CIRCUMVENTING DRUG RESISTANCE BY MOLECULAR DESIGN: EXACT COMBINATORIAL OPTIMIZATION AND DEEP GENERATIVE MODELS

A Dissertation

by

MOSTAFA KARIMI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Yang Shen
Committee Members,	P.R. Kumar
	Ulisses Braga-Neto
	Zhangyang Wang
Head of Department,	Miroslav M. Begovic

December 2020

Major Subject: Electrical Engineering

ABSTRACT

Drug resistance is a fundamental barrier to developing robust antimicrobial and anticancer therapies. Its first sign was observed in the 1940s, soon after discovering penicillin, the first modern antibiotic. This dissertation focuses on 1) principle-driven approach for overcoming mutational resistance in single protein target through combinatorial optimization; 2) principle- and data-driven approach for alternative pathway activation in molecular networks through deep generative models.

The energy landscape theorem for protein folding is the main principle utilized for overcoming mutational resistance in a single protein target. iCFN, an exact multistate protein design, has been developed through combinatorial optimization. iCFN has been employed for real-life applications such as engineering a HER2-specific antibody-drug conjugate and unraveling the biological mechanism underlying cancerous mutations in Estrogen receptors. Building upon iCFN, an optimal drug design approach has been developed to not only predict the resistant mutations but also overcome them by designing resistant drugs.

Drug combination therapy has been one of the strategies to overcome drug resistance since at least the 1940s. Drug combinations are effective at delaying the beginning or suppressing the expansion of the resistance. However, the chemical space has been estimated to be 10^{60}. Therefore, it is experimentally infeasible to experiment with all the compound combinations for every disease. So, a data- and principle-driven algorithm has been developed to generate drug combinations for molecular networks' desired disease. Firstly, we have developed DeepAffinity, an interpretable and accurate deep learning model for compound-protein affinity prediction. Later, it has been extended to DeepRelations, an explainable deep learning model for affinity and contact prediction. Secondly, we have designed hierarchical variation graph autoencoders (HGVAE) to jointly embed domain knowledge such as gene-gene, disease-disease, gene-disease networks. Finally, conditioning on disease representation and utilizing accurate compound-protein affinity predictor (DeepAffinity) a deep generative model has been developed through reinforcement learning and generative adversarial network to overcome resistance in molecular networks.

DEDICATION

I gratefully dedicate this dissertation to my mother, Zohreh and my wife, Masoumeh.

ACKNOWLEDGMENTS

First and foremost, I would like to genuinely thank my Ph.D. advisor, Dr. Yang Shen, for offering me immeasurable guidance, support, and encouragement throughout my Ph.D. research. I also would like to express my deepest gratitude to my co-authors, especially Di Wu and Arman Hassanzadeh. I had the pleasure of working closely with them and they contributed significantly to my Ph. D. thesis. I would like to thank my Ph.D. committee members for their constructive and supportive comments.

Last but not least, I should especially thank my wonderful mother and marvelous wife, whose consistent encouragement and support, even from a couple of continents away, were unquestionably the main reasons for my success in this journey.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Dr. Yang Shen, Dr. P.R. Kumar, and Dr. Ulisses Braga-Neto of the Department of Electrical and Computer Engineering and Dr. Zhangyang Wang of the Department of Computer Science and Engineering at Texas A&M University. All work for the dissertation was completed by the student, under the advisement of Dr. Yang Shen of the Department of Electrical and Computer Engineering.

Funding Sources

My work was supported by the National Science Foundation, through NSF award CCF-1546278 and the National Institute of General Medical Sciences of the National Institutes of Health, through NIH award R35GM124952.

NOMENCLATURE

DEE	Dead-End Elimination
CPD	Computational Protein Design
CFN	Cost Function Network
WCSP	Weighted Constraint Satisfaction Problem
DFBB	Depth First Branch and Bound
EDAC	Existential Directed Arc Consistency
SCP	Side Chain Packing
LDS	Limited Discrepancy Search
TCR	T-Cell Receptor
ER	Estrogen Receptor
MAPK	Pitogen-Activated Protein Kinase
CDK	Cyclin-Dependent Kinase
GPCR	G-Protein-Coupled Receptor
ADC	Antibody-Drug Conjugate
DDR	DNA Damage Repair
TB	tuberculosis
HIV	Human Immunodeficiency Virus
PPI	Protein-Protein interaction
CPI	Compound-Protein Interaction
DDI	Drug-Drug Interaction
SASA	Solvent-Accessible Surface Area
VdW	Van der Waals
RLU	Relative Light Unit

WT	Wild Type
MT	Mutant
MD	Molecular Dynamics
FDA	Food and Drug Administration
PDB	Protein Data Bank
RF	Random Forest
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
HRNN	Hierarchical Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
MLP	Multi-Layer Perceptrons
FC	Fully Connected
GNN	Graph Neural Network
GCN	Graph Convolutional Network
GIN	Graph Isomorphism Network
GCNN	Graph Convolutional Neural Network
HVGAE	Hierarchical Variational Graph Auto-Encoders
SMILES	Simplified Molecular-Input Line-Entry System
SPS	Structural Property Sequence
SSE	Secondary Structure Element
RMSE	Root Mean Squared Errors
AUPRC	Area Under the Precision-Recall Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
AP	Average Precision

