MACHINE LEARNING FOR LARGE AND SMALL DATA BIOMEDICAL DISCOVERY

BY

YUNAN LUO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Associate Professor Jian Peng, Chair
Assistant Professor Mohammed El-Kebir
Professor Jiawei Han
Associate Professor Jianzhu Ma, Peking University
Dr. Hyunghoon Cho, Broad Institute of MIT and Harvard

Abstract

In modern biomedicine, the role of computation becomes more crucial in light of the ever-increasing growth of biological data, which requires effective computational methods to integrate them in a meaningful way and unveil previously undiscovered biological insights. In this dissertation, we introduce a series of machine learning algorithms for biomedical discovery. Focused on protein functions in the context of system biology, these machine learning algorithms learn representations of protein sequences, structures, and networks in both the small- and large-data scenarios. First, we present a deep learning model that learns evolutionary contexts integrated representations of protein sequence and assists to discover protein variants with enhanced functions in protein engineering. Second, we describe a geometric deep learning model that learns representations of protein and compound structures to inform the prediction of protein-compound binding affinity. Third, we introduce a machine learning algorithm to integrate heterogeneous networks by learning compact network representations and to achieve drug repurposing by predicting novel drug-target interaction. We also present new scientific discoveries enabled by these machine learning algorithms. Taken together, this dissertation demonstrates the potential of machine learning to address the small- and large-data challenges of biomedical data and transform data into actionable insights and new discoveries.

Acknowledgments

During my Ph.D. study, I have been enjoying reading the acknowledgments section in others' dissertations. There were supportive people and vivid stories behind everyone's journey of a Ph.D., and mine was not an exception. I have been extremely lucky to be surrounded by a number of advisors, colleagues, friends, and family both in and out of the University of Illinois Urbana-Champaign (UIUC). And now, I finally have the opportunity to write down my acknowledgments section to thank them all, with my greatest gratitude.

First and foremost, I would like to give my warmest thanks to my advisor, Professor Jian Peng. I still remember the first meeting I had with Jian during his visit to Tsinghua University in 2015. He told me, with enthusiasm and excitement, about his new algorithm for identifying target genes of Parkinson's disease. I was immediately impressed by his research and the potential impact that could be achieved. That summer, Jian hosted me as a visiting undergraduate student at UIUC, and I had a fun time working with him on two research projects, part of which was then further developed into some results presented in this dissertation. The experience was so joyous that it became a no-brainer for me to choose Jian as my advisor when I was applying to graduate schools. Throughout my five years at UIUC, Jian has been a never-ending source of guidance, support, and inspiration for me and kept passing his knowledge, caring, and encouragement to me. He is not only a visionary advisor with insightful advice and sharp questions but also a great colleague who knows the finest details of a problem and could help with any technical challenges I had encountered. Now when I am reading the thousands of emails and Skype text messages I exchanged with him over the past few years, plenty of great memories flashback: the detailed notes he sent about the trip from Chicago airport to Champaign before my first visit to UIUC, the extensive discussion when we were working on our first RECOMB submission, the numerous paper links he sent to guide me to get familiar with a new research topic, the clever ideas he proposed and key issues he pinpointed when I was struggling with research, his affirmative "sure, always happy to" every time I requested help from him, and his huge effort in supporting my job search. Those good old times very well illustrate how I become who I am today after five years working with Jian. During my Ph.D., Jian gave me tremendous freedom, which was truly a luxury, and I greatly appreciate his patience and belief when helping me grow my research independence. Looking back, what I learned from Jian is far beyond a degree, but the way to think, the philosophy to do research, and the career to pursue as a scholar. It was my privilege to have Jian as my advisor, and I am forever

grateful to him.

I also owe many thanks to my other mentors who have guided and led me through my Ph.D. I have special gratitude for Professor Jianzhu Ma. I started to work with Jianzhu in my second year and he remains my close collaborator since then. While we only met once in person during my Ph.D., we made innumerable Skype calls to brainstorm research ideas and discuss coding and writing. Working with Jianzhu was a precious experience for me and it really motivated me to think about how to formulate research problems. Thanks also go to Professor Mohammed El-Kebir. I worked with Mohammed on cancer genomics for a period of time. He is an amazing mentor on everything, from research, to writing and presentation, and to coding and paper reviewing. I will always remember the scenes he welcomed me into his office "Yunan, come on in!" and started a fruitful meeting with me. Working with Professor Kaiyu Guan on remote sensing and intelligent agriculture was another unique experience. I was strongly impressed and encouraged by his scholarship: self-discipline, determination, and energy. I believe that everyone knowing Kaiyu, including me, might never understand how one can accomplish so many things at the same time. I also closely worked with Professor Huimin Zhao on protein engineering, who taught me how to study scientific problems in an interdisciplinary context. It was my pleasure to collaborate with Professor Bonnie Berger in early years of my Ph.D., and I benefited a lot from every stage of the project.

I would also like to thank many other brilliant professors and researchers. I want to especially thank Professor Jianyang Zeng for introducing me to the field of computational biology when I was a sophomore at Tsinghua University and for providing me with great support even after I started my Ph.D. in the US. Special thanks also go to Professor Andrew Chi-Chih Yao for founding Yao Class and for creating opportunities for undergraduate students like me to do frontier research. It is my honor to have Professor Jiawei Han on my dissertation committee, and his diligence and passion for research have a deep influence on me. Thanks to Dr. Hoon Cho for being my external committee member. His paper with Jian was the first RECOMB paper I have read, which also inspired part of the work in this dissertation. I am also grateful to Professors Tandy Warnow and Saurabh Sinha for providing me with invaluable advice at many stages of my degree. Dr. Ali Madani hosted me remotely at Salesforce Research for a summer internship, which was truly a wonderful experience. During my job search, I was fortunate to have met with and received guidance from many researchers both at UIUC and other institutions, and I sincerely thank their help and support.

My life at Urbana-Champaign was not boring at all because of my close friends. Yang Liu and Qing Ye were the first two people I got to know after I arrived at Champaign. They

offered a lot to help me get used to the new life, and we have had the happiest time at Qing Ye's "EMM" home. I knew Sheng Wang even before I joined UIUC. We started to collaborate in late 2015, and he remained my peer mentor since then. Besides them, over my time at UIUC, the Peng group has been gradually growing into a large family, and I feel extremely lucky to spend a significant portion of my 20s with them: Shibi He, Wei Qian, Jiaqi Guan, Xiaoming Zhao, Jinglin Chen, Yuanyi Zhong, Haoxiang Wang, Yufeng Su, and Hantian Ding. I will never forget the happiness we shared at the 1218 office, the group BBQ, reunions during the pandemic, and most importantly, countless EA FIFA nights. Meng Zhang has been my roommate since we joined UIUC and he is the best roommate and the most hardworking doctorate student I have seen at UIUC. I greatly appreciate his help over the years. I also should thank a great number of my old friends. I had much more fun chatting with Kaike, my friend for over a decade, about everything. The group video calls with Zhe, Zhenxing, and Bowen always freed me from the stressful research life. Ruonan and Peng have provided great help to my study life in the U.S. I could not imagine a long journey without my dear friends. Thanks for everything they brought to me.

Throughout my Ph.D., I have been funded by research grants from National Science Foundation (NSF), Department of Energy (DOE), and the Aligning Science Across Parkinson's (ASAP) initiative initiative, as well as fellowships from UIUC CompGen and Baidu. This dissertation would not be possible without the hard work of my colleagues and collaborators, and I thank them for their contributions.

I dedicate this dissertation to my parents, for their constant love and support. Choosing an academic career seems a natural choice for me in some way, but I know is never easy for them. As their only child, they certainly hope I could be around but they have been very supportive of every decision I made. Thank you, mom and dad, for everything! Finally, thanks to Qing for the unfading love. I feel so lucky to have you as a partner and the best friend to experience every piece of life together and share the highs and the lows. My Ph.D. journey has never been daunting with your company. I also thank Qing's parents for their caring and support. My life is filled with courage and happiness because of you all!

Table of Contents

Chapter 1: Introduction

The rapid development of biotechnology has revolutionized biology and medicine by generating massive, multi-model datasets. The role of computation becomes even more critical in light of the ever-increasing growth of those datasets, which requires effective computational methods to integrate them in a meaningful way and unveil previously undiscovered biological insights.

Machine learning has emerged as a powerful tool to transform biomedical data into knowledge discovery. However, reasoning over massive biomedical data and translating advanced machine learning into solutions to key biomedical problems present several fundamental challenges. On one hand, biomedical data are large-scale, heterogeneous, high-dimensional, and noisy, and computational methods are needed to operationalize the data to enable comprehensive analysis. On the other hand, due to the required human efforts and costly experimental procedures, biomedical data often have limited annotations, which brings a significant challenge for data-driven methods such as machine learning.

My research aims to address the above large-data and small-data challenges by developing new machine learning algorithms. Specifically, I have been focusing on several research questions in system biology. The core principle of my research is to develop machine learning algorithms to integrate diverse data and knowledge into actionable representations and to reason over such representations to guide knowledge discovery and reveal new scientific insights. Key contributions of my research include: (i) Inventing algorithms to integrate large-scale heterogeneous data to disentangle out non-redundant information from data noise and to represent them in a way amenable to comprehensive analyses; (ii) Developing the domain-tailored machine learning methods that incorporate domain expertise and prior knowledge to improve accuracy and generalizability, particularly in low-data regimes; (iii) Designing computational methods to unlock new synergistic workflows that cannot be realized solely with existing wet-lab techniques and to accelerate biological discovery and design.

1.1 BACKGROUND AND MOTIVATION

My dissertation research focused on several research problems in system biology, during which I have identified unique challenges due to the large- or small-scale natures of data in the problem. In this section, I motivate some of the current challenges in developing machine learning algorithms for system biology. While there are challenges specific to each problem, here I try to present the common challenges from a data perspective, i.e., what are the key

computational challenges posed by either large-scale data or small-scale data.

1.1.1 Large-scale data: heterogeneity, high-dimensionality, and integration

The recent advent of high-throughput experimental techniques has enabled the generation of large-scale biomedical data, which we want to leverage to interrogate and understand biological systems. A hallmark of currently available biomedical data is their heterogeneity. Biomedical data are of various types, ranging from experimental readouts, curated annotations, and metadata, and no single data type is sufficient to characterize the whole biological system such as a protein-protein interaction (PPI) network. The key challenge here is how to operationalize and integrate large-scale heterogeneous data in their broadest sense to reduce noise and redundancy. The high dimensionality of those datasets further amplifies the complexity of the problem and requires fundamentally new computational methods.

Later in this dissertation, we will consider a problem where we use PPI network data to study how proteins and their interactions determine their functions in particular biological processes, physical or genetic interactions, or disease associations. As it is intractable to exhaustively characterize proteins through biological experiments, computational hypotheses that integrate genome-scale interaction networks have garnered great interest. The key intuition behind such approaches is that proteins that have similar topological roles in the network are more likely to perform similar functions. Challenges here have been to develop algorithms to (i) capture the topological similarity relationships between proteins, and (ii) integrate heterogeneous information (e.g., physical interactions, co-expression, and genetic interactions) from which separate networks can be constructed.

Challenges in capturing topological roles of proteins. A type of network diffusion algorithm called random walk with restart (RWR) has been extensively studied to infer protein functions [1, 2, 3, 4, 5]. The core idea is to propagate information along the network to exploit both direct and indirect connectivity between proteins. Typically, a distribution is built for each protein in relation to other proteins in the network, so that one can select the most related proteins in the distribution or select proteins that share similar distributions. While having been demonstrated to be useful for inferring protein function, such approaches are still far from satisfactory, partially due to the fact that biological interactomes are often noisy and incomplete. For example, even the yeast interactome data, the highest quality data gathered among all organisms, have a significant portion of false positive and false negative edges [6].

Challenges in integrating heterogeneous interaction networks. The common way that most existing methods combine heterogeneous interaction networks (e.g., physical in-

teraction, co-expression, and genetic interaction networks) has been summarizing multiple networks into a single network through weighted averaging [7] or Bayesian inference [8, 9, 10]. The issue of those approaches is that network-specific interaction patterns (e.g., tissue-specific protein modules) may be buried behind edges from other datasets in the final collapsed network. A simple solution to circumvent this issue is to concatenate all input networks to preserve all features [1, 7, 11]. However, doing so greatly increase the dimensionality of the input feature space, which often obscures the signals in the data. The noise in interaction network generated from high-throughput experiments further exacerbates this issue.

It is imperative to develop algorithms to address the challenges mentioned above. In this dissertation, I will describe an algorithm to integrate topological features from multiple heterogeneous interaction networks, while coping with the inherent noise in high-throughput data.

1.1.2 Small-scale data: auxiliary information, representation, and transferability

Biomedical data are also small in the sense that they are rarely annotated with function labels, making it difficult to reason the functions and dynamics of a biological system. For example, only 7% of the human proteome have been explored for therapeutic opportunities and linked to at least one approved drug [12], more than 95% of the reported human phosphorylation sites have no known up-stream kinase or biological function [13], and nearly half of Gene Ontology terms for humans have fewer than 10 annotations [14]. The primary reasons for the limited available annotations include i) the space to explore is tremendously large and infeasible to exhaustively enumerate even with latest biotechnology, ii) the experimental procedures that generate annotations are often costly and time-consuming, iii) currently available data such as function labels are biased towards well-characterized proteins, leaving novel proteins sparsely labeled. Therefore, most biomedical data, while having large data volumes, often have quite limited information volumes within the dataset and do not readily transfer to knowledge discovery. This brings a significant challenge for data-driven computational approaches, especially machine learning algorithms, as those approaches rely on a sufficient amount of data for model training to offer satisfactory prediction performance and to reveal scientific insights.

Below, I list three directions, which will be discussed in this dissertation, to mitigate the data scarcity issue when developing machine learning algorithms for biomedical problems.

Learning contextual representations. Representation learning has been a powerful strategy in recent deep learning applications in computer vision and natural language

processing. Examples include earlier "shallow" representation learning algorithms such as word2vec [15] and node2vec [16] and, more recently, "deep" representation learning algorithms such as BERT [17] and GPT-3 [18]. Similar ideas have been applied to the biological domain to learn better representations for protein sequences [19, 20]: instead of representing protein sequences with the popular one-hot encoding approach as in many traditional machine learning approaches, these methods represent the sequences with real-valued vectors learned from massive unlabeled sequence data, which presumably capture the intrinsic syntax and semantics of protein sequence. Recent studies have found that this representation can substantially improve many protein-related tasks [19, 20, 21, 22, 23].

Improving model transferability. Improving the transferability of machine learning models is another direction to mitigate the data scarcity issue. There are two main-stream approaches to develop transferable models. The first one is to employ meta-learning [24] and few-shot learning [25] strategies. The idea of meta-learning is to train the model on multiple related tasks so that it learns to capture features or feature interactions that lead to better transferability. The few-shot learning strategy enforces the model to give accurate predictions after seeing a few samples as training data. The second paradigm is the "pre-training and fine-tuning" approach that has become a common strategy in many vision and language applications. This is also closely related to the representation learning techniques mentioned above. Typically, a model is pre-trained on large-scale unlabeled data to fill in masked words in a sentence or missing regions of an image. This pre-training stage is also called self-supervised training. After that, the model is fine-tuned using the data of our target task. The rationale is that through the pre-training stage, the model learns a set of parameters as a good initialization by leveraging the massive unlabeled data, and in the fine-tuning stage the model can accurately predict for the target task by slightly tuning those parameters using the labeled data of the target task, instead of re-learning all parameters from scratch.

Incorporating auxiliary information. Incorporating related data as auxiliary information can facilitate the learning process of deep learning. Consider sequence-based protein contact prediction as an example. Although the goal is to directly predict pairwise residue contacts by only using the protein sequence as input, it has been found that incorporating co-evolution information of residues into deep learning models can enhance the prediction performance [26]. The reason is that pairwise residue contacts often correlate with residue co-evolution couplings [27, 28, 29, 30, 31, 32, 33]. Therefore, instead of letting the deep learning model learn to find this correlation from data by itself from scratch, one can incorporate this information as a prior. In this way, the model will not waste the data power to re-learn this prior, instead, it can make full use of the data to predict residue contact, which is the

ultimate goal. In addition to these motivating examples, there are many problems where auxiliary information can be integrated to enhance the predictive model. For example, in this dissertation, we will show how to incorporate evolution information to predict protein function from sequence.

The above three directions represent the approaches I took to tackle the small-data challenge in biomedical problems. In this dissertation, I will describe in detail how to develop those ideas into concrete methods in the contexts of specific problems.

1.2 OVERVIEW OF DISSERTATION RESEARCH

My dissertation research studied the integration of large-scale heterogeneous networks and also focused on addressing the challenges caused by small-scale annotations in problems of system biology. A unified theme of my work is that they are all centered around learning better representations (e.g., for protein/DNA sequences, protein structures, and molecule interaction networks) with novel machine learning algorithms. In many cases, my algorithms have successfully enabled several new discoveries. In this dissertation, I will unify my research projects as representation learning of the a multi-scale hierarchy of protein, namely sequences, structures, and networks, to study the functions of proteins in biological systems.

My first project [34] focuses on the representation learning of protein sequences, as an approach for addressing the challenge of data scarcity in protein function prediction. This project considers the problem of assisting protein engineering with deep learning. The computational question here can be formulated as a regression task: given a protein sequence, a deep learning model is used to predict the function levels (quantified by a numeric value) of the input protein. While having a simple setup, this problem presents several unique challenges: the predictive model needs to capture the non-independent mutational effects (epistasis) in sequences and should be able to generalize to unseen sequences as well. While those challenges could be addressed if we train the model using massive training data, in protein engineering, we often have limited function readouts as it is costly to carry the experiment to generate those labels. Here, we sought to incorporate related data, including evolutionarily related sequences and large-scale unlabeled sequences, to learn better representations of protein sequences. The evolutionary information, in particular, has been shown to correlate with epistasis of function. Therefore, encoding the evolutionary signal into the protein sequence representation potentially can inform the supervised learning process of function prediction. We used a Markov random field (MRF) to capture the single-site preference and co-evolution relationships within protein sequences. Learned parameters of the MRF were used as representations to encode protein sequences. We further integrated

representations that capture the global semantics of protein sequences by training a protein language model on millions of protein sequences. We develop ECNet, a deep learning model, to integrate those two representations to assist protein engineering [34]. We showed that ECNet predicts the sequence-function relationship more accurately compared to existing machine learning algorithms by using ∼50 deep mutagenesis scanning and random mutagenesis datasets. Moreover, we used ECNet to guide the engineering of TEM-1 β-lactamase and identified variants with improved ampicillin resistance with high success rates.

My second project [35] pivots into learning representations of protein structures for improving the prediction of kinase-drug binding affinity. Mutations in kinase proteins are related to many complex human diseases such as cancers and kinases are the primary targets of a wide range of compounds. However, due to the high structure similarity between kinases, the specificity of many existing drugs is limited, meaning that a drug may bind to another kinase that is not the primary target by design. Therefore, inferring the binding profile of a kinase has important implications in biomedicine. Developing computational methods for predicting protein-compound binding has been studied for years, especially in very recent years when deep learning has been making rapid progress. Many deep learning algorithms have been developed to learn features from raw data representations, such as protein sequence and molecule SMILES strings, for binding prediction. However, while the binding activity happens as a process in the 3D space, few methods have considered exploiting information in 3D structure data of proteins and compounds. Here, we leveraged state-of-the-art graph neural networks that can model the 3D structure data in an effective way to develop a deep learning algorithm called KDBNet for predicting binding affinity from structure data. For example, for protein structures, the graph neural networks capture the geometric features of the binding pocket, including the dihedral angles of the protein backbone, the local orientation frame of residues, and the distance and direction between atoms. By modeling the spatial data, KDBNet learns more informative features to predict the binding affinity. We further used an ensemble approach to estimate the uncertainty for every model prediction. Experiment results demonstrated that KDBNet, by integrating structure data, more accurately predicts the binding affinity than models that did not consider structure data. The uncertainty estimation also enabled KDBNet to prioritize leading kinase-drug pairs that have strong binding affinities.

My third project [36] aims to integrate large-scale heterogeneous networks using representation learning to predict drug-target interactions (DTIs), which is a key question in accelerating drug discovery with computational methods. The problem is formulated as a link prediction task in networks, where nodes represent entities such as drugs, proteins, diseases, and side-effects, and edges represent physical interactions, medical associations, or

similarity relationships between those entities. As this network data is high dimensional and noisy, we took a dimensionality reduction approach and used representation learning to obtain a low-dimensional vector to represent each node in the network, such that topologically similar nodes have similar embedding vectors. We also extended this algorithm to integrate multiple heterogeneous networks that share the same set of nodes but have different edges in individual networks. The representation learning algorithm was then coupled with a matrix completion algorithm to predict the interactions between drugs and target proteins. We found that this method, called DTINet, has superior integration ability and improves the prediction accuracy compared to several existing methods [36]. We applied DTINet to predict interactions that have not been reported in current databases to augment the current knowledge of DTIs. Furthermore, DTINet identified novel interactions between three drugs and two COX proteins, which were then been validated in experimental assays. We then extended DTINet to integrate protein sequence features and chemical structure features using recurrent neural networks and graph neural networks. A preliminary version of this new model has been the winning algorithm in the DREAM Challenge for Drug-Kinase Binding Prediction [37]. Recently, we also extended DTINet to a more comprehensive pipeline to repurpose potential drugs for SARS-CoV-2 [38].

I have completed a few other research projects during my Ph.D, which were excluded in this dissertation to keep it succinct and coherent. The excluded projects were also closely related to the scope of this dissertation, including representation learning for protein and DNA sequences [39, 40, 41, 42, 43], protein or other molecule structures [44, 45], and biological networks [46, 47, 48]. New methods developed in those projects have been applied in important scientific questions such as drug response prediction [49] and drug repositioning [38].

1.3 BROAD IMPACTS

Building connections between machine learning and biology Broadly, my thesis research focuses on developing machine learning algorithms for biological problems such as those related to system biology and human diseases. While machine learning, especially deep learning, is extensively used to study biology in the past few years, most of the existing work only applied machine learning as an off-the-shelf toolbox. My research, however, identifies specific challenges and key issues in problems of interests and develops domain-tailored machine learning models to address them. Examples include our novel machine learning algorithms to integrate large-scale heterogeneous data [36] and to incorporate prior knowledge to improve accuracy and generalizability [34]. These projects represent a step towards closing the gap between what can be solved by existing mahcine learning algorithms and the

actual challenges and problems in biology.

Learning transferable representations and models A particular focus of my thesis research is developing algorithms to learn transferable representations and machine learning models. This is motivated by the fact that labeled data are often expensive to generate in the laboratory, and that typically there is only limited data to train machine learning models. I translated the research advances of machine learning into solutions for several protein binding prediction problems. Those techniques, such as meta-learning and self-supervised learning, effectively mitigate the data scarcity issues in those protein binding problems. Furthermore, my solutions have been shown to be generic and applied in other biomedical problems as well, such as predicting drug response across biological contexts using meta-learning [49].

AI-assisted scientific discovery Some of my research have been applied in real-world applications to assist scientific discovery. For example, my DTINet algorithm for predicting drug-target interaction has been used to repurpose potential drugs for inflammatory diseases [36] and COVID-19 [38], and my ECNet algorithm for protein function prediction has been integrated to a protein engineering pipeline and successfully engineered several variants of TEM-1 protein with improved function [34].

1.4 ROADMAP OF DISSERTATION

In this dissertation, I will present my research on representation learning of protein sequences, structures, and networks, and the scientific discovery enabled by the algorithms I developed. Chapter 2 introduces a sequence representation learning framework that integrates evolutionary context of protein sequences, which discovered novel protein variants with enhanced function. Chapter 3 presents a new geometric deep learning algorithm that learns representations of protein structure data for improving the prediction of protein-compound binding affinity. Chapter 4 describes a representation learning method for integrating heterogeneous networks, which repurposed existing drugs for new targets. Chapter 5 concludes this dissertation with a brief discussion of future directions.

Chapter 2: Representation Learning of Protein Sequences

A protein is made up of a linear chain of amino acids. It is widely believed that the sequence of a protein contains the information that encodes its structure and function, and many computational methods have been developed to study protein structure and function using a sequence-first approach. For example, the AlphaFold algorithm predicts the three-dimensional structure of a protein from its sequence [50]. In this chapter, we focus on the representation learning of protein sequences. Traditionally, sequence data is represented using simple encoding strategies such as one-hot encoding in popular machine learning applications. However, these representations have many limitations in biological problems. The one-hot encoder puts equal weights on all amino acids, which fails to reflect the unique property, e.g., evolutionary conservation, of an individual amino acid at a particular position. We demonstrate in this chapter how to learn biology-inspired sequence representations using protein engineering as an example, where the goal is to optimize the function of a protein by introducing mutations.

2.1 INTRODUCTION

Protein engineering aims to create protein variants with improved or novel functions. One powerful protein engineering strategy is directed evolution, which consists of iterative cycles of mutagenesis and high-throughput screening or selection [51, 52, 53]. While directed evolution is highly successful, the protein sequence space that can be sampled by directed evolution is limited and developing an effective high-throughput screening or selection can require a significant experimental effort [54].

To address these limitations, machine learning (ML) algorithms have been developed to assist directed evolution, which led to many successfully engineered proteins [54, 55, 56, 57, 58, 59]. In ML-assisted directed evolution, a machine learning model is trained to learn the sequence-function relationship from sequence and screening data. In one round of directed evolution, the model simulates and predicts the fitness of all possible sequences, and a restricted list of best-performing variants is used as the starting point for the next round of directed evolution. In contrast to the classical directed evolution, ML-assisted directed evolution can escape from the local optimum by learning the entire functional landscape from data. It takes full advantage of all available sequence and screening data, including those of unimproved variants, thereby traversing the fitness landscape more efficiently.

A critical component of ML-guided directed evolution is to build a machine learning al-

gorithm that accurately maps sequence to function. Unlike the qualitative predictions that group protein sequences into different functional classes [60, 61, 62, 63], in protein engineering, a model is required to distinguish quantitative functional levels of closely related sequences. For example, in one round of directed evolution, the ML model needs to predict the fitness of a sequence that differs from the parent sequence by only one or very few single amino acids. Several ML algorithms have been developed to predict the mutational effects by leveraging the evolutionary information of homologous sequences [64, 65]. These methods built generative models to reveal the underlying constraints of the evolutionary process, which can then be used to infer which mutations are more tolerable or favorable than others. Because of the unsupervised nature, however, these methods are not able to leverage the fitness data of tested variants available during the directed evolution process and thus may have limited accuracy when guiding the protein engineering. More recently, inspired by the advances in natural language processing [17], an emerging trend is to pretrain a language model (LM) on large protein sequence datasets to learn the distribution of protein sequences [21, 22, 23, 63, 66, 67, 68, 69]. The protein sequences observed in nature today are the results of natural selection by evolution. Out of the possible mutations to a sequence, evolution samples those that preserve or improve the protein's fitness, such as stability, structure, and function. The underlying constraints or factors that determine protein's fitness have shaped the distribution of protein sequences. LMs are used to unravel the 'grammars' or 'semantics' of sequence generation by evolution. By being trained on natural sequences to predict the likelihood that a particular amino acid appears within a context, the language model learns representations that are semantically rich and encode structure, evolutionary and biophysical contexts [66]. Several recent studies found that the representations learned by LMs can be used to predict the sequence-function relationship in an unsupervised way [70, 71, 72]. It was also found that using the learned representation as the feature input to fine-tune a supervised model improves fitness prediction on multiple protein mutagenesis datasets [67]. However, as these models are trained on massive sequences such as those in UniProt [73] and Pfam [74], the learned representations only capture general context for a wide spectrum of proteins but may not be specific to the protein to be engineered. Lacking this specificity in the representation, the prediction model may not be effective in capturing the underlying mechanism (e.g., epistasis between residues) that determines the fitness of a protein and is not able to effectively prioritize best-performing variants to assist the directed evolution.

In this work, we developed ECNet (evolutionary context-integrated neural network), a deep learning model that guides protein engineering by predicting protein fitness from the sequence. We constructed a sequence representation that incorporated the local evolutionary

context specific to the protein to be engineered. This representation explicitly encodes the residue interdependencies of all residue pairs in the sequence, which informs our prediction model to quantify the effects of mutations – especially higher-order mutations – in the sequence. We further incorporated global evolutionary context from an LM model trained on large sequence databases to model the semantic grammar within protein sequences as well as other structure and stability relevant contexts. Finally, a recurrent neural network model, trained on the fitness data of screened variants, is used for the sequence-to-function modeling with both representations. Through extensive benchmarking experiments, we showed that ECNet outperforms existing methods on 50 deep mutagenesis datasets. Further experiments on combinatorial mutagenesis datasets demonstrated that ECNet enables generalization from low-order mutants to higher-order mutants. Moreover, ECNet was successfully used to engineer TEM-1 β-lactamase variants with improved resistance to ampicillin.

2.2 ECNET: EVOLUTIONARY CONTEXT-INTEGRATED DEEP SEQUENCE MODELING FOR PROTEIN ENGINEERING

2.2.1 Residue co-evolution correlates protein functional fitness

Mutations within the protein sequence can affect fitness in a non-independent way, which is also known as genetic interactions or epistasis. It was found that epistasis interactions, quantified by deep mutational scanning (DMS) of proteins, can be used to infer protein contacts and structures [75, 76]. As structurally proximal protein residues are often inferred from co-variation pairs from sequence evolution historically [77, 78], we hypothesized that co-evolution information can also be used to infer epistasis or fitness of proteins.

To test this hypothesis, we investigated the relationship between the co-evolution of residue pairs and the fitness of double mutants. We collected a DMS study that measured the fitness of double mutants of the human YAP65 WW domain [79]. We also quantified the strength of pairwise residue dependencies by fitting a direct coupling analysis model [80] to the homologous sequences of the WW domain (see the "Methods" section). We found that the strength of pairwise dependencies correlated with the fitness of double mutants (Spearman correlation 0.35; Fig 2.1a). Similar to a previous study [64], we also used the change of dependency strength (by contrasting the mutant sequence to the wild-type sequence) to predict the fitness of protein variants in a set of DMS studies [81]. We found that the predictions correlated with experimental data with a Spearman correlation ranging from 0.1 to 0.5 (Figure 2.1b). In addition, we observed a trend of increasing correlation score if a protein has more homologous sequences, presumably because abundant homologous

sequences lead to a more accurately fitted direct coupling analysis model. Overall, these results suggested that there are signals in the evolution information that we can leverage to predict protein fitness. This motivated us to integrate evolutionary information of protein sequences to empower a supervised model that predicts the fitness of protein variants in directed evolution.

2.2.2 Sequence-to-function modeling

We built a deep learning sequence-to-function model, ECNet, that learns the mapping from protein sequences to their respective functional measurements (Figure 2.1c, Supplementary Figure B.1) from data (e.g., fitness measured by deep mutational scanning). We used the LSTM neural network architecture and trained protein-specific models using large-scale deep mutational scanning datasets ("Methods").

Our model is mainly empowered by two informative protein representations, with one accounting for residue interdependencies of the specific protein of interest and the other capturing the general sequence semantics in the protein universe. Existing tools predict the conservation effects of mutations by considering each amino acid independently (e.g., PolyPhen-2 [82] and CADD [83]) while others exploit structure information (e.g., FoldX [84] and OSPREY [85]). However, the functions of proteins are often driven by the interdependencies between residues (e.g., epistasis) in the protein [86, 87], and not all the protein structures are solved. We thus explicitly modeled the pairwise interactions of all pairs of sites in a protein by extracting signals of evolutionary conservation from its homologous sequences or sequence families. We used a generative graphical model, fitted on the multiple sequence alignment (MSA) of the homologous sequences, to uncover the underlying constraints or interdependencies that define the family of homologous sequences. These constraints are the results of the evolutionary process under natural selection and may reveal clues on which mutations are more tolerable or favorable than others. The generative model generates a sequence $\boldsymbol{x} = (x_1, \ldots, x_L)$ with probability $p(\boldsymbol{x}) = \exp[E(\boldsymbol{x})]/Z$, where $E(\boldsymbol{x})$ is the 'energy function' of sequence $\mathbf{x}$ in the generative model and Z is a normalization constant. We applied CCMpred [80], which is based on a Markov random field (MRF) specification to model the residue dependencies in protein sequences. The energy function $E(\boldsymbol{x})$ of sequence $\boldsymbol{x}$ is defined as the sum of all pairwise coupling constraints $\boldsymbol{e}_{ij}$ and single-site constraints $\boldsymbol{e}_i$, where i and j are position indices along the protein sequence,

$$E(\boldsymbol{x}) = \sum_i \boldsymbol{e}_i(x_i) + \sum_{i \neq j} \boldsymbol{e}_{ij}(x_i, x_j). \tag{2.1}$$

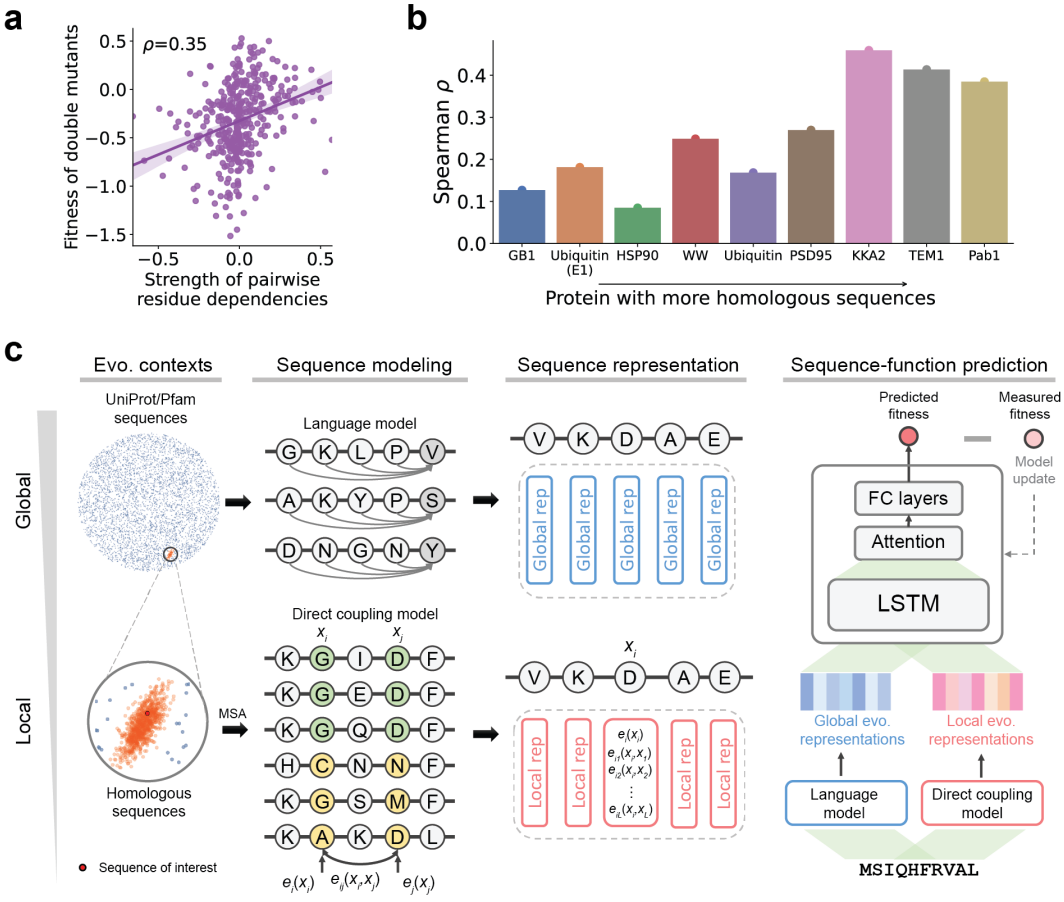


Figure 2.1: The motivation and overview of our evolutionary context-integrated sequence modeling method for protein engineering. (a) Sequence co-evolution data correlates with fitness measurements in deep mutational scanning studies. The scatter plot shows the relationship between the fitness measurement of double mutants and the co-variation strength of residues where the mutations were introduced. Each data point represents a double mutant. The error band indicates the 95% confidence interval of the regression line. (b) Sequence co-evolution data can be used to predict protein fitness. The bar plot shows the Spearman correlation between experimentally measured fitness and strength changes of co-variation. Proteins were sorted by the number of homologous sequences. c An overview of ECNet, our evolutionary context-integrated deep learning framework for protein engineering. ECNet integrates global and local evolutionary contexts to represent the protein sequence of interest. First, a language model is used to learn global semantic-rich global sequence representations from the protein sequence databases such as UniProt or Pfam. Next, a direct coupling analysis model is used to capture the dependencies between residues in protein sequences, which encodes the local evolutionary context. The global and local evolutionary representations are then combined as sequence representations and used as the input of a deep learning model that predicts the fitness of proteins. Quantitative fitness data measured by deep mutational scanning (DMS) are used to supervise the training of the deep learning model (MSA: multiple sequence alignment; Dim. reduction: dimensionality reduction; LSTM: long short-term memory network; FC layers: fully-connected layers; Evo. contexts: evolutionary contexts; Evo. representations: evolutionary representations; Global/Local rep: global/local representation).

When the MRF model is fit to data with proper regularizations, the residue interactions in protein sequences are explained by the direct coupling terms $\boldsymbol{e}_{ij}$. It has been shown that the magnitudes of $\boldsymbol{e}_{ij}$ terms can accurately predict protein contacts [26] and 3D structures [88]. For a protein sequence with length L, we encoded its i-th amino acid $\boldsymbol{x}_i$ by a vector, in which elements were set to the single-site term $\boldsymbol{e}_i(\boldsymbol{x}_i)$ and pairwise coupling terms $\boldsymbol{e}_{ij}(x_i, x_j)$ for $j = 1, \ldots, L$ (Figure 2.1c), and then dimensionality reduction techniques were used to project it into low rank ("Methods"). Encoding the protein sequence in this way directly incorporates the protein's evolutionary context, i.e., the effects of pairwise epistasis, which can inform machine learning models to predict the fitness of a sequence with single or higher-order combinatorial mutations.

In addition to the evolutionary sequence contexts specific to the protein of interest, global protein sequence contexts, i.e., those encoding structures and stabilities, can also inform our prediction model to predict the effects of mutations. For this purpose, we integrated general protein sequence representations from unsupervised protein contextual language models [21, 63, 66, 67]. Using a large corpus of protein sequences such as UniProt and Pfam, a language model learns to predict the likelihood of a particular amino acid appearing at a position given all other amino acids surrounding it as context. During the training, the language model gradually changes its internal dynamics (encoded as hidden state vectors) to maximize the prediction accuracy. It was found that a wide range of protein-relevant scientific tasks, including secondary structure prediction, contact prediction, and remote homology detection, can be improved by using the hidden state vectors of a language model as input features to fine-tune a supervised model for the specific task [21, 67]. Here, we also used the language model's hidden state vectors as another type of protein sequence representation for our prediction model to capture the global protein sequence context (Figure 2.1c; "Methods"), which is a complement to our local evolutionary context representation.

The local and global evolutionary representations are jointly used to model the protein sequence of interest. A deep learning model (recurrent neural network) then takes these sequence representations as input and learns the sequence-to-function relationship. Quantitative functional measurements (e.g., fitness data measured by DMS) are used to supervise the training of the deep learning model (Supplementary Figure B.1; "Methods").

2.3 METHODS

2.3.1 Datasets

We collected multiple large-scale deep mutational scanning (DMS) datasets and random mutagenesis datasets curated by previous publications.

Envision dataset. We first collected 12 DMS studies from Gray et al. [81], covering ten proteins and 28,545 fitness measurements of single amino acid variants. The fitness values were normalized such that wild-type-like variants have scores of one, and variants that are more (less) active than the wild type have scores greater (less) than one.

DeepSequence dataset. We also collected a set of DMS datasets compiled by Riesselman et al. [65]. We excluded a study of RNAs since it is out of the scope of this study. The resulting set consists of 39 DMS studies across 33 proteins. Most of these studies (37/39) provide the function values of single amino acid variants, and two studies provide the functional measurements of higher-order mutants. The functions measured in these studies include growth rate, enzyme function, protein stability, and peptide binding.

Single and double mutants datasets. To test the ability of ECNet to predict epistasis, we compiled multiple DMS studies that contain the fitness values of both single and double amino acid variants. We obtained the DMS data of the GB1 domain, WW domain, RRM domain, and FOS–JUN heterodimer from Rollins et al. [76], and the prion-like domain of TDP-43 from Bolognesi et al. [89]. A set of fitness of TEM-1 consecutive double mutants was also obtained from Gonzalez et al. [90].

Higher-order avGFP mutants dataset. We also collected a higher-order mutant dataset [91] to assess ECNet's generalizability to predict the effect of even higher-order variants. This study systematically assayed the local fitness landscape of the green fluorescent protein from Aequorea victoria (avGFP) by measuring the fluorescence of 50k derivatives of avGFP, with each sequence containing 1–15 amino acid substitution mutations.

Inhibitor-resistant TEM-1 variants. We compiled a list of TEM-1 variants that have been found to be inhibitor-resistant with supporting evidence in previous studies. The list was downloaded from `https://externalwebapps.lahey.org/studies/TEMTable.aspx` (see "Data availability" for accession). We excluded variants for which mutation information was labeled as "Not yet released". This resulted in 146 sequences that mostly contained two to five and up to ten mutations (average 3.3 mutations per sequence). To generate a list of random candidate variants for enhanced TEM-1 variants prioritization, we enumerated all combinations of amino acid mutations on all or a subset of the positions where mutations were introduced in this 146-sequence list. In total, we obtained 18,937 randomly generated candidate variants for TEM-1 variants prioritization.

Homologous sequences and fitness data of viral proteins. We used the homologous sequences of each viral protein collected in Hie et al. [71] as the training data of the unsupervised ECNet, including 44,851 unique influenza A hemagglutinin (HA) amino acid sequences observed in animal hosts, 57,730 unique HIV-1 Env protein sequences, and 4172 unique Spike and homologous protein sequences. We used the fitness data collected in Hie et al. [71] to

validate the unsupervised ECNet. The fitness data includes replication fitness of HA H1 WSN33 mutants from Doud and Bloom [92], replication fitness of six HA H3 strains (Bei89, Bk78, Bris07, HK68, Mos99, and NDako16) from Wu et al. [93], replication fitness of HIV Env BF520 and BG505 mutants from Haddox et al. [94], and K_d binding affinities between SARS-CoV-2 mutants and ACE2 from Starr et al. [95].

2.3.2 Inference of evolutionary couplings from multiple sequence alignments

We first searched homologous protein sequences of a given protein using HHblits available in the hh-suite [96]. We used the wild-type sequence of the given protein as the query sequence and searched against the uniclust-30 database (version uniclust30_2018_08) for three iterations. We used a maximum pairwise sequence identity of 99% and a coverage cutoff of 50%. Other parameters were set as default. The search results were formatted to the A3M multiple sequence alignment (MSA) format.

To identify the co-evolutionary residue pairs in a protein, we used a statistical model to exploit the evolutionary sequence conservation and model all pairwise interdependencies of residues. The model identifies the evolutionary couplings by learning a generative model of the MSA of homologous sequences using a Markov random field. Given the MSA of homologous sequences, the couplings are learned by maximizing the likelihood of observed sequences in the MSA, which is defined as

$$L(\boldsymbol{e}) = \frac{1}{Z} \prod_{n=1}^{N} \prod_{i=1}^{L} \left[\exp\left(\boldsymbol{e}_i(\boldsymbol{x}_i^n) + \sum_{j=1, j\neq i}^{L} \boldsymbol{e}_{ij}(\boldsymbol{x}_i^n, \boldsymbol{x}_j^n) \right) \right] \tag{2.2}$$

where the single-site constraints $\boldsymbol{e}_i$ and the pairwise coupling constraints $\boldsymbol{e}_{ij}$ are parameters of the model, $\boldsymbol{x}_i^n$ is the i-th amino acid in the n-th sequence, Z is the normalization constant, N is the number of homologous sequences and L is the number of columns in the MSA (number of amino acids in the query sequence). The direct optimization of this likelihood is computationally intractable due to the computation of the normalization constant that increases exponentially—20^L sequences need to be considered. It was thus adopted to maximize the site-factored pseudo-likelihood of the MSA, which has a running time complexity $O(NL^2)$ where N is the number of sequences in the MSA. We refer the interested readers to previous studies [80, 97, 98] for the details of the optimization. In this work, we used CCMPred [80], a GPU-based algorithm maximizing the pseudo-likelihood (plus regularization terms), to optimize the generative model. The evolutionary couplings are learned as parameters of the Markov random field.

2.3.3 Local evolutionary context representation with evolutionary couplings

By fitting the graphical model to the MSA of homologous sequences of a protein, we obtained the coupling matrix $\boldsymbol{e}_{ij}$ that quantifies the co-constraints of all possible 20^2 amino acid combinations between positions i and j in the sequence. In particular, the term $\boldsymbol{e}_{ij}(x_i, x_j)$ is the pairwise emission potential of the Markov random field for amino acid x_i occurring at position i while amino acid x_j occurring at position j. We used the site preference vector $\boldsymbol{e}_i$ and the coupling matrix $\boldsymbol{e}_{ij}$ to construct a data representation that encodes the co-evolution information of a protein.

Specifically, the i-th amino acid x_j in the protein was represented by an $(L+1)$-long 'local evolutionary representation':

$$\boldsymbol{v}_i = [\boldsymbol{e}_i(x_i), \boldsymbol{e}_{i1}(x_i, x_1), \boldsymbol{e}_{i2}(x_i, x_2), ..., \boldsymbol{e}_{iL}(x_i, x_L)] \tag{2.3}$$

The full representation of a protein sequence was thus obtained by stacking local evolutionary representations for all positions, resulting in an L by $(L+1)$ matrix. As we have shown, the pairwise potentials in the matrix $\boldsymbol{e}_{ij}$ correlated with the fitness measured in DMS experiments (Figure 2.1a, b). We thus expect that using the local evolutionary representations derived from $\boldsymbol{e}_i$ and $\boldsymbol{e}_{ij}$ as a data representation of amino acids will inform the sequence-to-function prediction model to better capture the residue dependencies and the sequence-to-function relationship.

The length of the local evolutionary representation is roughly equal to the length of the protein sequence, which may raise an overfitting issue when the protein length is long while the number of functional measurements used as training data is low. Therefore, we used a dimensionality reduction approach to transform the $(L+1)$-long vector into a fixed-length d-dimensional vector ($d < L$), where d is independent of the length of the protein sequence. This is done by applying a linear layer in the neural network to reduce the dimensionality of local evolutionary representations $\boldsymbol{v}_i$. Hereinafter, we will refer to the transformed vector $\boldsymbol{v}_i$ as local evolutionary representation unless otherwise specified.

2.3.4 Pre-trained protein sequence representation model

Very recently, self-supervised models have provided powerful protein sequence representations that facilitate scientific advances, including protein engineering, structure prediction, and remote homology detection. These language models [21, 63, 66, 67], without using labeled data, are trained on natural sequences from large protein databases such as Pfam [74] and UniProt [73] to predict the next amino acid character given all previous amino acid

characters in the protein sequence or predict randomly masked amino acids using the rest as given context. During the model training, these models progressively adapt their parameters to maximize the prediction accuracy, resulting in a representation of protein sequences that capture intrinsic semantics in protein sequences and interdependencies among amino acids.

In this work, we integrated the amino acid representations produced by a transformer model in TAPE, one of the most powerful self-supervised sequence representation models [21]. The representations capture the global evolutionary context from the massive protein sequence data the model was trained on, which is complementary to our evolutionary representations that capture the local evolutionary context specific to the target protein. The TAPE model applied a Transformer architecture [99] and was trained on Pfam data to predict a masked amino acid using the remaining ones as input. We downloaded the pre-trained weights of the TAPE model from `https://github.com/songlab-cal/tape`. For an input sequence, TAPE generates a 768-dimensional vector representation for each amino acid. We refer to the reprojected TAPE representations as global evolutionary representations.

