the resulting uncertainty introduced is not accounted for in the KIBA score. This means evaluating the accuracy on the downstream task becomes inaccurate, as there is no notion of reliability of a single KIBA score incorporated into the model. As Killiokoski et al.[49] pointed out, IC_{50}s can be augmented with K_i, but care should be taken when looking at SAR/QSAR models in terms of the maximally achievable performance due to the introduced noise.

More generally, the kinase data sets are a good starting point for model development as they are publicly available and have sufficient volume of data to train the DL, making them an attractive choice for initial model development, it is critical to understand their inherent limitations. Specifically, although they serve as a baseline model, these data sets alone will not provide the breadth required to develop a globally applicable binding affinity prediction algorithm across diverse protein families and ligand. Future work will require data sets beyond BindingDB,[50] improvements around data representation, and architctural improvements that allow learning of joint protein and ligand interactions.

We have seen that the DL models do not learn information that captures the protein and ligand interaction features but are biased toward learning from the ligand features. The current approaches to encode proteins from sequences and contact maps in the form of 1D and 2D encodings with CNN or GNN architectures are not sufficient to capture protein features to build a robust binding affinity prediction tool, and future work in this direction is required. The most obvious starting point is making use of 3D protein−ligand interaction features from 3D complex structures, for which there already is a body of work.[51,52] However, Volkov et al.[48] highlighted challenges with current complex-based models (3D), namely, that they do not necessarily learn the physics of protein−ligand binding. They found that explicit description of protein−ligand interactions from complexes provides no clear advantage compared to the corresponding interaction-agostic models based solely on ligand or protein descriptors. Furthermore, Volkov et al.[48] discussed hidden protein and ligand biases in the PDBbind[9] data set for training complex-based models showing that these models have partly memorized the input data and did not learn the features that correspond to protein−ligand interactions. One avenue to explore in the future is to jointly learn features for making predictions; in this way, the DL model uses the joint features corresponding to protein−ligand interaction properties. This could be done, e.g., by including physics-based 3D snapshots in training the DL models to predict the binding affinity, similar to what has been explored to active learning approaches incorporating molecular dynamics-based binding affinity predictions.[53]

■ ASSOCIATED CONTENT

Data Availability Statement

All data for the experiments carried out and instructions on how to reproduce this work can be found at https://github.com/meyresearch/DL_protein_ligand_affinity.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.3c01208.

Architectures for the graph neural networks, methods used, and additional experimental results (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Antonia S. J. S. Mey − *EaStCHEM School of Chemistry, University of Edinburgh, Edinburgh EH9 3FJ, U.K.*; orcid.org/0000-0001-7512-5252; Email: antonia.mey@ed.ac.uk

Authors

Rohan Gorantla − *School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, U.K.; EaStCHEM School of Chemistry, University of Edinburgh, Edinburgh EH9 3FJ, U.K.*

Alžbeta Kubincová − *Exscientia, Schrödinger Building, Oxford OX4 4GE, U.K.*

Andrea Y. Weiße − *School of Biological Sciences, University of Edinburgh, Edinburgh EH9 3FF, U.K.; School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, U.K.*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.3c01208

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the United Kingdom Research and Innovation (grant EP/S02431X/1), UKRI Centre for Doctoral Training in Biomedical AI at the University of Edinburgh, School of Informatics, and Exscientia Plc, Oxford. The authors thank John Chodera for his discussions and insights during the project.

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

REFERENCES

(1) Brown, N. *Artificial Intelligence in Drug Discovery*; Royal Society of Chemistry, 2020; Vol. 75.
(2) Kimber, T. B.; Chen, Y.; Volkamer, A. Deep learning in virtual screening: recent applications and developments. *Int. J. Mol. Sci.* **2021**, *22*, 4435.
(3) Wang, R.; Lu, Y.; Wang, S. Comparative evaluation of 11 scoring functions for molecular docking. *J. Med. Chem.* **2003**, *46*, 2287−2303.
(4) Stanzione, F.; Giangreco, I.; Cole, J. C. Use of molecular docking computational tools in drug discovery. *Prog. Med. Chem.* **2021**, *60*, 273−343.
(5) Mey, A. S.; Allen, B. K.; Macdonald, H. E. B.; Chodera, J. D.; Hahn, D. F.; Kuhn, M.; Michel, J.; Mobley, D. L.; Naden, L. N.; Prasad, S.; Rizzi, A.; Scheen, J.; Shirts, M. R.; Tresadern, G.; Xu, H. Best Practices for Alchemical Free Energy Calculations. *Living J. Mol. Sci.* **2020**, *2*, 18378.
(6) Mey, A. S.; Juárez-Jiménez, J.; Hennessy, A.; Michel, J. Blinded predictions of binding modes and energies of HSP90-α ligands for the 2015 D3R grand challenge. *Bioorg. Med. Chem.* **2016**, *24*, 4890−4899.
(7) Hahn, D. F.; Bayly, C. I.; Boby, M. L.; Macdonald, H. E. B.; Chodera, J. D.; Gapsys, V.; Mey, A. S.; Mobley, D. L.; Benito, L. P.; Schindler, C. E.; Tresadern, G.; Warren, G. L. Best practices for constructing, preparing, and evaluating protein-ligand binding affinity benchmarks. *Living J. Mol. Sci.* **2022**, *4*, 1497−1498.
(8) Lyu, J.; Irwin, J. J.; Shoichet, B. K. Modeling the expansion of virtual screening libraries. *Nat. Chem. Biol.* **2023**, *19*, 712−718.
(9) wwPDB consortium. Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Res.* **2019**, *47* (D1), D520−D528.
(10) Tang, J.; Szwajda, A.; Shakyawar, S.; Xu, T.; Hintsanen, P.; Wennerberg, K.; Aittokallio, T. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *J. Chem. Inf. Model.* **2014**, *54*, 735−743.
(11) Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P. Comprehensive analysis of kinase inhibitor selectivity. *Nat. Biotechnol.* **2011**, *29*, 1046−1051.
(12) Turzo, S. B. A.; Hantz, E. R.; Lindert, S. Applications of machine learning in computer-aided drug discovery. *QRB Discov* **2022**, *3*, No. e14.
(13) Zhao, L.; Zhu, Y.; Wang, J.; Wen, N.; Wang, C.; Cheng, L. A brief review of protein−ligand interaction prediction. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 2831−2838.
(14) Wallach, I.; Dzamba, M.; Heifets, A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv*, October 10, 2015, arXiv:1510.02855. .
(15) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. Development and evaluation of a deep learning model for protein−ligand binding affinity prediction. *Bioinformatics* **2018**, *34*, 3666−3674.
(16) Jiménez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. K. deep: protein−ligand absolute binding affinity prediction via 3d-convolutional neural networks. *J. Chem. Inf. Model.* **2018**, *58*, 287−296.
(17) Jones, D.; Kim, H.; Zhang, X.; Zemla, A.; Stevenson, G.; Bennett, W. D.; Kirshner, D.; Wong, S. E.; Lightstone, F. C.; Allen, J. E. Improved protein−ligand binding affinity prediction with structure-based deep fusion inference. *J. Chem. Inf. Model.* **2021**, *61*, 1583−1592.
(18) Ahmed, A.; Mam, B.; Sowdhamini, R. DEELIG: A deep learning approach to predict protein-ligand binding affinity. *Bioinform. Biol. Insights* **2021**, *15*, 1−9.
(19) Wang, K.; Zhou, R.; Tang, J.; Li, M. GraphscoreDTA: optimized graph neural network for protein−ligand binding affinity prediction. *Bioinformatics* **2023**, *39* (6), btad340.
(20) Jin, Z.; Wu, T.; Chen, T.; Pan, D.; Wang, X.; Xie, J.; Quan, L.; Lyu, Q. CAPLA: improved prediction of protein−ligand binding affinity by a deep learning approach based on a cross-attention mechanism. *Bioinformatics* **2023**, *39* (2), btad049.
(21) Kalakoti, Y.; Yadav, S.; Sundar, D. TransDTI: transformer-based language models for estimating DTIs and building a drug recommendation workflow. *ACS Omega* **2022**, *7*, 2706−2717.
(22) Nguyen, T.; Le, H.; Quinn, T. P.; Nguyen, T.; Le, T. D.; Venkatesh, S. GraphDTA: Predicting drug−target binding affinity with graph neural networks. *Bioinformatics* **2021**, *37*, 1140−1147.
(23) Jiang, M.; Li, Z.; Zhang, S.; Wang, S.; Wang, X.; Yuan, Q.; Wei, Z. Drug−target affinity prediction using graph neural network and contact maps. *RSC Adv.* **2020**, *10*, 20701−20712.
(24) Öztürk, H.; Özgür, A.; Ozkirimli, E. DeepDTA: deep drug−target binding affinity prediction. *Bioinformatics* **2018**, *34*, 821−829.
(25) Öztürk, H.; Ozkirimli, E.; Özgür, A. WideDTA: prediction of drug-target binding affinity. *ArXiv*, February 4, 2019, arXiv:1902.04166. .
(26) Michel, M.; Menéndez Hurtado, D.; Elofsson, A. PconsC4: fast, accurate and hassle-free contact predictions. *Bioinformatics* **2019**, *35*, 2677−2679.
(27) Lin, X.; Zhao, K.; Xiao, T.; Quan, Z.; Wang, Z.-J.; Yu, P. S. DeepGS: Deep Representation Learning of Graphs and Sequences for Drug-Target Binding Affinity Prediction. *ECAI* **2020**, *325*, 1301−1308.
(28) Karimi, M.; Wu, D.; Wang, Z.; Shen, Y. DeepAffinity: interpretable deep learning of compound−protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics* **2019**, *35*, 3329−3338.
(29) Zhang, S.; Jiang, M.; Wang, S.; Wang, X.; Wei, Z.; Li, Z. SAG-DTA: prediction of drug−target affinity using self-attention graph network. *Int. J. Mol. Sci.* **2021**, *22*, 8993.
(30) Nguyen, T. M.; Nguyen, T.; Le, T. M.; Tran, T. Gefa: early fusion approach in drug-target affinity prediction. *IEEE/ACM Trans Comput. Biol. Bioinform* **2022**, *19*, 718−728.
(31) Yang, Z.; Zhong, W.; Zhao, L.; Chen, C. Y.-C. ML-DTI: mutual learning mechanism for interpretable drug−target interaction prediction. *J. Phys. Chem. Lett.* **2021**, *12*, 4247−4261.
(32) Rao, R.; Meier, J.; Sercu, T.; Ovchinnikov, S.; Rives, A. Transformer protein language models are unsupervised structure learners. *Biorxiv*, December 15, 2020, .
(33) Kanev, G. K.; de Graaf, C.; Westerman, B. A.; de Esch, I. J.; Kooistra, A. J. KLIFS: an overhaul after the first 5 years of supporting kinase research. *Nucleic Acids Res.* **2021**, *49*, D562−D569.
(34) Jumper, J.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583−589.
(35) He, T.; Heidemeyer, M.; Ban, F.; Cherkasov, A.; Ester, M. SimBoost: a read-across approach for predicting drug−target binding affinities using gradient boosting machines. *J. Cheminformatics* **2017**, *9*, 1−14.
(36) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* 31st Conference on Neural Information Processing Systems (NIPS 2017).
(37) The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic Acids Res.* **2017**, *45*, D158−D169.
(38) Di Paola, L.; De Ruvo, M.; Paci, P.; Santoni, D.; Giuliani, A. Protein contact networks: an emerging paradigm in chemistry. *Chem. Rev.* **2013**, *113*, 1598−1613.
(39) Jasmin Güven, J.; Molkenthin, N.; Mühle, S.; Mey, A. S J S What geometrically constrained models can tell us about real-world protein contact maps. *Phys. Biol.* **2023**, *20*, 046004.
(40) Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI* **2015**, 234−241.
(41) Varadi, M.; et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res.* **2022**, *50*, D439−D444.
(42) Naughton, F. B.; Alibay, I.; Barnoud, J.; Barreto-Ojeda, E.; Beckstein, O.; Bouysset, C.; Cohen, O.; Gowers, R. J.; MacDermott-Opeskin, H.; Matta, M.; Melo, M. N.; Reddy, T.; Wang, L.; Zhuang, Y. MDAnalysis 2.0 and beyond: fast and interoperable, community driven simulation analysis. *Biophys. J.* **2022**, *121*, 272a−273a.

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

(43) Hagberg, A.; Swart, P.; S Chult, D. *Exploring network structure, dynamics, and function using NetworkX; tech. rep.*; Los Alamos National Lab.(LANL), Los Alamos, NM (United States),2008.
(44) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S.; Rives, A. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, July 21, 2022. .
(45) Bento, A. P.; Hersey, A.; Félix, E.; Landrum, G.; Gaulton, A.; Atkinson, F.; Bellis, L. J.; De Veij, M.; Leach, A. R. An open source chemical structure curation pipeline using RDKit. *J. Cheminformatics* **2020**, *12*, 1−16.
(46) Kipf, T. N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv*, September 9, 2016, arXiv:1609.02907. .
(47) Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom* **2020**, *21*, 1−13.
(48) Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. On the frustration to predict binding affinities from protein−ligand structures with deep neural networks. *J. Med. Chem.* **2022**, *65*, 7946−7958.
(49) Kalliokoski, T.; Kramer, C.; Vulpetti, A.; Gedeck, P. Comparability of mixed IC50 data−a statistical analysis. *PloS One* **2013**, *8*, No. e61007.
(50) Gilson, M. K.; Liu, T.; Baitaluk, M.; Nicola, G.; Hwang, L.; Chong, J. BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res.* **2016**, *44*, D1045−D1053.
(51) Seo, S.; Choi, J.; Park, S.; Ahn, J. Binding affinity prediction for protein−ligand complex using deep attention mechanism based on intermolecular interactions. *BMC Bioinform* **2021**, *22*, 1−15.
(52) Born, J.; Huynh, T.; Stroobants, A.; Cornell, W. D.; Manica, M. Active site sequence representations of human kinases outperform full sequence representations for affinity prediction and inhibitor generation: 3D effects in a 1D model. *J. Chem. Inf. Model.* **2022**, *62*, 240−257.
(53) Khalak, Y.; Tresadern, G.; Hahn, D. F.; de Groot, B. L.; Gapsys, V. Chemical space exploration with active learning and alchemical free energies. *J. Chem. Theory Comput.* **2022**, *18*, 6259−6270.

https://doi.org/10.1021/acs.jcim.3c01208
J. Chem. Inf. Model. 2024, 64, 2496−2507

Correction to "From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction"

Rohan Gorantla,[†,‡] Alžbeta Kubincová,[¶] Andrea Y. Weiße,[§,†] and Antonia S. J. S. Mey[*,‡]

†*School of Informatics, University of Edinburgh, EH8 9AB, UK*

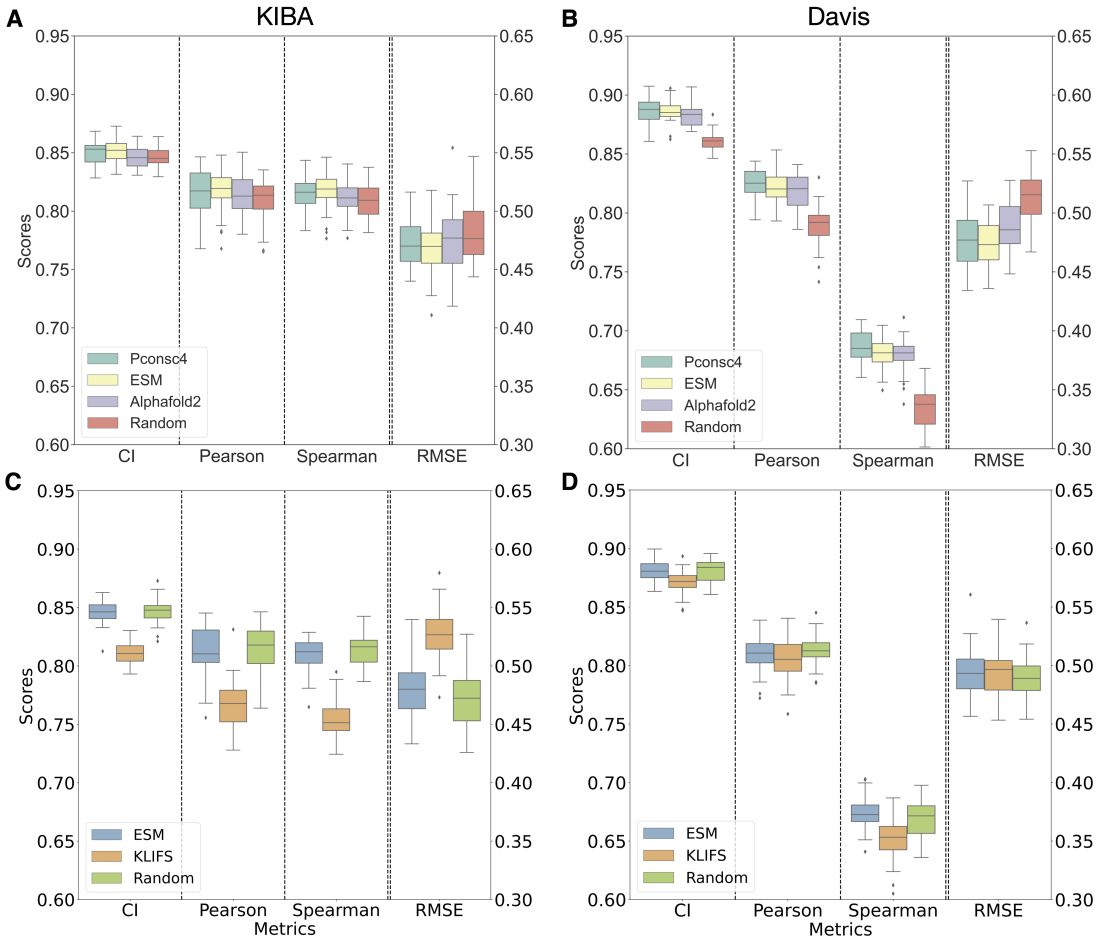
‡*EaStCHEM School of Chemistry, University of Edinburgh, EH9 3FJ, UK*

¶*Exscientia, Schrödinger Building, Oxford, OX4 4GE, UK*

§*School of Biological Sciences, University of Edinburgh, EH9 3FF, UK*

E-mail: antonia.mey@ed.ac.uk

Figure 4, panels C and D are identical. Panel C was accidentally duplicated from panel D. The values reported in the text remain accurate and were not affected by this error. The correct version of Figure 4 is provided below. The caption remains unchanged, as it still accurately describes the updated figure panel and this error does not affect the results, discussion, or conclusions of the paper.


A
KIBA
B
Davis
C
D
Scores
Metrics
CI
Pearson
Spearman
RMSE
Pconsc4
ESM
Alphafold2
Random
KLIFS

Chapter 5

Learning Binding Affinities via Fine-tuning of Protein and Ligand Language Models

This chapter is based on the work described in the following publication - Gorantla, R., Gema, A. P., Yang, I. X., Serrano-Morrás, Á., Suutari, B., Jiménez, J. J., & Mey, A. S. *bioRxiv 2024.11.01.621495* (2024) and under peer review.

In this chapter, I introduce BALM, a deep learning framework designed to predict **b**inding **a**ffinities using protein sequences and ligand SMILES by fine-tuning pretrained protein and ligand **l**anguage **m**odels. This work builds on the insights from Chapter 4, where I studied the impact of proteins and ligand information on deep learning frameworks for binding affinity predictions and highlighted the need for more robust and generalizable prediction methods.

As a next step, I address three key challenges via the BALM framework at model, data, and evaluation levels to make the deep learning models more useful for practical screening. At the model level, BALM employs a novel approach that optimizes cosine similarity (representing experimental affinities) in a shared embedding space between protein and ligand representations, moving beyond earlier concatenation-based frameworks that provided poor results. BALM incorporates

efficient fine-tuning strategies, enabling rapid adaptation to new targets with minimal computational overhead and training data requirements. BALM introduces more stringent evaluation protocols by validating performance on larger more complex datasets, implementing challenging data splits and comprehensive metrics that better reflect real-world screening scenarios. The framework's reliability is validated through systematic comparisons with established methods, such as molecular docking and experimental binding data across diverse protein targets, providing a more realistic assessment of model performance in drug discovery applications.

Learning Binding Affinities via Fine-tuning of Protein and Ligand Language Models

Rohan Gorantla,[†,‡] Aryo Pradipta Gema,[†] Ian Xi Yang,[‡] Álvaro Serrano-Morrás,[¶,§] Benjamin Suutari,[‖] Jordi Juárez-Jiménez,[¶,§] and Antonia S. J. S. Mey[*,‡]

†*School of Informatics, University of Edinburgh, Crichton Street, Edinburgh, EH8 9AB, Midlothian, United Kingdom*

‡*EaStCHEM School of Chemistry, University of Edinburgh, David Brewster Road, Edinburgh, EH9 3FJ, Midlothian, United Kingdom*

¶*Unitat de Fisicoquímica, Departament de Farmàcia i Tecnologia Farmacèutica, i Fisicoquímica, Facultat de Farmàcia i Ciències de l'Alimentació, Universitat de Barcelona (UB), Av. Joan XXIII, 27-31, 08028 Barcelona, Spain*

§*Institut de Química Teòrica i Computacional (IQTC), Facultat de Química i Física, Universitat de Barcelona (UB), C. Martí i Franquès, 1, 08028 Barcelona, Spain*

‖*Independent Researcher, 37027, United States*

E-mail: antonia.mey@ed.ac.uk

Abstract

Accurate *in-silico* prediction of protein-ligand binding affinity is essential for efficient hit identification in large molecular libraries. Commonly used structure-based methods such as docking often fail to rank compounds effectively, and free energy-based approaches, while accurate, are too computationally intensive for large-scale screening.

Existing deep learning models struggle to generalize to new targets or drugs, and current evaluation methods often do not accurately reflect real-world performance. We introduce **BALM**, a deep learning framework that predicts **b**inding **a**ffinity using pretrained protein and ligand **l**anguage **m**odels. We also propose improved evaluation strategies with diverse data sets and metrics to assess model performance to new targets better. Using the BindingDB dataset, BALM generalises unseen drugs, scaffolds, and targets. In few-shot scenarios for targets such as *USP7* and *Mpro*, it outperforms traditional machine learning and docking methods, including AutoDock Vina. Adoption of our target-based evaluation methods will allow a more stringent evaluation of machine learning-based scoring tools. Our protein prediction framework shows good performance, is computationally efficient, and is highly adaptable within this evaluation setting, making it practical for early-stage drug discovery screening.

Introduction

Identifying hit compounds from ultra-large libraries for accelerating target-based drug discovery hinges on the precision of *in-silico* methods for predicting protein-ligand binding affinity (BA). Structure-based approaches such as giga-docking[1,2] are commonly used for virtual screening.[3,4] While docking is often a good approach to generate chemically reasonable poses, it usually falls short in rank-ordering compounds.[5] Free energy-based methods[6–8] offer more reliable predictions but are computationally costly for hit discovery in ultra-large libraries and are usually restricted to lead optimization stages. Limited size and diversity of screening libraries have long been a bottleneck for the detection of novel potent ligands and for the whole process of drug discovery.[9] With recent advancements in on-demand libraries such as Enamine,[4,10] scaling up the virtual screening capabilities from million to a trillion compounds is possible. The chances of these libraries containing more potent ligands with better physicochemical properties increase by increasing the search-space. However, the number of potential decoys that need to be filtered out will also increase. Effective hit

identification in ultra-large libraries requires computationally efficient and accurate *in-silico* methods. Machine learning advances, particularly deep learning (DL), accelerate drug discovery stages, from virtual screening[11–15] to finding hits to optimizing lead compounds with active learning[16–18] and *de-novo* design[19–22] techniques.

In recent years, many DL strategies have been developed for predicting binding affinity,[12,13,23–26] yet challenges remain at the model, data, and evaluation levels. At the model level, DL methods typically fall into two categories - complex-based and sequence-based. Complex-based models leverage three-dimensional (3D) protein-ligand structural data, often derived from databases such as PDBBind.[27] Although the use of 3D structures can potentially capture detailed interaction information, complex-based models often fail to generalize due to the limited availability of high-quality 3D complexes and biases within structural data.[28] On the sequence-based side, models are trained using 1D protein sequences and SMILES representations of ligands. This makes them more flexible in terms of data requirements due to large sequence databases but is limited to a much smaller pool of interaction data in comparison to typical language model training datasets. Furthermore, investigations into understanding what these models learn from input protein sequence and ligand SMILES data have shown a dependence on ligand information and neglect of combined information on protein and ligand for prediction.[29] Challenges at the data and evaluation levels further hinder DL model performance in binding affinity prediction. Commonly used datasets for training these DL models combine IC50 and K_i data from various assays, introducing significant noise due to differences in how experiments are performed.[30] This undermines the reliability of the model. Furthermore, evaluating these models by randomly splitting data across train and test sets often results in data leakage, where similar compounds end up in both training and test sets, leading to overestimating the model's generalizability.[31] We address these three key challenges at **model, data,** and **evaluation levels** to provide a DL framework for practical screening purposes.

At the model level, we introduce **BALM**, a novel sequence-based deep learning method

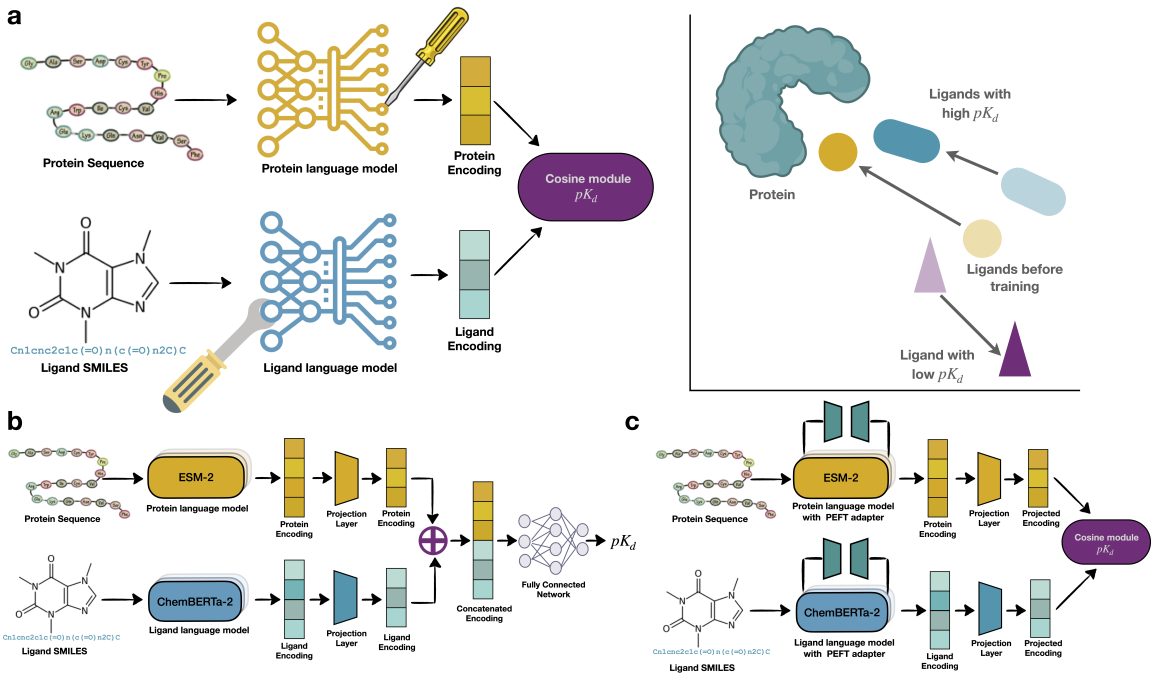


Figure 1: **Overview of BALM architecture.** **(a)** BALM learns by optimizing the distance between protein and ligand encodings using a cosine similarity metric directly representing binding affinity (pK_d). Cosine similarity is maximized for binding interactions and minimized for non-binding interactions. Protein sequences and ligand SMILES strings are encoded using language models trained on extensive protein and ligand databases. ESM-2[32] is used for encoding protein sequences, and ChemBERTa-2[33] for extracting features from ligand SMILES. These encoded features are projected into a shared latent space via linear layers. The model then optimizes the cosine similarity between protein and ligand embeddings to learn binding affinity. **(b)** The baseline model is based on a pipeline from previous studies[13,18,24,34,35] where features are first extracted from protein sequences and SMILES strings using a deep learning architecture. Then these features are concatenated and passed through a linear layer to predict binding affinity. The model is trained using a Mean Squared Error (MSE) loss function to predict the binding affinity. **(c)** BALM with parameter-efficient fine-tuning (PEFT). PEFT adapters are added to the protein (LoKr) and ligand (LoHa) language models, allowing selective fine-tuning of small subsets of additional parameters while keeping the main model weights frozen. The fine-tuned embeddings are then projected and optimized using cosine similarity similar to the original BALM for affinity prediction.

for predicting protein-ligand **b**inding **a**ffinity using pre-trained protein and ligand **l**anguage **m**odels. The key novelty over previous metric learning-based methods for binding affinity prediction,[36–38] is that BALM learns binding affinities based on the distance between proteins and ligands in feature space. BALM operates on pairs of data (protein sequences, ligand SMILES) as shown in Fig. 1 to learn an embedding space that directly represents the binding affinity pK_d. BALM builds on advances in protein[32] and ligand[33] language models, which learn representations in an unsupervised manner from large-scale datasets such as UniRef[39] for proteins and PubChem[40] for ligands. BALM uses these protein (ESM-2) and ligand (ChemBERTa-2) language models to extract features and incorporates parameter-efficient fine-tuning (PEFT)[41–46] to adapt these embeddings specifically for binding affinity prediction. This is achieved by tuning only a small fraction of model parameters and significantly reducing computational costs for training. By embedding protein-ligand pairs in a shared feature space that reflects interaction strength, BALM optimizes the cosine similarity (a distance metric) to directly predict binding affinity (pK_d).

To ensure the best data integrity from the BindingDB dataset, we only use K_d values for training and remove any assay limits in the standard dataset. While reducing the size of the training set, this makes it of higher quality. In terms of improved evaluation strategies, we systematically test BALM's performance across challenging data splits within the BindingDB dataset,[47,48] including zero-shot (cold drug and cold target), scaffold, and random splits. Our experiments demonstrate that BALM outperforms our literature-inspired baseline model across all splits, showcasing generalization to unseen drugs, scaffolds, and targets on typically reported aggregate evaluation metrics such as RMSE and correlation coefficients. Incorporating PEFT further enhances BALM's performance, highlighting the benefits of parameter-efficient fine-tuning for producing reliable affinity models.

Crucially, the true picture of a model's utility in a drug discovery scenario cannot be evaluated by reporting its performance on aggregated protein targets.[8] We evaluate zero-shot predictions across individual protein targets and suggest using Fisher-transformed correla-

tions to capture target-specific performance. This metric captures the significant variability in zero-shot predictions across individual protein targets, particularly in cold target settings where protein-ligand data are sparse. By utilizing Fisher-transformed[49] correlations to capture target-specific performance, we obtain a clearer view of BALM's practical applicability and limitations, offering a model evaluation approach that could inform the development and fair assessment of future DL methods. To further assess BALM's adaptability, we evaluated its performance in practical few-shot learning scenarios on two unknown (to the model) targets—*USP7*, a ubiquitin-specific protease,[50] and *Mpro*, the main protease of SARS-CoV-2.[51,52] Using a pre-trained BALM model, we fine-tuned the embeddings for these targets, observing rapid and notable performance gains with only a small fraction of additional data, highlighting BALM's potential in real-world settings with limited experimental data. We benchmarked BALM against simple machine learning models, such as Gaussian Processes, which depend heavily on feature selection and kernel choice, impacting generalizability and performance. BALM's performance is robust across different targets without further tuning embedding requirements, emphasizing its practicality in real-world screening scenarios. We further compared BALM's performance with traditional structure-based docking methods, including AutoDock Vina[53] and rDock,[54] on the Leak Proof PDBBind dataset,[55] which was designed to eliminate test set data leakage. BALM outperformed these methods in the rescoring of crystal structures in predictive accuracy. Lastly, we evaluated pre-trained BALM on the Free Energy Benchmark,[8] which includes data representative of the lead optimization stage, featuring congeneric series for three challenging targets (*MCL1*, *HIF2A*, *SYK*). While BALM struggled on *HIF2A* and *SYK*, it showed competitive accuracy by few-shot fine-tuning on the larger *MCL1* dataset. For highly similar ligands, as in lead optimization datasets, BALM's predictions cluster within a narrower range as already observed in other models,[26] pointing towards some current limitations. This work demonstrates how, under rigorous evaluation metrics, large language models represent an attractive solution for fast and accurate protein property prediction.

Results

BALM learns better than baseline models across challenging data splits

We investigated the performance of the BALM model (Fig. 1a) on the BindingDB dataset[47,48] and compared it with a consensus Baseline model[13,18,24,34,35] using different data splits. We focused on the BindingDB dataset, specifically looking at binding affinities measured according to K_{d},[48] which contains around 48,000 non-covalent interactions involving 1,090 protein targets and 9,900 ligands. To ensure stable model training, we transformed the $\mathrm{K_d}$ values into pK_{d} values, following recommendations from previous studies.[24,29,35] To address significant bias caused by binding assay limits and the resulting skewed range of affinity predictions (Fig. S1), we cleaned the dataset by removing assay limits. This resulted in discarding around 21,000 interactions around a $pK_{mathrmd}$ of 5. After this cleaning step, the dataset was reduced to about 25,000 interactions, involving around 1,070 targets and 9,200 ligands. To evaluate the efficacy of each model, we considered random, cold target, cold drug, and scaffold dataset splits, as discussed in more detail in the data section. For the random split, interactions were randomly divided into train and test sets. The cold target split involved grouping protein targets into distinct sets to assess performance on unseen proteins, while the cold drug split assigned ligands similarly to test novel compounds. For the scaffold split, drugs were categorized by their two-dimensional Murcko scaffolds[56] using RDKit's `MurckoScaffold` module,[57] testing the model's ability to generalize to new chemical scaffolds. Target and drug-specific data splits are key for generalizability. We evaluated the models' ability to generalize using the concordance index (CI), Pearson correlation, Spearman rank correlation, and root mean squared error (RMSE). To compare the performance of two trained models on the test set, we assessed statistical significance using paired t-tests, with $p < 0.05$ considered significant.

The Baseline model (Fig. 1b) follows a pipeline from previous studies[13,18,24,34,35] where

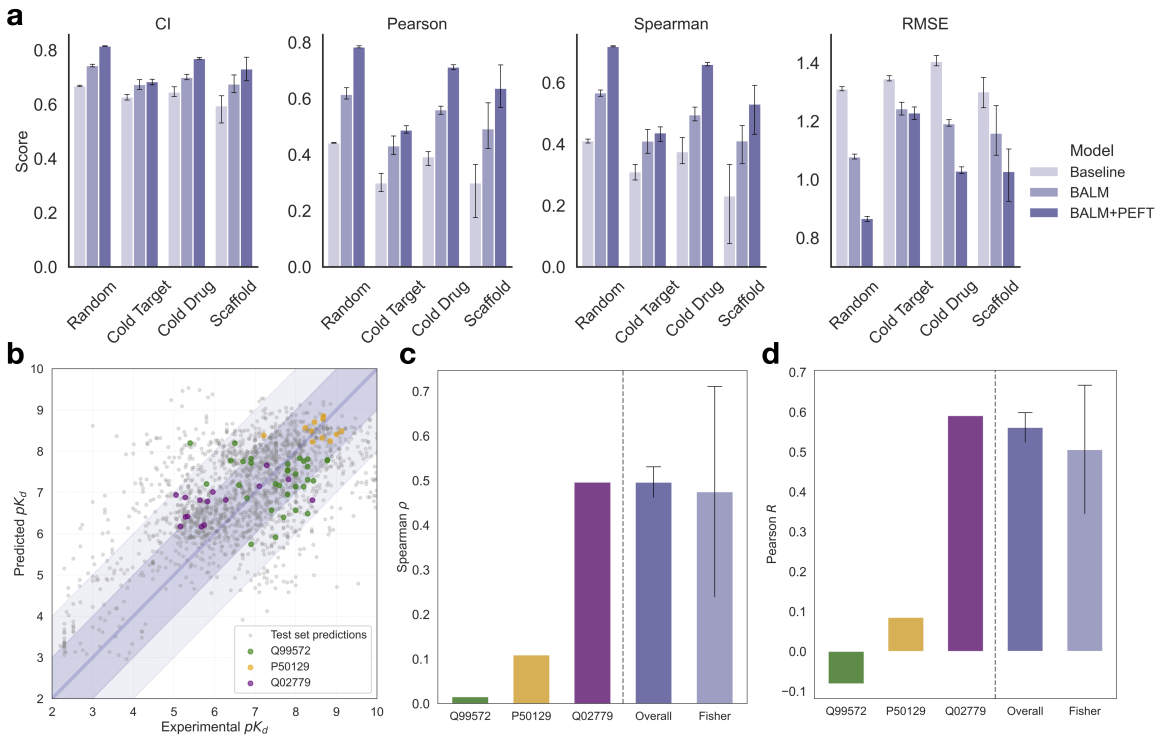


Figure 2: **BALM performs better than the baseline model on challenging data splits, and parameter-efficient fine-tuning further improves its performance. (a)** Performance comparison of Baseline, BALM, and BALM+PEFT models on the filtered BindingDB dataset using four data splits - Random, Cold Target, Cold Drug, and Scaffold. Error bars indicate the standard deviation that was observed during three random variations in the data splits and model initializations. Across all splits, BALM+PEFT (dark blue) demonstrates the best performance, as reflected by increased Pearson correlation and reduced RMSE. **(b)** Scatter plot showing zero-shot predictions of the BALM+PEFT variant on three randomly selected targets from the test set in cold target split. Grey points represent predictions across the entire test set, while coloured points highlight specific targets: *Q99572* (green), *P50129* (yellow), and *Q02779* (purple). **(c)** Spearman correlation (ρ) bar plot for selected test targets compared to overall dataset performance and Fisher transformed mean performance. *Overall* ρ indicates the model's correlation calculated across all test set predictions, while the *Fisher* ρ aggregates individual correlations for each target by averaging correlations at the target level. The *Fisher* ρ provides a more realistic picture of model performance, accounting for target-specific variability and showing higher variance, as some targets may have better predictions (high Spearman) while others show worse performance (low Spearman). Larger standard deviations in the *Fisher transformed* values reflect these differences across targets. **(d)** Pearson correlation (R) bar plot for selected test targets, similar to panel (c). The *Overall* R reflects the model's global performance across the test set, while the *Fisher* transformed aggregated R gives a view of target-specific correlations. *Fisher* aggregation helps highlight the performance distribution across targets, and large standard deviations for the *Fisher* suggest that some targets are predicted well while others are predicted poorly, indicating variability in the model's zero-shot performance across unseen targets.

features are extracted from protein sequences and SMILES strings using a DL architecture. These features are then concatenated and passed through a linear layer to predict binding affinity, with the Mean Squared Error (MSE) loss function used to learn pK_d. For a fair comparison, we employed protein and ligand language models and utilized projected encodings from the projection layer similar to BALM. These encoded representations are concatenated and passed through a fully connected layer to predict binding affinity. Both the baseline and BALM models were run with three random seeds to ensure the model performance was consistent with random variation in the data splits or initialization.

Fig. 2a provides an overview of the evaluation metrics performance of the BALM (medium blue) and Baseline (light blue) models across four data splits. In the Random split, BALM outperforms the baseline model in all evaluation metrics with statistical significance.

While the performance of BALM was in general better than the Baseline across all splits and metrics, the results on the most challenging splits were underwhelming.

Previous studies[44–46] have demonstrated that parameter-efficient fine-tuning (PEFT) can tailor the performance of pre-trained language models for systems not used during training by updating a small fraction of the parameters. We investigated the use of four different parameter-efficient adapters (LoRA, LoHa, LoKr, and IA3), which are added to the language models as shown in Fig. 1c., to further improve the performance of BALM. Using the BindingDB data with random splitting, we evaluated separately the effect of the different PEFT adapters on the ligand and the protein language models (Fig. S2), using three different values of the rank matrix hyperparameter r (8, 16 and 32) for the three methods that required this hyperparameter. LoHa (r= 16) showed the highest performance boost for ligand-only fine-tuning, improving Pearson correlation by 9.4%, while LoKr (r= 8) protein-only fine-tuning increased the value of the Pearson correlation by 18.2%. Combining the best-performing methods for both ligand and protein yielded the BALM+PEFT model, which demonstrated a substantial 23.4% improvement over the performance of the non-optimized version of BALM. Beyond the random split, the BALM+PEFT variant con-

sistently outperformed the initial model across all splits and metrics (Fig. 2a). Improved performance in the cold target and cold drug splits are noteworthy, with increases in Pearson correlation of approximately 20% and 10% respectively and a decrease of ca. 5% in RMSE in both cases.

BALM+PEFT is a fine-tuned model that, according to the monitored metrics, displays high predictive power, and reliability, and seems highly generalisable to new targets and chemical scaffolds, as shown by its performance in the cold protein and cold target splits. However, evaluating the global performance of these models, grouping different protein targets and mixing extremely different chemical scaffolds does not truly inform the models' performance in situations usually faced during drug discovery efforts. This is the ability to accurately screen millions of compounds of unknown affinity to a single, oftentimes previously unseen, macromolecular target. To this end, we investigated the performance of BALM+PEFT in the affinity prediction in the cold-target setting in more detail.

Zero-shot performance is not reliable for all targets, and commonly used cumulative metrics do not give the real picture in cold target-setting

We evaluated the zero-shot performance of the BALM+PEFT model in the cold target split using selected targets to understand its performance on unseen proteins. Fig. 2b shows the zero-shot predictions for three targets of pharmacological relevance: the P2X purinoreceptor 7 (P2RX7, Uniprot ID: *Q99572*, the serotonin receptor 2A (HTR2A *P50129*), and the Mitogen-activated Protein kinase kinase kinase 10 (MAP3K10, *Q02779*). Notably, there is significant variability in the values of both Spearman and Pearson correlation coefficients with respect to experimental values for these protein targets, with values ranging close to 0 for the P2RX7 (green dots) and HTR2A (yellow dots) and very close to the average performance of the model in the MAP3K10 case (purple dots). Interestingly, employing the

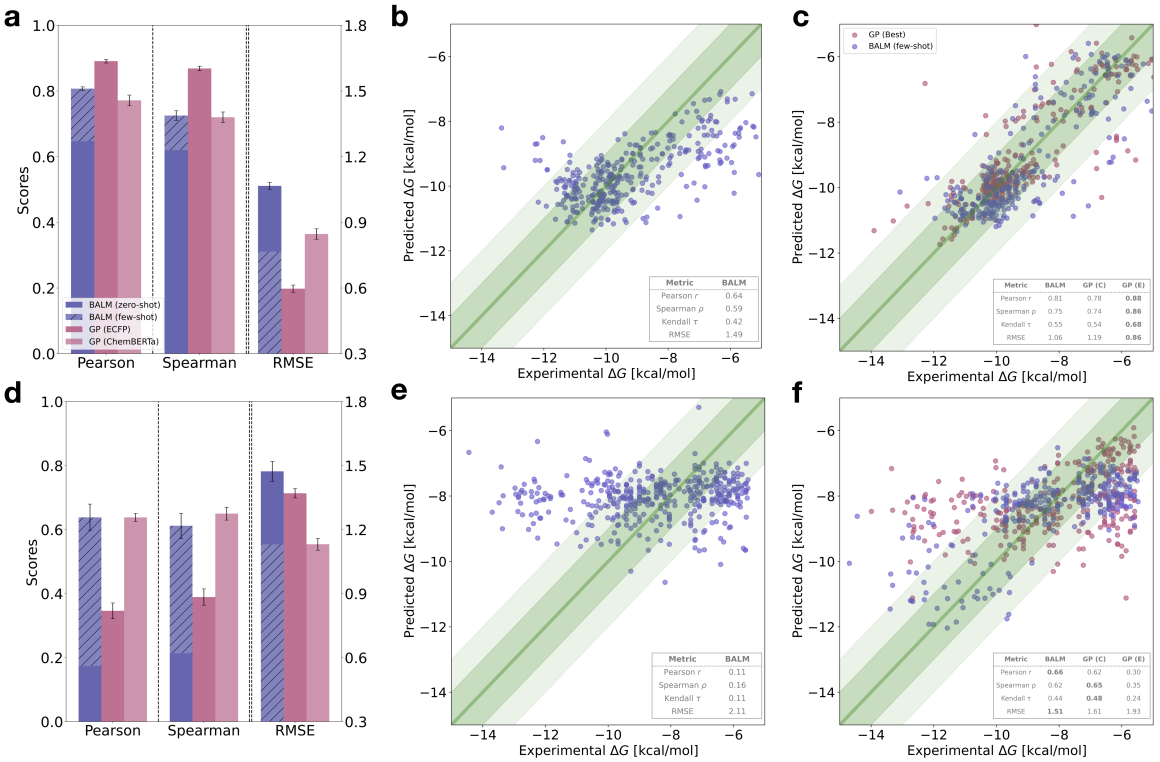


Figure 3: **Zero-shot performance of pre-trained BALM+PEFT model and few-shot comparison of BALM with Gaussian Process (GP) models on *USP7* and *Mpro* targets. (a, d)** Performance metrics for Pearson correlation, Spearman correlation, and RMSE on *USP7* (a) and *Mpro* (d) targets, comparing BALM in zero-shot (solid) and few-shot (patterned) settings with GP models trained using ECFP8 fingerprints (GP (E), Tanimoto kernel) and ChemBERTa embeddings (GP (C), RBF kernel). The pre-trained BALM+PEFT model is fine-tuned by retraining only the projection layer. Error bars indicate standard deviation over three random seeds. **(b, e)** Scatter plots showing zero-shot model predictions of pre-trained BALM+PEFT model for *USP7* (b) and *Mpro* (e) targets. Experimental ΔG values (kcal/mol) are on the x-axis and predicted ΔG values are on the y-axis. Only 20% of the test set (selected randomly) is shown for readability. **(c, f)** Few-shot BALM+PEFT predictions (blue) and GP predictions (GP (Best), red) for *USP7* (c) and *Mpro* (f) targets, highlighting BALM's robust performance across different GP configurations. We show 20% of the test set (selected randomly) for readability. GP (Best) refers to the optimal GP configuration for each target: GP (C) for *USP7* and GP (E) for *Mpro*, with metrics displayed for direct comparison.

often-used cumulative correlations over the whole dataset seems to overestimate the reliability of the model in cold target-setting and fails to highlight target-dependent deviations, making them a potentially misleading metric of the overall model performance. In contrast, averaging Fisher-transformed correlation coefficients and then back-transforming the average to the Pearson coefficient can help identify the greater dispersion across different targets (see Supplementary information for more details). The variability in the performance of zero-shot predictions is also replicated outside the BindingDB data set, as demonstrated for the test cases of the Ubiquitin carboxyl-terminal hydrolase 7[50] (USP7, *Q93009*) and the Main protease domain of the SARS-CoV-2 virus replicase polyprotein 1a (Mpro, *P0DTC1*).[51,52] In both cases we transformed the available experimental IC50 into pIC50. For evaluation, we computed a change in free energy ΔG of binding for a final comparison between experimental and computed values. Since in the single target sencario experiments come from a single assay we can use the Cheng-Prusoff equation to convert to ΔG, please see the supplementary information for more details. For the two single targets we obtained significantly different performance of the BALM-PEFT model: zero-shot $r = 0.64$ and RMSE of 1.49 kcal/mol for the USP7 case (Fig. 3b) and $r = 0.11$, and the RMSE is higher at 2.11 kcal/mol for the Mpro (Fig. 3e), indicating that zero-shot performance inconsistencies are related to the model and not the dataset composition. One thing to note is that the homodimer stoichiometry is not taken into account in training or for the prediction of this model.