QSAR	Quantitative Structure Activity Relationship
RL	Reinforcement Learning
DO	Disease Ontology
GO	Gene Ontology
GWAS	Genome-Wide Association Studies
GAN	Generative Adversarial Network
WGAN	Wasserstein Generative Adversarial Network
GS-WGAN	Generalized Sliced Wasserstein Generative Adversarial Network
WD	Wasserstein Distance
GSWD	Generalized Sliced Wasserstein Distance
GRT	Generalized Radon Transform
PPO	Proximal Policy Optimization
KS	Kolmogorov-Smirnov

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Drug resistance is a major obstacle in fighting against many diseases such as HIV [2], TB [3], cancers [4, 5] and so on. Drug discovery and design is a costly ($\sim$billions of USD) [6] and lengthy ($\sim$12 years) [7] process with low success rates (3.4% phase-1 oncology compounds make it to approval and market) [8].Therefore, it is worrisome if the newly developed drug loses its potency due to resistance shortly after its introduction to the market. Antibiotic resistance is the tangible example of this crisis, when a year after FDA approval of Ceftaroline (fifth generation drug), the resistance occurs and [9]. This dissertation focuses on overcoming drug resistance due to a single protein target mutation and alternative (same) pathway (re-)activation in molecular networks.

ERα, which plays a vital role in the prevention and treatment of the majority of breast cancers, is a concrete example of mutational resistance [10]. Suppression of estrogen production and Direct inhibition of ERs through selective ER modulators (SERMs) or selective ER degraders (SERDs) are the two main classes of therapeutics ERα-based breast cancer. However, after long-term exposure to these drugs, mutational resistance (Y537S, D538G) may occur [11, 12], and these drugs will lose their efficacy. The primary and challenging questions are 1) whether we can anticipate the mutational resistance; 2) understand the biological mechanism behind them; 3) Overcome the resistance by considering the mutational resistance while designing drugs.

On the other hand, BRAF V600-mutant melanoma is a concrete example of the same pathway reactivation for drug resistance [13, 14]. Targeting both MEK and BRAF in patients with BRAF V600-mutant melanoma (through Dabrafenib + Trametinib) shows less resistance with respect to targeting MAPK or BRAF alone [13, 14]. Melanoma and many other diseases show that their resistances are multifactorial [15, 16], multiple drugs targeting multiple components simultaneously could confer less resistance than individual drugs targeting components separately.

This dissertation introduces a **principle-driven** [17] approach to overcome mutational resistance in **single protein target**. Also, it introduces a **data- and principle-driven** [18] approach to overcome drug resistance for alternative (same) pathway (re-)activation in **molecular networks**.

These approaches complement each other in various settings:

- **Scope.** The principle-driven approach design drugs for a *single protein target*. However, the data- and principle-driven approach design drug combinations for *molecular networks*.

- **Objective function.** The principle-driven approach's objective function is rooted in *explicit energy models* between drugs-protein. However, the data- and principle-driven approach 's objective function is *implicitly learned from data*.

- **Aim.** The principle-driven approach is developed to *anticipate and overcome resistant mutations*. However, the data- and principle-driven approach overcome the *alternative (same) pathway (re-) activation*.

- **Formulation.** The principle-driven approach has been developed in *combinatorial optimization* framework. But, the data- and principle-driven approach has been developed through *machine learning* perspective. Particularly, it is grounded in *reinforcement learning (RL), generative adversarial networks (GANs), Bayesian modeling, and attention mechanism.*

The dissertation is structured as follows. Firstly in chapter 2, principle-driven approaches have been developed through combinatorial optimization to overcome the mutational resistance in a single protein target. Secondly, in chapters 2 and 3, data- and principle-driven approaches have been developed for drug combinations generation through reinforcement learning to overcome alternative pathway activation in molecular networks. More specifically:

- **Principle-driven approach for overcoming mutational resistance in single protein target.** iCFN (interconnected cost function networks) [17], a novel and the efficient exact algorithm has been developed for generic multistate computational protein design (with substate ensembles for both positive and negative states). iCFN has been utilized for scientific discovery in various real-life applications such as 1) Mutagenesis analysis of Pan-cancer DNA Damage Repair Deficiency; 2) Engineering a HER2-specific antibody-drug conjugate to increase lysosomal delivery and therapeutic efficacy; 3) Anticipating Cancer Mutations and unraveling mechanism for Estrogen Receptors. Furthermore, taking one step further, an optimal single and cocktail drug design approach has been developed to not only predict the resistant mutations but also overcome them by designing drugs. Three novel optimal drug design formulations have been introduced for different mutational resistance scenarios. Three drug design applications are 1) specificity over proteins: designing drugs to bind to "targeted" proteins rather than "off-targets" 2) Specificity over states of a protein: designing a drug to bind to "desired" state rather than "undesired" state of a protein 3) specificity over drugs: designing a drug to compete with another compound (such as a natural compound in our body) for binding to a "targeted" protein. Novel lower bounds with theoretical proofs and computational complexity analysis have been developed for each drug design problem. Furthermore, formulations and lower bounds have been extended to optimal drug cocktail design.