2.3.5 Sequence-to-function neural network model

Model architecture. We built a deep learning model for the sequence-to-function prediction. The model receives as input features (amino acid characters and evolutionary representations) of the protein sequences and produces the predicted functional measurements of proteins as output. The backbone of our model is a bidirectional long short-term memory network (BiLSTM) [100] integrated with a two-layer fully-connected neural network. Hyperparameters of the model were decided through a grid search in an independent experiment (see "Training details"). Amino acids in the input sequences were one-hot encoded and passed through a 20-dimensional embedding layer. The amino acid embeddings were then concatenated with the evolutionary representations position-wisely before being input to the LSTM module. We used a single-layer LSTM with a hidden dimension d_L as the default setting in this work. One hidden state vector was produced by the LSTM for every amino acid in the sequence. To integrate the TAPE representations into our model, we reprojected them to d_p-dimensional vectors using a linear fully-connected layer, which were then concatenated with the hidden state vector produced by the LSTM model for each amino acid. We summarized these concatenated vectors into a single vector using a weighted averaging approach, where the averaging weights were learned from the data by using a self-attention layer [99]. This vector was then passed to a top module to predict the functional measurements. The top module is a two-layer fully-connected neural network with tanh activation. The hidden dimensions of the two layers were set to d_h and 1, respectively. To facilitate

the model training, we added a batch normalization layer [101] before the fully-connected layers. We also applied a dropout [102] layer after the first fully-connected layer to prevent overfitting. To improve the model's robustness and prediction accuracy, we used an ensemble approach to output the prediction, in which three replicas of ECNet models were trained using the same hyperparameters and training data, and their output scores were averaged as the final prediction.

Training details. We cast the task of predicting the functional values of proteins as a regression problem, and the objective was to minimize the difference between the predicted and experimentally measured functional values. We trained our deep learning model using the Adam optimizer [103] with default parameters. Mean squared error (squared L_2 norm) was used as the loss function. To select the hyperparameters of ECNet, we performed a small-scale grid search using the training data of a protein, such that 7/8 of the training data was used to train a model with a specific set of hyperparameters, and the remaining 1/8 data was used as the validation set to select the hyperparameters. The test set was not used for hyperparameter selection. We tested the LSTM's dimension of $d_L = 32, 64$, and 128, the top layer dimension of $d_h = 32, 64$, and 128, the reprojected embedding dimension of $d_p = 128$ and 256. In general, we found that $d_L = 128$, $d_h = 128$, and $d_p = 128$ are reasonably good defaults and can be used for a new protein. Nevertheless, a careful grid search of hyperparameters for the new protein would further improve the model performance. Unless otherwise specified, the batch size was set to 128 and the maximum number of training epochs was set to 2000 with an early stop if the performance has not been improved for 1000 epochs. Model training was performed on an Nvidia TITAN X GPU. The time required to train a single model depends on the training data size of each protein, ranging from 0.5 to 6 h. For the ensemble model with three replicas, the required time thus ranges from 2 to 20 h.

Auxiliary classification objective. While the prediction of functional measurements is a regression problem by definition, the skewed distribution of the training data may lead to a biased predictor. For example, in the Envision dataset [81], only 18% of TEM-1 variants are more active than the wild-type sequence (positive effects) while the remaining are less active than the wild-type sequence (negative effects). In this case, a model optimized using a regression objective (e.g., minimizing the mean squared error) tends to fit the negative effects more but be less sensitive to the error from the prediction of positive effects. However, the main goal of machine learning-guided protein engineering is to identify the variants with an enhanced property than the wild-type sequence. Hence, it is critical to mitigate this type of bias in the prediction model. We addressed this issue by introducing an auxiliary classification objective. We binned the functional measurements using their 10-quantiles as

breakpoints, i.e., grouping the measurements into 10 bins with equal size. In the model training, we encouraged the model to accurately predict not only the absolute functional measurement but also which bin the measurement is in. Jointly, the classification objective forces the model to treat each bin of functional measurements equally and the regression objective forces the model to predict the measurements as close to the observed values as possible. In the implementation, we added a second top module into the deep learning model, which also receives the summarized LSTM hidden state vector as input and its output is ten numbers indicating the predicted probability that the measurement should fall in each of the bins. The overall loss function is $L = L_r + \alpha L_c$ where L_r is the loss of the regression objective, L_c is the cross-entropy loss of a ten-class classification, and α is a constant used to balance the scales of the two losses, which was set as $\alpha = 0.1$ in this work. We used this hybrid loss when training the model for prioritizing novel TEM-1 variants and used the regression loss for other benchmarking experiments.

2.3.6 Unsupervised ECNet based on language model training

While the vanilla ECNet is a supervised model and requires function or fitness data to train the predictor, we also developed an unsupervised extension of ECNet that does not need any direct fitness measurements as training data but is still able to produce reasonably accurate predictions. This unsupervised model is useful when the fitness data of a protein is unavailable or not sufficient to train an accurate supervised predictor. The predictions of the unsupervised ECNet can be used as an approximation of fitness and guide the selection of variants to screen in the first round of directed evolution, after which the experimental screening data can be used to train a more accurate, supervised ECNet model.

The main idea of the unsupervised ECNet is to train a model that learns the evolutionary preferences from the homologous sequences of the protein of interest. Those homologous sequences are the results of long-course evolution and might reveal evolutionary preferences about which mutations are more viable or tolerable than others. This approach is motivated by the recent advances of deep learning for human languages, in which algorithms called language models are developed to learn intrinsic semantics and grammar constraints of natural languages like English from large text corpora.

The model architecture of the unsupervised ECNet is also based on a bidirectional LSTM (BiLSTM), as in the supervised ECNet, but with a different training objective. Here, we use an objective similar to that used in Hie et al. [71] to train a protein language model. Precisely, we are given a protein sequence $\boldsymbol{x} = (x_1, \ldots, x_L)$, $x_i \in \mathcal{X}$,$i \in [L]$, where $\mathcal{X}$ is the alphabet of all possible amino acids. Let $\tilde{x}_i$ denote a point-mutation at position i and the

mutated sequences as $x(\tilde{x}_i) = (x_1, \ldots, x_{i-1}, \tilde{x}_i, x_{i+1}, \ldots, x_L)$. The language model aims to predict the probability of an amino acid appearing at a position considering its surrounding context, i.e., $p(x_i|x_{[L]\setminus\{i\}})$, where $x_{[L]\setminus\{i\}} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_L)$ represents the sequence context. The context is encoded using a latent real-valued vector $\boldsymbol{z}_i = f_e(x_{[L]\setminus\{i\}})$, where $f_e : \mathcal{X}^{L-1} \to \mathbb{R}^D$ is an embedding function that maps discrete sequences into a D-dimensional continuous space. Here the embedding function was instantiated by a bidirectional LSTM neural network and the outputs of the final LSTM layers were concatenated to form the embedding vector, i.e.,

$$\boldsymbol{z}_i = [\mathrm{LSTM}_f(g_f(x_1, \ldots, x_{i-1})); \mathrm{LSTM}_r(g_r(x_{i+1}, \ldots, x_L))] \tag{2.4}$$

where g_f is the output of preceding layers that proceed the input in the forward direction, LSTM_f is the final layer of the forward-directed LSTM, and g_r and LSTM_r are defined similarly but for the reverse direction. The embedding vector $\boldsymbol{z}_i$ is transformed into a probability through a learner transformation and a softmax function, i.e.,

$$p(x_i|x_{[L]\setminus\{i\}}) = p(x_i|\boldsymbol{z}_i) = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{b}) \tag{2.5}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are learned parameters. We used a two-layer BiLSTM with 256 units in this work.

We demonstrated the utility of the unsupervised ECNet on viral proteins. We trained three unsupervised ECNet models for influenza HA, HIV Env, and SARS-CoV-2 Spike proteins using the unsupervised ECNet model. One epoch in the training consisted of the prediction of every token at all positions and in all sequences in the training set. The output probability $p(x_i|x_{[L]\setminus\{i\}})$ is used as the predicted score and to correlate with the fitness score of a mutant. If a variant has multiple mutations, the product of the probabilities of the individual point mutations was used as the predicted score of unsupervised ECNet.

2.3.7 Baseline methods

We compared ECNet against several existing baseline methods, including supervised and unsupervised models.

Yang et al. (Doc2Vec). Yang et al. [104] proposed a learned protein embedding to represent a protein sequence in a 64-dimensional vector using a Doc2Vec model [105] trained on the UniProt database. The representation vector is used as the input feature to fit a Gaussian process-based regressor to predict the functional measurement. Following a previ-

ous work [66], we also used the four best-performing models as chosen in Yang et al. [104], including the original model ($k = 3, w = 7$), the scrambled model ($k = 3, w = 5$), the random model ($k = 3, w = 7$), and the uniform model ($k = 4, w = 1$). The pre-trained models were downloaded from `http://cheme.caltech.edu/kkyang/models/` and protein representation vectors were generated using the code available at `https://github.com/fhalab/embeddings_reproduction`. The best performance across the four models was reported as the final performance of the Doc2Vec model.

Envision. Envision is a supervised method proposed in Gray et al. [81] that predicts the functional measurements of protein variants. Each variant was annotated with 27 biological, structural, and physicochemical features, which were used as input to train a gradient boosting regression model using large-scale mutagenesis data. We downloaded the source code of Envision from `https://github.com/FowlerLab/Envision2017`.

EVmutation. EVmutation is an unsupervised statistical model proposed by Hopf et al. [64]. It explicitly models the co-variations between all pairs of residues in the protein by fitting a pairwise undirected graphical model to the multiple sequence alignment (MSA) of all homologous sequences of the protein of interest. The model then quantifies the effect of single or high-order substitution mutations using the log-ratio of sequence probabilities between the mutant and wild-type sequences. In this work, we used the workflow implemented in EVcouplings (`https://github.com/debbiemarkslab/EVcouplings`) to generate the predictions of EVmutation.

DeepSequence. Similar to EVmutation, DeepSequence [65] is also a generative model that predicts the effects of mutations in an unsupervised manner. However, unlike EVmutation explicitly modeling pairwise dependencies, DeepSequence uses a latent model, fitted on the MSA of homologous sequences of a protein, to capture higher-order dependencies of residues in the protein. The effects of mutations are also predicted by the log-ratio of mutant likelihood to wild-type likelihood.

Autoregressive model. Generative models of protein sequences such as EVmutation and DeepSequence are dependent on the alignment of homologous sequences, which may introduce artifacts and lose important information caused by indels in the alignment. A generative autoregressive model was proposed by Riesselman et al. [106] to predict the mutation effects in protein sequence, without the requirement of multiple sequence alignment.

TAPE. We used TAPE [21], a language model (LM) trained on Pfam sequences to generate global context representations of protein sequences. We extracted the hidden state vectors, one for each amino acid in the sequence, from the TAPE model. We used the same top module as in our model (i.e., self-attention layer and fully-connected layers) to take the representations as input and predict the functional measurements.

UniRep. UniRep [66] first trains an unsupervised protein language model on UniRef50 sequences. The model is then fine-tuned using homologous sequences of a studied protein (called evotuning). The model is used to generate a vector representation for each protein sequence. These representations are used as the input of a top supervised model such as a ridge or LASSO regression to predict the fitness of mutants.

CSCS. CSCS [71] is an unsupervised model that is specifically designed to predict viral escapes. It also trains a language model on viral protein sequences and computes two scores to quantify the effect of a mutation, one is the grammaticality of the mutation, defined as the model predicted probability of an amino acid a position in the sequence, and the other is the semantic change of the mutation, defined as the L1 distance between the embeddings of the mutated sequence and the wild-type sequence. In our experiment, we used the grammaticality of a mutant to correlate its fitness, as this was shown to outperform the prediction based on semantic changes in the CSCS study.

2.3.8 Benchmarking experiments

To assess ECNet's performance, we compared ECNet to other baselines using the original benchmark datasets that these methods were tested on in their publications. We ensured that the training and test sets are not overlapped in our experiments.

Benchmarks on the Envision dataset. We compared ECNet to the gradient boosting regression algorithm (denoted as 'Envision') proposed in the Envision dataset paper [81]. For each protein, we used 80% of the DMS data to train ECNet or other methods and the remaining 20% data to evaluate the model's performance. Spearman correlation was used as the evaluation metric. We used grid search to optimize the hyperparameters of ECNet. Note that Envision used 27 biological, structural, and physicochemical features to build the prediction model while our model only used the protein sequence to predict the functional measurements. To test the model's ability to identify variants that are more active than the wild-type sequence, we also converted the task into a classification problem, in which protein sequences with a function score greater than the wild-type sequence (with a function score 1) were labeled as positive samples, and the remaining sequences as negative samples. We used the AUROC score as the metric for this classification evaluation. We also compared ECNet to the Yang et al. (Doc2Vec) model on this dataset.

Benchmarks on the DeepSequence dataset. We compared ECNet to EVmutation, DeepSequence, and the Autoregressive model on the DeepSequence dataset that these methods have been tested on. The predictions made by these unsupervised approaches were collected from previous studies [65, 106]. For ECNet, we performed five-fold cross-validation on this dataset

and reported the average performance over all the five folds. Hyperparameters of ECNet were optimized using an inner-loop cross-validation. We used Spearman correlation as the evaluation metric. We also compared ECNet to supervised models that used global context representations (TAPE) or locally-fine-tuned global context representations (UniRep) on this dataset.

Evaluation of variants prioritization. We designed a simulation experiment of variants prioritization to assess how accurate and efficient ECNet is in retrieving high-performing variants [107]. We collected three large DMS datasets of proteins avGFP [91], GB1 [108], and Pab1 [109], each with the fitness data of single and high-order mutants. We trained the model using 90% of the data (randomly sampled) and asked the model to predict the fitness for the remaining 10% data. For comparison, we ran One-hot encoder, Evmutation, and UniRep to predict the same test variants. We also included the ideal model, which used the ground-truth ranking of fitness values to rank variants, and the null model, which ranked variants with a random order, as references for the evaluation. For each protein, we repeated the experiments ten times. We calculated the recall as the evaluation metric, which is defined as the fraction of true top 100 variants the model recovered in its list of top K predictions. This value of K can be interpreted as the sequencing budget in actual experiments of directed evolution. We also used the maximum fitness (normalized by rank) observed in the top K predictions as an additional metric to assess ECNet's ability in identifying the variants with the highest possible fitness. To demonstrate the efficiency of ECNet's prioritization, we computed its efficiency gain over random sampling as a function of budget K, which was quantified as the ratio between the recall of ECNet and the recall of the null model.

Benchmarks of unsupervised ECNet. We trained unsupervised ECNet models using the language model objective on the homologous sequences of three viral proteins, including influenza HA, HIV Env, and SARS-CoV-2 Spike, respectively. Hyperparameters of ECNet were optimized using an inner-loop cross-validation. The trained ECNet models were evaluated using fitness datasets for mutants of several proteins, including HA H1 WSN33, six HA H3 strains, BG505 and BF520 HIV Env, and SARS-CoV-2 Spike. The performance was evaluated using Spearman correlation between the output probability of a mutation given by unsupervised ECNet and the fitness score of that mutation. As our training objective followed that of CSCS, a protein language model developed to predict the escape of viral mutations, we compared our model to CSCS and validated that our model achieved a comparable performance as CSCS. For reference, we also trained a supervised ECNet on the viral proteins using five-fold cross-validation and found that the supervised model substantially improved the performance.

Effects of training data size. To investigate the effects of training data size, we randomly

withheld 10% of data of each DMS study in the DeepSequence dataset as the test set. For the remaining 90% data, we trained two separate ECNet models by using all of them as training data (denoted as 100%) or randomly sampling 1/4 of them as training data (denoted as 25%). We also trained an unsupervised ECNet model without using any DMS data (denoted as 0%). The three models were all evaluated on the same test set.

Comparison to randomly generated TEM-1 variants. We trained an ECNet model and used it to predict the fitness of 146 TEM-1 inhibitor-resistant variants and a set of randomly generated variants. For every of the 146 TEM-1 inhibitor-resistant variants, we generate ten random variants that have the same mutated sites as the inhibitor-resistant variant but the alternative amino acid at each position is re-sampled uniformly from the 20-amino acid set. We ensured that the generated random variants do not overlap with any of the 146 variants.

2.3.9 Prioritized high-order TEM-1 variants using ECNet

We trained an ECNet model using DMS data of low-order TEM-1 variants and used the model to prioritize new high-order variants that were likely to have enhanced fitness. We sourced the training data from Firnberg et al. [110] and Gonzalez et al. [90], which measured fitness values of 98.2% (2536/2583) of all possible point mutants and 12.0% (12,374/102,855) of all possible consecutive double mutants, respectively. In both studies, the fitness of TEM-1 was defined as its resistance to ampicillin (Amp). We randomly sampled 95% of the combined datasets to form the training set and used the remains as the validation set. We used the default hyperparameters mentioned above for ECNet. The model was trained for 2000 epochs with early stopping if its performance on the validation set did not improve for 1000 epochs.

We used the trained model to predict the fitness of variants beyond single and consecutive double mutants and to identify new variants with improved fitness (as compared to the wild type). To reduce the exponential search space (20^{286} sequences) of possible sequences, we focused on a restricted subspace where the mutations only occur on plausible function-related sites of TEM-1 documented in the literature. The detailed steps are as below. (1) We compiled a list of inhibitor-resistant TEM-1 variants supported by evidence in previous studies (see "Datasets"). The list contains 146 TEM-1 variants, each with 2–11 mutations with respect to TEM-1 wild-type sequence, covering 72 positions and 99 unique amino acid (AA) changes. (2) Based on each of the 146 variants, we generated new variants by considering all subset combinations of mutated positions of this variant and enumerating all AA changes that have appeared in these positions in the list. (3) From the newly generated sequences, we removed sequences that are identical to sequences in the 146-variant list,

resulting in 18,937 sequences that form our restricted search space. We call these sequences "candidate variants". (4) We applied the trained ECNet model to predict the fitness for each of the candidate variants. Variants predicted to have lower fitness than the wild-type fitness were removed. (5) We used FoldX38 to compute the change of structure stability for each variant. The PDB structure of wild-type TEM-1 was used as the template (PDB ID: 1XPB) and refined by the 'RepairPDB' function of FoldX. The 'BuildModel' procedure of FoldX was applied to mutate residues and compute the change of stability. Variants with a large change of stability ($|\Delta\Delta G| > 3$ kcal/mol) were removed. (6) The remaining variants are sorted based on their predicted fitness, and we refer to those variants as prioritized variants.

To generate prioritized variants for experimental validations, we built two versions of ECNet models, a base version (ECNet-base) and an ensemble version (ECNet-ensemble). In ECNet-base, we trained three independent predictors using the default neural network architecture of ECNet, but each with a different loss objective, namely regression loss, classification loss, and regression loss with an auxiliary classification loss, respectively. The intersection of variants prioritized by the three predictors formed the final prioritized variants of ECNet-base. The use of three loss objectives here follows the intuition that the regression loss encourages the model to approximate the fitness of all variants and the classification loss focuses the model on identifying variants with fitness higher than the wild type. Combining the three predicted lists can prioritize more reliable predictions. We selected the top 28 variants in the intersected list for experimental validations. In ECNet-ensemble, we followed the same procedure as in ECNet-base but trained five replicates of predictive models for each loss objective. The predicted fitness of a variant was averaged over the five replicates. From the prioritized list by ECNet-ensemble, we selected 9 variants that were ranked at the top but different from what have been selected from the list predicted by ECNet-base.

In total, we used ECNet to identify 37 TEM-1 variants that were likely to demonstrate improved fitness as compared to the wild type. These variants contained two to six mutations (average 3.02 mutations per sequence), covering 22 positions in the TEM-1 sequence.

2.3.10 Experimental validation of prioritized TEM-1 variants

Materials and general methods. Molecular biology reagents and chemicals were purchased from Fisher Scientific, Sigma-Aldrich, GOLDBIO, or New England Biolabs, Inc., unless specified otherwise. *Escherichia coli* DH5α (New England Biolabs, MA) was cultured in Luria–Bertani broth. DNA sequencing was performed at ACGT (Wheeling, IL). Primers were ordered from Integrated DNA Technologies (Coralville, IA) and listed in Supplementary

Data 1 of Luo et al. [34]. Plasmid pSkunk3-BLA was purchased from Addgene (plasmid 61531). PacBio Barcoded Universal Primers (Part Number: 101-629-100) was purchased from Pacific Biosciences (Menlo Park, CA).

TEM-1 mutant creation TEM-1 mutants were constructed by overlapping PCR using primers (Supplementary Data 1 of Luo et al. [34]) carrying targeted single or multiple mutation sites. Briefly, DNA fragments were PCR amplified by primers carrying targeted single or multiple mutation sites using pSkunk3-BLA plasmid as template and then gel purified. The purified DNA fragments were further fused by overlapping PCR to provide DNA fragments with complete TEM-1 gene fragments flanked by restriction enzyme (*BamH*I and *Spe*I) digestion sites. After gel purification, the fused DNA fragments and pSkunk3-BLA plasmid were digested by *BamH*I and *Spe*I. The digested TEM-1 gene with mutations and pSkunk3-BLA plasmid was gel purified and ligated by T4 DNA ligase and then transformed into *Escherichia coli* DH5α competent cells. The single colonies from the transformation plates were picked and cultured overnight at 37 °C. The mutation sites of mutants were confirmed by DNA sequencing using primers listed in Supplementary Data 1 of Luo et al. [34]. In some cases, the constructed TEM-1 plasmids were used as PCR templates for creating other variants.

Ampicillin resistance assay of TEM-1 mutants. The plasmids harboring the genes encoding wild-type TEM-1, positive controls, and ECNet's predicted variants were mixed with equal concentrations and transformed into *Escherichia coli* DH5α competent cells (six replicates). After incubation at 37 °C overnight, the colonies from each of the transformation plates were immersed by ice-cold LB medium which was further scratched and pooled, yielding 10 ml cell suspension in LB medium. Inoculated 0.5 mL of the cell suspension into 50 mL LB medium supplemented with streptomycin (50 μg/ml) to allow the OD600 of cultures to reach 0.5 with shaking at 37 °C. The cells were then washed twice by 1 volume of ice-cold 1X PBS and then resuspended in 1 volume of ice-cold 1X PBS. Finally, 100 μL cells were spread onto each of the freshly prepared LB agar plates with different concentrations of ampicillin (0, 300, 1500, and 3000 μg/ml), streptomycin (50 μg/ml), and IPTG (0.3 mM). The plates were incubated at 37 °C overnight. The colonies from each plate were then pooled and miniprepped to provide a plasmid mixture for each plate. The concentrations of the plasmids were determined by QubitTM 4 Fluorometer (Invitrogen). PCR amplification of the targeted region with the same amount of total plasmids from each plate as templates by using target-specific primers tailed with a universal sequence (Supplementary Data 2) and Phusion Hot Start II High-Fidelity PCR Master Mix (Thermofisher, F-565S) was performed according to the manufacturer's instructions. The amplicons were then PCR barcoded by PacBio Barcoded Universal Primers using Phusion Hot Start II High-Fidelity PCR Master

Mix. After gel purification, the barcoded amplicons were pooled with equal concentrations which were further used for SMRTbell library construction with the SMRTbell Express Template Prep Kit 2.0. PacBio sequencing with Sequel II System was then performed at the Roy J. Carver Biotechnology Center at University of Illinois at Urbana-Champaign. About two million reads with a mean read length of 949 bp were obtained from the sequencing. The reads were error corrected with circular consensus and demultiplexed. Further bioinformatic analysis revealed read numbers of individual mutants from the corresponding plates with various concentrations of ampicillin. The fitness of each mutant at a certain ampicillin concentration was determined based on the ratio of the relative abundance of the mutant to wild-type TEM-1 in the plate with the related concentration of ampicillin and the relative abundance of the variant to wild-type TEM-1 in the plate without ampicillin.

Fitness calculation. The PacBio read data was processed using the TADA workflow (`https://github.com/h3abionet/TADA`). The fitness of a TEM-1 variant is determined by the ratio between the relative abundance of variants under the selection of a specific concentration and without selection. More precisely, for the fitness value $f_c(\mathrm{MT})$ of a variant MT at concentration $c(c = 300, 1500,$ or $3000\ \mu\mathrm{g/mL})$ is calculated as

$$f_c(\mathrm{MT}) = \frac{N_c(\mathrm{MT})/N_c(\mathrm{WT})}{N_0(\mathrm{MT})/N_0(\mathrm{WT})}, \tag{2.6}$$

where WT is the wild type and $N_c(\cdot)$ is the read count of a mutant under concentration c.

2.3.11 Data availability

The following datasets generated or curated in previous publications were used: Envision dataset (`https://doi.org/10.1016/j.cels.2017.11.003`); DeepSequence dataset (`https://doi.org/10.1038/s41592-018-0138-4`); single and double mutants fitness (`https://doi.org/10.1038/s41588-019-0432-9`, `https://doi.org/10.1016/j.jmb.2019.03.020`, `https://doi.org/10.1038/s41467-019-12101-z`); TEM-1 single-mutation and double-mutation mutants fitness data (`https://doi.org/10.1093/molbev/msu081`, `https://doi.org/10.1016/j.jmb.2019.03.020`); high-order avGFP fitness (`https://doi.org/10.1038/nature17995`); inhibitor-resistant TEM-1 variants (downloaded from `https://externalwebapps.lahey.org/studies/TEMTable.aspx` and deposited at `https://doi.org/10.6084/m9.figshare.16516608.v1`); homologous sequences and fitness data of viral proteins (`https://doi.org/10.1126/science.abd7331`); PDB structure of TEM-1 (PDB ID: 1XPB). The fitness data of TEM-1 validation experiment is available in Supplementary Data 2 of Luo et al. [34].

2.3.12 Code availability

The source code of ECNet is available at `https://github.com/luoyunan/ECNet` and on Zenodo at `https://doi.org/10.5281/zenodo.5294461`. ECNet was built on Python 3.7, PyTorch 1.4.0, Numpy 1.18.5, Scipy 1.4.1, Numba 0.45.1, Bio Python 1.78, SciKit-Learn 0.24.1, Pandas 1.2.3, msgpack-python 0.5.6, and TAPE 0.4.

2.4 RESULTS

2.4.1 Accurate prediction of functional fitness landscape of proteins

To validate the ECNet, we performed multiple benchmarking experiments to assess the ability of ECNet in predicting the functional fitness from protein sequences.

We first compared our evolutionary context representation to different representation schemes for protein sequences or mutations. Yang et al. [104] proposed to use a Doc2Vec model [105], pre-trained on ∼500k UniProt sequences, to map an arbitrary sequence to a 64-dimensional real-valued vector. To directly test the utility of sequence representations, we used our deep learning model as the predictor for both our representation and the Doc2Vec representation of Yang et al. We compared the two approaches on the Envision dataset [81], composed of 12 DMS studies that generated fitness values of single amino acid variants of ten proteins ("Methods"). We found that ECNet consistently outperformed the approach of Yang et al on all the 12 datasets, with a relative improvement ranging from 16 to 60% in terms of the achieved Spearman correlation (Figure 2.2a). Since the Doc2Vec representation was learned from the UniProt dataset, the information it captured is mostly general protein properties but not the dependencies in the sequence that determine functions. In contrast, our evolutionary context representation explicitly models the epistasis of residue pairs in the sequence, which jointly influence the function in a non-independent way. This fine-grained information informed the prediction model to learn the sequence-function mapping more effectively and thus improved the prediction performance. We also compared our evolutionary context representation to the Envision model [81], which described a single amino acid substitution using 27 biological, structural, and physicochemical features. Compared to this approach, ECNet, without using these features, still improved the Spearman correlation for most of the proteins (Supplementary Figure B.2; Supplementary Table A.5). As protein engineering focuses on identifying variants with improved properties than the wild type, we further evaluated the model performance using a classification metric (AUROC score), in which variants with higher function measurements than the wild-type sequence are defined

as positive samples, and the remaining variants as negative samples. We observed similar improvements in AUROC scores for 11/12 protein DMS datasets (Figure 2.2b; Supplementary Table A.5). These results suggest that sequence contexts are more informative than the descriptors of mutated amino acids, which is critical in capturing the interdependencies between residues to predict the functions.

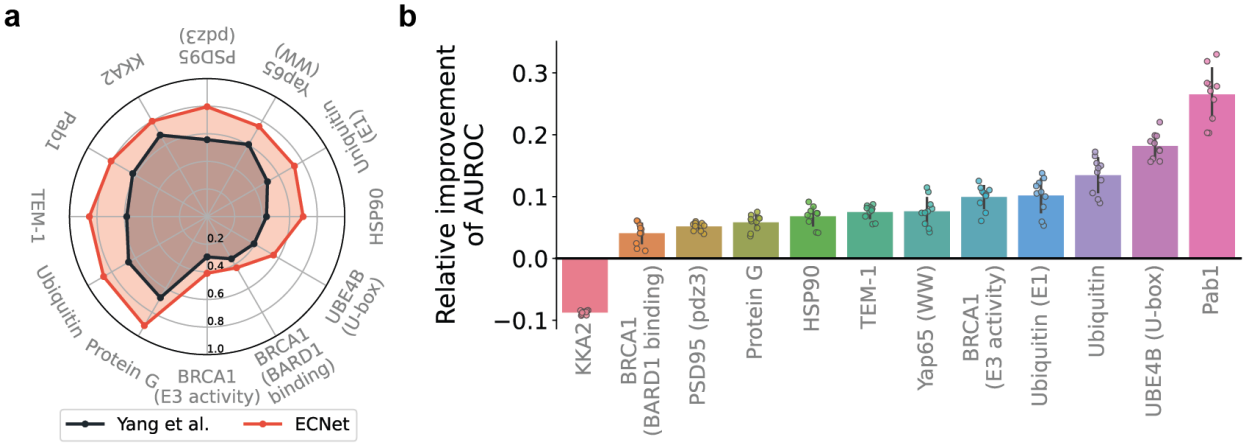


Figure 2.2: Comparisons to other protein variant representation methods. (a) Comparison to the approach from Yang et al. [104] that represents protein sequences with fixed-length vector representations by training a Doc2Vec model on the UniProt database. Spearman correlation was used as the evaluation metric. Performances were evaluated using five-fold cross-validation. (b) Comparison to the Envision model [81] that represents a variant with 27 biological, structural, and physicochemical descriptors. AUROC (area under the receiver operating characteristics) was used as the evaluation metric to assess the ability of the model in identifying variants with improved function compared to the wild type. Relative improvements achieved by ECNet over the Envision model were shown in the bar plot. Performances were evaluated using ten trials of five-fold cross-validation. The bar plot represented the mean±SD of the data.

Next, we compared ECNet to other sequence modeling approaches for mutational effects prediction on a larger set of DMS datasets previously curated [65]. We first compared it to three unsupervised methods, including EVmutation [78], DeepSequence [65], and Autoregressive [106]. These methods trained generative models on homologous sequences and predicted the mutation effects by calculating the log-ratio of sequence probabilities of mutant and wild-type sequences. As expected, ECNet, predicting the mutation effects using a supervised predictor, outperformed these methods across almost all proteins (Figure 2.3a), compared to EVmutation (median difference in Spearman correlation $\Delta\rho$=0.216), DeepSequence (median $\Delta\rho$=0.196), Autoregressive (median $\Delta\rho$=0.165). There were only two proteins on which ECNet did not clearly outperform other unsupervised methods. This is likely due to the relatively small number of function measurements available that we can use to train the supervised predictor (1777 and 985 measurements, respectively; median:

2721 across all proteins). We expect that a more regularized prediction model will achieve improved prediction performance for proteins with a small set of function measurements. We also compared ECNet to two supervised methods. One is TAPE that uses the sequence representations learned by the protein language model [21] as input to train a neural network that has the identical model architecture as ECNet. The other is UniRep [66], which uses the output of its own language model to train a top model based on ridge regression. We found that ECNet, by combining global LM representations, local evolutionary representations, and the raw sequence as input, achieved higher correlations than TAPE (median $\Delta\rho$=0.089) and UniRep (median $\Delta\rho$=0.109) that used LM representations alone for nearly all proteins (Figure 2.3b). We also performed an ablation analysis to dissect the performance of each representation component in our model's input and found that a model using joint representations outperformed a model using any individual representation (Supplementary Figure B.3). Furthermore, we simulated experiments where ECNet was trained on noisy training data and tested on noise-free data. We found that ECNet was robust against data noise (Supplementary Figure B.4). For example, ECNet's test correlation only decreased by 2% when the training data was perturbed by 10%. In contrast, a simple sequence representation such as one-hot encoding was impacted severely by data noise. Overall, tested on a large set of DMS data, ECNet significantly outperformed other sequence modeling methods, either unsupervised or supervised (Figure 2.3c; one-sided rank-sum test $P < 10^{-5}$), demonstrating its superior ability in predicting the fitness landscape of protein variants.

In addition to evaluating ECNet's performance of predicting fitness across all variants as shown above, we further designed an experiment to assess ECNet's ability to prioritize high-performing variants. To this end, we trained an ECNet model and applied it to predict and rank all variants in the randomly split test set based on their predicted fitness. We then calculated the fraction of the true top 100 variants that were ranked in the top K predictions of ECNet. This experiment simulated the process in directed evolution where we want to identify and synthesize the most promising variants for screening, given a sequencing budget of K variants [107]. On three DMS datasets of avGFP, GB1, and Pab1, we found that ECNet achieved higher recall (Supplementary Figure B.5a) and more efficiently discovered the variant with the highest fitness (Supplementary Figure B.5b) than UniRep and EVmutation. ECNet also achieved a 15–50$\times$ efficiency gain over a random sampling approach (Supplementary Figure B.5c), which is a widely used strategy in current directed evolution workflows. These results suggest that ECNet is an effective method to retrieve high-ranking variants for protein engineering and can potentially improve the efficiency of directed evolution in the laboratory.

As a supervised model, the performance of ECNet can be limited when the available DMS

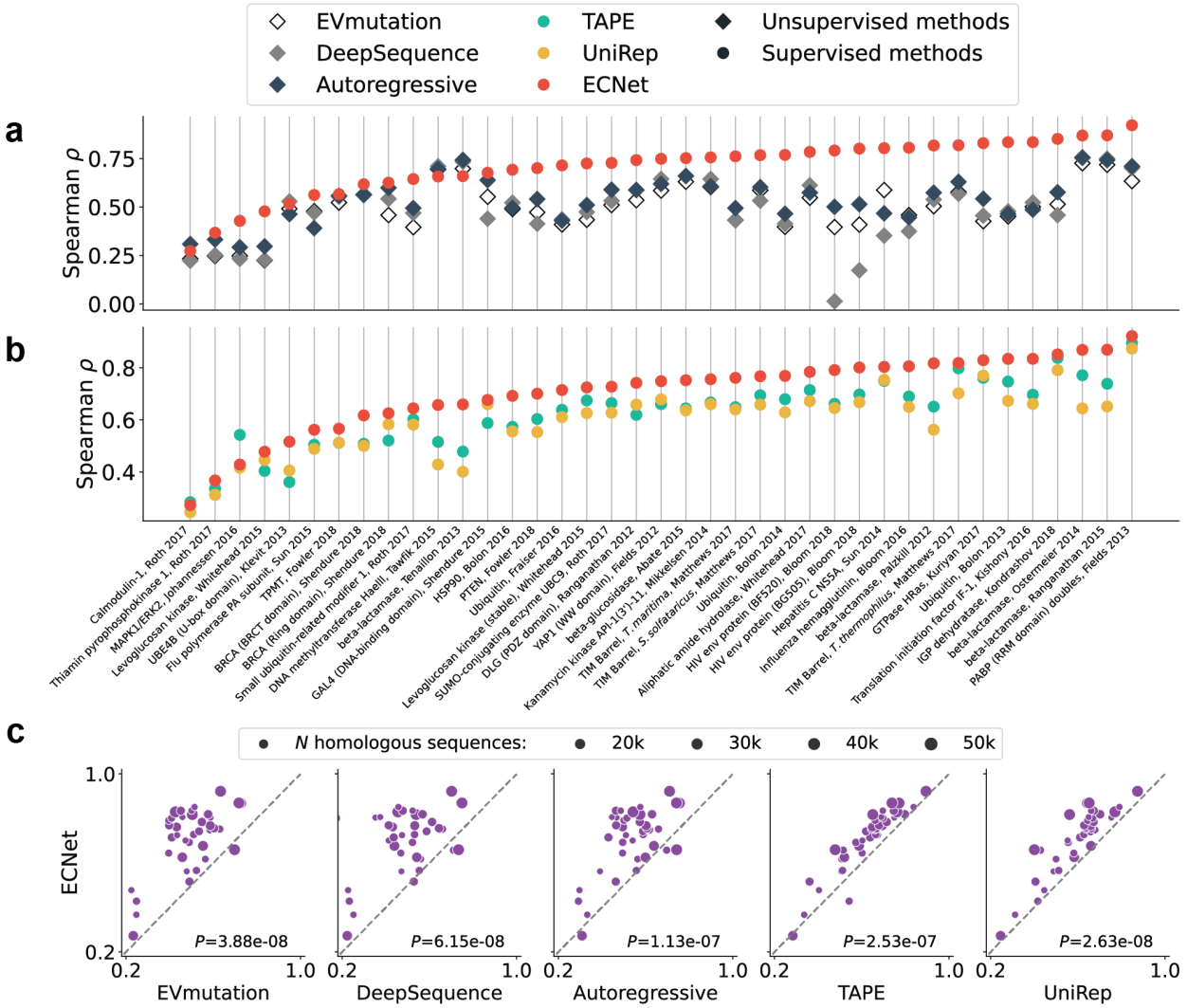


Figure 2.3: Comparisons to other sequence modeling approaches for mutation effects prediction. (a) Comparisons to three unsupervised generative models, EVmutation, DeepSequence, and Autoregressive. (b) Comparisons to two supervised models (UniRep and TAPE) that use a pre-trained protein language model to learn protein sequence representations and fine-tunes a supervised predictor using functional measurements. (c) Pairwise comparisons between ECNet and other methods. Each data point represents the performance on the DMS data of a protein and the dot size is proportional to the number of homologous sequences of the protein. Spearman correlation was used as the evaluation metric for all results in this figure. One-sided rank-sum test was used to test the statistical significance.

data is too scarce to train an accurate predictor. To address this challenge, we further built an unsupervised version of ECNet model that does not require any DMS data for training but is able to produce reasonably accurate predictions. Inspired by protein language models, we built unsupervised ECNet by training it on homologous sequences of the protein of interest using the language model objective. The predicted probability of an amino acid at a position

was used as the proxy of fitness prediction ("Methods"). Tested on four DMS studies covering ten viral protein strains, unsupervised ECNet achieved an average Spearman correlation of 0.37 (Supplementary Figure B.6). This unsupervised variant of ECNet is particularly useful when the target protein is novel and has very few available DMS data. For example, we observed that unsupervised ECNet achieved reasonably good performance (mean Spearman correlation 0.36; Supplementary Figure B.7) on the DeepSequence dataset without using any DMS data as the supervised signal. In addition, using a small number of DMS data (e.g., 25% of available data of each protein) to train a supervised ECNet model substantially improved the prediction performance (mean Spearman correlation 0.54; Supplementary Figure B.7), and using the full DMS data further boosted the results (mean Spearman correlation 0.71; Supplementary Figure B.7). We thus expect that unsupervised ECNet can select promising variants for screening in the first round of directed evolution, after which the screening data can be used to train a supervised ECNet for later rounds to improve the model accuracy and prioritize improved variants.

2.4.2 Generalization to higher-order variants from low-order variants data

Construction and screening of higher-order variants can require a significant amount of experimental effort and time. As a result, fitness measurements of single mutants were more prevalent in existing DMS studies as compared to those of double or higher-order mutants. It is thus highly desired in protein engineering that a machine learning model trained on fitness data of low-order variants can also accurately predict the fitness of higher-order variants. As such, the model can fully leverage the fitness data of screened low-order variants and prioritize higher-order variants that are likely to exhibit improved properties for the next round of directed evolution.

We thus assessed ECNet's performance on predicting the fitness of higher-order variants when only lower-order data were used for model training. We collected the fitness measurements of both single and double mutants of six proteins from previous DMS studies [79, 89, 90, 108, 109, 111]. We then trained our prediction model using single mutant data only and tested its performance on double mutants. ECNet achieved Spearman correlations ranging from 0.73 to 0.94 for the six proteins and outperformed the TAPE and the EVmutation methods (Figure 2.4a), suggesting its generalizability to the prediction of higher-order variants from low-order variants data. We also observed that the increased diversity of fitness landscape in the training data improved the prediction performance. For example, to predict the fitness of quadruple mutants of the avGFP protein53, we trained separate models using the fitness data of single, double, triple, or all three orders of mutants.

The test results suggested that a model trained on higher-order mutation data (from single to triple) achieved an increasing prediction performance, and the union of all-order mutation data further improved the prediction (Figure 2.4b). To further assess ECNet's ability, we used orthogonal data containing sequences of 146 TEM-1 variants that are known to be inhibitor-resistant ("Methods"). Sequences in this data contain two to ten (mean 3.3) amino acid substitutions compared to the TEM-1 protein. Based on these sequences, we generated ten times more random variants by enumerating all mutation combinations restricted to the positions where mutations were introduced in the 146 variants ("Methods"). We then trained our model on fitness data of TEM-1 single mutants data and used it to predict the fitness of the 146 TEM-1 variants as well as the randomly generated variants. We found that ECNet distinguished the inhibitor-resistant variants from the random variant background (Figure 2.4c; mean predicted fitness 0.79 vs. 0.48; one-sided rank-sum test $P < 10^{-5}$). This orthogonal validation further demonstrates the generalizability of ECNet, even trained on single mutants data, to the prediction for higher-order mutants.

It was shown that mutations within the sequence can have non-independent effects (epistasis) on fitness [86, 112]. The double mutant fitness f_{ij} may not always be equal to the sum of constituent single mutant fitness $f_i + f_j$, where f's are the (log-transformed) experimentally measured fitness of variants. Epistasis (ϵ) is quantified as the difference between the experimentally measured fitness and the expected fitness: $\epsilon = f_{ij} - (f_i + f_j)$. To analyze whether ECNet captures the interdependencies between mutations, we correlated the observed epistasis ϵ with predicted epistasis $\hat{\epsilon}$, which is defined as $\hat{\epsilon} = \hat{f}_{ij} - (\hat{f}_i + \hat{f}_j)$ where $\hat{f}$'s are predicted fitness. Compared with EVmutation that explicitly models epistasis using a generative model, the epistasis predicted by ECNet better correlated with the observed epistasis (Figure 2.4d; one-sided rank-sum test $P < 10^{-5}$). The epistasis captured by ECNet was also more accurate or comparable to that of TAPE (Figure 2.4d). These results suggest that ECNet captured the residue dependencies within sequences more accurately, and thus resulted in the superior prediction performances reported above.

2.4.3 Engineering of TEM-1 beta-lactamase using ECNet

To experimentally validate its utility in protein engineering, we applied ECNet to prioritize new higher-order TEM-1 β-lactamase variants that are likely to have improved fitness compared to the wild type. We trained ECNet using DMS data reported in previous studies [90, 110]. The datasets curated the fitness measurements of nearly all point-mutation variants and 12% of possible consecutive double-mutation variants of TEM-1. We performed in silico mutagenesis for several function-related sites of TEM-1curated in the literature and

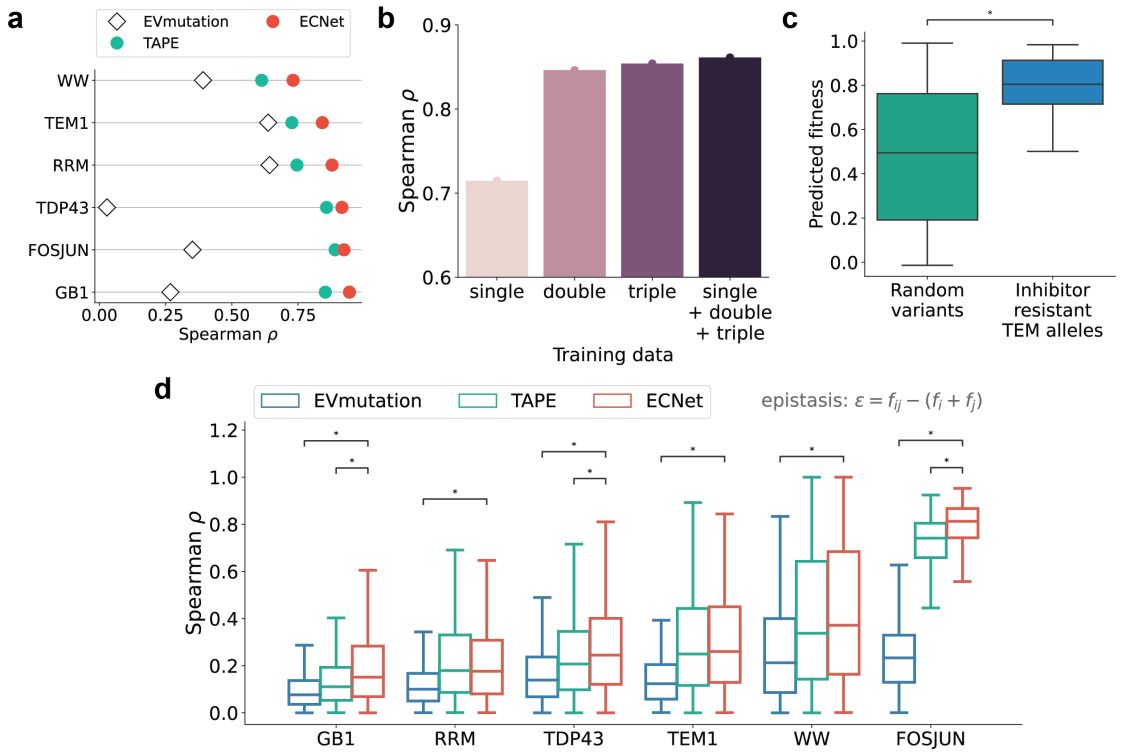


Figure 2.4: Accurate prediction of higher-order variants using a model trained on lower-order variants. (a) Prediction of the fitness of double mutants. For supervised methods (ECNet and TAPE), the prediction models were trained using fitness measurements of single mutants. (b) Prediction of quadruple mutants of avGFP using models trained on single, double, triple, and all three types of mutants. (c) The predicted fitness values of inhibitor-resistant TEM-1 variants ($n = 146$) were significantly higher (one-sided rank-sum test $P = 5.1 \times 10^{-32}$) than those of randomly generated background variants ($n = 1460$). (d) Spearman correlation of experimentally measured epistasis and predicted epistasis for double-mutation variants of GB1 ($n = 4455$), RRM ($n = 2700$), TDP43 ($n = 5166$), TEM-1 ($n = 841$), WW ($n = 1680$), and FOSJUN ($n = 3072$). In comparison to EVmutation, the Spearman correlations achieved by ECNet were significantly higher for all six proteins (one-sided rank-sum test P values: GB1: 6.7×10^{-72}, RRM: 4.7×10^{-35}, TDP43: 4.0×10^{-7}, TEM-1: 1.8×10^{-18}, WW: 6.7×10^{-17}, FOSJUN: 1.0×10^{-80}). In comparison to TAPE, ECNet was comparable for proteins RRM, TEM-1, and WW and achieved significantly higher correlations for proteins GB1, TDP43, and FOSJUN (one-sided rank-sum test P values 2.4×10^{-19}, 4.2×10^{-9}, and 2.2×10^{-51}, respectively). In box plots, the midline represents the median, the lower and upper hinges of the boxes correspond to the 25th and 75th percentiles, and the whiskers extend to 1.5 times the interquartile range from the hinges. The asterisk symbol $\star$ indicates P values $< 10^{-5}$.

their higher-order recombinations ("Methods"). We then applied ECNet to predict the fitness for all variants generated from the in silico mutagenesis. After removing structurally unstable variants, we selected 37 variants that were ranked at the top by either the standard ECNet model or an ensemble version of ECNet, which averages predictions of multiple replicates of ECNet models ("Methods"). The 37 top-performers were novel TEM-1 variants

and did not overlap with any variants in our training data or functional TEM-1 variants we collected from the literature. Despite that the training data only covered single and consecutive double mutants, these 37 variants sampled a diverse combination of mutation sites and contained higher-order mutants ranging from 2 to 6 mutations (Supplementary Data 2 of Luo et al. [34]).