With few-shot fine-tuning, the BALM framework rapidly adapts to new targets and can be used for screening compound libraries

While zero-shot performance is highly variable depending on the target, it is common in current drug discovery efforts to have access to a small set of high-quality data points from the very early stages of the hit-to-lead process. We hypothesized that in a real-life scenario, these data points could be incorporated into a pre-trained model to improve its adaptability to targets beyond the training set. To explore this extent, we used the BALM+PEFT

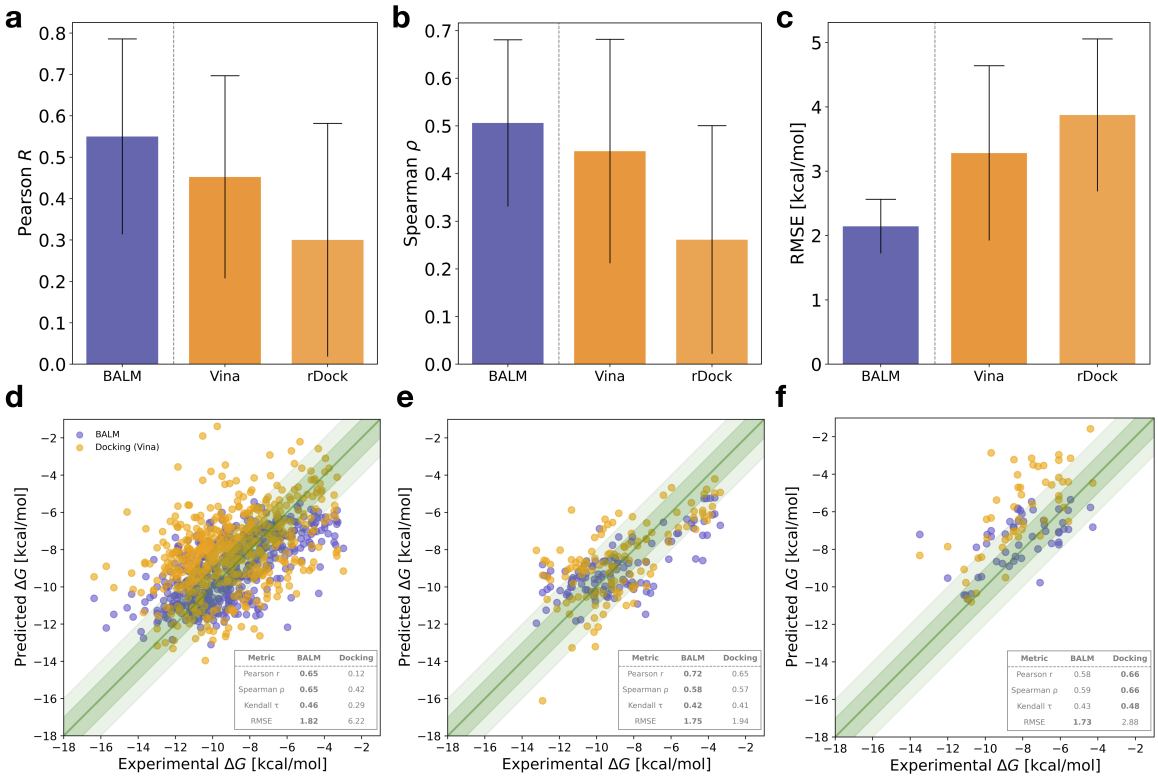


Figure 4: **Evaluation of BALM with respect to docking methods, AutoDock Vina[53] and rDock,[54] across various target families in the LP-PDBBind[55] test split.** The test set includes approximately 2,100 protein-ligand complexes spanning 12 diverse target families, with results shown here for representative families. Fisher-transformed metrics are used for comparing Pearson and Spearman correlations across target families, providing an accurate performance measure that accounts for variability within each family. AutoDock Vina consistently performs better than rDock across most target families. **(a)** Bar plot of Pearson correlation for each method across all families; error bars represent the standard deviation across target families. **(b)** Spearman correlation, showing the ranking accuracy of each method across families with similar error bars. **(c)** RMSE, directly comparing the predictive accuracy of each method, where lower values indicate higher precision in binding affinity predictions. **(d-f)** Scatter plots showing the performance of BALM (blue) versus AutoDock Vina (orange) for different target families. **(d)** For *Transferase* targets (545 PDB complexes), BALM achieves superior Pearson and Spearman correlations compared to Vina, demonstrating its effectiveness in predicting kinase-related binding affinities. **(e)** For *Chaperone* targets (119 PDB complexes), BALM outperforms Vina, especially in RMSE, which suggests higher prediction stability for this family. **(f)** In the case of *Oxidoreductase* targets (52 PDB complexes), docking methods show slightly better ranking performance, while BALM achieves a lower RMSE, indicating more precise binding affinity predictions for this target class.

model trained on BindingDB as the base model and fine-tuned only the linear projection layer, which accounts for 262,656 out of 152.5 million total parameters, or approximately 0.17% of the model's parameters. This should improve the adaptability of the model while being computationally efficient to implement. We compared the few-shot performance of BALM+PEFT with that of Gaussian Process (GP) models,[58] which are surrogate models that can produce reliable predictions with limited data[17,18] only reling on ligand information. We used two molecular representations to train the GP models: Extended-Connectivity Fingerprints (ECFP8) paired with a Tanimoto kernel and ChemBERTa embeddings paired with a radial basis function (RBF) kernel.

We evaluated the few-shot performance on the *USP7* (Fig. 3a,b,c) and *Mpro* (Fig. 3d,e,f), datasets. On the one hand, BALM+PEFT exhibits notable improvements in Pearson and Spearman correlations and a reduction in RMSE with few-shot fine-tuning using 20% of the data for training and the rest for testing. All the models are trained and tested with three different random seeds as before to estimate variability from training. In the zero-shot setting, BALM's Pearson correlation is 0.64 for *USP7* (Fig. 3a) and 0.11 for *Mpro* (Fig. 3d) that improves when trained on a few labelled examples to a Pearson correlation of 0.81 for *USP7* and 0.66 for *Mpro*, with RMSE values decreasing from 1.49 kcal/mol to 1.03 kcal/mol for *USP7* and from 2.11 kcal/mol to 1.51 kcal/mol for *Mpro*. On the other hand, the GP models, while displaying similar or even better accuracy than BALM+PEFT, are very sensitive to the choice of molecular representation, as shown in Fig. 3c and f. Specifically, in both datasets the Pearson correlation and RMSE worsen when switching from the ECFP8 fingerprint to the CheBERTa embedding. ECFP8+GP can be a naive starting point but will display more performance variability than a fine-tuned BALM+PEFT model. In contrast, the results obtained for BALM+PEFT demonstrate that, in a few-shot scenario, is a robust predictor independently of the choice of molecular representation. Furthermore, the model is very computationally efficient, with zero-shot predictions taking approximately 90 seconds per target for the 2,000 ligands in both USP7 and Mpro datasets on a single

Nvidia A100 GPU. The few-shot approach required between 14 and 25 additional minutes for retraining using between 10% and 30% of the data. Refer to Fig. S3 for a detailed performance comparison of the pre-trained BALM+PEFT model (zero-shot) and few-shot fine-tuning using 10%, 20%, and 30% of experimental data on *USP7* and *Mpro* targets. As such, BALM+PEFT's combination of zero-shot capability and robust performance with minimal additional data and fine-tuning on a small subset of parameters makes the model a robust tool for drug discovery screening scenarios, where rapid adaptation to novel targets and minimal dependency on feature engineering is essential to achieve the high-throughput required in current pharmaceutical industry settings.

BALM achieves better ranking than docking scores for a variety of target families

To further establish the potential of BALM+PEFT-like models to be incorporated in virtual screening pipelines, we investigated the performance of the BALM+PEFT on the LP-PDBBind dataset[55] and compared it to two well-established docking methods: AutoDock Vina[53] and rDock.[54] The LP-PDBBind dataset, which is derived from PDBBind v2020,[59] consists of 2,100 protein-ligand complexes from 12 protein families with experimentally measured binding affinities. LP-PDBBind is specifically designed to minimize similarities in sequence, structure, and ligand chemistry across training, validation, and test sets, minimizing data leakage and allowing the testing of the model's generalizability. For our evaluation, we employed the Clean Level 2 (CL2) split, as suggested by Li et al.,[55] due to its higher structural reliability.

In Fig. 4, we compare the general performance of zero-shot BALM+PEFT with that of both docking methods. For both docking methods existing crystal structure poses were scored with the most appropriate scoring function (see methods). Results are presented as bar plots for Fisher-transformed Pearson (Fig. 4a) and Spearman correlations (Fig. 4b), as well as RMSE (Fig. 4c) across target families. BALM+PEFT consistently achieves higher

Fisher-transformed Pearson and Spearman correlations compared to both docking methods, indicating stronger predictive and ranking performance across families. We also examined individual target families, comparing the performance of BALM+PEFT (in blue) versus Autodock Vina (Vina) (in orange) on three representative target families as identified in the LP-PDBBind dataset (Fig. 4d-f). For *Transferase* and *Chaperone* proteins (Fig. 4d), BALM+PEFT achieves Pearson correlation coefficients of 0.65 and 0.72 and RMSE values of 1.82 kcal/mol and 1,75 kcal/mol respectively, substantially outperforming Vina (Pearson r 0.12 / 0.65 and RMSE 6.22 kcal/mol / 1.94 kcal/mol). However, the ranking metrics for BALM ($\rho = 0.58$, $\tau = 0.42$) and Vina ($\rho = 0.57$, $\tau = 0.41$) are similar in the case of *Chaperone* proteins . For *Oxidoreductase* targets (Fig. 4f), Vina surpasses BALM+PEFT in ranking metrics (Spearman $\rho = 0.66$ vs. 0.59 for BALM), but BALM+PEFT achieves a lower RMSE of 1.73 kcal/mol versus Vina's 2.88 kcal/mol. In the case of more challenging protein families, such as those containing metals, membrane proteins, or transcription factors both methods achieve comparable results, although in general RMSE values are lower for the BALM+PEFT predictions. A full comparison across the 12 families is provided in the supplementary information both for BALM+PEFT vs. Vina (Fig. S4) and BALM+PEFT vs rDock (Fig. S5). The results suggest that BALM+PEFT offers a fast and high-accuracy alternative to scoring functions and, as such, can be envisaged as a tool in virtual screening protocols as a way to re-rank docking solutions before progressing to more computationally demanding methods, such as alchemical free energy methods. Furthermore, the performance of docking methods is very dependent on the availability of high-quality three-dimensional protein-ligand complex structures, while BALM+PEFT bypasses the need for that, making it particularly appealing for targets or compounds for which is not possible to obtain reliable three-dimensional structures.

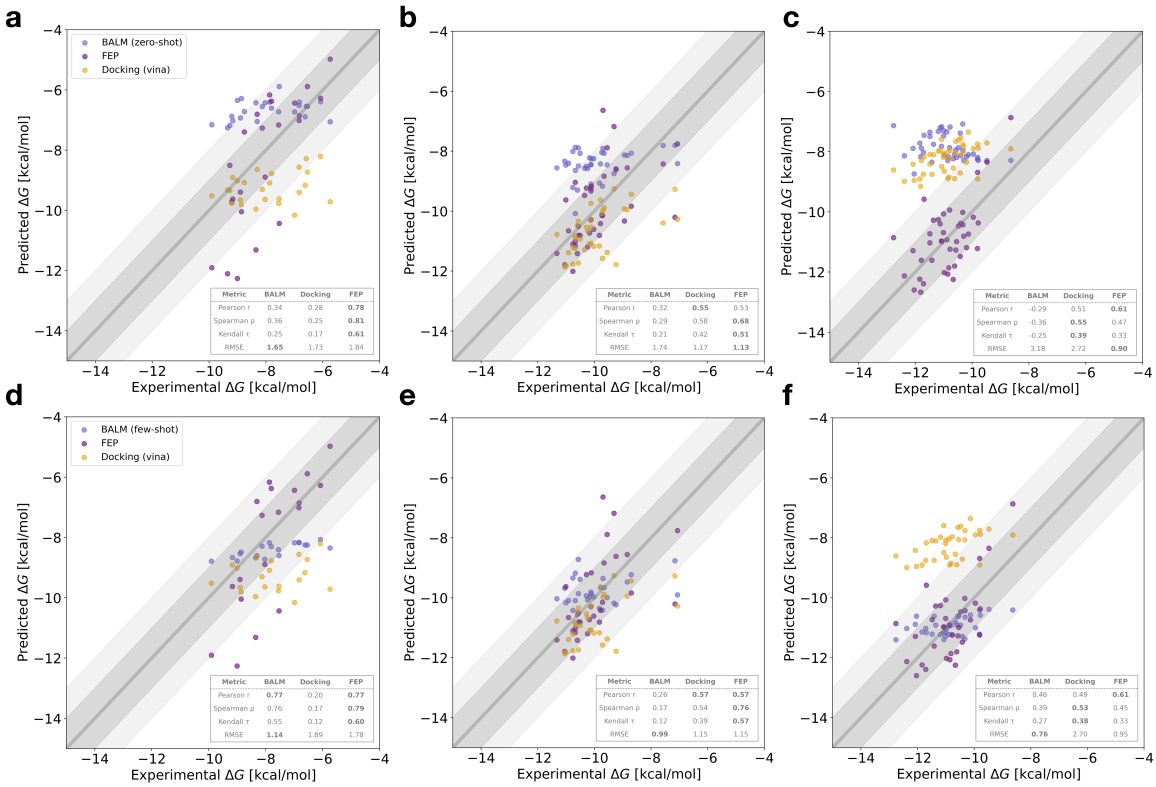


Figure 5: **Comparing BALM, Alchemical Free Energy (AFE) predictions, and AutoDock Vina across three benchmark protein-ligand targets.** The benchmark includes *MCL1* (25 ligands), *HIF2A* (37 ligands), and *SYK* (43 ligands). Panels **(a, b, c)** present the zero-shot predictions from BALM (blue) alongside AFE (yellow) and docking (orange) for each target. **(d, e, f)** illustrate the few-shot performance of BALM, fine-tuned with approximately 20% of the data. Each plot displays metrics for Pearson correlation, Spearman ρ, Kendall τ, and RMSE.

BALM on free energy benchmark datasets

Last, we compared the performance of BALM+PEFT with alchemical free energy (AFE) methods for the evaluation of relative binding free energies.[7] We used a subset of targets from the protein-ligand free energy benchmark curated by Hahn et al.,[8] which is composed of a congeneric series of ligands. Specifically, we selected three targets with the largest datasets, *MCL1*, *HIF2A*, and *SYK*, containing 25, 37, and 43 unique compounds, respectively, to compare BALM+PEFT's performance against AFE methods run as part of the OpenFE benchmark.[60] Additionally, we evaluated docking performance on these targets using AutoDock Vina, by rescoring the 3D poses presented in the benchmark. Fig. 5 shows BALM+PEFT's performance in zero-shot (a, b, c) and few-shot (d, e, f) modes for each target. In a zero-shot setting, the performance of the model is worse than both Vina and AFE, suggesting that although BALM+PEFT can predict binding free energy within a moderate error range, methods based on three-dimensional information can capture binding nuances more effectively, as recently pointed out by other authors.[5] The few-shot setting, using ca. 20% of the data for each target (5 ligands for *MCL1*, 7 for *HIF2A*, and 8 for *SYK*), generally reduces the RMSE across all targets, although the rank correlation trends are very target-dependent, being slightly better for *SYK* and *MCL1*, but worse for *HIF2A*. Regardless of the zero or few-shot setting, the predictions obtained with BALM+PEFT cluster within a narrower binding affinity range than Vina or AFE, suggesting either limitations in distinguishing small R-group modifications to a common chemical scaffold or an unintended optimization of its loss function, optimizing for RMSE at the cost of capturing the full dynamic range of binding energies. Globally, these results hint that BALM+PEFT faces challenges in effectively ranking congeneric series, which complicates its potential as a direct alternative to AFE for lead optimization stages. This may stem from the limited few-shot data (5-8 ligands), preventing the model from capturing subtle R-group differences and potentially focusing more on improving RMSE during training without maintaining rank order. As we used a small subset of data in this study, future exploration should involve using larger

datasets of congeneric series to better understand the impact of fine-tuning with data subsets of varying sizes. Additionally, implementing time-based splits of data collected over different periods could provide valuable insights by testing the model's performance retrospectively. These steps may help in refining the model's predictive capabilities and addressing its current limitations. However, the speed of inference and its higher accuracy with respect to methods such as molecular docking make the model an appealing alternative to popular re-ranking methods such as MM-GBSA,[61] aimed at filtering out low-quality compounds and obtaining a fast answer in those situations in which speed may trump accuracy.

Discussion

Drug discovery has many computational tools at its disposal in its early stages, from hit finding to lead-optimisation of a compound for a specific biomolecular target. Traditionally, these range from docking to simulation-based methods, and more recently include machine learning models such as AlphaFold3[62] to aid in structure generation of the desired protein targets. One key objective for computer-aided drug discovery is to successfully predict affinities and other ADMET properties (Absorption, Distribution, Metabolism, Excretion, and Toxicity) of large molecular libraries, and later of a lead molecule, towards a given biomolecular target. For affinity predictions, there has historically been a trade-off between prediction speed (e.g., through a docking score processing millions of compounds at a time) and accuracy (e.g., through alchemical free energy methods, processing 100s of compounds at a time). More recently, machine learning methods for affinity predictions have been taking an increasingly central stage by complementing traditional methods. A current issue is that many machine-learning-based affinity prediction methods are not tested against single targets, or evaluated on target-specific metrics. Issues in the methodologies adopted to assess recent AI-driven docking approaches were highlighted by the case of DiffDock, where a careful re-assessment of its performance according to best practices yielded lower success rates

than initially reported.[63,64] In this work, we propose fair and robust evaluation strategies to test models' performance and apply them to BALM, our new binding affinity prediction framework leveraging information on protein sequence and ligand SMILES.

As a sequence and SMILES-based model, BALM does not rely on structural data and can thus benefit from many large-language modelling tool developments such as parameter-efficient fine-tuning. In a target-specific setting, BALM with appropriate PEFT adapters outperforms baseline models on specific protein targets and can be further fine-tuned using a few-shot data scenario. This makes it a viable tool to be used alongside other computational methods for large-scale screening campaigns against individual targets. BALM is also fast and cost-effective, with few-shot learning on 100-300 compounds achieved in 10-20 minutes on typical GPU architectures such as Nvidia A100s. For targets without good crystallographic data available, BALM is a good alternative to traditional docking-based virtual screening methods. Current limitations preclude BALM from achieving alchemical free energy accuracy on congeneric ligand series often found in lead-optimisation campaigns. This can be addressed in multiple ways in the future looking at diverse ligand embeddings and other fine-tuning strategies. While its current performance does not provide a full replacement of traditional methods, BALM can be used alongside a docking campaign for consensus scoring and even replacement of computationally costly docking protocols where poor structural data is available.

Overall, BALM is likely to perform well in scenarios of single globular proteins with sufficient data availability for both fine-tuning and few-shot learning. In situations where proteins are homodimers, multimers, form part of a membrane, or have otherwise more complicated behavior (e.g., binding mediated by co-factors), BALM can complement more advanced strategies such as AFE. All of these situations constitute realistic drug discovery scenarios but may be captured well with language model-based approaches. For this reason, it is crucial that unbiased evaluation strategies are adopted to assess the performance of ML models. In this context, we show that looking at target-specific evaluation metrics and

leak-free training of the framework is key to determining how well an ML model performs on a realistic affinity prediction task. In this work, we set a standard baseline on how ML-based affinity models should be evaluated. Only through the broad adoption of unbiased evaluation strategies such as those presented in our work, the community will obtain a real sense of the advance of machine-learning-based affinity predictions.

Methods

Model overview and objective. BALM is designed to predict protein-ligand binding affinity by taking ligand SMILES strings and protein sequences as the input. Our method begins by encoding the protein sequences and ligand SMILES by integrating protein and ligand language models. These language models trained on extensive ligand[40] and protein databases[39] encapsulate the chemical properties inherent in SMILES strings and physicochemical, functional, and evolutionary patterns in protein sequences. The encodings obtained from the language models are then projected through a linear layer to a shared latent space. The core idea behind BALM is to maximize the cosine similarity between protein-ligand embeddings for a binding interaction and minimize the same for non-binding interactions, thus learning protein-ligand interaction features. The model learns to understand the relationship between protein and ligand embeddings and quantifies it through a cosine similarity score, a proxy for the binding affinity score (pK_d). The model learns by minimizing the mean-squared error (MSE) loss.

Encoding and projecting protein and ligand features. We use ESM-2[32] model for encoding protein features and ChemBERTa-2[33] for encoding ligands. It is important to note that our focus is not to evaluate the performance of protein or ligand language models but to leverage their learned representations. The ESM-2 model is trained on sequences from the UniRef database[39] with 65 million unique sequences using bidirectional transformer architecture.[65] ESM-2 learns in an unsupervised manner through masked language modelling

(MLM) objective, where 15% of amino acids are masked in the input sequence, and it is tasked with predicting these missing positions. We use the 150 million parameter model as it strikes a balance between computational efficiency and performance.

ChemBERTa-2 is based on the RoBERTa[66] transformer model, using both MLM similar to ESM-2 by masking 15% tokens and multi-task regression (MTR) pretraining on 200 downstream molecular properties. We use the model trained on the largest dataset size using the MTR objective as it outperformed the other model configurations in the comparative study.[33] For a given input protein sequence, I_P, and ligand SMILES, I_L, we encode and project to get embeddings of the same dimensions as shown below:

$$\mathbf{x}_{\mathrm{P}} = \mathbf{W}_{\mathrm{P}} \cdot f(I_{\mathrm{P}}) + \mathbf{b}_{\mathrm{P}}, \tag{1}$$

$$\mathbf{x}_{\mathrm{L}} = \mathbf{W}_{\mathrm{L}} \cdot g(I_{\mathrm{L}}) + \mathbf{b}_{\mathrm{L}}. \tag{2}$$

Here, $f(I_{\mathrm{P}}) \in R^{d_{\mathrm{P}}}$ extracts the protein features using ESM-2 and similarly, $g(I_{\mathrm{L}}) \in R^{d_{\mathrm{L}}}$ extracts the ligand features using ChemBERTa-2. We then transform them separately into $\mathbf{x}_{\mathrm{P}}, \mathbf{x}_{\mathrm{L}} \in R^K$ using a single fully connected layer (with a ReLU activation). These layers are parameterized with weight matrices $W_{\mathrm{P}} \in R^{K \times d_{\mathrm{P}}}$ for proteins and $W_{\mathrm{L}} \in R^{K \times d_{\mathrm{L}}}$ for ligands, and bias vectors $\mathbf{b}_{\mathrm{P}}, \mathbf{b}_{\mathrm{L}} \in R^K$. As the encodings derived from the ESM-2 and the ChemBERTa-2 model have different dimensions, 640 (150M model) and 384, respectively, we employ linear layers to map both sets of embeddings to a shared latent space of dimension K (256) suitable for similarity computations. Both ESM-2 and ChemBERTa-2 models are accessible via the Hugging face transformers library,[67] under the model names `facebook/esm2_t30_150M_UR50D`, and `DeepChem/ChemBERTa-77M-MTR`, respectively.

Training with cosine similarity and MSE. In the training phase of BALM, the cosine similarity metric is used to measure the affinity between the projected protein and ligand embeddings. BALM utilizes cosine similarity as the core metric for learning protein-ligand binding as a regression problem. BALM defines the binding score by computing the cosine

similarity of the protein and ligand embeddings:

$$\sigma(\mathbf{x}_{\mathrm{P}}, \mathbf{x}_{\mathrm{L}}) = \frac{\mathbf{x}_{\mathrm{P}} \cdot \mathbf{x}_{\mathrm{L}}}{\|\mathbf{x}_{\mathrm{P}}\| \|\mathbf{x}_{\mathrm{L}}\|}, \tag{3}$$

where $\mathbf{x}_{\mathrm{P}}$ and $\mathbf{x}_{\mathrm{L}}$ denote the embeddings of the protein and ligand, respectively. The cosine similarity σ ranges from -1 to 1, with values closer to 1 indicating a stronger binding (high pK_d). $\mathbf{x}_{\mathrm{P}}^i$ and $\mathbf{x}_{\mathrm{L}}^i$ are the projected embeddings of the protein and ligand, respectively. BALM is trained by minimizing the MSE between predicted binding scores given by $\sigma(\mathbf{x}_{\mathrm{P}}, \mathbf{x}_{\mathrm{L}})$ and experimental affinities (y_i) adjusted to the -1 to 1 scale :

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \sigma(\mathbf{x}_{\mathrm{P}}^i, \mathbf{x}_{\mathrm{L}}^i) \right)^2, \tag{4}$$

where N is the number of samples.

Parameter-efficient fine-tuning on BALM. PEFT allows adapting pre-trained language models to new domains by fine-tuning a small subset of additional parameters while keeping the initial model parameters fixed. This reduces computational requirements and prevents catastrophic forgetting. Given a pre-trained model T with parameters θ, and a downstream task $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, where (x_i, y_i) represents an input and ground-truth pair for task $\mathcal{D}$, PEFT introduces a small set of trainable parameters $\Delta\theta$, where $|\Delta\theta| \ll |\theta|$. The objective is to adapt θ to task $\mathcal{D}$ by optimizing:

$$\min_{\Delta\theta} E_{(x_i, y_i) \in \mathcal{D}} \mathcal{L}(T_{\theta + \Delta\theta}(\hat{y}_i | x_i), y_i), \tag{5}$$

where $\hat{y}_i$ denotes the predicted affinities and $\mathcal{L}$ denotes the MSE loss function reflecting the model's performance on task $\mathcal{D}$. Various PEFT methods have been proposed, for more details refer to the survey papers.[41,42]

In this work, we use the reparameterised and additive PEFT methods[41,42] to fine-tune BALM's protein (ESM-2) and ligand (ChemBERTa-2) language models for binding affinity

prediction. The reparameterized PEFT methods we used in the study include Low-Rank Adaptation (LoRA),[68] LoRA with Hadamard product (LoHa),[69] and LoRA with Kronecker product (LoKr),[70] applied to the key, query, and value matrices. An additive PEFT method, Infused Adapter by Inhibiting and Amplifying Inner Activations (IA^3),[71] is also studied for fine-tuning BALM. These methods are implemented using the Hugging face `PeftModel` class.[67,72] We use four PEFT methods in our study, and these are discussed in detail in the Appendix **??**.

Data for the study

To benchmark the performance of our models, we utilized several publicly available datasets, including BindingDB,[47,48] LP-PDBBind,[55] and other datasets specific to protein-ligand systems such as *USP7*,[50] *Mpro*,[51,52] and three targets from protein-ligand free energy benchmark[8] - *MCL1*, *HIF2A*, and *SYK*. These datasets encompass a wide range of binding affinity measurements and chemical diversity, enabling us to compare with docking and alchemical free energy methods. These datasets provide a comprehensive benchmark for strategically evaluating the machine learning model's capability to predict binding affinity under various conditions, including zero-shot and few-shot scenarios. We release all the cleaned datasets used in our work as `BALM-Benchmark` on Hugging face.[67]

BindingDB dataset provides experimentally measured binding affinities of protein-ligand interactions, we focus on the K_d version[48] due to inconsistencies in IC50 and K_i measures.[30] This version contains 52,284 interactions with 10,665 ligand SMILES and 1,413 protein sequences. We filter out sequences with more than 1024 residues for computational efficiency with the ESM-2 model.[32] The K_d values were transformed into pK_d for stable training, following previous works.[24,29,35] We were left with approximately 1100 targets and 48,000 interaction data after the filtering. Additionally, to avoid bias from assay limits (Fig. S1), we removed the top five most frequent limits. This reduced the dataset to about 25,000 interactions involving approximately 1,070 targets and 9,200 ligands. We used around 70%

of these interaction data for training, 10% interactions for validation, and 20% interactions for testing. Four data splits (Random, Cold Target, Cold Drug, and Scaffold) were used to evaluate model generalizability. In the *random split*, commonly used in prior works,[24,29,35] protein-ligand interaction pairs are randomly distributed across training, validation, and test sets, ensuring each subset is a statistically representative sample. The *cold target split* segregates protein targets into distinct groups for training, validation, and testing, allowing evaluation on unseen protein targets. Similarly, the *cold drug split* randomly allocates drugs to training, validation, and test sets, assigning all protein-ligand pairs linked to each ligand to the corresponding set, enabling performance evaluation on unseen drugs. Lastly, the *scaffold split* employs the Murcko scaffolds[56] concept, using the `MurckoScaffold` module from RDkit[57] to bin drugs by their core scaffolds, ensuring no structural overlap across sets. Approximately 4570 scaffolds were used from the BindingDB dataset for this purpose.

LP-PDBBind dataset is derived from the publicly available PDBBind v2020[59] dataset, a curated set of approximately 20,000 protein-ligand complex structures (3D) with experimentally measured binding affinities. The dataset has been reorganized to minimize sequence, structural interaction patterns, and chemical similarity across training, validation, and test splits.[55] It has also been cleaned to remove covalent-bound ligand-protein complexes, ligands with rare atomic elements, and structures with steric clashes while maintaining consistency in reported binding free energies. Recently, Jores et al.[73] provided more detailed analysis of the LP-PDBBind's data splits. Furthermore, the dataset has been categorized based on the quality of structures into Clean Levels 1, 2, and 3, consisting of 14,324, 7,985, and 4,404 entries, respectively. We utilize CL1 for training, while CL2 is employed for validation and testing due to its higher reliability, as suggested by Li et al.[55]

USP7 dataset is curated by Shen et al.,[50] evaluates inhibitors targeting the ubiquitin-specific protease 7 (*USP7*). The dataset consists of over 4000 ligands with associated binding affinities collected from ChEMBL.[74] After processing, the final dataset comprises 1,799 unique ligands with measured experimental affinities, represented as IC50 values. These

values were then transformed into pIC50 values for uniformity and stability in training.

Mpro dataset is derived from the COVID Moonshot project focusing on inhibitors of the SARS-CoV-2 main protease (*Mpro*),[51,52] includes data from multiple design sprints. We filtered to remove assay limits, and the cleaned dataset contains 2,062 unique ligands with experimentally determined IC50 values. We converted IC50 values to pIC50 values similar to the *USP7* target. Further details about the data curation can be found here.[51,52]

Protein-ligand free energy benchmark curated by Hahn et al.[8] contains about 21 targets for benchmarking alchemical free energy calculations. We selected three targets, *MCL1*, *HIF2A*, and *SYK*, to benchmark the machine learning model performance as compared to the alchemical energy calculations,[60] provided through the OpenFE (open free energy) consortium. The *HIF2A* dataset contains 37 ligands, the *MCL1* dataset includes 25 ligands, and the *SYK* dataset consists of 43 unique ligands.

Acknowledgement

RG and APG were supported by the United Kingdom Research and Innovation (grant EP/S02431X/1), UKRI Centre for Doctoral Training in Biomedical AI at the University of Edinburgh, School of Informatics. RG was also supported by Exscientia Plc, Oxford, now Recusrion. Experiments from this work are conducted mainly on the Edinburgh International Data Facility[1] and supported by the Data-Driven Innovation Programme at the University of Edinburgh.

Supporting Information Available

Supporting Information Available: Dataset analysis, additional experimental results and methods used are provided.

[1]https://edinburgh-international-data-facility.ed.ac.uk/

Data and Code Availability

All datasets curated for this study are publicly available via the Hugging Face `BALM-Benchmark` collection. The datasets can be accessed at `https://huggingface.co/datasets/BALM/BALM-benchmark`, and the pre-trained models are accessible on Hugging Face at `https://huggingface.co/BALM`. All the code containing scripts for data processing, model training, and evaluation is publicly accessible on GitHub at `https://github.com/meyresearch/BALM`

References

(1) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. Y.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K.; Tolmachev, A. A. Ultra-large library docking for discovering new chemotypes. *Nature* **2019**, *566*, 224–229.

(2) Stanzione, F.; Giangreco, I.; Cole, J. C. Use of molecular docking computational tools in drug discovery. *Prog. Med. Chem.* **2021**, *60*, 273–343.

(3) Shoichet, B. K. Virtual screening of chemical libraries. *Nature* **2004**, *432*, 862–865.

(4) Beroza, P.; Crawford, J. J.; Ganichkin, O.; Gendelev, L.; Harris, S. F.; Klein, R.; Miu, A.; Steinbacher, S.; Klingler, F.-M.; Lemmen, C. Chemical space docking enables large-scale structure-based virtual screening to discover ROCK1 kinase inhibitors. *Nat. Commun.* **2022**, *13*, 6447.

(5) Errington, D.; Schneider, C.; Bouysset, C.; Dreyer, F. A. Assessing interaction recovery of predicted protein-ligand poses. *arXiv:2409.20227* **2024**,

(6) Robo, M. T.; Hayes, R. L.; Ding, X.; Pulawski, B.; Vilseck, J. Z. Fast free energy estimates from λ-dynamics with bias-updated Gibbs sampling. *Nat. Commun.* **2023**, *14*, 8515.

(7) Mey, A. S.; Allen, B. K.; Macdonald, H. E. B.; Chodera, J. D.; Hahn, D. F.; Kuhn, M.; Michel, J.; Mobley, D. L.; Naden, L. N.; Prasad, S.; Rizzi, A.; Scheen, J.; Shirts, M. R.; Tresadern, G.; Xu, H. Best Practices for Alchemical Free Energy Calculations. *Living J. Mol. Sci.* **2020**, *2*, 18378.

(8) Hahn, D. F.; Bayly, C. I.; Boby, M. L.; Macdonald, H. E. B.; Chodera, J. D.; Gapsys, V.; Mey, A. S.; Mobley, D. L.; Benito, L. P.; Schindler, C. E.; Tresadern, G.; Warren, G. L. Best practices for constructing, preparing, and evaluating protein-ligand binding affinity benchmarks. *Living J. Mol. Sci.* **2022**, *4*, 1497–1497.

(9) Sadybekov, A. V.; Katritch, V. Computational approaches streamlining drug discovery. *Nature* **2023**, *616*, 673–685.

(10) Grygorenko, O. O.; Radchenko, D. S.; Dziuba, I.; Chuprina, A.; Gubina, K. E.; Moroz, Y. S. Generating multibillion chemical space of readily accessible screening compounds. *Iscience* **2020**, *23*.

(11) Hadfield, T. E.; Scantlebury, J.; Deane, C. M. Exploring the ability of machine learning-based virtual screening models to identify the functional groups responsible for binding. *J. Cheminformatics* **2023**, *15*, 84.

(12) Zhang, Y.; Li, S.; Meng, K.; Sun, S. Machine Learning for Sequence and Structure-Based Protein–Ligand Interaction Prediction. *J. Chem. Inf. Model.* **2024**,

(13) Kimber, T. B.; Chen, Y.; Volkamer, A. Deep learning in virtual screening: recent applications and developments. *Int. J. Mol. Sci.* **2021**, *22*, 4435.

(14) Smer-Barreto, V.; Quintanilla, A.; Elliott, R. J. R.; Dawson, J. C.; Sun, J.; Campa, V. M.; Lorente-Macías, Á.; Unciti-Broceta, A.; Carragher, N. O.; Acosta, J. C.; others Discovery of senolytics using machine learning. *Nat. Commun.* **2023**, *14*, 3445.

(15) Rodriguez, S.; Hug, C.; Todorov, P.; Moret, N.; Boswell, S. A.; Evans, K.; Zhou, G.; Johnson, N. T.; Hyman, B. T.; Sorger, P. K.; others Machine learning identifies candidates for drug repurposing in Alzheimer's disease. *Nat. Commun.* **2021**, *12*, 1033.

(16) Gusev, F.; Gutkin, E.; Kurnikova, M. G.; Isayev, O. Active learning guided drug design Lead optimization based on relative binding free energy modeling. *J. Chem. Inf. Model.* **2023**, *63*, 583–594.

(17) Thompson, J.; Walters, W. P.; Feng, J. A.; Pabon, N. A.; Xu, H.; Goldman, B. B.; Moustakas, D.; Schmidt, M.; York, F. Optimizing active learning for free energy calculations. *Artif. Intell. Life Sci.* **2022**, *2*, 100050.

(18) Gorantla, R.; Kubincova, A.; Suutari, B.; Cossins, B. P.; Mey, A. S. Benchmarking active learning protocols for ligand binding affinity prediction. *J. Chem. Inf. Model.* **2024**, *64*, 1955–1965.

(19) Huang, L.; Xu, T.; Yu, Y.; Zhao, P.; Chen, X.; Han, J.; Xie, Z.; Li, H.; Zhong, W.; Wong, K.-C.; others A dual diffusion model enables 3D molecule generation and lead optimization based on target pockets. *Nat. Commun.* **2024**, *15*, 2657.

(20) Anstine, D. M.; Isayev, O. Generative models as an emerging paradigm in the chemical sciences. *J. Am. Chem. Soc.* **2023**, *145*, 8736–8750.

(21) Sattari, K.; Li, D.; Kalita, B.; Xie, Y.; Lighvan, F. B.; Isayev, O.; Lin, J. De novo molecule design towards biased properties via a deep generative framework and iterative transfer learning. *Digit. Discov.* **2024**,

(22) Runcie, N. T.; Mey, A. S. SILVR: Guided diffusion for molecule generation. *J. Chem. Inf. Model.* **2023**, *63*, 5996–6005.

(23) Chatterjee, A.; Walters, R.; Shafi, Z.; Ahmed, O. S.; Sebek, M.; Gysi, D.; Yu, R.; Eliassi-

Rad, T.; Barabási, A.-L.; Menichetti, G. Improving the generalizability of protein-ligand binding predictions with AI-Bind. *Nat. Commun.* **2023**, *14*, 1989.

(24) Öztürk, H.; Özgür, A.; Ozkirimli, E. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics* **2018**, *34*, 821–829.

(25) Jiménez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *J. Chem. Inf. Model.* **2018**, *58*, 287–296.

(26) Shen, L.; Feng, H.; Li, F.; Lei, F.; Wu, J.; Wei, G.-W. Knot data analysis using multiscale Gauss link integral. *Proc. Natl. Acad. Sci. USA* **2024**, *121*, e2408431121.

(27) Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *J. Med. Chem.* **2004**, *47*, 2977–2980.

(28) Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. On the frustration to predict binding affinities from protein–ligand structures with deep neural networks. *J. Med. Chem.* **2022**, *65*, 7946–7958.

(29) Gorantla, R.; Kubincova, A.; Weiße, A. Y.; Mey, A. S. From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction. *J. Chem. Inf. Model.* **2024**, *64*, 2496–2507.

(30) Landrum, G. A.; Riniker, S. Combining IC50 or K i Values from Different Sources Is a Source of Significant Noise. *J. Chem. Inf. Model.* **2024**,

(31) Backenköhler, M.; Groß, J.; Wolf, V.; Volkamer, A. Guided docking as a data generation approach facilitates structure-based machine learning on kinases. *J. Chem. Inf. Model.* **2024**, *64*, 4009–4020.

(32) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; others Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **2023**, *379*, 1123–1130.

(33) Ahmad, W.; Simon, E.; Chithrananda, S.; Grand, G.; Ramsundar, B. Chemberta-2: Towards chemical foundation models. *arXiv:2209.01712* **2022**,

(34) Nguyen, T.; Le, H.; Quinn, T. P.; Nguyen, T.; Le, T. D.; Venkatesh, S. GraphDTA: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics* **2021**, *37*, 1140–1147.

(35) Jiang, M.; Li, Z.; Zhang, S.; Wang, S.; Wang, X.; Yuan, Q.; Wei, Z. Drug–target affinity prediction using graph neural network and contact maps. *RSC Adv.* **2020**, *10*, 20701–20712.

(36) Luo, D.; Liu, D.; Qu, X.; Dong, L.; Wang, B. Enhancing Generalizability in Protein–Ligand Binding Affinity Prediction with Multimodal Contrastive Learning. *J. Chem. Inf. Model.* **2024**,

(37) Kaufman, B.; Williams, E. C.; Underkoffler, C.; Pederson, R.; Mardirossian, N.; Watson, I.; Parkhill, J. Coati: Multimodal contrastive pretraining for representing and traversing chemical space. *J. Chem. Inf. Model.* **2024**, *64*, 1145–1157.

(38) Singh, R.; Sledzieski, S.; Bryson, B.; Cowen, L.; Berger, B. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proc. Natl. Acad. Sci. U.S.A.* **2023**, *120*, e2220778120.

(39) Suzek, B. E.; Wang, Y.; Huang, H.; McGarvey, P. B.; Wu, C. H.; Consortium, U. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **2015**, *31*, 926–932.

(40) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; E Bolton, E. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **2019**, *47*, D1102–D1109.

(41) Han, Z.; Gao, C.; Liu, J.; Zhang, S. Q. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. *arXiv:2403.14608* **2024**,

(42) Xu, L.; Xie, H.; Qin, S.-Z. J.; Tao, X.; Wang, F. L. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv:2312.12148* **2023**,

(43) Sultan, A.; Sieg, J.; Mathea, M.; Volkamer, A. Transformers for molecular property prediction: Lessons learned from the past five years. *J. Chem. Inf. Model.* **2024**, *64*, 6259–6280.

(44) Dutt, R.; Ericsson, L.; Sanchez, P.; Tsaftaris, S. A.; Hospedales, T. Parameter-efficient fine-tuning for medical image analysis: The missed opportunity. *arXiv:2305.08252* **2023**,

(45) Gema, A.; Daines, L.; Minervini, P.; Alex, B. Parameter-efficient fine-tuning of llama for the clinical domain. *arXiv:2307.03042* **2023**,

(46) Gema, A. P.; Hong, G.; Minervini, P.; Daines, L.; Alex, B. Edinburgh Clinical NLP at SemEval-2024 Task 2: Fine-tune your model unless you have access to GPT-4. *arXiv:2404.00484* **2024**,

(47) Gilson, M. K.; Liu, T.; Baitaluk, M.; Nicola, G.; Hwang, L.; Chong, J. BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res.* **2016**, *44*, D1045–D1053.