- **data- and principle-driven approach for drug combinations generation.** For the first time, interpretable (DeepAffinity [19]) and explainable (DeepRelation [20]) deep learning models have been developed for fast and accurate compound-protein affinity and contact predictions. Various applications of the proposed models have been illustrated, such as 1) Binding site prediction; 2) discovering the underling mechanism behind protein specificity; 3) Structure-activity relationship (SAR) and lead optimization; 4) Assisting ligand-protein docking. Particularly, DeepAffinity has been used as a fast and accurate compound protein affinity oracle inside drug combinations generation. A data- and principle-driven deep generative model has been developed for faster, broader, and deeper exploration of drug combination space by following the principle underlying FDA approved drug combinations [18]. More specifically, Hierarchical Variational Graph Auto-Encoders (HVGAE) has been developed for jointly embedding disease-disease network and gene-gene networks. Secondly, a reinforcement-learning graph-set generator model has been developed for drug combination design by utilizing both gene/disease embedding and network principles.

2. PRINCIPLE-DRIVEN APPROACH FOR OVERCOMING MUTATIONAL RESISTANCE IN SINGLE PROTEIN TARGET*

2.1 Overview

A principle-driven approach for overcoming mutational resistance has been developed in this chapter. This is the first approach proposed in this dissertation for overcoming resistance. More specifically, in this approach, 1) we aim to design drugs for single protein target; 2) we have objective functions rooted in explicit energy terms; 3) we are in the pursuit of developing exact and efficient algorithms through combinatorial optimization. This chapter is structured as follows. In section 2.2, iCFN an efficient and exact algorithm for multi-state protein design is introduced. Its efficiency and accuracy are assessed through benchmark and known proteins. In section 2.3, iCFN has been utilized for scientific discovery in real-life applications. iCFN has been used for Mutagenesis analysis, "acid-switched" ADC design, and anticipating cancer mutations in ERα. In section 2.4, iCFN has been extended for optimal drug design to not only anticipate the resistance mutation but also to consider it while designing drugs. An optimal single drug and drug cocktail designs have been formulated as a minimax problem. Efficient and guaranteed lower bounds have been developed for exact DFBB approaches.

2.2 iCFN: an efficient exact algorithm for multistate protein design

2.2.1 Introduction

Designing proteins of desired structures, properties, or functions would enable unraveling and modulating biological systems and allow for a wide array of applications. Although this important problem remains challenging, progress has been made with computational protein design (CPD), sometimes together with experimental approaches such as directed evolution. CPD introduces an

*Reprinted with permission from "Engineering a HER2-specific antibody–drug conjugate to increase lysosomal delivery and therapeutic efficacy" by J. C. Kang, W. Sun, P. Khare, M. Karimi, X. Wang, Y. Shen, R. J. Ober, E. S. Ward, 2019. Nature biotechnology 37, no. 5, 523-526, Copyright 2019 Nature Research.

*Reprinted with permission from "iCFN: an efficient exact algorithm for multistate protein design" by M. Karimi, Y. Shen 2018. Bioinformatics 34, no. 17, i811-i820, Copyright 2018 Oxford University Press.

automated and accelerated way to tackle the problem. More importantly, it can rationally generate a tangible number of designs (and their underlying mechanistic hypotheses) which can be experimentally tested to refine our knowledge.

CPD is often formulated as an optimization problem where a utility or objective function summarizing a single or multiple design objectives is optimized over protein sequence space. As protein functions are largely determined by structures and dynamics, evaluating an objective function for any given sequence often involves energy minimization over structures (or conformations). There are two cases of structure-based CPD problems. The first is single-state design that only considers one desired (or "positive") state (e.g. stability of a given, fixed backbone conformation). However, there are two limitations with single-state design: (1) without the explicit consideration of an undesired (or "negative") state, a designed binder may not be foldable or no specificity can be achieved; (2) without the consideration of multiple positive or negative substates such as conformations, folds, oligomers and on/off-target binding, no multiple sub-objectives, positive or negative, can be accomplished and over-simplified assumptions often have to be made (for instance, a fixed backbone despite that flexible protein structures exist in an ensemble of conformational substates [21, 22]). Rather, the second case of CPD – multistate design – removes both limitations by considering both positive and negative states *and* allowing multiple substates for either state [23]. Some "multistate" design studies only remove one limitation thus we emphasize the difference between "state" and "substate" here to remove possible confusions.

Even single-state CPD is extremely challenging. With protein backbones fixed and side chains discretized as rotamers [24], the resulting combinatorial optimization problem is NP-hard [25] thus has no polynomial-time algorithm. Over the past two decades, three types of algorithms have been developed for single-state design: heuristic, approximation, and exact algorithms, among which only exact algorithms can guarantee the global optimum.

Heuristic algorithms include genetic algorithms [26] and Markov chain Monte Carlo (MCMC) that can generate good-quality feasible solutions efficiently. In particular, MCMC is used in the very popular Rosetta software [27] and has led to many successful applications [28, 29, 30, 31, 32, 33]. Approximation algorithms include relaxed integer programming [34] and loopy belief propagation [35, 36] that solve approximate forms of the problem.