We created those 37 variants and nine previously reported TEM-1 mutants which had demonstrated strong resistance against ampicillin to serve as positive controls [90, 110] (Supplementary Data 1 of Luo et al. [34]). We plated the library containing these 37 variants and positive controls on LB agar plates with ampicillin of various concentrations (300, 1500, and 3000 μg/mL) to test their resistance capacity against ampicillin. Further, PacBio sequencing was performed to determine the relative abundance of these variants before and after selection, as a proxy of their fitness (Figure 2.5a; "Methods"). The fitness of each mutant at a certain ampicillin concentration was calculated based on the ratio of the relative abundance of the mutant to wild-type TEM-1 in the plate with the related concentration of ampicillin and the relative abundance of the mutant to wild-type TEM-1 in the plate without ampicillin ("Methods" and Supplementary Data 2 of Luo et al. [34]). We observed that most of the variants prioritized by ECNet demonstrated improved fitness as compared to the wild type (Figure 2.5b). The improvements were observed at various concentrations of ampicillin (300, 1500, and 3000 μg/mL) and were reproducible across different replicates. Notably, ECNet has identified variants that improved the wild-type fitness by up to $\sim$8-fold, which was substantially higher than the best performers we had in the training data (positive controls in Figure 2.5b). We also found that the ensemble model of ECNet achieved robust predictions, with a mean hit rate (fraction of predicted variants with fitness higher than the wild type) 0.52, 0.91, and 0.94 for concentrations 300, 1500, and 3000 μg/mL, respectively (Figure 2.5c).

Despite being trained on the data of single mutants and consecutive double mutants, ECNet prioritized novel and higher-order TEM-1 mutants that showed improved resistance against ampicillin. The validation results suggested that the evolutionary contexts enable ECNet to discover higher-order mutants that have not been observed in the training data. The results also demonstrate the potential of ECNet to be integrated into the existing protein engineering workflows to guide the discovery of enhanced variants.

2.5 DISCUSSION

A critical challenge in machine learning-guided protein engineering is the development of a machine learning model that accurately maps protein sequences to functions for un-

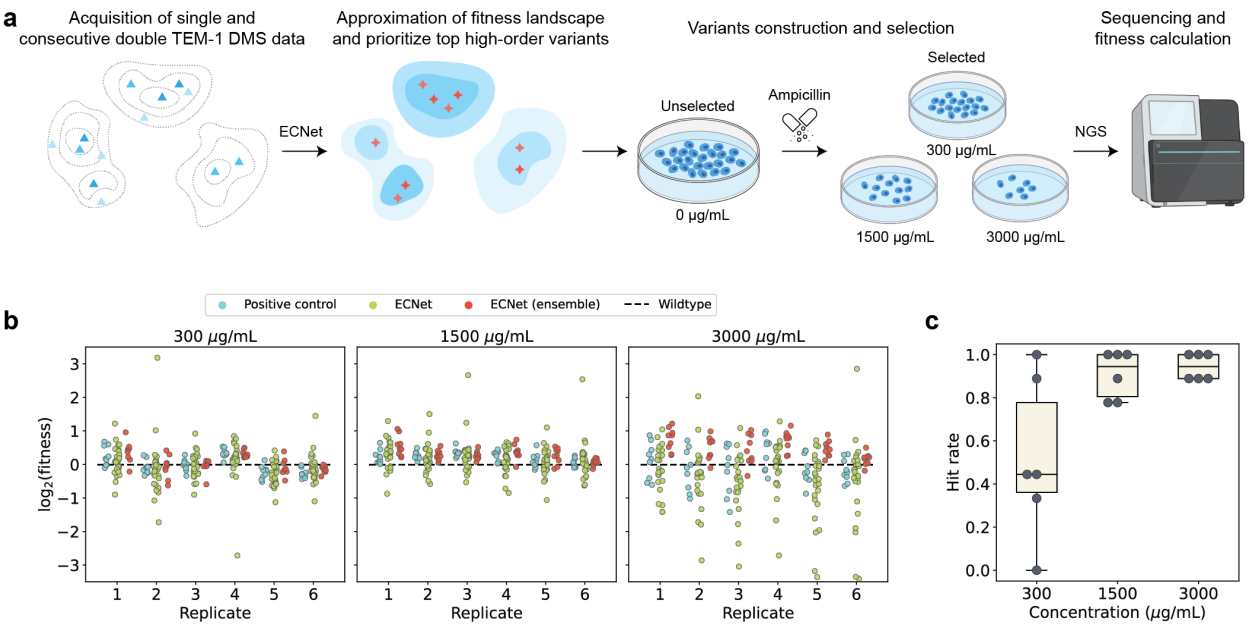


Figure 2.5: ECNet enables the rapid engineering of TEM-1. (a) The workflow of using ECNet to predict enhanced TEM-1 variants. The ECNet model was trained on fitness data of single and consecutive double TEM-1 mutants and applied to prioritized higher-order mutants; top-ranked TEM-1 variants were constructed and their fitness (resistance against ampicillin) was measured (DMS: deep mutational scanning; NGS: next-generation sequencing). (b) Fitness values of predicted TEM-1 variants at different ampicillin concentrations. Results from six replicates are shown. Fitness values of variants prioritized by an ensemble version of ECNet (averaged predictions of multiple replicates of ECNet models) are colored separately. Top-performing single or consecutive double mutants in the training data are labeled as positive controls. The Black dashed line represents the fitness of wild-type TEM-1. (c) Hit rate (fraction of predicted variants with fitness higher than the wild type) of the ensemble ECNet model. Each point represents a replica ($n = 6$ replicates in total). The midline of box plots represents the median, the lower and upper hinges of the boxes correspond to the 25th and 75th percentiles, and the whiskers extend to 1.5 times the interquartile range from the hinges.

seen variants. While models have been developed for the qualitative classification of protein sequences into function classes, such as those in the Critical Assessment of Functional Annotation (CAFA) challenge [60], in protein engineering prediction models are required to provide a more fine-grained characterization of protein functions, which distinguishes the quantitative function levels of closely related sequences (e.g., single-site mutants of wild-type protein with sequence similarity $> 99\%$). The function prediction in protein engineering is also different from predicting the deleteriousness [113] or instability [84] of variants—to assist protein engineering, the machine learning model needs to prioritize variants that are not only structurally stable and non-deleterious but also with enhanced properties. Furthermore, as the protein sequence space is tremendous in size, it is desired to have a machine learning

model that navigates the fitness landscape effectively and can generalize from regions of low-order variants to regions in the landscape where higher-order variants with improved function may exist. All these factors render it uniquely challenging to develop a machine learning model that can be used to guide protein engineering strategies such as directed evolution and rational design.

In this work, we have presented a high-performance method, ECNet, that predicts protein function levels from sequence to facilitate the process of protein engineering. Supervised machine learning models have been explored recently to predict protein sequence-function relationships [107, 114, 115, 116]. As in those studies, in this work, we mainly focused on improving the protein function by introducing point mutations, while introducing insertion/deletion was also explored in other work [106]. Our machine learning model uniquely used a biologically-motivated sequence modeling approach to learn the sequence-function relationship, leading to superior performances in predicting the fitness of protein variants. Benchmarked on a large set of deep mutational scanning studies, ECNet outperformed multiple existing machine learning models for protein engineering. Further, ECNet accurately captures the epistasis effects of mutations within protein sequences and can be generalized to predict higher-order mutants' functions by learning from the data of lower orders. We applied ECNet to engineer TEM-1 β-lactamase and experimentally validated that it successfully identified variants with enhanced ampicillin resistance with high hit rates.

ECNet's prediction performance is impacted by the MSA characteristics and DMS data properties. For example, ECNet predicts better for proteins with more homologous sequences (Supplementary Figs. 8 and 9a), for sites that are more conserved within a protein (Supplementary Fig. 9b), and for proteins that have a more complete DMS dataset (Supplementary Fig. 9c-d). In our additional tests that used both a sequence site-wise strategy and a per-site AA-wise train/test split strategy61 to assess ECNet's generalizability, we found that, despite the challenging setting, ECNet still outperformed the DeepSequence when predicting for new mutation sites (Supplementary Fig. 10). Further investigation revealed that the exploration-exploitation trade-off of training data also influenced the model performance (Supplementary Fig. 11). This implies that the design of more effective training data should be taken into account when developing ML algorithms to assist protein engineering, especially when the experimental test budget is limited62.

We expect ECNet to be a practical tool for ML-guided protein engineering. In a round of directed evolution, the sequence-to-function model can be applied, potentially coupled with other sequence design algorithms63,64,65, to select the next set of variants to screen. In addition, given its generalizability to higher-order mutants from lower-order mutants, the model can fully leverage the screening data of low-order mutants, including that of both

improved and unimproved variants, generalize to distant regions in the fitness landscape where higher-order variants with improved properties may exist, and prioritize promising higher-order mutants to screen in the next round, in which the screened data can be used to further improve the model, hereby forming an iterative loop of directed evolution to discover improved variants.

Chapter 3: Representation Learning of Protein Structures

The amino acid chain of a protein can fold into a structure in the 3D space, which is known as protein folding and driven by the physicochemical properties of the arranged amino acids. The 3D shape is important for a protein to fulfill a certain function, such as binding to a substrate. In this chapter, we present a method for representation learning of protein structures and show that integrating the 3D structure of proteins improves prediction of protein binding, for which 3D geometric information better characterizes the activity than sequence information.

3.1 INTRODUCTION

Proteins serve as drug targets for therapeutic purposes. It was estimated that only 11% of human proteome can be targeted by drugs or small molecules, leaving a large proportion to be explored for therapeutic opportunities [12]. A group of proteins called kinase is of particular interest as drug targets because they are tractable in drug development and have diverse pharmacological implications in a wide range of diseases, such as cancers and infectious diseases [117, 118]. Protein kinases present high evolutionary conservation in sequence and structure. Most of the kinase inhibitors, however, bind to conserved ATP-binding pockets of kinases, thus resulting in extensive target promiscuity [119]. Chemical compounds that inhibit a single kinase are still rare despite significant research efforts devoted to target-based drug discovery in recent years [120]. The multi-target activities contribute to therapeutic responses as well as adverse off-target or toxic consequences. Mapping out the target binding profiles is therefore critical to uncover new therapeutic effects and to better predict and manage possible adverse effects. Unfortunately, even with automated high-throughput profiling assays, it is still infeasible to exhaustively measure the compound-target binding activities because of the enormously large chemical space.

Computational approaches, especially machine learning (ML) methods, have emerged as alternative solutions to accelerate the mapping of compound-protein interaction profiles [42]. Early studies include several bipartite graph-based methods that formulated the prediction problem as a recommendation system-like task [36, 121, 122, 123, 124]. These methods computed the similarity between compounds or proteins based on simple features, e.g., molecule fingerprints or sequences alignment scores, so as to predict the interaction of new protein-drugs based on known, similar proteins and drugs. With the rapid advances of deep learning in recent years, a line of studies [125, 126, 127, 128, 129] leveraged deep neural

networks to automatically learn features from raw representations of compounds and proteins in a fully data-driven way, as known as end-to-end learning. The data representations widely used by deep learning methods are 1D features such as protein sequences and molecule SMILES strings [125, 126]. Recent approaches suggested that incorporating 2D features, including molecular graphs and protein contact maps, could clearly boost the prediction accuracy [127, 128, 129, 130]. While the compound-protein binding, in essence, is a physical process in the three-dimensional space, there are very few existing studies incorporating 3D structure information to further improve the protein-drug binding prediction, in part due to the limited availability of protein structure data, compared to sequence data, and the lack of predictive models that can effectively utilize 3D data for binding prediction. Fortunately, the bottleneck of data availability becomes less severe for kinase proteins due to their biological importance. In fact, kinases are one of the most representative protein families in the Protein Data Bank (PDB) database [131] and the number of solved kinase structures keeps increasing at a fast rate [132, 133]. In parallel, the recent progress in graph deep learning offers new solutions to effectively model 3D data such as the PDB structures of proteins. Jointly, there are great opportunities and pressing needs to develop new methods that integrate 3D structure information to improve predictions of kinase-drug binding affinity.

The primary importance of ML approaches for compound-protein binding prediction is to accelerate the discovery of compounds or targets. With an accurate ML predictive model, researchers can perform virtual screening by applying the model to generate hypotheses about binding activities and select candidates with the best-predicted activities for downstream validation. However, as data-driven approaches, those methods are largely impacted by the intrinsic noise and bias in the data and susceptible to pathological failures in out-of-distribution regimes. To address this issue, a solution would be to quantify the uncertainty of model predictions as a confidence assessment in support of human decisions, as higher novelty often comes with a higher risk of failure. Uncertainty estimation started to be recognized as a critical property of ML algorithms [134, 135, 136], yet the majority of existing methods of compound-protein binding prediction only provide the point-estimate prediction of affinity without quantifying the uncertainty [125, 126, 127, 128, 129]. In the context of compound or target discovery, simply choosing the top candidates for downstream validation only based on point-estimate predictions may lead to false positives. Hie et al. [130] took the initiative to introduce uncertainty estimation for prioritizing strong-binding compound-protein pairs using the Gaussian process, but quantifying uncertainty with more expressive deep neural networks has not been explored for kinase-drug binding prediction.

Here, we develop KDBNet, a deep learning algorithm that integrates 3D structure information for predicting the binding affinity of kinase-drug binding as well as estimating the predic-

tion uncertainty. KDBNet represents the 3D protein and molecule structure data as graphs and uses graph neural networks to learn structure representations from binding pocket structures of proteins and atom coordinates of molecules. Compared to other algorithms that only consider 1D or 2D representations of proteins and compounds, KDBNet learns more explicit features from 3D input data directly, which better respects the nature of binding activities happening in the 3D space. In contrast to recent methods that relied on solved protein-compound binding complex to predict the binding affinity [137, 138, 139, 140, 141, 142], KDBNet is more flexible in that it only requires separate structures rather than a binding complex as input, thus it can be applied to large molecule libraries and a wide range of proteins for which their binding complex have not been solved. We built KDBNet as an ensemble model of multiple replicates of individual neural networks, which not only improves the prediction accuracy and robustness but also allows us to estimate the uncertainties of model predictions. We further applied an uncertainty recalibration technique to refine the uncertainty estimates, improving KDBNet's utility in applications of ML-guided discovery of proteins and targets. Benchmarking on public datasets of kinase-drug binding affinity measurements, we observed that KDBNet achieved substantially more accurate predictions than existing models that only used 1D or 2D representations of proteins and drugs. Experiments also suggested that KDBNet's uncertainty estimates were largely consistent with respect to prediction errors, meaning that predictions with lower uncertainty are often more accurate. Furthermore, we found the uncertainty estimates were also well-calibrated, providing statistically meaningful confidence intervals of individual predictions. Finally, we extended KDBNet into a Bayesian optimization framework and demonstrated that it enables data-efficient active learning and accelerated exploration and exploitation of strong-binding kinase-drug pairs.

3.2 KDBNET: CALIBRATED DEEP LEARNING FOR KINASE-DRUG BINDING AFFINITY PREDICTION

In this work, we develop KDBNet (kinase-drug binding neural network), a deep learning model that integrates 3D structures to predict binding affinities between kinases and small-molecule compounds. We focus on this particular group of proteins (kinase) due to its pharmacological implications for therapeutics of many diseases such as cancer [12, 143] and the availability of comprehensive compound-kinase binding datasets [144, 145]. As an overview (Figure 3.1), KDBNet receives three-dimensional structures of proteins and compounds and represents them as two graphs, where the graph's nodes are protein residues or molecule atoms, and the edges encode residue contacts or atom distance. A set of fea-

tures, reflecting the spatial or chemical properties of the input protein and compound, are also derived for each node and edge in the protein and molecule graphs. Next, KDBNet uses two graph neural networks to learn structure representations of the input kinase and compound, which are then combined to predict the binding affinity through another fully connected neural network. We further equip KDBNet's prediction with uncertainty estimation by training an ensemble of models and estimating the uncertainty using the variance of individual models' predictions.

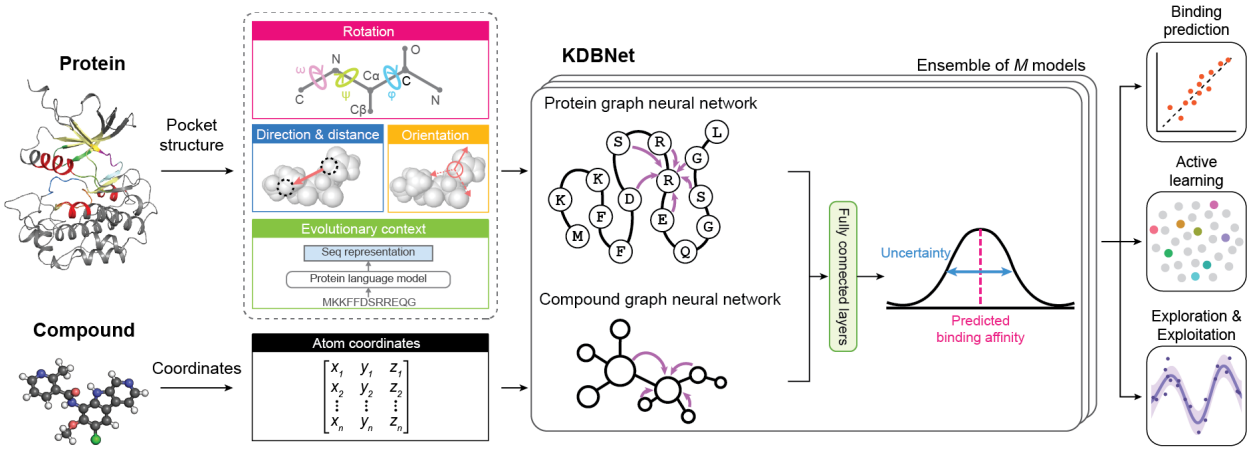


Figure 3.1: Overview of KDBNet. KDBNet is a neural network that integrates protein 3D structure and compound 3D structure for predicting compound-protein binding affinity. KDBNet derives a set of features, including sequence, evolutionary representations, and 3D-invariant geometric features, based on the input 3D structure and uses a graph neural network to learn structure-aware representations of a protein. For the input compound, KDBNet uses a 3D-equivariant graph neural network to directly learn structure representations from the compound's coordinates in the 3D space. The representations of the input protein and are then used to predict the binding affinity as well as the uncertainty of the prediction.

3.3 METHODS

3.3.1 Representation of protein structure

The human genome encodes more than 500 protein kinases that share a similar fold (3D structure) with a small N-terminal lobe and a large C-terminal lobe, consisting of several conserved α-helices and β-strands. Connecting the two lobes is a key activation loop formed by 20-30 residues, typically starting with the DFG motif (Asp-Phe-Gly) and ending with the APE motif (Ala-Pro-Glu). This activation segment primarily coordinates the substrate-binding (e.g., to small-molecule compounds) and enzymatic activities of protein kinases.

Given the crucial role of protein structure in binding activity, we hypothesize that it would be more informative to incorporate 3D protein structure data to predict kinase-drug binding affinity, compared to predicting only from protein sequences [125, 129].

The 3D PDB structure of a protein is given as 3D coordinates of the backbone $\mathcal{C} = \{\boldsymbol{c}_i \in \mathbb{R}^3\}_{i=1}^N$, where N is the number of residues and $\boldsymbol{c}_i$ is the coordinate of the C_α atom of the i-th residue. We represent the protein structure as a graph $\mathcal{G}_p = (\mathcal{V}_p, \mathcal{E}_p)$ where nodes $\mathcal{V}_p$ are residues and edges $\mathcal{E}_p$ indicate residue contacts. In this work, we defined the residue contact using a distance cutoff of 8Å between C_α atoms [26].

To make the structure graph representation more informative, we associate every node or edge in the graph with a feature vector. Intuitively, we want our node and edge features to be i) *invariant* to rotation and translation so that the features will not depend on coordinate systems defined in different PDB structure inputs, and ii) *informative* about the local structure, as unique structural motifs may lead to distinct binding affinities. Here, we derive a set of invariant spatial features following a previous study [146]. We further extended their approach to include other features that encode sequence and evolutionary properties of residues. The constructions of node and edge features are detailed below.

Node features: For every residue, we build three types of features: i) sequence feature, ii) geometric feature, and iii) evolutionary feature. The sequence feature is a one-hot representation to indicate the amino acid (AA) type (out of the total 20 possible AAs) of the residue. For geometric features, we computed the three dihedral angles $(\phi_i, \psi_i, \omega_i)$ based on the backbone coordinates of residue i. These angles were encoded as a vector of cosine and sine values: $\boldsymbol{v}_i = (\sin\phi_i, \sin\psi_i, \sin\omega_i, \cos\phi_i, \cos\psi_i, \cos\omega_i)$. Lastly, for evolutionary features, we ran ESM [67], a recent protein language model trained on 250 million sequences, to generate the embedding for each residue. The ESM embeddings have been shown to encode structural, functional, and evolutionary properties of the protein and can improve a wide range of protein-related prediction tasks, such as function and structure prediction [34, 67]. Those three features are concatenated together as the node feature for a residue.

Edge features: To characterize the local structure surrounding residue i, we create edge features that describe the spatial relationships between residue i and its neighbors (residues j's). In particular, we compute an orientation matrix $\boldsymbol{O}_i$ that defines the local coordinate frame for residue i:

$$\boldsymbol{O}_i = [\boldsymbol{b}_i, \boldsymbol{n}_i, \boldsymbol{b}_i \times \boldsymbol{n}_i], \quad \text{where} \quad \boldsymbol{u}_i = \frac{\boldsymbol{c}_i - \boldsymbol{c}_{i-1}}{\|\boldsymbol{c}_i - \boldsymbol{c}_{i-1}\|}, \quad \boldsymbol{b}_i = \frac{\boldsymbol{u}_i - \boldsymbol{u}_{i+1}}{\|\boldsymbol{u}_i - \boldsymbol{u}_{i+1}\|}, \quad \boldsymbol{n}_i = \frac{\boldsymbol{u}_i \times \boldsymbol{u}_{i+1}}{\|\boldsymbol{u}_i \times \boldsymbol{u}_{i+1}\|}, \tag{3.1}$$

where $\boldsymbol{c}_i \in \mathbb{R}^3$ is the coordinates of residue i. For an edge edge (i, j) we consider an edge representation that reflects the local distance, direction, orientation, and relative positions [146]:

$$\boldsymbol{e}_{ij} = \left(\text{RBF}(\|\boldsymbol{c}_j - \boldsymbol{c}_i\|), \quad \boldsymbol{O}_i^T \frac{\boldsymbol{c}_j - \boldsymbol{c}_i}{\|\boldsymbol{c}_j - \boldsymbol{c}_i\|}, \quad \boldsymbol{q}(\boldsymbol{O}_i^T \boldsymbol{O}_j), \quad E_{\text{pos}}(\boldsymbol{c}_j - \boldsymbol{c}_i) \right), \tag{3.2}$$

The edge features $\boldsymbol{e}_{ij}$ has four components: (i) The first part $\text{RBF}(\|\boldsymbol{c}_j - \boldsymbol{c}_i\|)$ is the distance encoding embedded into radial basis functions (RBFs). We use 16 RBFs with centers evenly spaced between 0 and 8Å. (ii) The second term is the direction encoding that corresponding to the relative direction of $\boldsymbol{c}_j$ in the local frame of residue i. (iii) The third term is the orientation encoding of the quaternion representation $\boldsymbol{q}(\cdot)$ of the spatial rotation matrix $\boldsymbol{O}_i^T \boldsymbol{O}_j$. (iv) The last term $E_{\text{pos}}(\boldsymbol{c}_j - \boldsymbol{c}_i)$ encodes the relative distance and direction between residues i and j. We used the relative positional encoding [147], an extension of the positional encoding introduced in the Transformer model [99]. The relative positional embedding represents the vector pointing to $\boldsymbol{c}_j$ from $\boldsymbol{c}_i$ through a sinusoidal function. We keep the sign of the distance vector $\boldsymbol{c}_j - \boldsymbol{c}_i$ because protein sequence structures are generally asymmetric.

3.3.2 Representation of molecule structure

KDBNet also incorporates the 3D molecular structure of compounds to predict binding affinities. Similarly, given the 3D coordinates of atoms in the molecule, we represent the molecule structure as a graph $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$ where nodes $\mathcal{V}_d$ are atoms of the molecule and edges $\mathcal{E}_d$ are defined for a pair of atoms if their distance is less than 4.5Å [148]. We found in local tests that, compared to simply defining edges based on the actual chemical bonds of the molecule, using the distance cutoff to define edges yielded more expressive representations and better performance. As molecules do not have a natural backbone as in proteins, we do not derive the angle, orientation, and direction features for atoms as we did in the protein graph. Instead, we directly use the 3D coordinates of atoms or edge vectors as node features and edge features, allowing the graph neural network in KDBNet to learn meaningful geometric representations of the molecule in a data-driven way. The node and edge features of the molecule structure are detailed below.

Node features: For every atom, we include a vector-valued feature and a scalar-valued feature as its node feature. The vector feature is the atom coordinates $\boldsymbol{c}_i \in \mathbb{R}^3$. The scalar feature is a list of 66 descriptors of chemical properties [128, 129, 141], including the atom type, bond degree, number of hydrogen bonds, number of implicit hydrogen bonds, and whether the atom is aromatic (Supplementary Table A.4).

Edge features: For an edge between atoms i and j, we also create a vector feature and a scalar feature. The vector feature is the unit vector in the direction of $\boldsymbol{c}_j - \boldsymbol{c}_i$, and the scalar feature $\text{RBF}(\|\boldsymbol{c}_j - \boldsymbol{c}_i\|)$ is the pairwise distance embedded into 16 Gaussian RBFs

with centers evenly spaced between 0 and 4.5Å.

3.3.3 KDBNet model architecture

Now, we introduce the neural network architecture of KDBNet. The major components of KDBNet are two graph neural networks (GNNs) to learn structure representations from the input protein and compound, respectively. The representations produced by the two GNNs are then passed to a fully connected neural network to predict the binding affinity between the input protein and compound.

Protein graph neural network For the protein GNN, we use Graph Transformer [149], an effective GNN architecture adapted from the vanilla Transformer model for text data [99], to model the kinase structure. Given the protein structure graph $\mathcal{G}_p = \{\mathcal{V}_p, \mathcal{E}_p\}$, a graph transformer model builds L graph convolution layers. The i-th layer is a non-linear transformation function that transforms node i's embedding $\boldsymbol{h}_i^{(\ell-1)} \in \mathbb{R}^{d_{\ell-1}}$ to $\boldsymbol{h}_i^{(\ell)} \in \mathbb{R}^{d_\ell}$ for $i \in [N]$, $\ell \in [L]$, where d_ℓ is the embedding's dimension at layer ℓ, N is the number of nodes in $\mathcal{G}_p$, and L is the total number of layers in the GNN. In particular, when $\ell = 0$ the embedding $\boldsymbol{h}_i^{(0)} \in \mathbb{R}^{d_0}$ is just the node feature of residue i. In addition, we have edge features of each edge (i, j) denoated as $\boldsymbol{e}_{ij} \in \mathbb{R}^{d_e}$.

Formally, in the i-th Graph Transformer layer of the GNN, the hidden representation $\boldsymbol{h}_i^{(\ell)}$ is updated by performing a message passing between node i and its neighbors

$$\boldsymbol{h}_i^{(\ell)} = \boldsymbol{W}_1^{(\ell)} \boldsymbol{h}_i^{(\ell-1)} + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} (\boldsymbol{W}_2^{(\ell)} \boldsymbol{h}_j^{(\ell-1)} + \boldsymbol{W}_3^{(\ell)} \boldsymbol{e}_{ij}), \tag{3.3}$$

where $\mathcal{N}(i)$ is the set of neighbor nodes of i in the graph, $\boldsymbol{W}_1^{(\ell)} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$, $\boldsymbol{W}_2^{(\ell)} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$, and $\boldsymbol{W}_3^{(\ell)} \in \mathbb{R}^{d_e \times d_\ell}$ are learnable parameters of the GNN, and $\alpha_{i,j}$ is the attention weight used to aggregate messages. The weights $\alpha_{i,j}$ are computed using self-attention:

$$\alpha_{i,j} = \text{softmax}\left([(\boldsymbol{W}_4^{(\ell)} \boldsymbol{h}_i^{(\ell-1)})^\top (\boldsymbol{W}_5^{(\ell)} \boldsymbol{h}_j^{(\ell-1)} + \boldsymbol{W}_3^{(\ell)} \boldsymbol{e}_{ij})] / \sqrt{d_\ell} \right), \tag{3.4}$$

where $\boldsymbol{W}_4^{(\ell)} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ and $\boldsymbol{W}_5^{(\ell)} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ are learnable parameters and d_ℓ is the length of vector $\boldsymbol{h}_i^{(\ell)}$.

We stacked three Graph Transformer layers and used the Leaky ReLU activation function [150] between two adjacent layers. After the final layer, we use the global add pooling operation as the readout function to aggregate all node representations into a summary representation $\boldsymbol{h}^p \in \mathbb{R}^{256}$ of the input protein: $\boldsymbol{h}^p = \text{ADD}(\{\boldsymbol{h}_i^{(L)} | i = 1, \ldots, N\})$.

Molecule graph neural network Given the molecule structure graph $\mathcal{G}_d = \{\mathcal{V}_d, \mathcal{E}_d\}$, we also use a GNN to learn the representation for the input molecule. Recall that in graph $\mathcal{G}_d$, we associate each node and edge with both geometric vector features (e.g., 3D coordinates) and scalar features (e.g., descriptors of chemical properties). We thus use a specialized layer, named geometric vector perceptrons (GVPs) [151], to build the molecule GNN. The key advantage of GVP is that it has special consideration for 3D data in design and allows KDBNet to learn structure representations directly from the raw atom coordinates in $\mathbb{R}^3$, without requiring the construction of features invariant to rotations and translation, such as relative direction embeddings. In the GNN, the GVP layer can be used as a drop-in replacement of MLPs (multi-layer perceptrons), such as $\boldsymbol{W}_k^{(\ell)}$ in the protein GNN (Eqn. 3.3).

Formally, we use the tuple $\boldsymbol{v}_i = (\boldsymbol{v}_i^v, \boldsymbol{v}_i^s)$ to denote the node feature of atom i, where $\boldsymbol{v}_i^v \in \mathbb{R}^{\mu \times 3}$ is a list of vector features in $\mathbb{R}^3$ and $\boldsymbol{v}_i^s \in \mathbb{R}^{\nu}$ is a list of scalar features. The edge feature $\boldsymbol{e}_{ij} = (\boldsymbol{e}_{ij}^v, \boldsymbol{e}_{ij}^s)$ of edge (i, j) has similar meaning. The molecule GNN transforms the node and edge features through L graph convolution layers to obtain the representation of the input molecule. Specifically, in the i-th layer, each node aggregates "messages" (embeddings) from neighboring nodes and edges and then updates its own representations:

$$\boldsymbol{h}_i^{(\ell)} = \boldsymbol{h}_i^{(\ell-1)} + g\left(\boldsymbol{h}_i^{(\ell-1)} + \frac{1}{|\mathcal{N}(i)|}\sum_{j \in \mathcal{N}(i)} \boldsymbol{m}_{ji}^{(\ell)}\right), \tag{3.5}$$

where $g(\cdot)$ is a sequence of three GVP layers, $\mathcal{N}(i)$ is the set of neighbor nodes of i in the $\mathcal{G}_d$, $\boldsymbol{h}_i^{(\ell)}$ is the embedding of node i in layer ℓ (in particular, $\boldsymbol{h}_i^{(0)} = \boldsymbol{v}_i$ is the node feature), and $\boldsymbol{m}_{ji}^{(\ell)}$ is the "message" passed from node j to node i, computed using another sequence of GVP layers: $\boldsymbol{m}_{ji}^{(\ell)} = g(\text{concat}(\boldsymbol{h}_j^{(\ell-1)}, \boldsymbol{e}_{ji}))$. Similar to the protein GNN, after the final layer of the molecule GNN, we also apply the global add pooling operation to aggregate all node representations into a scalar representation $\boldsymbol{h}^d \in \mathbb{R}^{128}$ of the input drug.

Prediction module and training details The two representations of protein and drug, $\boldsymbol{h}^p$ and $\boldsymbol{h}^d$, are first projected to dimension 128 using two fully connected layers with sizes 1024 and 128 and a dropout rate of 0.25. The two projected embeddings are then concatenated and passed to a 2-layer fully connected neural network with sizes 1024 and 512 and a dropout rate of 0.25, followed by a single scalar output as the predicted binding affinity between the input protein and drug. The training objective of KDBNet is to minimize the mean squared error (MSE) between the predicted binding affinity and the true affinity value. The model is trained using the Adam optimizer with a learning rate of 0.0005. Using inner-loop cross-validation on the training data, we decided to use three layers with sizes 128, 256,

and 256 for the protein GNN and three layers with uniform size 128 for the molecule GNN, which were robust across different settings in our experiments. Other hyperparameters, such as the dimensions of hidden layers, were also selected by performing a small-scale grid search using nested cross-validation on training data only. We trained all models for 500 epochs.

3.3.4 Uncertainty quantification

From a practical perspective, it is desirable that a machine learning model can also provide an associated uncertainty, in addition to the predicted affinity, so that researchers are able to assess how reliable is the hypothesis and how likely it will succeed in experimental validation. For that reason, we also equipped KDBNet with an uncertainty quantification module. This was achieved by training an ensemble of $M = 8$ independent model replicates [134], which has been widely demonstrated as an effective way to estimate uncertainty [152]. Specifically, let $\hat{y}_k(\boldsymbol{x}_i)$ be prediction given by the k-th individual model, where $\boldsymbol{x}_i$ represents a kinase-drug pair. KDBNet's final prediction of binding affinity $\mu(\boldsymbol{x}_i)$ and its estimated uncertainty $\sigma(\boldsymbol{x}_i)$ are given by the mean and standard deviation of the individual model's predictions:

$$\mu(\boldsymbol{x}_i) = \frac{1}{M}\sum_{k=1}^{M}\hat{y}_k(\boldsymbol{x}_i), \quad \sigma(\boldsymbol{x}_i)^2 = \frac{1}{M}\sum_{k=1}^{M}(\hat{y}_k(\boldsymbol{x}_i) - \mu(\boldsymbol{x}_i))^2 \tag{3.6}$$

KDBNet estimates the uncertainty by computing the standard deviation of predictions given by individual neural networks in the ensemble. The standard deviation quantifies the uncertainty within the predictive model, as known as epistemic uncertainty. It is possible to extend KDBNet to quantify the uncertainty in the data, known as aleatoric uncertainty, by modeling the affinity measurement as a Gaussian variable and decoupling the Gaussian variance into epistemic and aleatoric uncertainties [134]. In addition to the miscalibration area-minimizing approach used in the work, other calibration techniques [153, 154, 155] for deep learning models can also be combined with KDBNet to calibrate the uncertainty estimation. One limitation of KDBNet is the additional computation cost incurred due to the training of multiple independent neural networks for the purpose of uncertainty estimation. While this can be addressed by exploring the recent uncertainty quantification strategy that only requires training a single model [156], in our current implementation, we trained multiple independent neural networks in parallel on multiple GPUs, which has reduced the training time that would have been required by a sequential training.

3.3.5 Error curve at various uncertainty cutoffs

One way to evaluate the uncertainty estimation is by considering how well are they consistent with the prediction error. A meaningful uncertainty estimation should produce low errors on the subset of high-confidence predictions. We followed an evaluation scheme in previous studies [156, 157] to test whether the estimated uncertainties correlate with prediction errors, we sorted predictions $\hat{y}_i$ in an order represented by index $\{r_i\}$, such that $\hat{y}_{r_i}$ has the r-th highest estimated uncertainty (i.e., $\hat{y}_{r_1}$ is the most uncertain prediction while $\hat{y}_{r_n}$ is the most confident prediction with n being the total number of test samples). For every value i, we computed the cumulative mean squared error (MSE) for the test samples $\{y_{r_j} : j \geq i\}$. This metric evaluates the model's prediction accuracy at various confidence cutoffs. For example, setting $i = \lceil 0.5n \rceil$ gives the MSE at the 50% confidence cutoff. We repeated the evaluation with five independent trials and plotted the cumulative MSE as a function of different confidence cutoffs.

3.3.6 Uncertainty recalibration

There are two mainstream definitions of regression calibration in the literature: confidence interval-based calibration [153] and error-based calibration [154]. Under confidence-based calibration, a model is said to be well-calibrated if $e\%$ of its predictions fall in the $e\%$ predicted confidence interval ($0 \leq e \leq 1$) [153], while the error-based calibration defines a well-calibrated model if its uncertainty estimate of a prediction, in expectation, equals the prediction errors [154]. Several approaches have been proposed to recalibrate regression models [153, 154, 155]. The general idea is to learn a post hoc transformation function, which receives the model's predicted uncertainties as input and outputs the transformed uncertainty estimates that would be better calibrated. In our method, we used a simple yet effective scaling approach [154, 158] to recalibrate the uncertainty estimates. Specifically, we transformed the model's output $(\mu(\boldsymbol{x}_i), \sigma(\boldsymbol{x}_i))$ to $(\mu(\boldsymbol{x}_i), r\sigma(\boldsymbol{x}_i))$, where r is the scaling factor to be learned. Note that the model's prediction of binding affinity $\mu(\boldsymbol{x}_i)$ does not change. To learn the scaling factor r, we introduced an optimization problem where the objective is to minimize the miscalibration area. The recalibration is a post hoc process, meaning that the model's predicted uncertainties are fixed and only r is optimized. As suggested previously [153], the recalibration was performed on a held-out validation set that has not been used for model training. We used Brent's method [159] implemented in the SciPy package [160] to solve this single-variable optimization.

3.3.7 Calibration curve and miscalibration area

The ranking-based evaluation mentioned above does not take into account the actual values of uncertainty estimates. Another important and more strict criterion for uncertainty estimation is calibration, which relates to the magnitude of uncertainty estimates and assesses whether the predicted probability distributions are consistent with the observed empirical frequencies. The ideal can be illustrated by an intuitive example: if the well-calibrated model predicts a $x\%$ confidence interval, empirically the true data points would fall within this interval $x\%$ of the time. Having data points falling within the interval more or less $x\%$ time corresponds to over- or under-confident uncertainty estimation.

To evaluate the calibration of uncertainty estimates, we followed a widely used procedure used in previous studies [156, 157, 161]. We interpreted each prediction and its uncertainty as the mean and variance of a Gaussian distribution. Given a confidence interval of level e, we computed the $x\%$ confidence interval boundaries of each data point x_i using the inverse CDF function F_i^{-1} of Gaussian distribution, i.e., $L_i = F_i^{-1}(0.5 - e/2)$ and $R_i = F_i^{-1}(0.5 + e/2)$. For a calibrated model, we would expect that the fraction of ground truth data points that fall in that interval is e. To find the empirical or observed fraction, we count the fraction of ground-truth data points falling in the interval, that is $\hat{f}_e = |\{y_i | L_i \leq y_i \leq R_i\}|$. Plotting $\hat{f}_e$ against e for $0 \leq e \leq 1$ gives a curve called calibration curve. The calibration curve of a perfectly calibrated predictive model is the diagonal line. Therefore, to quantify the degree of uncertainty calibration, we computed the area between the model's calibration curve and the parity line, which is called the miscalibration area. We repeated the evaluation with five independent trials to obtain the mean calibration curve and miscalibration area.

3.3.8 Kinase structure and sequence

We consider both the structure and sequence of a kinase in KDBNet. Most protein kinases share a common structural fold that consists of two lobes: an N-terminal lobe with five beta-sheets and a C-helix, and a C-terminal lobe formed by six alpha-helices. Connecting the two lobes is a flexible region that forms the ATP and substrate binding site. The activation loop in this region, typically in a length of 20-30 residues, is crucial for binding activity. The same protein kinase may have different conformations because the loop can fold into catalytically active and inactive states. Therefore, there are multiple major structure conformations for a protein kinase in the Protein Data Bank (PDB) database.

We first selected the representative PDB structure for a kinase. In a recent study [162], Modi and Dunbrack define a classification and nomenclature for the active and inactive states

of a protein kinase. The authors further released the Kincore database [163] that presents the conformation classification for kinase structures in the PDB. For each kinase, we then selected the conformation with the most PDB structures as the representative conformation of this kinase. If that conformation has multiple PDB structures, we used the one with the highest resolution and the lowest number of missing residues in the structure.

After selecting the representative PDB structure for each kinase, we obtained the pocket of each structure by utilizing the KLIFS database [133]. Based on the analysis of 1200 kinase-ligand binding crystal structures, KLIFS defined a pocket formed by 85 residues that cover the binding sites to a wide range of kinase inhibitors. We extracted the substructure of the pocket from the PDB structure of each kinase. In KDBNet, we only used this substructure, instead of the entire structure of the kinase, as the structure information of a kinase to the model, because 1) residues in the pocket directly interact with the drug molecule, largely determining the binding activity; and 2) structure elements outside the pocket, i.e., the N- and C-terminal lobes, are relatively conserved across different kinases and may not directly coordinate the binding as they are relatively far from the binding sites. In total, we downloaded the pocket structure of 283 kinases.

The sequence of amino acids of the 85 residues in the binding pocket of a kinase was obtained from its reference protein sequence in UniProt. We did not the associated sequence in the PDB file because may contain missing or inaccurate residues. To do this, we mapped the PDB pocket sequence to the full UniProt sequence using pairwise local alignment (score matrix: BLOSUM62, gap open penalty: 10, and gap extend penalty: 0.5). We successfully mapped 281 out of 283 structures to the UniProt sequences.

3.3.9 Kinase-compound binding datasets

For evaluation, we used two public datasets of experimental measurements of binding affinity, Davis [144] and KIBA [145], which were widely used to benchmark previous methods [124, 125, 128, 129, 130]. The Davis study contains binding affinity measurements (K_d values ranging from 0.1 to 10,000 nM) of kinase-compound pairs, while the KIBA dataset derived a score to integrate three bioactivity values, including K_i, K_d, and IC_{50}. We removed compounds and kinases in both datasets that do not have 3D structures in PDB or PubChem databases (statistics in Tables A.3a and A.3b). Due to the importance of kinases in therapeutics, we expect that more kinases will have a solved structure in the near future. We obtained the Davis [144] and KIBA [145] datasets curated in the Therapeutics Data Commons resource [164]. For kinases that have multiple mutant types in the Davis dataset, we only keep the minimal K_d values (strongest binding), as suggested by the KIBA study [145].

Raw K_d values in the Davis dataset were transformed to pK_d values $\text{p}K_d = -\log_{10}(K_d/10^9)$ to facilitate the model training as suggested by previous studies [125, 141]. For model training and MSE-based evaluation in this work, we used pK_d as the binding affinity labels, but for the evaluation of top acquisition experiment, we converted the labels back to the original K_d values, consistent with a similar experiment performed in Hie et al. [130].

We used three train-test split settings to evaluate prediction performance: i) New-protein split: 20% of kinases were completed withheld as the test set, which simulates the drug repositioning scenario where the model predicts for unseen proteins; ii) New-Drug split: 20% of drugs were completed withheld as test set, which simulates the drug discovery scenario where the model predicts for unseen drugs; iii) Both-new split: 20% of kinases and 20% drugs were completed withheld as test set, which simulates the scenario where the model predicts for unseen proteins and drugs.

3.3.10 Baseline methods

We compared KDBNet to the following baseline methods.

DeepDTA. DeepDTA [125] is a deep learning model that receives 1D protein sequences and 1d compound SMILES strings as inputs and uses two branches of convolutional neural networks (CNNs) to process the protein and compound input, respectively. Embeddings produced by the two CNNs are concatenated and passed to a multi-layer fully connected neural network to predict binding affinity.

GraphDTA. GraphDTA [129] is a deep learning model that represents proteins by 1D sequences and compounds by 2D molecule graphs. GraphDTA has a similar model architecture as DeepDTA but replaces the compound CNN with a graph neural network (GNN).

DGraphDTA. DGraphDTA [128] extends both DeepDTA and GraphDTA in that it uses 2D protein contact maps and 2D molecule graphs as input. For the model architecture, DGraph uses two GNNs to learn features of the input protein and compound, respectively.

KronRLS. KronRLA [165] is a kernel-based method that predicts kinase-drug binding affinity based on drug similarity and protein similarity. The drug similarities were computed based on the Tanimoto distance of pairwise molecule fingerprints. The protein similarities were computed using the normalized Smith-Waterman alignment score [124].

GP and *GP-MLP*. Both GP and GP-MLP [130] implement the Gaussian process regression to predict kinase-drug binding affinity. GP directly uses the Gaussian process to predict the affinity, while GP-MLP first uses a multi-layer perception (MLP) to fit the affinity and then the Gaussian process to fit the residuals of MLP prediction. Both GP and GP-MLP use language model-based protein sequence features [68] and graph neural network-based

compound features [166] generated by independent pre-trained models.

3.4 RESULTS

We performed several evaluation experiments to show that KDBNet provides accurate affinity predictions of kinase-drug binding as well as informative and calibrated uncertainty estimation of the predictions. Furthermore, we demonstrated two applications of KDBNet, including an active learning for data-efficient learning guided by uncertainties and Bayesian optimization for accelerated discovery.

3.4.1 Accurate prediction of kinase-drug binding affinity

We first assess KDBNet's performance in predicting kinase-drug binding affinity using the Davis and KIBA datasets. We created three five-fold cross-validation settings to simulate out-of-distribution scenarios where the training and test sets do not share any drugs or proteins, or both (Figure 3.2a and Methods). We compared KDBNet with several state-of-the-art methods for predicting kinase-drug binding affinity, including three deep learning-based methods [125, 128, 129], a Gaussian process (GP)-based method [130], and a kernel-based method [165]. Those baseline methods do not consider 3D structure information, rather they only used 1D and 2D representations or pairwise similarities of compounds and proteins (Methods).