(48) Huang, K.; Fu, T.; Gao, W.; Zhao, Y.; Roohani, Y.; Leskovec, J.; Coley, C. W.; Xiao, C.;

Sun, J.; Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv:2102.09548* **2021**,

(49) Fisher, R. Frequency Distribution of the Values of the Correlation Coefficient in Samples from an Indefinitely Large Population. *Biometrika* **1915**, *10*, 507–521.

(50) Shen, W.-f.; Tang, H.-w.; Li, J.-b.; Li, X.; Chen, S. Multimodal data fusion for supervised learning-based identification of USP7 inhibitors: a systematic comparison. *J. Cheminform.* **2023**, *15*, 1–16.

(51) Achdout, H.; Aimon, A.; Bar-David, E.; Morris, G. COVID moonshot: open science discovery of SARS-CoV-2 main protease inhibitors by combining crowdsourcing, high-throughput experiments, computational simulations, and machine learning. *BioRxiv* **2020**,

(52) Boby, M. L.; Fearon, D.; Ferla, M.; Filep, M.; Koekemoer, L.; Robinson, M. C.; Consortium, T. C. M.; Chodera, J. D.; Lee, A. A.; London, N.; von Delft, A.; von Delft, F. Open science discovery of potent noncovalent SARS-CoV-2 main protease inhibitors. *Science* **2023**, *382*, eabo7201.

(53) Trott, O.; Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **2010**, *31*, 455–461.

(54) Ruiz-Carmona, S.; Alvarez-Garcia, D.; Foloppe, N.; Gago, F.; Gervais, G.; Irwin, J.; Sverrisson, F.; Tounge, B.; Tresadern, G.; Morley, S. rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLoS Comput. Biol.* **2014**, *10*, e1003571.

(55) Li, J.; Guan, X.; Zhang, O.; Sun, K.; Wang, Y.; Bagni, D.; Head-Gordon, T. Leak Proof PDBBind: A Reorganized Dataset of Protein-Ligand Complexes for More Generalizable Binding Affinity Prediction. *arXiv:2308.09639* **2023**,

(56) Bemis, G. W.; Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **1996**, *39*, 2887–2893.

(57) Bento, A. P.; Hersey, A.; Félix, E.; Landrum, G.; Gaulton, A.; Atkinson, F.; Bellis, L. J.; De Veij, M.; Leach, A. R. An open source chemical structure curation pipeline using RDKit. *J. Cheminform.* **2020**, *12*, 1–16.

(58) Deringer, V. L.; Bartók, A. P.; Bernstein, N.; Wilkins, D. M.; Ceriotti, M.; Csányi, G. Gaussian process regression for materials and molecules. *Chem. Rev.* **2021**, *121*, 10073–10141.

(59) Liu, Z.; Li, Y.; Han, L.; Li, J.; Liu, J.; Zhao, Z.; Nie, W.; Liu, Y.; Wang, R. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics* **2015**, *31*, 405–412.

(60) Baumann, H. M.; Henry, M. M.; Ries, B.; Swenson, D. W. H.; Eastwood, J. R. B.; Gowers, R. J.; Alibay, I. Openfe 1.0rc Release: Benchmarking Results. 2024; `https://doi.org/10.5281/zenodo.13959654`, Accessed: October 21, 2024.

(61) Genheden, S.; Ryde, U. The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities. *Expert Opin. Drug Discov.* **2015**, *10*, 449–461.

(62) Abramson, J.; Adler, J.; Dunger, J.; Evans, R.; Green, T.; Pritzel, A.; Ronneberger, O.; Willmore, L.; Ballard, A. J.; Bambrick, J.; others Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **2024**, *630*, 493–500.

(63) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv:2210.01776* **2022**,

(64) Jain, A. N.; Cleves, A. E.; Walters, W. P. Deep-Learning Based Docking Methods: Fair Comparisons to Conventional Docking Workflows. *arXiv:2412.02889* **2024**,

(65) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805* **2018**,

(66) Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692* **2019**,

(67) Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; others Huggingface's transformers: State-of-the-art natural language processing. *arXiv:1910.03771* **2019**,

(68) Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv:2106.09685* **2021**,

(69) Yeh, S.-Y.; Hsieh, Y.-G.; Gao, Z.; Yang, B. B.; Oh, G.; Gong, Y. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. *arXiv:2309.14859* **2023**,

(70) He, X.; Li, C.; Zhang, P.; Yang, J.; Wang, X. E. Parameter-Efficient Model Adaptation for Vision Transformers. Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023. 2023; pp 817–825.

(71) Liu, H.; Tam, D.; Muqeeth, M.; Mohta, J.; Huang, T.; Bansal, M.; Raffel, C. A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 1950–1965.

(72) Mangrulkar, S.; Gugger, S.; Debut, L.; Belkada, Y.; Paul, S.; Bossan, B. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. `https://github.com/huggingface/peft`, 2022.

(73) Joeres, R.; Blumenthal, D. B.; Kalinina, O. V. DataSAIL: Data Splitting Against Information Leakage. *bioRxiv* **2023**, 2023–11.

(74) Mendez, D. et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* **2019**, *47*, D930–D940.

Chapter 6

Benchmarking Active Learning Protocols for Ligand Binding Affinity Prediction

This chapter is based on the work described in the following publication - Gorantla, R., Kubincova, A., Suutari, B., Cossins, B. P., & Mey, A. S. *J. Chem. Inf. Model.* **64**, 6, 1955–1965 (2024).

In this chapter, I shifted focus from developing machine learning models for large-scale screening to supporting high-accuracy physics-based methods such as alchemical free energy (AFE) calculations, which are used in lead optimisation stages. While AFE calculations provide excellent accuracy for free energy predictions (RMSE $\sim$ 1 kcal/mol), their computational cost makes them impractical for evaluating large compound sets during lead optimization. To address this limitation, I investigated how machine learning, specifically active learning (AL), can be used to intelligently prioritize compounds for both AFE calculations and experimental assays, reducing the overall computational and experimental burden in lead optimization.

Active learning involves several key components, including the choice of surrogate models, acquisition functions for compound selection, and strategies for balancing exploration and exploitation of chemical space during iterative learning. The in-

vestigation systematically evaluates various components of AL workflows using four diverse protein targets (TYK2, USP7, D2R, Mpro). I assess different surrogate models, including Gaussian Process regression and pretrained graph neural networks, analyzing their effectiveness in guiding compound selection under varying conditions of data availability. I assessed the effect of parameters such as initial training set size, batch sizes for iterative learning, and strategies for balancing exploration versus exploitation of chemical space.

Through comprehensive benchmarking experiments, I demonstrate how different AL protocols influence the identification of top binders while minimizing the number of required AFE calculations. The findings provide practical guidelines for implementing these hybrid approaches in lead optimization campaigns, showing how machine learning can effectively support physics-based methods by reducing computational overhead while maintaining prediction reliability.

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

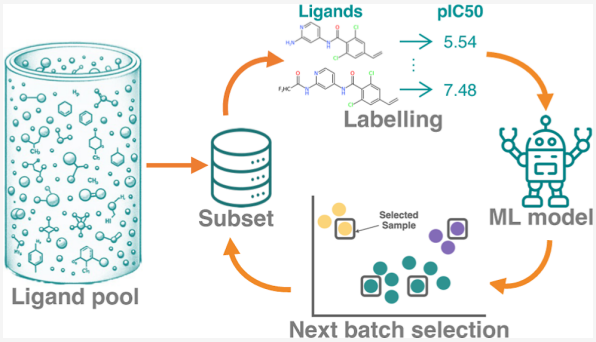
pubs.acs.org/jcim Article

Benchmarking Active Learning Protocols for Ligand-Binding Affinity Prediction

Rohan Gorantla,[‖] Alžbeta Kubincová,[‖] Benjamin Suutari, Benjamin P. Cossins, and Antonia S. J. S. Mey*

Cite This: J. Chem. Inf. Model. 2024, 64, 1955−1965 Read Online

ABSTRACT: Active learning (AL) has become a powerful tool in computational drug discovery, enabling the identification of top binders from vast molecular libraries. To design a robust AL protocol, it is important to understand the influence of AL parameters, as well as the features of the data sets on the outcomes. We use four affinity data sets for different targets (TYK2, USP7, D2R, Mpro) to systematically evaluate the performance of machine learning models [Gaussian process (GP) model and Chemprop model], sample selection protocols, and the batch size based on metrics describing the overall predictive power of the model (R2, Spearman rank, root-mean-square error) as well as the accurate identification of top 2%/5% binders (Recall, F1 score). Both models have a comparable Recall of top binders on large data sets, but the GP model surpasses the Chemprop model when training data are sparse. A larger initial batch size, especially on diverse data sets, increased the Recall of both models as well as overall correlation metrics. However, for subsequent cycles, smaller batch sizes of 20 or 30 compounds proved to be desirable. Furthermore, adding artificial Gaussian noise to the data up to a certain threshold still allowed the model to identify clusters with top-scoring compounds. However, excessive noise (<1σ) did impact the model's predictive and exploitative capabilities.



INTRODUCTION

Active learning (AL) is a semisupervised machine learning (ML) method, which makes use of a model to guide the selection of new samples to label unlabeled data of interest in an iterative process. In the context of computational drug discovery, this method has been used to identify potent inhibitors in small-molecule libraries at a fraction of the cost associated with a systematic potency screen.[1−3]

The identification of drug candidates requires a balance of novelty from exploring a new chemical space with optimization of known leads by means of small substitutions. The tension between exploration and exploitation is reflected in AL campaigns,[2] and combinations of the two strategies are common, although the exact procedures vary widely.[4,5] An exploration strategy aims to select samples that are representative of the underlying chemical space in order to construct a good potency model.[6] Exploitative strategy, on the other hand, aims to retrieve a high amount of potent compounds by means of a greedy acquisition based on the predicted binding affinity.

Traditionally, AL has been considered in the late stages of lead optimization to select compounds for synthesis. Retrospective studies on affinity data were able to retrieve top binders by using information from a small subset of the data.[4,5] The procedure was also combined with an automated synthesis setup to select products from a matrix resulting from two types of reagents.[7] Despite being successful, the throughput in these approaches was low (tens of compounds selected for labeling), and the small sizes of the libraries employed (hundreds of compounds) are often associated with a restricted chemical space.

Computational potency prediction methods have evolved over the last four decades from traditional docking,[8,9] alchemical free energy (AFE) techniques[10,11] to more recently ML approaches.[12−14] However, the use of AL applications together with computational potency estimation such as virtual screening[15−18] or relative binding free energy (RBFE) calculations using molecular dynamics simulations[19−23] only emerged in the past 8 years, driven by the increase in automation and throughput of computational tools for drug discovery. In these cases, 100s to 1000 compounds are selected out of pools containing up to 100,000 samples. The sheer size of the compound pool goes hand in hand with a high degree of diversity compared to low-throughput use cases, putting more

Received: February 7, 2024
Accepted: February 23, 2024
Published: March 6, 2024

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

strain on the AL pipeline and necessitating a careful selection of molecular features, ML models, and acquisition methods.[15,19,22,23] In addition to the challenge posed by data set sizes and diversity, using RBFEs or docking scores in lieu of experimental binding affinities introduces errors of systematic and stochastic nature, which are often not well characterized in advance.

Although AL presents an opportunity to quickly identify an active chemical space in large ligand pools, a routine application of this method in the pharmaceutical industry requires establishing a robust protocol that is transferable between different data sets. Previous AL studies used RBFE as their labeling tool of choice and only investigated ligands for a single target.[19,21−23] The scarcity of large public RBFE data sets and the cost and difficulty of generating them are additional hurdles for establishing robust AL−RBFE benchmarks. Furthermore, none of the studies mentioned above considered cost as a factor in the selection of a protocol, resulting in very large initial batches or exploration phases.[19,22] They also compared protocols that require a variable amount of RBFE data.[21] The difference between data sets, their sizes and generation procedures, as well as applied AL protocols, and different combinations of metrics to evaluate the performance of AL make it difficult to compare literature protocols and identify best-practice approaches.

The aim of this study is to evaluate AL protocols in a rigorous manner by using four publicly available data sets for benchmarking. The main focus is on the AL design and not on the method used for labeling data. Different strategies for labeling are possible, such as docking,[9] AFE methods (relative or absolute methods),[10] experimental measurements, or an ML property prediction model.[14] How to choose best labeling strategies and how to mix different ones will not be evaluated here. Instead, we use four different data sets, where we already have labels provided. The chosen data sets differ in their protein targets, kind of potency measurement (ΔG from RBFE or experimental K_i/IC_{50}), size (600 to 10000 samples), and degree of diversity. The Tyk2 data set has labels from predicted RBFE values (ΔG), which are converted to binding affinities (K_i), while all other data sets comprise experimental values. Our primary objective is to investigate how the diversity and size of data sets affect the efficacy of AL. We use a wide range of metrics to gain a holistic perspective, ranging from conventional regression metrics, such as R2, to assess the overall performance of the ML model, to Recall and F1 scores for the top 2%/5% binders to assess the exploitative capabilities of a model and the degree of exhaustion of the active chemical space. To investigate the benefit of pretrained model architectures for AL, we compare a fine-tuned Chemprop (CP) model with Gaussian process (GP) regression, which is a common choice for AL.[21,22] Finally, the total number of acquired samples is always kept constant throughout all experiments to compare AL protocols at a fixed cost. With our prelabeled data, we do not have a choice how to label the data, but any labeling method can be used, such as a docking score, an experimental value, or an RBFE calculation. However, mixing different methods will require care in accounting for accuracy or trustworthiness of the labeling method used and is beyond the scope of this work.

The paper is structured as follows. In the Methods section, we give a detailed overview of the data sets and provide details of the AL procedure, ML models, and metrics. In the Results and Discussion section, the GP and CP models are first benchmarked to assess differences in their predictive power between data sets. Next, we compare AL procedures by varying the size of the initial batch and the method for its acquisition. Thereafter, we identify an optimal batch size for cycles that come after the initial batch(es). Finally, we assess the robustness of AL toward stochastic noise in the potency data.

METHODS

Data Sets. We used four publicly available binding affinity data sets that encompass the following protein targets: Tyrosine Kinase 2 (TYK2); a G protein-coupled receptor target, Dopamine Receptor D2 (D2R); and two proteases, Ubiquitin-Specific Protease 7 (USP7) and SARS-CoV-2 Main Protease (Mpro). Table 1 provides the number of ligands in each data set as well as other information relevant for subsequent AL experiments. Figure 1A shows the distribution of the measured affinities associated with each data set. All the data sets used in our study are accessible at https://github.com/meyresearch/ActiveLearning_BindingAffinity.

Table 1. Summary of Data Set Characteristics for Protein Targets Used in Our Study[a]

target	ligands	binding measure	% data for AL	top 5%	top 2%
TYK2	9997	pK_i	3.6	500	200
USP7	4535	pIC_{50}	7.9	227	90
D2R	2502	pK_i	14.4	125	50
Mpro	665	pIC_{50}	54.1	33	13

[a]Total number of ligands, the binding measure used for training and inference, the percentage of data utilized for AL based on a consistent sample of 360 compounds acquired over AL cycles, and the count of compounds in the top 5% and top 2% fraction of the dataset.

The TYK2 data set was derived from the work of Thompson et al.,[21] which focused on optimizing the AL methodologies for RBFE calculations. This data set comprises 10000 congeneric molecules targeting the TYK2 kinase, all of which were synthesized using an aminopyrimidine core scaffold. The data set was initially populated with 573 TYK2 inhibitors, which were subsequently decomposed into unique R-groups at three attachment points. These groups were then combinatorially assembled to create an initial library of 203406 unique compounds, which was then filtered down to 10000 molecules based on a set of "drug-like" properties. $\Delta\Delta G$ values obtained from RBFE calculations were converted to pK_i values, providing a more interpretable metric for binding affinity, using

$$pK_i = \frac{-(\Delta\Delta G + \Delta G_{ref})}{RT\ln 10} \quad (1)$$

where pK_i is the negative logarithm of the inhibition constant, which is a measure of the binding affinity of a ligand for its target (TYK2). $\Delta\Delta G$ is the free energy difference of binding between a reference compound and a second compound. ΔG_{ref} is the absolute binding free energy of the reference compound (−47.778 kJ/mol for TYK2), R is the universal gas constant, approximately equal to 8.314 J/(mol·K), and T is the absolute temperature. In Figure 1B, the Uniform Manifold Approximation and Projection (UMAP)[24] of the TYK2 data set shows clear clusters that capture variations in R-groups attached to the core scaffold. We can see that most of the active compounds are located within the two upper clusters. Figure S1 in the Supporting Information highlights that the majority of the top 2% binders are situated here.

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

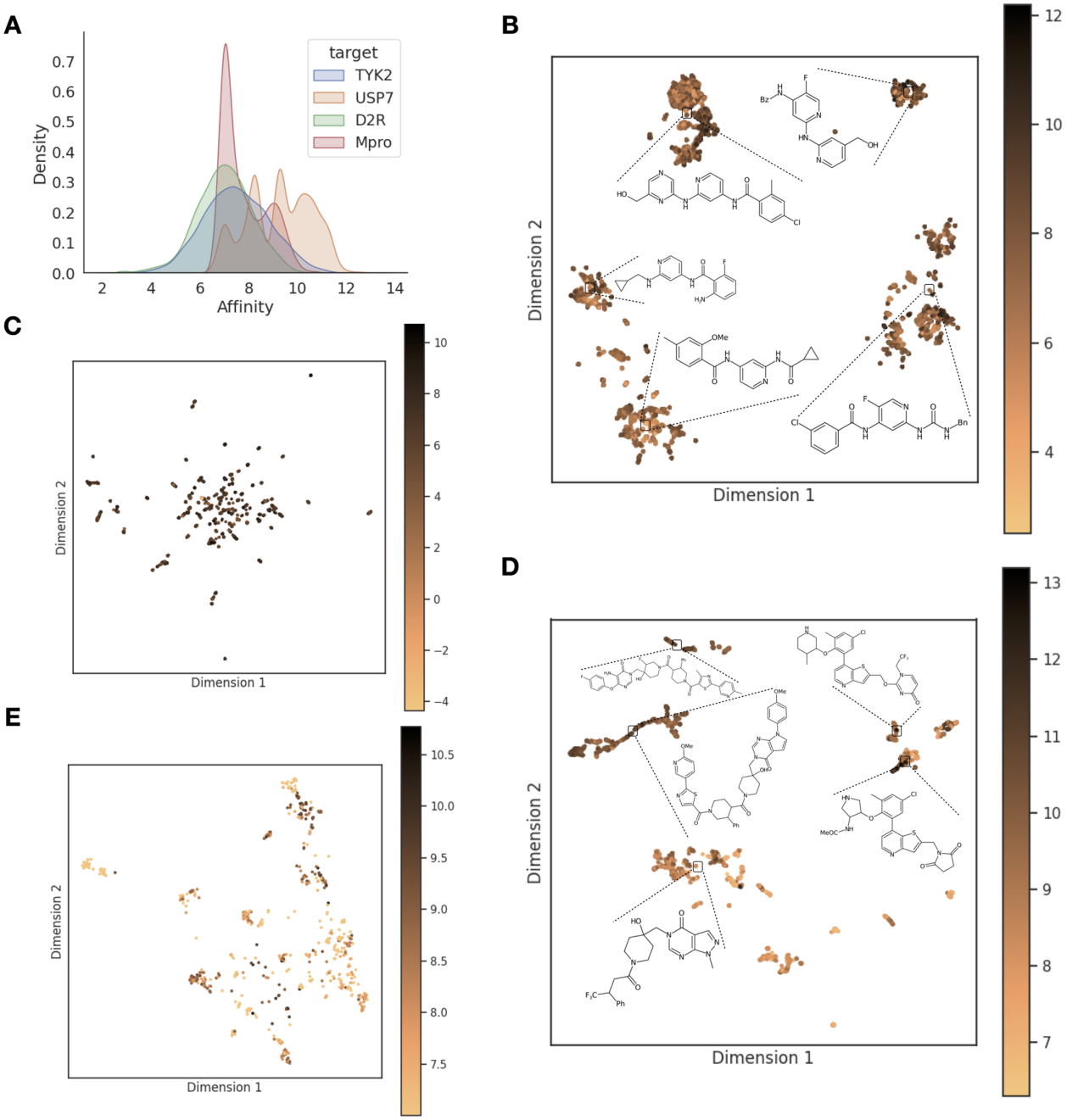


Figure 1. Distribution of affinity scores and UMAP projections for four protein targets. (A) Kernel density estimation plot illustrating the distribution of affinity scores for each target data set: pK_i values for TYK2 and D2R and pIC_{50} values for USP7 and Mpro. The standard deviations (1 σ) for the pKi/pIC_{50} values are as follows: 1.36 for TYK2, 1.31 for USP7, 1.44 for D2R, and 0.91 for Mpro. (B) UMAP projection of the TYK2 data set with overlaid cluster centroid compounds. (C) UMAP visualization of the D2R data set. (D) UMAP projection of the USP7 data set with overlaid cluster centroid compounds. (E) UMAP representation of the Mpro data set.

The USP7 data set was curated by Shen et al.,[25] with the primary objective of building a classification model to distinguish active from inactive inhibitors. The SMILES of over 4000 ligands together with their experimental affinities including K_i, K_d, and IC_{50} were collected from ChEMBL.[26] Duplicate SMILES were aggregated into unique entries using the Open Babel package 2.3.1[27] by Shen et al.[25] All experimental results with varying units were converted to IC_{50} values for each SMILES, which, for the scope of our study, were translated to pIC_{50} values. As evident from Figure 1A, the USP7 data set exhibits multiple assay minima. Moreover, the UMAP visualization with cluster centroids in Figure 1D highlights the data set's diverse core scaffolds and R-groups, showing the presence of heterogeneity. However, scaffolds tend to be well-preserved within a cluster. A significant portion

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

Journal of Chemical Information and Modeling pubs.acs.org/jcim Article

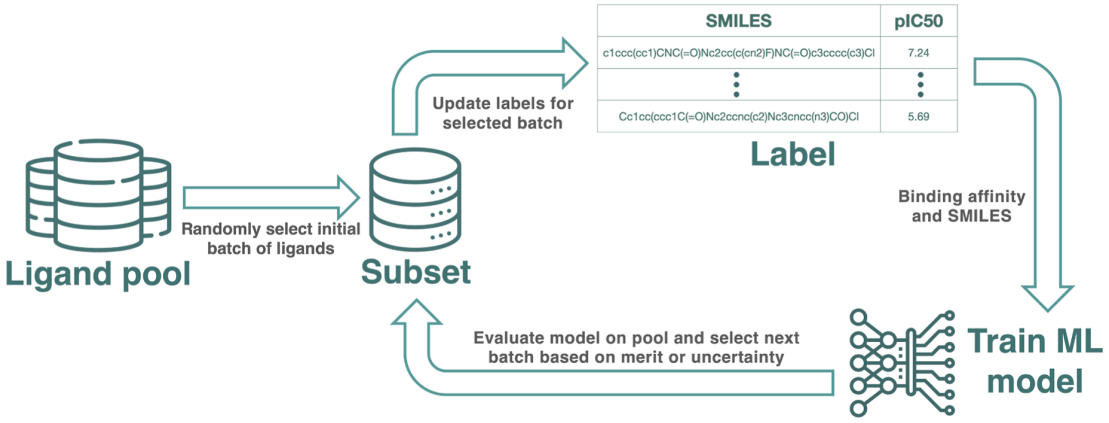


Figure 2. Schematic overview of the AL pipeline. An AL cycle begins with a randomly chosen batch from the available pool of compounds, followed by labeling and model training on the subset with labels. The subsequent batch for the next AL cycle is strategically chosen based on model predictions and uncertainties using exploration, exploitation, or random strategies.

of the top active compounds can be found in the upper regions of the UMAPs.

The D2R data set is a subset of the ACNet data set,[28] which was curated from the ChEMBL[26] database (version 28) on 190 targets to study the performance of ML models on data from activity cliffs. Zhang et al.[28] categorized matched molecular pairs as activity cliff if the difference in potency is $pK_i \geq 2$. We selected the D2R target due to the high number of associated activity data, making it particularly suitable for our study. The ACNet data set was constructed by screening over 17 million activities, only retaining compounds tested against single human targets in direct interaction binding assays and filtering data with low assay confidence. The data set uses assay-independent equilibrium constants (pK_i) as the measure of potency. We retained the pK_i values and averaged over duplicate entries with the same SMILES to ensure data consistency, leading to a reduction from 4121 to 2502 entries. From the UMAP in Figure 1D, the D2R data set appears to encompass a heterogeneous assortment of compounds, with a large number of congeneric series of 10−20 compounds each. The absence of distinct clusters and the dispersed distribution of binding scores suggest intricate structure−activity relationships.

The Mpro data set is part of the COVID Moonshot project,[29] which focuses on the development of inhibitors for the SARS-CoV-2 main protease. The data set provides experimental pIC_{50} values, which are an amalgamation derived from single enantiomers and racemic mixtures. The project consists of several design cycles (sprints) in which medicinal chemists select compounds for synthesis out of a database with public submissions. The UMAP depicted in Figure 1E shows a diverse composition of the Mpro data set. The absence of distinct clusters and dispersed distribution of top active compounds highlight its intricate nature. Importantly, with only 665 compounds, this data set is significantly smaller than the other data sets in our study, offering a distinct context for AL pipeline investigations in low-data settings.

AL Protocols. AL is a ML paradigm designed to optimize the selection of samples for training models (Figure 2). It is particularly useful for problems where labels can be calculated for every data point, but the associated computational cost is high, as is the case for RBFE calculations or testing the data point in the lab. The key difference from conventional ML consists of splitting the training process into several AL cycles such that a model trained on a subset of samples informs the selection of the next batch to be added to its training set.

The initial batch of compounds was selected at random, as this is a common strategy in previously published studies.[15,21] This choice also ensures that the distribution of the training data matches the pool used for inference, which is not the case in diversity-based selection methods.

In a real AL use case, labels for the selected compounds can be acquired by means of RBFE calculations. We take the advantage of experimental potency values from the literature instead of RBFE calculations. This provides robust insight into AL on retrospective data.

A model is then trained on a subset of labeled samples and used to make predictions on the unlabeled data. On the basis of these predictions, a strategic subset of samples is selected for labeling, often employing strategies aimed at either "exploration" of the chemical space or "exploitation" of promising regions within it. These newly labeled samples are then incorporated into the training set for the next AL cycle.

In each AL cycle, we employ one of the strategies for sample selection: random sampling, exploration, and exploitation. Random selection involves choosing compounds arbitrarily from the remaining data. Exploitation focuses on selecting compounds with the highest predicted potency, thereby exhausting high-potency areas in the chemical space. Exploration selects compounds with the highest prediction uncertainty, aiming to sample broadly across the chemical space to gain a more global understanding, thereby potentially identifying new promising areas in the chemical space.

To ensure a fair evaluation for different targets, we always acquire a total of 360 compounds for labeling over the whole AL procedure, irrespective of the data set, model, or selection protocol. The number of AL cycles is always adjusted to fit this total depending on the batch sizes. Additionally, to account for variability, each experiment was conducted three times by using different seeds for the initial batch selection.

ML Models. *GP Regression.* We chose GP regression[30] for its ability to provide both expected values and uncertainty estimates for predictions, as well as its proven performance in a previous AL study on the TYK2 target by Thompson et al.[21] GP is a nonparametric Bayesian approach that provides a probabilistic framework for making predictions. GPs make use

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

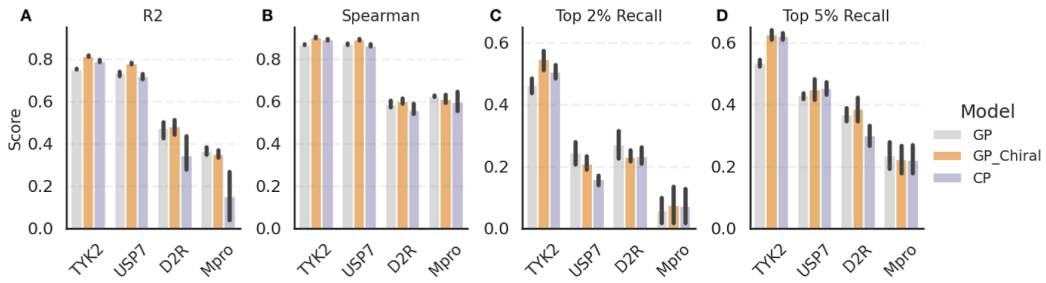


Figure 3. Benchmarking GP and CP models on four target data sets (TYK2, USP7, D2R, Mpro) used in our AL study. We use 20% of the data set for training the models and the remaining 80% as a test set to calculate the metrics. (A) R2, (B) Spearman ρ, (C) Recall for top 2% compounds, and (D) Recall for top 5% compounds. Data sets are sorted based on their size, i.e., from large to small (see also Table 1). While both models exhibit robust predictive power, the CP model is more sensitive to the data set size than the GP model, and the introduction of chirality descriptors offers limited enhancement in model accuracy.

of a kernel function to measure the similarity between the data points. The kernel function is used to construct a covariance matrix of observed features, which is used to make predictions for the unseen data. In our implementation, we use the Tanimoto similarity kernel, which is particularly useful for measuring the similarity between sets, making it suitable for binary fingerprints. Our implementation of GP is based on the GPyTorch version 1.10[31] library.

By default, ligands were featurized using ECFP8 fingerprints from OpenEye's OEChem toolkit version 3.4.0.0.[32] To assess the influence of chiral descriptors on the GP model performance, an alternative featurization using Morgan fingerprints with chirality descriptors was also assessed as a part of benchmarking. Morgan fingerprints were generated using RDkit[33] and have a radius of 4. Both ECFP8 and Morgan fingerprints were hashed to 4096 bits.

Chemprop. CP is a message-passing neural network designed for molecular property prediction. The model operates by iteratively updating the atom and bond features of a molecule through message-passing layers (and hence does not rely on molecular fingerprints, such as GP). This allows the model to capture both local and global structural information. CP has been shown to perform well on a variety of molecular property prediction tasks, including solubility, toxicity, and binding affinity.[34] Monte Carlo dropout[35] was used to provide a measure of uncertainty. Our implementation is based on the Chemprop package version 1.6.1,[36] and we use a model pretrained on potency data across 1788 targets, including Kinases, GPCRs, and Proteases. The data for pretraining the model are mostly taken from ChEMBL[26] for targets having more than 200 interaction data points. For our work, we unfreeze the entire encoder and train the model in each AL cycle for 500 epochs with a batch size of 50 and a learning rate of 0.0001 for 10 warmup epochs, ramping up to a maximum of 0.001 for the remaining epochs using a Noam scheduler. These hyperparameters for fine-tuning the CP model are empirically determined.

Analysis. We used a range of metrics to evaluate the performance of our AL benchmark. These metrics are selected to assess both the exploitative and predictive aspects of the model. Whereas exploitation is governed by the predictive prowess of the model on the high end of the potency range, regression metrics reflect its performance on the bulk of the data.

To define Recall and F1 scores in this context, the selection process is converted to a classification task. A compound is considered "True" if it has been acquired for labeling in any of the previous AL cycles and "False" otherwise. We further categorize compounds into "active" and "inactive" based on their relative ordering, specifically focusing on the top 2% and top 5% of compounds. The Recall metric is calculated according to eq 2

$$\text{Recall} = \frac{\text{TP}}{r \cdot N_{\text{tot}}} \tag{2}$$

Here, TP represents the number of true positives, r is the fraction of compounds considered true (either 0.02 for the top 2% or 0.05 for the top 5%), and N_{tot} is the total number of samples in the given data set. The expectation value of the TP for a random selection is given by $r\,N_{\text{acq}}$, where N_{acq} represents the number of compounds acquired or selected in the current AL cycle. The F1 score is particularly useful for assessing the model's ability to correctly identify the most promising compounds (TP) while minimizing the selection of false positives (FP). We compute the F1 score according to eq 3

$$F_1 = \frac{2 \cdot \text{TP}}{N_{\text{acq}} + r \cdot N_{\text{tot}}} \tag{3}$$

Here, we used the relations N_{acq} = TP + FP and $r\,N_{\text{tot}}$ = TP + FN. False negatives (FN) represent the promising compounds that the model failed to identify.

To assess the predictive power of the models, we use coefficient of determination R2, Spearman ρ, and the root-mean-square error (RMSE). To visualize the high-dimensional fingerprints on two-dimensional maps, we use UMAP[24] as a dimensionality reduction technique using the `umap-learn` package version 0.5.3.[37] In contrast to widely used tSNE plots, UMAPs do not rely on a fixed cutoff and instead keep the number of neighbors constant (here, 50), which is an arguably better choice for preserving the global structure of the data.

RESULTS AND DISCUSSION

Model Benchmarking. Figure 3 compares GP and CP models trained on a 5-fold split, where the training set was made up of 20% of the data and metrics were calculated on the test sets containing the remaining 80%. Error bars represent a 95% confidence interval across these folds. The calculated metrics are the coefficient of determination R2 (A), Spearman ρ (B), and the Recall of top 2% (C) and 5% (D) samples across the test sets. To evaluate the impact of chiral molecules, we considered a GP model trained on Morgan fingerprints

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

Journal of Chemical Information and Modeling pubs.acs.org/jcim Article

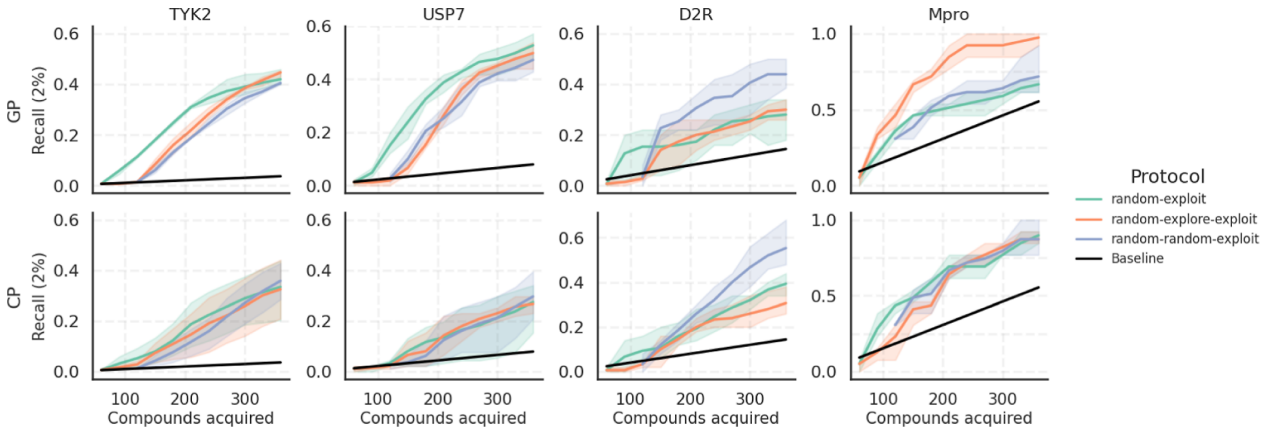


Figure 4. Top 2% Recall achieved with different AL protocols for initial sample selection across the four target data sets. The "random−exploit" protocol acquires 60 compounds at random before switching to exploitation, "random−explore−exploit" acquires another 60 compounds using the prediction uncertainty following the initial batch selected at random, and "random−random−exploit" starts with 120 compounds selected at random. The shaded area displays the variance over three repeats initialized with a different random seed for the initial sample selection. The baseline shows the expectation value for the Recall upon random acquisition. The protocol yielding the best Recall is consistent between the two models for each data set but not between different data sets.

with chirality descriptors in addition to achiral ECFP8 fingerprints with the same size and radius.

The goal of this work is to assess the performance of the two models on each data set in the limit of a large training set. However, the training sets amount to a fraction of the pool rather than being comparable in size to reflect the total number of compounds typically acquired in AL. Table 1 shows that the subsequent AL experiments acquire less than 20% of the data in total except for Mpro, where over half of the data are acquired. Given the lack of bias in a random sample selection, this analysis provides an upper bound for the model performance in terms of regression. It also gives an estimate for the Recall given a large amount of training data.

For each data set, all of the models demonstrated predictive capabilities, with an R2 larger than 0.3 (with the exception of CP trained on Mpro data), Spearman ρ over 0.5, and top 5% Recalls of 0.2 or more. This suggests that it is possible to train a predictive model even on the more challenging data sets given enough training data sampled broadly over the available chemical space. There is, however, a clear trend in model performance concerning the data set size. R2 and top 5% Recall monotonously decrease with decreasing size of the data set. The trend is also present, to a weaker extent, for Spearman ρ and the top 2% Recall, which is more pronounced for CP compared to that for GP. This observation aligns with the understanding that CP, being a deep learning model, benefits from larger data sets.

Both Mpro and D2R data sets presented challenges due to their heterogeneous nature. The lack of distinct clustering in the UMAP projections (Figure 1C,E) for these data sets, especially for D2R, indicates the diverse composition of compounds, making them potentially harder to fit with ML models. Spearman ρ for these two data sets is indeed lower than those for TYK2 and USP7. However, the comparably large training set accounts for diversity to some degree, which is why differences between different types of data sets are not very pronounced in this benchmark.

The GP model using chirality descriptors showed comparable performance to the model trained on achiral fingerprints, suggesting that introducing chirality representation did not significantly enhance the models' performance. For Mpro and D2R compounds, which have a significant number of chiral centers, the performance remained comparable between both the fingerprints. Similarly, the USP7 data set, which contains only a few chiral compounds, showed no significant improvement with the chirality descriptors. It is likely that chirality does not play an important role in the structure−activity relationship of these series or that stereochemistry information is missing for some of the data (such as TYK2).

AL Strategies. In this section, we investigate the impact of batch sizes, sample acquisition strategies, and the exploration−exploitation trade-off in simulated AL scenarios. The focus herein lies on protocols with a distinct exploration phase, which aims to select diverse samples for building a robust and predictive model, followed by greedy acquisition using the merit predicted by the model (exploitation). Splitting the AL protocol into these two stages simplifies the evaluation of the individual benefits of each phase. Note, however, that the interleaving of the two strategies has also been studied in literature studies[5,20,23] yielding good results. We always acquire a total of 360 compounds for labeling to fairly evaluate selection protocols that differ in their batch sizes.

Selection of Initial Samples. In an AL setting, it is essential to understand the influence of the initial sample selection strategy as it guides the trajectory of subsequent AL cycles. We explore initial batch sizes and strategies for the selection of these compounds using three distinct selection protocols. All of them are initiated with 60 samples selected at random (following the suggestion from Thompson et al.[21] in the TYK2 study) and use a batch size of 30 compounds for exploitation. The "random−exploit" protocol immediately switches to exploitation from the first cycle on. Both the "random−explore−exploit" and "random−random−exploit" protocols acquire 60 more compounds after the initial selection with the purpose of improving coverage of the chemical space, resulting in an effective initial batch size of 120. More precisely, the "random−random−exploit" protocol selects the additional 60 compounds at random, whereas the "random−explore−exploit" protocol utilizes the model's prediction uncertainty to

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

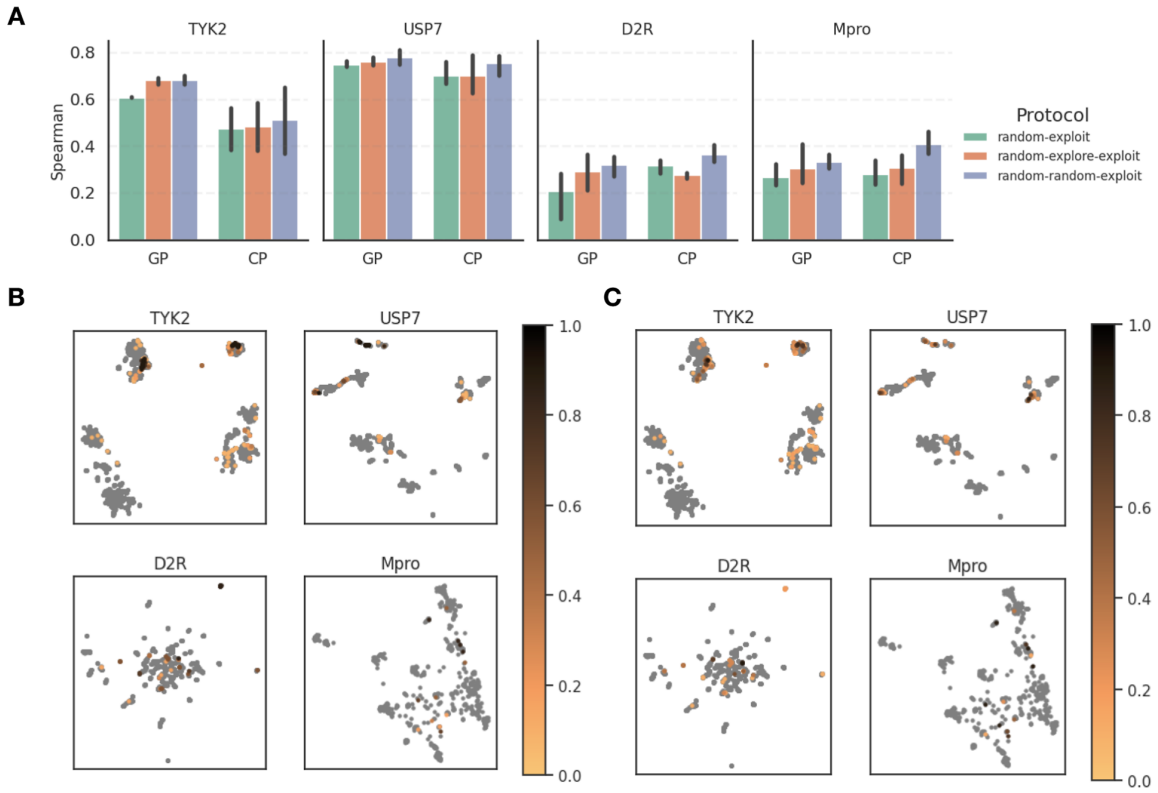


Figure 5. Evaluation of model performance with Spearman ρ and UMAPs showing compound acquisition across AL protocols. (A) Spearman ρ of the final models trained on all 360 selected compounds, emphasizing the enhanced predictability with larger initial batch sizes of 120 in the "random−explore−exploit" and "random−random−exploit" protocols. (B) UMAP projection of the GP model selection displaying the top 2% compounds in each data set, colored by the frequency of their acquisition. UMAP shows consistent acquisition of compounds in dense clusters and the challenge of identifying sparse clusters. (C) UMAP of the CP model, highlighting similar trends in compound acquisition. Overall, we can see that the GP model is more consistent in compound selection than the CP model across multiple AL runs.

select the additional compounds over two exploration cycles. Due to the constraint on the total number of acquired compounds, the "random−explore−exploit" and "random−random-exploit" protocols acquire fewer compounds over the exploitation phase than "random−exploit".

Figure 4 shows the top 2% Recall (eq 2) across the four data sets for the three protocols and both GP and CP models, and the top 5% Recall is shown in Figure S2. Figure 5A shows the corresponding Spearman ρ values of the final models trained on all 360 selected compounds. A more detailed plot showing the Spearman ρ as a function of the number of compounds acquired can be found in Figure S3, and the equivalent plots for the R2 and the RMSE are also shown in Figures S4 and S5, respectively. A general observation is that a larger initial batch size of 120 with the "random−explore−exploit" and "random−random−exploit" protocols yields more predictive models and therefore augments the likelihood of pinpointing active compounds, as evidenced by the higher Spearman ρ (Figure 5A) and larger slopes in the Recall curves (Figures 4 and S2). The same trend is also seen for R2 and RMSE in Figures S4 and S5, respectively. However, the increase in the initial batch size comes at the expense of an increased initial training cost for the model. The F1 score (Figures S6 and S7) decreases in later cycles for all data sets except TYK2 in response to diminishing precision, which is indicative of an increased cost per identified top compound. This decline is more pronounced for smaller data sets.

For the TYK2 and USP7 data sets, a smaller initial batch size was found to be more adequate with GP, given that the explorative protocols only caught up after around 300 acquired compounds. In contrast, the more heterogeneous D2R and Mpro data sets greatly benefited from a larger initial batch. As can be seen from Figure S1, D2R predominantly contains top compounds in small dense clusters, while Mpro's top compounds are more dispersed as singletons. This distinction can explain the opposite trends in terms of selection protocols for these two data sets. However, such nuances are not known a priori and cannot be used to inform the choice of the exploration method. The drop of predictive power (Figures S2 and S4) on Mpro over the course of AL is associated with the small size of this data set compared to the number of acquired training samples as well as the exploitative acquisition, which causes a potency imbalance between training and pool data. In comparison to the GP model, the CP model underperformed on all data sets except D2R, although its predictive performance tends to be more consistent between the three protocols. By contrast, CP outperformed GP on D2R, where the underlying global model is likely to account well for the diversity of these compounds.