Despite progress in heuristic or approximation algorithms for single-state design, there is a critical need for exact algorithms due to two major reasons. First, the guarantee of the global optimum from exact algorithms assures that biophysical models and mechanistic hypotheses underlying the formulation can be isolated from search algorithms and improved based on design success or failure; and the guarantee of a gap-free list of the top sub-optimum directly addresses uncertainty in those biophysical models (such as free energy calculation). Second, the performance gap between exact and heuristic algorithms widens as the size of single-state design grows [37] and this gap will be even wider for multistate design whose size grows further with the number of substates.

The first and the most known framework of exact algorithms is dead-end elimination (DEE) followed by A* [38, 39, 40, 41, 42]. DEE is widely used to prune the search space; and A* [43] is a tree search algorithm for enumerating a gap-free ordered list in the pruned space. Original DEE criteria [44, 45] have evolved to more powerful albeit more costly ones [46, 47, 48].

Furthermore, the DEE framework has been extended by the Donald group to first consider flexible side-chain rotamers in minDEE [49] and iMinDEE [50], then locally flexible backbones within voxel boxes [51], and recently locally flexible backbones and side-chain rotamers in DEEPer [52]. Other promising extensions include deriving tighter bounds in BroMAP [53] and dynamic A* [54] as well as exploiting the sparseness of protein residue contact maps in AND/OR branch-and-bound search [55].

Recently, a new framework of exact algorithms called cost function network (CFN) has been introduced to re-formulate single-state design as a weighted constraint satisfaction problem (WCSP) [56, 57] modelled through a cost function network (CFN) and to solve it using depth first branch-and-bound (DFBB) [58, 59]. CFN is shown to be significantly faster than other exact methods or solve problems of sizes unprecedented to DEE/A* [37, 60]. Various local consistencies have been developed for lower bounding in DFBB [61, 62, 63, 64, 65, 66], among which existential directed arc consistency (EDAC) is used the most for its balance between tightness and cost in practice.

However, for multistate protein design with substate ensembles, no exact algorithm exists except an extension of DEE/A* – COMETS (Constrained Optimization of Multi-state Energies by Tree Search) [67]. Progress has been focused on heuristic or approximation algorithms [23, 68, 69, 70, 71, 72, 73]. For multistate design, DEE has been extended to type-dependent DEE where only rotamers of the same amino-acid type can prune each other [74]. For multistate design where the objective function is a linear combination of substate energies, COMETS incrementally searches for the lowest-scoring sequence with A* by exploiting new lower bounds and generates the top few sequences.

Here we present iCFN (interconnected cost function networks), a novel and efficient exact algorithm for generic multistate CPD (with substate ensembles for both positive and negative states). Our optimization formulation is general enough for various design tasks. And our algorithm guarantees a gap-free list of the best sequences and conformations with unprecedented efficiency for practical, large-scale multistate CPD problems. Specifically, we have adopted the formulation of WCSP and the model of CFN for each substate; and represented the coupled WCSPs as interconnected CFNs over a tree of sequences, substates, and rotamers (values).

Then we have derived novel lower bounds with theoretical proofs and complexity analysis; and we have designed DFBB-based tree search that allows positive and negative designs to inform each other and substates within and across states to prune each other. Finally, we have applied iCFN to designing a T-cell receptor (TCR) to specifically recognize an antigen peptide and avoid another while allowing all molecules' backbones to be globally flexible. For the resulting multistate CPD problems of unprecedented sizes to exact methods, iCFN drastically improves the efficiency and accuracy compared to state-of-the-art methods and provides new insights into the importance of backbone flexibility in CPD and molecular mechanisms of binding specificity.

2.2.2 Materials and Methods

2.2.3 Formulation

We will first introduce and formulate various cases of computational protein design of increasing computational complexity and biophysical relevance. Bold-faced notations in lower cases indicate vectors.

2.2.3.1 Single-state design with a single substate

A simple CPD is to find the best sequence $\mathbf{s}$ that optimally accommodates a desired (positive) substate as measured in an objective function $f(\mathbf{x})$ of protein structure $\mathbf{x}$. Proteins are often assumed to have fixed backbones and discrete side-chain rotamers $\mathbf{r}$ at selected, mutable or flexible residues. So the only non-fixed part of the structure $\mathbf{x}$ consists of these side-chain rotamers $\mathbf{r}$. The objective function, often some form of energy functions, is usually assumed to be a sum of constant, singleton, and pairwise terms:

$$f(\mathbf{r}) = c + \sum_{i} E(i_r) + \sum_{i<j} E(i_r, j_s), \tag{2.1}$$

where i_r and j_s denote rotamers r and s at residue i and j, respectively. An example of $f(\cdot)$ is the energy $E(\cdot)$ of protein structure $\mathbf{x}$ to stabilize a desired, fixed backbone structure potentially for a desired function.

The resulting optimization problem can be formulated as

$$\mathbf{s}^* = arg \min_{\mathbf{s} \in \mathcal{S}} \min_{\mathbf{r} \in \mathcal{R}(\mathbf{s})} f(\mathbf{r}), \tag{2.2}$$

where the set $\mathcal{S}$, capturing the sequence design space, is the Cartesian product of the sets of amino-acid types allowed over all residues; and $\mathcal{R}(\cdot)$, capturing the rotamer library, is the Cartesian product of the rotamer sets over all mutable or flexible residues of a sequence.