The results of cross-validation experiments (Figures 3.2 and B.12) suggested that KDBNet was consistently better than other methods in terms of several metrics including Pearson correlation, Spearman correlation, and mean squared error (one-sided rank-sum test $P < 10^{-3}$). The improvements achieved by KDBNet were also consistent across different split settings. For example, on the new-protein test set, KDBNet received a Pearson correlation of 0.77, in contrast with 0.52 and 0.58 for DGraphDTA and GP, respectively. Compared with other methods, KDBNet also achieved higher Spearman correlation and lower MSE in all settings. We had similar observations on the larger KIBA dataset, where KDBNet still outperformed the baseline methods in three split settings and all metrics (Figure B.14). The improvements of KDBNet mainly stem from the direct modeling of 3D structures of protein and molecule in the neural network, confirmed by our ablation study where structure data was dropped (Figure B.13). The 3D structure data provide more explicit information related to the binding activity, which might not be fully reflected by the 1D or 2D features used by the baseline methods. Compared to GP which used fixed feature embeddings, the improvements of KDBNet also suggested that learning embeddings in an end-to-end

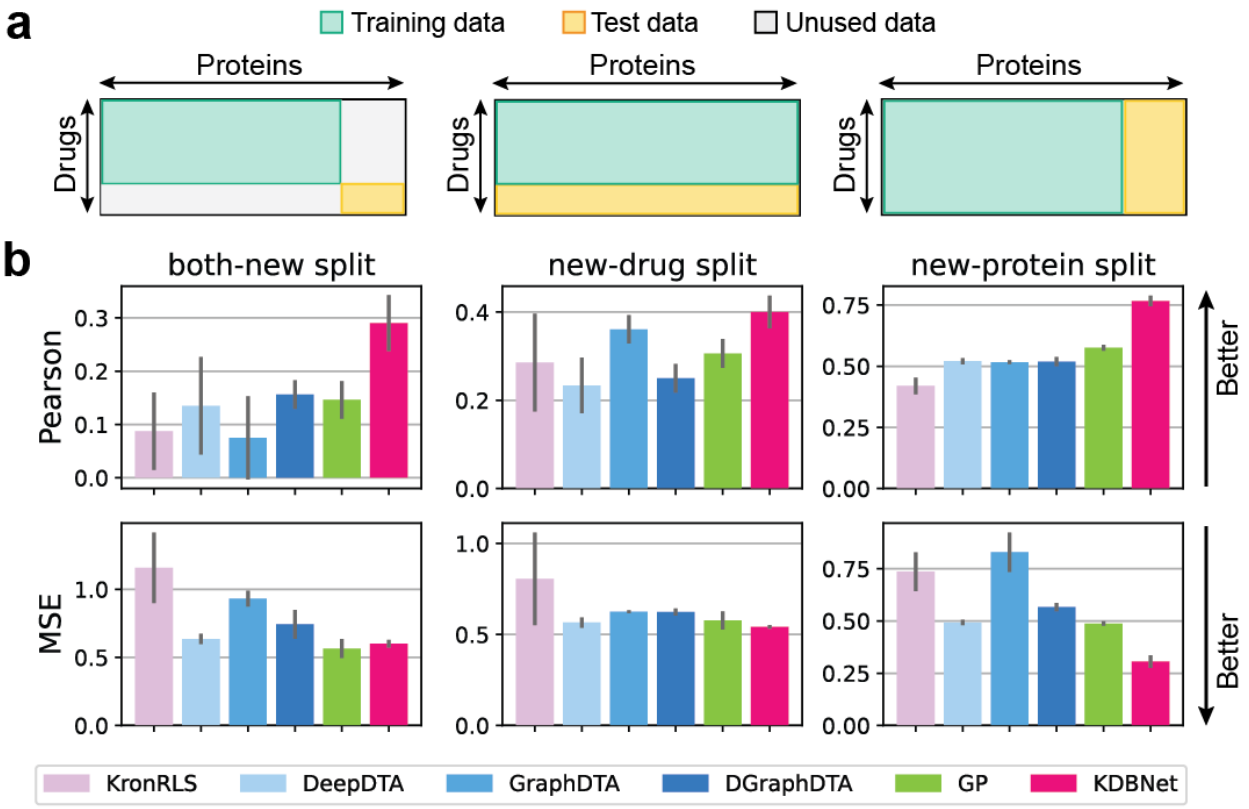


Figure 3.2: KDBNet achieves accurate prediction of kinase-drug binding affinity. (**a**) Three train-test split settings of five-fold cross-validation where the model is evaluated on data of unseen drugs, proteins, or both. (**b**) Comparison of prediction performance of KDBNet with KronRLS, DeepDTA, GraphDTA, DGraphDTA, and GP on the three train-test split settings. Performances were evaluated using Pearson correlation and mean squared error (MSE) between predicted and true pK_d values.

manner captures the features more effectively. Integrating 3D information also enables KDBNet to generalize to predictions for unseen proteins and unseen drugs or both. Overall, these results demonstrated that KDBNet, by incorporating 3D structure data and leveraging geometry-aware deep learning, had substantial performance improvements in kinase-drug binding prediction as compared to multiple existing methods.

3.4.2 Informative and calibrated uncertainty estimation

After evaluating the prediction accuracy of KDBNet, we next performed experiments to demonstrate the uncertainty estimation of KDBNet is accurate and calibrated. First, we sought to investigate whether KDBNet's uncertainty estimate is indicative of prediction accuracy. On the Davis dataset, we ranked all KDBNet's predictions by their associated

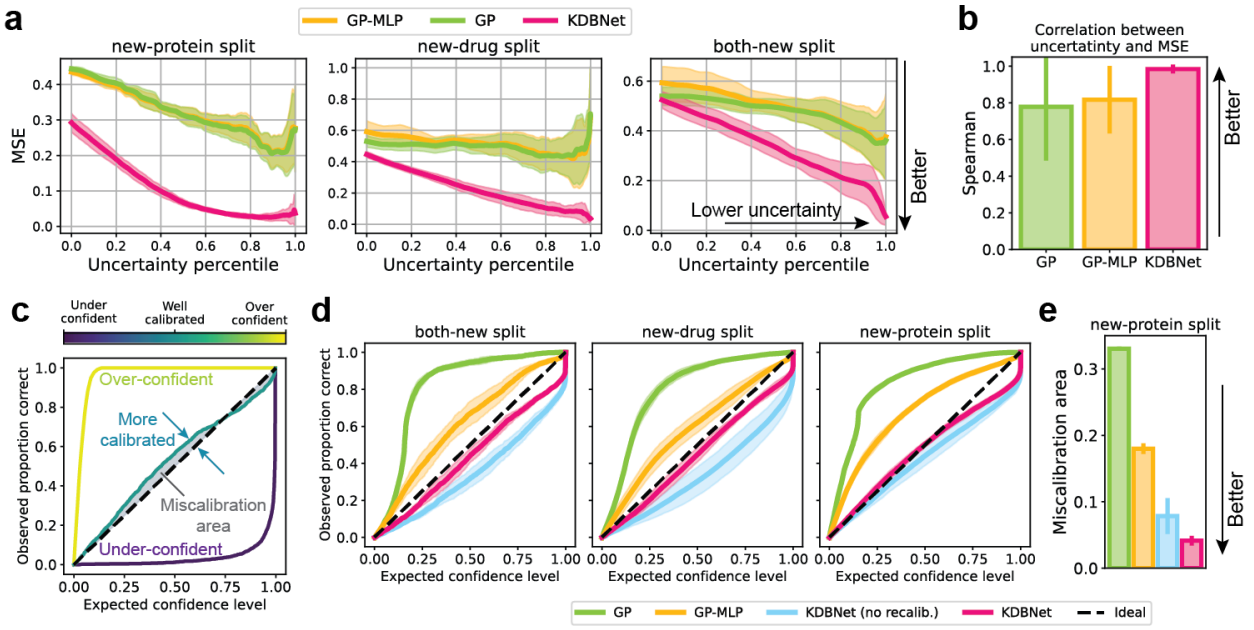


Figure 3.3: KDBNet provides accurate and calibrated uncertainty estimation. (**a**) Prediction errors of KDBNet, GP, and GP-MLP, measured as mean squared error, at different cutoffs of uncertainty percentiles. The x-axis represents the sorted uncertainty such that the 100% percentile is the lowest uncertainty (highest confidence). (**b**) Spearman correlation between the estimated uncertainty and the prediction error measured in MSE on the both-new test set. (**c**) An illustration of the calibration curve. For a confidence interval of confidence level e ($0 \leq e \leq 1$), the curve shows the expected fraction and the observed fraction of test points that fall within that interval. The diagonal line corresponds to the calibration curve of a perfectly calibrated model. The area between a curve and diagonal line (miscalibration area) is used to quantify the uncertainty calibration, and lower values indicate better calibration. (**d**) Calibration curves of KDBNet, KDBNet without recalibration, GP, and GP-MLP on test sets. (**e**) Miscalibration area of KDBNet, GP, and GP-MLP on the new-protein test set.

uncertainty estimates (Methods). We observed that there was a clear trend that KDBNet's predictions with lower uncertainty had lower prediction errors. The same trend was observed in different train-test split settings (Figure 3.3a). Compared to the two Gaussian process-based methods, GP and GP-MLP [130] (Methods), KDBNet achieved substantially lower MSE across different uncertainty percentiles (Figure 3.3a) and higher correlations between the estimated uncertainty and prediction errors (Figures 3.3b and B.15a). These results suggested that KDBNet's uncertainty estimates were correctly ranked with respect to prediction errors. In the top 10% of predictions where KDBNet had the most confidence (uncertainty percentile 90%-100%), its average MSE of predictions were 0.03, 0.07, and 0.10 for the new-protein, new-drug, and both-new split settings, respectively, in contrast to the global average MSEs 0.30, 0.54, and 0.60 for the three splits, respectively. This suggested that KDBNet's predictions were very accurate when it has high confidence.

The above evaluation validated that the *ranking* of KDBNet's uncertainty estimates is indicative. We next proceeded to assess if the *magnitude* of the uncertainty estimates is quantitatively similar to the true error. Models that are over-confident or under-confident usually produce uncertainty estimates that too small or large, and it is difficult to interpret them as statistically meaningful credible intervals. This issue is known as miscalibration in uncertainty quantification [153]. Ideally, we hope the model's uncertainty estimation is well-calibrated, meaning that, for example, if we predict a 95% confidence interval, we want the true values to fall within the interval 95% of the time. Following previous studies [136, 153, 161], we computed calibration curves that compare the observed fraction of data points falling in a confidence interval with the expected fraction (Methods). To quantify the degree of uncertainty calibration, we calculated the area between the model's calibration curve and the diagonal line corresponding to the perfectly calibrated model (Figure 3.3c), which is called miscalibration area and a lower value indicating a better calibration. We observed that KDBNet's calibration curves were more close to the ideal diagonal curve (Figure 3.3d), yielding substantially lower miscalibration areas than the GP-based methods (Figures 3.3e and B.15b). The recalibration technique of KDBNet also led to effective improvements, clearly pushing the calibration curves more close to the diagonal and decreasing the miscalibration area (Figures 3.3d-e and B.15b; rank-sum test $P < 10^{-3}$). These results suggested that KDBNet's uncertainty estimates were calibrated and scaled with errors.

Together, the two sets of experiments demonstrated that the uncertainty estimates of KDBNet were accurate with respect to prediction errors and well-calibrated. The accurate quantification of uncertainty has important implications for iterative ML-guided experiment design, where the uncertainty estimates can be used to guide the data acquisition and candidate prioritization, as we will illustrate in the next section.

3.4.3 Uncertainty-guided active learning for data-efficient learning

Having verified that KDBNet provides informative and calibrated uncertainty estimation, we sought to assess the value of uncertainty in machine learning guided discovery. The first natural application is active learning, where the objective is to select training samples intelligently such that better prediction performance could be achieved with less training data. Active learning simulates the scenario in drug discovery in which researchers repeat the select-validate cycle to discover promising compounds. In analogous to human experts relying on intuitive confidence to acquire and test new samples, KDBNet uses its estimated uncertainty to perform iterative training and selection (Figure 3.4a). Specifically, we started training KDBNet on a random 1% subset of KIBA training data. At each subsequent round,

KDBNet predicted binding affinities and uncertainties for the remainder of training data, and then ranked them based on the score function $s(\boldsymbol{x}) = \sigma(\boldsymbol{x})$, where $\sigma(\boldsymbol{x})$ is the predicted uncertainty for sample $\boldsymbol{x}$ (hereinafter referred to as "explore" strategy). We then added the top T samples with the greatest uncertainties to the training set and retrained KDBNet with the expanded training set from scratch. Two other types of score functions $s(\boldsymbol{x})$ were considered for comparison: i) "greedy": samples of higher predicted affinity receive higher scores, $s(\boldsymbol{x}) = \mu(\boldsymbol{x})$; ii) "random": samples receive random scores, $s(\boldsymbol{x}) \sim \mathcal{U}(0, 1)$. The performance was evaluated on the "both-new" test set.

We found that KDBNet achieved efficient active learning by using its estimated uncertainty to acquire new training samples, reaching the performance on par with full data training by using only 50% data (Figrue 3.4b). Noticeably, the KDBNet's performance drastically increased in the first few rounds compared to the random selection, suggesting that the uncertainty-based active learning is more efficient than the brute-force random search. Furthermore, in contrast to the greedy strategy that kept seeking samples with the highest affinities, the explore strategy focused on samples that could promote the diversity of the training set and best address the uncertainties of the model, thereby exhibiting faster rates of performance improvement and higher efficiency gains (performance improvement over random selection) across all active learning stages (Figure 3.4b-c). These results suggested that, enabled by uncertainty quantification, KDBNet has achieved sample-efficient active learning for data acquisition and model training, a highly desired capability in model-guided experimental design where the naive enumeration of the search space is costly or even intractable.

3.4.4 Bayesian optimization for accelerated exploration and exploitation

As another application of uncertainty estimation, we integrated KDBNet with Bayesian optimization for the exploration and exploitation of strong-binding kinase-drug pairs. In the active learning experiments, new samples were acquired solely based on uncertainty to promote the diversity of the training set. However, for prioritization or discovery of high-performing candidates, one can combine the predicted mean values and uncertainties to guide the data acquisition. A principle framework to achieve this goal is Bayesian optimization, which allows us to prioritize candidates in high confidence, high desirability regions ("exploitation") or to probe in potentially high desirability regions although with less confidence ("exploration"), as illustrated in Figure 3.4d. In Bayesian optimization, a widely used way to combine predictive scores and uncertainties is through an acquisition function called upper confidence bound (UCB) with the form $\mathsf{UCB} = \mathsf{score} + \beta \cdot \mathsf{uncertainty}$, where

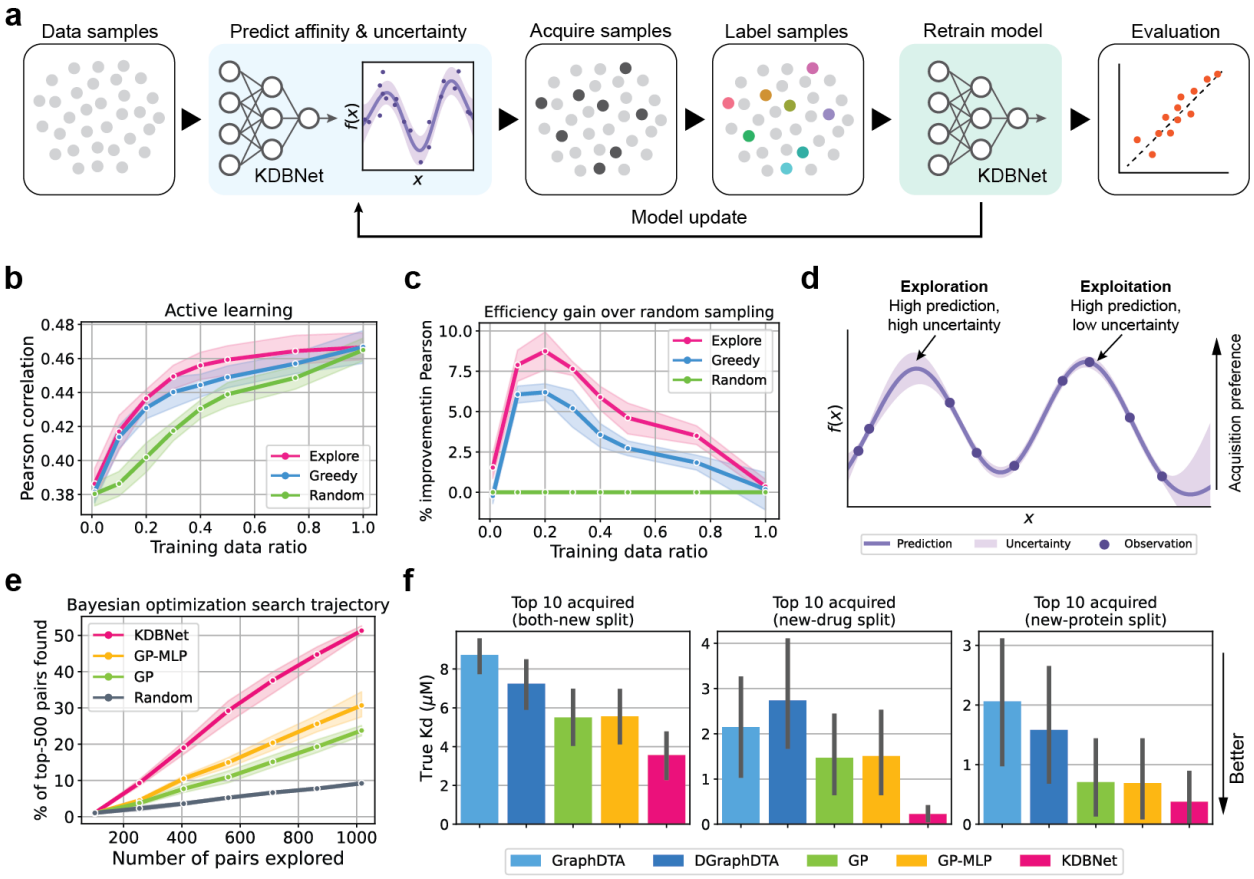


Figure 3.4: Leveraging uncertainty for active learning, exploration, and exploitation. (**a**) Schematic visualization of the active learning process that consists of multiple rounds of model training, data acquisition, and model evaluation. (**b**) Active learning performance in Pearson correlation on the KIBA both-new test set at different rounds. The explorative sampling is compared to the greedy and random sampling strategies. (**c**) The efficiency gain of the explorative and greedy samplings over the random sampling, defined as the relative improvement in Pearson correlation. (**d**) Schematic illustration of data acquisition based on KDBNet's prediction and uncertainty. One can exploit regions with high-confidence, high-desirability samples, or explore in potentially high-desirability regions with less model confidence. (**e**) Exploration using KDBNet and upper confidence bound (UCB) acquisition function with a Bayesian optimization framework. Curves represent the performance trajectory, measured by the percentage of top-500 binding affinities found as a function of the number of kinase-compound pairs explored in the Davis dataset. (**f**) Exploitation using KDBNet and Bayesian optimization. True K_d values of the top 10 kinase-drug pairs prioritized by each model are shown. A lower K_d value means a stronger binding affinity.

the constant β controls the tradeoff between exploitation and exploration.

High-recall exploration: We first evaluated KDBNet's ability in terms of exploration. We designed an experiment on the Davis dataset where the aim is to identify kinase-compound pairs with the strongest binding affinity by only observing the ground truth binding affinities for a small subset of pairs. We initiated the data acquisition process by

training KDBNet on 1% kinase-compound pairs (∼100 pairs). The subsequent steps were performed as in the active learning experiments, but with UCB as the score function, defined as $\text{UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \beta\sigma(\boldsymbol{x})$ with $\beta = 1$. Intuitively, this score function promotes samples with high binding affinity and high uncertainties. Since our goal is to identify strong-binding pairs as many as possible, we want to explore "good" regions that also have some variability (uncertainty) as this would increase the likelihood of discovering even better samples. We observed that KDBNet yields substantial improvements over the random exploration baseline, as measured by the recall of top-500 kinase-compound pairs (top 1%) as a function of the number of pairs explored. Specifically, KDBNet retrieved 50% of the top-500 pairs from the pool of 10k pairs after exploring only 1k pairs. KDBNet also outperformed the two GP-based baselines by a clear margin. This experiment demonstrated the applicability of KDBNet within the Bayesian optimization framework for accelerating the exploration and discovery of strong-binding kinase-compound pairs.

High-confidence exploitation: Next, we performed an analysis to evaluate KDBNet's ability in terms of exploitation, i.e., how strong are the binding affinities of top kinase-compound pairs prioritized by KDBNet. This mimics the real scenario of a biological discovery process, where researchers typically focus on only a few top compounds or proteins for further validation rather than testing the entire unexplored space. To directly test this, we simulated an experiment on the Davis dataset where a model was asked to prioritize kinase-compound pairs that have the strongest binding affinity from the test data. For KDBNet, we defined the UCB score function as $\text{UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \beta(-\sigma(\boldsymbol{x}))$ with $\beta = 1$. Note that we added a negation sign before the uncertainty term. Intuitively, this score function encouraged KDBNet to prioritize pairs with strong binding affinity and low uncertainty. We used the greedy score function for baselines without uncertainty estimation and the rank-based UCB function for GP-based baselines [130]. We showed the binding affinities of the top 10 kinase-drug pairs acquired by different methods in K_d values in Figure 3.4f, where lower K_d values indicate stronger binding affinities. We found that KDBNet, on average, has retrieved pairs with stronger binding affinity, outperforming other baseline methods in all three split settings. KDBNet successfully prioritized kinase-drug pairs with a mean K_d value lower than 0.5 μM for both the new-protein and new-drug split settings, and a mean K_d of 3.5 μM for the most challenging both-new split setting. For reference, a K_d value lower than 3 μM was considered as a very strong binding in the original stud [144]. In addition, we observed that uncertainty-based methods (KDBNet, GP, and GP-MLP) had a clear advantage over the other two no-uncertainty methods in terms of prioritization of strong-binding kinase-compound pairs. As the binding affinity datasets often contain intrinsic noise or imbalanced measurements towards frequent compounds or proteins, ML models

could be largely impacted by those issues and generate uncertain predictions. Therefore, top predictions given by uncertainty-agnostic models could be false positive in many cases. In contrast, the estimated uncertainty provides another dimension of information, in addition to the mean prediction of affinity, for prioritizing lead candidates that are more likely to succeed with high confidence.

3.5 DISCUSSION

In this work, we have presented KDBNet, a geometric deep learning algorithm for predicting protein-drug affinity. KDBNet integrates the 3D structure data of both proteins and compounds and models them using structure-aware graph neural networks. Experiments have demonstrated that KDBNet is more accurate than several existing deep learning methods in predicting kinase-drug binding affinity. KDBNet also provides well-calibrated uncertainties that scale with prediction errors and confidence intervals that are statistically indicative. We further validated the KDBNet's utility in active learning and Bayesian optimization for prioritizing kinase-drug pairs with strong binding affinities. We anticipate that KDBNet could facilitate the reliable and robust deployment of machine learning-guided discovery of kinase-drug pairs with novel or enhanced binding activities.

While we focused on the binding activities of kinases, the methodology, in principle, can be applied for general proteins. We considered kinase-drug binding in this work considering the pharmacological importance of kinases and the availability of public datasets of binding affinity measurements. The problem setting in this work was different, and actually less restrictive, than another line of recent studies on binding affinity prediction based on protein-compound binding complex [137, 138, 139, 140, 141, 142], in that KDBNet does not require binding complex structures as input and can be generalized to predict for novel molecules or targets without solved complex structures. We emphasize that the problem setup considered in this work is different from that in recent studies on binding affinity prediction based on protein-compound binding complex [137, 138, 139, 140, 141, 142]. The prerequisite of those methods is the solved structure of binding complex [167] for every protein-compound pair, of which the availability is rather limited. Due to the same reason, it is difficult to apply those methods in active learning or Bayesian optimization frameworks to predict the binding affinities for novel molecules or targets for which binding complex structures do not exist. KDBNet is less restricted by data availability than those methods in that it only requires separate structures as input rather than the binding complex. We note that this is not a critical limitation of KDBNet. First, the 3D structures s of compounds are largely available: it is estimated that 92% of compounds available in the PubChem database have

3D structure [168]. Second, kinases are one of the most representative protein families in the Protein Data Bank (PDB) database, and due to their biological and pharmacological importance, the available solved structures of kinases are keeping increasing in number, especially for kinases that are widely used as drug targets [133, 163]. Third, AlphaFold has been shown to predict the 3D protein structure in near experimental resolution [50, 169] and its structure prediction might serve as a reasonable proxy for less-studied kinases without solved structures.

KDBNet estimates the uncertainty by computing the standard deviation of predictions given by individual neural networks in the ensemble. The standard deviation quantifies the uncertainty within the predictive model, as known as epistemic uncertainty. It is possible to extend KDBNet to quantify the uncertainty in the data, known as aleatoric uncertainty, by modeling the affinity measurement as a Gaussian variable and decoupling the Gaussian variance into epistemic and aleatoric uncertainties [134]. In addition to the miscalibration area-minimizing approach used in the work, other calibration techniques [153, 154, 155] for deep learning models can also be combined with KDBNet to calibrate the uncertainty estimation. One limitation of KDBNet is the additional computation cost incurred due to the training of multiple independent neural networks for the purpose of uncertainty estimation. While this can be addressed by exploring the recent uncertainty quantification strategy that only requires training a single model [156], in our current implementation, we trained multiple independent neural networks in parallel on multiple GPUs, which has reduced the training time that would have been required by a sequential training.

Chapter 4: Representation Learning of Protein Networks

Proteins cooperate with each other to carry out a large number of functions in cells. For example, kinase proteins form signaling networks to propagate environmental signals across cells to coordinate cellular processes. In Chapter 2 and Chapter 3, we discussed the representation learning of protein sequences and structures. In this chapter, we introduce a representation learning method for protein networks and more broadly biological networks. We show that network representations not only summarize multi-source information but also mitigate the noise and incompleteness of individual network data.

4.1 INTRODUCTION

Computational prediction of drug-target interactions (DTIs) has become an important step in the drug discovery or repositioning process, aiming to identify putative new drugs or novel targets for existing drugs. Compared to *in vivo* or biochemical experimental methods for identifying new DTIs, which can be extremely costly and time-consuming [170], *in silico* or computational approaches can efficiently identify potential DTI candidates for guiding *in vivo* validation, and thus significantly reduce the time and cost required for drug discovery or repositioning. Traditional computational methods mainly depend on two strategies, including the molecular docking-based approaches [171, 172] and the ligand-based approaches [173]. However, the performance of molecular docking is limited when the 3D structures of target proteins are not available, while the ligand-based approaches often lead to poor prediction results when a target has only a small number of known binding ligands.

In the past decade, much effort has been devoted to developing machine learning-based approaches for computational DTI prediction. A key idea behind these methods is the "guilt-by-association" assumption, that is, similar drugs may share similar targets and vice versa. Based on this intuition, DTI prediction is often formulated as a binary classification task, which aims to predict whether a drug-target interaction is present or not. A straightforward classification-based approach is to consider known DTIs as labels and incorporate the chemical structures of drugs and primary sequences of targets as input features (or kernels). Most existing prediction methods mainly focus on exploiting information from homogeneous networks. For example, Bleakley and Yamanishi [174] applied a support vector machine (SVM) framework to predict DTIs based on a bipartite local model (BLM). Mei et al. [175] extended this framework by combining BLM with a neighbor-based interaction-profile inferring (NII) procedure (called BLMNII), which is able to learn the DTI features

from neighbors and predict interactions for new drug or target candidates. Xia et al. [176] proposed a semi-supervised learning method for DTI prediction, called NetLapRLS, which applies Laplacian regularized least square and incorporates both similarity and interaction kernels into the prediction framework. van Laarhoven et al. introduced a Gaussian interaction profile (GIP) kernel-based approach coupled with regularized least square (RLS) for DTI prediction [177, 178].

In addition to chemical and genomic data, previous works have incorporated pharmacological or phenotypic information, such as side-effect [179, 180], transcriptional response data [181], drug-disease associations [182], public gene expression data [183] and functional data [184] for DTI prediction. Heterogeneous data sources provide diverse information and a multi-view perspective for predicting novel DTIs. For instance, the therapeutic effects of drugs on diseases can generally reflect their binding activities to the targets (proteins) that are related to these diseases and thus can also contribute to DTI prediction. Therefore, incorporating heterogeneous data sources, e.g., drug-disease associations, can potentially boost the accuracy of DTI prediction and provide new insights into drug repositioning. Despite the current availability of heterogeneous data, most existing methods for DTI prediction are limited to only homogeneous networks or bipartite DTI models, and cannot be directly extended to take into account heterogeneous nodes or topological information and complex relations among different data sources.

Several computational strategies have been introduced to integrate heterogeneous data sources to predict DTIs. A network-based approach for this purpose is to fuse heterogeneous information through a network diffusion process and directly use the obtained diffusion distributions to derive the prediction scores of DTIs [182, 185]. A meta-path based approach has also been proposed to extract the semantic features of DTIs from heterogeneous networks [186]. A collaborative matrix factorization has been developed to project the heterogeneous networks into a common feature space, which enables one to use the aforementioned homogeneous network-based methods to predict new DTIs from the resulting single integrated network [187]. However, these approaches generally fail to provide satisfactory integration paradigms. First, directly using the diffusion states as the features or prediction scores may easily suffer from the bias induced by the noise and high dimensionality of biological data and thus possibly lead to inaccurate DTI predictions. In addition, the hand-engineered features, such as meta-paths, often require expert knowledge and intensive effort in feature engineering and hence prevent the prediction methods from being scaled to large-scale datasets. Moreover, collapsing multiple individual networks into a single network may cause substantial loss of network-specific information, since edges from multiple data sources are mixed without distinction in such an integrated network.

In this paper, we present DTINet, a novel network integration pipeline for DTI prediction. DTINet not only integrates diverse information from heterogeneous data sources (e.g., drugs, proteins, diseases, and side-effects), but also copes with the noisy, incomplete, and high-dimensional nature of large-scale biological data by learning low-dimensional but informative vector representations of features for both drugs and proteins. The low-dimensional feature vectors learned by DTINet capture the context information of individual networks, as well as the topological properties of nodes (e.g., drugs or proteins) across multiple networks. Based on these low-dimensional feature vectors, DTINet then finds an optimal projection from drug space onto target space, which enables the prediction of new DTIs according to the geometric proximity of the mapped vectors in a unified space. We have demonstrated the integration capacity of DTINet by unifying multiple networks related to drugs and proteins, and shown that incorporating additional network information can significantly improve the prediction accuracy. In addition, through comprehensive tests, we have demonstrated that DTINet can achieve substantial performance improvement over other state-of-the-art prediction methods. Furthermore, we have experimentally validated the new interactions predicted by DTINet between three drugs and the cyclooxygenase (COX) proteins that have not been reported in the literature (to the best of our knowledge), and demonstrated the potential novel applications of these drugs in preventing inflammatory diseases. All these results demonstrate that DTINet can offer a practically useful tool to predict unknown DTIs from complex heterogeneous networks, which may provide new insights into drug discovery or repositioning and understanding of mechanisms of drug action.

4.2 DTINET: HETEROGENEOUS NETWORK INTEGRATION FOR DRUG-TARGET INTERACTION USING REPRESENTATION LEARNING

We develop a new computational pipeline, called DTINet, to predict novel drug-target interactions (DTIs) and thus identify new indications of old drugs from a heterogeneous network (Supplementary Figure B.16), which is constructed based on the following known information (Methods): drug-protein interactions, drug-drug interactions, drug-disease associations, drug-side-effect associations, drug-drug similarities, protein-disease associations, protein-protein interactions, and protein-protein similarities. DTINet (Figure 4.1) first performs a network diffusion algorithm (e.g., random walk with restart, RWR [188]) on each network to obtain a distribution (also called "diffusion state") of each drug or protein node, which captures its topological relations to all other nodes in the heterogeneous network (Methods). Taking both local and global connectivity patterns into account, this step characterizes the underlying topological context and inherent connection profiles of each drug or

protein node in the network. When the diffusion states of two nodes are close to each other, it implies that they are in similar positions with respect to other nodes in the network.

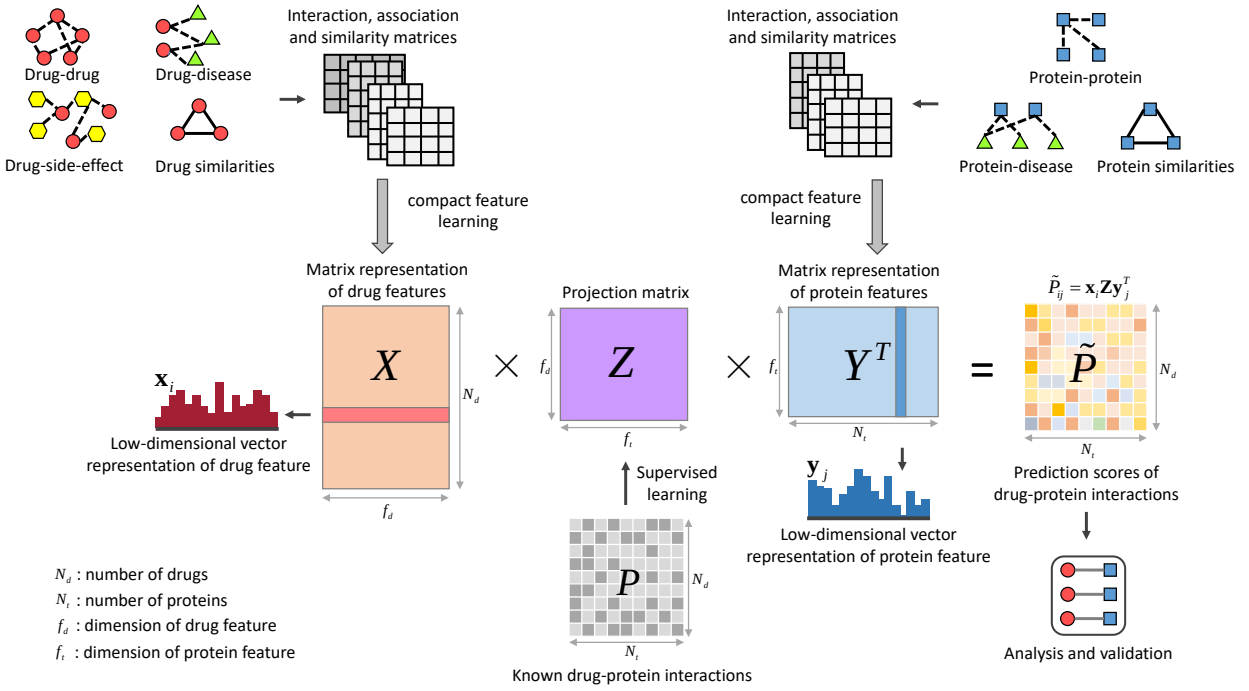


Figure 4.1: The flowchart of the DTINet pipeline. DTINet first integrates a variety of drug-related information sources to construct a heterogeneous network and applies a compact feature learning algorithm to obtain a low-dimensional vector representation of the features describing the topological properties for each node. With the learned compact features X and Y for drugs and proteins (i.e., each row in X and Y represents the feature vector of a drug and a protein, respectively), DTINet then finds the best projection from drug space onto protein space, such that the projected feature vectors of drugs are geometrically close to the feature vectors of their known interacting proteins. The projection matrix Z is learned to minimize the difference between the known interaction matrix P and XZY^T. After that, DTINet infers new interactions for a drug by sorting its target candidates based on their geometric proximity to the projected feature vector of this drug in the projected space. The predicted new drug-target interactions can be further analyzed and experimentally validated.

A key observation in the above network diffusion algorithm is that the originally computed diffusion states are not entirely accurate, in part due to the noisy, incomplete, and high-dimensional nature of biological data. To cope with this issue, DTINet further applies the diffusion component analysis (DCA) method [189, 190] to approximate the obtained diffusion distribution by constructing a model parameterized by a low-dimensional vector representation for each drug or protein node (Figure 4.1). These low-dimensional vector representations are obtained by minimizing the difference between the diffusion distributions of individual networks and the corresponding model distributions simultaneously (Figure 4.2).

Such a process is also called *compact feature learning* [191] (more details will be described below) and the resulting low-dimensional vector is also called the *feature vector*. Intuitively, the low-dimensional feature vector obtained from compact feature learning encodes the relational properties (e.g., similarity), association information, and topological context of each drug (or protein) in the heterogeneous network. Akin to principal component analysis (PCA), which seeks the intrinsic low-dimensional linear structure of the data to best explain the variance, DCA learns a low-dimensional vector representation for all nodes such that their connectivity patterns in the heterogeneous network are best interpreted.

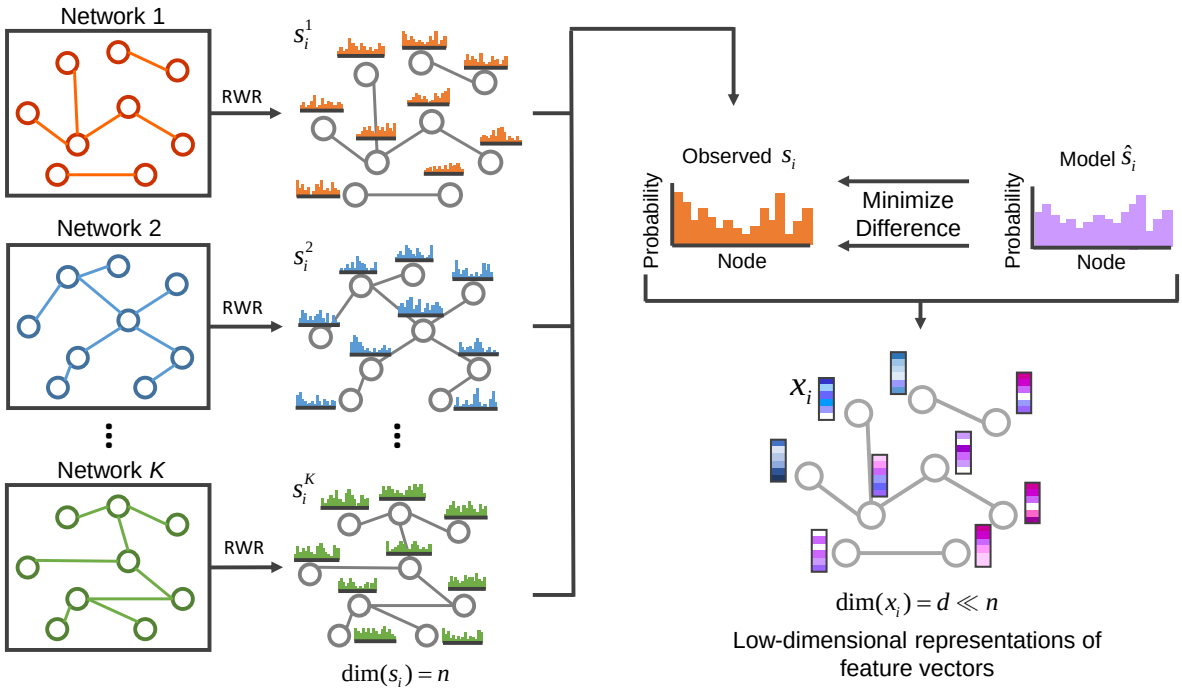


Figure 4.2: Schematic illustration of compact feature learning. The random walk with restart (RWR) algorithm is first used to compute the diffusion states of individual networks. Then the low-dimensional representations of feature vectors for individual nodes are obtained by minimizing the difference between the diffusion states s_i and the parameterized multinomial logistic models $\hat{s}_i$. The learned low-dimensional feature vectors encode the relational properties (e.g., similarity), association information, and topological context of each node in the heterogeneous network.

After obtaining the low-dimensional feature vectors of both drugs and proteins, DTINet finds the best projection from drug space onto protein space, such that the mapped feature vectors of drugs are geometrically close to their known interacting targets (Figure 4.1). After that, DTINet infers new interactions for a drug by ranking its target candidates according to their proximity to the projected feature vector of this drug. A key insight of this approach is that the drugs (or proteins) with similar topological properties in the heterogeneous network are more likely to be functionally correlated. For example, those drugs that are close in the

directions of their feature vectors are more likely to act on the same target, and vice versa. This intuition allows us to predict unknown drug-target interactions by fully exploiting our previous knowledge about known drug-target interactions.

4.3 METHODS

4.3.1 Datasets

A total of four types of nodes and six types of edges, representing diverse drug-related information, were collected from the public databases and used to construct the heterogeneous network for our drug-target interaction (DTI) prediction task.

Nodes. We extracted the drug nodes from the DrugBank database (Version 3.0) [192] and the protein nodes from the HPRD database (Release 9) [193]. The disease nodes were obtained from the Comparative Toxicogenomics Database [194]. The side-effect nodes were collected from the SIDER database (Version 2) [195]. In addition, we excluded those isolated nodes; in other words, we only considered those nodes which had at least one edge (see below) in the network.

Edges. We imported the known drug-target interactions as well as drug-drug interactions from DrugBank (Version 3.0) [192]. The protein-protein interactions were downloaded from the HPRD database (Release 9) [193]. The drug-disease and protein-disease associations were extracted from the Comparative Toxicogenomics Database [194]. We also included the drug-side-effect associations from the SIDER database (Version 2) [195].

4.3.2 Construction of a heterogeneous network by integrating diverse drug-related information

Compiling various curated public drug-related databases, we constructed a heterogeneous network, which includes 12,015 nodes and 1,895,445 edges in total, for predicting missing drug-target interactions (Figure 4.1, Supplementary Tables A.1-A.2). The heterogeneous network integrates four types of nodes (i.e., drugs, proteins, diseases, and side-effects) and six types of edges (i.e., drug-protein interactions, drug-drug interactions, drug-disease associations, drug-side-effect associations, protein-disease associations and protein-protein interactions). Based on chemical structures of drugs and primary sequences of proteins, we also built up multiple similarity networks to further augment the network heterogeneity, providing our drug-target prediction task with diverse information and from a multiple-views perspective.

4.3.3 Compact feature learning for drugs and targets

DTINet applies diffusion component analysis (DCA) [189], an algorithm that combines network diffusion (i.e., random walk with restart) with dimensionality reduction, to learn the low-dimensional vector representations of the drug and target features that capture the intrinsic topological properties of a heterogeneous network. DCA has been generalized into Mashup, a new method for integrating multiple heterogeneous interactomes [190].

Random walk with restart revisited. Random walk with restart (RWR), a network diffusion algorithm, has been extensively applied to analyze the complex biological network data [1, 3, 4, 5, 185]. Different from conventional random walk methods, RWR introduces a pre-defined restart probability at the initial node for every iteration, which can take into consideration both local and global topological connectivity patterns within the network to fully exploit the underlying direct or indirect relations between nodes. Formally, let $\mathbf{A}$ denote the weighted adjacency matrix of a molecular interaction network with n drugs (or targets). We also define another matrix $\mathbf{B}$, in which each element $\mathbf{B}_{i,j}$ describes the probability of a transition from node i to node j, that is,

$$\mathbf{B}_{i,j} = \frac{\mathbf{A}_{i,j}}{\sum_{j'} \mathbf{A}_{i,j'}} \tag{4.1}$$

Next, let $\mathbf{s}_i^t$ be an n-dimensional distribution vector in which each element stores the probability of a node being visited from node i after t iterations in the random walk process. Then RWR from node i can be defined as:

$$\mathbf{s}_i^{t+1} = (1 - p_r)\mathbf{s}_i^t\mathbf{B} + p_r\mathbf{e}_i, \tag{4.2}$$

where $\mathbf{e}_i$ stands for an n-dimensional standard basis vector with $\mathbf{e}_i(i) = 1$ and $\mathbf{e}_i(j) = 0, \forall j \neq i$, and p_r stands for the pre-defined restart probability, which actually controls the relative influence between local and global topological information in the diffusion process, with higher values emphasizing more on the local structures in the network. At some fixed point of the iterating process, we can obtain a stationary distribution $\mathbf{s}_i^\infty$ of RWR, which we refer to as the "diffusion state" $\mathbf{s}_i$ for node i (i.e., $\mathbf{s}_i = \mathbf{s}_i^\infty$), being consistent with the notation of previous work [189]. Intuitively, the jth element of diffusion state, denoted by $\mathbf{s}_{ij}$, represents the probability of RWR starting node i and ending up at node j in equilibrium. When two nodes have similar diffusion states, it generally implies that they have similar positions with respect to other nodes in the network, and thus probably share similar functions.

The dimensionality reduction framework. The diffusion states resulting from the aforementioned RWR process may not be entirely accurate, partially due to the low quality and high dimensionality of biological data. A small number of missing or fake interactions in

the network can significantly affect the results of the diffusion process [196]. Moreover, it is generally inconvenient to directly use the high dimensionality of the diffusion states for the topological features, especially for our heterogeneous network based prediction task. To address this issue, DTINet employs a dimensionality reduction scheme, called diffusion component analysis (DCA), to reduce the dimensionality of the feature space and capture those important topological features from the diffusion states. With the goal of denoise and dimensionality reduction, DCA approximates each diffusion state $\mathbf{s}_i$ with a multinomial logistic model based on a latent vector representation whose dimensionality is much lower than that of the original n-dimensional vector representing the diffusion states. Specifically, the probability assigned to node j in the diffusion state of node i is now modeled as

$$\hat{\mathbf{s}}_{ij} = \frac{\exp\left(\mathbf{w}_i^T \mathbf{x}_j\right)}{\sum_{j'} \exp\left(\mathbf{w}_i^T \mathbf{x}_{j'}\right)} \tag{4.3}$$

where $\forall i,\ \mathbf{x}_i, \mathbf{w}_i \in \mathbb{R}^d$ for $d \ll n$. We refer to $\mathbf{w}_i$ as the *context feature* and $\mathbf{x}_i$ as the *node feature* for node i, both describing the topological properties of the network. If $\mathbf{x}_i$ and $\mathbf{w}_j$ point to a similar direction and thus have a large inner product, it is likely that node j is frequently visited in a random walk starting from node i. DCA takes a set of the observed diffusion states $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ as input and optimizes over $\mathbf{w}$ and $\mathbf{x}$ for all nodes, using the Kullback-Leibler (KL) divergence (also called relative entropy) to guide the optimization, that is,

$$\min_{\mathbf{w},\mathbf{x}} C(\mathbf{s}, \hat{\mathbf{s}}) = \frac{1}{n} \sum_{i=1}^{n} D_{KL}(\mathbf{s}_i \parallel \hat{\mathbf{s}}_i), \tag{4.4}$$

where $D_{KL}(\cdot \parallel \cdot)$ denotes the KL-divergence between two distributions. The DCA framework uses a standard quasi-Newton method L-BFGS [197] to solve this optimization problem.

Integration of heterogeneous network information. The above dimensionality reduction framework can be naturally extended to integrate multiple network data from heterogeneous sources. Given K similarity networks in a heterogeneous framework constructed from diverse information, DCA first performs RWR on individual networks separately and then obtains the network-specific diffusion states $\mathbf{s}_i^{(k)}$ for each node i in every network k. After that, it also constructs a multinomial logistic distribution to model the diffusion states:

$$\hat{\mathbf{s}}_{ij}^{(k)} = \frac{\exp\left(\mathbf{w}_i^{(k)T} \mathbf{x}_j\right)}{\sum_{j'} \exp\left(\mathbf{w}_i^{(k)T} \mathbf{x}_{j'}\right)}, \tag{4.5}$$

where each node i is assigned with a network-specific vector representation $\mathbf{w}_i^{(k)}$, which

represents the context feature of node i in network k, and the node feature vectors $\mathbf{x_i}$ are allowed to be shared globally across all K networks. Finally, DCA optimizes the following objective function,

$$\min_{\mathbf{w},\mathbf{x}} C(\mathbf{s}, \hat{\mathbf{s}}) = \sum_{k=1}^{K} \frac{1}{n} \sum_{i=1}^{n} D_{KL} \left(\mathbf{s}_i^{(k)} \parallel \hat{\mathbf{s}}_i^{(k)} \right), \tag{4.6}$$

which can also be solved by the quasi-Newton L-BFGS method [197]. Although the divergence terms of individual networks are given equal weights in the above objective function, it is possible to weight them differently to emphasize the relative importance of individual networks.

To make the DCA framework more scalable to large biological networks, DTINet employs a variant of DCA, called clusDCA [198], which uses an alternative objective function that can be optimized efficiently based on singular value decomposition (SVD). Briefly, for each drug or protein, clusDCA is able to learn a low-dimensional vector representation that corresponds to a solution minimizing the difference between the observed diffusion states and the model distribution under the L_2-norm in log space [198].

In our tests, we observed stable performance of DTINet for different values of the restart probability p_r between 0.5 and 0.8 (Figure B.24a). For all the test results shown in the Results section, the restart probability p_r was set to 0.8. After dimensionality reduction, we learned an f_d-dimensional vector for drugs and an f_t-dimensional vectors for targets. We observed robust results over a wide range of choices for the f_d and f_t parameters (Figure B.23). In the tests, we set $f_d = 100$ and $f_t = 400$, which were equal to 10-20% of the dimensionality of the original vectors describing the diffusion states.