Figure 5B,C shows UMAPs where the top 2% compounds in each data set are colored by the frequency of their acquisition (averaged over different protocols and random seeds for initialization) using GP or CP, respectively. A breakdown by protocol for each data set can also be found in Figures S8−S11,

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

Journal of Chemical Information and Modeling pubs.acs.org/jcim Article

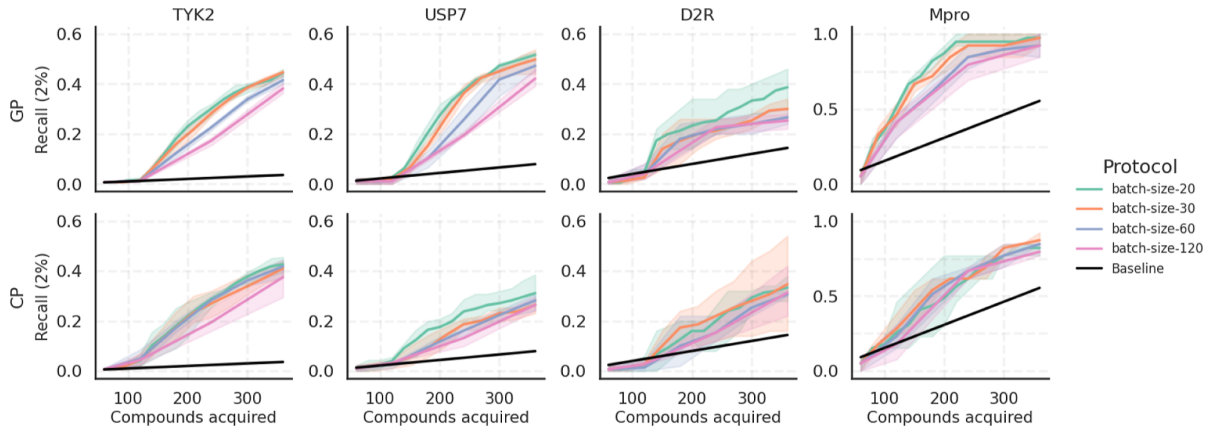


Figure 6. Comparison of the top 2% Recall across AL protocols varying in their batch size. The "batch-size-20" protocol uses three exploration batches and 12 exploitation batches of 20; "batch-size-30" employs two exploration and eight exploitation batches of 30; "batch-size-60" employs a single exploration and four exploitation batches of 60; and "batch-size-120" employs one exploration batch of 60 and two exploitation batches of 120. The comparison suggests that the Recall improves with decreasing batch size across all data sets with both models.

which shows that there is little difference between compounds identified by different protocols. A common trend in Figure 5B,C is a consistent acquisition of compounds in dense clusters, while sparse clusters remain difficult to track with both models. Overall, we could see that the selection with the GP model is slightly more consistent across multiple AL runs.

We also evaluated the performance of AL relative to training models on a larger proportion of the data, as was the case in the Model Benchmarking section. The recall achieved by AL using a smaller fraction of the data (Figure 4) is comparable to or better than the models trained in a single batch on 20% of each data set (Figure 3), demonstrating the benefits of partitioning the training process into multiple stages. On the other hand, R2 and Spearman ρ are lower for models constructed using AL (except for the Mpro data set due to its size). It might be tempting to conclude that the size of the training set is the deciding factor for a model's predictive quality, but this is not entirely true. Figures S3 and S4 show that the R2 and Spearman ρ stop improving after the start of the exploitation phase, and the effective performance of the final models is comparable to that of the models constructed only from the initial 60 or 120 samples. Similarly, the models constructed from a large randomly chosen batch better account for the variety in the data than samples selected using an exploitative strategy. As stated before, the amount of diversity in the data is strongly linked to the speed of convergence of a model with an increasing number of training samples. On the TYK2 and USP7 data sets, GP models trained on 360 samples reach more than 70% of the rank-ordering performance than that of corresponding models trained on more than twice as many samples (CP underperforms on TYK2, reaching only about 50% of the performance of the larger model). By contrast, this ratio drops to 30−50% for D2R, which also benefited greatly from an increase in the initial batch size. In summary, AL yields greater benefits for compound pools, which display a high degree of similarity and strong structure−activity relationships.

Batch Size for Exploitation. To systematically investigate the influence of the batch size in exploitation cycles, we fixed the initial sample selection at 60 samples chosen randomly and the subsequent 60 samples selected based on the exploration strategy. We then evaluated four distinct batch sizes for the subsequent AL cycles, namely, 20, 30, 60, and 120. Given our constraint of acquiring a total of 360 compounds, smaller batch sizes necessitate a greater number of AL cycles to reach this total. The protocols varied in batch sizes, with "batch-size-20" using three exploration batches of 20 each followed by 12 exploitation batches of 20, "batch-size-30" using two exploration and eight exploitation batches of 30, "batch-size-60" using one exploration and four exploitation batches of 60, and "batch-size-120" combining one exploration batch of 60 with two exploitation batches of 120.

Figure 6 shows the top 2% Recall across different batch sizes. We can see a clear trend where smaller batch sizes consistently outperform larger ones across all data sets, and this holds true for both GP and CP models. The same trend can be seen in Figure S12, which shows the top 5% Recall, and also considering F1 scores (Figures S13 and S14), indicating that smaller batch sizes favor both precision and recall. The same trends are reflected by the metrics Spearman ρ, R2, and RMSE across all data sets, except for Mpro (as seen in Figures S15−S17). This trend can be understood by considering the ratio of pool and training sizes. When the pool of available data significantly outnumbers the training set, the model benefits from small, incremental additions to its training data, making a batch size of 1 theoretically optimal. This is because each new data point refines the models understanding. However, in practical scenarios, using extremely small batch sizes might not be feasible as it would require acquiring potency data in a time-consuming serial manner rather than a more efficient parallel approach. This advantage of smaller batches diminishes when the pool size is roughly equal to the training size, where the predictive power of a model decreases upon imbalancing the data by using an exploitative acquisition strategy.

Modeling Noise on Labels. The measurement of binding affinities by experimental and computational means is subject to noise although being fed to the model as "true" values. To systematically study the influence of noise in training data, we introduced Gaussian noise to each data set. The noise was generated by sampling random numbers from a Gaussian distribution with a mean of zero and a standard deviation (σ) ranging from zero to two times the standard deviation of the underlying potency data, meaning that the amounts of noise vary between data sets in absolute terms. Our investigation

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

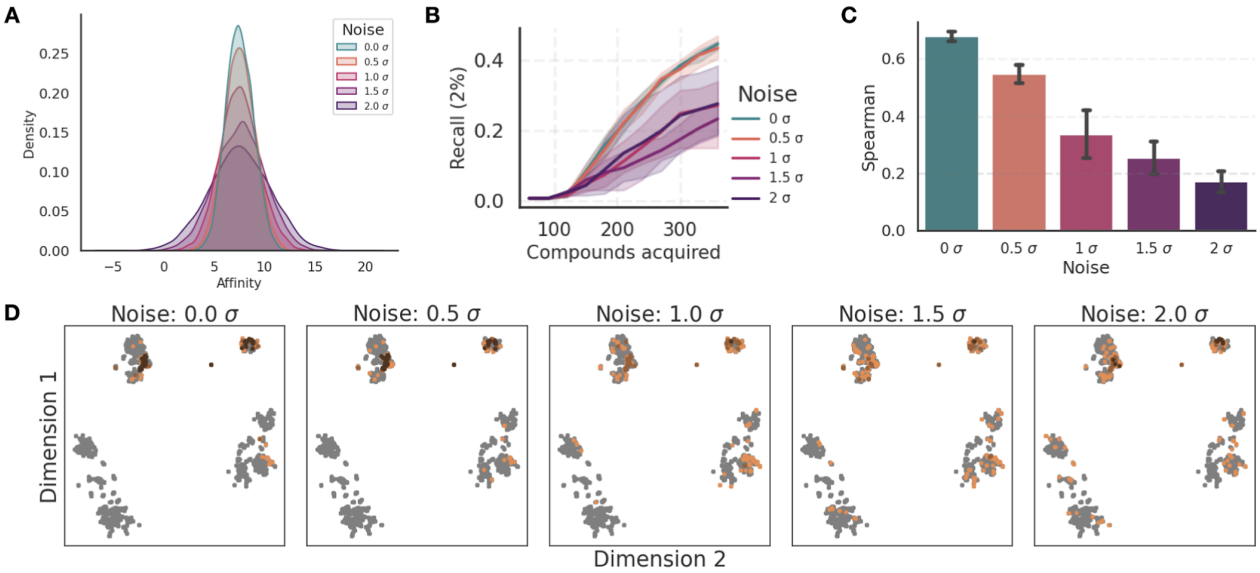


Figure 7. Analysis of the influence of Gaussian noise on the outcomes of AL using the GP model on the TYK2 data set. The standard deviation of the added Gaussian noise was scaled with respect to the standard deviation of TYK2 pK_i values, with factors ranging from 0 (no noise) to 2. (A) Kernel density estimation plot of the pK_i distribution across varying noise magnitudes. (B) Top 2% Recall, highlighting a noticeable decline at increased noise levels. (C) Spearman ρ revealing diminished model predictability with increasing noise. (D) UMAP visualization of the compounds selected in the exploitation phase, colored by the acquisition frequency across three distinct AL iterations with randomized initializations. The UMAPs emphasize the AL framework's capability to consistently identify top-binding compound clusters, even amidst noise interference.

considers the noise multipliers 0, 0.5, 1, 1.5, and 2. Similar to the batch size modulation, we fixed the initial sample selection at 60 samples chosen randomly and the subsequent 60 samples selected by using the exploration strategy. We ran the AL pipeline three times with different random seeds for selection of the initial batch. The transformation of the potency distribution at each noise level for the TYK2 data set is visually presented in Figure 7A. Similar transformations for other data sets, namely, USP7, D2R, and Mpro, are shown in Figures S18A, S19A, and S20A, respectively, and the UMAPs of the noisy data are shown in Figure S21. An overarching observation from Figure S21 is that the introduction of stochastic noise retains the macroscopic structure of the clusters while smoothing out the microscopic features within these clusters across all target data sets.

The introduction of noise has a pronounced effect on the regression performance and Recall of the constructed model. As depicted in Figure 7B, the top 2% Recall rapidly decays when the noise level exceeds 1 σ. This trend of declining Recall with increasing noise is consistently observed across other data sets, as shown in Figure S18B for USP7, Figure S19B for D2R, and Figure S20B for Mpro.

Furthermore, Spearman ρ drops with increasing noise levels, as shown in Figure 7C. However, the models remain predictive even with high amounts of noise, maintaining a positive Spearman ρ coefficient. This trend also holds true for USP7 and D2R (Figures S18C and S19C) but not for Mpro, where Spearman ρ drops below zero for a noise addition of 2 σ, as shown in Figure S20C.

AL is relatively consistent in identifying the top binders between independent repeats even under noisy conditions. Figure 7D shows compounds acquired in the exploitation phase colored by the fraction of three AL repeats that selected them. Similar observations can be made for other data sets, as shown in Figure S18D for USP7, Figure S19D for D2R, and Figure S20D for Mpro. These findings are consistent with the work of Bellamy et al.[38] for synthetic affinity data, who found a better Recall in terms of the true (noiseless) labels than considering the noisy values used to train the model. This robustness underscores the fact that while noise may introduce perturbations, learning the overarching structure−activity relationship trends remains preserved. Interestingly, the presence of noise might even be beneficial for exploration, aiding in overcoming potency barriers in the chemical space. However, while noise can enhance exploration, it hampers the exploitative power of the model.

A similar trend of decaying Recall with increasing noise also is observed for the CP model, as depicted in Figure S22. Spearman ρ for the CP model also exhibits a decline with increasing noise levels (Figure S23). However, the CP model is far less consistent in identifying relevant clusters compared to the GP model in the presence of noise (Figure S24), especially for TYK2 and USP7.

In conclusion, while the presence of stochastic noise impairs the performance of AL, the models demonstrate a commendable ability to identify large-scale trends. However, the Gaussian noise simulation in this study primarily accounts for stochastic noise. Systematic errors are not accounted for. They can particularly be detrimental, as they introduce a bias in the activity landscape, potentially misguiding the AL process. One approach to investigating the effect of systematic bias in AL could involve SMARTS filtering to isolate compounds with specific functional groups and then introduce noise with a nonzero mean.

CONCLUSIONS AND OUTLOOK

We comprehensively evaluated the effect of different parameters for AL protocols based on a range of metrics. These capture the identification of top binders, the predictive quality of the underlying ML model, and a qualitative analysis

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

of the identified clusters in the chemical space. Simulating identical AL runs on different binding affinity data sets allowed us to assess different aspects of the AL strategies, as well as to capture trends with respect to the size and composition of a data set.

Using RBFE to label a chemical library of 5000 compounds, each calculation averaging 8 GPU hours at a rate of $2.50 per GPU hour, results in a total cost of $100 K. Using AL and RBFE to label just 300 compounds incurs a cost of $6k and can yield a comparably predictive model. AL can identify top binders using a significantly smaller fraction of the data to train. We observed a pronounced dependence on the data set size, although the diversity of compounds in a data set is the most decisive factor contributing to the margin of profit achievable by AL. This may be reduced even further with reliable docking or ML labeling protocols in the future. A similar trend was identified when varying the initial batch size for AL, where the more diverse D2R and Mpro data sets benefited from a large initial exploration phase compared to TYK2 and USP7.

The results suggest that certain design strategies will help in designing the most useful AL protocols. These findings have implications for the design of ligand pools for AL. The TYK2 and USP7 data sets, which consistently led to predictive models and a high Recall, are made up of few distinct scaffolds and a large number of substitutions or even a combinatorial enumeration of substituents in the case of TYK2. The confinement of chemical space is essential for a small number of samples to represent a sizable pool. In practice, however, a combinatorial exploration of substituents is not always desirable when a strict filtering of compounds is necessary due to constraints on the physicochemical properties. In such cases, increasing the batch size is a more defensive approach to achieve success with AL. In the present study, using the model uncertainty in the initial exploration phase did not yield any observable benefits over selecting the compounds at random.

For the exploitation phase, our findings consistently suggest that training in small batches results in the highest Recall. The performance gains, however, are incremental when the batch size is reduced below 30 samples. Very small batches are also undesirable from a practical perspective due to the increase in the number of AL cycles and the overall turnaround time.

The addition of noise to potency data was found to have detrimental effects on the exploitative power of AL if the variance of the noise is equal to or larger than the variance of the affinity values. On the other hand, the noise did not prevent the GP model from finding large-scale active regions in the chemical space, even when its variance exceeded the underlying signal. The CP model was more affected by noise in the data and lost its predictive power with high levels of noise. Together with the fact that CP outperformed GP on the AL runs for D2R, whereas GP performed better in the general case, it is likely to be more sensitive to the local structure of chemical space and, at the same time, is more vulnerable to noise.

Using a Gaussian model noise is a drastic simplification to account for the entirety of errors that may be introduced by the labeling methods, such as RBFE, and understanding their nature for a given ligand series is crucial. A validation prior to running AL with the chosen labeling method (e.g., AL−RBFE) is highly recommended, as it may not only provide insight into the magnitude of stochastic errors relative to the dynamic range of RBFE values but also reveal systematic offsets for certain functional groups, which can alter the entire course of an AL run. One aspect to look at in the future is data initialization beyond starting with 60 random samples consistently. Depending on the data set, preclustering data and selecting randomly from clusters may be desirable. This will introduce additional choices around data representation and clustering method choice [e.g., using density-based spatial clustering of applications with noise (DBSCAN) on UMAP data]. However, depending on the data set, clustering may not be helpful. For example, varying ϵ for DBSCAN in Figure S31 does not give good clusters for D2R but may give rise to better results for, e.g., Tyk2. As only some data sets show clear clusters, using just random selection is the most straightforward approach when investigating different data sets and will avoid any bias introduced from clustering. Overall, understanding the relationships between data, models, and selection strategies in AL pipelines paves the way for establishing protocols for choosing these parameters in an automated fashion.

■ ASSOCIATED CONTENT

Data Availability Statement

All data for the experiments carried out can be found at https://github.com/meyresearch/ActiveLearning_BindingAffinity.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.4c00220.

Data set analysis and additional experimental results (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Antonia S. J. S. Mey − *EaStCHEM School of Chemistry, University of Edinburgh, Edinburgh EH9 3FJ, U.K.;* orcid.org/0000-0001-7512-5252; Email: antonia.mey@ed.ac.uk

Authors

Rohan Gorantla − *School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, U.K.; EaStCHEM School of Chemistry, University of Edinburgh, Edinburgh EH9 3FJ, U.K.; Exscientia, Oxford OX4 4GE, U.K.*

Alžbeta Kubincová − *Exscientia, Oxford OX4 4GE, U.K.*

Benjamin Suutari − *Exscientia, Oxford OX4 4GE, U.K.*

Benjamin P. Cossins − *Exscientia, Oxford OX4 4GE, U.K.*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c00220

Author Contributions

‖R.C. and A.K. contributed equally to this work.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the United Kingdom Research and Innovation (grant EP/S02431X/1), UKRI Centre for Doctoral Training in Biomedical AI at the University of Edinburgh, School of Informatics and Exscientia Plc, Oxford.

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

REFERENCES

(1) Reker, D. Practical considerations for active machine learning in drug discovery. *Drug Discovery Today* **2019**, *32*−*33*, 73−79.
(2) Reker, D.; Schneider, G. Active-learning strategies in computer-assisted drug discovery. *Drug Discovery Today* **2015**, *20*, 458−465.
(3) Yu, J.; Li, X.; Zheng, M. Current status of active learning for drug discovery. *Artif. Intell. Life Sci.* **2021**, *1*, 100023.
(4) Ahmadi, M.; Vogt, M.; Iyer, P.; Bajorath, J.; Fröhlich, H. Predicting Potent Compounds via Model-Based Global Optimization. *J. Chem. Inf. Model.* **2013**, *53*, 553−559.
(5) Varela, R.; Walters, W. P.; Goldman, B. B.; Jain, A. N. Iterative Refinement of a Binding Pocket Model: Active Computational Steering of Lead Optimization. *J. Med. Chem.* **2012**, *55*, 8926−8942.
(6) Fusani, L.; Cabrera, A. C. Active learning strategies with COMBINE analysis: new tricks for an old dog. *J. Comput. Aided Mol. Des.* **2019**, *33*, 287−294.
(7) Desai, B.; Dixon, K.; Farrant, E.; Feng, Q.; Gibson, K. R.; van Hoorn, W. P.; Mills, J.; Morgan, T.; Parry, D. M.; Ramjee, M. K.; Selway, C. N.; Tarver, G. J.; Whitlock, G.; Wright, A. G. Rapid Discovery of a Novel Series of Abl Kinase Inhibitors by Application of an Integrated Microfluidic Synthesis and Screening Platform. *J. Med. Chem.* **2013**, *56*, 3033−3047.
(8) Stanzione, F.; Giangreco, I.; Cole, J. C. Use of molecular docking computational tools in drug discovery. *Prog. Med. Chem.* **2021**, *60*, 273−343.
(9) Wang, R.; Lu, Y.; Wang, S. Comparative evaluation of 11 scoring functions for molecular docking. *J. Med. Chem.* **2003**, *46*, 2287−2303.
(10) Mey, A. S. J. S.; Allen, B. K.; Bruce Macdonald, H. E.; Chodera, J. D.; Hahn, D. F.; Kuhn, M.; Michel, J.; Mobley, D. L.; Naden, L. N.; Prasad, S.; Rizzi, A.; Scheen, J.; Shirts, M. R.; Tresadern, G.; Xu, H. Best Practices for Alchemical Free Energy Calculations [Article v1.0]. *Living J. Mol. Sci.* **2020**, *2*, 18378.
(11) Hahn, D. F.; Bayly, C. I.; Boby, M. L.; Bruce Macdonald, H. E.; Chodera, J. D.; Gapsys, V.; Mey, A. S.; Mobley, D. L.; Benito, L. P.; Schindler, C. E.; Tresadern, G.; Warren, G. L. Best Practices for Constructing, Preparing, and Evaluating Protein-Ligand Binding Affinity Benchmarks [Article v1.0]. *Living J. Mol. Sci.* **2022**, *4*, 1497.
(12) Kimber, T. B.; Chen, Y.; Volkamer, A. Deep learning in virtual screening: recent applications and developments. *Int. J. Mol. Sci.* **2021**, *22*, 4435.
(13) Lyu, J.; Irwin, J. J.; Shoichet, B. K. Modeling the expansion of virtual screening libraries. *Nat. Chem. Biol.* **2023**, *19*, 712−718.
(14) Gorantla, R.; Kubincova, A.; Weiße, A. Y.; Mey, A. S. J. S. From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction. *J. Chem. Inf. Model.* **2023**.
(15) Graff, D. E.; Shakhnovich, E. I.; Coley, C. W. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chem. Sci.* **2021**, *12*, 7866−7881.
(16) Fujiwara, Y.; Yamashita, Y.; Osoda, T.; Asogawa, M.; Fukushima, C.; Asao, M.; Shimadzu, H.; Nakao, K.; Shimizu, R. Virtual Screening System for Finding Structurally Diverse Hits by Active Learning. *J. Chem. Inf. Model.* **2008**, *48*, 930−940.
(17) Yang, Y.; Yao, K.; Repasky, M. P.; Leswing, K.; Abel, R.; Shoichet, B. K.; Jerome, S. V. Efficient Exploration of Chemical Space with Docking and Deep Learning. *J. Chem. Theory Comput.* **2021**, *17*, 7106−7119.
(18) Berenger, F.; Kumar, A.; Zhang, K. Y. J.; Yamanishi, Y. Lean-Docking: Exploiting Ligands' Predicted Docking Scores to Accelerate Molecular Docking. *J. Chem. Inf. Model.* **2021**, *61*, 2341−2352.
(19) Konze, K.; Bos, P. H.; Dahlgren, M. K.; Leswing, K.; Tubert-Brohman, I.; Bortolato, A.; Robbason, B.; Abel, R.; Bhat, S. Reaction-based Enumeration, Active Learning, and Free Energy Calculations to Rapidly Explore Synthetically Tractable Chemical Space and Optimize Potency of Cyclin-dependent Kinase 2 Inhibitors. *J. Chem. Inf. Model.* **2019**, *59*, 3782−3793.
(20) Ghanakota, P.; Bos, P. H.; Konze, K. D.; Staker, J.; Marques, G.; Marshall, K.; Leswing, K.; Abel, R.; Bhat, S. Combining Cloud-Based Free-Energy Calculations, Synthetically Aware Enumerations, and Goal-Directed Generative Machine Learning for Rapid Large-Scale Chemical Exploration and Optimization. *J. Chem. Inf. Model.* **2020**, *60*, 4311−4325.
(21) Thompson, J.; Walters, W. P.; Feng, J. A.; Pabon, N. A.; Xu, H.; Maser, M.; Goldman, B. B.; Moustakas, D.; Schmidt, M.; York, F. Optimizing active learning for free energy calculations. *Artif. Intell. Life Sci.* **2022**, *2*, 100050.
(22) Gusev, F.; Gutkin, E.; Kurnikova, M. G.; Isayev, O. Active learning guided drug design lead optimization based on relative binding free energy modeling. *J. Chem. Inf. Model.* **2023**, *63*, 583−594.
(23) Khalak, Y.; Tresadern, G.; Hahn, D. F.; de Groot, B. L.; Gapsys, V. Chemical Space Exploration with Active Learning and Alchemical Free Energies. *J. Chem. Theory Comput.* **2022**, *18*, 6259−6270.
(24) McInnes, L.; Healy, J.; Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction **2018**, arXiv:1802.03426
(25) Shen, W.-f.; Tang, H.-w.; Li, J.-b.; Li, X.; Chen, S. Multimodal data fusion for supervised learning-based identification of USP7 inhibitors: a systematic comparison. *J. Cheminform.* **2023**, *15*, 5−16.
(26) Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Félix, E.; Magariños, M.; Mosquera, J.; Mutowo, P.; Nowotka, M.; et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* **2019**, *47*, D930−D940.
(27) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J. Cheminform.* **2011**, *3*, 33.
(28) Zhang, Z.; Zhao, B.; Xie, A.; Bian, Y.; Zhou, S. Activity Cliff Prediction: Dataset and Benchmark. **2023**, arXiv:2302.07541 (accessed Sep 10, 2023).
(29) Achdout, H.; Aimon, A.; Bar-David, E.; Morris, G. COVID moonshot: open science discovery of SARS-CoV-2 main protease inhibitors by combining crowdsourcing, high-throughput experiments, computational simulations, and machine learning. **2020**, BioRxiv
(30) Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2018**, *85*, 1−16.
(31) Gardner, J.; Pleiss, G.; Weinberger, K. Q.; Bindel, D.; Wilson, A. G. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. *NeurIPS*, 2018; Vol. *31*.
(32) OpenEye Scientific Software. *OEChem. TK*. 2023. http://www.eyesopen.com (accessed Aug 30, 2023).
(33) Bento, A. P.; Hersey, A.; Félix, E.; Landrum, G.; Gaulton, A.; Atkinson, F.; Bellis, L. J.; De Veij, M.; Leach, A. R. An open source chemical structure curation pipeline using RDKit. *J. Cheminform.* **2020**, *12*, 51.
(34) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370−3388.
(35) Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of the 33rd International Conference on Machine Learning* 2016; pp 1050−1059.
(36) Heid, E.; Greenman, K. P.; Chung, Y.; Li, S.-C.; Graff, D. E.; Vermeire, F. H.; Wu, H.; Green, W. H.; McGill, C. J. *Chemprop: A Machine Learning Package for Chemical Property Prediction*, 2023.
(37) McInnes, L.; Healy, J.; Saul, N.; Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw.* **2018**, *3*, 861.
(38) Bellamy, H.; Rehim, A. A.; Orhobor, O. I.; King, R. Batched Bayesian Optimization for Drug Design in Noisy Environments. *J. Chem. Inf. Model.* **2022**, *62*, 3970−3981.

https://doi.org/10.1021/acs.jcim.4c00220
J. Chem. Inf. Model. 2024, 64, 1955−1965

Chapter 7

Conclusion & Future Work

The work presented in this thesis studies and develops machine learning methods for protein-ligand binding affinity prediction. Binding affinity, which quantifies interaction strength between proteins and ligands, is crucial for both identifying initial hits and optimizing lead compounds as part of optimizing ADME properties. While experiments remain the crucial step for developing new drugs, computational methods can aid in identifying the right compounds for experiments. Current computational approaches range from accurate but expensive physics-based methods, for example, alchemical free energy calculations to relatively faster but less reliable methods such as molecular docking. Machine learning offers a promising middle ground to provide reliable solutions for property predictions. The work presented in this thesis follows two main themes - (1) developing deep learning frameworks for fast and reliable binding affinity prediction during hit discovery[255,292] (Chapters 4 and 5) and (2) optimizing physics-based methods through active learning during lead optimization[246] (Chapter 6).

Deep learning frameworks for binding affinity prediction. In Chapter 4, the systematic investigation of state-of-the-art deep learning models up till 2022 provided key insights into how these models learn from protein and ligand data. The study demonstrated that protein encodings, whether derived from sequence information (1D) or enriched with structural data from contact maps (2D), had minimal impact on binding predictions. This observation remained consistent across multiple

protein contact generation methods, including AlphaFold2[45], Pconsc4[293], and ESM-1b[244], with no statistically significant differences. In contrast, ligand-based features were identified as the primary drivers of model performance, with substantial performance degradation observed when ligand information was perturbed through node and edge property randomization. Different approaches to combining protein and ligand encodings showed no significant impact on performance, further confirming the model's dependence on ligand information.

These concatenation-based approaches inherently treat the protein and ligand encoders as independent modules, permitting the model to draw predominantly on ligand features when they suffice to minimize the mean squared error during training, thereby potentially overlooking biophysical interactions. As the study used kinase-centric data, conserved sequence motifs across kinases may have produced similar protein embeddings; restricting the protein input to only interaction residues (KLIFS[294]) rather than complete sequences did not strengthen the protein-specific signal. Consequently, none of the fusion strategies tested could utilize richer 2D contact-map encodings to outperform simple 1D sequence embeddings, and the close parity between these modalities suggests the training objective did not incentivize learning distinct interaction patterns. Comparable limitations have been reported for 3D complex–based affinity predictors in Volkov et al.[54] observed that explicit 3D descriptors often provide no clear advantage over interaction-agnostic ligand or protein models, in part due to hidden biases in resources such as PDBbind[155]. We used Wilcoxon signed-rank tests and bootstrapped confidence intervals across all performance metrics to ensure that our conclusions were not artefacts of chance or label noise. This work also demonstrates the need to expand benchmarking beyond kinase-centric datasets by leveraging datasets with diverse protein families and to implement more stringent split protocols (e.g., cold-target or scaffold splits) that better reflect prospective screening challenges.

Building on these insights about the shortcomings of current deep learning approaches Chapter 5 introduces the BALM framework, which aims to address the

limitations of previous approaches at the model, data and evaluation levels. BALM utilizes pretrained protein and ligand language models (ESM-2[243] and ChemBERTa-2[199]) that have learned molecular representations from large-scale databases[167,242]. The framework optimizes cosine similarity in a shared embedding space between protein and ligand representations, moving beyond concatenation-based approaches, which had demonstrably failed at learning joint embedding spaces. Through efficient fine-tuning strategies, BALM enables rapid adaptation to new targets while maintaining computational efficiency.

The evaluation of BALM on the BindingDB[154,295] dataset demonstrated improved performance across challenging data splits, including zero-shot predictions for unseen drugs, scaffolds and targets. Overall, zero-shot on unseen targets is challenging and is not always reliable. However, BALM exhibited a notable performance boost when fine-tuned with limited additional data on unseen targets, achieving rapid adaptation with minimal training data. Additionally, comparisons with molecular docking methods, such as AutoDock Vina[296], validated BALM's effectiveness, especially where structural data availability is limited in the early screening stages. The findings also revealed limitations in BALM's ability to capture fine-grained structural differences within congeneric ligand series, especially during lead optimization. BALM's predictions tended to cluster within a narrower affinity range, suggesting challenges in differentiating subtle R-group variations. The evaluation protocols developed in Chapter 5 further address issues related to cumulative metrics, such as Pearson or Spearman correlations, which may obscure individual target performance in zero-shot settings. Using Fisher-transformed correlations to capture target-specific variability offers a more realistic assessment of model performance, emphasizing the need for granular evaluation metrics. Additionally, systematic data splits— cold drug, cold target, and scaffold are recommended to better reflect practical screening scenarios in drug discovery. The evaluation criteria and datasets used in the BALM enable a comprehensive assessment of future deep learning frameworks.

Furthermore, while the BALM framework demonstrates clear gains, there re-

mains room for refinement at the level of the loss function and data partitioning strategies. In particular, using a purely MSE loss optimises for RMSE but does not explicitly encourage correct rank ordering among similar ligands. This issue becomes apparent when attempting to prioritise congeneric series during lead optimisation. Incorporating a ranking component (e.g., pairwise or listwise ranking loss) alongside MSE could, therefore, improve the model's ability to discriminate subtle potency differences. Likewise, moving beyond current data splits to time-based splits, where earlier compounds train the model and later analogues are held out, would more faithfully emulate prospective discovery campaigns and test the model's temporal generalisation. Also, as BALM sometimes struggles when ligands share highly similar scaffolds, augmenting its input with orthogonal ligand descriptors (e.g., ECFP fingerprints that capture explicit substructures) alongside chemical-language embeddings may further enhance sensitivity to R-group modifications.

Several future research directions can help further the capability of these models. A fundamental challenge lies in the availability of high-quality binding affinity data. Currently, one of the largest publicly available datasets, BindingDB, contains approximately 25,000 protein-ligand interactions (after removing assay limits) involving around 1,100 unique targets and 9,200 unique ligands. However, developing more robust deep learning models requires significantly larger datasets that span multiple target families and include a congeneric series of ligands for individual targets. Such comprehensive datasets will enable models to better learn and capture subtle structural differences that impact binding affinity. This is particularly crucial for applications in lead optimization, where models need to distinguish between highly similar compounds.

The development of multimodal frameworks presents another promising direction. While current approaches primarily rely on sequence or complex information independently, integrating diverse data types, including protein sequences, 3D structures, interaction networks from AlphaFold[297] and their dynamics information in the form of snapshots, could provide a more comprehensive understanding of protein-

ligand binding. Information from multiple modalities can help build a base model, and this model can later be used to predict binding affinity with the available modality during the test time or fine-tune for a target-specific screening. Another key area for future investigation is the mechanistic interpretability of deep learning models and the exploration of prediction uncertainty. Understanding how these models arrive at their affinity predictions remains less explored. Methods for quantifying prediction uncertainty could be particularly valuable, enabling the identification of cases where model predictions may be less reliable. This uncertainty information could be leveraged to develop hybrid approaches that combine deep learning with physics-based methods. For instance, cases with high uncertainty could trigger more accurate but computationally intensive physics-based calculations, whose results could then be used for few-shot learning to improve the base model performance.

Active learning to support physics-based methods in lead optimization is a first step in this direction. In Chapter 6, I explored how machine learning can support alchemical free energy (AFE) calculations during lead optimization. While AFE calculations provide excellent accuracy (RMSE $\sim$ 1 kcal/mol), their computational cost is high to search large compound spaces[52]. The systematic evaluation of active learning protocols across four diverse protein targets (TYK2[250], USP7[254], D2R[298], Mpro[299]) revealed several key insights about efficiently prioritizing compounds for AFE calculations.

The investigation demonstrated that both Gaussian Process (GP) models and graph neural networks can effectively guide compound selection, though their performance characteristics differ. GP models showed superior performance and more consistent compound selections in the limited data scenarios I tested. This finding was consistent across multiple evaluation metrics that captured both the identification of top binders and the overall predictive quality of the models.

The study identified optimal batch sizes for different scenarios - larger initial batches improved overall performance with diverse datasets, while smaller batches

proved more effective in later iterations. When reducing batch sizes below 30 samples, performance gains became incremental, suggesting a practical lower limit given the increased number of AL cycles and overall turnaround time. From a computational perspective, this batch size threshold aligns well with AFE calculation workflows, as it enables efficient parallelization of multiple calculations simultaneously. Having more calculations run in parallel, while more computationally demanding, can provide free energy estimates faster which can be used in the next AL cycle. Dataset characteristics strongly influenced these outcomes, with more diverse compound sets benefiting from larger initial exploration phases compared to focused sets with limited scaffold diversity.

A critical finding was the impact of dataset composition on active learning effectiveness. While dataset size played a role, the chemical diversity of compounds emerged as the decisive factor in determining the potential benefits of active learning. This observation held true across different models and sampling strategies, highlighting the importance of considering dataset characteristics when designing active learning protocols.

The comprehensive evaluation of active learning protocols revealed key relationships between data characteristics, model behaviour, and selection strategies. While these insights provide a foundation for parameter selection in AL pipelines, several critical areas require investigation before achieving fully automated protocols.

A primary direction for future research involves understanding the sensitivity of GP models to different molecular representations and kernel choices. While my work utilized ECFP fingerprints, the findings from Chapter 5 highlighted that GP model performance can vary significantly based on fingerprint and kernel selection. Future studies should systematically evaluate different featurization methods and kernel functions across diverse datasets to establish more robust guidelines for model configuration. Another crucial challenge identified in both Chapters 5 and 6 relates to capturing subtle structural modifications. The GP model showed limited effectiveness in scenarios involving single-core scaffolds with R-group variations, mirroring

BALM's challenges with congeneric series. This consistent limitation across different approaches suggests a fundamental challenge in computational methods for lead optimization. Developing models that can reliably distinguish small structural changes while maintaining computational efficiency remains an important research direction. One potential approach could be investigating how pretrained models such as BALM could be adapted for active learning scenarios, with a specific focus on improving their sensitivity to subtle structural variations.

The optimization of sampling strategies presents another promising research direction. While my current work examined exploration and exploitation strategies independently in each cycle, future research could investigate dynamic combinations of these strategies within individual cycles. Understanding how to adjust these combinations based on dataset characteristics and chemical space complexity could lead to more efficient compound selection protocols. This could include developing adaptive strategies that automatically adjust based on the diversity of the remaining compound pool and the current stage of optimization. These research directions aim to address fundamental challenges in active learning for drug discovery, working towards more automated and reliable protocols that can effectively support physics-based methods in lead optimization campaigns.

This thesis establishes machine learning as a bridge between high-throughput screening and accurate physics-based simulations. BALM's efficiency in screening large compound libraries and ability to generalise to unseen targets with few-shot training and active learning's cost reductions for AFE calculations together address critical bottlenecks across the drug discovery pipeline. While deep learning models cannot yet replace physics-based methods for lead optimisation, they provide intelligent prioritisation tools—guiding experiments and simulations toward the most promising candidates. These advances highlight the potential of machine learning when grounded in rigorous evaluation and biological realism, paving the way for more automated, data-driven drug discovery workflows.

Chapter 8

Appendix

8.1 Responsible Research and Innovation

Our research focuses on developing deep learning methods for estimating protein-ligand binding affinity, which could have potential direct and indirect societal benefits. By accelerating the drug discovery phase, these methods could expedite the development of treatments for currently untreatable diseases while significantly reducing costs. Given that drug research is partially funded by government resources, ML-based binding affinity estimation techniques can optimize experimental planning and ensure efficient use of taxpayer money.

The research adheres to key principles of responsible innovation through several aspects. At the technical level, our work emphasizes reproducibility and reliability through rigorous evaluation protocols and comprehensive benchmarking against established methods. The development of BALM incorporates parameter-efficient strategies that reduce computational overhead, addressing both economic and environmental considerations. Our active learning protocols further demonstrate responsible resource utilization by optimizing the selection of compounds for expensive physics-based calculations.

We acknowledge potential environmental impacts, particularly the carbon footprint associated with GPU-based computational resources required for deep learning models. However, these environmental costs are mitigated through careful experi-

mental design and efficient model architectures. The parameter-efficient fine-tuning strategies implemented in BALM significantly reduce computational requirements compared to traditional approaches. Furthermore, by improving the efficiency of drug discovery processes, our methods could potentially reduce the overall environmental impact of pharmaceutical research and development.

The research also addresses broader societal responsibilities through its commitment to open science and knowledge sharing. All developed methods and evaluation protocols are documented and made publicly available, enabling broader scientific community engagement and validation. This transparency supports the collective advancement of the field while ensuring the responsible development and application of these technologies.

We remain mindful of potential misuse scenarios, particularly when designing molecules for unintended purposes. While our methods focus specifically on therapeutic applications, we acknowledge that similar techniques could potentially be adapted for developing other bioactive compounds. However, existing regulatory frameworks and quality control measures in pharmaceutical development provide important safeguards against misuse.

Our approach to responsible innovation extends to data handling and model development. We carefully consider the potential impact on model predictions, implementing evaluation strategies that assess performance across diverse chemical spaces and target families. This attention to data quality and representation helps ensure the development of more equitable and reliable prediction methods.

Through these various dimensions, our research demonstrates a commitment to responsible innovation while advancing the technical capabilities of computational drug discovery methods. The work contributes to the broader goal of making drug development more efficient and accessible while maintaining high standards of scientific rigour and ethical consideration.

8.2 Supplementary Information - From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction

Supporting Information

From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction

Rohan Gorantla,[†,‡] Alžbeta Kubincová,[¶] Andrea Y. Weiße,[§,†] and Antonia S. J. S. Mey[*,‡]

†*School of Informatics, University of Edinburgh, EH8 9AB, UK*

‡*EaStCHEM School of Chemistry, University of Edinburgh, EH9 3FJ, UK*

¶*Exscientia, Schrödinger Building, Oxford, OX4 4GE, UK*

§*School of Biological Sciences, University of Edinburgh, EH9 3FF, UK*

E-mail: antonia.mey@ed.ac.uk

1

Supporting Information Available

Methods

Deep Learning model architectures and implementation details

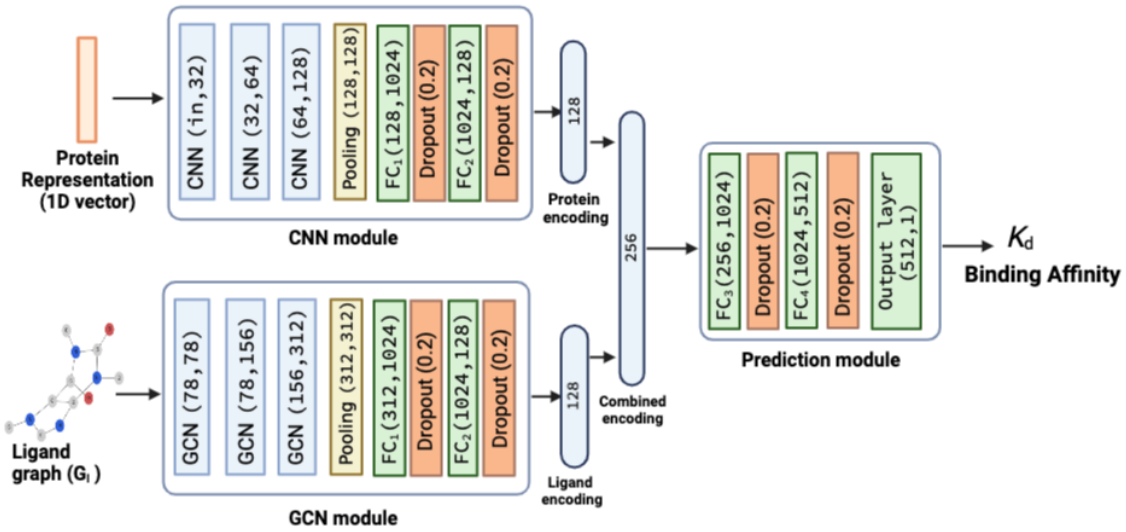


Figure S1: Convolutional neural network (CNN), GCN, and binding affinity prediction modules for studying the impact of 1D protein encodings and ligand graphs. The neural network layers in both the modules are shown along with their input and output channel sizes, i.e., *(input channel size, output channel size)*. The GCN module and prediction module architecture is the same as Figure S2. 1D protein representations (`in` is 1785 for KLIFS and 1280 for ESM) and ligand graphs are passed through the CNN module and GCN module respectively. For the proteins, features are first extracted from three 1D CNN layers with stride as 1, and increasing kernel sizes $[4, 8, 12]$. The output from the 1D CNN layers is then passed through the pooling layer and the pooling output is passed through two fully connected (FC) layers. The flattened feature vector of 128 dimensions is given as input to the first FC_1 layer in CNN module with 1024 neurons and outputs a feature vector of 1×1024 dimension. A dropout layer is added after the FC layer with a dropout rate of 0.2. The output from FC_1 layer is passed to FC_2 layer with 128 neurons. We obtain 128-dimensional protein and ligand encodings which are then combined to form a 256-dimensional embedding. We implement the 1D-CNN layer with `torch.nn.Conv1d()`, pooling layer with `torch.nn.AdaptiveMaxPool1d()`, FC and output layers with `torch.nn.Linear()` and the dropout layer with `torch.nn.Dropout()` class.

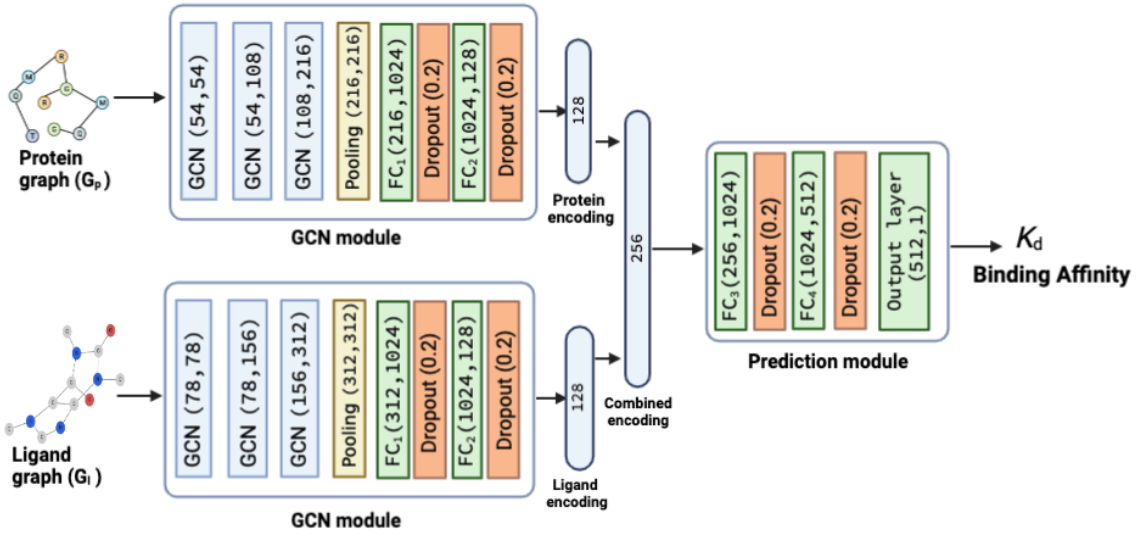


Figure S2: Graph convolutional network (GCN) and binding affinity prediction modules for studying the impact of protein and ligand graphs. The neural network layers in both the modules are shown along with their input and output channel sizes, i.e., *(input channel size, output channel size)*. Protein and ligand graphs are passed through the GCN module, here features are first extracted from 3 GCN layers and then passed through the pooling layer to get a single column vector of dimension **g** (**g** is 216 and 312 for protein and ligand graphs, respectively). These vectors from the pooling layer are then passed through two fully connected (FC) layers. The flattened feature vector of dimension **g** is given as input to the first FC_1 layer with 1024 neurons and outputs a feature vector of 1×1024 dimension. A dropout layer is added after the FC layer with a dropout rate of 0.2 to avoid overfitting and co-adaptation issues. The output from FC_1 layer is passed to FC_2 layer with 128 neurons. We obtain 128-dimensional protein and ligand encodings which are then combined to form a 256-dimensional embedding. We pass the 256-dimensional embedding into the prediction module with two FC layers and one output layer. The FC_3 and FC_4 layers have 1024 and 512 neurons, respectively, and a dropout layer is added after them. The output layer is similar to an FC layer with one neuron, and it takes 512-dimensional embedding to predict the binding affinity score. We implemented GCN layer with `torch_geometric.nn.GCNConv()`, pooling layer with `torch_geometric.nn.global_mean_pool()`, FC and output layers with `torch.nn.Linear()` and the dropout layer with `torch.nn.Dropout()` class. For studying element-wise product, we take an element-wise product of the 128-dimensional protein and ligand encoding to obtain the a 128-dimensional vector which is fed to the FC_3 layer instead of 256-dimensional vector in the case of concatenation. Similarly, for combined concatenation and element-wise product vector, the FC_3 layer will get an input vector of 384 dimensions.