2.2.3.2 Single-state design with substate ensembles

This slightly more complex case also considers just positive design (or state) but considers an ensemble of positive substates rather than one. Such treatment leads to more accurate biophysical models and more design capabilities. For instance, it allows for treating a protein backbone flexible when these substates correspond to backbone conformers; and it allows for designing binding profiles when these substates correspond to various ligand-bound states. We give the formulation as

$$\mathbf{s}^* = arg \min_{\mathbf{s}\in\mathcal{S}} \min_{p\in\mathcal{P}} \min_{\mathbf{r}\in\mathcal{R}_p(\mathbf{s})} f_p(\mathbf{r}), \tag{2.3}$$

where $\mathcal{P}$ is the set of positive substates and $f_p(\cdot)$ is the objective function for the p^{th} positive substate. Minimizing over $\mathcal{P}$ substate objective functions $f_p(\cdot)$ maintains an "OR" relationship among them for the overall (positive) objective. For instance, in the case of $f_p(\cdot)$ being energies $E_p(\cdot)$, it ensures choosing the ground substate with its sequence and conformation being optimized simultaneously. One can also maximize over $\mathcal{P}$.

2.2.3.3 Multistate design with a single substate per state

Compared to the two previous cases with positive state only, this case considers both positive and negative states each represented by a single substate. In other words, it is to find the best sequence that specifically accommodates a desired positive substate rather than an undesired negative substate. The objective function here can be the gap between the two substate objective functions $f^{+}(\cdot)$ and $f^{-}(\cdot)$. The formulation is given as

$$\mathbf{s}^* = arg \min_{\mathbf{s}\in\mathcal{S}} \big(\min_{\mathbf{r}\in\mathcal{R}^{+}(\mathbf{s})} f^{+}(\mathbf{r}) - \min_{\mathbf{r}\in\mathcal{R}^{-}(\mathbf{s})} f^{-}(\mathbf{r}) \big), \tag{2.4}$$

where superscripts "+" and "-" indicate positive and negative quantities, respectively. For instance, this formulation allows for binding affinity design when positive and negative substate objective functions are energies of a protein in one bound and one unbound state, respectively. It also allows for binding specificity design when these substate objective functions are binding energies to one target and one off-target, respectively.

2.2.3.4 Multistate design with substate ensembles

The most generic formulation, for which our exact algorithm will solve, considers both positive and negative states explicitly *and* considers an ensemble of substates for either state. The formulation can be written as

$$\mathbf{s}^* = arg\min_{\mathbf{s}\in\mathcal{S}} \big(\min_{p\in\mathcal{P}} \min_{\mathbf{r}\in\mathcal{R}_p^+(\mathbf{s})} f_p^+(\mathbf{r}) - \min_{q\in\mathcal{Q}} \min_{\mathbf{r}\in\mathcal{R}_q^-(\mathbf{s})} f_q^-(\mathbf{r})\big), \tag{2.5}$$

where $\mathcal{P}$ and $\mathcal{Q}$ denote the positive and the negative substate ensemble with p and q being the positive and negative substate index, respectively. Moreover, constraints on substate objective functions can be introduced and addressed (e.g. those linear ones in our T-cell receptor design).

This generic formulation includes all aforementioned formulations as special cases. It helps improve the accuracy of biophysical models and strengthen the capability to design for multiple desired substates over multiple undesired ones. For instance, one can design a tight binder that can fold using protein-complex and binder alone as positive and negative states in conformational ensembles, respectively, as our XRCC1 design does in Sec. 2.2.7. One can also design a protein that specifically binds to a target "ensemble" rather than an off-target one with $f_p(\cdot)$ and $f_q(\cdot)$ being energies for the p^{th} target and q^{th} off-target, respectively, as our T-cell receptor design does in Sec. 2.2.8. The $\min$ operator over all positive substates can be replaced by $\max$ for multi-specificity and solved similarly.

2.2.4 iCFN for multistate design with substate ensembles

With the generic formulation given, we proceed to introduce our exact algorithms based on cost function networks (CFN). CFN is the state-of-the-art approach to single-state protein design with a single substate [58, 59]. We extend CFN for multistate design with substate ensembles. We first design a tree structure of sequences, substates, and rotamers and a tree-search algorithm using CFN as a corner stone, which leads to a reduced version of the ultimate iCFN. Here CFN is used to solve each substate energy minimization problem for any given sequence, a problem also known as side chain packing (SCP). We further improve the reduced version to interconnected CFNs (iCFN) by deriving novel bounding schemes across CFNs.