4.3.4 The optimization process of DCA

For simplicity, here we only show the optimization process of DCA for a single input network. The optimization of DCA with multiple networks is a simple extension. To optimize the following objective function of DCA,

$$\min_{\mathbf{w},\mathbf{x}} C(\mathbf{s}, \hat{\mathbf{s}}) = \frac{1}{n} \sum_{i=1}^{n} D_{KL}(\mathbf{s}_i \parallel \hat{\mathbf{s}}_i). \tag{4.7}$$

We first express the formula in terms of $\mathbf{w}$ and $\mathbf{x}$ based on the definition of KL-divergence

and $\hat{\mathbf{s}}$, that is,

$$C(\mathbf{s}, \hat{\mathbf{s}}) = \frac{1}{n} \sum_{i=1}^{n} \left[H(\mathbf{s}_i) - \sum_{j=1}^{n} \mathbf{s}_{ij} \left(\mathbf{w}_i^T \mathbf{x}_j - \log \left(\sum_{j'=1}^{n} \exp\{\mathbf{w}_i^T \mathbf{x}_{j'}\} \right) \right) \right], \tag{4.8}$$

where $H(\cdot)$ denotes the entropy. Then we compute the gradients of this objective with respect to the parameters $\mathbf{w}$ and $\mathbf{x}$, respectively,

$$\nabla_{\mathbf{w}_i} C(\mathbf{s}, \hat{\mathbf{s}}) = \frac{1}{n} \sum_{j=1}^{n} (\hat{\mathbf{s}}_{ij} - \mathbf{s}_{ij}) \mathbf{x}_j, \tag{4.9}$$

$$\nabla_{\mathbf{x}_i} C(\mathbf{s}, \hat{\mathbf{s}}) = \frac{1}{n} \sum_{j=1}^{n} (\hat{\mathbf{s}}_{ji} - \mathbf{s}_{ji}) \mathbf{w}_j. \tag{4.10}$$

This objective function can be solved using a standard quasi-Newton L-BFGS method to find the low-dimensional vector representations $\mathbf{w}$ and $\mathbf{x}$. Throughout our tests, the vectors $\mathbf{w}$ and $\mathbf{x}$ were initialized with uniform random values in $[-0.05, 0.05]$.

To make the DCA framework more scalable to large biological networks, we use a more efficient matrix factorization based approach to decompose the diffusion states and obtain their low-dimensional vector representations. Based on the definition of $\hat{\mathbf{s}}_{ij}$, we have

$$\log \hat{\mathbf{s}}_{ij} = \mathbf{x}_i^T \mathbf{w}_j - \log \sum_{j'} \exp\{\mathbf{w}_i^T \mathbf{x}_{j'}\}. \tag{4.11}$$

The first term in the above equation corresponds to the low-dimensional approximation of $\hat{\mathbf{s}}_{ij}$, and the second term is a normalization factor, which ensures that $\hat{\mathbf{s}}_i$ is a well defined distribution. We relax the constraint that the entries in $\hat{\mathbf{s}}_i$ must sum to one by dropping the second term, that is

$$\log \hat{\mathbf{s}}_{ij} = \mathbf{x}_i^T \mathbf{w}_j. \tag{4.12}$$

In addition, instead of minimizing the relative entropy between the original and approximated diffusion states, we use the sum of squared errors as the objective function:

$$\min_{\mathbf{w},\mathbf{x}} C(\mathbf{s}, \hat{\mathbf{s}}) = \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{x}_i^T \mathbf{w}_j - \log \hat{\mathbf{s}}_{ij})^2. \tag{4.13}$$

This resulting objective function can be optimized by singular value decomposition (SVD). To avoid taking the logarithm of zero, we add a small positive constant $\frac{1}{n}$ to $\mathbf{s}_{ij}$ and compute

the logarithm diffusion state matrix $\mathbf{L}$ as:

$$\mathbf{L} = \log(\mathbf{S} + \mathbf{Q}) - \log(\mathbf{Q}), \tag{4.14}$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ with $\mathbf{Q}_{ij} = \frac{1}{n}$, $\forall i, j$, and $\mathbf{S} \in \mathbb{R}^{n \times n}$ is the concatenation of $\mathbf{s}_1, \ldots, \mathbf{s}_n$. With SVD, we decompose $\mathbf{L}$ into three matrices:

$$\mathbf{L} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{4.15}$$

To obtain the low-dimensional vectors $\mathbf{w}_j$ and $\mathbf{x}_i$ of d dimensions, we simply choose the first d singular vectors in $\mathbf{U}_d$, $\mathbf{V}_d$ and the first d singular values in $\mathbf{\Sigma}_d$. More precisely, let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ denote a matrix where each row represents the corresponding low-dimensional feature vector representation of each node in the network, and let $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_n]^T$ denote a matrix where each row represents the corresponding vector of the context features. Then, $\mathbf{X}$ and $\mathbf{W}$ can be computed as:

$$\mathbf{X} = \mathbf{U}_d \mathbf{\Sigma}_d^{1/2}, \quad \mathbf{W} = \mathbf{V}_d \mathbf{\Sigma}_d^{1/2}. \tag{4.16}$$

To integrate heterogeneous network data, we extend the above single-network DCA to a multiple-network case. More specifically, let $\mathbf{L} = \{\mathbf{L}^1, \ldots, \mathbf{L}^K\}$ be the set of logarithm diffusion state matrices based on the set of diffusion states $\mathbb{S} = \{\mathbf{S}^1, \ldots, \mathbf{S}^K\}$ from K input networks. Then, we optimize the following objective function:

$$\min_{\mathbf{w},\mathbf{x}} C(\mathbb{S}, \widehat{\mathbb{S}}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{r=1}^{K} (\mathbf{x}_i^T \mathbf{w}_j^r - \log \hat{\mathbf{s}}_{ij}^r)^2, \tag{4.17}$$

where we assign a network-specific feature $\mathbf{w}_i^r$ for each node i in network r, and the node features $\mathbf{x}_i$ are shared across all K networks. This objective function can also be optimized by SVD.

4.3.5 Projection from drug space onto target space

We use the low-dimensional vector representations of both drug and protein features obtained from compact feature learning to predict new drug-target interactions. Based on the intuition that geometric proximity in the feature vector space may reflect the functional relevance, we apply a matrix completion approach [199] to obtain a projection matrix that maps the low-dimensional feature vectors from drug space onto protein space, such that

the projected feature vectors of drugs are geometrically close to the vectors of their known interacting proteins.

Formally, we use $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{N_d}]^T$, $\mathbf{x}_i \in \mathbb{R}^{f_d}, i = 1, \ldots, N_d$ to denote the matrix representation of the drug features (i.e., each row i represents the corresponding feature vector of drug i), and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{N_t}]^T$, $\mathbf{y}_i \in \mathbb{R}^{f_t}, i = 1, \ldots, N_t$, to denote the matrix representation of the protein features (i.e., each row i represents the corresponding feature vector of protien i), where N_d and N_t stand for the numbers of drugs and proteins, respectively. Let $\mathbf{P}$ be a drug-target interaction matrix, where each entry $\mathbf{P}_{ij} = 1$ if drug i is known to interact with protein j, and $\mathbf{P}_{ij} = 0$ otherwise. We set up a bilinear function to learn the projection matrix $\mathbf{Z}$ between drug space and target space to predict the unknown drug-target interactions in $\mathbf{P}$ (i.e., those zero-valued entries). In particular, the bilinear function is formulated as:

$$\mathbf{X}\mathbf{Z}\mathbf{Y}^T \approx \mathbf{P}, \tag{4.18}$$

where $\mathbf{P} \in \mathbb{R}^{N_d \times N_t}$ stands for the drug-target interaction matrix, $\mathbf{X} \in \mathbb{R}^{N_d \times f_d}$, $\mathbf{Y} \in \mathbb{R}^{N_t \times f_t}$ are obtained from the compact feature learning stage (i.e., the network diffusion and dimensionality reduction processes), and $\mathbf{Z} \in \mathbb{R}^{f_d \times f_t}$ is the projection matrix to be learned. We then use the formula below to measure the likelihood of the pairwise interaction score between drug i and protein j:

$$\text{score}(i, j) = \mathbf{x}_i \mathbf{Z} \mathbf{y}_j^T, \tag{4.19}$$

where a larger score(i, j) suggests that drug i is more likely to interact with protein j.

Although the projection matrix $\mathbf{Z}$ is of dimension $f_d \times f_t$, there typically exist significant correlations between those feature vectors of drugs or proteins that are geometrically close in space, which can thus greatly reduce the number of effective parameters required to model drug-target interactions. To take into account this issue, we impose a low-rank constraint on $\mathbf{Z}$ to learn only a small number of latent factors, by considering a low-rank decomposition of the form $\mathbf{Z} = \mathbf{G}\mathbf{H}^T$, where $\mathbf{G} \in \mathbb{R}^{f_d \times f_k}$ and $\mathbf{H} \in \mathbb{R}^{f_t \times f_k}$. This low-rank constraint not only alleviates the overfitting problem but also computationally benefits the optimization process [200]. The optimization problem with such a low-rank constraint on the original projection matrix $\mathbf{Z}$ is NP-hard to solve. A standard relaxation of the low-rank constraint is to minimize the trace norm (i.e., sum of singular values) of the matrix $\mathbf{Z} = \mathbf{G}\mathbf{H}^T$, which is equivalent to minimize the Frobenius norms $\frac{1}{2}(\|\mathbf{G}\|_F^2 + \|\mathbf{H}\|_F^2)$. Therefore, factoring $\mathbf{Z}$ into $\mathbf{G}$ and $\mathbf{H}$ can be accomplished by solving the following optimization problem:

$$\min_{\mathbf{G},\mathbf{H}} \sum_{(i,j)} \|\mathbf{P}_{ij} - \mathbf{x}_i \mathbf{G}\mathbf{H}^T \mathbf{y}_j^T\|_2^2 + \frac{\lambda}{2}(\|\mathbf{G}\|_F^2 + \|\mathbf{H}\|_F^2), \tag{4.20}$$

where λ is a regularization parameter, which controls the tradeoff between the minimization of the squared loss on known interaction pairs (i, j) and the Frobenius norms. The optimization problem can be solved by alternating minimization [199]. We evaluated the prediction performance of DTINet with respect to different choices of the latent dimensionality parameter f_k and observed stable performance of DTINet over a wide range values of this parameter (Supplementary Figure B.24b). In our test, we set the value of the latent dimensionality parameter to $f_k = 50$, which was roughly equal to 50% of the dimension of the feature vectors of drugs, and 10% of the dimension of the feature vectors of proteins. The performance of DTINet was also robust to different choices of the regularization parameter λ. We observed that the performance was not influenced much when varying λ from 10^{-3} to 10^2 (Supplementary Figure B.24c). In our test, we did not fine-tune the value of λ and simply set $\lambda = 1$. In principle, we could always further improve the performance by carefully tuning these parameters using an inner-loop cross-validation on training data.

4.3.6 Time complexity of DTINet

When learning the low-dimensional representations of nodes in a heterogeneous network, performing the random walk with restart on each single network takes time complexity $O(n^3)$, where n is the number of nodes in the network. Next, the SVD operation on the matrix of diffusion states takes $O(Kn^3)$ time, where K stands for the total number of individual networks in the heterogeneous framework. Therefore, the running time for learning the compact representations of drugs and targets is $O(K_d N_d^3)$ and $O(K_t N_t^3)$, respectively, where K_d and K_t stand for the total numbers of similarity networks for drugs and targets, respectively, and N_d and N_t stand for the total numbers of drugs and targets, respectively. The matrix completion step for learning the projection matrix takes $O((q + N_d f_d + N_t f_t) f_k^2)$, where q stands for the number of non-zero entries in the known drug-target interaction matrix, f_d and f_t stand for the dimensions of the low-dimensional vector representations of drugs and targets, respectively, and f_k stands for the latent rank parameter of the matrix completion [199]. Thus, the overall time complexity of DTINet is $O(K_d N_d^3 + K_t N_t^3 + (q + N_d f_d + N_t f_t) f_k^2)$. In practice, K_d, K_t and f_k are usually small and can be regarded as constants.

4.3.7 Construction of similarity networks

For the input homogeneous interaction networks (e.g., drug-drug interaction network), we compute the "diffusion state" of each drug or target by directly running the RWR algorithm on each of these networks. For the association networks, i.e., drug-side-effect, drug-disease,

and protein-disease association networks, we construct the corresponding similarity networks based on the Jaccard similarity coefficient and then run the RWR process on these similarity networks. Jaccard similarity is a common statistic used to characterize the similarity between two sets of objects. Taking the drug-side-effect association network as an example, we use the following formula to measure the similarity between drug i and drug j:

$$\mathbf{S}(i,j) = \frac{|SE_i \cap SE_j|}{|SE_i \cup SE_j|}, \tag{4.21}$$

where SE_i denotes the set of side-effects of drug i. Then we run the RWR procedure on this similarity network to obtain the diffusion states of drugs. In the same manner, we can construct the similarity networks of proteins.

In addition to the above interaction or association-based similarity networks, we construct a drug similarity network based on the chemical structures of drugs, in which the similarity score between a pair of two drugs is calculated using the Tanimoto coefficient [201] using the product-graphs of their chemical structures. We also construct a protein similarity network based on genome sequences, in which the similarity score between a pair of two proteins is computed using the Smith-Waterman score [202] based on their primary sequences.

Overall, we construct four similarity networks for drugs, based on (i) drug-drug interactions, (ii) drug-disease associations, (iii) drug-side-effect associations, and (iv) chemical structures. Similarly, we construct three similarity networks for proteins, based on (i) protein-protein interactions, (ii) protein-disease associations, and (iii) genome sequences. With these similarity networks, we can learn the low-dimensional feature vector representations of drugs and proteins, by first performing diffusion separately on individual networks and then jointly optimizing the feature vectors under the compact feature learning framework.

4.3.8 Baseline Methods

We compare our method against four previously-proposed methods, including the bipartite local models (BLMNII), the Laplacian regularized least square (NetLapRLS), the heterogeneous network model (HNM) and the collaborative matrix factorization (CMF). We briefly describe these methods below.

1. Bipartite local model with neighbor-based interaction-profile inferring (BLMNII) [175]: This method is a combination of the bipartite local model (BLM) and the neighbor-based interaction-profile inferring (NII). The BLM framswork models the drug-target

interaction prediction task as a binary classification problem in a bipartite graph. Suppose that we want to predict whether drug d_i interacts with target t_j. The BLM method first focuses on drug d_i and assigns a label $+1$ to all the known targets that interact with drug d_i, and -1 otherwise. Then BLM uses the protein similarity matrix as a kernel matrix to train a support vector machine (SVM). Such a process is also performed in a reverse way, that is, BLM also labels each known drug by whether it interacts with target t_j or not, and then trains an SVM based on the drug similarity matrix. The final prediction of whether drug d_i interacts with target t_j is then derived based on the average prediction score from both directions. The NII procedure incorporates the neighbors interaction profiles into the BLM method to train the model and enable the prediction for new drugs or targets.

2. Laplacian regularized least square (NetLapRLS) [176]: This method employs a semi-supervised learning algorithm, i.e., Laplacian regularized least square, for DTI prediction, which utilizes available labeled data of DTI pairs and incorporates similarity and interaction kernels to improve the prediction. NetLapRLS attempts to estimate the interaction scores $\mathbf{F}_d$ and $\mathbf{F}_t$ based on the drug and protein domains, respectively. For example, the interaction scores $\mathbf{F}_d$ are obtained by minimizing the squared loss between the known DTI matrix $\mathbf{P}$ and $\mathbf{F}_d$ with a regularized term of $\mathbf{F}_d$ and $\mathbf{S}_d$, where $\mathbf{S}_d$ is the similarity network of drugs. The final prediction $\mathbf{F}$ is obtained by averaging the results derived from both $\mathbf{F}_d$ and $\mathbf{F}_t$.

3. Heterogeneous network model (HNM) [182]: This method builds a three-layer heterogeneous network consisting of three types of nodes: drug, target, and disease nodes. Then it iteratively propagates interaction or association information in the heterogeneous network using random walk with restart. The iterative updating rule is given by

$$\mathbf{W}_{td}^{k+1} = \alpha \mathbf{W}_{td}^{k} \times (\mathbf{W}_{dd} \times \mathbf{W}_{ds}^{k} \times \mathbf{W}_{ss} \times \mathbf{W}_{ds}^{kT}) + (1-\alpha)\mathbf{W}_{td}^{0}, \tag{4.22}$$

$$\mathbf{W}_{ds}^{k+1} = \alpha (\mathbf{W}_{td}^{kT} \times \mathbf{W}_{tt} \times \mathbf{W}_{td}^{k} \times \mathbf{W}_{dd}) \times \mathbf{W}_{ds}^{k} + (1-\alpha)\mathbf{W}_{ds}^{0}, \tag{4.23}$$

where $\mathbf{W}_{td}^{k}$ and $\mathbf{W}_{ds}^{k}$ stand for the weights on the target-drug and drug-disease association links in the kth iteration, respectively; $\mathbf{W}_{td}^{0}$ and $\mathbf{W}_{ds}^{0}$ represent the target-drug and drug-disease association matrices defined by the input data; $\mathbf{W}_{dd}$ stores both drug interaction and similarity information, which is basically computed from the averaging result derived from both of the drug-drug interaction and drug-drug similarity matrices; $\mathbf{W}_{ss}$ represents the disease-disease similarity matrix; and $\mathbf{W}_{tt}$ represents the

protein-protein interaction matrix derived from the input data. The final DTI prediction scores are obtained from matrix $\mathbf{W}_{td}$ after convergence.

4. Collaborative Matrix factorization (CMF) [187]: This method learns the feature vector matrices $\mathbf{X}$ and $\mathbf{Y}$ for drugs and targets, respectively, by minimize the following objective function:

$$\min_{\mathbf{X},\mathbf{Y}} \|\mathbf{P} - \mathbf{X}\mathbf{Y}^T\|_F^2 + \lambda_m(\|\mathbf{X}\|_F^2 + \|\mathbf{Y}\|_F^2) + \lambda_d\|\mathbf{S}_d - \mathbf{X}\mathbf{X}^T\|_F^2 + \lambda_t\|\mathbf{S}_t - \mathbf{Y}\mathbf{Y}^T\|_F^2, \quad (4.24)$$

where S_d and S_t represent the drug and target similarity matrices, respectively, and λ_m, λ_d and λ_t represent the regularization coefficients.

Here, HNM and CMF are designed to integrate heterogeneous information, while BLMNII and NetLapRLS mainly focus on solving the DTI prediction problem on a single network. To make a fair comparison, we first summarized our heterogeneous network into a single network for both BLMNII and NetLapRLS, using the following integration process. In particular, we combined multiple networks into a single network by assigning the edge weight $p_{i,j} = 1 - \prod_k (1 - p_{i,j}^{(k)})$, where $p_{i,j}^{(k)} \in [0, 1]$ is the interaction probability or similarity between node i and node j in network $k \in \{1, 2, \ldots, K\}$, where K stands for the total number of networks.

4.3.9 Computational docking analyses

In our structure-based modeling studies, we used the docking program Autodock [172] to infer the possible binding modes of the new predicted interactions between three drugs (i.e., telmisartan, chlorpropamide and alendronate) and the COX proteins. The protein structures used in our docking studies were downloaded from the Protein Data Bank [203] (PDB IDs 3kk6 and 3qmo for COX-1 and COX-2, respectively). The three-dimensional structures of the above three drugs were obtained from the ZINC database [204].

4.3.10 Experimental validation

Reagents. LPS (L-2360) and 4% sterile thioglycollate were purchased from Sigma-Aldrich (St. Louis, MO, USA). IFN-γ(315-05) was purchased from PeproTech (New York, USA). Chlorpropamide (S4166), telmisartan (S1738), alendronate (S1624) and ibuprofen (S1638) were purchased from Selleck Chemicals (Houston, TX, USA). COX Fluorescent Activity Assay Kit (700200), arachidonic acid (90010), indomethacin (70270), human recombinant

COX-1 and COX-2 enzymes (17616 and 60122) and prostaglandin E2 (PGE 2) ELISA kit (514010) were purchased from Cayman Chemical Company (Ann Arbor, MI, USA). Cyclooxygenase 2 (ab62331) antibody was obtained from Abcam (Cambridge, MA, USA). [^{3}H] celecoxib was purchased from Hartmann Analytics (Braunschweig, Germany).

Animals. C57BL/6J mice (10 weeks old) were obtained from Vital River (Beijing, China) and were housed under controlled temperature (22 °C ± 2 °C) and humidity (40-60%) with a 12 h light/dark cycle. For each experiment, three mice were randomly injected intraperitoneally with 1 ml of 4% sterile thioglycollate and sacrificed 3 days later. All animal surgery was performed under anesthesia by Avertin (250 mg/Kg), and anesthetized animals were sacrificed by cervical dislocation at the end of the experiments. All experiments were performed in accordance with guidelines of the Institute for Laboratory Animal Research of Tsinghua University. The experimental procedures were approved by the Administrative Committee of Experimental Animal Care and Use of Tsinghua University, licensed by the Science and Technology Commission of Beijing Municipality (SYXK-2014-0024), and they conformed to the National Institute of Health guidelines on the ethical use of animals.

Cell culture. Peritoneal macrophages were isolated from peritoneum by lavage using 20 ml DMEM and seeded into 6 well plates using one hundred million cells/well in DMEM of 10% FBS. Non-adherent cells were removed 6 h later, whereas adherent cells were refed with DMEM of 10% FBS and allowed to recover overnight. Macrophages were treated with chlorpropamide, telmisartan and alendronate for 24h and then pre-incubated with DMEM-10% FBS for 2h before treatment of LPS (10 ng/ml). Macrophages were pre-treated with DuP-697 and SC-560 for 12 h before treatment with telmisartan, alendronate and chlorpropamide and then incubated with IFN-γ (10 ng/ml) for 12 h following LPS stimulation (10 ng/ml) for 6 h. The concentrations of telmisartan, alendronate and chlorpropamide treatment were determined based on previous research [205, 206, 207], while those of the chemical probe Dub-697 and the known NSAID ibuprofen were determined according to the indications of the assay kit and previous binding studies in the literature [208, 209, 210], respectively. Cells were harvested for subsequent analysis.

COX fluorescent activity. Following stimulation, kidneys were harvested from mice, and macrophages from the above treatment were homogenized in 5 ml of cold PBS containing protease inhibitors and centrifuged at 10,000 g for 15 minutes at 4 °C. The supernatant was assayed by the COX fluorescent activity assay kit according to the manufacturer's instructions.

Human recombinant enzyme assays. The selectivity of inhibition *in vitro* for telmisartan, alendronate and chlorpropamide was evaluated using the recombinant human COX-1 and COX-2 enzyme assays as previously described in [211]. In particular, the recombinant en-

zymes were pre-incubated with various concentrations of telmisartan, alendronate and chlorpropamide for 10 minutes at 25 °C. Then the 10 μM arachidonic acid was added to start the reaction and allowed the process to proceed for 10 minutes. The reaction was terminated by diluting the reaction into buffer containing 25 μM indomethacin. The final levels of PGE 2 were measured by ELISA.

Radioligand-based binding assays. The binding assays of Hood et al. [212] were used to assess the direct binding activity to COX-2 by measuring the competitive binding of the radiolabeled inhibitor [^{3}H] celecoxib to the target enzyme. The murine monoclonal COX-2-specific antibody was coated onto 96-well Immulon 2HB microtiter plates (Thermo Scientific, Waltham, USA) and incubated overnight at 37 °C. The coated plates were washed with Dulbecco's phosphate-buffered saline (D-PBS) and blocked by 10% skim milk to avoid non-specific binding. The recombinant COX-2 enzyme binding buffer was added to plates and incubated for 2 h at 37 °C and these antibody-captured enzyme-coated plates were washed with D-PBS. To measure the competitive binding activity with celecoxib, compounds at their IC50 concentrations were incubated with [^{3}H] celecoxib and allowed to compete for the binding to COX-2 for 2 h. After that, the incubation was halted by aspiration and washed twice with cold D-PBS. The 50 μl of 10% SDS was added into plates for 1 hour at 37 °C. At last, the COX-2 bound radioligand was transferred into the liquid scintillation vial for quantitation using the liquid scintillation spectrometry.

Real-time PCR analysis. Total RNA was extracted from the whole-cell lysates using the Trnzol-A$^+$ reagent (Tiangen, Cat. no. DP421, China). Reverse transcription was performed using TIANScript RT Kit (Tiangen, Cat. no. KR104-02, China). All real-time PCR reactions were carried out on ABI ViiA$^{\text{TM}}$ 7 Real-Time System (Life Technologies, USA) using TransStart Top Green qPCR SuperMix (Transgen, Cat. no. AQ131-03, China). The formula $2^{-\Delta\Delta\text{Ct}}$ was used to calculate the relative expression. The expression of the housekeeping gene GAPDH was used as an internal control.

Statistical analysis. Statistical analyses were performed using GraphPad Prism software (Version 6.0). Values were presented as mean ± SD. Every analysis was performed for three independent experiments, each of which was performed with triplicates. Data were analyzed using a one-way analysis of variance (ANOVA), followed by a Newman–Keuls multiple comparison test. Statistical significances were calculated and indicated. *: $P < 0.05$, **: $P < 0.01$, ***: $P < 0.001$.

4.3.11 Code and data availability

The source code of DTINet and the input heterogeneous network data can be downloaded from `https://github.com/luoyunan/DTINet`.

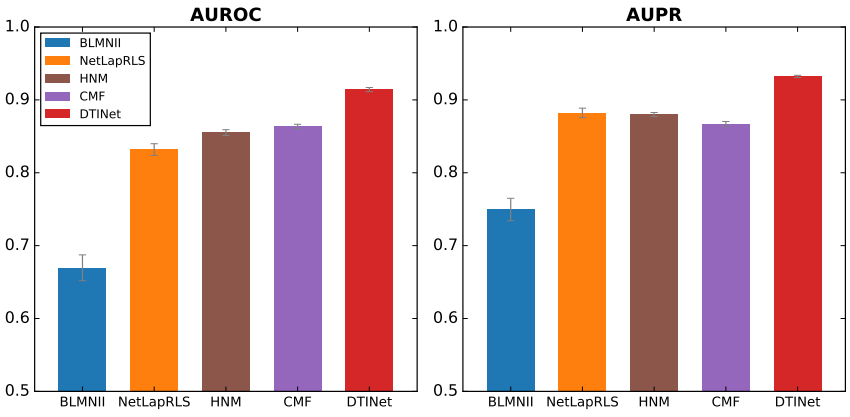
4.4 RESULTS

4.4.1 DTINet yields accurate drug-target interaction prediction

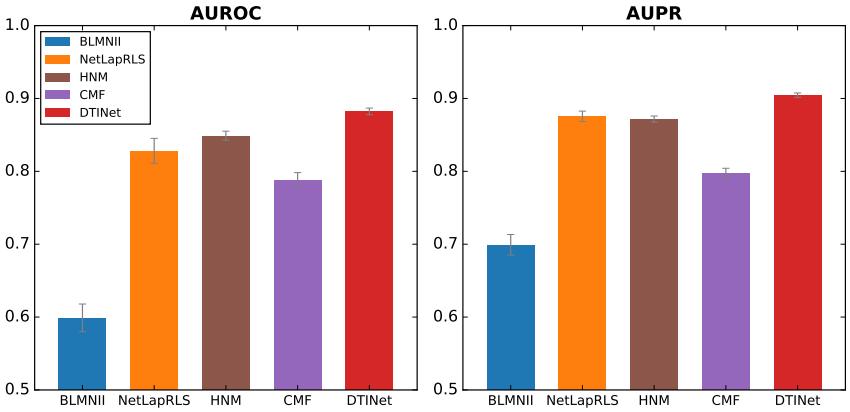
We first evaluated the prediction performance of DTINet using a ten-fold cross-validation procedure, in which a randomly chosen subset of 10% of the known interacting drug-target pairs and a matching number of randomly sampled non-interacting pairs were held out as the test set, and the remaining 90% known interactions and a matching number of randomly sampled non-interacting pairs were used to train the model. We compared DTINet with four state-of-the-art methods for DTI prediction, including BLMNII [175], NetLapRLS [176], HNM [182] and CMF [187]. Our comparative results showed that DTINet consistently outperformed other existing methods, with 5.9% higher AUROC and 5.7% higher AUPR than the second-best method (Figure 4.3a). Compared to HNM, which predicts DTIs based on a modified version of random walk in a complete heterogeneous network, DTINet achieved 6.9% higher AUROC (85.51% for HNM) and 5.9% higher AUPR (87.98% for HNM), presumably because HNM only uses the original diffusion states for prediction, which is not entirely accurate, while DTINet applies a novel dimensionality reduction on the diffusion states and thus is able to capture the underlying structural properties of the heterogeneous network.

To mimic a practical situation in which a drug-target interaction matrix is often sparsely labeled with only a few known DTIs, we also performed two additional cross-validation tests, in which the negative set in the test data either contained negative samples nine times more than the positive ones or all remaining non-interacting drug-target pairs that were not in the training data (Supplementary Figure B.17). In these two settings with imbalanced datasets, the known drug-target interactions (i.e., positive samples) composed only 10% and 0.18% of the whole dataset, respectively. In these two tests, although the AUPR scores of all methods dropped when compared to the previous test (Figure 4.3a), we observed that DTINet still achieved much higher AUPR than other methods, e.g., about 100% higher than the second-best method when considering all non-interacting drug-target pairs as the negative set; Supplementary Figure B.17). As studied in previous works [121, 177, 213], AUROC is likely to be an overoptimistic metric to evaluate the performance of a prediction algorithm, especially on highly-skewed data, while AUPR can provide a better assessment in this scenario. Thus, the noticeable performance improvement of DTINet in terms of AUPR over other prediction methods demonstrated its superior ability in predicting new DTIs in sparsely labeled networks.

The originally collected datasets may contain homologous proteins or similar drugs, which



(a)



(b)

Figure 4.3: DTINet outperforms other state-of-the-art methods for DTI prediction. We performed a ten-fold cross-validation procedure to compare the prediction performance of DTINet to that of four state-of-the-art DTI prediction methods, i.e., HNM, CMF, and the extended versions of BLMNII and NetLapRLS. Performance of each method was assessed by both the area under ROC curve (AUROC) and the area under precision-recall curve (AUPRC). (a) All methods were trained and tested on the original collected dataset (see the main text), without removing any homologous protein. (b) All methods are trained and tested on a modified dataset, in which homologous proteins were excluded. A pair of two proteins are said to be homologous if their sequence identity score is above 40%. All results were summarized over 10 trials and expressed as mean ± SD.

raised a potential concern that the good performance of prediction methods might result from easy predictions. To investigate this issue, we performed the following additional tests (Figure 4.3b and Supplementary Figure B.18): (1) the removal of DTIs involving homologous proteins (sequence identity scores $> 40\%$); (2) the removal of the DTIs with similar drugs (Tanimoto coefficients $> 60\%$); (3) the removal of the DTIs with the drugs sharing similar side-effects (Jaccard similarity scores $> 60\%$); (4) the removal of the DTIs

with the drugs or proteins associated with similar diseases (Jaccard similarity scores $> 60\%$); and (5) the removal of the DTIs with either similar drugs (Tanimoto coefficients $> 60\%$) or homologous proteins (sequence identity scores $> 40\%$). In the above tests, the removal operations can further reduce the potential redundancy in the DTIs that may cause the inflated evaluation performance in cross-validation. The test results under the above settings showed that DTINet was robust against the removal of homologous proteins or similar drugs in the training data and still consistently outperformed other methods (Figure 4.3b and Supplementary Figure B.18). We also removed the DTIs with homologous proteins in a skewed dataset in which the known interacting drug-target pairs composed only 10% of the whole dataset, and observed similar results (Supplementary Figure B.18e). Other threshold values for drug similarity scores and protein identity scores were also evaluated, and similar trends were observed (results not shown). Taken together, these results demonstrated that DTINet can still achieve decent performance and outperform other prediction methods even without the presence of similar drugs or targets.

The random split of training and test data in the conventional cross-validation procedure may raise another concern due to "popular" drugs [199]. Since the drugs that are well-connected (i.e., with large degrees) to proteins in the drug-target interaction network tend to be predicted more easily and thus may result in the inflated high recall rates, it is important to seek a proper evaluation procedure and metric to assess the performance of prediction methods under more realistic drug repositioning scenarios. To this end, we first hid all DTIs in which the related drugs have new MOAs (mechanism of actions) discovered within five years as of the time that the DrugBank database Version 3.0 (which was used to construct our heterogeneous network) was released. According to this criterion, we held out 255 DTIs related to 79 drugs as the test set. As in the previous studies [199, 214], we used "recall @ top-k" as the evaluation metric, which is defined as the fraction of true interacting targets that were retrieved in the list of top-k predictions for a drug. The motivation of using this metric was that a method that can accurately recover the true interacting targets in the list of top-k predictions is generally desired and useful for the downstream experimental validation. We found that DTINet achieved much better performance than other methods in recovering the true interacting targets for a given drug at different values of rank k (Supplementary Figure B.19a). For example, DTINet recovered 59% of the true DTIs within the list of top 150 predictions (corresponding to $\sim 10\%$ of the total 1,512 proteins), which was about 34% higher than the second-best method. In the second setting, we evaluated the prediction performance of different methods on those singleton drugs, which have only one interacting known target in our dataset. Such a setting can be considered as a difficult case in computational drug repositioning, in that all these singleton drugs have

no known interacting targets available in the training data (corresponding to those rows with no known entries in the drug-target interaction matrix). Thus, there was no redundancy resulting from homologous proteins or similar drugs between training and test data during the cross-validation procedure. This setting can be used to assess the performance of prediction methods on those DTIs that are relatively less well studied and characterized. We observed that DTINet retrieved $\sim 50\%$ of true DTIs for a singleton drug in the list of top-150 predictions, in contrast to a fraction less than 28% for other methods (Supplementary Figure B.19b). Overall, the test results under the above two settings demonstrated the superior ability of DTINet in integrating heterogeneous information into the prediction of new DTIs in real drug repositioning scenarios, which implied that DTINet can provide a practically useful tool for computational drug repositioning and drug discovery.

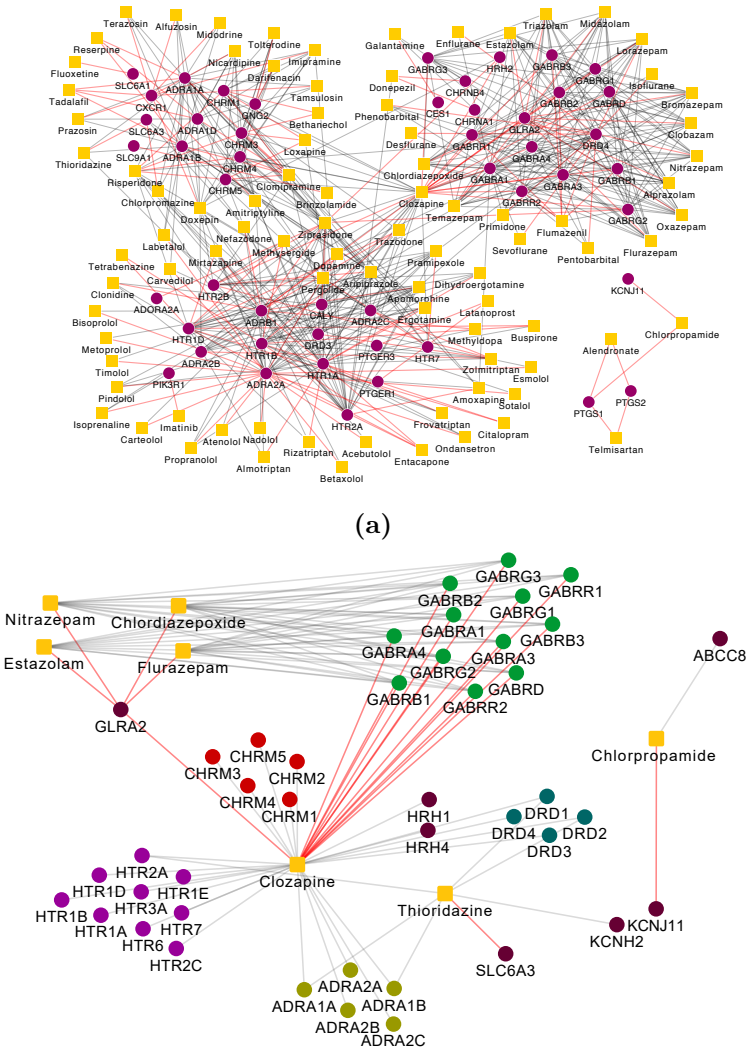
DTINet copes with the noise and incompleteness in the high-dimensional data by learning the compact representations that capture the most explanatory features. To directly evaluate the robustness of DTINet under this setting, we randomly perturbed the topological structures in the network data. In particular, 10% randomly sampled edges in the heterogeneous network were perturbed, by adding new edges or deleting existing interaction (or association) edges. Compared to NetLapRLS, DTINet achieved more robust performance against the incompleteness or noise in the network data (Supplementary Figure B.20). This result demonstrated the robustness of DTINet in extracting the relevant latent topological patterns even under the setting of noisy network data.

Our further comparative study showed that integrating multiple networks derived from the feature vectors of drugs or proteins by DTINet can greatly improve the prediction performance over individual single networks (Supplementary Figure B.21). Our comparison demonstrated that, even without multiple networks integration, DITNet still outperformed the state-of-the-art single network-based method NetLapRLS on individual similarity networks. This result emphasized DTINet's ability to fully exploit useful topological information from high-dimensional and noisy network data via a compact learning procedure, even only given a single network as input. In addition, we observed that DTINet achieved much better prediction performance than the extended version of NetLapRLS, when integrating multiple networks into a heterogeneous one. These results indicated that integrating multiple networks into DTI prediction is not a trivial task, while the network integration procedure of DTINet can simultaneously and effectively capture the underlying topological structures of multiple networks, leading to the improved accuracy of DTI prediction. Moreover, in terms of time complexity, DTINet runs fast and only takes roughly cubic time.

4.4.2 DTINet identifies novel drug-target interactions

We also predicted the novel drug-target interactions using the whole heterogeneous network (in which drug and targets have at least one known interacting pair) as training data and outputted the list of top predictions (Supplementary Data 1 in Luo et al. [36]). We excluded those easy predictions in which the targets have sequence identity scores above 40% from the homologous proteins in training data. Among the list of top 150 predictions (Figure 4.4a and Supplementary Data 1 in Luo et al. [36]), we found that many of them can also be supported by the previous known experimental or clinical evidence in the literature (Figure 4.4b and Supplementary Data 3 in Luo et al. [36]). For example, new predictions showed that clozapine can act on the gamma-aminobutyric acid (GABA) receptors, an essential family of channel proteins that modulate cognitive functions (Figure 4.4b). This new prediction can be supported by the previous studies which showed that clozapine can have a direct interaction with the GABA B-subtype (GABA-B) receptors [215] and antagonize the GABA A-subtype (GABA-A) receptors in the cortex [216]. More examples of such novel predictions which can be supported from the previous studies in the literature can be found in Supplementary Data 3 of Luo et al. [36].

Next, we focused on those novel drug-target interactions among the list of top 150 predictions from DTINet, for which we rarely found known experimental support in the literature. Among the list of these top 150 predictions, most of the new predicted DTIs were relevant (i.e., connected) to the previously known interactions except the interactions between three drugs, including telmisartan, chlorpropamide and alendronate, and the prostaglandin-endoperoxide synthase (PTGS) proteins, which are also called cyclooxygenase (COX) proteins (Figure 4.4a). COX is a family of enzymes responsible for prostaglandin biosynthesis [217], and mainly includes COX-1 and COX-2 in human, both of which can be inhibited by nonsteroidal anti-inflammatory drugs (NSAIDs) [218]. Apparently, it was difficult to use the correlations between nodes within the DTI network to explain the predicted interactions between these three drugs and the COX proteins. On the other hand, these new DTIs had relatively high prediction scores in the list of the top 150 predictions (Supplementary Data 1 in Luo et al. [36]). In addition, the COX proteins provide a class of important targets in a wide range of inflammatory diseases [219]. Despite the existence of numerous known NSAIDs used as COX inhibitors, many of them are associated with the cardiovascular side-effects [220, 221]. Thus, it is always important to identify alternative COX inhibitors from existing drugs with less side-effects. Given these facts, it would be interesting to see whether the predicted interactions between these three drugs and the COX proteins can be further validated.



(a)

(b)

Figure 4.4: Network visualization of the drug-target interactions predicted by DTINet. (a) Visualization of the overall drug-target interaction network involving the top 150 predictions (Supplementary Data 1 in Luo et al. [36]). Target and drugs are shown in purple circles and yellow boxes, respectively. (b) Network visualization of several examples of novel DTI predictions which can be supported by known experimental or clinical evidence in the literature. The drugs are shown in yellow boxes, white different families of their interacting targets are shown in circles with different colors. In both (a) and (b), known drug-target interactions are marked by grey edges, while the new predicted interactions are shown by red edges.

Among the aforementioned three drugs, telmisartan has been known as an angiotensin II receptor antagonist that can be used to treat hypertension [222], chlorpropamide has been known as a sulfonylurea drug that acts by increasing insulin to treat type 2 diabetes mellitus [223], and alendronate has been known as a bisphosphonate drug mainly used for treating bone disease, such as osteoporosis and osteogenesis imperfect [224, 225]. Despite

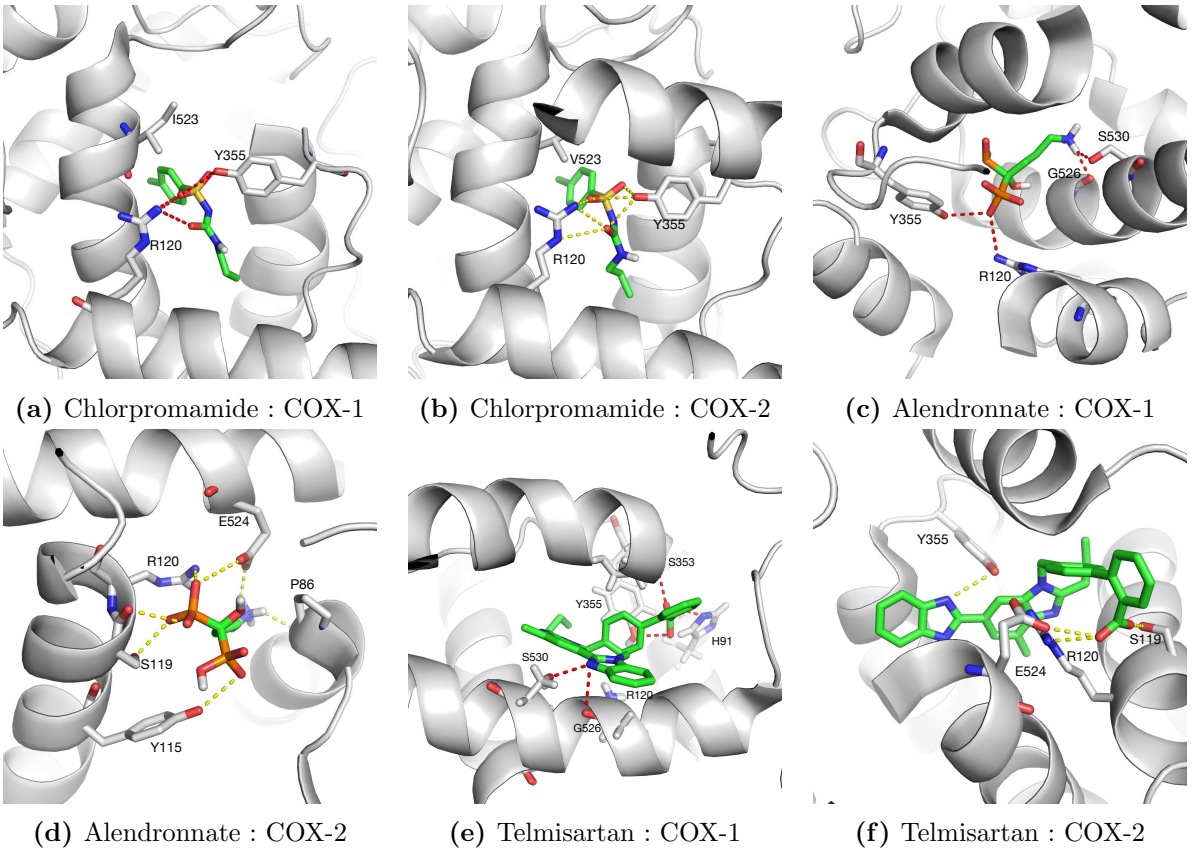
our current understanding about the functions of COX-1 and COX-2 proteins and the known indications of telmisartan, chlorpropamide and alendronate, it still remains largely unknown whether these three drugs can also interact with the COX proteins. According to the top 150 predictions by DTINet (Figure 4.4a and Supplementary Data 1 in Luo et al. [36]), these three drugs can act on the COX proteins. We will further present our validation results on the predicted interactions between these three drugs and COX proteins in the next sections.

4.4.3 Computational docking suggests the binding modes for the predicted drug-target interactions

Our docking studies ("Methods") showed that the three drugs (i.e., telmisartan, alendronate and chlorpropamide) were able to dock to the structures of both COX-1 (PDB ID: 3kk6) and COX-2 (PDB ID: 3qmo), and displayed different binding patterns (Figure 4.5). In particular, all three drugs were fitted into the active sites of both COX-1 and COX-2. More specifically, chlorpropamide displayed similar configurations when binding to COX-1 and COX-2 (Figures 4.5a and 4.5b), by forming hydrogen bonds with both residues R120 and Y355, which created a conserved pocket as in those for common NSAIDs [226, 227]. On the other hand, the substitution of V119 in COX-1 by S119 in COX-2 allowed the formation of a different hydrogen bond network in the binding pocket. Moreover, telmisartan and alendronate interacted with residue S530 in addition to residues R120 and Y355 when docked to COX-1 (Figures 4.5c and 4.5e), while they were both able to bind to residue S119 when docked to COX-2 (Figures 4.5d and 4.5f). Thus, a subtle difference between the binding pockets of those two enzymes may result in different binding modes even for the same drug. These docking results may provide important hints for understanding the structural basis of the predicted drug-target interactions and thus help reveal the underlying molecular mechanisms of drug action.

4.4.4 Experimental validation of the top-ranked drug-target interactions predicted by DTINet

We further sought to experimentally validate the bioactivities of the COX inhibitors predicted by DTINet ("Methods"). First, we tested their inhibitory potencies on the mouse kidney lysates using the COX fluorescent activity assays. Similar dose-dependent repression of COX activity was observed for the three drugs (Figures 4.6a-4.6c). The IC50 values of telmisartan, alendronate and chlorpropamide for COX activity were measured at 56.14 μM, 160.2 μM and 289.5 μM, respectively. The measured IC50 values of the three drugs especially



(a) Chlorpromamide : COX-1 **(b)** Chlorpromamide : COX-2 **(c)** Alendronnate : COX-1

(d) Alendronnate : COX-2 **(e)** Telmisartan : COX-1 **(f)** Telmisartan : COX-2

Figure 4.5: The docked poses for the predicted interactions between three drugs (chlorpropamide, alendronate and telmisartan) and the COX proteins (COX-1 and COX-2). (a) Chlorpromamide vs. COX-1; (b) Chlorpromamide vs. COX-2; (c) Alendronnate vs. COX-1; (d) Alendronnate vs. COX-2; (e) Telmisartan vs. COX-1; (f) Telmisartan vs. COX-2. The protein structures of COX-1 and COX-2 were downloaded from the Protein Data Bank (PDB IDs 3kk6 and 3qmo for COX-1 and COX-2, respectively). The structures of the small molecules were obtained from the ZINC [204]. The clocking program Autodock [172] was used for the docking modeling. Hydrogen bonds were computed by PyMOL [228] and represented by the red and yellow dashed lines in COX-1 and COX-2, respectively.

telmisartan were comparable to those of many common NSAIDs, such as celecoxib (COX-1: 82 μM; COX-2: 6.8 μM), ibuprofen (COX-1: 12 μM, COX-2: 80 μM) and rofecoxib (COX-1: >100 μM; COX-2: 25 μM) [229, 230]. Probably alendronate and chlorpropamide were relatively weak inhibitors of COX. It is worth noting that the order of the experimentally measured IC50 values of these three drugs was consistent with the ranking of prediction scores in DTINet (Supplementary Data 1 in Luo et al. [36]).