Node features in a protein graph

Position-Specific Scoring Matrix (PSSM) provides the per-residue evolution patterns in the sequence profile [1]. By first counting the instances of each residue at each position, a position

frequency matrix $\mathbf{M}^{\mathrm{PFM}}$ is generated using the equation below [2]

$$M_{\mathrm{k,j}}^{\mathrm{PFM}} = \sum_{i=1}^{Z} I\left(S_{i,j} = k\right), \tag{1}$$

where S is a set of Z aligned sequences for a protein sequence with length of L_{p}, k belongs to residue symbols set, $i = (1, 2, \ldots, Z), j = (1, \ldots, L_{\mathrm{p}})$ and $I(x)$ is an indicator function when the condition x is satisfied and 0 otherwise. A position probability matrix $\mathbf{M}^{\mathrm{PPM}}$ is then computed from the $\mathbf{M}^{\mathrm{PFM}}$ matrix using the following equation

$$M_{\mathrm{k,j}}^{\mathrm{PPM}} = \frac{M_{\mathrm{k,j}}^{\mathrm{PFM}} + \frac{c}{4}}{Z + c}, \tag{2}$$

where c is the added pseudo count that is empirically set to 0.8 similar to [2] to avoid matrix entries with a value of 0 [3]. The $\mathbf{M}^{\mathrm{PPM}}$ matrix is then utilized to compute 21 PSSM features. For computing PSSM features, we need an aligned protein sequence in PSICOV [4] format. We used the aligned protein sequences in PSICOV format provided by Jiang et al. [2]. Table S1 below contains information about the rest of the node features.

Ligand randomizations

In the point randomization process, we enumerate the presence of certain atoms (such as Cl, F, Br, and (=O)) within the string, and selectively modify up to four atoms. This can involve substituting one halogen atom with another or removing a (=O) atom. In cases where none of the enumerated atoms exist, a Cl atom is appended at the beginning of the SMILES string. The appending of chlorine was influenced by its prevalent incorporation in drug-like molecules due to its effects on lipophilicity, electronic distribution, and steric hindrance, impacting binding affinity and pharmacokinetics [5]. While chlorine' capability to expand its octet is noteworthy [5], its frequent representation in medicinal chemistry primarily drove its selection. This variation helps us ascertain if small changes in ligand structure can influence binding affinity prediction and identify if the model accurately captures these structural

Table S1: Residue node features in a protein graph

Node Features	Dimensions
One-hot encoding of the residue symbol	21
Position-specific scoring matrix	21
Whether the residue is aromatic	1
Whether the residue is aliphatic	1
Whether the residue is polar neutral	1
Whether the residue is acidic charged	1
Whether the residue is basically charged	1
Residue weight	1
Negative of the logarithm of the dissociation constant for the −COOH group	1
Negative of the logarithm of the dissociation constant for the −NH group	1
Negative of the logarithm of the dissociation constant for any other group in the molecule	1
pH at the isoelectric point	1
Hydrophobicity of residue (pH = 2)	1
Hydrophobicity of residue (pH = 7)	1
Total	54

alterations.

KIBA metric

Kinase inhibitor bioactivity (KIBA) score is a continuous value of binding affinity developed by Jing et al. [6] integrating the information from biological activity of kinase inhibitors from K_{i}, IC_{50}, and K_{d} into a single bioactivity score [6]. Lower KIBA score denotes higher binding affinity. The KIBA score can be defined based on K_{d} or K_{i}, or the average of them, depending on the availability of the bioactivity types [6] using the equation below

$$\begin{aligned}
\mathrm{KIBA} &= \frac{\mathrm{IC}_{50}}{1 + H_{\mathrm{i}}\left(\mathrm{IC}_{50}/K_{\mathrm{i}}\right)} && \text{if } K_{\mathrm{i}} \text{ and } IC_{50} \text{ are present} \\
&= \frac{\mathrm{IC}_{50}}{1 + H_{\mathrm{d}}\left(\mathrm{IC}_{50}/K_{\mathrm{d}}\right)} && \text{if } K_{\mathrm{d}} \text{ and } IC_{50} \text{ are present} \\
&= \left(\frac{\mathrm{IC}_{50}}{1 + H_{\mathrm{i}}\left(\mathrm{IC}_{50}/K_{\mathrm{i}}\right)} + \frac{\mathrm{IC}_{50}}{1 + H_{\mathrm{d}}\left(\mathrm{IC}_{50}/K_{\mathrm{d}}\right)}\right)/2 && \text{if } K_{\mathrm{i}},\ K_{\mathrm{d}} \text{ and } IC_{50} \text{ are present,}
\end{aligned} \tag{3}$$

Algorithm 1: Point Randomization of Ligands

Input : `SMILES` string, S_i

Initialization: Get the count of Cl, F, Br, C and (=O) atoms from the input S_i and set to #Cl, #F, #Br, #C and #(=O) respectively.

$N \leftarrow$ #Cl + #F + #Br + #(=O);

if $N > 0$ AND $N \leq 4$ **then**

  Randomly select n changes to be made from the range of 1 to N possible changes;

  while $n \neq 0$ **do**

    Randomly select an atom from Cl, F, Br, and (=O) which are present in S_i;

    if *Halogen (i.e., Cl) is selected* **then**

      Replace selected halogen (Cl) from S_i with one of other halogens (F or Br) randomly to obtain an updated `SMILES` string S_i ;

      #Cl $\leftarrow$ #Cl $-$ 1 ; `/* Update the Halogen count */`

      $n \leftarrow n - 1$;

      `continue`

    end

    if *(=O) is selected* **then**

      Remove one (=O) from S_i to obtain an updated `SMILES` string S_i ;

      #(=O) $\leftarrow$ #(=O) $-$ 1;

      $n \leftarrow n - 1$;

      `continue`

    end

  end

else

  $S_i \leftarrow$ Cl+ S_i ;

end

Output: Updated `SMILES` string, S_o

where H_i and H_d are the parameters that determine the weights of IC_{50} in the model-based adjustments for K_i and K_d.

Table S2: Overview of various experiments performed in our study

Protein encoding	**Ligand encoding**	**Combining method**
To study the impact of 1D and 2D protein encodings		
Pconcs4	Original ligand graph	Concatenation
Alphafold2	Original ligand graph	Concatentation
ESM	Original ligand graph	Concatentation
Random	Original ligand graph	Concatentation
KLIFS (1D)	Original ligand graph	Concatentation
ESM (1D)	Original ligand graph	Concatentation
To study the impact of ligand encodings		
Pconcs4	Point randomisation	Concatentation
Pconcs4	Random node features	Concatentation
Pconcs4	Random sampling	Concatentation
Assessing the impact of combining methods		
Pconcs4	Original ligand graph	Element-wise product
Pconcs4	Original ligand graph	Concatenation + Element-wise product

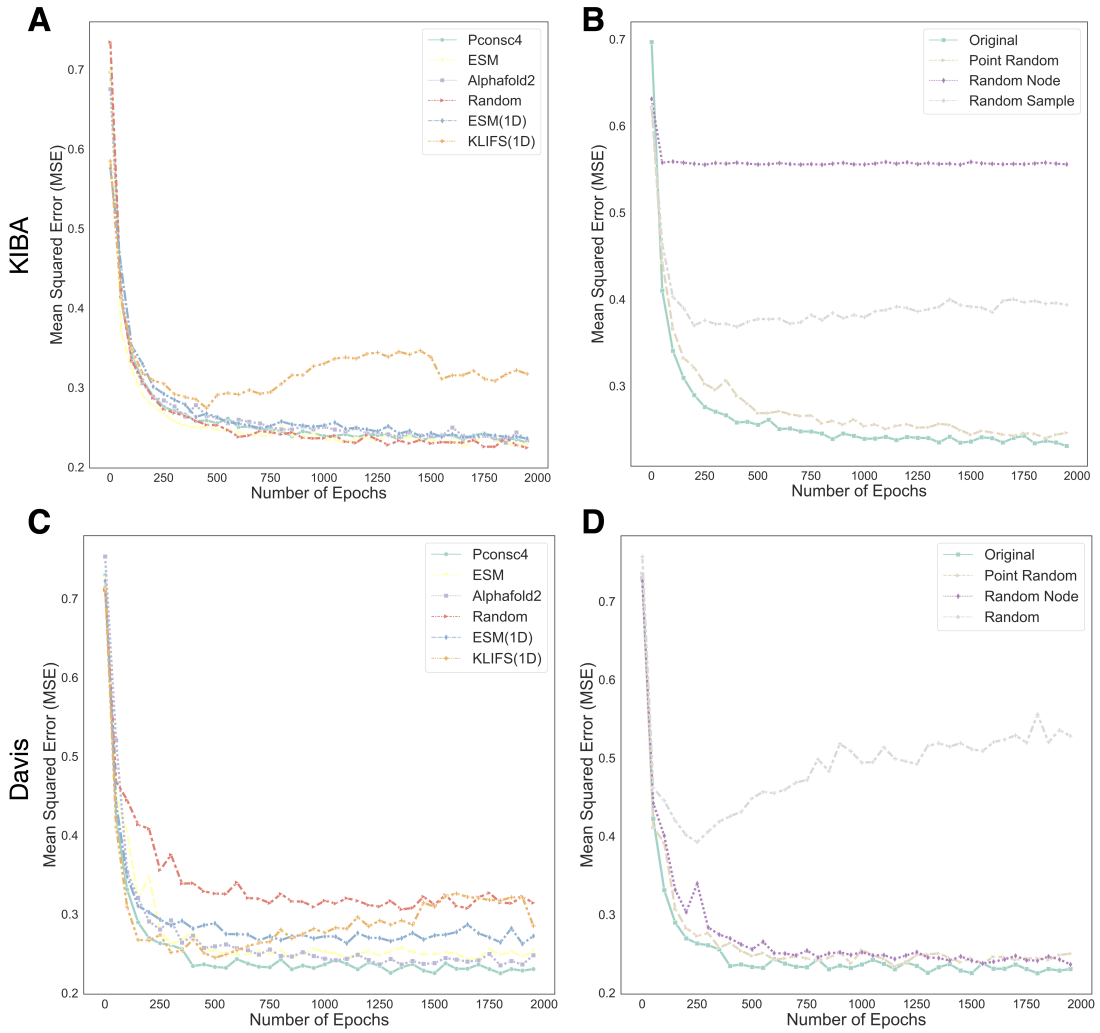


Figure S3: Mean squared error (MSE) curves on validation data during the training of DL models using various protein and ligand encodings. **A, C** MSE curves for protein encodings on the KIBA and Davis dataset. **B, D** MSE curves for ligand encodings on the KIBA and Davis dataset.

Results

Contact map evaluation

The Matthews Correlation Coefficient (MCC) [7] is a widely used metric to assess the quality of binary classifiers, including those used in protein contact prediction. In this context, MCC measures the agreement between predicted and true contact maps, which are binary matrices

indicating the presence or absence of contact between pairs of residues in a protein. MCC can be calculated from a confusion matrix that summarizes the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) obtained by comparing the predicted contact map to the true contact map. The MCC is given by

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP}) \times (\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{TN} + \text{FN})}} \tag{4}$$

MCC values range from -1 to 1 , with 1 indicating perfect agreement, 0 indicating random prediction, and -1 indicating complete disagreement between predicted and true contact maps. One advantage of using MCC in cases with imbalanced datasets is that it takes into account both true positives and true negatives, as well as false positives and false negatives, to provide a balanced assessment of the performance of the classifier. This is particularly important when the positive and negative classes are not balanced in the dataset, as metrics such as accuracy can be misleading. In the context of contact map prediction, the true contacts (positive class) are typically much rarer than the non-contacts (negative class), resulting in an imbalanced dataset.

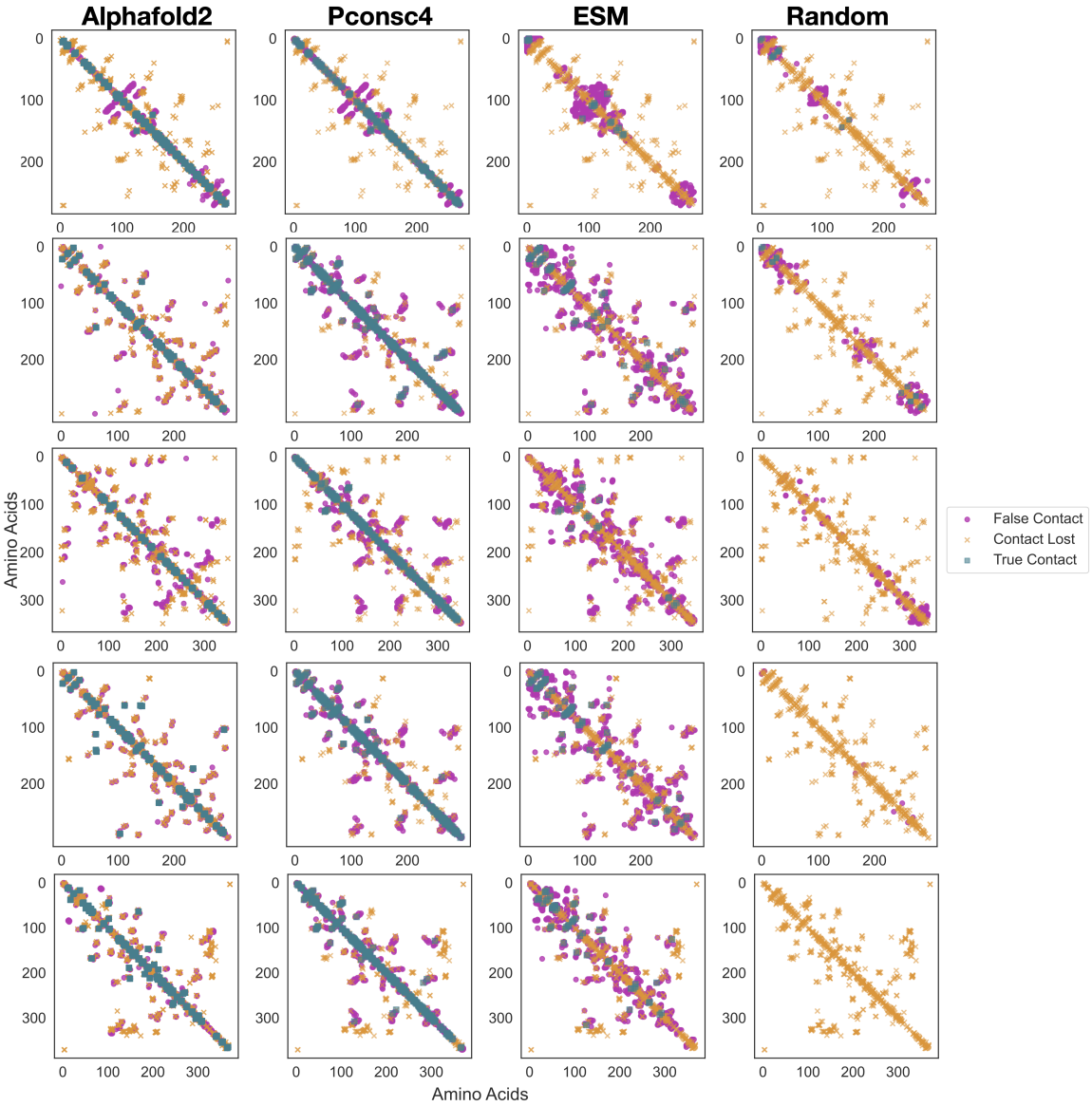


Figure S4: Visual illustration of contact maps obtained from various kinases present in KIBA and Davis datasets as compared to the contact maps obtained from PDB structures of kinases. The true contacts, false contact and contacts lost are highlighted.

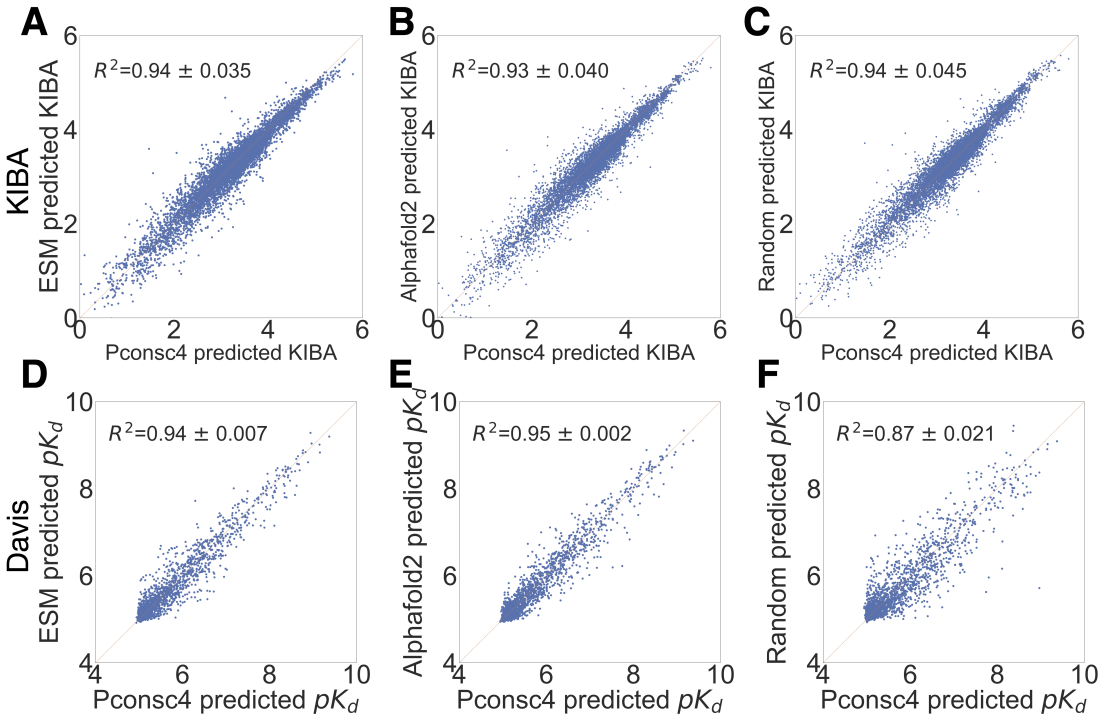


Figure S5: Correlation between binding affinity predictions given by the graph-DL model with different protein encoding methods- **A, B, and C:** KIBA and **D, E, and F:** Davis. The protein encoding based on random contact map is also strongly correlated with Pconsc4 methods. All these scatter plots show that the BA predictions are not impacted by the protein encodings obtained from various contact map methods and function in the same way on both the datasets.

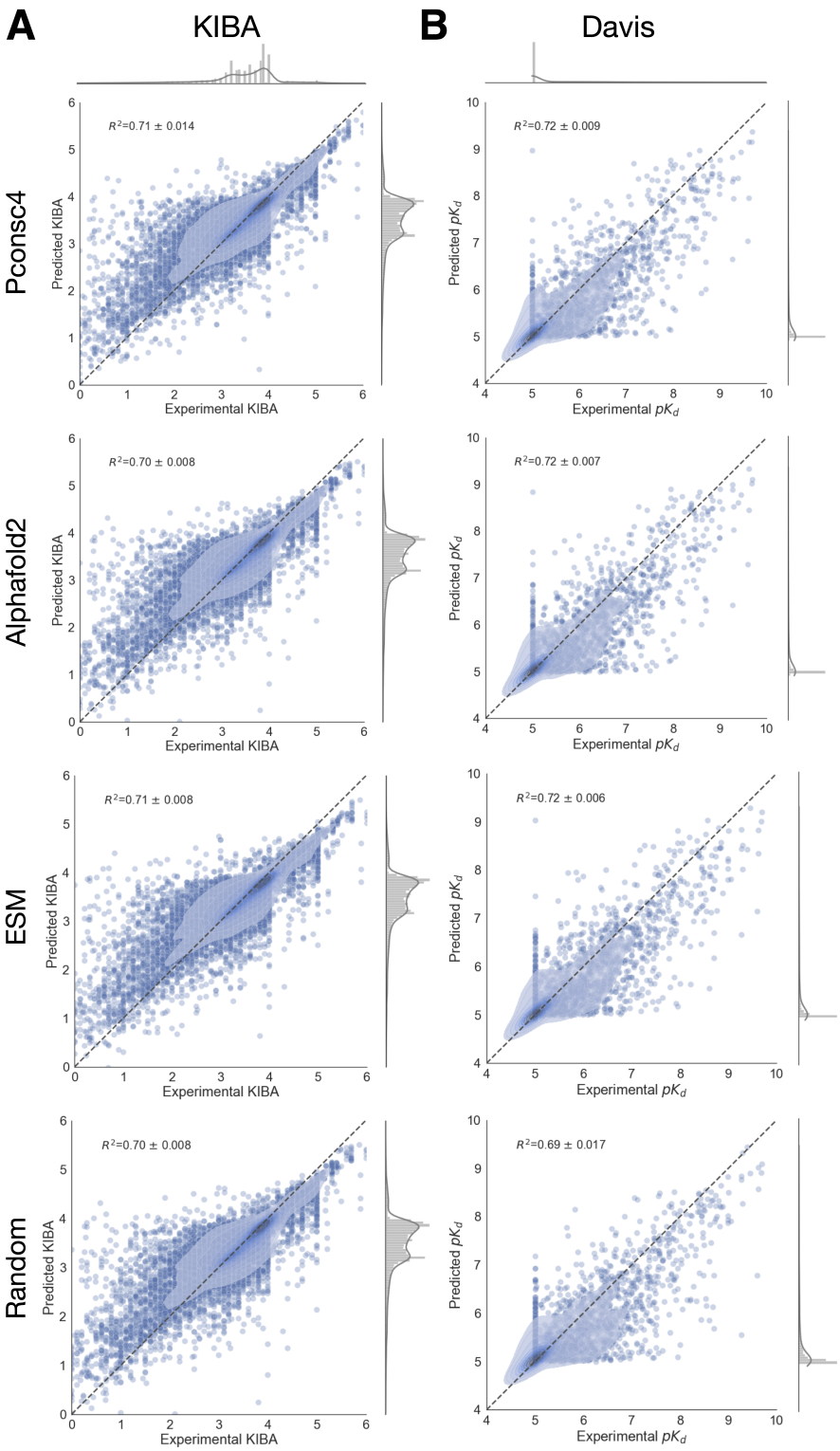


Figure S6: Experimental vs. predicted binding affinities of DL model trained using four different 2D encodings generated from contact maps on **A:** KIBA and **B:** Davis datasets.

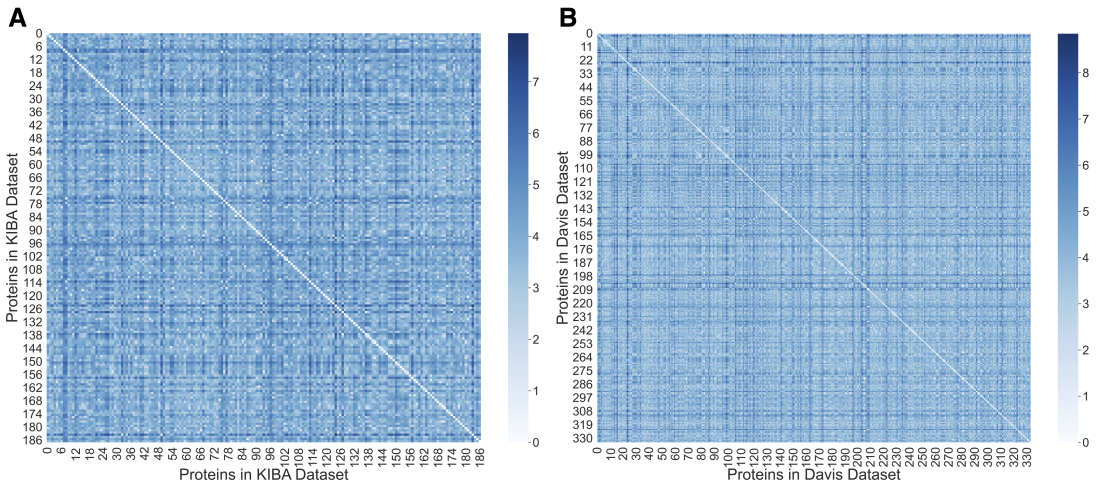


Figure S7: Euclidean distance between the ESM embeddings of proteins in the Davis and KIBA datasets. These embeddings capture 95% variance among the kinases in the Davis and KIBA datasets, and from the heatmap we can see that the Euclidean distance between the ESM embeddings of kinases is considerable, indicating a substantial distance between them in the Euclidean space. **A:** ESM embeddings from 188 KIBA proteins. **B:** ESM embeddings from 334 Davis proteins.

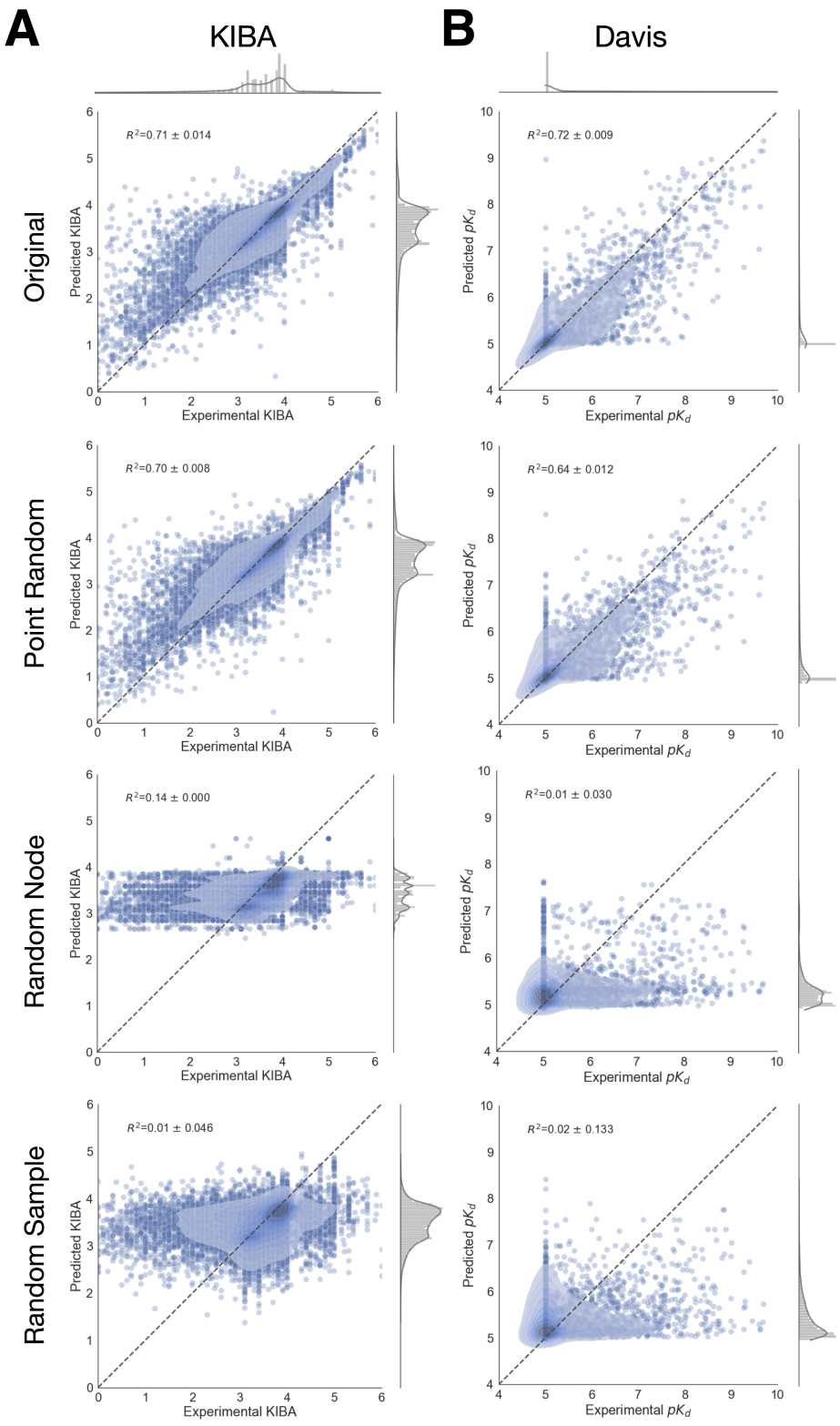


Figure S8: Experimental vs. predicted binding affinities of DL model trained using various perturbations of ligand graph encodings on both **A:** KIBA and **B:** Davis datasets.

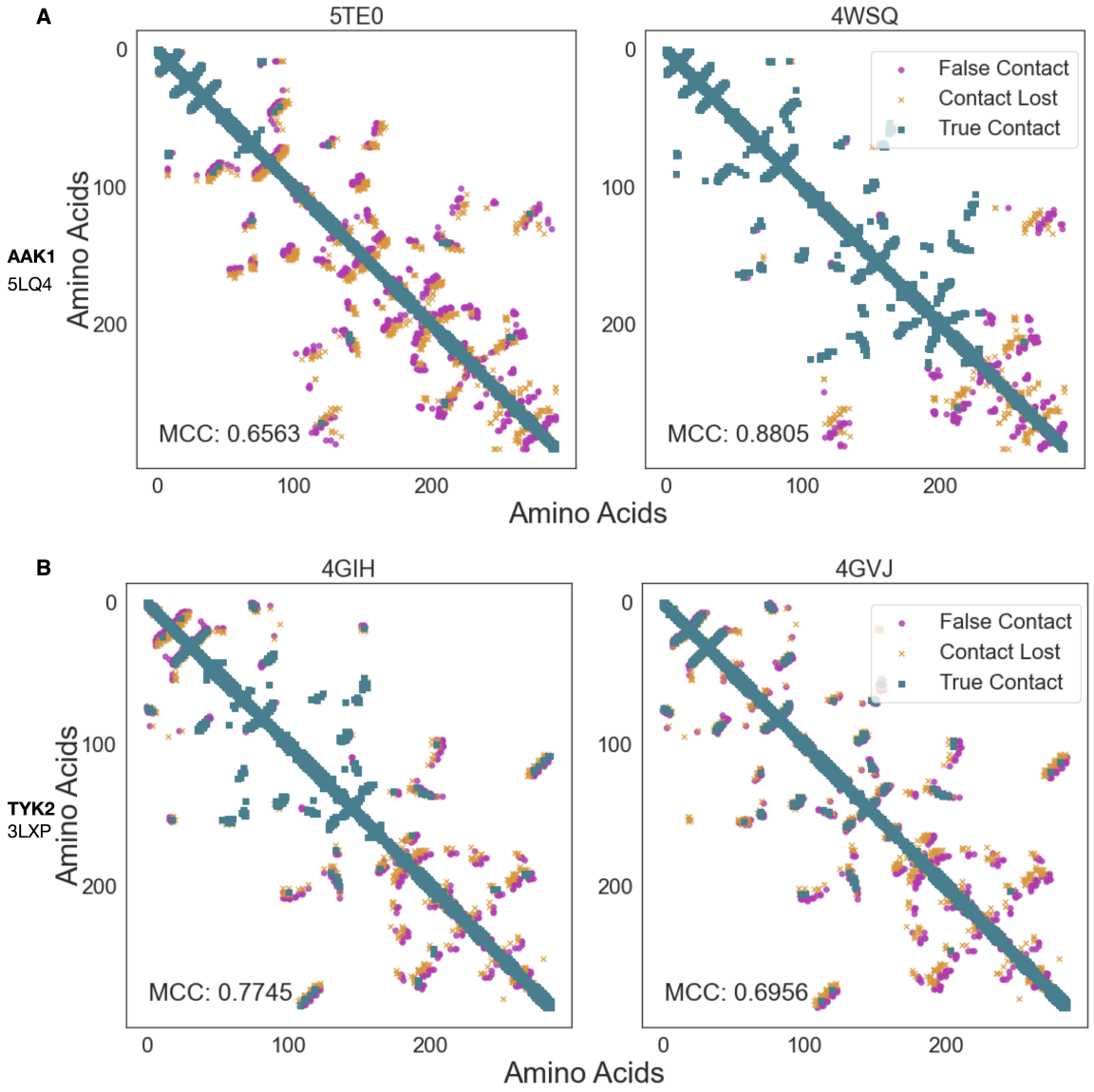


Figure S9: Comparing contact maps obtained from different x-ray structures- **A:** AAK1 and **B:** TYK2 kinases. For AAK1, we have used 5LQ4 as reference PDB structure and compared it with 5TE0 and 4WSQ PDBs. Similarly, in the case of TYK2, we have used 3LXP as reference PDB structure and compared it with 4GIH and 4GVJ PDBs. We have also shared MCC values comparing each PDB with the reference PDB.

References

(1) Guo, Y.; Wu, J.; Ma, H.; Wang, S.; Huang, J. Comprehensive Study on Enhancing Low-Quality Position-Specific Scoring Matrix with Deep Learning for Accurate Protein Structure Property Prediction: Using Bagging Multiple Sequence Alignment Learning. *J. Comput. Biol.* **2021**, *28*, 346–361.

(2) Jiang, M.; Li, Z.; Zhang, S.; Wang, S.; Wang, X.; Yuan, Q.; Wei, Z. Drug–target affinity prediction using graph neural network and contact maps. *RSC Adv.* **2020**, *10*, 20701–20712.

(3) Nishida, K.; Frith, M. C.; Nakai, K. Pseudocounts for transcription factor binding sites. *Nucleic Acids Res.* **2009**, *37*, 939–944.

(4) Jones, D. T.; Buchan, D. W.; Cozzetto, D.; Pontil, M. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics* **2012**, *28*, 184–190.

(5) Fang, W.-Y.; Ravindar, L.; Rakesh, K.; Manukumar, H.; Shantharam, C.; Alharbi, N. S.; Qin, H.-L. Synthetic approaches and pharmaceutical applications of chloro-containing molecules for drug discovery: A critical review. *Eur. J. Med. Chem.* **2019**, *173*, 117–153.

(6) Tang, J.; Szwajda, A.; Shakyawar, S.; Xu, T.; Hintsanen, P.; Wennerberg, K.; Aittokallio, T. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *J. Chem. Inf. Model.* **2014**, *54*, 735–743.

(7) Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 1–13.

8.3 Supplementary Information - Learning Binding Affinities via Fine-tuning of Protein and Ligand Language Models

Supporting Information

Learning Binding Affinities via Fine-tuning of Protein and Ligand Language Models

Rohan Gorantla,[†,‡] Aryo Pradipta Gema,[†] Ian Xi Yang,[‡] Álvaro Serrano-Morrás,[¶,§] Benjamin Suutari,[‖] Jordi Juárez-Jiménez,[¶,§] and Antonia S. J. S. Mey[*,‡]

†*School of Informatics, University of Edinburgh, Crichton Street, Edinburgh, EH8 9AB, Midlothian, United Kingdom*

‡*EaStCHEM School of Chemistry, University of Edinburgh, David Brewster Road, Edinburgh, EH9 3FJ, Midlothian, United Kingdom*

¶*Unitat de Fisicoquímica, Departament de Farmàcia i Tecnologia Farmacèutica, i Fisicoquímica, Facultat de Farmàcia i Ciències de l'Alimentació, Universitat de Barcelona (UB), Av. Joan XXIII, 27-31, 08028 Barcelona, Spain*

§*Institut de Química Teòrica i Computacional (IQTC), Facultat de Química i Física, Universitat de Barcelona (UB), C. Martí i Franquès, 1, 08028 Barcelona, Spain*

‖*Independent Researcher, 37027, United States*

E-mail: antonia.mey@ed.ac.uk

Additional Results

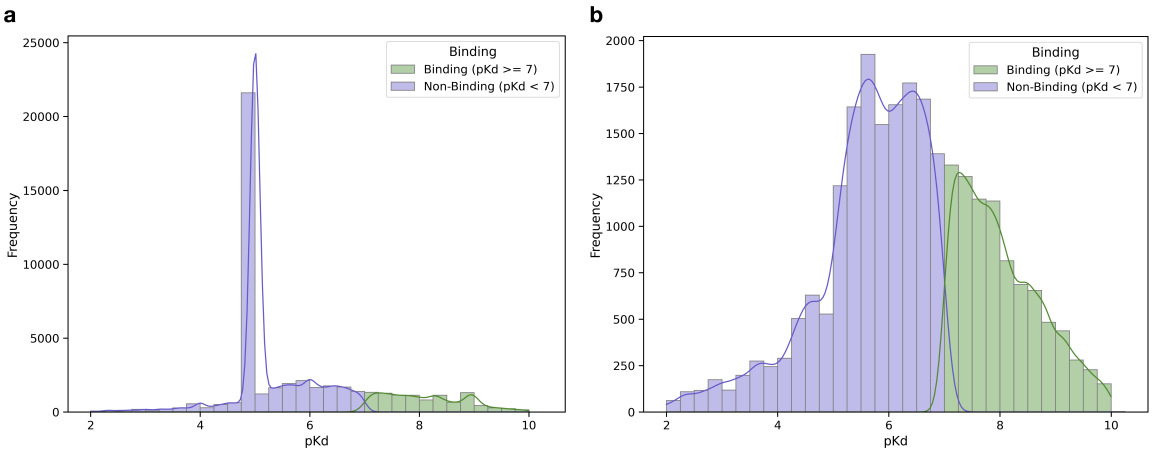


Figure S1: **BindingDB distribution of pK_d values before and after removing assay limits. (a)** The distribution of binding affinities (pK_d) in the original BindingDB dataset shows significant skewness, primarily caused by the dominance of interactions with assay limits, where the most frequent value ($pK_d = 4.99$) accounts for a large portion of the data. **(b)** After removing the top five most frequent assay limits, the dataset demonstrates a more balanced distribution of binding affinities, providing a better foundation for model training and reducing prediction bias.

BindingDB data distribution

Parameter-efficient fine-tuning significantly improves BALM's performance For both ligand and protein-only fine-tuning, we tested various PEFT methods and ranks. LoHa and LoKr showed improvement in performance across all ranks as compared to the no fine-tuning model, while LoRA showed improvement with rank 8. However, the performance dropped at higher ranks. LoHa seemed to be stable, with rank 16 giving the best improvement of approximately 9.4%, showing a significant improvement over the BALM model without fine-tuning. Other ranks for LoHa also provided notable improvements, achieving around 8.7% (rank 8) and 8.3% (rank 32). IA3, while beneficial, offered a moderate gain of about 5.0%. In protein-only fine-tuning, LoKr seemed to be consistent across all ranks, with the highest improvement of approximately 18.2% at rank 8. Other ranks for LoKr also showed significant improvements, achieving gains of approximately 17.4% (ranks 16 and 32). IA3 and LoHa also demonstrated substantial improvements, with LoHa (rank 8) achieving

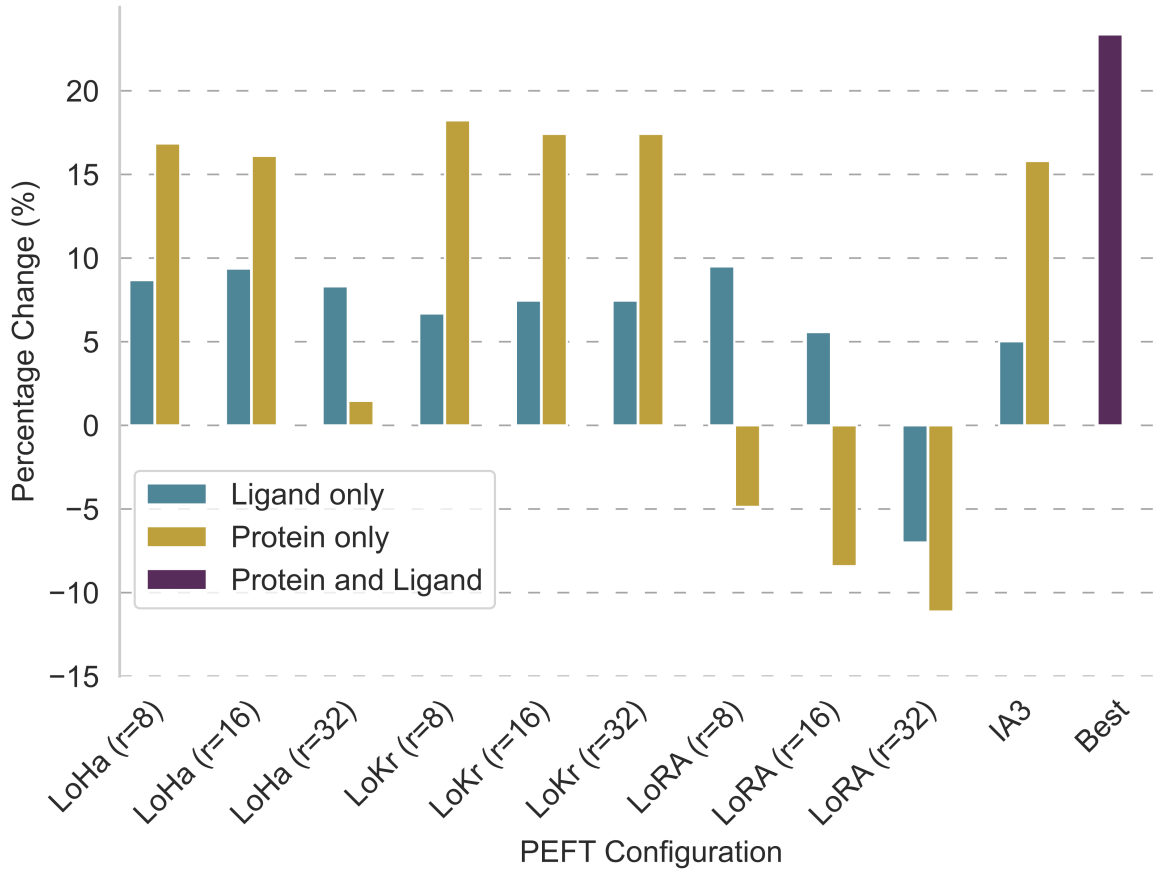


Figure S2: **Comparative analysis of parameter-efficient fine-tuning (PEFT) methods for protein and ligand language models.** This study evaluates the performance of various PEFT methods, specifically LoRA, LoHa, LoKr, and IA^3, in fine-tuning protein and ligand language models within the BALM framework. The performance of these methods is measured using Pearson correlation and compared against BALM without fine-tuning. Initially, we fine-tune protein and ligand language models separately in the BALM framework. Subsequently, the most effective fine-tuning methods (denoted as Best in the plot) for protein (LoKr) and ligand (LoHa) models are applied to assess their combined impact on the performance of BALM+PEFT.

around 16.8% gain and LoHa (rank 16) showing a gain of 16.1%. In contrast, LoRA showed a drop in performance. Finally, we combined the best-performing fine-tuning methods, LoHa for ligands and LoKr for proteins, and chose rank 16 for both to keep it consistent to create the BALM+PEFT variant. This combination yielded a performance gain of around 23.4% compared to BALM with no fine-tuning.