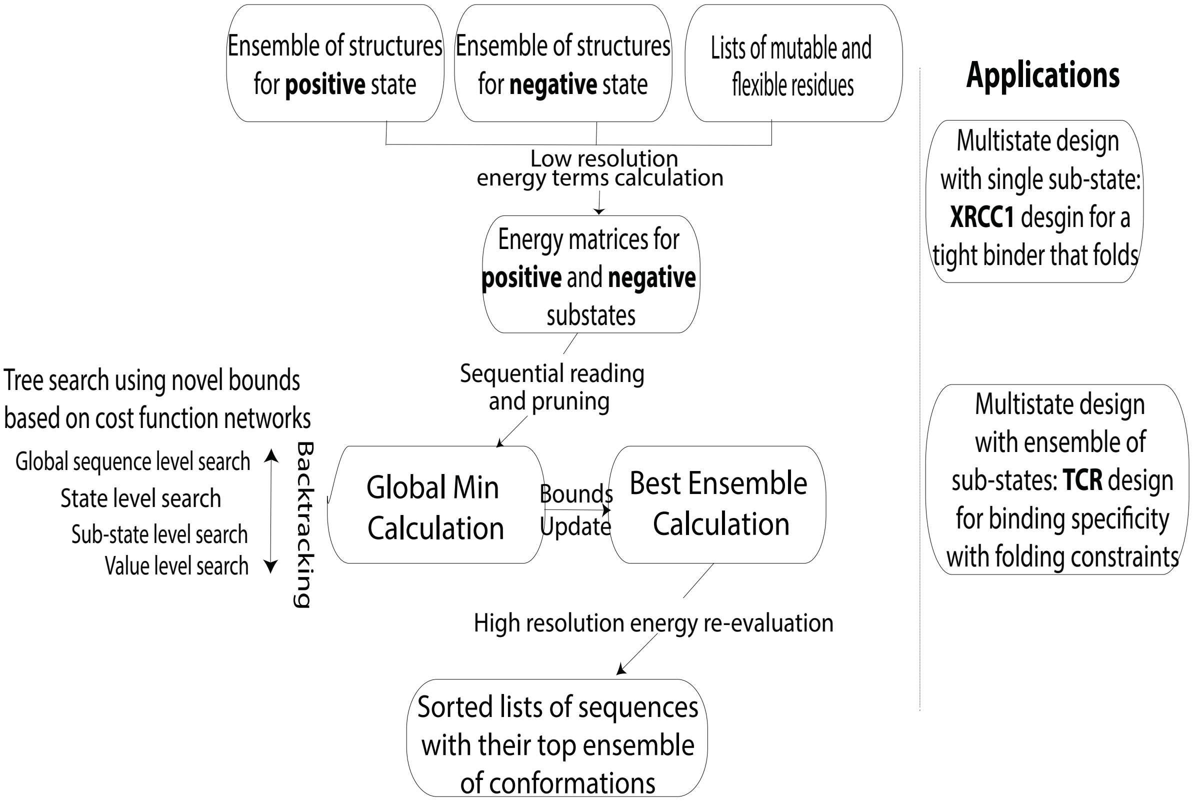

Ensemble of structures for positive state
Ensemble of structures for negative state
Lists of mutable and flexible residues
Low resolution energy terms calculation
Energy matrices for positive and negative substates
Sequential reading and pruning
Tree search using novel bounds based on cost function networks
Global sequence level search
State level search
Sub-state level search
Value level search
Backtracking
Global Min Calculation
Bounds Update
Best Ensemble Calculation
High resolution energy re-evaluation
Sorted lists of sequences with their top ensemble of conformations
Applications
Multistate design with single sub-state: XRCC1 desgin for a tight binder that folds
Multistate design with ensemble of sub-states: TCR design for binding specificity with folding constraints

Figure 2.1: Schematic illustration of the study: algorithm design and application.

A high-level schematic illustration of iCFN is shown in Fig. 2.1. For either positive or negative state, iCFN first reads data (singleton and pairwise energy values) and prunes the search space using type-dependent DEE within and across substates sequentially. Individual substate designs are reformulated as weighted constraint satisfaction problems (WCSP) and modeled with interconnected cost function networks over a tree representation of the search space. Using a depth-first branch-and-bound (DFBB) approach, iCFN then searches over sequence space with newly proposed and proven lower bounds to prune partially or fully defined sequences and searches over substate-rotamer space for un-pruned, fully-defined sequences (i.e. side chain packing). After the global optimum is found, it will redo DEE pruning and DFBB search with updated bounds and relaxed energy thresholds for an ensemble of the best sequences in an ensemble of the best positive and negative conformations. Next we will explain these steps in more details.

2.2.4.1 Sequential reading and pruning of rotamers

iCFN first sequentially reads data and incrementally prunes rotamers for either state. When reading each substate, it prunes rotamers within the substate using type-dependent DEE – Goldstein and single split DEE for all substates as well as single pair and single double DEE just for positive substates. It then prunes rotamers for substates read so far using our extended, across-substate type-dependent DEE. This approach drastically reduces peak memory usage to store substate rotamers. We provide its pseudocode in Algorithm 1.