Next, the tissue extracts from the mouse kidney and the peritoneal macrophages were used for COX selective inhibition assays. Relative inhibition of COX-1 and COX-2 activities was distinguished using SC-560, a potent and selective COX-1 inhibitor, and Dup-697, a po-

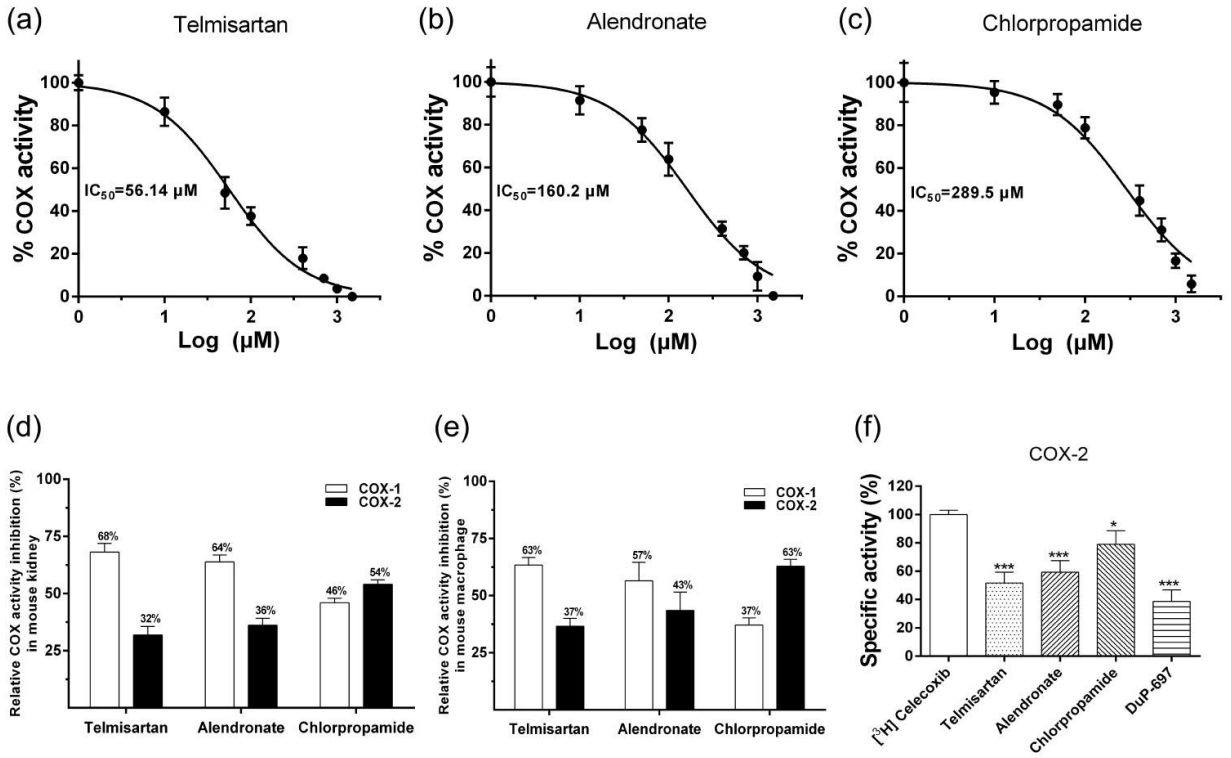


Figure 4.6: Inhibitory effects of telmisartan, alendronate and chlorpropamide on COX activity measured by COX inhibition assays. (a)-(c) The inhibition rates of telmisartan, alendronate and chlorpropamide measured by the COX fluorescent activity assays on the mouse kidney lysates. (d) and (e) The relative COX activity inhibition rates of telmisartan, chlorpropamide and alendronate on COX-1 and COX-2, measured by the COX fluorescent activity assays on the tissue extracts from both kidney (d) and macrophage (e) lysates. (f) The results on the competitive binding of [^{3}H] celecoxib for different COX-2 inhibitors measured by the radioligand-based binding assays. Control: the radioactivity of the sample with [^{3}H] celecoxib only. *: $P < 0.05$, **: $P < 0.01$, ***: $P < 0.001$, , Newman-Keuls multiple comparison test. Here, data show the mean with the standard deviation of three independent experiments, each of which was performed with triplicates.

tent and time-dependent of COX-2 inhibitor, respectively. Overall, the assays on the tissue extracts from mouse kidney showed that telmisartan and alendronate had slightly higher inhibition rates on COX-1 (68% and 64%, respectively) than COX-2 (32% and 36%, respectively), while chlorpropamide had a slightly higher inhibition rate on COX-2 (54%) than on COX-1 (46%) (Figure 4.6d). Similar patterns of COX inhibition selectivity with these drugs were also observed in the peritoneal macrophages (Figure 4.6e). To further evaluate the selectivity of these predicted drugs on COX-1 and COX-2, we also used the human recombinant enzyme assays to measure the levels of PGE 2 under COX-1 and COX-2 catalyses, respectively. The assay results showed that telmisartan, alendronate and chlorpropamide

had IC50 values of 41.97 μM, 90.73 μM and 223.5 μM for COX-1, respectively, and 91.75 μM, 184.1 μM and 151.9 μM for COX-2, respectively (Supplementary Figures B.22). Such results were also consistent with the IC50 values measured by the previous selective inhibition assays (Figure 4.6d-4.6e). These validation results were also in line with the observation that the predicted scores of these novel DTIs output by DTINet were actually not that far away (Supplementary Data 1 in Luo et al. [36]). Overall, the above inhibition assays showed that these three drugs identified by DTINet had a certain level of inhibition affinity and and may act as non-selective COX inhibitors on the family of COX proteins.

To further validate the predicted drug-target interactions, we also applied the radioactive isotope labeled with the COX-2 selective inhibitor [^{3}H] celecoxib for the competitive binding assays. As shown in Figure 4.6f, telmisartan, alendronate, chlorpropamide and DuP-697 (a standard COX-2 inhibitor) inhibited the binding of [^{3}H] celecoxib to COX-2 by about 48.33%, 40.67%, 14.00% and 61.33% at their IC50 concentrations, respectively. These assay results provided another piece of evidence to confirm that these drugs may have direct interactions with the COX proteins.

The COX inhibitors have been extensively used as nonsteroidal anti-inflammatory drugs (NSAIDs), thus we further tested the effects of the above three drugs on inflammatory responses and thus examined their potential applications in treating inflammatory diseases. Lipopolysaccharide (LPS) was used to stimulate the cultured peritoneal macrophages for the cellular inflammation model. In addition to those three drugs (i.e., telmisartan, chlorpropamide and alendronate) predicted by DTINet, we also considered the potent COX-2 inhibitor Dup-697 and the well-known NSAID ibuprofen for comparison.

A large number of proinflammatory factors can be generated during the inflammation process [231]. We consequently tested whether the three drugs can suppress the expression of various inflammatory factors in response to LPS stimulation (Figure 4.7). For TNF-α and IL-6, telmisartan exhibited a strong inhibitory effect on the LPS induced expression (Figures 4.7a and 4.7b). Meanwhile, the induction of the important cytokine IL-1β was also attenuated by each of the three drugs in the peritoneal macrophages (Figure 4.7c). In particular, telmisartan displayed the strongest suppression effect on IL-1β among all COX inhibitors. For IL-12p35, although both alendronate and telmisartan significantly inhibited its production induced by LPS, telmisartan had a much stronger suppression effect than other COX inhibitors (Figure 4.7d). The LPS-induced production of the immunological defensive factors such as CXCL-1 and iNOS were significantly restrained by the treatment of any of these three drugs (Figures 4.7e and 4.7f), which was similar to the results of both Dup-697 and ibuprofen. In summary, these results showed that telmisartan, chlorpropamide and alendronate can reduce the expressions proinflammatory factors in mouse peritoneal

macrophages. The observed anti-inflammation effects of these three drugs further extended the above inhibition assay studies and demonstrated their potential applications in preventing inflammatory disease.

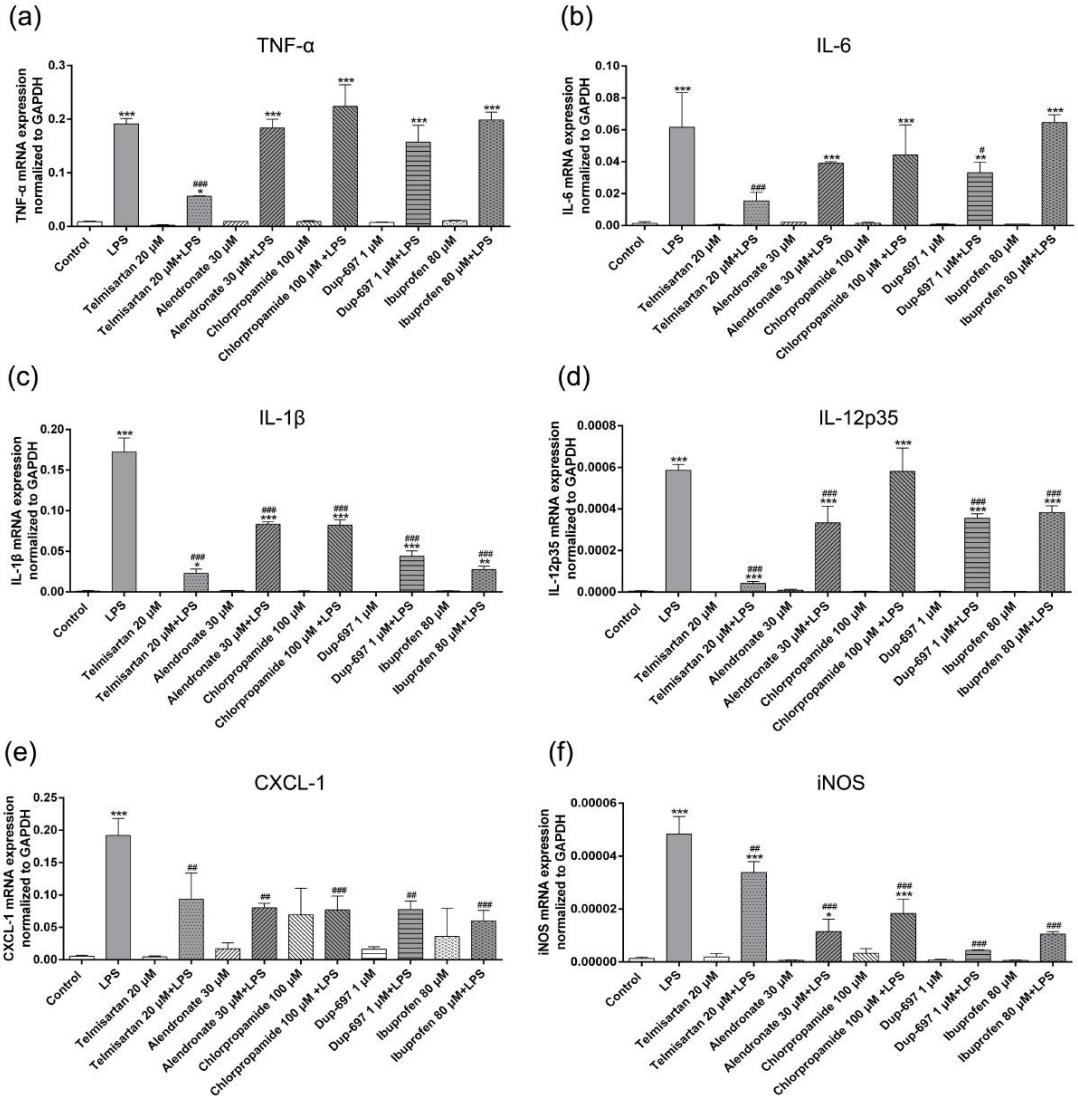


Figure 4.7: The real-time PCR (RT-PCR) analyses of the proinflammatory factors on the LPS-stimulated macrophages. (a)-(f) The RT- PCR analysis of mRNA expressions of TNFα, IL-6, IL-1β, IL-12p35, CXCL-1 and iNOS normalized relative to that of GAPDH, respectively. Control, macrophages without LPS treatment. *: $P < 0.05$; **: $P < 0.01$; $***: P < 0.001$, compared to the samples without LPS treatment. ##: $P < 0.01$; ### : $P < 0.001$, compared to the samples treated with LPS. n=3. Newman-Keuls multiple comparison test was used. Here, data show the mean with the standard deviation of three independent experiments, each of which was performed with triplicates. The concentrations of the COX inhibitors were determined according to the indications of the assay kits and the previous binding studies in the literature (see "Methods").

Taken together, the above experimental assays validated the novel interactions between

the three drugs (i.e., telmisartan, alendronate and chlorpropamide) and the COX proteins predicted by DTINet, which further demonstrated the accuracy of its prediction results and thus provided strong evidence to support its excellent predictive power. In addition, the experimentally validated interactions between these three drugs and the COX proteins can provide great opportunities for drug repositioning, i.e., finding the new functions (i.e., anti-inflammatory effects) of these drugs, and offer new insights into the understanding of their molecular mechanisms of drug action or side-effects of these drugs.

4.5 DISCUSSION

Recent advances in large-scale experimental approaches, e.g., mass spectrum-based methods [232, 233, 234, 235], have made great contributions to drug development and drug target identification with high throughput and accuracy. Nevertheless, these methods can only test one chemical at a time to determine the interacting proteins. In addition, they are still costly even in a high-throughput manner. Compared to these proteomics-based methods, computational approaches can allow high-throughput prediction for both drugs and targets, learning their intrinsic features and inferring the interactions between all potential drug-target pairs simultaneously. Based on the computational methods, we can identify a list of promising candidates and thus greatly reduce the huge search space of drug-target pairs that need to be validated by wet-lab experiments.

Previously, Guney et al. [236] showed that the network-based proximity of known drug targets and disease-associated proteins on the interactome can provide a good indicator for studying drug-disease associations and drug efficacy. Cheng et al. [237] also developed a network-based pipeline to predict new indications of existing drugs, which assumed that a drug can be applied to specific cancer types if the significantly mutated genes are enriched in those differentially expressed genes induced by the drug. These two methods were mainly used to study the drug-disease relationships but did not directly provide the information of new drug-target interactions, which instead was the major goal of our framework. Although the drug-disease relationships may provide more direct indications of existing drugs, knowing the explicit drug-target interactions can shed light on the underlying pharmacological mechanisms, which are important for understanding both the therapeutic and adverse effects of the corresponding drugs. In addition, the aforementioned two approaches simply focused on the distances between the disease-related and drug-related proteins, which would be sensitive to the incompleteness of known targets, disease genes and underlying protein-protein interactions. On the other hand, based on the systematic integration of heterogeneous network information, in principle, our approach can achieve better and more robust prediction

performance (e.g., with fewer false positive predictions) by considering diverse information from various types of network features.

Among the three drugs whose interactions with the COX proteins have been validated experimentally in our study, telmisartan displayed unique pleiotropic roles in addition to the renin-angiotensin system (RAS)-inhibition effects as an angiotensin II AT1 receptor antagonist/blocker. It has been reported that telmisartan acts as a selective modulator of the peroxisome proliferator-activated receptors (PPAR-γ and δ) [238]. Our findings probably add novel insights into its anti-inflammatory effects as a COX inhibitor. Several studies have indicated that telmisartan ameliorates the neuronal, airways, and coronary plaque inflammatory responses [239, 240]. Our findings provide direct evidence to support its interaction on the COX proteins. Its inhibitory effects on COX and inflammatory cytokine production may partially explain its anti-inflammatory indications.

For chlorpropamide, a sulfonylurea to increase the secretion of insulin to treat type 2 diabetes, there are few reports about its anti-inflammatory effects. Our findings indicate that chlorpropamide can also be a COX inhibitor though with weak binding affinity (IC50 around 300 μM), which may have implications on its adverse drug reactions on hematological changes, such as thrombocytopenia and granulocytopenia as in the hematologic syndromes induced by other cox inhibitors [241, 242]. Alendronate, another drug that we have tested, is a bisphosphonate drug and potent inhibitor of bone resorption used for the treatment of metabolic bone diseases. Previous studies have shown that alendronate can also suppress the production of inflammatory cytokines and matrix metalloproteinases in alveolar macrophages for its anti-inflammatory effects [243]. Our findings about its COX inhibition suggest that it may interact with COX for its immunological effects. Overall, we have combined the computational analysis with experimental validation to discover novel drug-target interactions. Our findings are particularly helpful for understanding the unknown pharmacological effects of existing drugs and identifying their potential new applications.

A future direction of our work is to include more heterogeneous network data in our framework. While we used only four domains (i.e., drugs, proteins, diseases, and side-effects) of information in this work, we highlight that DTINet is a scalable framework in that more additional networks can be easily incorporated into the current prediction pipeline. Other biological entities of different types, such as gene expression, pathways, symptoms, and Gene Ontology (GO) annotations, can also be integrated into the heterogeneous network for DTI prediction. Although it was only applied to predict missing DTIs in this work, DTINet is a versatile approach and definitely can also be applied to various link prediction problems, e.g., predictions of drug-side-effect associations, drug-drug interactions and protein-disease associations.

Chapter 5: Conclusions and Future Directions

The progress in biotechnology has been enabling the generation of large-scale, multi-model, holistic data to characterize biological systems. Despite the exciting opportunity, extracting biologically meaningful signals from noisy, incomplete, high-dimensional high-throughput data remains challenging. Computational approaches, exemplified by statistical methods and machine learning algorithms, have great promise of translating biomedical data into knowledge by processing, integrating, and analyzing them in an effective and efficient way. In this dissertation, I developed new machine learning algorithms to address unique challenges in those problems and to assist scientific discovery. In particular, I have focused on system biology and studied the sequence-structure-function relationship of proteins, which has important implications for understanding the mechanism of human diseases and for designing novel therapeutics. I used representation learning as a key technique to exploit values in biomedical data, including protein sequence datasets, protein and molecule 3D structures, and heterogeneous networks of biomedical relationships. I first developed a deep learning framework to learn representations that reflect evolutionary contexts for protein sequences, and the framework successfully discovered new protein variants with enhanced drug resistance (Chapter 2). I also developed a geometric deep learning model that learns representations of protein and compound structures for protein-compound binding affinity, which is able to prioritize high-confidence, high-affinity binding pairs (Chapter 3). Lastly, I developed a machine learning algorithm to integrate heterogeneous networks by learning compact network representations, and that algorithm also enabled the discovery of new targets for existing drugs (Chapter 4). Taken together, the results presented in this dissertation demonstrate the potential of machine learning to transform biomedical data into knowledge discovery.

Machine learning, especially deep learning, has evolved rapidly over the past years and achieved exciting progress in biology and medicine as well. Remarkable examples include AlphaFold [50] which predicts highly accurate 3D protein structure from a sequence using deep learning. As we have seen repeatedly in this dissertation, there are unique challenges in biological domain problems that do not exist in other popular machine learning applications such as image classification. Because of this, directly applying off-the-shelf machine learning algorithms developed for generic purposes or problems in other domains, would only achieve limited performance or unsatisfactory results, as we have shown in this dissertation. Therefore, developing domain-tailored methods becomes critical for addressing biomedical problems using machine learning. Looking forward, I envision many opportunities that

can harness heterogeneous biological data and leverage intelligent computational models to improve diagnostics and therapeutics for human diseases, which I outline below.

Transferable machine learning for biomedicine in low-data regimes In many important biomedical problems, such as those related to protein function and design, disease mutations studies, and functional genomics, the effective sample sizes are much smaller than what we expect for popular machine learning applications in vision and natural language. In some cases, the data are harder and even impossible to obtain due to ethical and practical reasons, such as the controlled experiments of gene knockdown in humans. These challenges underscore the importance of developing transferable machine learning models for data-limited scenarios. The role of transferable learning, including meta-learning, few-shot learning, and self-supervised training techniques, will be attractive to address small-data issues and reason about never-before-seen data in important biomedical applications, such as annotating novel types of single-cells and generating functional protein sequences based on limited examples.

Human-AI workflows for biological discovery and design Biological discovery and design (e.g., designing protein sequences or chemical molecules) is a problem of inference from incomplete and imperfect information, for which artificial intelligence (AI) techniques are well-suited. It is highly useful to develop human-AI workflows where data-driven AI models are combined with human knowledge to deliver robust and reliable strategies for biological discovery and design. Two directions of investigation can be expanded under this topic. One of them is the biological discovery with domain-tailored machine learning models, e.g., models to predict sequence-function relationships of proteins or the structure-property relationships of molecules. These models can be used to prioritize novel protein sequences or molecule structures that are likely to demonstrate improved properties. In particular, the models can be combined with techniques such as the Gaussian process to quantify uncertainty in predictions, which helps focus experimental efforts on hypotheses with a high likelihood of success or alerts researchers to experiments with greater novelty although also with a greater risk of failure. The other direction is model-based biological design, in which a query oracle (e.g., a trained neural network or a lab procedure that predicts/measures a property) is given and the goal is to design protein sequences or molecule structures to achieve the desired property. In applications such as protein engineering, most in-lab protocols of directed evolution create the variants library by introducing random mutations, which requires iterated rounds of selection to obtain the desired proteins. Computational solutions that design variants in a more intelligent way would thus greatly facilitate current

lab techniques in terms of cost and efficiency. For this purpose, one can combine machine learning models with Bayesian optimization to explore and exploit the search space and generate promising candidates for synthesis and validation. In both directions, AI models are expected to be integrated into current lab procedures to form a human-AI workflow (e.g., active learning loop) where AI predictions and human efforts iteratively refine the biological hypotheses.

Trustworthy machine learning for biomedicine In the long term, machine learning has the potential to support or provide decision-making in biomedicine. Using machine learning in the decision-making process of biomedicine, with or without human supervision, requires responsible and trustworthy models. There has been growing interest from researchers and practitioners to develop machine learning models that are not only accurate, but also explainable, robust, and privacy-preserving. Future directions along this line include (1) Transparency and explainability: To be accepted into decision-making processes in biomedicine, models should be designed to explicitly demonstrate their decision process when possible and to allow for interventions if necessary. (2) Model robustness: To achieve reliable decision-making, machine learning models and their explanations should be robust against distribution shifts and missing counterfactuals. (3) Privacy-preserving: Increased collaborations and data sharing will enhance machine learning models for biomedicine while posing concerns of individual privacy and intellectual property; fundamentally new algorithmic solutions based on techniques such as multi-party computation, differential privacy, and federated learning can facilitate the sharing of biomedical data without divulging sensitive privacy. Overall, we hope to move beyond black-box predictions and provide trustworthy machine learning tools that enable decision-making in biomedicine.

References

[1] M. Cao, C. M. Pietras, X. Feng, K. J. Doroschak, T. Schaffner, J. Park, H. Zhang, L. J. Cowen, and B. J. Hescott, "New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence," *Bioinformatics*, vol. 30, no. 12, pp. i219–i227, 2014.

[2] E. Glaab, A. Baudot, N. Krasnogor, R. Schneider, and A. Valencia, "Enrichnet: network-based gene set enrichment analysis," *Bioinformatics*, vol. 28, no. 18, pp. i451–i457, 2012.

[3] S. Köhler, S. Bauer, D. Horn, and P. N. Robinson, "Walking the interactome for prioritization of candidate disease genes," *The American Journal of Human Genetics*, vol. 82, no. 4, pp. 949–958, 2008.

[4] C.-S. Liao, K. Lu, M. Baym, R. Singh, and B. Berger, "Isorankn: spectral methods for global alignment of multiple protein networks," *Bioinformatics*, vol. 25, no. 12, pp. i253–i258, 2009.

[5] S. Navlakha and C. Kingsford, "The power of protein interaction networks for associating genes with diseases," *Bioinformatics*, vol. 26, no. 8, pp. 1057–1063, 2010.

[6] A.-C. Gavin, P. Aloy, P. Grandi, R. Krause, M. Boesche, M. Marzioch, C. Rau, L. J. Jensen, S. Bastuck, B. Dümpelfeld et al., "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, no. 7084, pp. 631–636, 2006.

[7] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris, "Genemania: a real-time multiple association network integration algorithm for predicting gene function," *Genome biology*, vol. 9, no. 1, pp. 1–15, 2008.

[8] I. Lee, U. M. Blom, P. I. Wang, J. E. Shim, and E. M. Marcotte, "Prioritizing candidate disease genes by network-based boosting of genome-wide association data," *Genome research*, vol. 21, no. 7, pp. 1109–1121, 2011.

[9] A. K. Wong, A. Krishnan, V. Yao, A. Tadych, and O. G. Troyanskaya, "Imp 2.0: a multi-species functional genomics portal for integration, visualization and prediction of protein functions and networks," *Nucleic acids research*, vol. 43, no. W1, pp. W128–W133, 2015.

[10] D. Szklarczyk, A. L. Gable, D. Lyon, A. Junge, S. Wyder, J. Huerta-Cepas, M. Simonovic, N. T. Doncheva, J. H. Morris, P. Bork et al., "String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets," *Nucleic acids research*, vol. 47, no. D1, pp. D607–D613, 2019.

[11] T. Milenković and N. Pržulj, "Uncovering biological network function via graphlet degree signatures," *Cancer informatics*, vol. 6, pp. CIN–S680, 2008.

[12] T. I. Oprea, C. G. Bologa, S. Brunak, A. Campbell, G. N. Gan, A. Gaulton, S. M. Gomez, R. Guha, A. Hersey, J. Holmes et al., "Unexplored therapeutic opportunities in the human genome," *Nature reviews Drug discovery*, vol. 17, no. 5, pp. 317–332, 2018.

[13] E. J. Needham, B. L. Parker, T. Burykin, D. E. James, and S. J. Humphrey, "Illuminating the dark phosphoproteome," *Science signaling*, vol. 12, no. 565, 2019.

[14] S. Wang, H. Cho, C. Zhai, B. Berger, and J. Peng, "Exploiting ontology graph for predicting sparsely annotated gene function," *Bioinformatics*, vol. 31, no. 12, pp. i357–i364, 2015.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[16] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[18] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[19] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church, "Unified rational protein engineering with sequence-based deep representation learning," *Nature methods*, vol. 16, no. 12, pp. 1315–1322, 2019.

[20] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *bioRxiv*, 2019. [Online]. Available: https://www.biorxiv.org/content/10.1101/622803v4

[21] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel, and Y. S. Song, "Evaluating protein transfer learning with tape," *Advances in neural information processing systems*, vol. 32, p. 9689, 2019.

[22] A. Madani, B. McCann, N. Naik, N. S. Keskar, N. Anand, R. R. Eguchi, P.-S. Huang, and R. Socher, "Progen: Language modeling for protein generation," *arXiv preprint arXiv:2004.03497*, 2020.

[23] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rihawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger et al., "Prottrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[24] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, 2020.

[25] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[26] Y. Liu, P. Palmedo, Q. Ye, B. Berger, and J. Peng, "Enhancing evolutionary couplings with deep convolutional neural networks," *Cell systems*, vol. 6, no. 1, pp. 65–74, 2018.

[27] M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, and E. Aurell, "Improved contact prediction in proteins: using pseudolikelihoods to infer potts models," *Physical Review E*, vol. 87, no. 1, p. 012707, 2013.

[28] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, "Direct-coupling analysis of residue coevolution captures native contacts across many protein families," *Proceedings of the National Academy of Sciences*, vol. 108, no. 49, pp. E1293–E1301, 2011.

[29] H. Kamisetty, S. Ovchinnikov, and D. Baker, "Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era," *Proceedings of the National Academy of Sciences*, vol. 110, no. 39, pp. 15 674–15 679, 2013.

[30] D. T. Jones, D. W. Buchan, D. Cozzetto, and M. Pontil, "Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments," *Bioinformatics*, vol. 28, no. 2, pp. 184–190, 2012.

[31] D. T. Jones, T. Singh, T. Kosciolek, and S. Tetchner, "Metapsicov: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins," *Bioinformatics*, vol. 31, no. 7, pp. 999–1006, 2015.

[32] L. Kaján, T. A. Hopf, M. Kalaš, D. S. Marks, and B. Rost, "Freecontact: fast and free software for protein contact prediction from residue co-evolution," *BMC bioinformatics*, vol. 15, no. 1, pp. 1–6, 2014.

[33] D. S. Marks, L. J. Colwell, R. Sheridan, T. A. Hopf, A. Pagnani, R. Zecchina, and C. Sander, "Protein 3d structure computed from evolutionary sequence variation," *PloS one*, vol. 6, no. 12, p. e28766, 2011.

[34] Y. Luo, G. Jiang, T. Yu, Y. Liu, L. Vo, H. Ding, Y. Su, W. W. Qian, H. Zhao, and J. Peng, "ECNet is an evolutionary context-integrated deep learning framework for protein engineering," *Nature Communications*, vol. 12, no. 1, Sep. 2021.

[35] Y. Luo and J. Peng, "Calibrated geometric deep learning improves kinase-drug binding prediction," *Under submission*, 2021.

[36] Y. Luo, X. Zhao, J. Zhou, J. Yang, Y. Zhang, W. Kuang, J. Peng, L. Chen, and J. Zeng, "A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information," *Nature communications*, vol. 8, no. 1, pp. 1–13, 2017.

[37] A. Cichonska, B. Ravikumar, R. J. Allaway, S. Park, F. Wan, O. Isayev, S. Li, M. Mason, A. Lamb, M. Jeon et al., "Crowdsourced mapping of unexplored target space of kinase inhibitors," *BioRxiv*, pp. 2019–12, 2020.

[38] Y. Ge, T. Tian, S. Huang, F. Wan, J. Li, S. Li, X. Wang, H. Yang, L. Hong, N. Wu, E. Yuan, Y. Luo, L. Cheng, C. Hu, Y. Lei, H. Shu, X. Feng, Z. Jiang, Y. Wu, Y. Chi, X. Guo, L. Cui, L. Xiao, Z. Li, C. Yang, Z. Miao, L. Chen, H. Li, H. Zeng, D. Zhao, F. Zhu, X. Shen, and J. Zeng, "An integrative drug repositioning framework discovered a potential therapeutic agent targeting COVID-19," *Signal Transduction and Targeted Therapy*, vol. 6, no. 1, Apr. 2021. [Online]. Available: https://doi.org/10.1038/s41392-021-00568-6

[39] Y. Luo, J. Zeng, B. Berger, and J. Peng, "Low-density locality-sensitive hashing boosts metagenomic binning," in *RECOMB*, vol. 9649. NIH Public Access, 2016, p. 255.

[40] Y. Luo, J. Ma, Y. Liu, Q. Ye, T. Ideker, and J. Peng, "Deciphering signaling specificity with deep neural networks." in *RECOMB*. Springer, 2018, pp. 266–268.

[41] Y. Luo, J. Ma, X. Zhao, Y. Su, Y. Liu, T. Ideker, and J. Peng, "Mitigating data scarcity in protein binding prediction using meta-learning." in *RECOMB*. Springer, 2019, pp. 305–307.

[42] A. Cichońska, B. Ravikumar, R. J. Allaway, F. Wan, S. Park, O. Isayev, S. Li, M. Mason, A. Lamb, Z. Tanoli et al., "Crowdsourced mapping of unexplored target space of kinase inhibitors," *Nature communications*, vol. 12, no. 1, pp. 1–18, 2021.

[43] P. Sashittal*, Y. Luo*, J. Peng, and M. El-Kebir, "Characterization of SARS-CoV-2 viral diversity within and across hosts," *bioRxiv:2020.05.07.083410*, May 2020. [Online]. Available: https://doi.org/10.1101/2020.05.07.083410

[44] Y. Su*, Y. Luo*, X. Zhao, Y. Liu, and J. Peng, "Integrating thermodynamic and sequence contexts improves protein-rna binding prediction," *PLoS computational biology*, vol. 15, no. 9, p. e1007283, 2019, presented at GLBIO'19.

[45] X. Liu, Y. Luo, P. Li, S. Song, and J. Peng, "Deep geometric representations for modeling effects of mutations on protein-protein binding affinity," *PLoS computational biology*, vol. 17, no. 8, p. e1009284, 2021.

[46] Y. Luo, S. Wang, J. Xiao, and J. Peng, "Large-scale integration of heterogeneous pharmacogenomic data for identifying drug mechanism of action." in *PSB*. World Scientific, 2018, pp. 44–55.

[47] Y. Luo, J. Peng, and J. Ma, "When causal inference meets deep learning," *Nature Machine Intelligence*, vol. 2, no. 8, pp. 426–427, 2020, news & Views.

[48] W. W. Qian, N. T. Russell, C. L. W. Simons, Y. Luo, M. D. Burke, and J. Peng, "Integrating deep neural networks and symbolic inference for organic reactivity prediction," *chemRxiv:11659563*, Jan. 2020. [Online]. Available: https://doi.org/10.26434/chemrxiv.11659563.v1

[49] J. Ma, S. H. Fong, Y. Luo, C. J. Bakkenist, J. P. Shen, S. Mourragui, L. F. Wessels, M. Hafner, R. Sharan, J. Peng et al., "Few-shot learning creates predictive models of drug response that translate from high-throughput screens to individual patients," *Nature Cancer*, pp. 1–12, 2021.

[50] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko et al., "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[51] F. H. Arnold, "Design by directed evolution," *Acc. Chem. Res.*, vol. 31, no. 3, pp. 125–131, Mar. 1998.

[52] H. Zhao, L. Giver, Z. Shao, J. A. Affholter, and F. H. Arnold, "Molecular evolution by staggered extension process (StEP) in vitro recombination," *Nat. Biotechnol.*, vol. 16, no. 3, pp. 258–261, Mar. 1998.

[53] P. A. Romero and F. H. Arnold, "Exploring protein fitness landscapes by directed evolution," *Nat. Rev. Mol. Cell Biol.*, vol. 10, no. 12, pp. 866–876, Dec. 2009.

[54] K. K. Yang, Z. Wu, and F. H. Arnold, "Machine-learning-guided directed evolution for protein engineering," *Nat. Methods*, vol. 16, no. 8, pp. 687–694, Aug. 2019.

[55] Z. Wu, S. B. J. Kan, R. D. Lewis, B. J. Wittmann, and F. H. Arnold, "Machine learning-assisted directed protein evolution with combinatorial libraries," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 116, no. 18, pp. 8852–8858, Apr. 2019.

[56] C. N. Bedbrook, K. K. Yang, J. E. Robinson, E. D. Mackey, V. Gradinaru, and F. H. Arnold, "Machine learning-guided channelrhodopsin engineering enables minimally invasive optogenetics," *Nat. Methods*, Oct. 2019.

[57] C. N. Bedbrook, K. K. Yang, A. J. Rice, V. Gradinaru, and F. H. Arnold, "Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization," *PLoS Comput. Biol.*, vol. 13, no. 10, p. e1005786, Oct. 2017.

[58] P. A. Romero, A. Krause, and F. H. Arnold, "Navigating the protein fitness landscape with gaussian processes," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 110, no. 3, pp. E193–201, Jan. 2013.

[59] S. Biswas, G. Kuznetsov, P. J. Ogden, N. J. Conway, and others, "Toward machine-guided design of proteins," *bioRxiv*, 2018.

[60] N. Zhou, Y. Jiang, T. R. Bergquist, A. J. Lee, B. Z. Kacsoh, A. W. Crocker, K. A. Lewis, G. Georghiou, H. N. Nguyen, M. N. Hamid et al., "The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens," *Genome biology*, vol. 20, no. 1, pp. 1–23, 2019.

[61] J. Upmeier zu Belzen, T. Bürgel, S. Holderbach, F. Bubeck, L. Adam, C. Gandor, M. Klein, J. Mathony, P. Pfuderer, L. Platz, M. Przybilla, M. Schwendemann, D. Heid, M. D. Hoffmann, M. Jendrusch, C. Schmelas, M. Waldhauer, I. Lehmann, D. Niopek, and R. Eils, "Leveraging implicit knowledge in neural networks for functional dissection and engineering of proteins," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 225–235, May 2019.

[62] S. Wang, H. Cho, C. Zhai, B. Berger, and J. Peng, "Exploiting ontology graph for predicting sparsely annotated gene function," *Bioinformatics*, vol. 31, no. 12, pp. i357–64, June 2015.

[63] M. L. Bileschi, D. Belanger, D. Bryant, T. Sanderson, B. Carter, D. Sculley, M. A. DePristo, and L. J. Colwell, "Using deep learning to annotate the protein universe," *bioRxiv*, p. 626507, July 2019.

[64] T. A. Hopf, J. B. Ingraham, F. J. Poelwijk, C. P. I. Schärfe, M. Springer, C. Sander, and D. S. Marks, "Mutation effects predicted from sequence co-variation," *Nat. Biotechnol.*, vol. 35, no. 2, pp. 128–135, Feb. 2017.

[65] A. J. Riesselman, J. B. Ingraham, and D. S. Marks, "Deep generative models of genetic variation capture the effects of mutations," *Nat. Methods*, vol. 15, no. 10, pp. 816–822, Oct. 2018.

[66] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church, "Unified rational protein engineering with sequence-based deep representation learning," *Nat. Methods*, Oct. 2019.

[67] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma et al., "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, 2021.

[68] T. Bepler and B. Berger, "Learning protein sequence embeddings using information from structure," *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[69] S. Biswas, G. Khimulya, E. C. Alley, K. M. Esvelt, and G. M. Church, "Low-n protein engineering with data-efficient deep learning," *Nature Methods*, vol. 18, no. 4, pp. 389–396, 2021.

[70] B. L. Hie, K. K. Yang, and P. S. Kim, "Evolutionary velocity with protein language models," *bioRxiv*, 2021.

[71] B. Hie, E. D. Zhong, B. Berger, and B. Bryson, "Learning the language of viral evolution and escape," *Science*, vol. 371, no. 6526, pp. 284–288, 2021.

[72] J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, and A. Rives, "Language models enable zero-shot prediction of the effects of mutations on protein function," *bioRxiv*, 2021.

[73] UniProt Consortium, "UniProt: a worldwide hub of protein knowledge," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D506–D515, Jan. 2019.

[74] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart, E. L. L. Sonnhammer, L. Hirsh, L. Paladin, D. Piovesan, S. C. E. Tosatto, and R. D. Finn, "The pfam protein families database in 2019," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D427–D432, Jan. 2019.

[75] J. M. Schmiedel and B. Lehner, "Determining protein structures using deep mutagenesis," *Nat. Genet.*, vol. 51, no. 7, pp. 1177–1186, July 2019.

[76] N. J. Rollins, K. P. Brock, F. J. Poelwijk, M. A. Stiffler, N. P. Gauthier, C. Sander, and D. S. Marks, "Inferring protein 3D structure from deep mutation scans," *Nat. Genet.*, vol. 51, no. 7, pp. 1170–1176, July 2019.

[77] S. Ovchinnikov, H. Kamisetty, and D. Baker, "Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information," *Elife*, vol. 3, p. e02030, 2014.

[78] T. A. Hopf, C. P. I. Schärfe, J. P. G. L. M. Rodrigues, A. G. Green, O. Kohlbacher, C. Sander, A. M. J. J. Bonvin, and D. S. Marks, "Sequence co-evolution gives 3D contacts and structures of protein complexes," *Elife*, vol. 3, Sep. 2014.

[79] C. L. Araya, D. M. Fowler, W. Chen, I. Muniez, J. W. Kelly, and S. Fields, "A fundamental protein property, thermodynamic stability, revealed solely from large-scale measurements of protein function," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 109, no. 42, pp. 16 858–16 863, Oct. 2012.

[80] S. Seemayer, M. Gruber, and J. Söding, "CCMpred–fast and precise prediction of protein residue-residue contacts from correlated mutations," *Bioinformatics*, vol. 30, no. 21, pp. 3128–3130, Nov. 2014.

[81] V. E. Gray, R. J. Hause, J. Luebeck, J. Shendure, and D. M. Fowler, "Quantitative missense variant effect prediction using Large-Scale mutagenesis data," *Cell Syst*, vol. 6, no. 1, pp. 116–124.e3, Jan. 2018.

[82] I. A. Adzhubei, S. Schmidt, and L. Peshkin, "ramensky ve, gerasimova a., bork p., kondrashov AS, sunyaev sr," *Nat. Methods*, vol. 7, no. 4, pp. 248–249, 2010.

[83] M. Kircher, D. M. Witten, P. Jain, B. J. O'Roak, G. M. Cooper, and J. Shendure, "A general framework for estimating the relative pathogenicity of human genetic variants," *Nat. Genet.*, vol. 46, no. 3, pp. 310–315, Mar. 2014.

[84] J. W. H. Schymkowitz, F. Rousseau, I. C. Martins, J. Ferkinghoff-Borg, F. Stricher, and L. Serrano, "Prediction of water and metal binding sites and their affinities by using the Fold-X force field," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 102, no. 29, pp. 10 147–10 152, July 2005.

[85] P. Gainza, K. E. Roberts, I. Georgiev, R. H. Lilien, D. A. Keedy, C.-Y. Chen, F. Reza, A. C. Anderson, D. C. Richardson, J. S. Richardson, and B. R. Donald, "OSPREY: protein design with ensembles, flexibility, and provable algorithms," *Methods Enzymol.*, vol. 523, pp. 87–107, 2013.

[86] M. S. Breen, C. Kemena, P. K. Vlasov, C. Notredame, and F. A. Kondrashov, "Epistasis as the primary factor in molecular evolution," *Nature*, vol. 490, no. 7421, pp. 535–538, Oct. 2012.

[87] D. M. McCandlish, P. Shah, and J. B. Plotkin, "Epistasis and the dynamics of reversion in molecular evolution," *Genetics*, vol. 203, no. 3, pp. 1335–1351, July 2016.

[88] D. S. Marks, L. J. Colwell, R. Sheridan, T. A. Hopf, A. Pagnani, R. Zecchina, and C. Sander, "Protein 3D structure computed from evolutionary sequence variation," *PLoS One*, vol. 6, no. 12, p. e28766, Dec. 2011.

[89] B. Bolognesi, A. J. Faure, M. Seuma, J. M. Schmiedel, G. G. Tartaglia, and B. Lehner, "The mutational landscape of a prion-like domain," *Nat. Commun.*, vol. 10, no. 1, p. 4162, Sep. 2019.

[90] C. E. Gonzalez and M. Ostermeier, "Pervasive pairwise intragenic epistasis among sequential mutations in TEM-1 β-Lactamase," *J. Mol. Biol.*, vol. 431, no. 10, pp. 1981–1992, May 2019.

[91] K. S. Sarkisyan, D. A. Bolotin, M. V. Meer, D. R. Usmanova, A. S. Mishin, G. V. Sharonov, D. N. Ivankov, N. G. Bozhanova, M. S. Baranov, O. Soylemez, N. S. Bogatyreva, P. K. Vlasov, E. S. Egorov, M. D. Logacheva, A. S. Kondrashov, D. M. Chudakov, E. V. Putintseva, I. Z. Mamedov, D. S. Tawfik, K. A. Lukyanov, and F. A. Kondrashov, "Local fitness landscape of the green fluorescent protein," *Nature*, vol. 533, no. 7603, pp. 397–401, May 2016.

[92] M. B. Doud and J. D. Bloom, "Accurate measurement of the effects of all amino-acid mutations on influenza hemagglutinin," *Viruses*, vol. 8, no. 6, p. 155, 2016.

[93] N. C. Wu, J. Otwinowski, A. J. Thompson, C. M. Nycholat, A. Nourmohammad, and I. A. Wilson, "Major antigenic site b of human influenza h3n2 viruses has an evolving local fitness landscape," *Nature communications*, vol. 11, no. 1, pp. 1–10, 2020.

[94] H. K. Haddox, A. S. Dingens, S. K. Hilton, J. Overbaugh, and J. D. Bloom, "Mapping mutational effects along the evolutionary landscape of hiv envelope," *Elife*, vol. 7, p. e34420, 2018.

[95] T. N. Starr, A. J. Greaney, S. K. Hilton, D. Ellis, K. H. Crawford, A. S. Dingens, M. J. Navarro, J. E. Bowen, M. A. Tortorici, A. C. Walls et al., "Deep mutational scanning of sars-cov-2 receptor binding domain reveals constraints on folding and ace2 binding," *Cell*, vol. 182, no. 5, pp. 1295–1310, 2020.

[96] M. Steinegger, M. Meier, M. Mirdita, H. Vöhringer, S. J. Haunsberger, and J. Söding, "HH-suite3 for fast remote homology detection and deep protein annotation," *BMC Bioinformatics*, vol. 20, no. 1, p. 473, Sep. 2019.

[97] M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, and E. Aurell, "Improved contact prediction in proteins: using pseudolikelihoods to infer potts models," *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, vol. 87, no. 1, p. 12707, Jan. 2013.

[98] H. Kamisetty, S. Ovchinnikov, and D. Baker, "Assessing the utility of coevolution-based residue–residue contact predictions in a sequence- and structure-rich era," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 110, no. 39, pp. 15 674–15 679, Sep. 2013.

[99] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[100] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[101] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv*, Feb. 2015.

[102] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[103] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization. arxiv. org," *Mach. Learn.*, 2014.

[104] K. K. Yang, Z. Wu, C. N. Bedbrook, and F. H. Arnold, "Learned protein embeddings for machine learning," *Bioinformatics*, vol. 34, no. 15, pp. 2642–2648, Aug. 2018.

[105] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *arXiv*, May 2014.

[106] J.-E. Shin, A. J. Riesselman, A. W. Kollasch, C. McMahon, E. Simon, C. Sander, A. Manglik, A. C. Kruse, and D. S. Marks, "Protein design and variant prediction using autoregressive generative models," *Nature communications*, vol. 12, no. 1, pp. 1–11, 2021.

[107] S. Gelman, P. A. Romero, and A. Gitter, "Neural networks to learn protein sequence-function relationships from deep mutational scanning data," *bioRxiv*, 2020.

[108] C. A. Olson, N. C. Wu, and R. Sun, "A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain," *Curr. Biol.*, vol. 24, no. 22, pp. 2643–2651, Nov. 2014.

[109] D. Melamed, D. L. Young, C. E. Gamble, C. R. Miller, and S. Fields, "Deep mutational scanning of an RRM domain of the saccharomyces cerevisiae poly(a)-binding protein," *RNA*, vol. 19, no. 11, pp. 1537–1551, Nov. 2013.

[110] E. Firnberg, J. W. Labonte, J. J. Gray, and M. Ostermeier, "A comprehensive, high-resolution map of a gene's fitness landscape," *Molecular biology and evolution*, vol. 31, no. 6, pp. 1581–1592, 2014.

[111] G. Diss and B. Lehner, "The genetic landscape of a physical interaction," 2018.

[112] B. Lehner, "Molecular mechanisms of epistasis within and between genes," *Trends Genet.*, vol. 27, no. 8, pp. 323–331, Aug. 2011.

[113] P. Rentzsch, D. Witten, G. M. Cooper, J. Shendure, and M. Kircher, "CADD: predicting the deleteriousness of variants throughout the human genome," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D886–D894, Jan. 2019.

[114] H. Song, B. J. Bremer, E. C. Hinds, G. Raskutti, and P. A. Romero, "Inferring protein sequence-function relationships with large-scale positive-unlabeled learning," *Cell Systems*, vol. 12, no. 1, pp. 92–101, 2021.

[115] Y. Saito, M. Oikawa, H. Nakazawa, T. Niide, T. Kameda, K. Tsuda, and M. Umetsu, "Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins," *ACS synthetic biology*, vol. 7, no. 9, pp. 2014–2022, 2018.

[116] C. Hsu, H. Nisonoff, C. Fannjiang, and J. Listgarten, "Combining evolutionary and assay-labelled data for protein fitness prediction," *bioRxiv*, 2021.

[117] M. M. Attwood, D. Fabbro, A. V. Sokolov, S. Knapp, and H. B. Schiöth, "Trends in kinase drug discovery: Targets, indications and inhibitor design," *Nature Reviews Drug Discovery*, pp. 1–23, 2021.