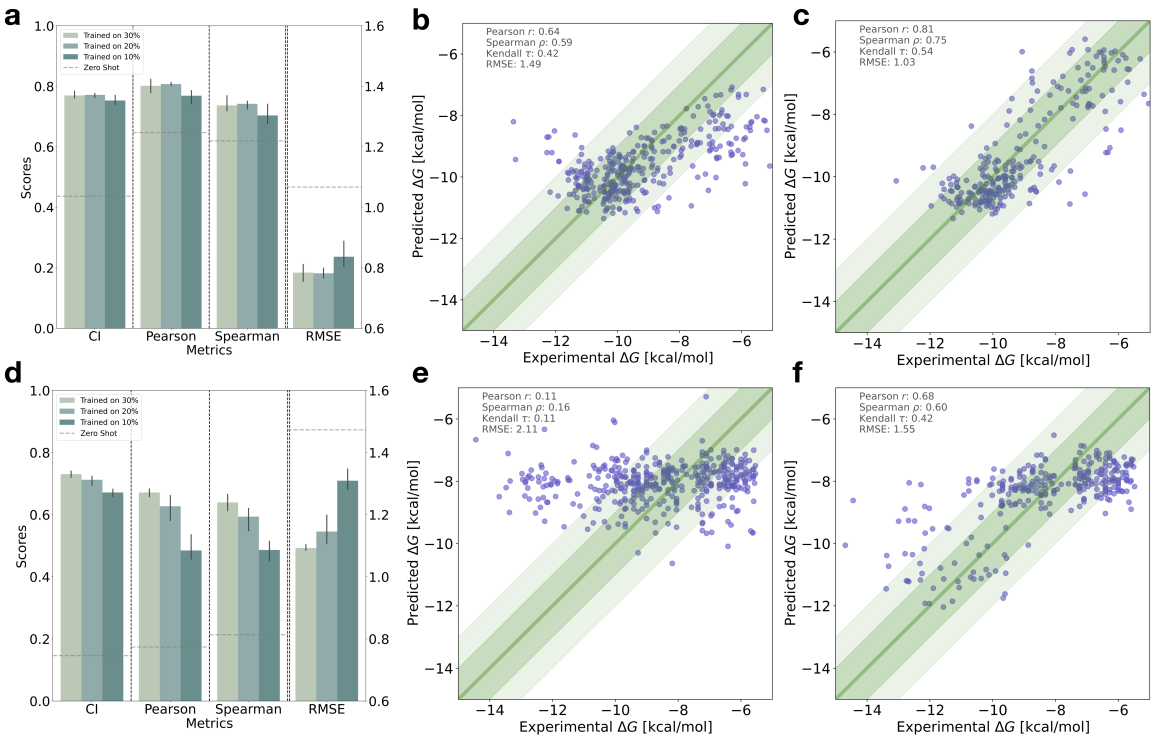


Figure S3: **Zero-shot and few-shot performance of the BALM+PEFT model on *Mpro* and *USP7* targets. (a, d)** Performance comparison of the pre-trained BALM+PEFT model (zero-shot) and few-shot fine-tuning using 10%, 20%, and 30% of experimental data. Testing is performed on the remaining *USP7* (a) and *Mpro* (d) data. The model is fine-tuned by retraining only the projection layer, and the data is split randomly with three different seeds. Error bars indicate the standard deviation across different splits, and performance metrics are reported for Concordance Index (CI), Pearson correlation, Spearman rank correlation, and Root Mean Squared Error (RMSE). **(b, e)** Scatter plots showing zero-shot model predictions for *USP7* (b) and *Mpro* (e) targets. Experimental ΔG values (kcal/mol) are on the x-axis and predicted ΔG values are on the y-axis. Only 20% of the test set (selected randomly) is shown for readability. **(c, f)** Scatter plots showing few-shot model performance on 20% of the training data for *USP7* (c) and *Mpro* (f) targets, with experimental ΔG on the x-axis and predicted ΔG on the y-axis. Metrics for Pearson R, Spearman ρ, Kendall τ, and RMSE are displayed in the top-left corner of each scatter plot. Only 20% of the test set (selected randomly) is shown for readability.

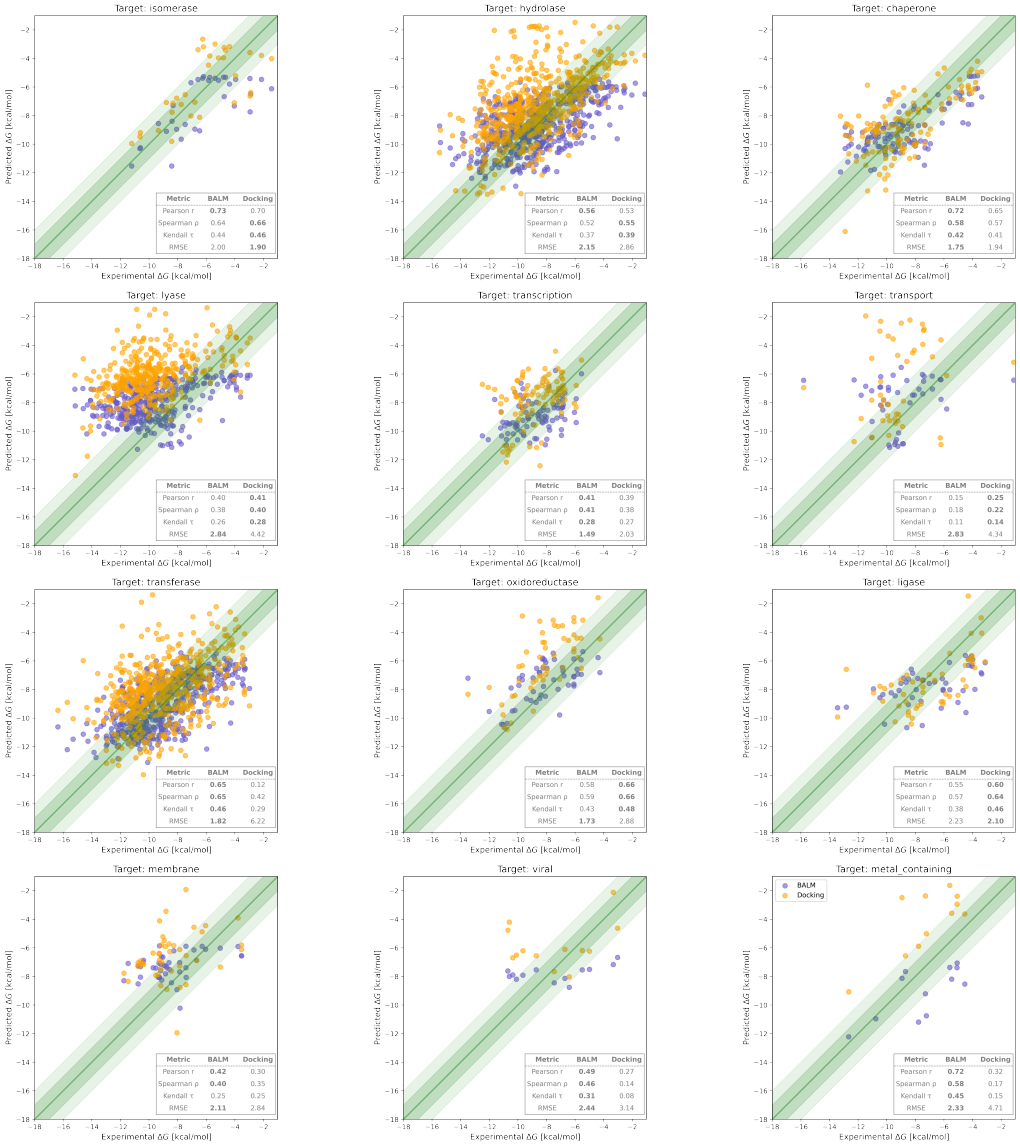


Figure S4: **Comparing BALM's (purple) and Autodock Vina's (orange) performance across different target families in the LP-PDBBind dataset.** Scatter plots show the predicted versus experimental binding affinities (ΔG) for various target types in the zero-shot setting using the BALM+PEFT model. Target families include a wide range of protein classes, such as isomerases, hydrolases, and membrane proteins. Pearson correlation (r), Kendall τ Spearman rank correlation (ρ), and RMSE are shown for each target type.

Figure S5: **Comparing BALM's (purple) and rDock (orange) performance across different target families in the LP-PDBBind dataset.** Scatter plots show the predicted versus experimental binding affinities (ΔG) for various target types in the zero-shot setting using the BALM+PEFT model. Target families include a wide range of protein classes, such as isomerases, hydrolases, and membrane proteins. Pearson correlation (r), Kendall τ Spearman rank correlation (ρ), and RMSE are shown for each target type.

Additional Methods

PEFT techniques

The strategies for fine-tuning the protein and ligand language models used in the study are discussed below.

- **LoRA** introduces $\Delta\mathbf{W} = \mathbf{BA}$, where $\mathbf{B} \in R^{p\times r}$ and $\mathbf{A} \in R^{r\times q}$ are trainable low-rank matrices, with $r \ll \min(p, q)$. This modification is added to the pre-trained weight $\mathbf{W}_0$, yielding an updated forward pass:

$$\mathbf{h}' = \mathbf{W}_0\mathbf{h} + \mathbf{b} + \gamma(\mathbf{BA})\mathbf{h}, \tag{1}$$

  where $\mathbf{h}$ represents the input and $\mathbf{h}'$ represents the output after the forward pass using the updated weights. γ is a scaling factor to control the initialization of $\mathbf{B}$ and $\mathbf{A}$, balancing pre-trained knowledge and new adaptations, with $\gamma = \alpha/r$ for some defined α, and r is the rank of the matrix.

- **LoHa** uses $\Delta\mathbf{W} = (\mathbf{B}_1\mathbf{A}_1) \odot (\mathbf{B}_2\mathbf{A}_2)$, enhancing the rank of the update matrix and thus the fine-tuning capacity without significantly increasing parameter count. The forward pass becomes:

$$\mathbf{h}' = \mathbf{W}_0\mathbf{h} + \mathbf{b} + \gamma[(\mathbf{B}_1\mathbf{A}_1) \odot (\mathbf{B}_2\mathbf{A}_2)]\mathbf{h}, \tag{2}$$

  where $\odot$ denotes the Hadamard (element-wise) product.

- **LoKr** employs $\Delta\mathbf{W} = \mathbf{C} \otimes (\mathbf{BA})$, leveraging the multiplicative rank property of Kronecker products to extend beyond low-rank constraints while maintaining parameter efficiency. The forward pass is updated as:

$$\mathbf{h}' = \mathbf{W}_0\mathbf{h} + \mathbf{b} + \gamma(\mathbf{C} \otimes (\mathbf{BA}))\mathbf{h}, \tag{3}$$

where $\otimes$ denotes the Kronecker product.

- **IA^3** method rescales the inner activations of the self-attention mechanism in transformer layers using learned vectors. IA^3 introduces three learnable rescaling vectors $\mathbf{l}_k \in R^{d_k}$, $\mathbf{l}_v \in R^{d_v}$, and $\mathbf{l}_f \in R^{d_{\text{ff}}}$, applied to key, value, and feed-forward network (FFN) activations, respectively. Unlike LoRA, LoKr, and LoHa, which modify weight matrices, IA^3 directly scales internal activations. The adjusted self-attention mechanism becomes:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left(\frac{\mathbf{Q}(\mathbf{l}_k \odot \mathbf{K}^T)}{\sqrt{d_k}} \right) (\mathbf{l}_v \odot \mathbf{V}), \tag{4}$$

  where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are the query, key, and value matrices, respectively, and d_k is the dimension of the key vectors. In the FFN, the adaptation is represented as:

$$\text{FFN}(\mathbf{h}') = (\mathbf{l}_{\text{ff}} \odot \theta(\mathbf{W}_1 \mathbf{h}))\mathbf{W}_2, \tag{5}$$

  where $\mathbf{l}_{\text{ff}}$ is the learned scaling vector applied to the output of the first FFN layer, $\mathbf{W}_1$ and $\mathbf{W}_2$ are FFN weight matrices, and θ is the FFN activation function.

Fisher-transformed correlations for target-wise evaluation

Cumulative metrics on test sets provide a broad snapshot of overall model performance across all targets, but they can obscure important performance variations at the individual target level. To overcome this, we apply the Fisher z-transformation to both Pearson R and Spearman ρ correlation coefficients to enable more reliable comparisons across individual protein targets. This transformation,

$$z = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right), \tag{6}$$

converts the bounded correlation coefficient $r \in [-1, 1]$ into an unbounded variable z, which is approximately normally distributed for large sample sizes.[1] By stabilizing the variance

and reducing bias inherent in raw correlation values, the Fisher transformation provides a clearer view of target-wise performance, which is critical to understanding the variability across targets in the test set.

Conversion between IC_{50} or K_d and Gibbs Free Energy (ΔG)

The conversion from experimental IC_{50} or dissociation constant (K_d) values to Gibbs binding free energy (ΔG) was carried out using the thermodynamic relationship,

$$\Delta G = RT \ln(K_d \text{ or } IC_{50}), \tag{7}$$

where R is the gas constant ($R = 1.9872041 \times 10^{-3}$ kcal mol^{-1} K^{-1}), and T is the temperature in Kelvin (typically 298.15 K unless otherwise stated). When converting IC_{50} to K_d, it is assumed that IC_{50} approximates K_d under conditions of competitive inhibition with negligible enzyme or receptor concentration compared to IC_{50}. IC_{50} or K_d values provided in micromolar (µM) or nanomolar (nM) units were first converted to molar units (M) before calculating ΔG.

Docking approach for LP-PDBBind

The LP-PDBBind[2] database provided the native receptor conformation and ligand binding poses extracted from their protein data bank (PDB) entries. When multiple alternative conformations were reported in the crystallographic data, the first alternative conformation was selected. The tautomers and protonation states of the protein-ligand complexes were obtained using the Molecular Operating Environment suite v2022-2[3] while maintaining the structural waters and cosolvents. Non-standard residues, isotopes, and cosolvents (such as crystallization buffer molecules like polyethylene glycol, ethanediol, phosphates, etc.) were appropriately removed or replaced with standard elements or residues. Coordination ions within a 6 Å radius of the ligand were retained.

For the evaluation with rDock,[4] a cavity was generated for each system using the reference ligand method. Then, the ligands underwent a single-step simplex minimization, and the `SCORE` and `SCORE.INTER` were reported. For the evaluation with AutoDock Vina v1.2.5,[5] the charges of the receptors and ligands were estimated using the Gasteiger model with MGLtools[6] and then re-scored using Vina within the automatically detected cavity.

References

(1) Fisher, R. Frequency Distribution of the Values of the Correlation Coefficient in Samples from an Indefinitely Large Population. *Biometrika* **1915**, *10*, 507–521.

(2) Li, J.; Guan, X.; Zhang, O.; Sun, K.; Wang, Y.; Bagni, D.; Head-Gordon, T. Leak Proof PDBBind: A Reorganized Dataset of Protein-Ligand Complexes for More Generalizable Binding Affinity Prediction. *arXiv:2308.09639* **2023**,

(3) ULC, C. C. G. Molecular Operating Environment (MOE). version v2022.2, Chemical Computing Group: Montreal, QC, Canada, 2022; `https://www.chemcomp.com`.

(4) Ruiz-Carmona, S.; Alvarez-Garcia, D.; Foloppe, N.; Gago, F.; Gervais, G.; Irwin, J.; Sverrisson, F.; Tounge, B.; Tresadern, G.; Morley, S. rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLoS Comput. Biol.* **2014**, *10*, e1003571.

(5) Eberhardt, J.; Santos-Martins, D.; Tillack, A.; Forli, S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *J. Chem. Inf. Model.* **2021**, *61*, 3891–3898.

(6) Morris, G.; Huey, R.; Lindstrom, W.; Sanner, M.; Belew, R.; Goodsell, D.; Olson, A. Autodock4 and AutoDockTools4: automated docking with selective receptor flexibility. *J. Comput. Chem.* **2009**, *30*, 2785–2791.

8.4 Supplementary Information - Benchmarking Active Learning Protocols for Ligand-Binding Affinity Prediction

Supporting Information

Benchmarking Active Learning Protocols for Ligand Binding Affinity Prediction

Rohan Gorantla,[†,‡,¶,§] Alžbeta Kubincová,[¶,§] Benjamin Suutari,[¶] Benjamin P. Cossins,[¶] and Antonia S. J. S. Mey[*,‡]

†*School of Informatics, University of Edinburgh, EH8 9AB, UK*

‡*EaStCHEM School of Chemistry, University of Edinburgh, EH9 3FJ, UK*

¶*Exscientia, Schrödinger Building, Oxford, OX4 4GE, UK*

§*These authors contributed equally to this work.*

E-mail: antonia.mey@ed.ac.uk

Acronyms

- **AL** - Active learning
- **GP** - Gaussian process
- **CP** - Chemprop
- **UMAP** - Uniform Manifold Approximation and Projection
- **RMSE** - Root Mean Square Error

Datasets

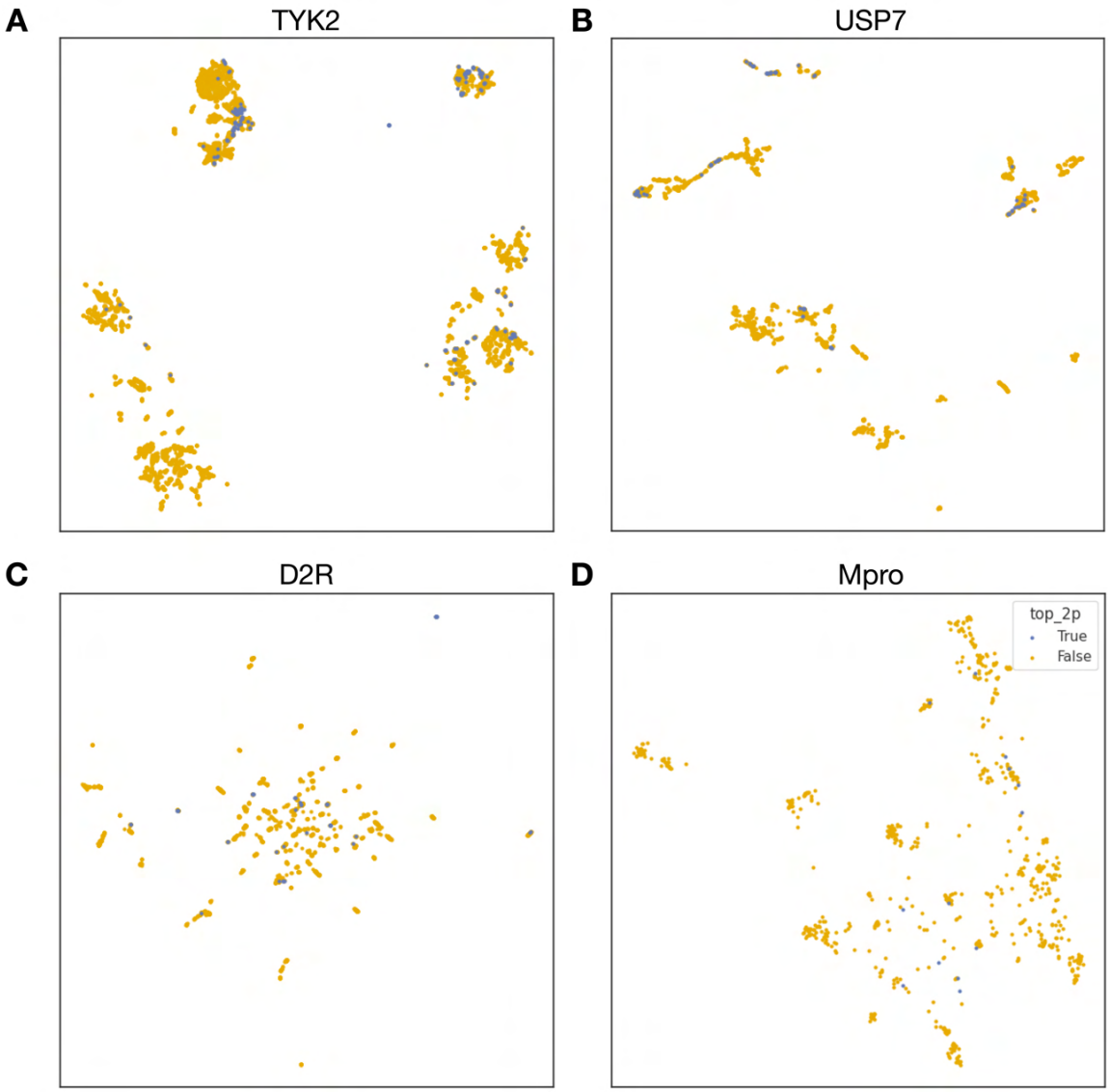


Figure S1: UMAP projection highlighting the top 2% binders in datasets used in our study- **A:** TYK2, **B:** USP7, **C:** D2R, and **D:** Mpro. Blue markers denote the top 2% compounds and the remaining compounds present in the dataset in yellow. We can see that the top 2% compounds for both TYK2 and USP7 targets are present in the dense clusters on the top. In the D2R dataset, the top 2% compounds are present in small clusters, and these small clusters are scattered over the entire chemical space with a few top binders in each. We can see top binders as singletons for Mpro dataset.

Results of AL strategies

Selection of Initial Samples

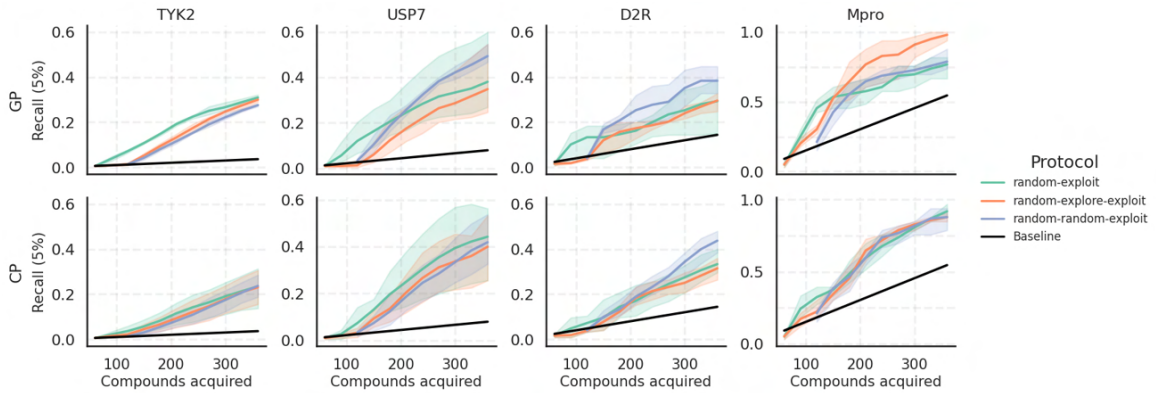


Figure S2: 5% Recall using different AL protocols. Here the compounds acquired shows the number of compounds selected over several AL cycles. The shaded area is variation over 3 AL runs with different seeds, and that baseline is the expectation value given a random selection.

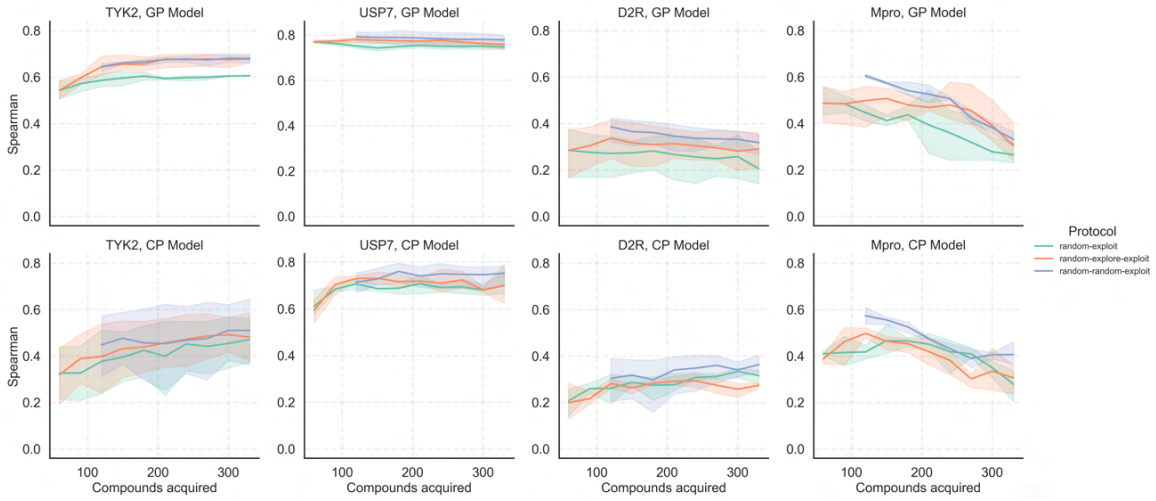


Figure S3: Spearman ρ using different AL protocols on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

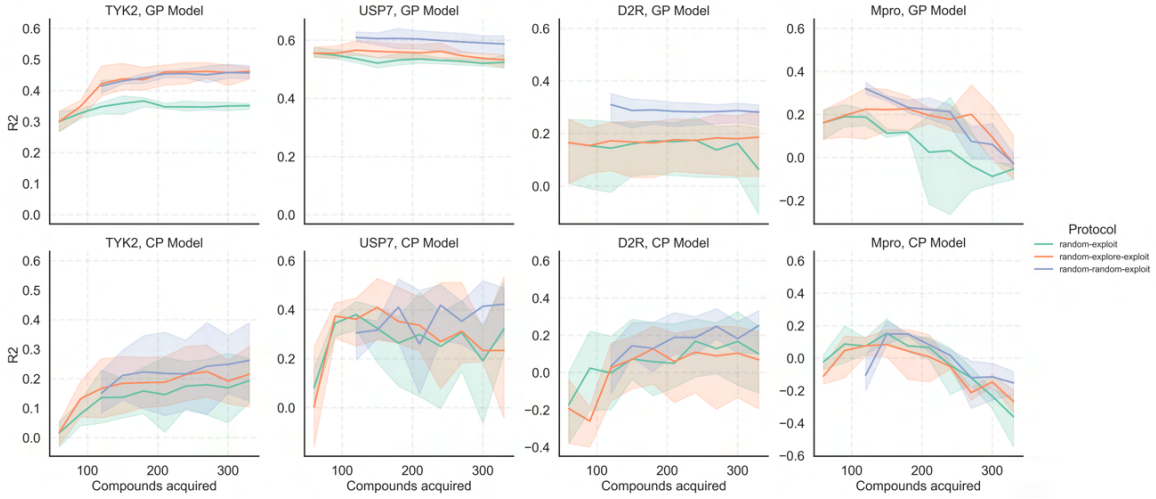


Figure S4: Coefficient of determination, R2 score using different AL protocols on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

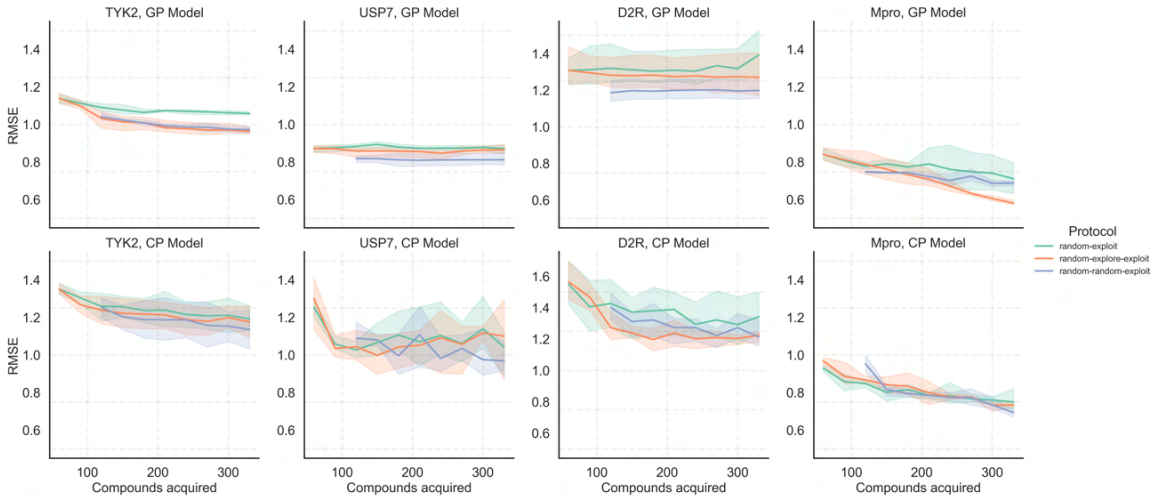


Figure S5: RMSE using different AL protocols on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

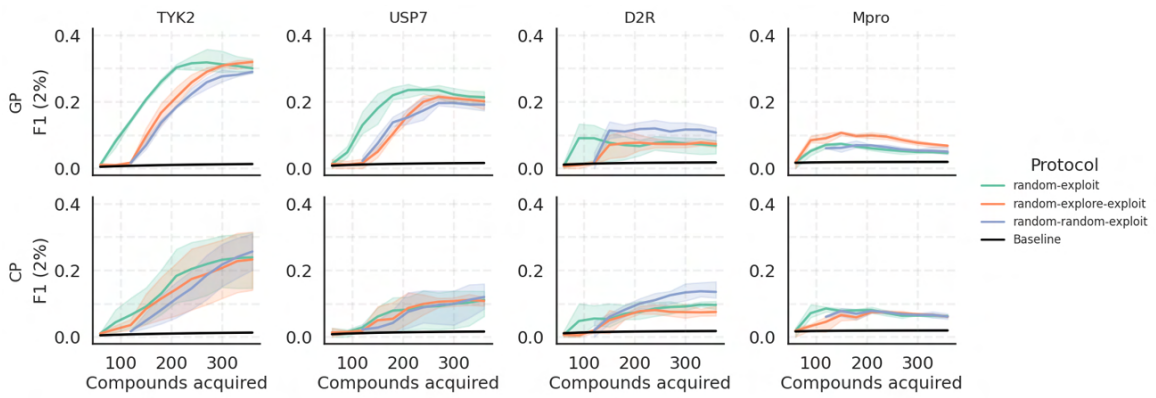


Figure S6: Comparing F1 score for top 2% compounds for GP and CP models using different AL protocols on all four target datasets.

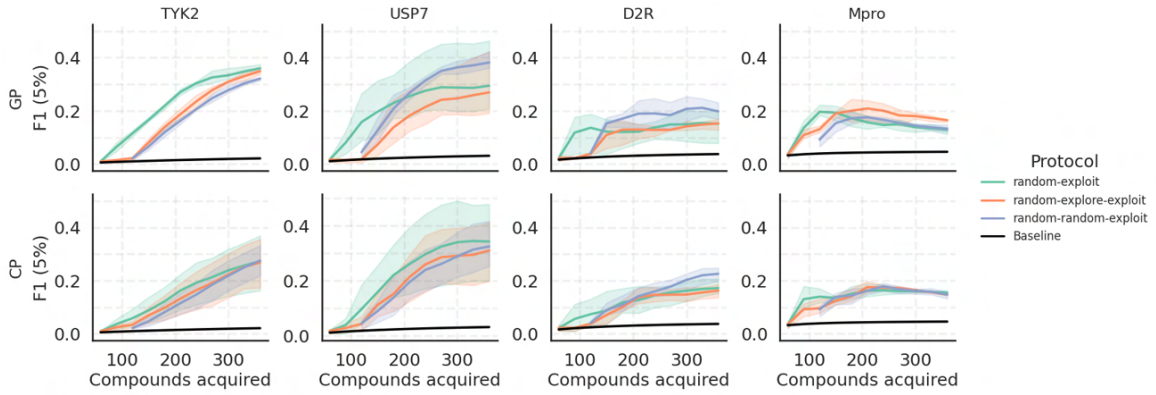


Figure S7: Comparing F1 score for top 5% compounds for GP and CP models using different AL protocols on all four target datasets.

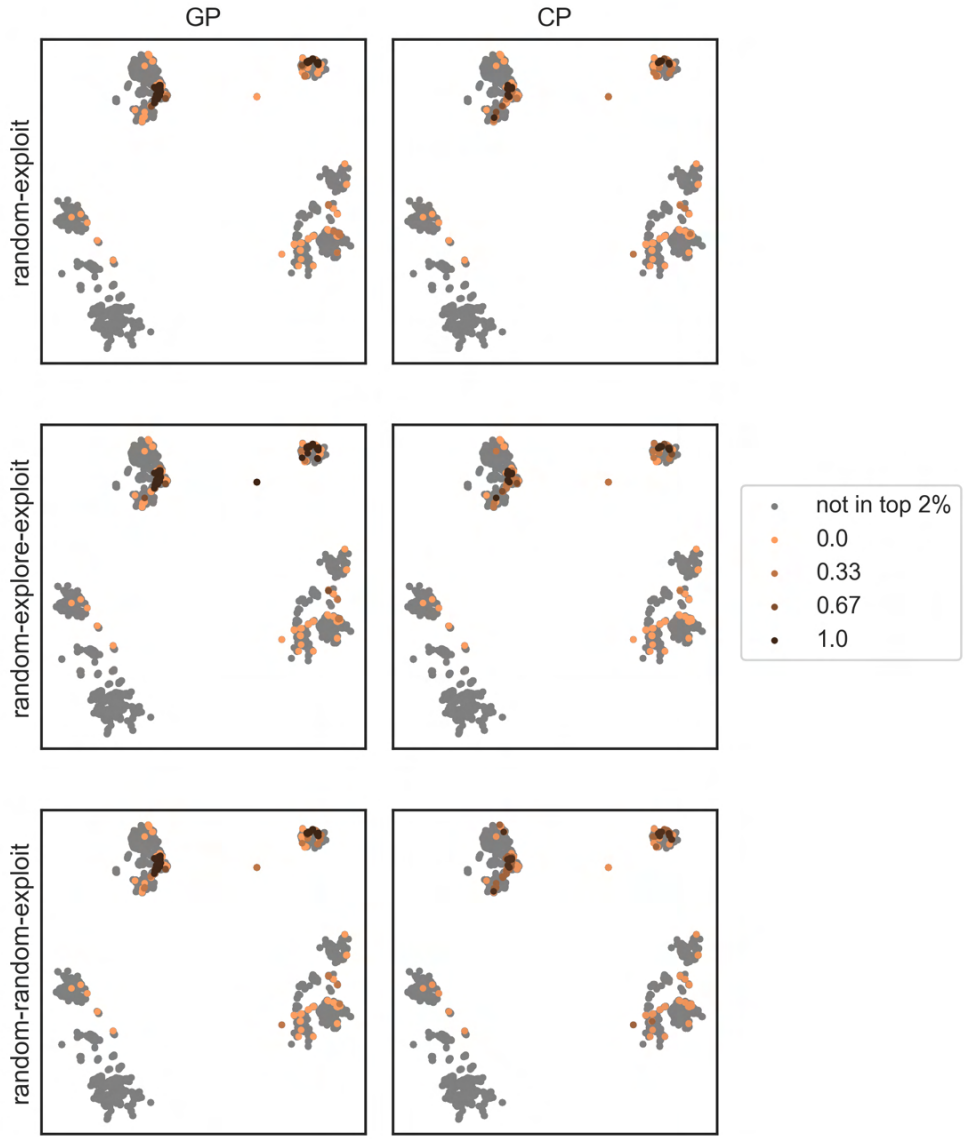


Figure S8: Top 2% TYK2 compounds on UMAP colored by the fraction of the 3 AL runs where they were acquired using GP and CP models.

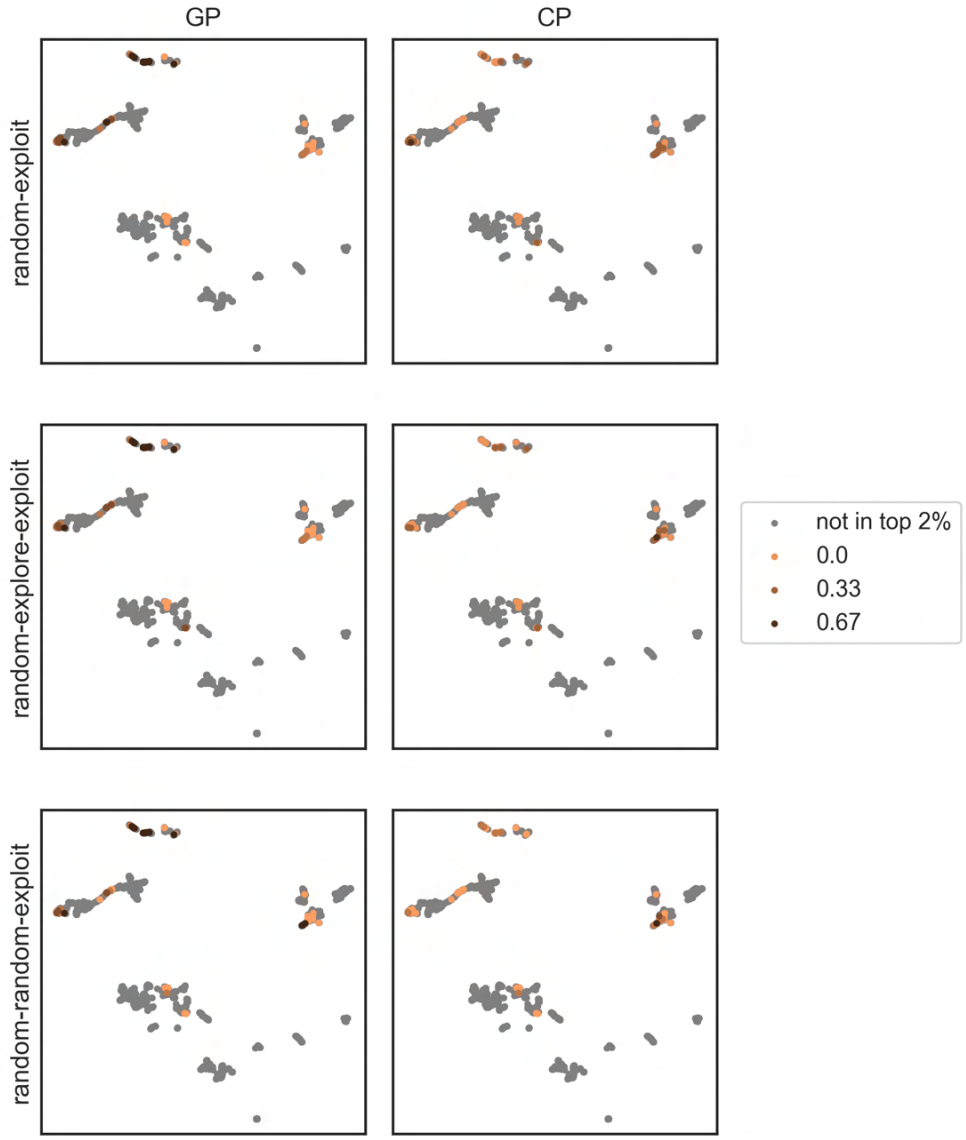


Figure S9: Top 2% USP7 compounds on UMAP colored by the fraction of the 3 AL runs where they were acquired using GP and CP models.

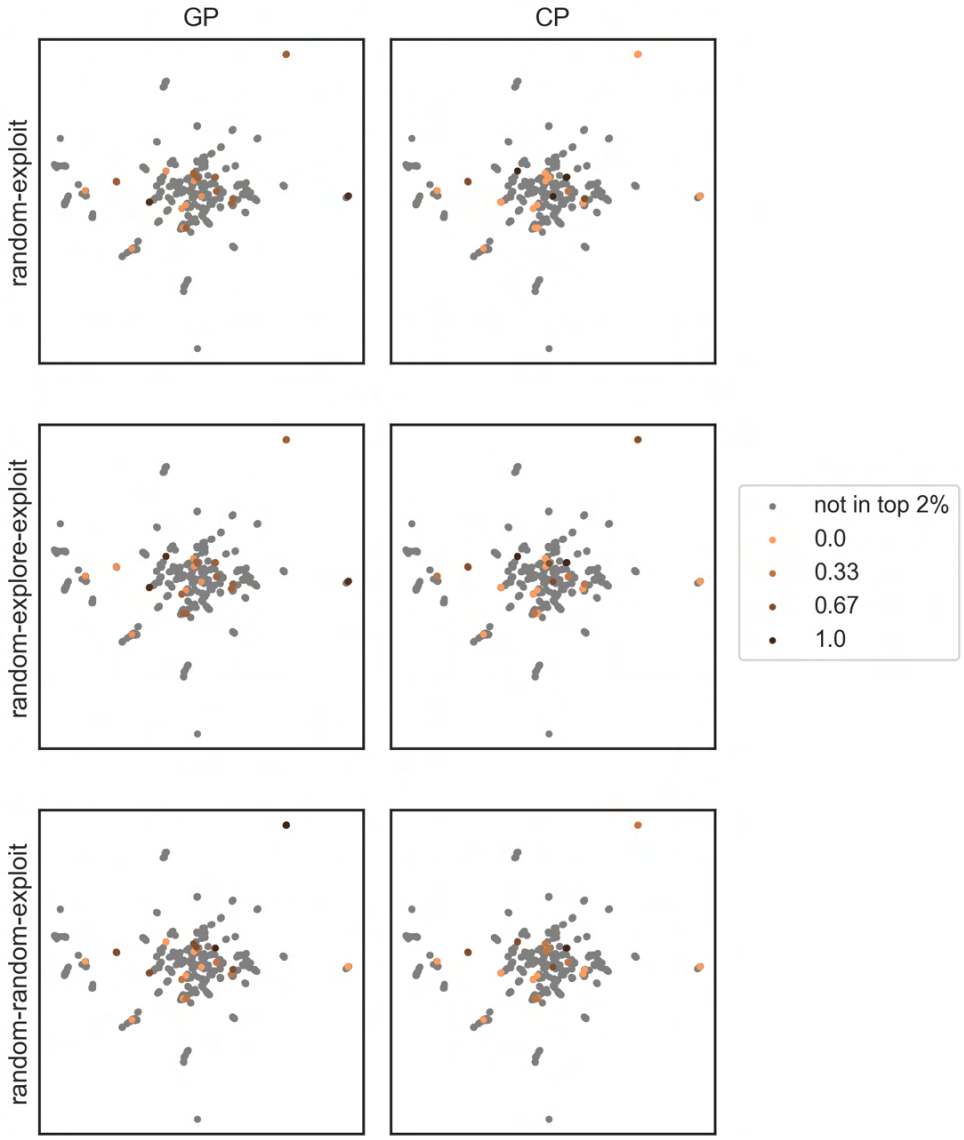


Figure S10: Top 2% D2R compounds on UMAP colored by the fraction of the 3 AL runs where they were acquired using GP and CP models.

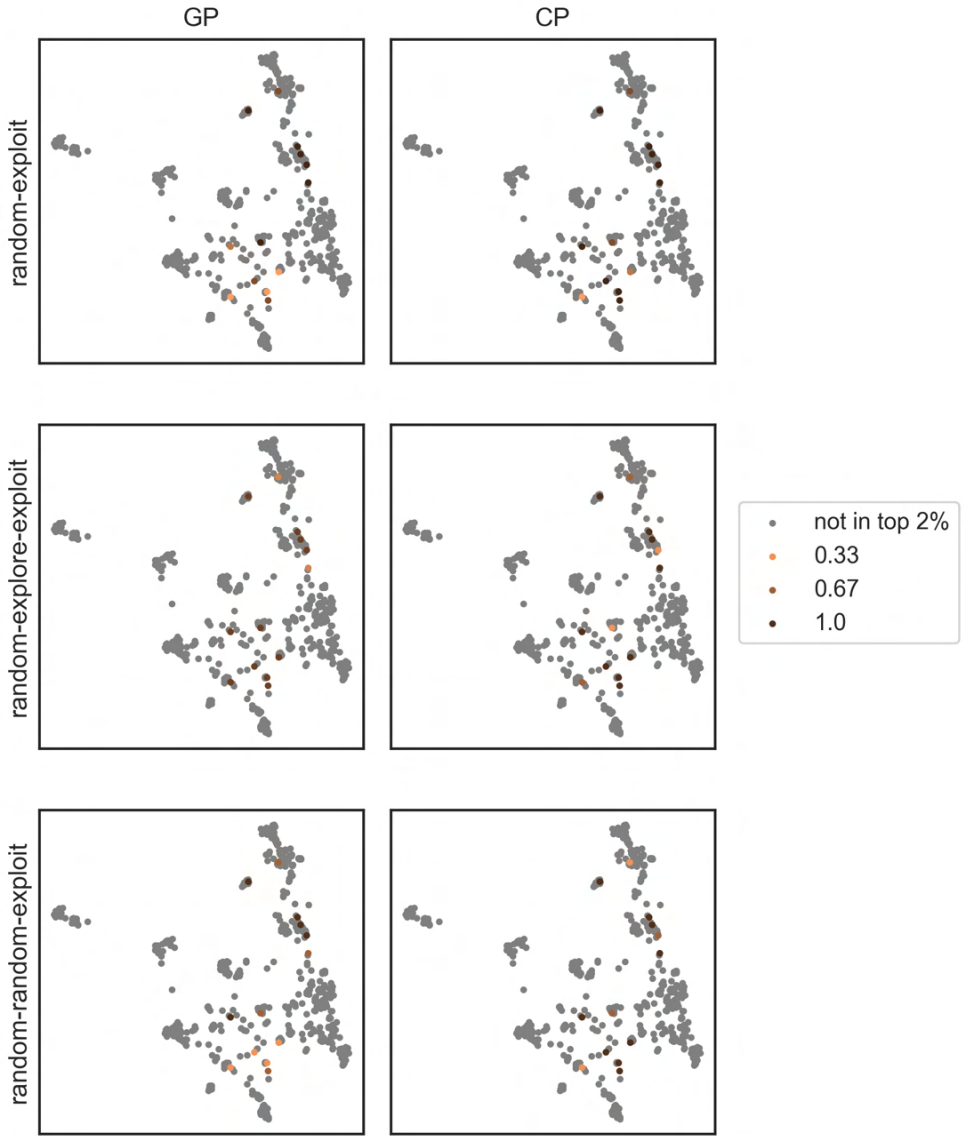


Figure S11: Top 2% Mpro compounds on UMAP colored by the fraction of the 3 AL runs where they were acquired using GP and CP models.