Algorithm 1 Pseudocode for sequential reading and pruning.

```
for i = 1: N do
  Read_substate(i)
  Type_Dep_DEE(i,$\delta$)
  if Is_it_first_substate() then
    Create_main_substate()
  else
    Substate_Dep_DEE(i,$\delta$)
    if Is_new_substate_pruned() == 0 then
      Concat_to_main_substate()
    end if
  end if
end for
```

We extend within-substate type-dependent DEE [74] to across-substate type-dependent DEE as follows:

Theorem 1. *Rotamer i_a of position i in substate 1, provably pruned by rotamer i_b of the same position in substate 2, is not part of the optimal solution if both substates are of the same state,* *both rotamers are of the same amino-acid type, and the following criterion holds:*

$$
\begin{aligned}
&L_1(i_a) = c_1 + E_1(i_a) + \sum_{j,j\neq i} \min_{s_1} \big(E_1(j_{s_1}) + E_1(i_a, j_{s_1})\big) \\
&+ \sum_{j>k,k\neq i,j\neq i} \min_{s_1,u_1} E_1(j_{s_1}, k_{u_1}) \\
&> U_2(i_b) = c_2 + E_2(i_b) + \sum_{j,j\neq i} \max_{s_2} \big(E_2(j_{s_2}) + E_2(i_b, j_{s_2})\big) \\
&+ \sum_{j>k,k\neq i,j\neq i} \max_{s_2,u_2} E_2(j_{s_2}, k_{u_2})
\end{aligned}
\tag{2.6}
$$

Proof. Following Eq. 2.1, the energy of substate 1 (used as a subscript) with rotamer i_a at residue i and its upper bound can be written as:

$$
\begin{aligned}
&c_1 + \sum_{m} E_1(m_r) + \sum_{m<j} E_1(m_r, j_s) \\
&= c_1 + E_1(i_a) + \sum_{m\neq i} E_1(m_r) + \sum_{j\neq i} E_1(i_a, j_s) + \\
&\sum_{m<j,m\neq i,j\neq i} E_1(m_r, j_s) \\
&> c_1 + E_1(i_a) + \sum_{j,j\neq i} \min_{s} \big(E_1(j_s) + E_1(i_a, j_s)\big) + \\
&\sum_{j>m,m,j\neq i} \min_{s,r} E_1(j_s, m_r) \triangleq L_1(i_a)
\end{aligned}
\tag{2.7}
$$

By doing the same for rotamer i_b in substate 2:

$$\begin{aligned}
& c_2 + \sum_{m} E_2(m_r) + \sum_{m<j} E_2(m_r, j_s) \\
& = c_2 + E_2(i_b) + \sum_{m \neq i} E_2(m_r) + \sum_{j \neq i} E_2(i_b, j_s) + \\
& \sum_{m<j, m \neq i, j \neq i} E_2(m_r, j_s) \\
& < c_2 + E_2(i_b) + \sum_{j, j \neq i} \max_{s} \big(E_2(j_s) + E_2(i_b, j_s) \big) + \\
& \sum_{j>m, m \neq i, j \neq i} \max_{s,r} E_2(j_s, m_r) \triangleq U_2(i_b)
\end{aligned} \tag{2.8}$$

Therefore, if $L_1(i_a) > U_2(i_b)$, then i_a is pruned by i_b and cannot be part of the global optimum.

A natural extension for the top δ kcal/mol ensemble is that rotamer i_a of substate 1 is pruned by rotamer i_b of substate 2 if $L_1(i_a) > U_2(i_b) + \delta$. □

An extension for the top δ kcal/mol ensemble is that rotamer i_b of substate 2 prunes rotamer i_a of substate 1 if $L_1(i_a) > U_2(i_b) + \delta$. In some applications especially for the optimal ensemble, these across-substate DEEs can increase computational cost more than they add pruning power and thus can be disregarded in iCFN as we later do for TCR design.

2.2.4.2 Global sequence-level search

The second stage of iCFN performs DFBB search over the sequence space, which is represented in a hierarchical tree structure together with states, substates, and conformations. The overall search strategy is illustrated in Fig. 2.2. Beginning with a completely undefined sequence indicated by all ”X”, it splits the current sequence space (parent node) into two subspaces (child nodes), based on the first amino acid being valine (V) or not. It then evaluates the lower bound on the right child corresponding to a partially defined sequence and determines whether to prune its entire subtree or to split it again. The so-called binary branching repeats until reaching a sequence-level leaf node (i.e., a fully-defined sequence) whose lower bound is evaluated for pruning. If the