[118] P. Cohen, D. Cross, and P. A. Jänne, "Kinase drug discovery 20 years after imatinib: progress and future directions," *Nature Reviews Drug Discovery*, pp. 1–19, 2021.

[119] S. M. Hanson, G. Georghiou, M. K. Thakur, W. T. Miller, J. S. Rest, J. D. Chodera, and M. A. Seeliger, "What makes a kinase promiscuous for inhibitors?" *Cell chemical biology*, vol. 26, no. 3, pp. 390–399, 2019.

[120] C. H. Arrowsmith, J. E. Audia, C. Austin, J. Baell, J. Bennett, J. Blagg, C. Bountra, P. E. Brennan, P. J. Brown, M. E. Bunnage et al., "The promise and peril of chemical probes," *Nature chemical biology*, vol. 11, no. 8, pp. 536–541, 2015.

[121] K. Bleakley and Y. Yamanishi, "Supervised prediction of drug–target interactions using bipartite local models," *Bioinformatics*, vol. 25, no. 18, pp. 2397–2403, 2009.

[122] M. C. Cobanoglu, C. Liu, F. Hu, Z. N. Oltvai, and I. Bahar, "Predicting drug–target interactions using probabilistic matrix factorization," *Journal of chemical information and modeling*, vol. 53, no. 12, pp. 3399–3409, 2013.

[123] X. Zheng, H. Ding, H. Mamitsuka, and S. Zhu, "Collaborative matrix factorization with multiple similarities for predicting drug-target interactions," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1025–1033.

[124] A. Cichonska, B. Ravikumar, E. Parri, S. Timonen, T. Pahikkala, A. Airola, K. Wennerberg, J. Rousu, and T. Aittokallio, "Computational-experimental approach to drug-target interaction mapping: a case study on kinase inhibitors," *PLoS computational biology*, vol. 13, no. 8, p. e1005678, 2017.

[125] H. Öztürk, A. Özgür, and E. Ozkirimli, "Deepdta: deep drug–target binding affinity prediction," *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, 2018.

[126] M. Karimi, D. Wu, Z. Wang, and Y. Shen, "Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks," *Bioinformatics*, vol. 35, no. 18, pp. 3329–3338, 2019.

[127] M. Tsubaki, K. Tomii, and J. Sese, "Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences," *Bioinformatics*, vol. 35, no. 2, pp. 309–318, 2019.

[128] M. Jiang, Z. Li, S. Zhang, S. Wang, X. Wang, Q. Yuan, and Z. Wei, "Drug–target affinity prediction using graph neural network and contact maps," *RSC Advances*, vol. 10, no. 35, pp. 20 701–20 712, 2020.

[129] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, "Graphdta: Predicting drug–target binding affinity with graph neural networks," *Bioinformatics*, vol. 37, no. 8, pp. 1140–1147, 2021.

[130] B. Hie, B. D. Bryson, and B. Berger, "Leveraging uncertainty in machine learning accelerates biological discovery and design," *Cell systems*, vol. 11, no. 5, pp. 461–477, 2020.

[131] P. W. Rose, A. Prlić, A. Altunkaya, C. Bi, A. R. Bradley, C. H. Christie, L. D. Costanzo, J. M. Duarte, S. Dutta, Z. Feng et al., "The rcsb protein data bank: integrative view of protein, gene and 3d structural information," *Nucleic acids research*, p. gkw1000, 2016.

[132] O. P. Van Linden, A. J. Kooistra, R. Leurs, I. J. De Esch, and C. De Graaf, "Klifs: a knowledge-based structural database to navigate kinase–ligand interaction space," *Journal of medicinal chemistry*, vol. 57, no. 2, pp. 249–277, 2014.

[133] G. K. Kanev, C. de Graaf, B. A. Westerman, I. J. de Esch, and A. J. Kooistra, "Klifs: an overhaul after the first 5 years of supporting kinase research," *Nucleic acids research*, vol. 49, no. D1, pp. D562–D569, 2021.

[134] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[135] H. Zeng and D. K. Gifford, "Quantification of uncertainty in peptide-mhc binding prediction improves high-affinity peptide selection for therapeutic design," *Cell systems*, vol. 9, no. 2, pp. 159–166, 2019.

[136] A. P. Soleimany, A. Amini, S. Goldman, D. Rus, S. N. Bhatia, and C. W. Coley, "Evidential deep learning for guided molecular property prediction and discovery," *ACS central science*, vol. 7, no. 8, pp. 1356–1367, 2021.

[137] J. Lim, S. Ryu, K. Park, Y. J. Choe, J. Ham, and W. Y. Kim, "Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation," *Journal of chemical information and modeling*, vol. 59, no. 9, pp. 3981–3988, 2019.

[138] L. Zheng, J. Fan, and Y. Mu, "Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction," *ACS omega*, vol. 4, no. 14, pp. 15 956–15 965, 2019.

[139] J. Zhou, S. Li, L. Huang, H. Xiong, F. Wang, T. Xu, H. Xiong, and D. Dou, "Distance-aware molecule graph attention network for drug-target binding affinity prediction," *arXiv preprint arXiv:2012.09624*, 2020.

[140] H. Hassan-Harrirou, C. Zhang, and T. Lemmin, "Rosenet: improving binding affinity prediction by leveraging molecular mechanics energies with an ensemble of 3d convolutional neural networks," *Journal of chemical information and modeling*, vol. 60, no. 6, pp. 2791–2802, 2020.

[141] S. Li, F. Wan, H. Shu, T. Jiang, D. Zhao, and J. Zeng, "Monn: a multi-objective neural network for predicting compound-protein interactions and affinities," *Cell Systems*, vol. 10, no. 4, pp. 308–322, 2020.

[142] S. Li, J. Zhou, T. Xu, L. Huang, F. Wang, H. Xiong, W. Huang, D. Dou, and H. Xiong, "Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 975–985.

[143] K. Ali, D. R. Soond, R. Pineiro, T. Hagemann, W. Pearce, E. L. Lim, H. Bouabe, C. L. Scudamore, T. Hancox, H. Maecker et al., "Inactivation of PI3K p110δ breaks regulatory t-cell-mediated immune tolerance to cancer," *Nature*, vol. 510, no. 7505, pp. 407–411, 2014.

[144] M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber, and P. P. Zarrinkar, "Comprehensive analysis of kinase inhibitor selectivity," *Nature biotechnology*, vol. 29, no. 11, pp. 1046–1051, 2011.

[145] J. Tang, A. Szwajda, S. Shakyawar, T. Xu, P. Hintsanen, K. Wennerberg, and T. Aittokallio, "Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis," *Journal of Chemical Information and Modeling*, vol. 54, no. 3, pp. 735–743, 2014.

[146] J. Ingraham, V. Garg, R. Barzilay, and T. Jaakkola, "Generative models for graph-based protein design," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[147] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, 2018.

[148] R. J. Townshend, M. Vögele, P. Suriana, A. Derry, A. Powers, Y. Laloudakis, S. Balachandar, B. Jing, B. Anderson, S. Eismann et al., "Atom3d: Tasks on molecules in three dimensions," *arXiv preprint arXiv:2012.04035*, 2020.

[149] Y. Shi, Z. Huang, W. Wang, H. Zhong, S. Feng, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," *arXiv preprint arXiv:2009.03509*, 2020.

[150] A. L. Maas, A. Y. Hannun, A. Y. Ng et al., "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30. Citeseer, 2013, p. 3.

[151] B. Jing, S. Eismann, P. Suriana, R. J. Townshend, and R. Dror, "Learning from protein structure with geometric vector perceptrons," *International Conference on Learning Representations (ICLR)*, 2021.

[152] A. Ashukha, A. Lyzhov, D. Molchanov, and D. Vetrov, "Pitfalls of in-domain uncertainty estimation and ensembling in deep learning," *International Conference on Learning Representations (ICLR)*, 2020.

[153] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2796–2804.

[154] D. Levi, L. Gispan, N. Giladi, and E. Fetaya, "Evaluating and calibrating uncertainty prediction in regression tasks," *arXiv preprint arXiv:1905.11659*, 2019.

[155] H. Song, T. Diethe, M. Kull, and P. Flach, "Distribution calibration for regression," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5897–5906.

[156] A. Amini, W. Schwarting, A. Soleimany, and D. Rus, "Deep evidential regression," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 14 927–14 937.

[157] G. Scalia, C. A. Grambow, B. Pernici, Y.-P. Li, and W. H. Green, "Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction," *Journal of chemical information and modeling*, vol. 60, no. 6, pp. 2697–2717, 2020.

[158] Y. Chung, I. Char, H. Guo, J. Schneider, and W. Neiswanger, "Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification," *arXiv preprint arXiv:2109.10254*, 2021.

[159] R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function," *The Computer Journal*, vol. 14, no. 4, pp. 422–425, 1971.

[160] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright et al., "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[161] K. Tran, W. Neiswanger, J. Yoon, Q. Zhang, E. Xing, and Z. W. Ulissi, "Methods for comparing uncertainty quantifications for material property predictions," *Machine Learning: Science and Technology*, vol. 1, no. 2, p. 025006, 2020.

[162] V. Modi and R. L. Dunbrack, "Defining a new nomenclature for the structures of active and inactive kinases," *Proceedings of the National Academy of Sciences*, vol. 116, no. 14, pp. 6818–6827, 2019.

[163] V. Modi and R. Dunbrack, "Kincore: a web resource for structural classification of protein kinases and their inhibitors," *bioRxiv*, 2021.

[164] K. Huang, T. Fu, W. Gao, Y. Zhao, Y. Roohani, J. Leskovec, C. W. Coley, C. Xiao, J. Sun, and M. Zitnik, "Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development," *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021.

[165] T. Pahikkala, A. Airola, S. Pietilä, S. Shakyawar, A. Szwajda, J. Tang, and T. Aittokallio, "Toward more realistic drug–target interaction predictions," *Briefings in bioinformatics*, vol. 16, no. 2, pp. 325–337, 2015.

[166] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," in *International conference on machine learning*. PMLR, 2018, pp. 2323–2332.

[167] Z. Liu, Y. Li, L. Han, J. Li, J. Liu, Z. Zhao, W. Nie, Y. Liu, and R. Wang, "Pdb-wide collection of binding data: current status of the pdbbind database," *Bioinformatics*, vol. 31, no. 3, pp. 405–412, 2015.

[168] "Pubchem3d release notes." [Online]. Available: https://pubchemdocs.ncbi.nlm.nih.gov/pubchem3d

[169] K. Tunyasuvunakool, J. Adler, Z. Wu, T. Green, M. Zielinski, A. Žídek, A. Bridgland, A. Cowie, C. Meyer, A. Laydon et al., "Highly accurate protein structure prediction for the human proteome," *Nature*, vol. 596, no. 7873, pp. 590–596, 2021.

[170] S. Whitebread, J. Hamon, D. Bojanic, and L. Urban, "Keynote review: in vitro safety pharmacology profiling: an essential tool for successful drug development," *Drug discovery today*, vol. 10, no. 21, pp. 1421–1433, 2005.

[171] B. R. Donald, *Algorithms in structural molecular biology*. MIT Press, 2011.

[172] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, and A. J. Olson, "Autodock4 and autodocktools4: Automated docking with selective receptor flexibility," *Journal of computational chemistry*, vol. 30, no. 16, pp. 2785–2791, 2009.

[173] M. J. Keiser, B. L. Roth, B. N. Armbruster, P. Ernsberger, J. J. Irwin, and B. K. Shoichet, "Relating protein pharmacology by ligand chemistry," *Nature biotechnology*, vol. 25, no. 2, pp. 197–206, 2007.

[174] K. Bleakley and Y. Yamanishi, "Supervised prediction of drug–target interactions using bipartite local models," *Bioinformatics*, vol. 25, no. 18, pp. 2397–2403, 2009.

[175] J.-P. Mei, C.-K. Kwoh, P. Yang, X.-L. Li, and J. Zheng, "Drug–target interaction prediction by learning from local information and neighbors," *Bioinformatics*, vol. 29, no. 2, pp. 238–245, 2013.

[176] Z. Xia, L.-Y. Wu, X. Zhou, and S. T. Wong, "Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces," *BMC systems biology*, vol. 4, no. Suppl 2, p. S6, 2010.

[177] T. van Laarhoven, S. B. Nabuurs, and E. Marchiori, "Gaussian interaction profile kernels for predicting drug–target interaction," *Bioinformatics*, vol. 27, no. 21, pp. 3036–3043, 2011.

[178] T. van Laarhoven and E. Marchiori, "Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile," *PloS one*, vol. 8, no. 6, p. e66952, 2013.

[179] M. Campillos, M. Kuhn, A.-C. Gavin, L. J. Jensen, and P. Bork, "Drug target identification using side-effect similarity," *Science*, vol. 321, no. 5886, pp. 263–266, 2008.

[180] S. Mizutani, E. Pauwels, V. Stoven, S. Goto, and Y. Yamanishi, "Relating drug–protein interaction network with drug side effects," *Bioinformatics*, vol. 28, no. 18, pp. i522–i528, 2012.

[181] F. Iorio, R. Bosotti, E. Scacheri, V. Belcastro, P. Mithbaokar, R. Ferriero, L. Murino, R. Tagliaferri, N. Brunetti-Pierri, A. Isacchi et al., "Discovery of drug mode of action and drug repositioning from transcriptional responses," *Proceedings of the National Academy of Sciences*, vol. 107, no. 33, pp. 14 621–14 626, 2010.

[182] W. Wang, S. Yang, X. Zhang, and J. Li, "Drug repositioning by integrating target information through a heterogeneous network model," *Bioinformatics*, vol. 30, no. 20, pp. 2923–2930, 2014.

[183] M. Sirota, J. T. Dudley, J. Kim, A. P. Chiang, A. A. Morgan, A. Sweet-Cordero, J. Sage, and A. J. Butte, "Discovery and preclinical validation of drug indications using compendia of public gene expression data," *Science translational medicine*, vol. 3, no. 96, pp. 96ra77–96ra77, 2011.

[184] F. Yang, J. Xu, and J. Zeng, "Drug-target interaction prediction by integrating chemical, genomic, functional and pharmacological data," in *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*. NIH Public Access, 2014, p. 148.

[185] X. Chen, M.-X. Liu, and G.-Y. Yan, "Drug–target interaction prediction by random walk on the heterogeneous network," *Molecular BioSystems*, vol. 8, no. 7, pp. 1970–1978, 2012.

[186] G. Fu, Y. Ding, A. Seal, B. Chen, Y. Sun, and E. Bolton, "Predicting drug target interactions using meta-path-based semantic network analysis," *BMC bioinformatics*, vol. 17, no. 1, p. 1, 2016.

[187] X. Zheng, H. Ding, H. Mamitsuka, and S. Zhu, "Collaborative matrix factorization with multiple similarities for predicting drug-target interactions," in *KDD*, 2013.

[188] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *ICDM*, 2006.

[189] H. Cho, B. Berger, and J. Peng, "Diffusion component analysis: Unraveling functional topology in biological networks," in *Research in Computational Molecular Biology*, ser. Lecture Notes in Computer Science, T. M. Przytycka, Ed. Springer International Publishing, 2015, vol. 9029, pp. 62–64. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16706-0_9

[190] H. Cho, B. Berger, and J. Peng, "Compact integration of multi-network topology for functional analysis of genes," *Cell Systems*, vol. 3, no. 6, pp. 540–548, 2016.

[191] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[192] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu, Y. Djoumbou, R. Eisner, A. Guo, and D. S. Wishart, "Drugbank 3.0: a comprehensive resource for 'omics' research on drugs," in *NAR*, 2011.

[193] T. S. K. Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, L. Balakrishnan, A. Marimuthu, S. Banerjee, D. S. Somanathan, A. Sebastian, S. Rani, S. Ray, C. J. H. Kishore, S. Kanth, M. Ahmed, M. K. Kashyap, R. Mohmood, Y. L. Ramachandra, V. Krishna, B. A. Rahiman, S. Mohan, P. Ranganathan, S. Ramabadran, R. Chaerkady, and A. Pandey, "Human protein reference database—2009 update," in *NAR*, 2009.

[194] A. P. Davis, C. G. Murphy, R. Johnson, J. M. Lay, K. Lennon-Hopkins, C. Saraceni-Richards, D. Sciaky, B. L. King, M. C. Rosenstein, T. C. Wiegers et al., "The comparative toxicogenomics database: update 2013," *Nucleic acids research*, vol. 41, no. D1, pp. D1104–D1114, 2013.

[195] M. Kuhn, M. Campillos, I. Letunic, L. J. Jensen, and P. Bork, "A side effect resource to capture phenotypic effects of drugs," *Molecular systems biology*, vol. 6, no. 1, p. 343, 2010.

[196] M. Kim and J. Leskovec, "The network completion problem: Inferring missing nodes and edges in networks." in *SDM*, vol. 11. SIAM, 2011, pp. 47–58.

[197] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[198] S. Wang, H. Cho, C. Zhai, B. Berger, and J. Peng, "Exploiting ontology graph for predicting sparsely annotated gene function," in *ISMB/ECCB*, 2015.

[199] N. Natarajan and I. S. Dhillon, "Inductive matrix completion for predicting gene–disease associations," *Bioinformatics*, vol. 30, no. 12, pp. i60–i68, 2014.

[200] H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon, "Large-scale multi-label learning with missing labels," in *ICML*, 2014.

[201] M. Hattori, Y. Okuno, S. Goto, and M. Kanehisa, "Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways," *Journal of the American Chemical Society*, vol. 125, no. 39, pp. 11 853–11 865, 2003.

[202] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.

[203] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic acids research*, vol. 28, no. 1, pp. 235–242, 2000.

[204] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, and R. G. Coleman, "Zinc: a free tool to discover chemistry for biology," *Journal of chemical information and modeling*, vol. 52, no. 7, pp. 1757–1768, 2012.

[205] J. Liu, S. Liu, C. Tanabe, T. Maeda, K. Zou, and H. Komano, "Differential effects of angiotensin II receptor blockers on aβ generation," *Neuroscience letters*, vol. 567, pp. 51–56, 2014.

[206] M. Tsubaki, T. Satou, T. Itoh, M. Imano, M. Yanae, C. Kato, R. Takagoshi, M. Komai, and S. Nishida, "Bisphosphonate-and statin-induced enhancement of OPG expression and inhibition of CD9, M-CSF, and RANKL expressions via inhibition of the Ras/MEK/ERK pathway and activation of p38MAPK in mouse bone marrow stromal cell line ST2," *Molecular and cellular endocrinology*, vol. 361, no. 1, pp. 219–231, 2012.

[207] J. A. Durr, J. Hensen, T. Ehnis, M. S. Blankenship et al., "Chlorpropamide upregulates antidiuretic hormone receptors and unmasks constitutive receptor signaling," *American Journal of Physiology-Renal Physiology*, vol. 278, no. 5, pp. F799–F808, 2000.

[208] E. E. Aeberhard, S. A. Henderson, N. S. Arabolos, J. M. Griscavage, F. E. Castro, C. T. Barrett, and L. J. Ignarro, "Nonsteroidal anti-inflammatory drugs inhibit expression of the inducible nitric oxide synthase gene," *Biochemical and biophysical research communications*, vol. 208, no. 3, pp. 1053–1059, 1995.

[209] M. Rosenstock, A. Danon, and G. Rimon, "Pghs-2 inhibitors, ns-398 and dup-697, attenuate the inhibition of pghs-1 by aspirin and indomethacin without altering its activity," *Biochimica et Biophysica Acta (BBA)-Molecular and Cell Biology of Lipids*, vol. 1440, no. 1, pp. 127–137, 1999.

[210] K. M. Stuhlmeier, H. Li, and J. J. Kao, "Ibuprofen: new explanation for an old phenomenon," *Biochemical pharmacology*, vol. 57, no. 3, pp. 313–320, 1999.

[211] J. K. Gierse, S. D. Hauser, D. P. Creely, C. Koboldt, S. H. Rangwala, P. C. Isakson, and K. Seibert, "Expression and selective inhibition of the constitutive and inducible forms of human cyclo-oxygenase," *Biochemical Journal*, vol. 305, no. 2, pp. 479–484, 1995.

[212] W. F. Hood, J. K. Gierse, P. C. Isakson, J. R. Kiefer, R. G. Kurumbail, K. Seibert, and J. B. Monahan, "Characterization of celecoxib and valdecoxib binding to cyclooxygenase," *Molecular pharmacology*, vol. 63, no. 4, pp. 870–877, 2003.

[213] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[214] U. M. Singh-Blom, N. Natarajan, A. Tewari, J. O. Woods, I. S. Dhillon, and E. M. Marcotte, "Prediction and validation of gene-disease associations using methods inspired by social network analyses," *PloS one*, vol. 8, no. 5, p. e58977, 2013.

[215] Y. Wu, M. Blichowski, Z. J. Daskalakis, Z. Wu, C. C. Liu, M. A. Cortez, and O. C. Snead III, "Evidence that clozapine directly interacts on the gabab receptor," *Neuroreport*, vol. 22, no. 13, pp. 637–641, 2011.

[216] A. Wassef, J. Baker, and L. D. Kochan, "Gaba and schizophrenia: a review of basic science and clinical studies," *Journal of clinical psychopharmacology*, vol. 23, no. 6, pp. 601–640, 2003.

[217] K. Uefuji, T. Ichikura, and H. Mochizuki, "Cyclooxygenase-2 expression is related to prostaglandin biosynthesis and angiogenesis in human gastric cancer," *Clinical cancer research*, vol. 6, no. 1, pp. 135–138, 2000.

[218] P. Rao and E. E. Knaus, "Evolution of nonsteroidal anti-inflammatory drugs (nsaids): cyclooxygenase (cox) inhibition and beyond," *Journal of Pharmacy & Pharmaceutical Sciences*, vol. 11, no. 2, pp. 81–110s, 2008.

[219] L. Minghetti, "Cyclooxygenase-2 (COX-2) in inflammatory and degenerative brain diseases," *Journal of Neuropathology & Experimental Neurology*, vol. 63, no. 9, pp. 901–910, 2004.

[220] P. M. Kearney, C. Baigent, J. Godwin, H. Halls, J. R. Emberson, and C. Patrono, "Do selective cyclo-oxygenase-2 inhibitors and traditional non-steroidal anti-inflammatory drugs increase the risk of atherothrombosis? meta-analysis of randomised trials," *Bmj*, vol. 332, no. 7553, pp. 1302–1308, 2006.

[221] S. Trelle, S. Reichenbach, S. Wandel, P. Hildebrand, B. Tschannen, P. M. Villiger, M. Egger, and P. Jüni, "Cardiovascular safety of non-steroidal anti-inflammatory drugs: network meta-analysis," *Bmj*, vol. 342, p. c7086, 2011.

[222] P. Gosse, "A review of telmisartan in the treatment of hypertension: blood pressure control in the early morning hours," *Vascular health and risk management*, vol. 2, no. 3, p. 195, 2006.

[223] B. Clarke and L. Duncan, "Comparison of chlorpropamide and metformin treatment on weight and blood-glucose response of uncontrolled obese diabetics," *The Lancet*, vol. 291, no. 7534, pp. 123–126, 1968.

[224] M. L. Bianchi, R. Cimaz, M. Bardare, F. Zulian, L. Lepore, A. Boncompagni, E. Galbiati, F. Corona, G. Luisetto, D. Giuntini et al., "Efficacy and safety of alendronate for the treatment of osteoporosis in diffuse connective tissue diseases in children," *Arthritis Rheum*, vol. 43, no. 9, pp. 1960–1966, 2000.

[225] L. A. DiMeglio and M. Peacock, "Two-year clinical trial of oral alendronate versus intravenous pamidronate in children with osteogenesis imperfecta," *Journal of Bone and Mineral Research*, vol. 21, no. 1, pp. 132–140, 2006.

[226] G. Rimon, R. S. Sidhu, D. A. Lauver, J. Y. Lee, N. P. Sharma, C. Yuan, R. A. Frieler, R. C. Trievel, B. R. Lucchesi, and W. L. Smith, "Coxibs interfere with the action of aspirin by binding tightly to one monomer of cyclooxygenase-1," *Proceedings of the National Academy of Sciences*, vol. 107, no. 1, pp. 28–33, 2010.

[227] A. J. Vecchio and M. G. Malkowski, "The structure of NS-398 bound to cyclooxygenase-2," *Journal of structural biology*, vol. 176, no. 2, pp. 254–258, 2011.

[228] Schrödinger, LLC, "The PyMOL molecular graphics system, version 1.8," November 2015.

[229] S. Kargman, E. Wong, G. M. Greig, J.-P. Falgueyret, W. Cromlish, D. Ethier, J. A. Yergey, D. Riendeau, J. F. Evans, B. Kennedy et al., "Mechanism of selective inhibition of human prostaglandin g/h synthase-1 and-2 in intact cells," *Biochemical pharmacology*, vol. 52, no. 7, pp. 1113–1125, 1996.

[230] M. Kato, S. Nishida, H. Kitasato, N. Sakata, and S. Kawai, "Cyclooxygenase-1 and cyclooxygenase-2 selectivity of non-steroidal anti-inflammatory drugs: investigation using human peripheral monocytes," *Journal of Pharmacy and Pharmacology*, vol. 53, no. 12, pp. 1679–1685, 2001.

[231] S. Ariasnegrete, K. Keller, and K. Chadee, "Proinflammatory cytokines regulate cyclooxygenase-2 mRNA expression in human macrophages," *Biochemical and biophysical research communications*, vol. 208, no. 2, pp. 582–589, 1995.

[232] S. Mehmood, J. Marcoux, J. Gault, A. Quigley, S. Michaelis, S. G. Young, E. P. Carpenter, and C. V. Robinson, "Mass spectrometry captures off-target drug binding and provides mechanistic insights into the human metalloprotease zmpste24," *Nature Chemistry*, 2016.

[233] H. Franken, T. Mathieson, D. Childs, G. M. Sweetman, T. Werner, I. Tögel, C. Doce, S. Gade, M. Bantscheff, G. Drewes et al., "Thermal proteome profiling for unbiased identification of direct and indirect drug targets using multiplexed quantitative mass spectrometry," *Nature protocols*, vol. 10, no. 10, pp. 1567–1593, 2015.

[234] A. Chernobrovkin, C. Marin-Vicente, N. Visa, and R. A. Zubarev, "Functional identification of target by expression proteomics (fitexp) reveals protein targets and highlights mechanisms of action of small molecule drugs," *Scientific reports*, vol. 5, p. 11176, 2015.

[235] M. M. Savitski, F. B. Reinhard, H. Franken, T. Werner, M. F. Savitski, D. Eberhard, D. M. Molina, R. Jafari, R. B. Dovega, S. Klaeger et al., "Tracking cancer drugs in living cells by thermal profiling of the proteome," *Science*, vol. 346, no. 6205, p. 1255784, 2014.

[236] E. Guney, J. Menche, M. Vidal, and A.-L. Barábasi, "Network-based in silico drug efficacy screening," *Nature communications*, vol. 7, 2016.

[237] F. Cheng, J. Zhao, M. Fooksa, and Z. Zhao, "A network-based drug repositioning infrastructure for precision cancer medicine through targeting significantly mutated genes in the human cancer genomes," *Journal of the American Medical Informatics Association*, 2016.

[238] S. C. Benson, H. A. Pershadsingh, C. I. Ho, A. Chittiboyina, P. Desai, M. Pravenec, N. Qi, J. Wang, M. A. Avery, and T. W. Kurtz, "Identification of telmisartan as a unique angiotensin ii receptor antagonist with selective pparγ–modulating activity," *Hypertension*, vol. 43, no. 5, pp. 993–1002, 2004.

[239] K. Sato, T. Yamashita, T. Kurata, Y. Fukui, N. Hishikawa, K. Deguchi, and K. Abe, "Telmisartan ameliorates inflammatory responses in shr-sr after tmcao," *Journal of Stroke and Cerebrovascular Diseases*, vol. 23, no. 10, pp. 2511–2519, 2014.

[240] T. V. Lanz, Z. Ding, P. P. Ho, J. Luo, A. N. Agrawal, H. Srinagesh, R. Axtell, H. Zhang, M. Platten, T. Wyss-Coray et al., "Angiotensin ii sustains brain inflammation in mice via tgf-β," *The Journal of clinical investigation*, vol. 120, no. 8, pp. 2782–2794, 2010.

[241] F. J. Giles, "The emerging role of angiogenesis inhibitors in hematologic malignancies," *ONCOLOGY-WILLISTON PARK THEN HUNTINGTON THE MELVILLE NEW YORK-*, vol. 16, no. 5; SUPP/4, pp. 23–29, 2002.

[242] M. M. Lubran, "Hematologic side effects of drugs," *Annals of Clinical & Laboratory Science*, vol. 19, no. 2, pp. 114–121, 1989.

[243] A. Töyräs, J. Ollikainen, M. Taskinen, and J. Mönkkönen, "Inhibition of mevalonate pathway is involved in alendronate-induced cell growth inhibition, but not in cytokine secretion from macrophages in vitro," *European journal of pharmaceutical sciences*, vol. 19, no. 4, pp. 223–230, 2003.

Appendix A: Supplementary Tables

Table A.1: The number of nodes of individual types in the constructed heterogeneous network.

Type of node	Count
Drug	708
Protein	1,512
Disease	5,603
Side-effect	4,192
Total	12,015

Table A.2: The size of individual interaction or association matrices in the constructed heterogeneous network.

Type of edge	Count
Drug-Protein	1,923
Drug-Drug	10,036
Drug-Disease	199,214
Drug-Side-effect	80,164
Protein-Protein	7,363
Protein-Disease	1,596,745
Total	1,895,445

Table A.3: Statistics of Davis and KIBA datasets. "Raw" refers to the raw dataset and "Processed" refers to the version with kinases and compounds without 3D structure being removed.

(a) Davis

	Proteins	Compounds	Pairs
Raw	442	68	30,056
Processed	226	68	14,464

(b) KIBA

	Proteins	Compounds	Pairs
Raw	229	2,111	118,254
Processed	160	2,086	8,9957

Table A.4: Node features of atoms in molecules. Features are encoded using one-hot encoding.

Index	Feature	Possible values	Dimension
1	Atom type	C, N, O, S, F, Si, P, Cl, Br, Mg, Na, Ca, Fe, As, Al, I, B, V, K, Tl, Yb, Sb, Sn, Ag, Pd, Co, Se, Ti, Zn, H, Li, Ge, Cu, Au, Ni, Cd, In, Mn, Zr, Cr, Pt, Hg, Pb, unknown	44
2	Atom's degree	0,1,...,5, ≥ 6	7
3	Number of H bound to the atom	0,1,...,5, ≥ 6	7
4	Number of implicit H bound to the atom	0,1,...,5, ≥ 6	7
5	Whether the atom is aromatic or not	True of False	1
	Total		66

Table A.5: Raw performance numbers on the Envision dataset. Performances of ECNet and the Envision model are evaluated using AUROC and Spearman correlation. The mean values of ten trials of five-fold cross-validation are listed.

	AUROC		Spearman correlation	
Protein	ECNet	Envision	ECNet	Envision
BRCA1 (E3 activity)	0.32	0.29	0.41	0.33
BRCA1 (BARD1 binding)	0.54	0.52	0.43	0.38
Ubiquitin (E1)	0.59	0.53	0.72	0.71
UBE4B (U-box)	0.48	0.41	0.52	0.48
PSD95 (pdz3)	0.86	0.82	0.79	0.75
Pab1	0.44	0.36	0.81	0.80
TEM-1	0.59	0.55	0.86	0.87
Ubiquitin	0.73	0.65	0.86	0.82
Yap65 (WW)	0.79	0.74	0.76	0.72
Protein G	0.88	0.83	0.91	0.87
HSP90	0.56	0.52	0.70	0.67
KKA2	0.80	0.88	0.80	0.88

Appendix B: Supplementary Figures

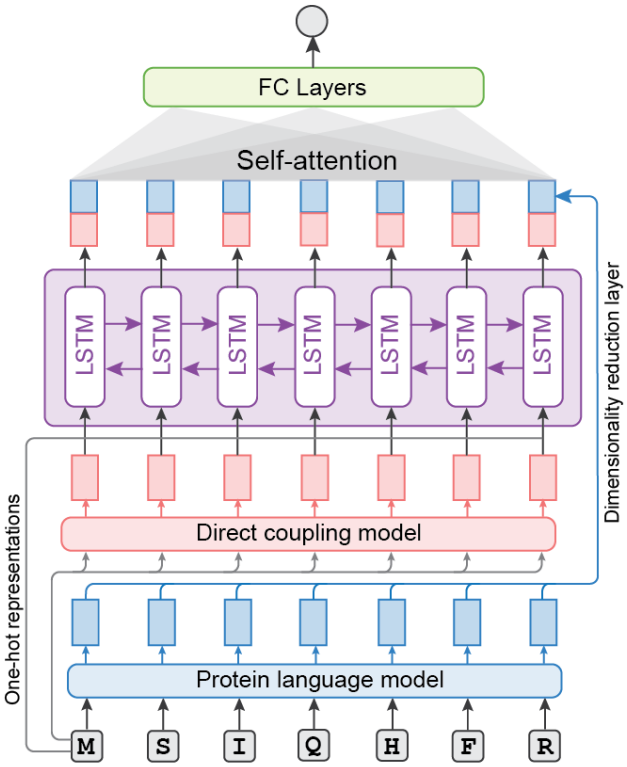


Figure B.1: The neural network architecture of ECNet. Given the input sequence, the protein language model is used to generate the global evolutionary contexts while the direct coupling model is used to generate the local evolutionary contexts. One-hot representations and the local evolutionary contexts are concatenated and passed to an LSTM. Embeddings produced by the LSTM are concatenated with global evolutionary contexts that have been projected by a linear dimensionality reduction layer. The top layers are composed of a self-attention layer, which summarizes the embeddings of all positions into a single embedding, and fully connected layers that output the final prediction. In the model training, only parameters of LSTM and top layers are updated, while parameters of the protein language model and direct coupling model are fixed. (LSTM: long short-term memory network; FC layers: fully-connected layers.)

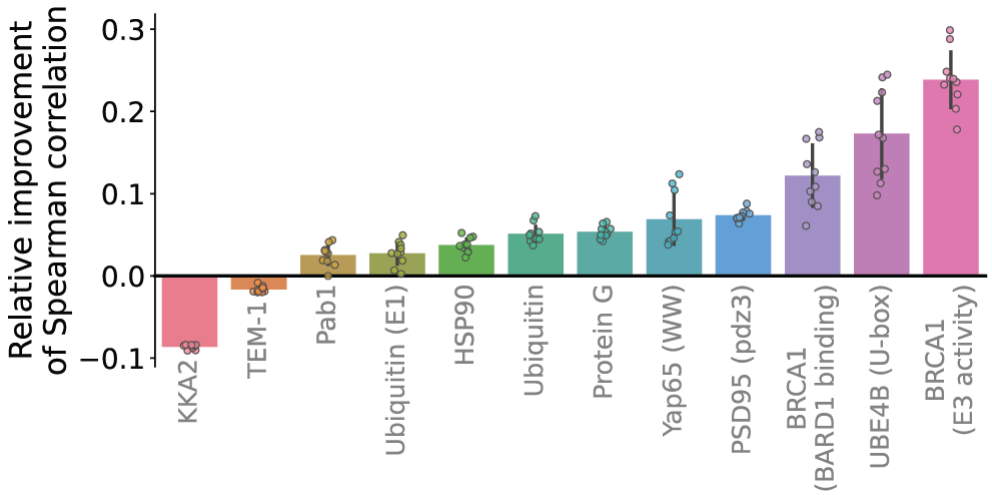


Figure B.2: Performance improvements on the Envision dataset. This bar plot shows the relative improvements of Spearman correlation achieved by ECNet as compared to the Envision model. Performances were evaluated using ten trials of five-fold cross-validation. The bar plot represented the mean±SD of the data.

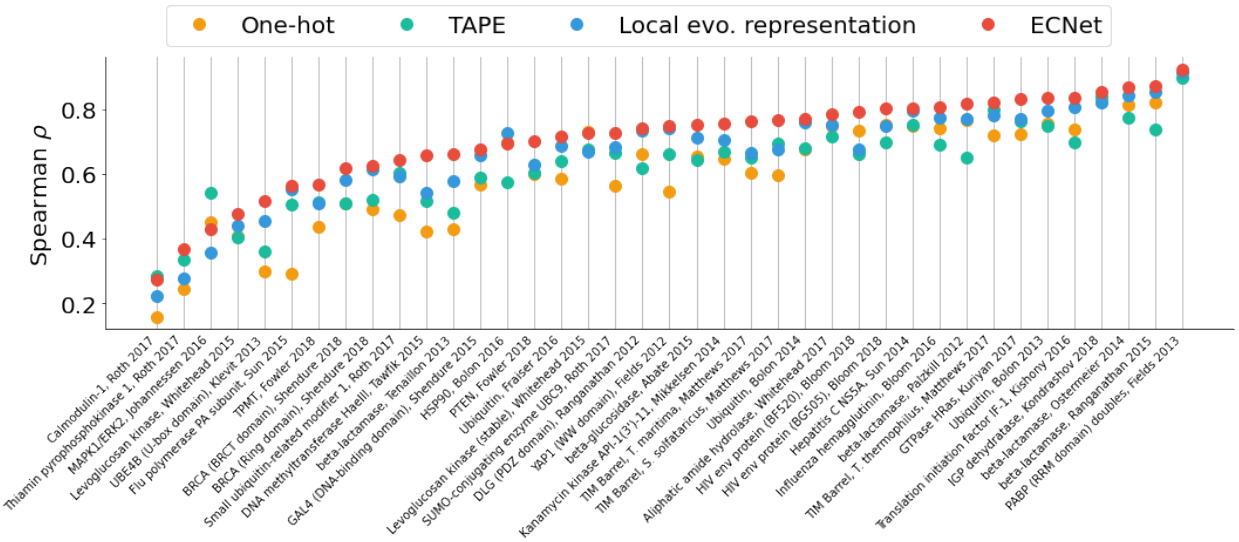


Figure B.3: Comparison of supervised learning with different representations. ECNet integrated multi-scale sequence representations, including the global evolutionary representation, the local evolutionary representation, and the one-hot representation of a protein sequence. We performed ablation analysis to compare ECNet to models that used individual sequence representation using five-fold cross-validation. Spearman correlation was used as the evaluation metric.

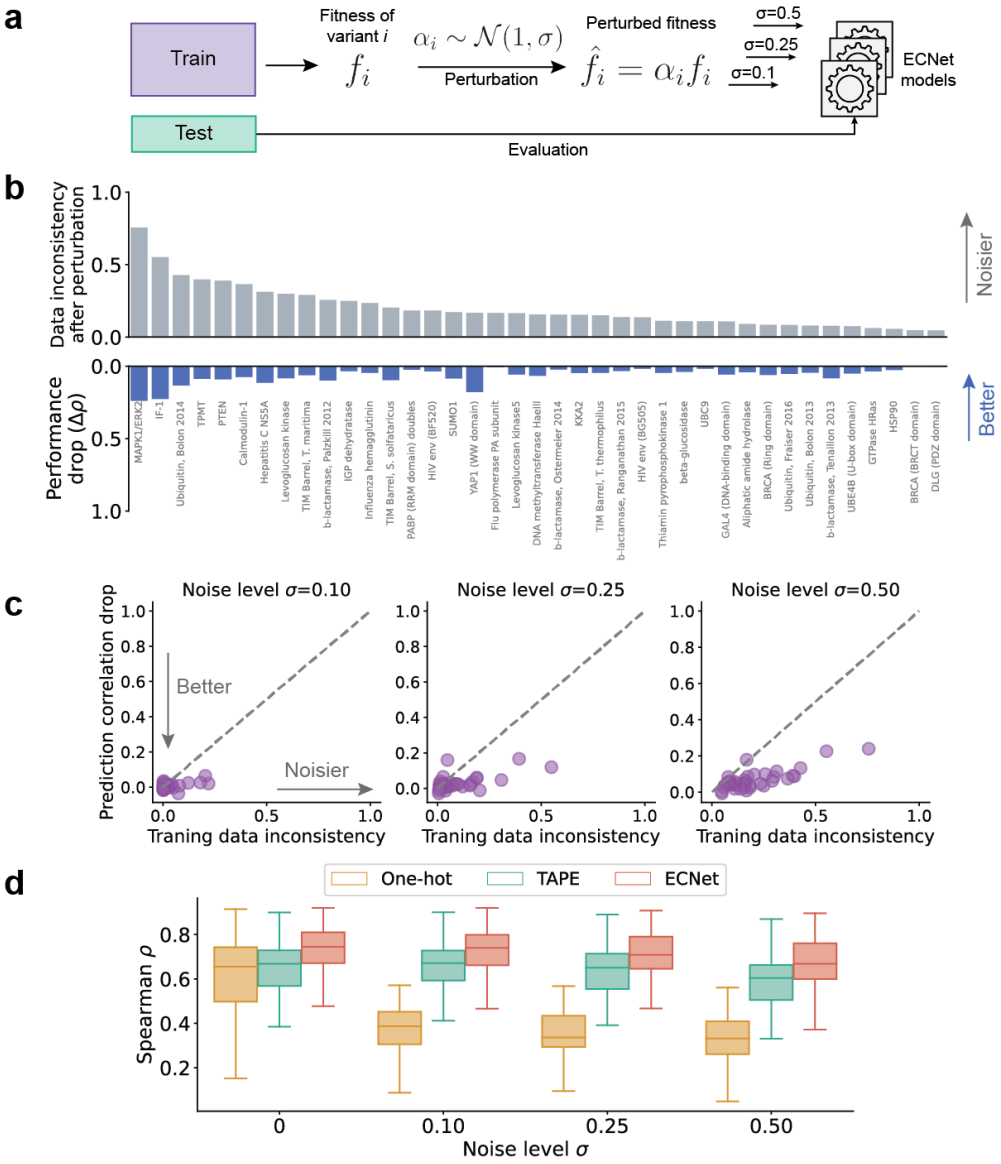


Figure B.4: Evaluation of ECNet on noisy data. (a) Schematic illustration of the evaluation protocol using simulated noisy data. A random fraction (20%) of the DMS dataset was withheld as the test set and the remaining variants are used as the training set. For each variant in the training set, its fitness value is perturbed by a multiplicative random noise α, i.e., the fitness value f_i of variant i became $\hat{f}_i = \alpha_i f_i$, where α_i was sampled from the normal distribution $\mathcal{N}(1, \sigma)$. The trained models were then evaluated on the noise-free test set. (b) Evaluation results for 39 proteins in the DeepSequence dataset for $\sigma = 0.5$. Top: data inconsistency caused by the perturbation for each protein. The inconsistency is defined as the $1 - r$, where r is the Spearman correlation between the pre- and post-perturbation fitness values of the training set. Bottom: ECNet's performance drop when trained on the noisy training data as compared to when it is trained on the noise-free training data. The performance was evaluated using Spearman correlation. (c) Relationship between the training data inconsistency and the performance drop for every protein in the DeepSequence dataset, at noise levels $\sigma = 0.1, 0.25$,and 0.5. (d) Prediction performances of ECNet, TAPE, and One-hot models on the DeepSequence dataset at noise levels $\sigma = 0, 0.1, 0.25$,and 0.5.

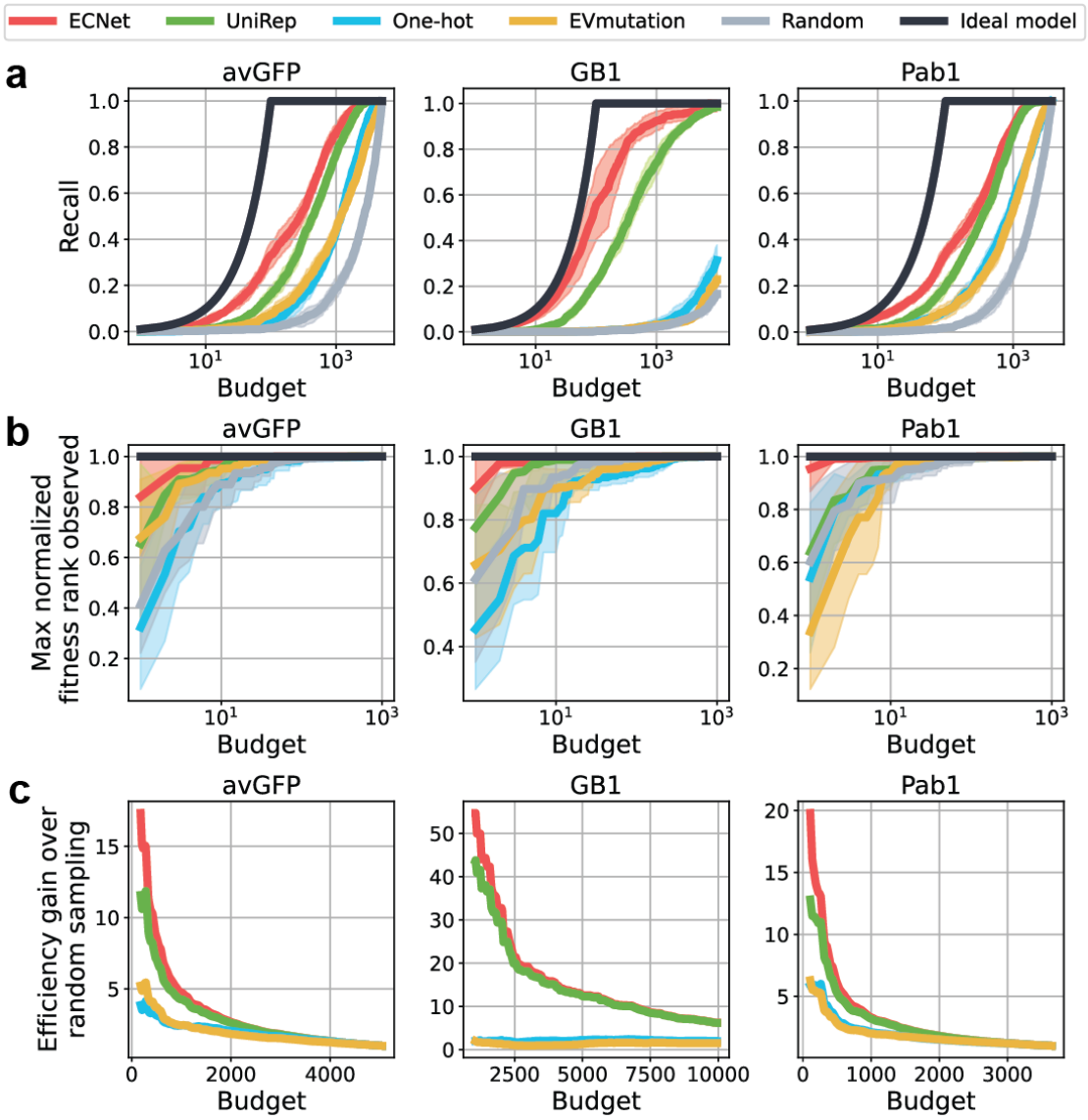


Figure B.5: Simulation results of using ECNet to prioritize high-performing variants for avGFP, GB1, and Pab1. (a) Recall versus sequence testing budget curves for each method. The recall is defined as the fraction of the true top-100 variants that were ranked within the top K predictions of a method, where K is a given testing budget (number of variants to test). ECNet was compared to i) UniRep, a supervised method, ii) One-hot, a supervised method that uses simple one-hot sequence representations, iii) EVmutation, an unsupervised method, iv) random model, which is a null model that assigns a random ranking to test variants, and v) ideal model, which ranks the variants using the ground-truth fitness score. (b) Maximum normalized fitness rank observed versus sequence testing budget curves for each method. Fitness scores of variants were normalized based on their rank into a value between 0 (the lowest fitness score) and 1 (the highest fitness score). (c) Efficiency gain of ECNet, UniRep, One-hot, and EVmutation over the random model with the given testing budget. The efficiency gain is defined as the ratio of a method's recall divided by the recall of the null model as a function of the testing budget. Error bands in (a) and (b) depict mean ± SD calculated over 10 independent replicates of the experiments. Curves in (c) were smoothed using an averaging window of size 50 along the x-axis.