Influence of batch size

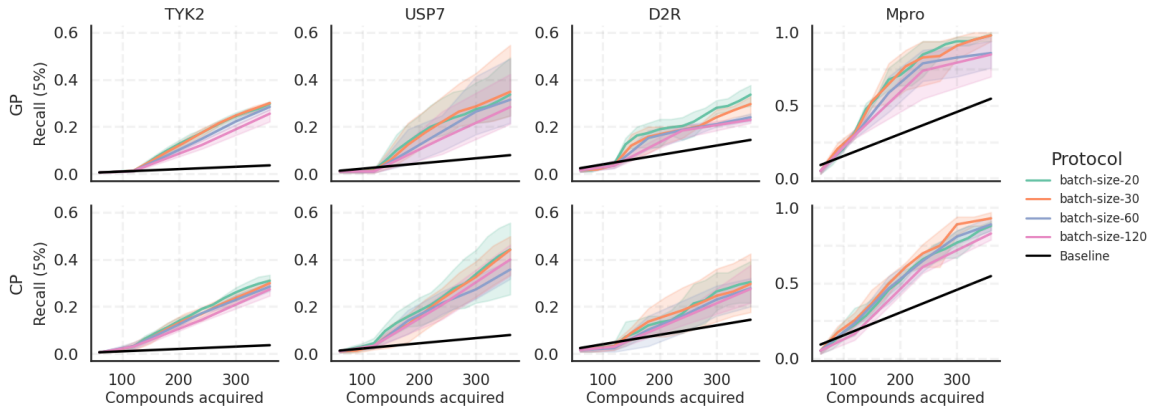


Figure S12: Comparing GP and CP model performance on 5% Recall with different batch sizes. The"batch-size-20" protocol employs three exploration batches and twelve exploitation batches of 20; "batch-size-30" employs two exploration and eight exploitation batches of 30; "batch-size-60" employs a single exploration and four exploitation batches of 60; "batch-size-120" employs one exploration batch of 60 and two exploitation batches of 120. Here, the compounds acquired show the number of compounds selected over several AL cycles. The shaded area is variation over 3 AL runs with different seeds, and that baseline is the expectation value given a random selection.

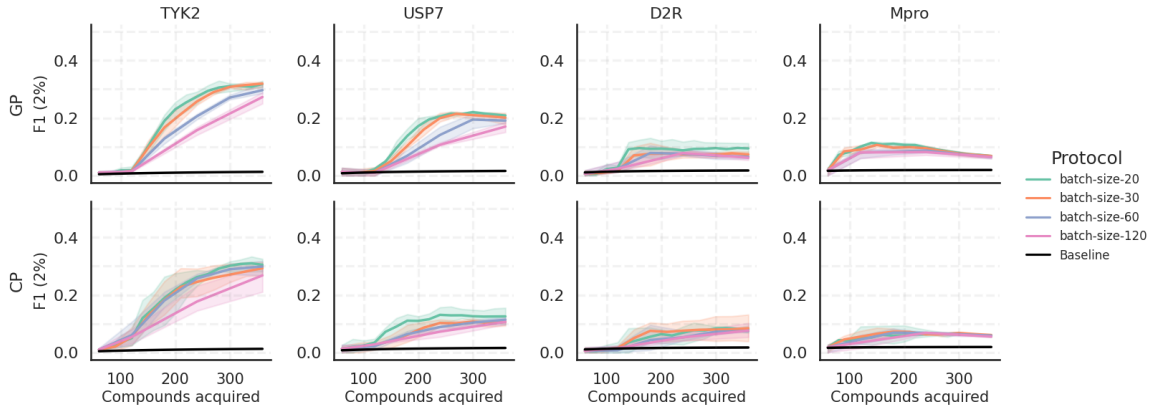


Figure S13: Comparing GP and CP model performance on F1 score (2%) with different batch sizes. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds, and that baseline is the expectation value given a random selection.

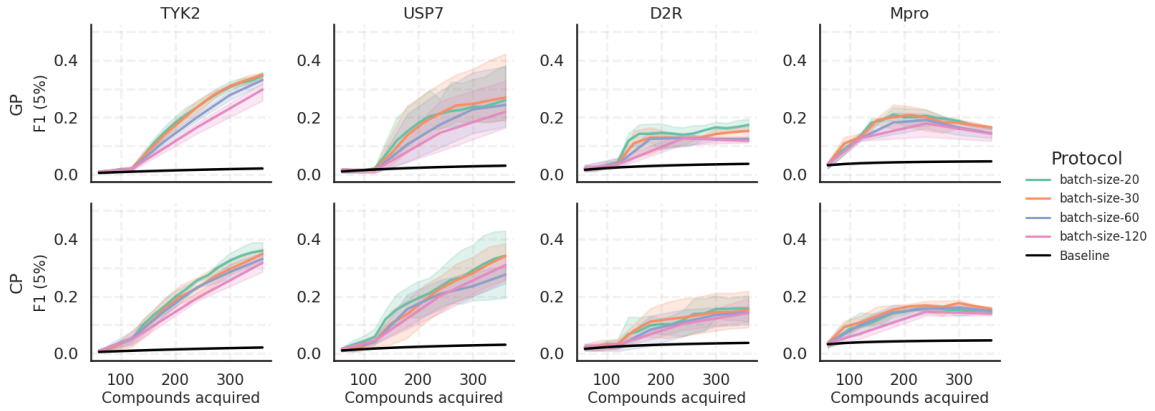


Figure S14: Comparing GP and CP model performance on F1 score (5%) with different batch sizes. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds, and that baseline is the expectation value given a random selection.

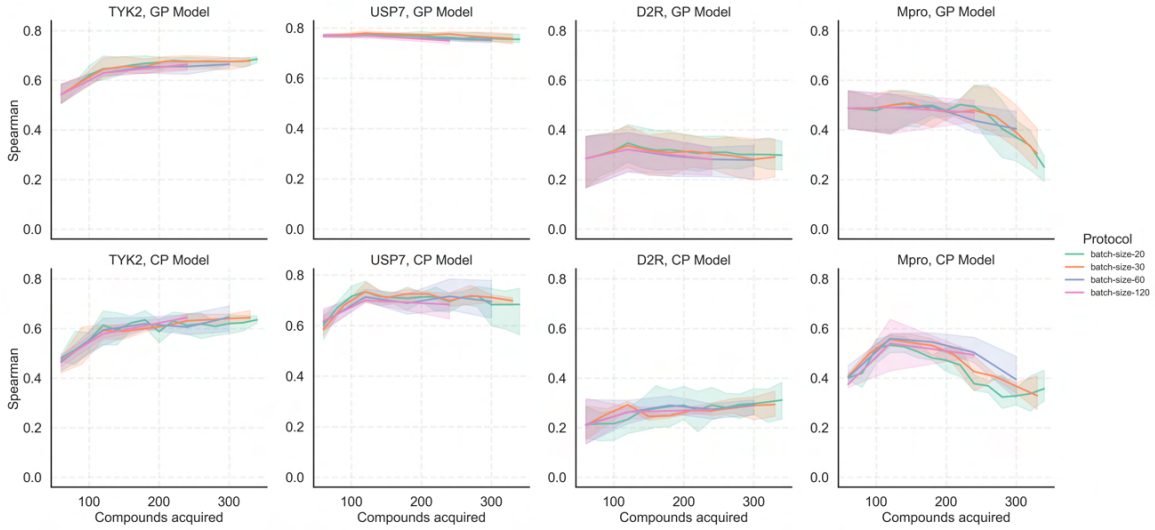


Figure S15: Spearman ρ using different AL protocols with varying batch sizes on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

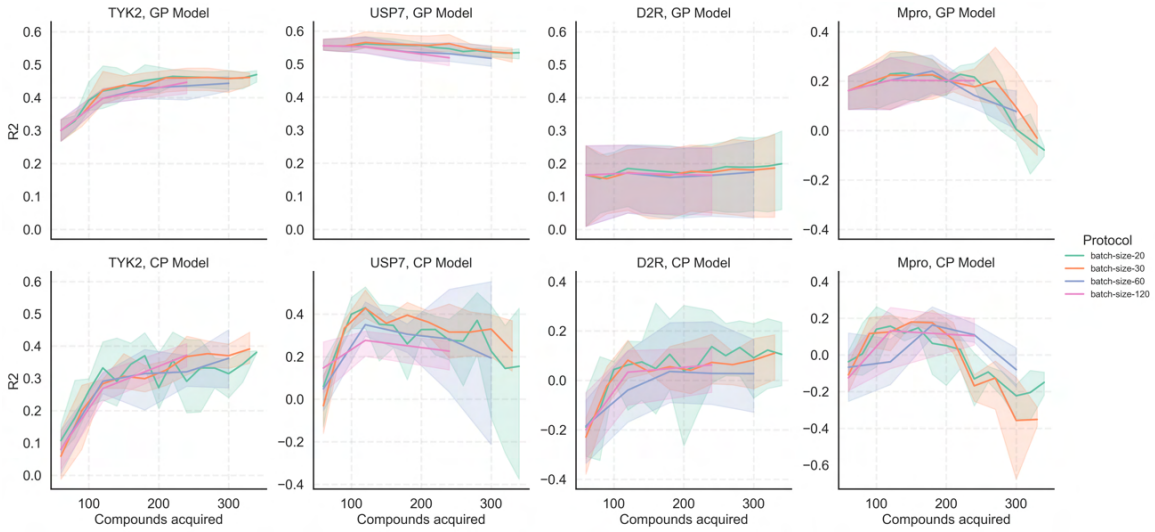


Figure S16: Coefficient of determination, R2 using different AL protocols with varying batch sizes on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

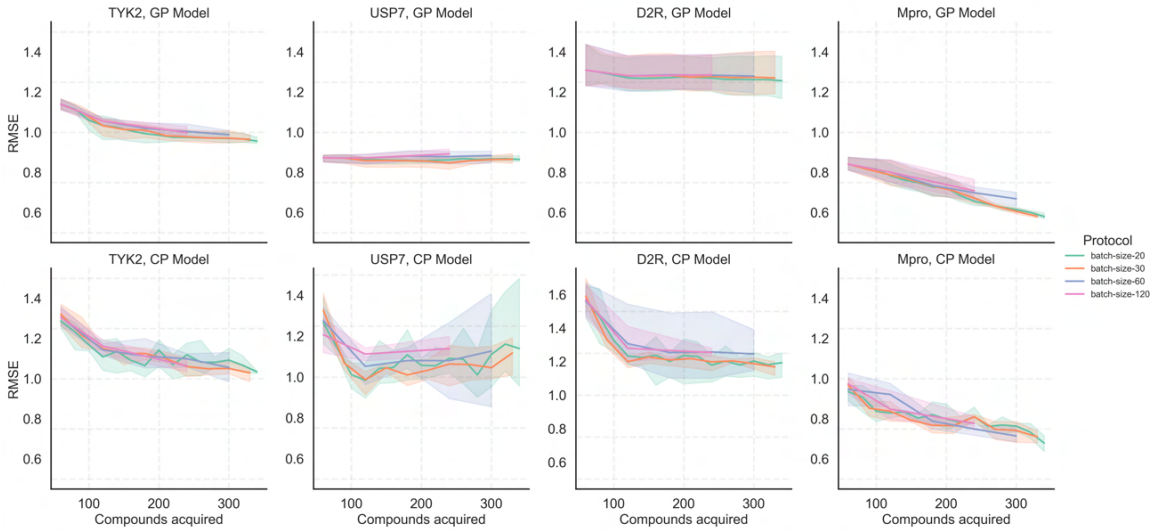


Figure S17: RMSE using different AL protocols with varying batch sizes on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

Modelling noise on labels

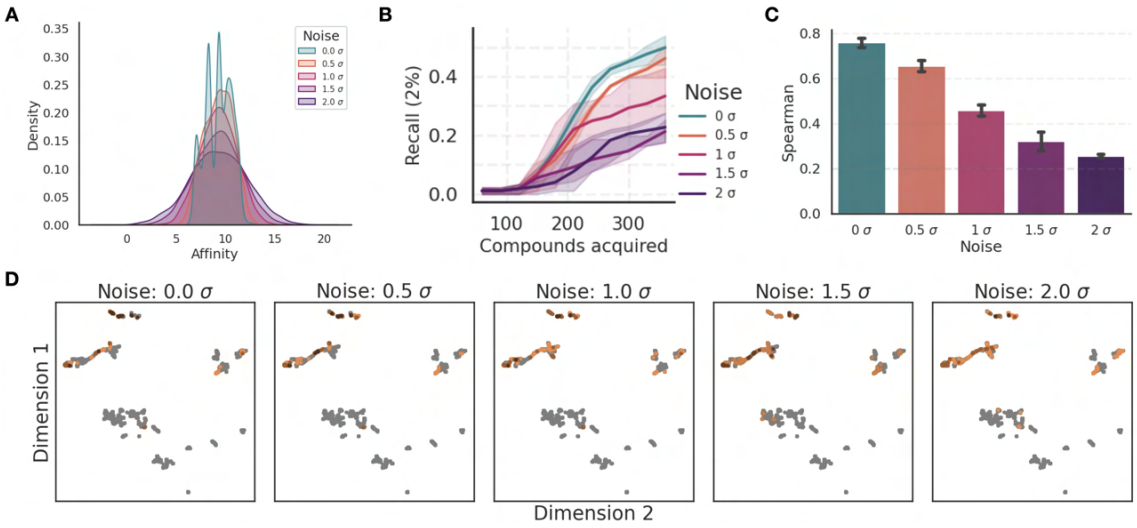


Figure S18: Analysis of the influence of Gaussian noise on the outcomes of AL using the GP model on the USP7 dataset. The standard deviation of the added Gaussian noise was scaled with respect to the standard deviation of USP7 affinities, with factors ranging from 0 (no noise) to 2. **A:** Kernel Density Estimation plot of the affinity score distribution across varying noise magnitudes. **B:** Top 2% Recall, highlighting a noticeable decline at increased noise levels. **C:** Spearman ρ revealing diminished model predictability with increasing noise. **D:** UMAP visualization of the compounds selected in the exploitation phase, colored by the acquisition frequency across three distinct AL iterations with randomized initializations. The UMAPs emphasize the AL framework's capability to consistently identify top-binding compound clusters, even amidst noise interference.

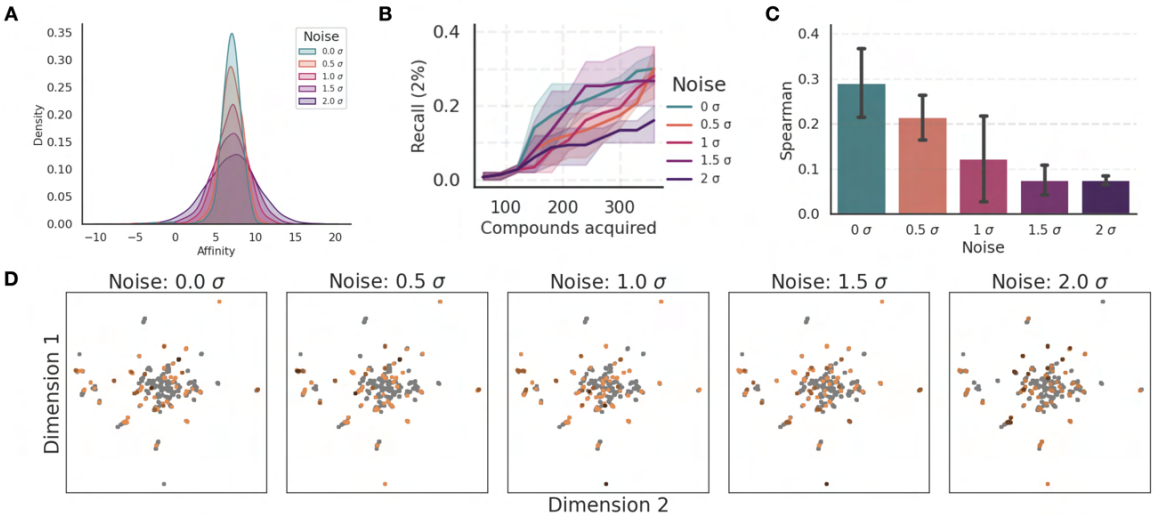


Figure S19: Analysis of the influence of Gaussian noise on the outcomes of AL using the GP model on the D2R dataset. The standard deviation of the added Gaussian noise was scaled with respect to the standard deviation of D2R affinities, with factors ranging from 0 (no noise) to 2. **A:** Kernel Density Estimation plot of the affinity score distribution across varying noise magnitudes. **B:** Top 2% Recall shown at different noise levels. **C:** Spearman ρ shown at different noise levels. **D:** UMAP visualization of the compounds selected in the exploitation phase, colored by the acquisition frequency across three distinct AL iterations with randomized initializations.

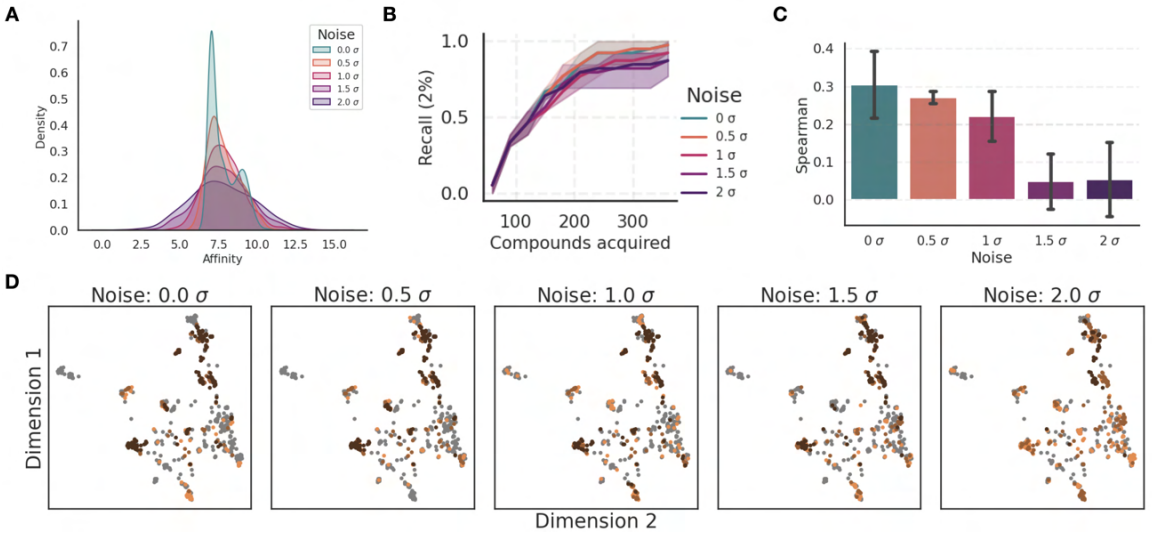


Figure S20: Analysis of the influence of Gaussian noise on the outcomes of AL using the GP model on the Mpro dataset. The standard deviation of the added Gaussian noise was scaled with respect to the standard deviation of Mpro affinities, with factors ranging from 0 (no noise) to 2. **A:** Kernel Density Estimation plot of the affinity score distribution across varying noise magnitudes. **B:** Top 2% Recall shown at different noise levels. **C:** Spearman ρ shown at different noise levels. **D:** UMAP visualization of the compounds selected in the exploitation phase, colored by the acquisition frequency across three distinct AL iterations with randomized initializations.

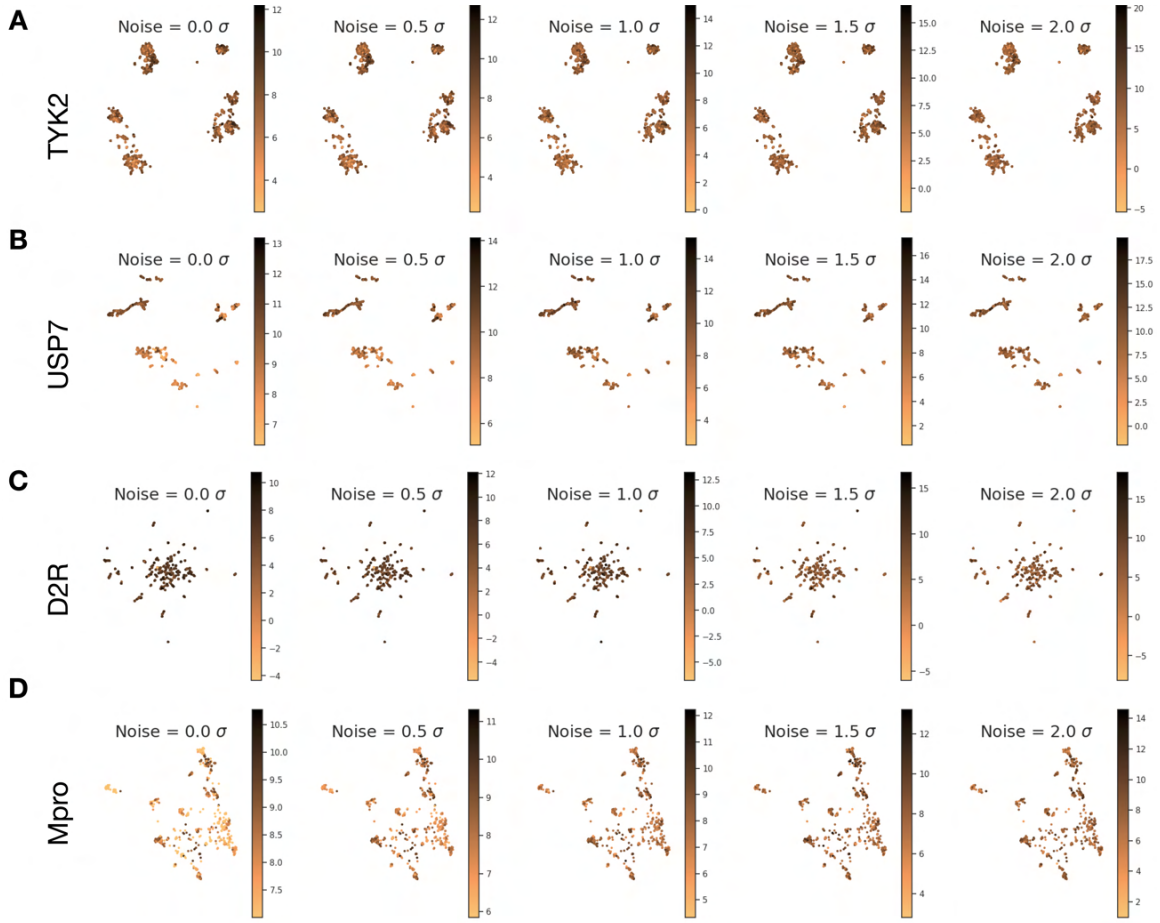


Figure S21: UMAPs showing the chemical space changes at different noise levels on for four datasets used in our study- **A:** TYK2, **B:** USP7, **C:** D2R, and **D:** Mpro. UMAP visualization of the compounds colored by the potency label at different noise levels.

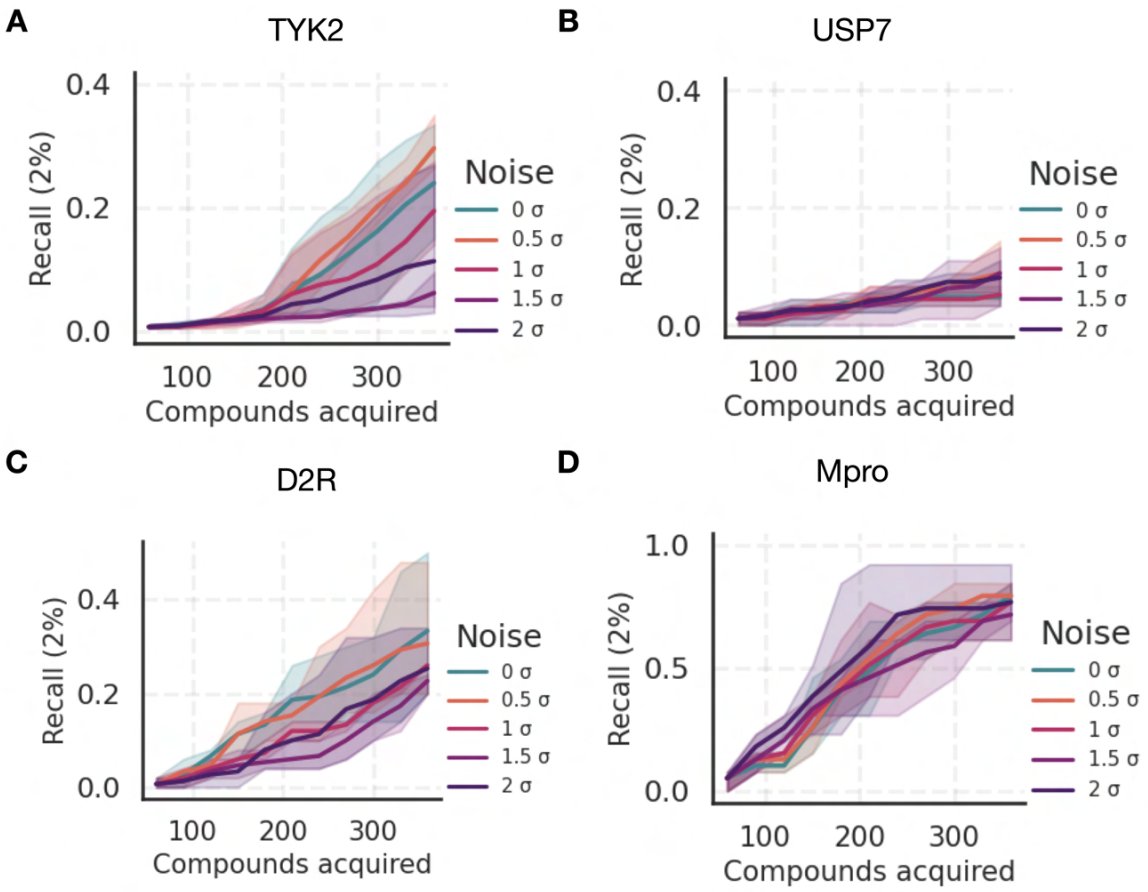


Figure S22: Top 2% Recall with CP models at different noise levels on for four datasets used in our study- **A:** TYK2, **B:** USP7, **C:** D2R, and **D:** Mpro.

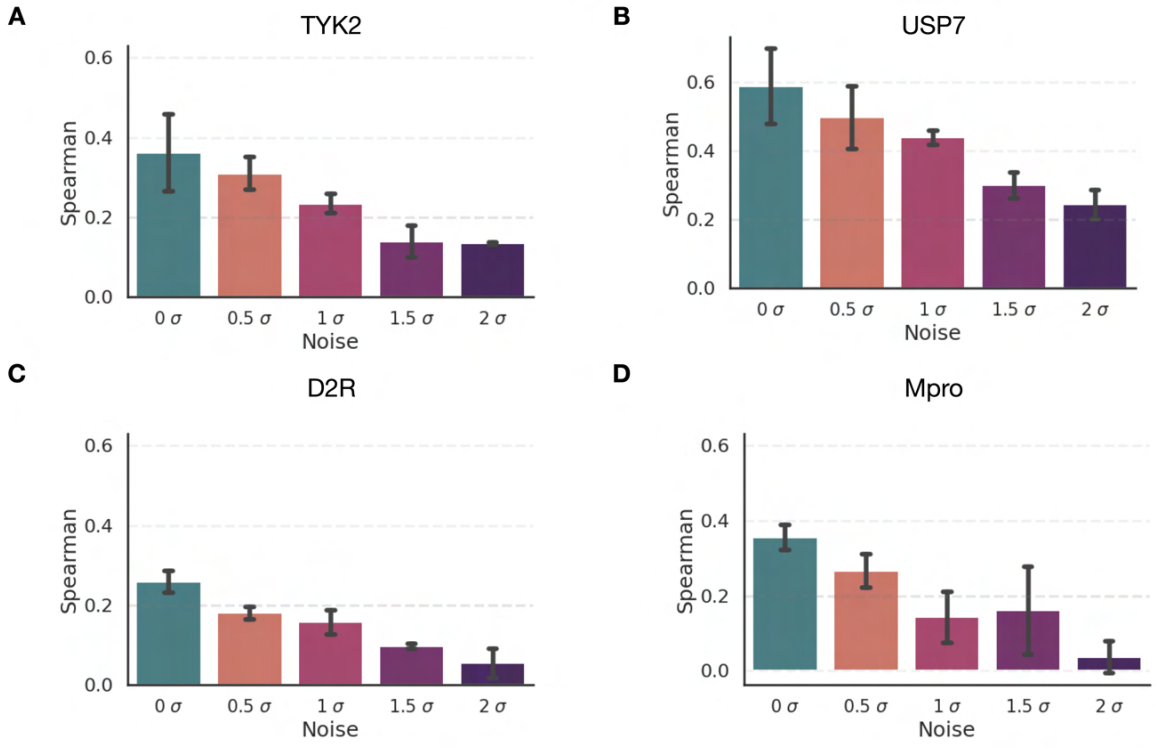


Figure S23: Spearman ρ for CP models at different noise levels on for four datasets used in our study- **A:** TYK2, **B:** USP7, **C:** D2R, and **D:** Mpro.

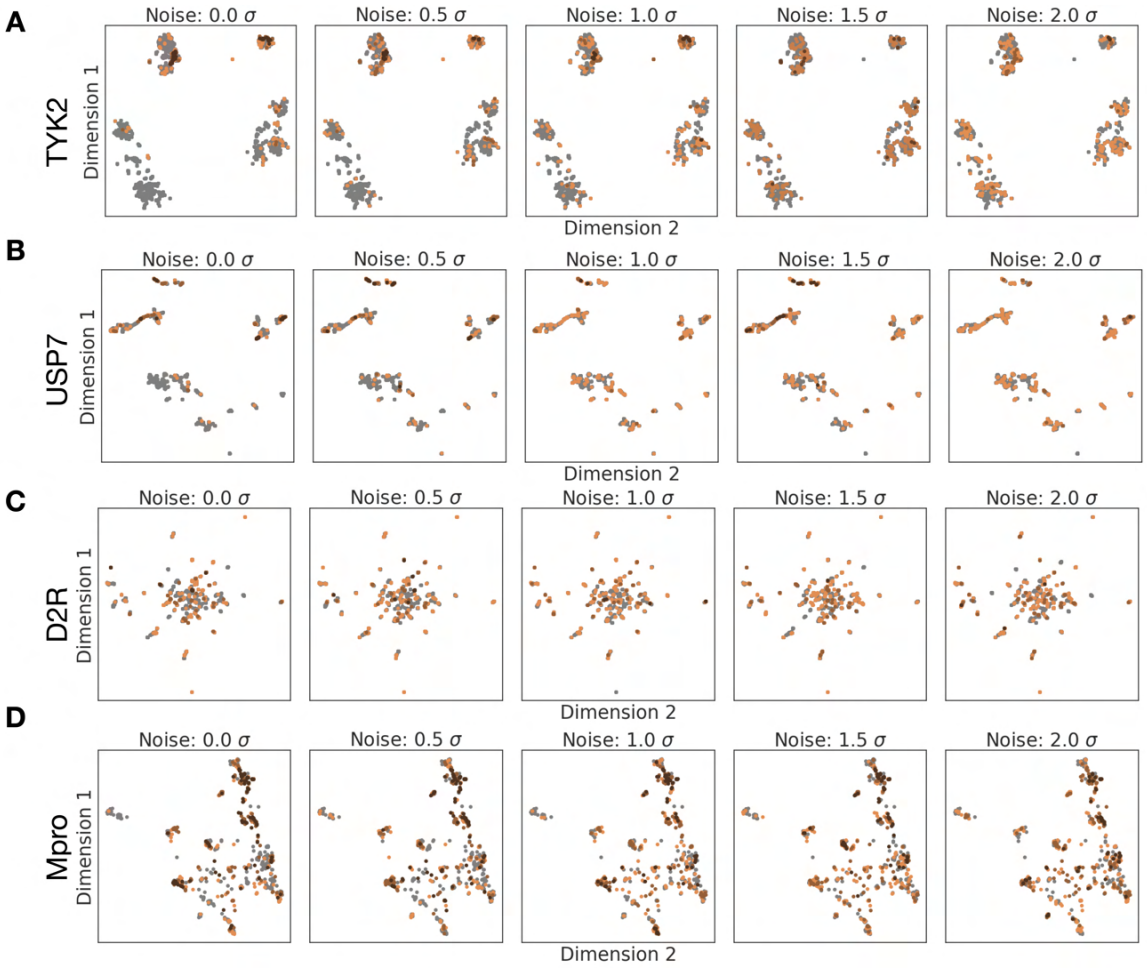


Figure S24: UMAPs using the CP model showing the chemical space changes at different noise levels on for four datasets used in our study- **A:** TYK2, **B:** USP7, **C:** D2R, and **D:** Mpro. UMAP visualization of the compounds selected in the exploitation phase, colored by the acquisition frequency across three distinct AL iterations with randomized initializations.

Selection of Initial Samples (on Training set)

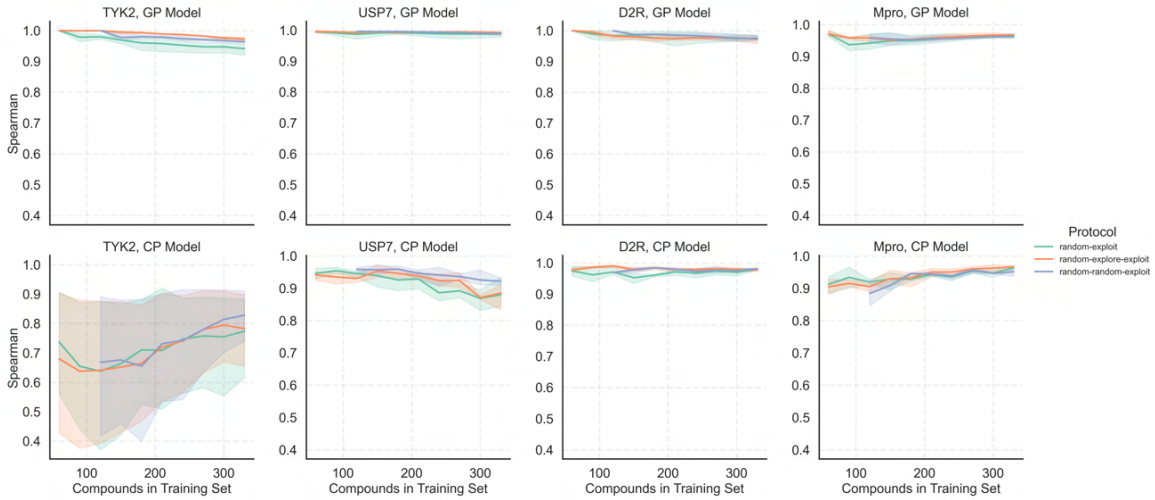


Figure S25: Spearman ρ using different AL protocols on the training set (size equal to Compounds acquired) on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

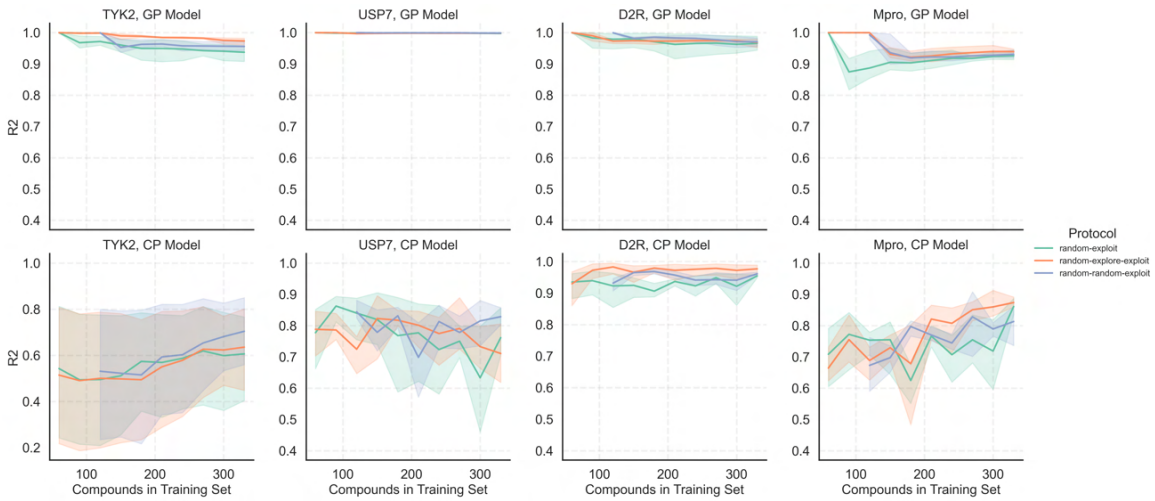


Figure S26: R2 on the training set (size equal to Compounds acquired) using different AL protocols on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

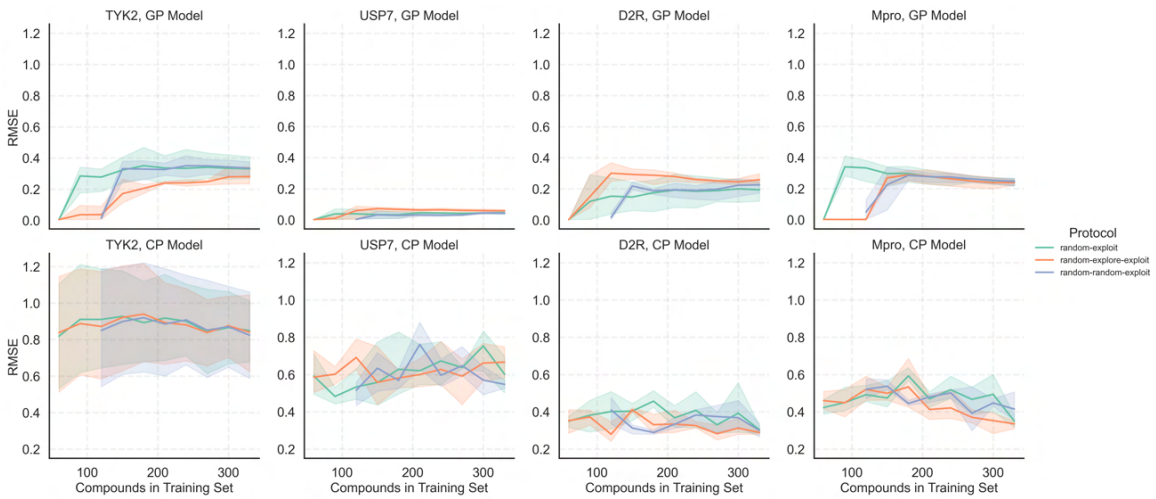


Figure S27: RMSE on the training set (size equal to Compounds acquired) using different AL protocols on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

Influence of batch size (on Training set)

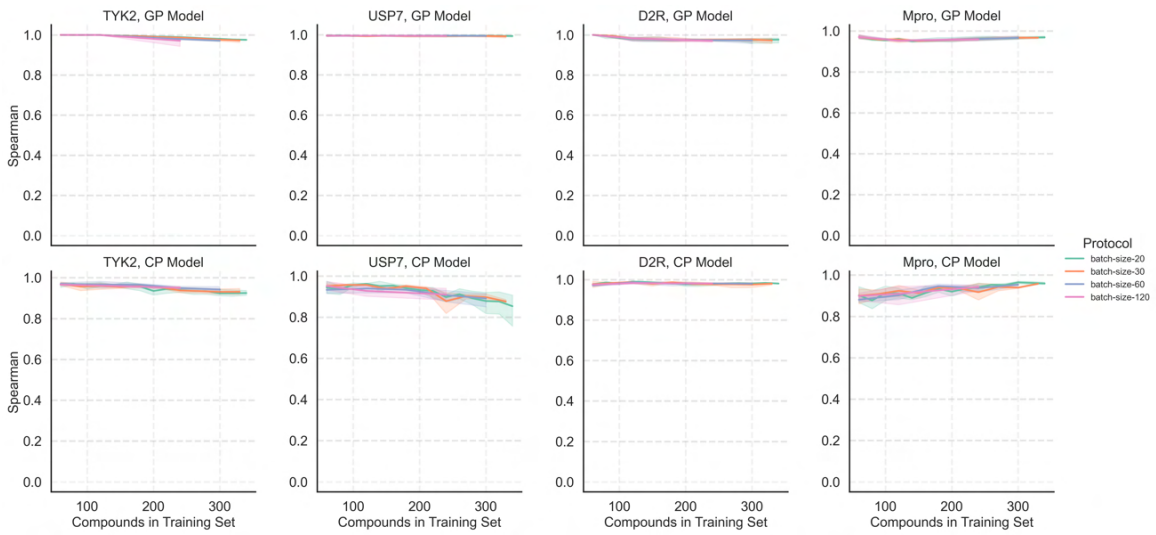


Figure S28: Spearman ρ on the training set (size equal to Compounds acquired) using different AL protocols with varying batch sizes on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

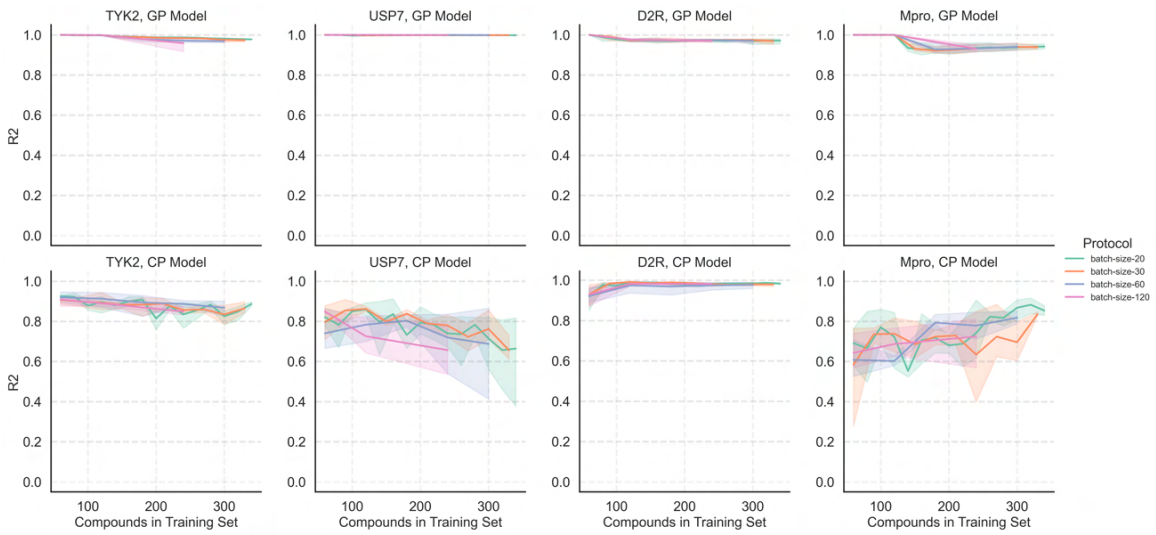


Figure S29: R2 on the training set (size equal to Compounds acquired) using different AL protocols with varying batch sizes on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

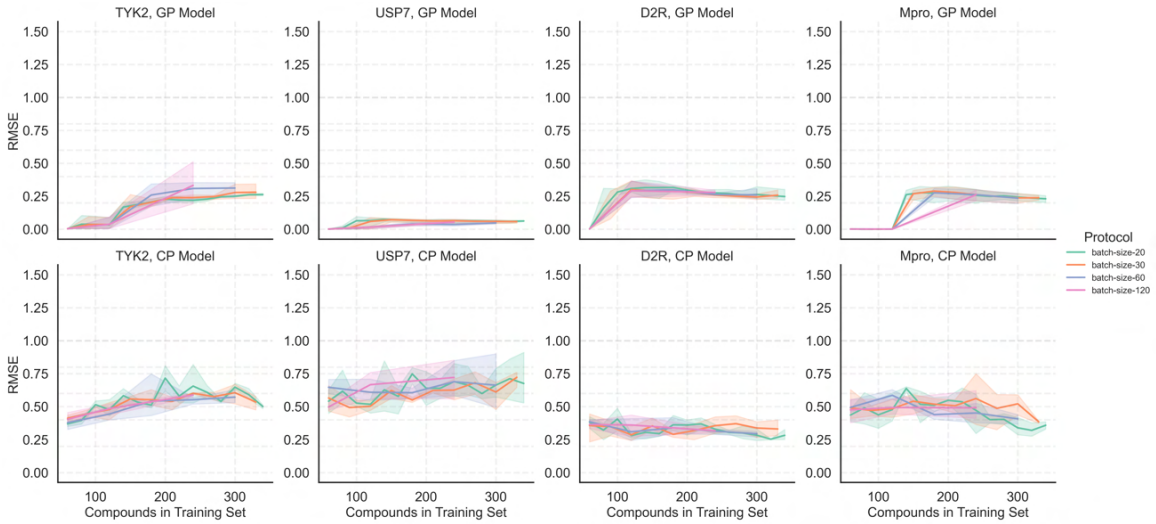


Figure S30: RMSE on the training set (size equal to Compounds acquired) using different AL protocols with varying batch sizes on all four target datasets with GP and CP models. Compounds acquired are cumulative over AL cycles. The shaded area is variation over 3 AL runs with different seeds.