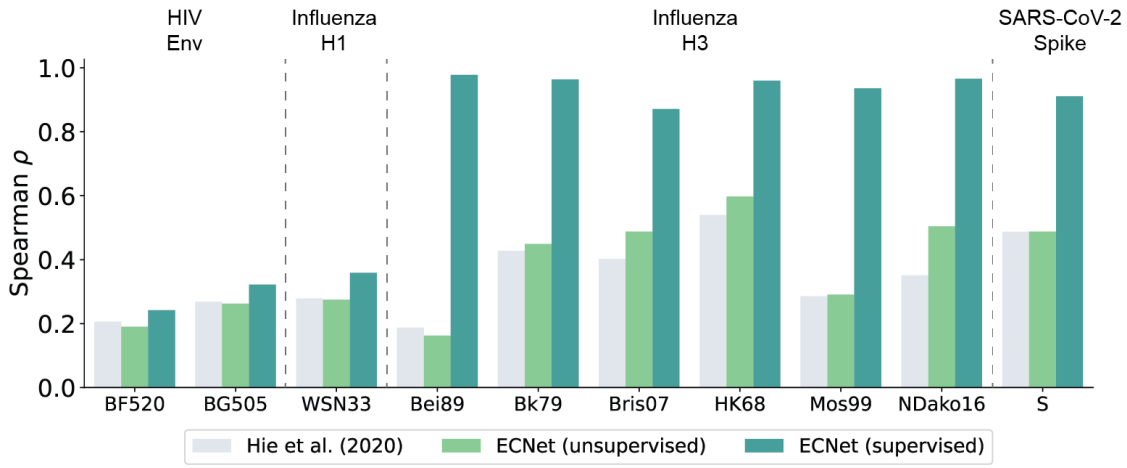


Figure B.6: Performance of unsupervised and supervised ECNet on viral proteins. The unsupervised ECNet is a model that learns from homologous sequences of the protein of interest to predict how tolerable or favorable a mutation is at a position, and it does not use any fitness data in the training process. The supervised ECNet model predicted the fitness value of the input sequence and was trained using existing fitness data. The performance of supervised ECNet was summarized as the average of a five-fold cross-validation.

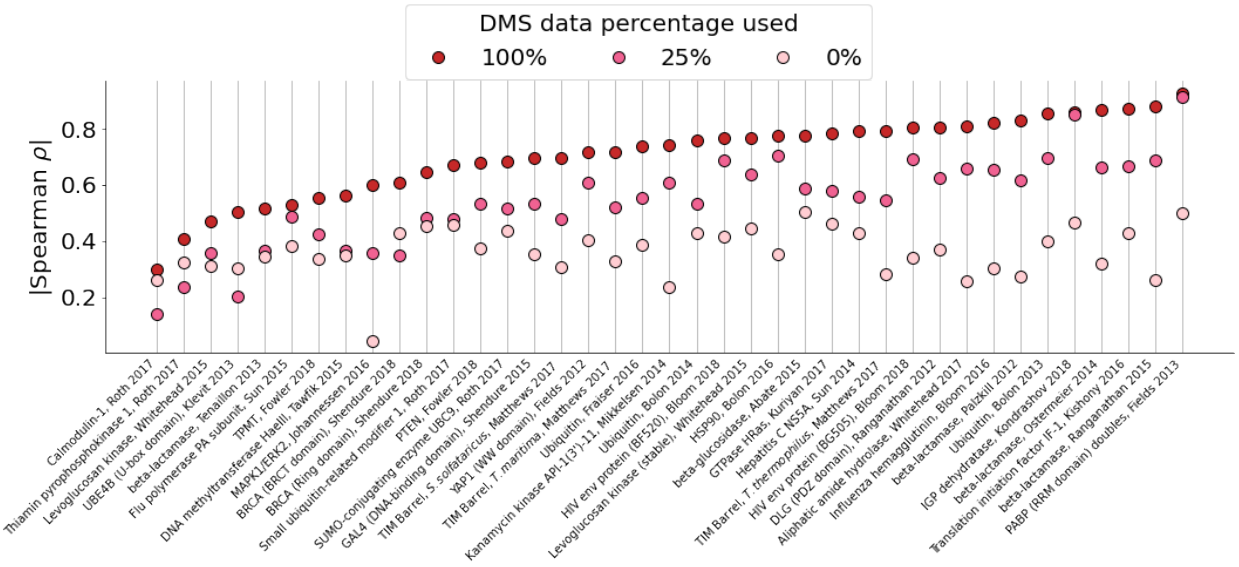


Figure B.7: Effect of DMS training data size on prediction performance. ECNet was trained in a supervised way with full (100%) or partial (25%) DMS data, or in an unsupervised way (0% DMS data), and then applied to predict the fitness for a holdout set of variants. The supervised ECNet model was trained using 100% or 25% DMS data, and three model replicas are trained on the same data and their predictions were averaged as the final predictions. The unsupervised ECNet model was trained on homologous sequences of the target protein to predict the probability of an amino acid showing at a position in the sequence (Methods). Performances were evaluated on the proteins in the DeepSequence dataset (Methods).

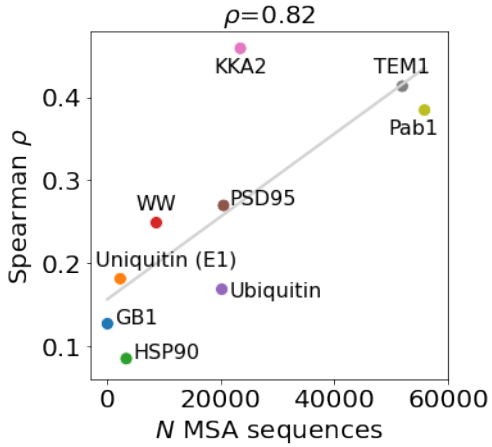


Figure B.8: Relationships between the number of sequences in MSA and the fitness prediction accuracy using co-variation.

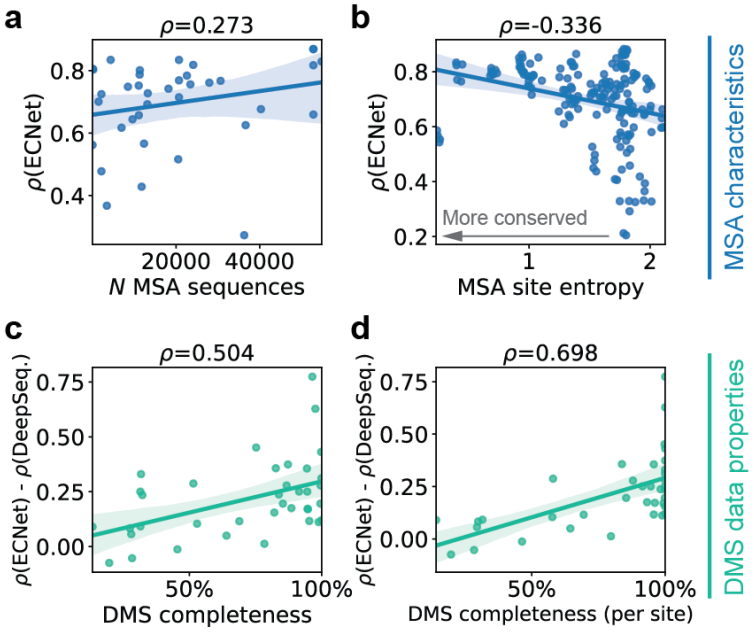


Figure B.9: Relationships between prediction performance and MSA/DMS data properties. (a) Correlation between the number of sequences in the MSA of homologous sequences and ECNet's prediction performance. Each point represents one protein in the DeepSequence dataset. (b) Correlation between site entropy in the MSA and ECNet's prediction performance. Each point represents one fold experiment, and its x-axis value indicates the average over the entropies of all sites in the test data in this fold. (c) Correlation between ECNet's performance improvements against DeepSequence and the completeness of single-mutation protein DMS data in the DeepSequence dataset. The completeness is defined as the percentage of screened variants among all possible single-mutation variants for a protein. (d) Same as (c) except that the DMS completeness is defined for each site.

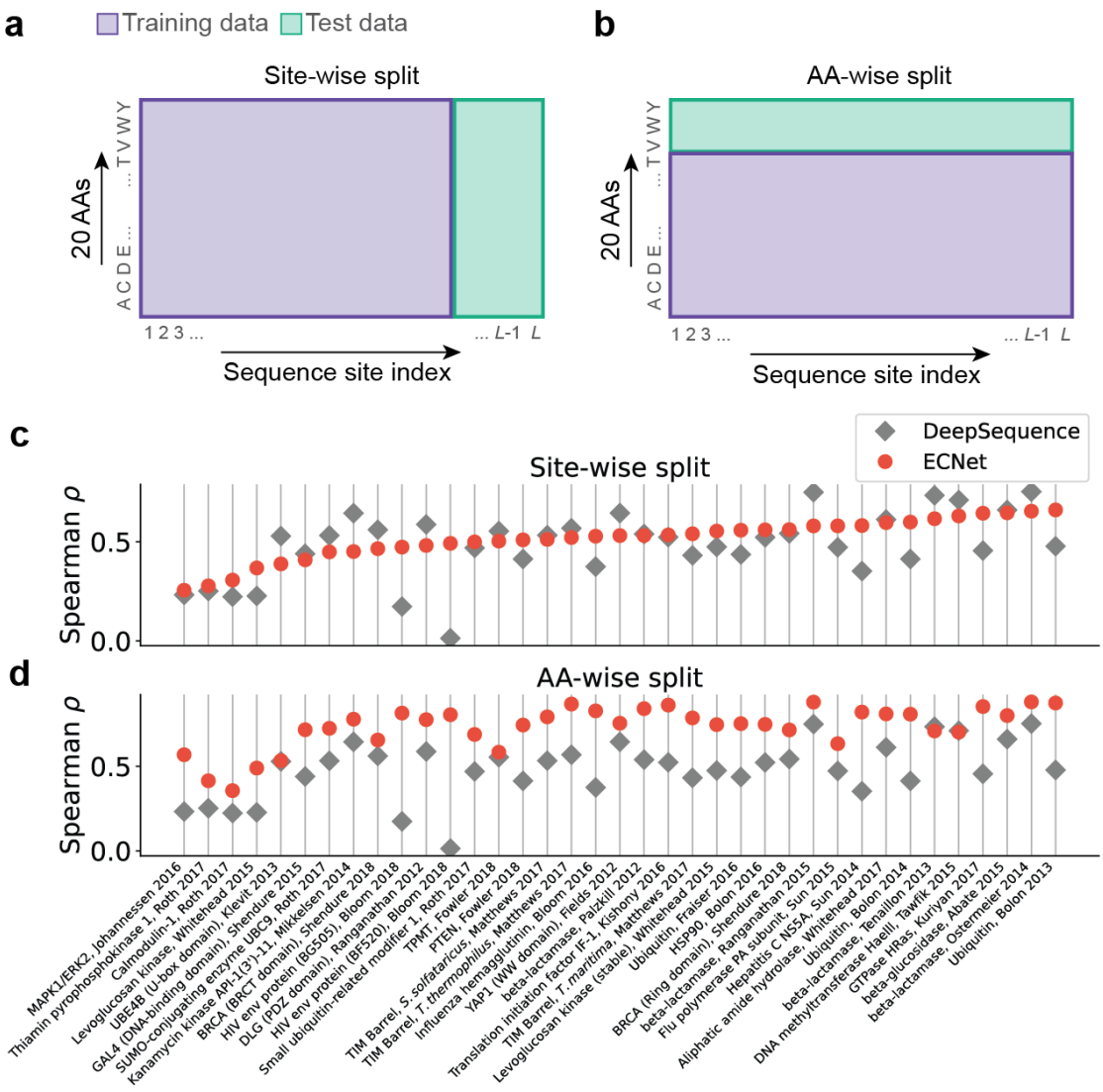


Figure B.10: Prediction performance on sequence site-wise and amino acid-wise (AA-wise) train/test data split. (a-b) Schematic visualizations of train/test split. A DMS dataset is visualized as a matrix where the x-axis represents the index of position in the sequence and the y-axis represents the amino acid (AA) types the site mutates to. To split strategies are considered: (a) Site-wise split: the deep mutational scanning (DMS) dataset was split based on the sequence position (site) in the sequence. Mutants in 80% of the sites were randomly sampled as training data and mutants in the remaining sites were used as test data. The partition of train/test sites was only for the schematic visualization purpose. The actual training sites are randomly sampled and not necessarily the 80% leftmost sites; (b) AA-wise split: the DMS dataset was split based on AA types of mutations. For each site, 80% of the mutations were randomly sampled and added to the training set and the remaining 20% mutations were added to the test set. The partition of train/test AA types was only for the schematic visualization purpose. The actual training mutations are randomly sampled and not necessarily the first 80% alphabetically ordered AA types. (c-d) Comparison of ECNet to DeepSequence on 37 single-mutation DMS datasets for both (c) site-wise split and (d) AA-wise split settings. The DeepSequence's performances are the same in (c) and (d) as it is an unsupervised model. (AA: amino acid; DMS: deep mutational scanning.)

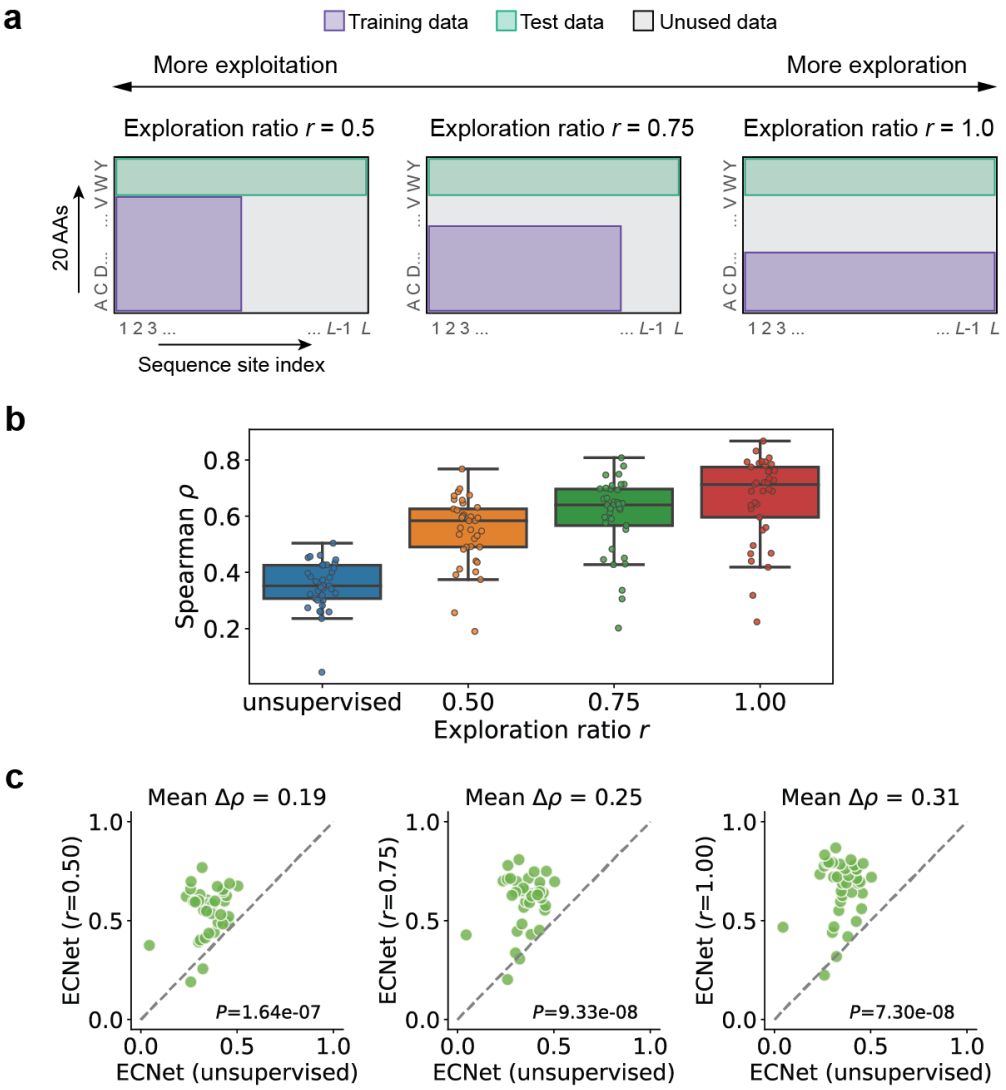


Figure B.11: The exploration-exploitation trade-off of training data design under a limited experimental budget. Given the limited experimental budget, we examined the effects of different training data designs, i.e., screening fewer sites but more mutations on each site, or more sites but fewer mutations on each site. (a) Schematic visualizations of the exploration-exploitation trade-off. Given a DMS dataset, 20% of mutations of each site were withheld as test data. The experimental budget (number of variants that can be tested) was set to 50% of the remaining variants. The allocation of the budget was controlled by an exploration ratio r, which we defined as the fraction of sites with at least one mutation being sampled in the training data. We varied the value of r =0.5, 0.75, and 1.0 but fixed the test budget (i.e., the area of the purple region remains the same). (b) ECNet's performance was assessed using 37 single-mutation DMS datasets curated in the DeepSequence dataset. (c) Pairwise performance comparison between ECNet at different exploration ratios and the unsupervised ECNet.

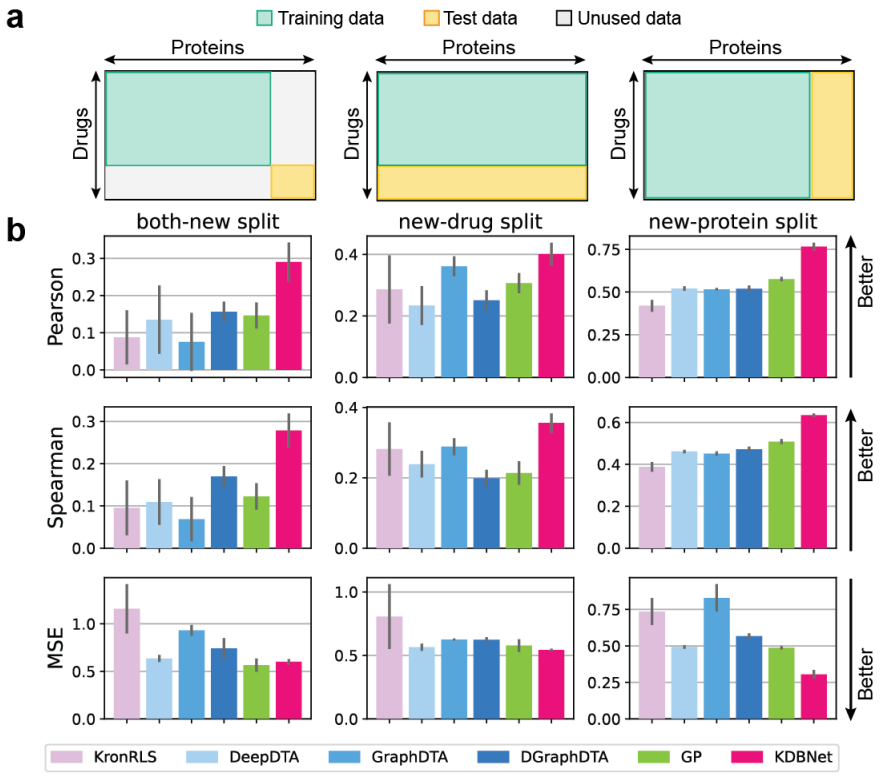


Figure B.12: Prediction performance evaluation on DAVIS dataset. Comparisons of prediction performance of KDBNet with KronRLS, DeepDTA, GraphDTA, and DGraphDTA on the DAVIS dataset using three train-test split settings. Performances were evaluated using three metrics, including Pearson correlation, Spearman correlation, and mean squared error (MSE) between predicted and true pK_d values. Results were averaged over 25 independent trials. The bar plot represented the mean±SEM of the data. Pearson and MSE results replicate Figure 3.2 for reference.

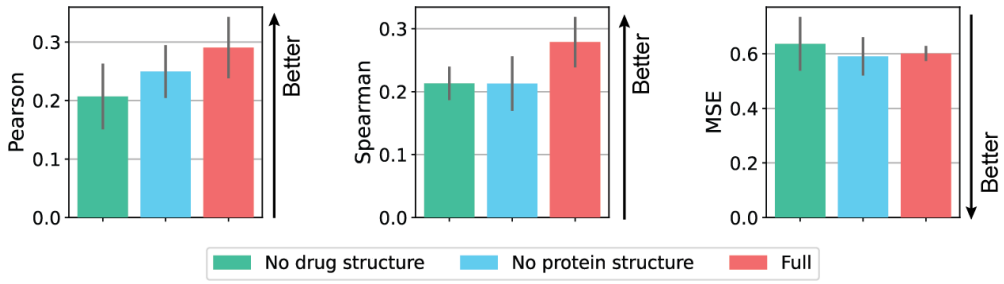


Figure B.13: Ablation evaluation of structure data. Comparisons between KDBNet variants that use or do not use 3D structure data. When protein structure was not used, the sequence was used as the representation of the input protein, and the protein GNN was replaced by a CNN. When drug structure is not used, the 2D molecule graph parsed from SMILES string was used as the representation of the input drug, and no 3D geometric features is used in the molecule GNN. The full model used both drug and protein structures.

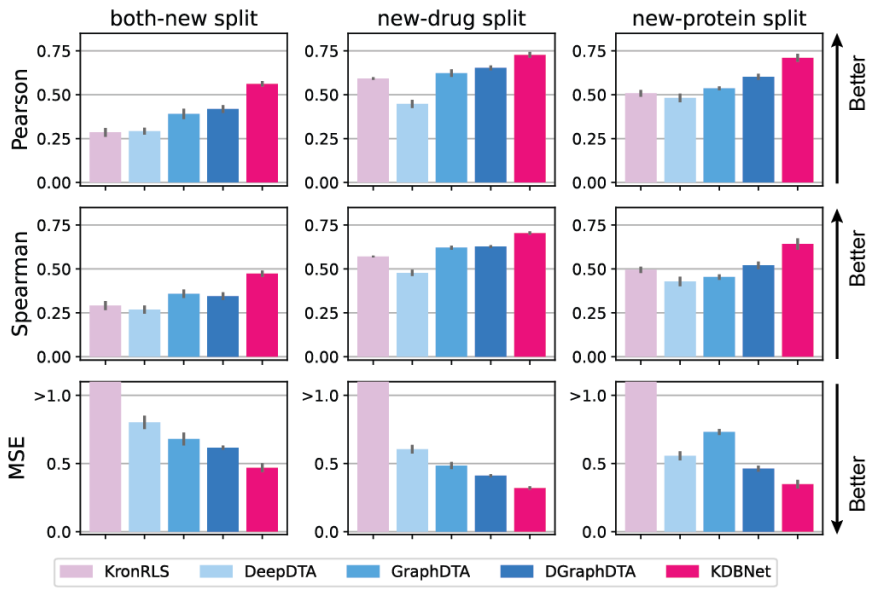


Figure B.14: Prediction performance evaluation on KIBA dataset. Comparisons of prediction performance of KDBNet with KronRLS, DeepDTA, GraphDTA, and DGraphDTA on the KIBA dataset using three train-test split settings. Performances were evaluated using three metrics, including Pearson correlation, Spearman correlation, and mean squared error (MSE) between predicted and true pK_d values. Results were averaged over 25 independent trials. The bar plot represented the mean±SEM of the data.

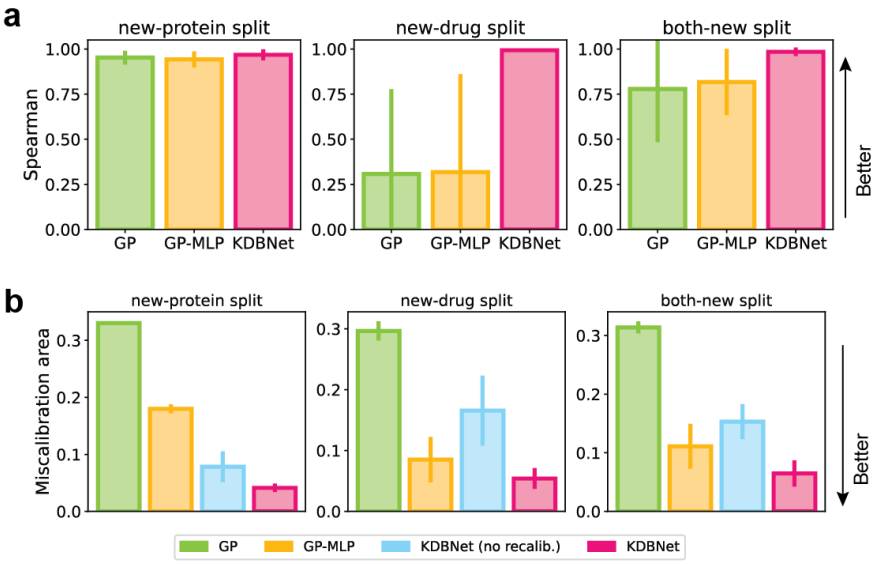


Figure B.15: Evaluation of uncertainty quantification using Spearman rank correlation and miscalibration area. (**a**) Spearman correlation between the estimated uncertainty and the prediction error measured in MSE on the three test sets of different split strategies. Higher correlations indicate more accurate uncertainty estimations with respect to prediction errors. (**b**) Miscalibration areas of KDBNet, GP, and GP-MLP on the three test sets. Lower values indicate more calibrated uncertainty estimations.

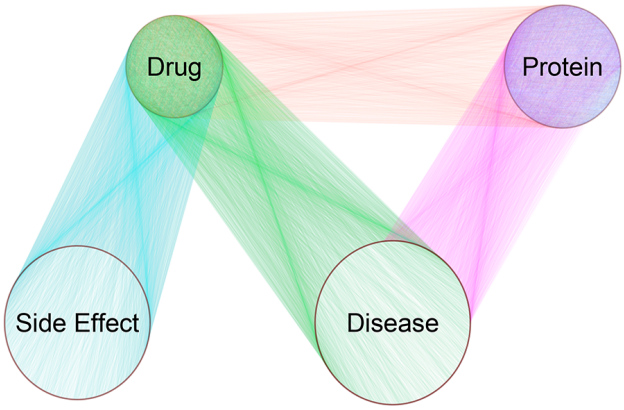
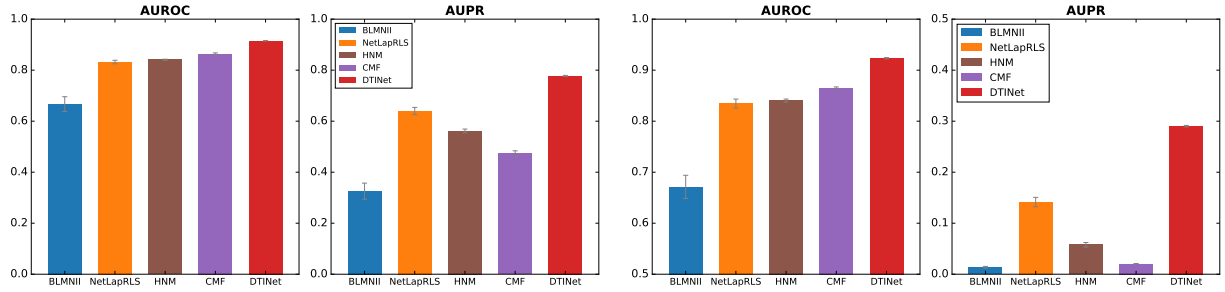


Figure B.16: The schema of the heterogeneous network constructed based on diverse data sources for DTINet development. Each edge between different types of nodes represents the pairwise interactions or associations between two nodes of the corresponding types (e.g., drug-protein interaction or protein-disease association). Each edge of a homogeneous drug (or protein) network represents the similarity between two drugs (or proteins) or their interaction.



(a) **(b)**

Figure B.17: Performance comparisons between DTINet and other state-of-the-art methods on skewed datasets. (a) The number of randomly chosen non-interacting drug-target pairs (i.e., negative samples) was 10 times more than the number of known interacting drug-target pairs (i.e., positive samples). (b) The negative set include all the remaining non-interacting drug-target pairs that were not in the training data. Here, during the training process, a randomly chosen subset of 90% known interactions and a matching number of non-interacting pairs were used to train the models, while during the test process, the remaining 10% known interactions and all of the non-interacting pairs that were not included in the training set were used to evaluate the prediction performance. All results were summarized over 10 trials of ten-fold cross-validation and expressed as mean ± SD.

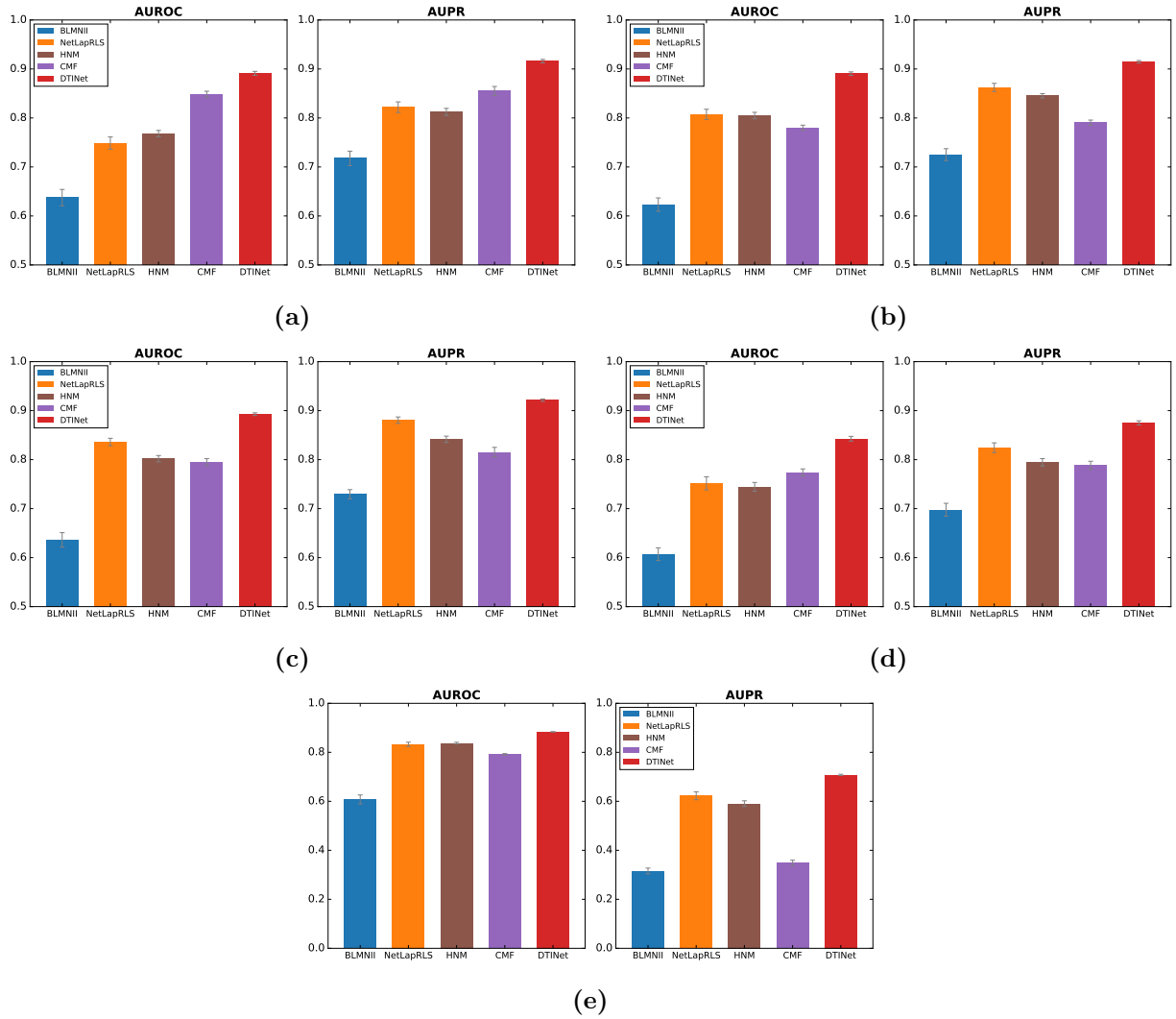
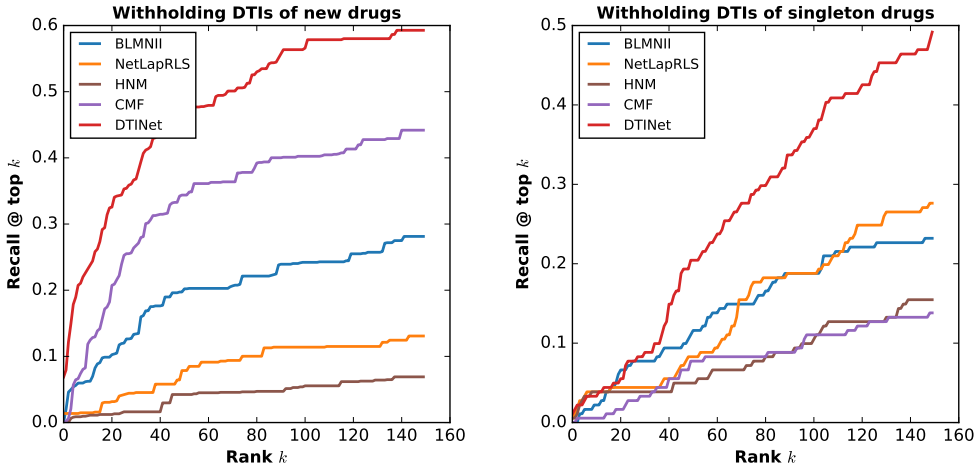


Figure B.18: Performance comparisons between different prediction approaches on the datasets after remaining similar drugs or / and targets. (a) The removal of DTIs with similar drugs (Tanimoto coefficients $> 60\%$), (b) The removal of DTIs with the drugs that share similar side-effects (Jaccard similarity scores $> 60\%$), (c) The removal of DTIs with the drugs or proteins associated with similar diseases (Jaccard similarity scores $> 60\%$), and (d) The removal of DTIs with either similar drugs (Tanimoto coefficients $> 60\%$) or homologous proteins (sequence identity scores $> 40\%$). These removal operations reduced the number of DTIs from 1,923 to 1,268, 1265, 1077 and 900 in (a), (b), (c) and (d), respectively. In (a)-(d), a matching number of randomly chosen non-interacting drug-target pairs (i.e., negative samples) with the known interacting drug-target pairs (i.e., positive samples) were used as training data. (e) The removal of the DTIs with homologous proteins (sequence similarity scores $> 40\%$) on a skewed dataset, in which known interacting drug-target pairs composed only 10% of the whole test data. All results were summarized over 10 trials of ten-fold cross-validation and expressed as mean $\pm$ SD.



(a) (b)

Figure B.19: The cumulative distributions of recall at top-k with respect to rank k when withholding the DTIs of (a) new drugs or (b) singleton drugs. The new drugs were those with new MOAs (mechanism of actions) discovered within the last five years as of the time that the DrugBank database Version 3.0 (which was used to construct our heterogeneous network) was released. The singleton drugs meant those with only one known interacting target in our dataset. The vertical axis, denoted by recall @ top-k, represents the fraction of the true interacting drug-target pairs that were retrieved in the list of top-k predictions for a drug. The maximum value of rank k was set to 150, which corresponded to roughly 10% of the total number of the targets (1,512).

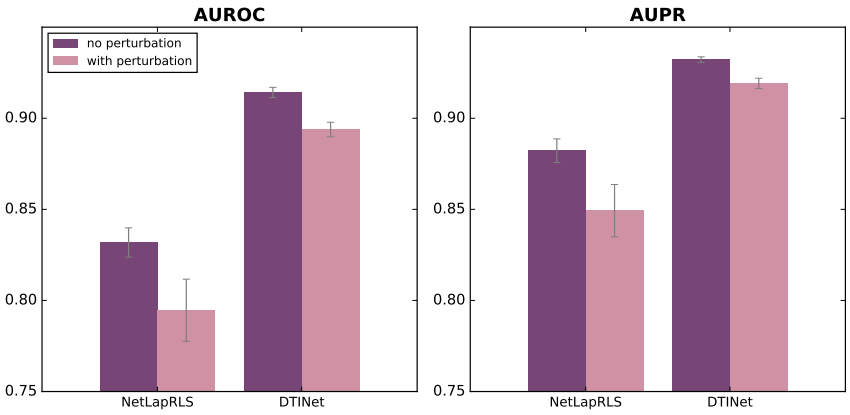
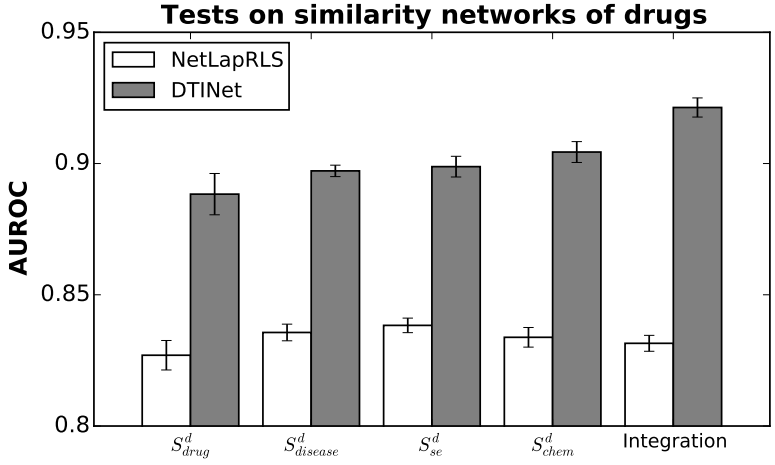
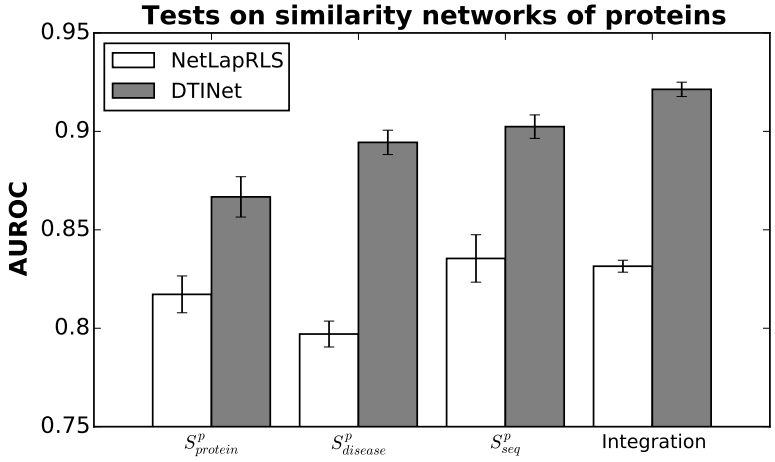


Figure B.20: Robustness of the prediction performance of DTINet with respect to the random perturbation of edges in the heterogeneous network. A fraction (10%) of randomly selected edges in the heterogeneous network were perturbed, by adding new edges or deleting existing edges. The ground truth drug-target interacting pairs used in the test data were not perturbed. All results were summarized over 10 trials of ten-fold cross-validation and expressed as mean ± SD.



(a)



(b)

Figure B.21: A comparative study on the prediction performance of DTINet and NetLapRLS on individual networks and their integration. (a) The test results on individual similarity networks of drugs and their integration, where S^d_{drug}, $S^d_{disease}$, S^d_{se} and S^d_{chem} represent the similarity networks in which the similarity score between a pair of drug nodes was computed based on the profiles of drug-drug interactions, drug-disease associations, drug-side-effect associations and chemical structures, respectively. (b) Tests on individual similarity networks of proteins and their integration, where $S^p_{protein}$, $S^p_{disease}$ and S^p_{seq} represent the similarity networks in which the similarity score between a pair of protein nodes was computed based on the profiles of protein-protein interactions, protein-disease associations and primary sequences, respectively. An extended version of NetLapRLS was used to combine all similarity networks to perform DTI prediction. All results were summarized over 10 trials of ten-fold cross-validation and expressed as mean ± SD.

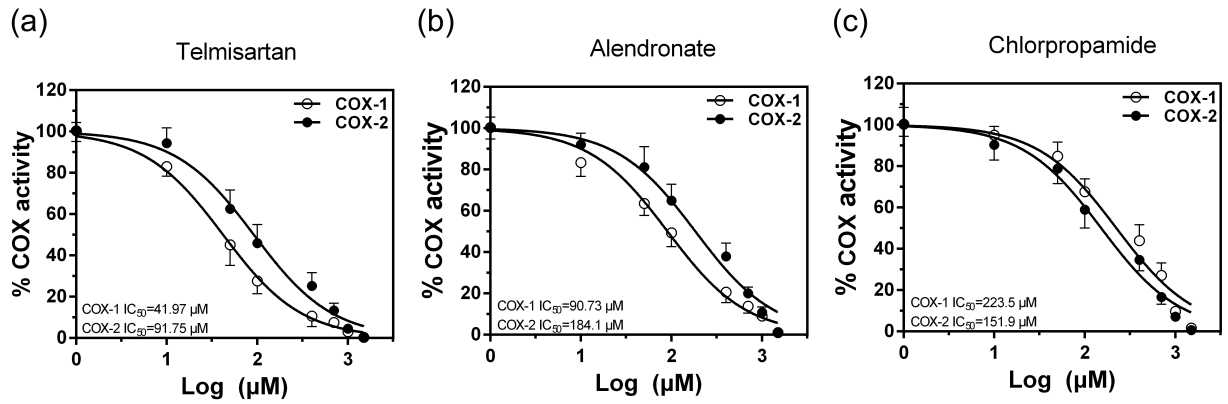


Figure B.22: The inhibitory effects of telmisartan (a), alendronate (b), and chlorpropamide (c) on COX-1 and COX-2 activities, measured by the human recombinant enzyme assays. The COX-1 (open squares) and COX-2 (closed circles) inhibitions were assessed by measuring the levels of PGE 2.

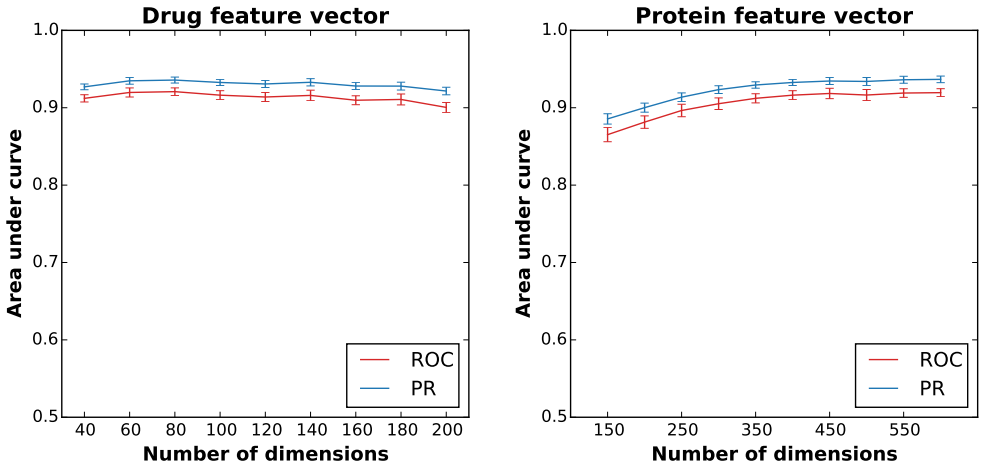


Figure B.23: Robustness of DTINet with respect to the number of dimensions of feature vectors. We evaluated the sensitivity of the prediction performance of DTINet with respect to different numbers of dimensions of the feature vectors of drugs (a) and proteins (b). We tested the dimensions of the feature vectors of drugs (f_d) and proteins (f_t) in a range that are roughly equal to 10%-30% of the dimensionality of the original vectors describing the diffusion states. DTINet had stable prediction performance over a wide range the dimensions of the feature vectors. Prediction performance was evaluated in terms of both the area under the receiver operating characteristic curve (ROC) and the area under the precision recall (PR) curve. All results were summarized over 10 trials of ten-fold cross-validation and expressed as mean ± SD.

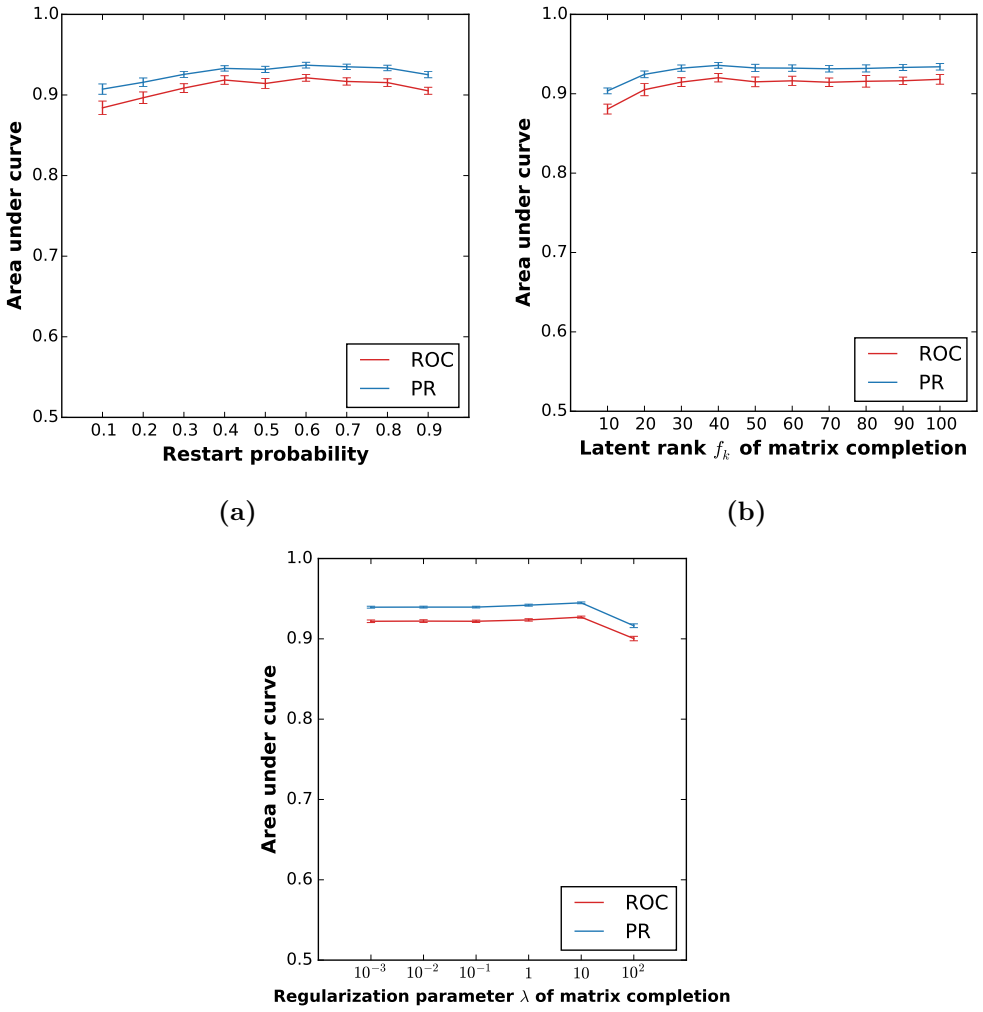


Figure B.24: Robustness of the prediction performance of DTINet with respect to different choices of parameters. We tested the prediction performance of DTINet using different values of (a) the restart probability, (b) the latent rank of matrix completion, and (c) the regularization parameter. Prediction performance was evaluated in terms of both the area under the receiver operating characteristic curve (ROC) and the area under the precision recall (PR) curve. All results were summarized over 10 trials of ten-fold cross-validation and expressed as mean ± SD.