Dataset Clustering

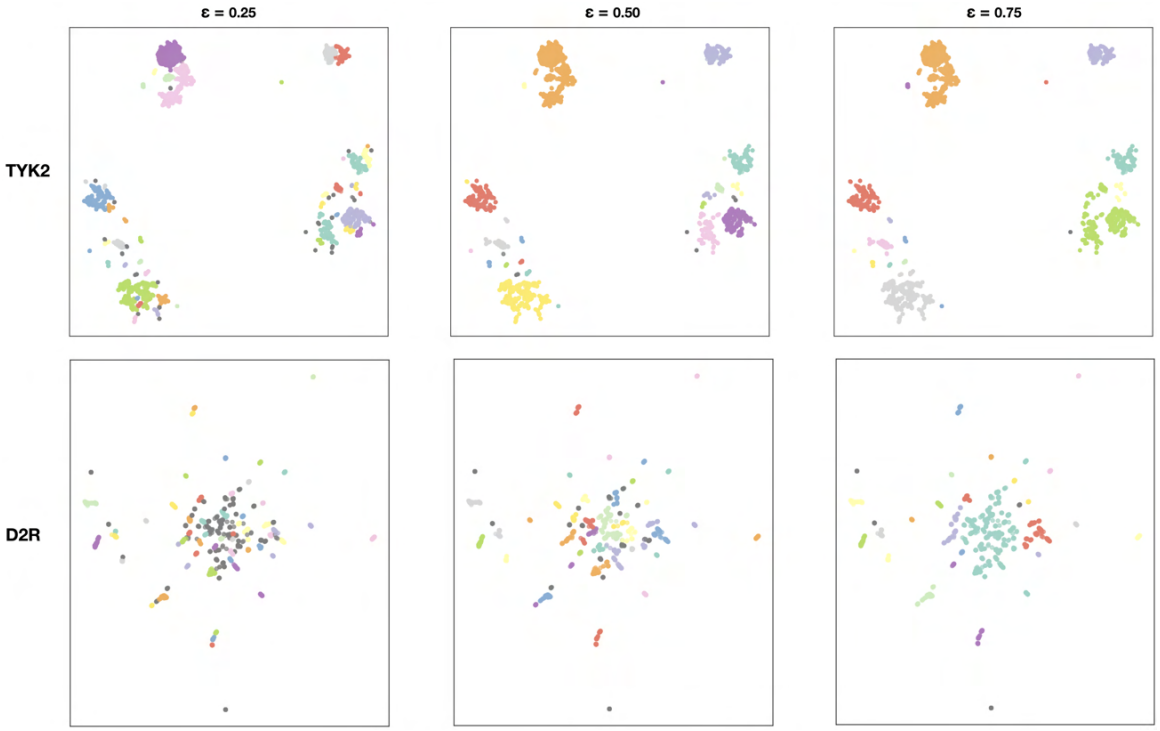


Figure S31: UMAP visualizations illustrating the clusters formed by varying DBSCAN epsilon values on the chemical space of TYK2 and D2R datasets. Each row showcases UMAPs generated using different epsilon values ($\epsilon = 0.25$, $\epsilon = 0.5$, and $\epsilon = 0.75$) with a minimum sample size of 20 for DBSCAN clustering. In the top row we have TYK2 dataset UMAPs colored according to cluster assignments at each epsilon value, and in the bottom row we can observe D2R dataset UMAPs, similarly colored by clustering results.

Bibliography

[1] J. A. DiMasi, H. G. Grabowski and R. W. Hansen, *J. Health Econ.*, 2016, **47**, 20–33.

[2] S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg and A. L. Schacht, *Nat. Rev. Drug Discov.*, 2010, **9**, 203–214.

[3] M. Hay, D. W. Thomas, J. L. Craighead, C. Economides and J. Rosenthal, *Nat. Biotechnol.*, 2014, **32**, 40–51.

[4] C. H. Wong, K. C. Siah and A. W. Lo, *Biostatistics*, 2019, **20**, 273–286.

[5] D. B. Catacutan, J. Alexander, A. Arnold and J. M. Stokes, *Nature Chemical Biology*, 2024, 1–14.

[6] A. Mullard, *2023 FDA Approvals*, 2024, `https://doi.org/10.1038/d41573-024-00001-x`, Accessed October 2, 2024.

[7] J. Myers and J. Baker, *Nat. Biotechnol.*, 2001, **19**, 727–730.

[8] P. Morgan, D. G. Brown, S. Lennard, M. J. Anderton, J. C. Barrett, U. Eriksson, M. Fidock, B. Hamrén, A. Johnson, R. E. March *et al.*, *Nat. Rev. Drug Discov.*, 2018, **17**, 167–181.

[9] N. Brown, *Artificial Intelligence in Drug Discovery*, Royal Society of Chemistry, Cambridge, UK, 2020.

[10] S. Barnett and J. D. Chodera, *GEN Biotechnol.*, 2024, **3**, 119–129.

[11] G. J. Weiner, *Nat. Rev. Cancer*, 2015, **15**, 361–370.

[12] N. Pardi, M. J. Hogan, F. W. Porter and D. Weissman, *Nat. Rev. Drug Discov.*, 2018, **17**, 261–279.

[13] T. Friedmann and R. Roblin, *Science*, 2000, **287**, 1957–1958.

[14] T. Langbein, W. A. Weber and M. Beheshti, *Hell. J. Nucl. Med.*, 2014, **17**, 242–248.

[15] S. L. Ginn, A. K. Amaya, I. E. Alexander, M. L. Edelstein and M. R. Abedi, *J. Gene Med.*, 2018, **20**, e3015.

[16] M. Lemurell, *A big future for small molecules: Targeting the undruggable*, `https://www.astrazeneca.com/r-d/next-generation-therapeutics/small-molecule.html`, 2022, Accessed: August 30, 2024.

[17] C. A. Lipinski, F. Lombardo, B. W. Dominy and P. J. Feeney, *Adv. Drug Deliv. Rev.*, 1997, **23**, 3–25.

[18] D. C. Swinney and J. Anthony, *Nat. Rev. Drug Discov.*, 2011, **10**, 507–519.

[19] A. L. Hopkins, *Nat. Chem. Biol.*, 2008, **4**, 682–690.

[20] P. Imming, C. Sinning and A. Meyer, *Nat. Rev. Drug Discov.*, 2006, **5**, 821–834.

[21] J. P. Hughes, S. Rees, S. B. Kalindjian and K. L. Philpott, *Br. J. Pharmacol.*, 2011, **162**, 1239–1249.

[22] J.-L. Reymond and M. Awale, *ACS Chem. Neurosci.*, 2012, **3**, 649–657.

[23] J. D. Durrant and J. A. McCammon, *BMC Biol.*, 2011, **9**, 1–9.

[24] A. C. Anderson, *Chem. Biol.*, 2003, **10**, 787–797.

[25] M. Levitt and A. Warshel, *Nature*, 1975, **253**, 694–698.

[26] M. Levitt, *Nat. Struct. Mol. Biol.*, 2001, **8**, 392–393.

[27] J. A. McCammon, B. R. Gelin and M. Karplus, *Nature*, 1977, **267**, 585–590.

[28] I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge and T. E. Ferrin, *J. Mol. Biol.*, 1982, **161**, 269–288.

[29] D. S. Goodsell and A. J. Olson, *Proteins*, 1990, **8**, 195–202.

[30] G. Jones, P. Willett, R. C. Glen, A. R. Leach and R. Taylor, *J. Mol. Biol.*, 1997, **267**, 727–748.

[31] H.-J. Böhm, *J. Comput. Aided Mol. Des.*, 1992, **6**, 61–78.

[32] V. Gillet, A. P. Johnson, P. Mata, S. Sike and P. Williams, *J. Comput. Aided Mol. Des.*, 1993, **7**, 127–153.

[33] G. Sliwoski, S. Kothiwale, J. Meiler and E. W. Lowe, *Pharmacol. Rev.*, 2014, **66**, 334–395.

[34] T. Fujita, J. Iwasa and C. Hansch, *J. Am. Chem. Soc.*, 1964, **86**, 5175–5180.

[35] Y. C. Martin, M. G. Bures, E. A. Danaher, J. DeLazzer, I. Lico and P. A. Pavlik, *J. Comput. Aided Mol. Des.*, 1993, **7**, 83–102.

[36] G. Jones, P. Willett and R. C. Glen, *J. Comput. Aided Mol. Des.*, 1995, **9**, 532–549.

[37] Y. Patel, V. J. Gillet, G. Bravi and A. R. Leach, *J. Comput. Aided Mol. Des.*, 2002, **16**, 653–681.

[38] A. Wlodawer and J. Vondrasek, *Annu. Rev. Biophys. Biomol. Struct.*, 1998, **27**, 249–284.

[39] F. Sohraby and H. Aryapour, Semin. Cancer Biol., 2021, pp. 249–257.

[40] T. Aoyama and H. Ichikawa, *J. Chem. Inf. Comput.*, 1992, **32**, 492–500.

[41] H. Liu, R. Zhang, X. Yao, M. Liu, Z. Hu and B. T. Fan, *J. Chem. Inf. Comput.*, 2003, **43**, 1288–1296.

[42] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan and B. P. Feuston, *J. Chem. Inf. Comput.*, 2003, **43**, 1947–1958.

[43] A. Cherkasov, E. N. Muratov, D. Fourches *et al.*, *J. Med. Chem.*, 2014, **57**, 4977–5010.

[44] T. B. Kimber, Y. Chen and A. Volkamer, *Int. J. Mol. Sci.*, 2021, **22**, 4435.

[45] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, *Nature*, 2021, **596**, 583–589.

[46] K. Swanson, P. Walther, J. Leitz, S. Mukherjee, J. C. Wu, R. V. Shivnaraine and J. Zou, *Bioinformatics*, 2024, **40**, btae416.

[47] X. Tang, H. Dai, E. Knight, F. Wu, Y. Li, T. Li and M. Gerstein, *Brief. Bioinform.*, 2024, **25**, year.

[48] H. H. Loeffler, J. He, A. Tibo, J. P. Janet, A. Voronov, L. H. Mervin and O. Engkvist, *J. Cheminform.*, 2024, **16**, 20.

[49] N. S. Pagadala, K. Syed and J. Tuszynski, *Biophys. Rev.*, 2017, **9**, 91–102.

[50] D. F. Hahn, C. I. Bayly, M. L. Boby, H. E. B. Macdonald, J. D. Chodera, V. Gapsys, A. S. Mey, D. L. Mobley, L. P. Benito, C. E. Schindler, G. Tresadern and G. L. Warren, *Living J. Mol. Sci.*, 2022, **4**, 1497–1497.

[51] A. Fischer, M. Smiesko, M. Sellner and M. A. Lill, *J. Med. Chem.*, 2021, **64**, 2489–2500.

[52] A. S. Mey, B. K. Allen, H. E. B. Macdonald, J. D. Chodera, D. F. Hahn, M. Kuhn, J. Michel, D. L. Mobley, L. N. Naden, S. Prasad, A. Rizzi, J. Scheen, M. R. Shirts, G. Tresadern and H. Xu, *Living J. Mol. Sci.*, 2020, **2**, 18378.

[53] G. Schneider, *Nat. Rev. Drug Discov.*, 2016, **17**, 97–113.

[54] M. Volkov, J.-A. Turk, N. Drizard, N. Martin, B. Hoffmann, Y. Gaston-Mathé and D. Rognan, *J. Med. Chem.*, 2022, **65**, 7946–7958.

[55] H. C. S. Chan, H. Shan, T. Dahoun, H. Vogel and S. Yuan, *Trends Pharmacol. Sci.*, 2019, **40**, 592–604.

[56] A. L. Hopkins and C. R. Groom, *Nat. Rev. Drug Discov.*, 2002, **1**, 727–730.

[57] D. L. Nelson and M. M. Cox, *Lehninger Principles of Biochemistry*, W. H. Freeman, New York, NY, 2005.

[58] G. A. Petsko and D. Ringe, *Protein Structure and Function*, New Science Press, London, UK, 2004.

[59] J. M. Berg, G. J. Gatto Jr, J. Hines, J. L. Tymoczko and L. Stryer, *Biochemistry*, Macmillan Higher Education, 2023.

[60] R. Trivedi and H. A. Nagarajaram, *Int. J. Mol. Sci.*, 2022, **23**, 14050.

[61] G. Rhodes, *Academic Press*, 2010.

[62] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, A. Tzur, B. Gautam, M. Hassanali *et al.*, *Nucleic Acids Res.*, 2011, **33**, D412–D415.

[63] W. Kühlbrandt, *Science*, 2014, **343**, 1443–1444.

[64] H. Zheng, J. Hou, M. D. Zimmerman, A. Wlodawer and W. Minor, *Expert Opin. Drug Discov.*, 2014, **9**, 125–137.

[65] J. Koehler Leman and G. Künze, *Int. J. Mol. Sci.*, 2023, **24**, 7835.

[66] Y. Liu, D. T. Huynh and T. O. Yeates, *Nat. Commun.*, 2019, **10**, 1864.

[67] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne, *Nucleic Acids Res.*, 2000, **28**, 235–242.

[68] wwPDB consortium, *Nucleic Acids Res.*, 2019, **47**, D520–D528.

[69] F. W. Pun, I. V. Ozerov and A. Zhavoronkov, *Trends Pharmacol. Sci.*, 2023, **44**, 669–682.

[70] X. Du and L. Zhang, *J. Mol. Biol.*, 2024, **433**, 2345–2356.

[71] M. Springer and A. Gupta, *Nat. Drug Discov.*, 2024, **25**, 134–155.

[72] L. Smith and P. Johnson, *Drug Discov. Today*, 2024, **29**, 12–34.

[73] M. Jinek, K. Chylinski, I. Fonfara *et al.*, *Science*, 2012, **337**, 816–821.

[74] Y. Hasin, M. Seldin and A. Lusis, *Genome Biol.*, 2017, **18**, 1–15.

[75] Y. You, X. Lai, Y. Pan, H. Zheng, J. Vera, S. Liu, S. Deng and L. Zhang, *Signal Transduct. Target. Ther.*, 2022, **7**, 156.

[76] J. P. Taylor-King, M. Bronstein and D. Roblin, *Clin. Pharm. Therap.*, 2024, **115**, 655–657.

[77] A. Lee, K. Lee and D. Kim, *Expert Opin. Drug Discov.*, 2016, **11**, 707–715.

[78] C. Liu, K. Xiao, C. Yu, Y. Lei, K. Lyu, T. Tian, D. Zhao, F. Zhou, H. Tang and J. Zeng, *PLoS Comput. Biol.*, 2024, **20**, e1011945.

[79] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis and S. Velankar, *Nucleic Acids Res.*, 2022, **50**, D439–D444.

[80] A. Marshall, A. Perl and D. Hunt, *Nat. Rev. Drug Discov.*, 2013, **12**, 103–119.

[81] D. C. Whitcomb, *Nat. Rev. Drug Discov.*, 2006, **5**, 462–469.

[82] M. Korn, C. Ehrt, F. Ruggiu, M. Gastreich and M. Rarey, *Curr. Opin. Struct. Biol.*, 2023, **80**, 102578.

[83] W. A. Warr, M. C. Nicklaus, C. A. Nicolaou and M. Rarey, *J. Chem. Inf. Model.*, 2022, **62**, 2021–2034.

[84] J. L. Melville, E. K. Burke and C. N. Hsu, *J. Chem. Inf. Model.*, 2009, **49**, 763–776.

[85] R. Macarron, M. N. Banks, D. Bojanic *et al.*, *Nat. Rev. Drug Discov.*, 2011, **10**, 188–195.

[86] L. M. Mayr and P. Fuerst, *Methods Mol. Biol.*, 2018, **1683**, 1–12.

[87] S. Kenny and J. Komisarof, *J. Biomol. Screen.*, 2021, **26**, 737–748.

[88] G. Schneider and P. Schneider, *Nat. Rev. Drug Discov.*, 2010, **9**, 273–286.

[89] K. A. Giuliano and P. A. Johnston, *Curr. Opin. Chem. Biol.*, 2021, **65**, 102–111.

[90] R. Edmondson, J. J. Broglie, A. F. Adcock and L. Yang, *Assay Drug Dev. Technol.*, 2014, **12**, 207–218.

[91] I. Chen and R. Hubbard, *Curr. Opin. Drug Discov. Dev.*, 2009, **12**, 374–383.

[92] A. L. Hopkins, G. M. Keserü, P. D. Leeson *et al.*, *Drug Discov. Today*, 2004, **9**, 430–431.

[93] P. J. Hajduk and J. Greer, *Nat. Rev. Drug Discov.*, 2007, **6**, 211–219.

[94] C. W. Murray and D. C. Rees, *Annu. Rep. Med. Chem.*, 2009, **44**, 251–267.

[95] Enamine, *Enamine's REAL database: Expanding the accessible chemical space for drug discovery*, `https://enamine.net/library-synthesis/real-compounds`, 2021, Accessed: October 2024.

[96] CHEMriya, *CHEMriya database: A new frontier for ultra-large chemical space exploration*, `https://chemriya.com`, 2020, Accessed: October 2024.

[97] J. J. Irwin and B. K. Shoichet, *J. Chem. Inf. Model.*, 2005, **45**, 177–182.

[98] GalaXi, *GalaXi: A new paradigm in ultra-large-scale virtual screening libraries*, `https://galaxi.com/library`, 2020, Accessed: October 2024.

[99] eXplore, *eXplore Library: Expanding the chemical space with computational design*, `https://explore.com`, 2020, Accessed: October 2024.

[100] P. Ertl and A. Schuffenhauer, *J. Chem. Inf. Model.*, 2009, **49**, 123–134.

[101] L. C. Blum and J.-L. Reymond, *J. Chem. Inf. Model.*, 2010, **50**, 616–629.

[102] L. Ruddigkeit, L. C. Blum and J.-L. Reymond, *J. Chem. Inf. Model.*, 2012, **52**, 2864–2875.

[103] A. Anderson and P. M. Hughes, *Drug Discov. Today*, 2009, **14**, 1150–1158.

[104] D. A. Smith, K. Beaumont, T. S. Maurer and L. Di, *Drug Discov. Today*, 2012, **17**, 318–327.

[105] S. Andersson, N. Blomberg and K. Nilsson, *Nat. Rev. Drug Discov.*, 2009, **8**, 554–555.

[106] S. D. Roughley and A. M. Jordan, *J. Med. Chem.*, 2011, **54**, 3451–3479.

[107] A. Hillisch, L. F. Pineda and R. Hilgenfeld, *Drug Discov. Today*, 2004, **9**, 659–669.

[108] X. Zheng, M. E. Gleave and K. P. Monaghan, *Future Med. Chem.*, 2014, **6**, 537–548.

[109] G. M. Keserű, D. A. Erlanson, G. Williams *et al.*, *J. Med. Chem.*, 2016, **59**, 8189–8206.

[110] O. B. Cox, T. Krojer, P. Collins *et al.*, *J. Chem. Biol.*, 2016, **23**, 747–757.

[111] M. Whittaker, L. Hoeferlin and A. Barker, *J. Med. Chem.*, 2010, **53**, 5333–5343.

[112] X. Liu and Y. Tang, *Drug Discov. Today*, 2017, **22**, 615–622.

[113] H. Lee and J. Kwon, *J. Pharmacol. Exp. Ther.*, 2018, **367**, 421–435.

[114] J. Adams, *Preclinical Drug Development*, Springer, 2015.

[115] J. Bailey, M. Thew and M. Balls, *Altern. Lab. Anim.*, 2014, **42**, 1–12.

[116] X. Liu and W. Zhang, *Eur. J. Drug Metab. Pharmacokinet.*, 2018, **43**, 173–184.

[117] B. Dermody and D. Tsuji, *Expert Opin. Drug Discov.*, 2016, **11**, 365–378.

[118] K. L. Chapman and H. H. Holzgrefe, *J. Toxicol.*, 2013, **2013**, 1–10.

[119] A. R. Boobis and S. M. Cohen, *Regul. Toxicol. Pharmacol.*, 2013, **65**, 320–329.

[120] J. A. Clayton and K. S. Collins, *Lab Anim.*, 2018, **47**, 141–149.

[121] M. Li and T. Xu, *Drug Dev. Ind. Pharm.*, 2019, **45**, 470–476.

[122] R. M. Califf, *Transl. Res.*, 2016, **171**, 1–12.

[123] U.S. Food and Drug Administration, *Clinical Trial Design: FDA Guidance for Industry*, FDA, 2020.

[124] P. B. Chapman and L. H. Einhorn, *Oncology*, 2015, **29**, 164–169.

[125] E. Kim and D. Lemmon, *Expert Opin. Investig. Drugs*, 2015, **24**, 719–730.

[126] H. Ford *et al.*, *J. Clin. Oncol.*, 2018, **36**, e18632–e18632.

[127] European Medicines Agency (EMA), *Eur. Regul. Aff.*, 2019, 1–6.

[128] M. K. Gilson, J. A. Given, B. L. Bush and J. A. McCammon, *Biophys. J.*, 1997, **72**, 1047–1069.

[129] D. F. Hahn, C. I. Bayly, H. E. B. Macdonald, J. D. Chodera, A. S. Mey, D. L. Mobley, L. P. Benito, C. E. Schindler, G. Tresadern and G. L. Warren, *arXiv:2105.06222*, 2021.

[130] K. C. Bulusu, R. Guha, D. J. Mason, R. P. Lewis, E. Muratov, Y. K. Motamedi, M. Cokol and A. Bender, *Drug Discov. Today*, 2016, **21**, 225–238.

[131] H. Öztürk, A. Özgür and E. Ozkirimli, *Bioinformatics*, 2018, **34**, i821–i829.

[132] F. L. Lambert, *J. Chem. Educ.*, 2002, **79**, 1241.

[133] I. Y. Ben-Shalom, S. Pfeiffer-Marek, K.-H. Baringhaus and H. Gohlke, *J. Chem. Inf. Model.*, 2017, **57**, 170–189.

[134] S. Wan, R. H. Stote and M. Karplus, *J. Chem. Phys.*, 2004, **121**, 9539–9548.

[135] B. Srinivasan and M. D. Lloyd, *Dose–Response Curves and the Determination of IC50 and EC50 Values*, 2024.

[136] R. L. Rich and D. G. Myszka, *J. Mol. Recognit.*, 2008, **21**, 355–400.

[137] M. Jerabek-Willemsen, T. André, R. Wanner, H. M. Roth, S. Duhr, P. Baaske and D. Breitsprecher, *J. Mol. Struct.*, 2014, **1077**, 101–113.

[138] A. Velazquez-Campoy and E. Freire, *Nat. Protoc.*, 2006, **1**, 186–191.

[139] J. Concepcion, K. Witte, C. Wartchow, S. Choo, D. Yao, S. Perspicace, J. Wei, P. Li, L. Wai and R. Varma, *Comb. Chem. High Throughput Screen.*, 2009, **12**, 791–800.

[140] D. M. Jameson and S. E. Seifried, *Methods*, 1999, **19**, 222–233.

[141] M. Pellecchia, I. Bertini, D. Cowburn, C. Dalvit, E. Giralt, W. Jahnke, T. L. James, S. W. Homans, C. Ludwig and J. W. Peng, *Nat. Rev. Drug Discov.*, 2008, **7**, 738–745.

[142] M. C. Lo, A. Aulabaugh, G. Jin, R. Cowling, J. Bard, M. Malamas and G. Ellestad, *Anal. Biochem.*, 2004, **332**, 153–159.

[143] D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.

[144] L. Zhao, Y. Zhu, J. Wang, N. Wen, C. Wang and L. Cheng, *Comput. Struct. Biotechnol. J.*, 2022, **20**, 2831–2838.

[145] B.-X. Du, Y. Qin, Y.-F. Jiang, Y. Xu, S.-M. Yiu, H. Yu and J.-Y. Shi, *Drug Discov. Today*, 2022, **27**, 1350–1366.

[146] X. Liu, S. Jiang, X. Duan, A. Vasan, C. Liu, C.-c. Tien, H. Ma, T. Brettin, F. Xia, I. T. Foster *et al.*, *arXiv:2410.00709*, 2024.

[147] T. Liu, Y. Lin, Y. Zhong, Y. Li, H. Zhang, L. Yu, X. Chen, X. Huang, L. Wu, J. Chen, Y. Wang *et al.*, *Nucleic Acids Res.*, 2007, **35**, D198–D201.

[148] G. A. Landrum and S. Riniker, *J. Chem. Inf. Model.*, 2024, **64**, 1560–1567.

[149] D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. P. Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka *et al.*, *Nucleic Acids Res.*, 2019, **47**, D930–D940.

[150] J. Tang, A. Szwajda, S. Shakyawar, T. Xu, P. Hintsanen, K. Wennerberg and T. Aittokallio, *J. Chem. Inf. Model.*, 2014, **54**, 735–743.

[151] J. T. Metz, E. F. Johnson, N. B. Soni, P. J. Merta, L. Kifle and P. J. Hajduk, *Nat. Chem. Biol.*, 2011, **7**, 200–202.

[152] T. He, M. Heidemeyer, F. Ban, A. Cherkasov and M. Ester, *J. Cheminform.*, 2017, **9**, 1–14.

[153] M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber and P. P. Zarrinkar, *Nat. Biotechnol.*, 2011, **29**, 1046–1051.

[154] K. Huang, T. Fu, W. Gao, Y. Zhao, Y. Roohani, J. Leskovec, C. W. Coley, C. Xiao, J. Sun and M. Zitnik, *arXiv:2102.09548*, 2021.

[155] Z. Liu, Y. Li, L. Han, J. Liu, Z. Zhao, W. Nie, Y. Liu and R. Wang, *J. Chem. Inf. Model.*, 2016, **56**, 595–607.

[156] J. Li, X. Guan, O. Zhang, K. Sun, Y. Wang, D. Bagni and T. Head-Gordon, *arXiv preprint arXiv:2308.09639*, 2023.

[157] L. Hu, M. L. Benson, R. D. Smith, M. G. Lerner and H. A. Carlson, *Proteins*, 2005, **60**, 333–340.

[158] S. Wagle, R. D. Smith, A. J. Dominic III, D. DasGupta, S. K. Tripathi and H. A. Carlson, *Sci. Rep.*, 2023, **13**, 3008.

[159] T. Siebenmorgen, F. Menezes, S. Benassou, E. Merdivan, K. Didi, A. S. D. Mourão, R. Kitel, P. Liò, S. Kesselheim, M. Piraud *et al.*, *Nat. Comput. Sci.*, 2024, 1–12.

[160] S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman, *J. Mol. Biol.*, 1990, **215**, 403–410.

[161] A. Krogh, M. Brown, I. S. Mian, K. Sjölander and D. Haussler, *J. Mol. Biol.*, 1994, **235**, 1501–1531.

[162] S. R. Eddy, *Bioinformatics*, 1998, **14**, 755–763.

[163] S. Hellberg, M. Sjöström, B. Skagerberg and S. Wold, *Biopolymers*, 1987, **26**, 423–439.

[164] R. Rao, J. Meier, T. Sercu, S. Ovchinnikov and A. Rives, *bioRxiv*, 2021, 2020.12.15.422761.

[165] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli *et al.*, *bioRxiv*, 2023, 2022.07.20.500902.

[166] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger *et al.*, *arXiv Prepr. arXiv:2007.06225*, 2020.

[167] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey and C. H. Wu, *Bioinformatics*, 2015, **31**, 926–932.

[168] M. Vendruscolo, E. Kussell and E. Domany, *Proc. Natl. Acad. Sci. USA*, 1997, **94**, 8427–8431.

[169] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa and M. Weigt, *Proc. Natl. Acad. Sci. USA*, 2011, **108**, E1293–E1301.

[170] J. K. Noel, P. C. Whitford and J. N. Onuchic, *J. Phys. Chem. B*, 2012, **116**, 8692–8702.

[171] G. Menichetti, P. Fariselli and D. Remondini, *Sci. Rep.*, 2016, **6**, 30367.

[172] J. J. Güven, N. Molkenthin, S. Mühle and A. S. Mey, *Phys. Biol.*, 2023, **20**, 046004.

[173] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le and S. Venkatesh, *Bioinformatics*, 2021, **37**, 1140–1147.

[174] M. Jiang, Z. Li, S. Zhang, S. Wang, X. Wang, Q. Yuan and Z. Wei, *RSC Adv.*, 2020, **10**, 20701–20712.

[175] R. Özçelik, D. van Tilborg, J. Jiménez-Luna and F. Grisoni, *ChemBioChem*, 2023, **24**, e202200776.

[176] M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri and D. R. Koes, *J. Chem. Inf. Model.*, 2017, **57**, 942–957.

[177] J. Jiménez, S. Doerr, G. Martínez-Rosell, A. S. Rose and G. De Fabritiis, *Bioinformatics*, 2017, **33**, 3036–3042.

[178] C. Isert, K. Atz and G. Schneider, *Current Opinion in Structural Biology*, 2023, **79**, 102548.

[179] E. N. Feinberg, D. Sur, Z. Wu, B. E. Husic, H. Mai, Y. Li, S. Sun, J. Yang, B. Ramsundar and V. S. Pande, *ACS Cent. Sci.*, 2018, **4**, 1520–1530.

[180] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals and G. E. Dahl, Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1263–1272.

[181] X. Kong, W. Huang and Y. Liu, *arXiv:2208.06073*, 2022.

[182] M. L. Connolly, *J. Appl. Crystallogr.*, 1983, **16**, 548–558.

[183] B. Lee and F. M. Richards, *J. Mol. Biol.*, 1971, **55**, 379–400.

[184] T. Cieplak and J. L. Wisniewski, *Molecules*, 2001, **6**, 915–926.

[185] D. Bajusz, A. Rácz and K. Héberger, *Comprehensive Medicinal Chemistry III*, Elsevier, Oxford, 2017, pp. 329–378.

[186] D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.

[187] J. L. Durant, B. A. Leland, D. R. Henry and J. G. Nourse, *J. Chem. Inf. Comput. Sci.*, 2002, **42**, 1273–1280.

[188] R. E. Carhart, D. H. Smith and R. Venkataraghavan, *J. Chem. Inf. Comput. Sci.*, 1985, **25**, 64–73.

[189] M. C. Jones, K. Pulapally and R. Desai, *J. Chem. Inf. Model.*, 2007, **47**, 1613–1622.

[190] P. Willett, J. M. Barnard and G. M. Downs, *J. Chem. Inf. Comput. Sci.*, 1998, **38**, 292–304.

[191] C. A. Lipinski, F. Lombardo, B. W. Dominy and P. J. Feeney, *Adv. Drug Deliv. Rev.*, 2012, **64**, 4–17.

[192] R. Mannhold, G. I. Poda, C. Ostermann and I. V. Tetko, *J. Pharm. Sci.*, 2009, **98**, 861–893.

[193] D. F. Veber, S. R. Johnson, H.-Y. Cheng, B. R. Smith, K. W. Ward and K. D. Kopple, *J. Med. Chem.*, 2002, **45**, 2615–2623.

[194] P. Ertl, B. Rohde and P. Selzer, *J. Med. Chem.*, 2000, **43**, 3714–3717.

[195] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, *Adv. Neural Inf. Process. Syst.*, 2013, **26**, year.

[196] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, *Adv. Neural Inf. Process. Syst.*, 2017, **30**, year.

[197] G. B. Goh, N. O. Hodas, C. Siegel and A. Vishnu, *arXiv:1712.02034*, 2017.

[198] S. Jaeger, S. Fulle and S. Turk, *J. Chem. Inf. Model.*, 2018, **58**, 27–35.

[199] W. Ahmad, E. Simon, S. Chithrananda, G. Grand and B. Ramsundar, *arXiv:2209.01712*, 2022.

[200] S. Honda, S. Shi and H. R. Ueda, *arXiv:1911.04738*, 2019.

[201] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko and K.-R. Müller, *Nat. Commun.*, 2017, **8**, 1389.

[202] F. A. Faber, L. Hutchison, B. Huang and O. A. Von Lilienfeld, *J. Chem. Theory Comput.*, 2017, **13**, 5255–5264.

[203] M. Skalic, J. Jiménez, D. Sabbadin and G. De Fabritiis, *J. Chem. Inf. Model.*, 2019, **59**, 1205–1214.

[204] F. Stanzione, I. Giangreco and J. C. Cole, *Prog. Med. Chem.*, 2021, **60**, 273–343.

[205] C. Shen, J. Ding, Z. Wang, D. Cao, X. Ding and T. Hou, *WIREs Comput. Mol. Sci.*, 2020, **10**, e1429.

[206] S. Genheden and U. Ryde, *Expert Opin. Drug Discov.*, 2015, **10**, 449–461.

[207] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell and A. J. Olson, *J. Comput. Chem.*, 2009, **30**, 2785–2791.

[208] J. Li, A. Fu and L. Zhang, *Interdiscip. Sci. Comput. Life Sci.*, 2019, **11**, 320–328.

[209] S.-Y. Huang, *Brief. Bioinform.*, 2018, **19**, 982–994.

[210] H. Zhao and D. Huang, *PloS one*, 2011, **6**, e19923.

[211] T. A. Halgren, *J. Comput. Chem.*, 1996, **17**, 553–586.

[212] S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta and P. Weiner, *J. Am. Chem. Soc.*, 1984, **106**, 765–784.

[213] J. A. Morrone, J. K. Weber, T. Huynh, H. Luo and W. D. Cornell, *J. Chem. Inf. Model.*, 2020, **60**, 4170–4179.

[214] M. E. Tuckerman, *J. Condens. Matter Phys.*, 2002, **14**, R1297.

[215] H. Gohlke, C. Kiel and D. A. Case, *J. Mol. Biol.*, 2004, **330**, 891–913.

[216] E. Wang, H. Sun, J. Wang, Z. Wang, H. Liu, J. Z. Zhang and T. Hou, *Chem. Rev.*, 2019, **119**, 9478–9508.

[217] S. Genheden and U. Ryde, *J. Comput. Chem.*, 2010, **31**, 837–846.

[218] M. R. Shirts and V. S. Pande, *Annu. Rev. Phys. Chem.*, 2007, **58**, 219–246.

[219] Y. Khalak, G. Tresadern, M. Aldeghi, H. M. Baumann, D. L. Mobley, B. L. de Groot and V. Gapsys, *Chem. Sci.*, 2021, **12**, 13958–13971.

[220] M. R. Shirts, D. L. Mobley, S. P. Brown and V. S. Pande, *J. Phys. Chem. B*, 2008, **112**, 617–627.

[221] C. Hansch, P. P. Maloney, T. Fujita and R. M. Muir, *Nature*, 1964, **194**, 178–180.

[222] A. Tropsha, *Mol. Inform.*, 2010, **29**, 476–488.

[223] C. Hansch and T. Fujita, *J. Am. Chem. Soc.*, 1962, **84**, 4334–4341.

[224] C. M. Bishop and H. Bishop, *Deep Learning: Foundations and Concepts*, Springer Nature, 2023.

[225] S. Scardapane, *arXiv:2404.17625*, 2024.

[226] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, Springer, 2006, vol. 4.

[227] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, *IEEE Trans. Neural Netw. Learn. Syst.*, 2021, **33**, 6999–7019.

[228] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie and L. Farhan, *J. Big Data*, 2021, **8**, 1–74.

[229] R. Gorantla, R. K. Singh, R. Pandey and M. Jain, 2019 IEEE BIBE, 2019, pp. 397–404.

[230] R. K. Singh and R. Gorantla, *PLoS One*, 2020, **15**, e0220677.

[231] R. K. Singh, R. Gorantla, S. G. R. Allada and P. Narra, *PLoS One*, 2022, **17**, e0276836.

[232] B. Khemani, S. Patil, K. Kotecha and S. Tanwar, *J. Big Data*, 2024, **11**, 18.

[233] A. Duval, S. V. Mathis, C. K. Joshi, V. Schmidt, S. Miret, F. D. Malliaros, T. Cohen, P. Lio, Y. Bengio and M. Bronstein, *arXiv:2312.07511*, 2023.

[234] C. Nauck, R. Gorantla, M. Lindner, K. Schürholt, A. S. Mey and F. Hellmann, *arXiv:2407.12419*, 2024.

[235] T. Lin, Y. Wang, X. Liu and X. Qiu, *AI Open*, 2022, **3**, 111–132.

[236] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, *arXiv:1810.04805*, 2018.

[237] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, *arXiv:1907.11692*, 2019.

[238] A. Sultan, J. Sieg, M. Mathea and A. Volkamer, *J. Chem. Inf. Model.*, 2024, **64**, 6259–6280.

[239] K.-D. Luong and A. Singh, *J. Chem. Inf. Model.*, 2024.

[240] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, *arXiv:1910.03771*, 2019.

[241] T. U. Consortium, *Nucleic Acids Res.*, 2021, **49**, D480–D489.

[242] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang and E. E Bolton, *Nucleic Acids Res.*, 2019, **47**, D1102–D1109.

[243] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli *et al.*, *Science*, 2023, **379**, 1123–1130.

[244] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido and A. Rives, *BioRxiv:10.1101/2022.07.20.500902*, 2022.

[245] G. Landrum, *RDKit: Open-source cheminformatics*, 2006, Available at: `https://www.rdkit.org`.

[246] R. Gorantla, A. Kubincova, B. Suutari, B. P. Cossins and A. S. Mey, *J. Chem. Inf. Model.*, 2024.

[247] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti and G. Csányi, *Chem. Rev.*, 2021, **121**, 10073–10141.

[248] B. Settles, Active learning and experimental design workshop in conjunction with AISTATS 2010, 2011, pp. 1–18.

[249] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen and X. Wang, *ACM Comput. Surv.*, 2021, **54**, 1–40.

[250] J. Thompson, W. P. Walters, J. A. Feng, N. A. Pabon, H. Xu, B. B. Goldman, D. Moustakas, M. Schmidt and F. York, *Artif. Intell. Life Sci.*, 2022, **2**, 100050.

[251] A. Orlov, T. Akhmetshin, D. Horvath, G. Marcou and A. Varnek, *ChemRxiv*, 2024.

[252] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos and E. Tuzhilina, *Nat. Rev. Methods Primers*, 2022, **2**, 100.

[253] L. McInnes, J. Healy, N. Saul and L. Großberger, *J. Open Source Softw.*, 2018, **3**, 861.

[254] W.-f. Shen, H.-w. Tang, J.-b. Li, X. Li and S. Chen, *J. Cheminform.*, 2023, **15**, 1–16.

[255] R. Gorantla, A. P. Gema, I. X. Yang, Á. Serrano-Morrás, B. Suutari, J. J. Jiménez and A. S. J. S. Mey, *bioRxiv*, 2024.

[256] E. R. Ziegel, *The elements of statistical learning*, 2003.

[257] D. Chicco and G. Jurman, *BMC Genom.*, 2020, **21**, 1–13.

[258] M. Hollander, D. A. Wolfe and E. Chicken, *Nonparametric statistical methods*, John Wiley Sons, 2013.

[259] O. Rainio, J. Teuho and R. Klén, *Sci. Rep.*, 2024, **14**, 6086.

[260] S. Midway, M. Robertson, S. Flinn and M. Kaller, *PeerJ*, 2020, **8**, e10387.

[261] D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, 10th edn, 2020.

[262] Q. McNemar, *Psychometrika*, 1947, **12**, 153–157.

[263] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Science & Business Media, 2nd edn, 2009.

[264] C. Cortes and V. Vapnik, *Mach. Learn.*, 1995, **20**, 273–297.

[265] L. Breiman, *Mach. Learn.*, 2001, **45**, 5–32.

[266] D. W. Jones, *J. Chem. Inf. Comput. Sci.*, 1992, **32**, 234–252.

[267] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl and V. Svetnik, *J. Chem. Inf. Model.*, 2015, **55**, 263–274.

[268] K. Yang, K. Swanson, W. Jin, C. W. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, W. M. Matheos *et al.*, *J. Chem. Inf. Model.*, 2019, **59**, 3370–3388.

[269] D. K. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik and R. P. Adams, Adv. Neural Inf. Process. Syst., 2015, pp. 2224–2232.

[270] H. Altae-Tran, B. Ramsundar, A. S. Pappu and V. Pande, *ACS Cent. Sci.*, 2017, **3**, 283–293.

[271] D. D. Wang, W. Wu and R. Wang, *J. Cheminform.*, 2024, **16**, 2.

[272] A. de Ruiter and C. Oostenbrink, *J. Chem. Inf. Model.*, 2017, **57**, 1118–1128.

[273] J. Jiménez, M. Skalic, G. Martinez-Rosell and G. De Fabritiis, *J. Chem. Inf. Model.*, 2018, **58**, 287–296.

[274] L. Zheng, J. Fan and Y. Mu, *ACS Omega*, 2019, **4**, 15956–15965.

[275] M. M. Stepniewska-Dziubinska, P. Zielenkiewicz and P. Siedlecki, *Bioinformatics*, 2018, **34**, 3666–3674.

[276] Y. Li, M. A. Rezaei, C. Li and X. Li, 2019 IEEE BIBM, 2019, pp. 303–310.

[277] D. Jones, H. Kim, X. Zhang, A. Zemla, G. Stevenson, W. D. Bennett, D. Kirshner, S. E. Wong, F. C. Lightstone and J. E. Allen, *J. Chem. Inf. Model.*, 2021, **61**, 1583–1592.

[278] E. N. Feinberg, D. Sur, Z. Wu, B. E. Husic, H. Mai, Y. Li, S. Sun, J. Yang, B. Ramsundar and V. S. Pande, *ACS Cent. Sci.*, 2018, **4**, 1520–1530.

[279] J. Son and D. Kim, *PLoS ONE*, 2021, **16**, e0249404.

[280] G. Liu, M. Singha, L. Pu, P. Neupane, J. Feinstein, H.-C. Wu, J. Ramanujam and M. Brylinski, *J. Cheminform.*, 2021, **13**, 1–17.

[281] M. Backenköhler, J. Groß, V. Wolf and A. Volkamer, *J. Chem. Inf. Model.*, 2024.

[282] Q. Feng, E. Dueva, A. Cherkasov and M. Ester, *arXiv:1807.09741*, 2018.

[283] M. M. Gromiha, *Protein bioinformatics: from sequence to function*, academic press, 2010.

[284] H. Öztürk, E. Ozkirimli and A. Özgür, *arXiv:1902.04166*, 2019.

[285] M. Karimi, D. Wu, Z. Wang and Y. Shen, *Bioinformatics*, 2019, **35**, 3329–3338.

[286] X. Lin, *arXiv:2003.13902*, 2020.

[287] S. Zhang, M. Jiang, S. Wang, X. Wang, Z. Wei and Z. Li, *Int. J. Mol. Sci.*, 2021, **22**, 8993.

[288] Z. Jin, T. Wu, T. Chen, D. Pan, X. Wang, J. Xie, L. Quan and Q. Lyu, *Bioinformatics*, 2023, **39**, btad049.

[289] P. Cohen, *Nat. Rev. Drug Discov.*, 2002, **1**, 309–315.

[290] L. Castelo-Soccio, H. Kim, M. Gadina, P. L. Schwartzberg, A. Laurence and J. J. O'Shea, *Nat. Rev. Immunol.*, 2023, **23**, 787–806.

[291] G. Manning, D. B. Whyte, R. Martinez, T. Hunter and S. Sudarsanam, *Science*, 2002, **298**, 1912–1934.

[292] R. Gorantla, A. Kubincova, A. Y. Weiße and A. S. Mey, *J. Chem. Inf. Model.*, 2024.

[293] M. Michel, D. Menéndez Hurtado and A. Elofsson, *Bioinformatics*, 2019, **35**, 2677–2679.

[294] G. K. Kanev, C. de Graaf, B. A. Westerman, I. J. de Esch and A. J. Kooistra, *Nucleic Acids Res.*, 2021, **49**, D562–D569.

[295] M. K. Gilson, T. Liu, M. Baitaluk, G. Nicola, L. Hwang and J. Chong, *Nucleic Acids Res.*, 2016, **44**, D1045–D1053.

[296] O. Trott and A. J. Olson, *J. Comput. Chem.*, 2010, **31**, 455–461.

[297] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick *et al.*, *Nature*, 2024, **630**, 493–500.

[298] Z. Zhang, B. Zhao, A. Xie, Y. Bian and S. Zhou, *Activity Cliff Prediction: Dataset and Benchmark*, 2023, arXiv:2302.07541 (accessed Sep 10, 2023).

[299] H. Achdout, A. Aimon, E. Bar-David and G. Morris, *BioRxiv*, 2020.

Please note the references above are only for Chapters 1, 2, 3 and 7. The references for the remaining chapters are present in the respective chapters itself.