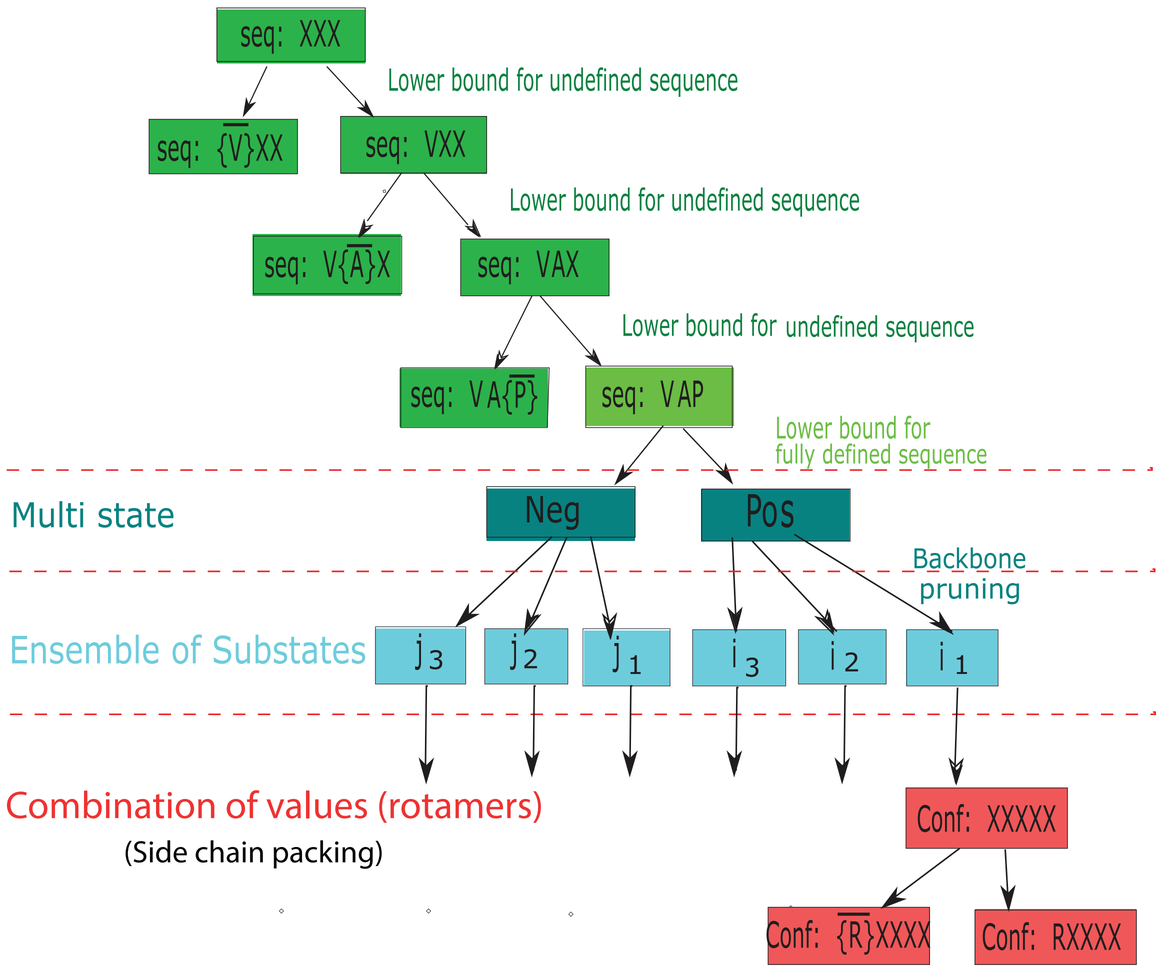


Figure 2.2: Schematic illustration of global sequence search.

sequence is not pruned, state, substate, and rotamer-level search follows with similar DFBB (next subsections). Then iCFN calculates the sequence's specificity score and updates the upper bound for optimal specificity if the score is lower than the best specificity so far.

Two types of lower bounds with proofs and complexity analysis are developed for iCFN to prune sub-trees of sequences (including their associated substates and rotamers) when the search reaches a sequence-level leaf node (i.e., a fully-defined sequence) or otherwise. They are not included in reduced iCFN to assess their sole contribution to numerical efficiency.

The first lower bound is generically applicable to all sequences, fully defined or not.

Theorem 2. *For any sequence space S, a lower bound of the objective function for multistate* *protein design with substate ensembles (Formulation in Eq. 2.5) is given by (× denotes Cartesian product):*

$$
\begin{aligned}
&\min_{(k,l)\in\mathcal{P}\times\mathcal{Q}} \Big(\Delta c_{kl} + \sum_{i} \min_{\mathbf{a}\in S(i)} \min_{(r,r')} \big(\Delta E_{kl}(i_{r,r'}) + \\
&\qquad \sum_{j>i} \min_{\mathbf{a}'\in S(j)} \min_{(s,s')} \Delta E_{kl}(i_{r,r'}, j_{s,s'})\big)\Big), \textit{ where}
\end{aligned} \tag{2.9}
$$

$$
\begin{aligned}
&\Delta c_{kl} = c^{+}_{k} - c^{-}_{l}, \\
&\Delta E_{kl}(i_{r,r'}) = E^{+}_{k}(i_r) - E^{-}_{l}(i_{r'}), \\
&\Delta E_{kl}(i_{r,r'}, j_{s,s'}) = E^{+}_{k}(i_r, j_s) - E^{-}_{l}(i_{r'}, j_{s'}),
\end{aligned} \tag{2.10}
$$

i.e., differences in constant, singleton, and pairwise energies between a positive substate k and a negative substate l.

Proof. For an arbitrary sequence $\mathbf{a}$ in the space S, its rotamer vector $\mathbf{r}$ is in the space of $\mathcal{R}_k(\mathbf{a})$ for substate k. The highest specificity is thus:

$$
\begin{aligned}
&\min_{\mathbf{a}\in S} \big(\min_{k\in\mathcal{P}} \min_{\mathbf{r}\in\mathcal{R}_k(\mathbf{a})} E^{+}_{k}(\mathbf{r}) - \min_{l\in\mathcal{Q}} \min_{\mathbf{r}'\in\mathcal{R}_l(\mathbf{a})} E^{-}_{l}(\mathbf{r}')\big) \\
&\geqslant \min_{\mathbf{a}\in S} \min_{(k,l)\in\mathcal{P}\times\mathcal{Q}} \min_{(\mathbf{r},\mathbf{r}')\in\mathcal{R}_k(\mathbf{a})\times\mathcal{R}_l(\mathbf{a})} \big(E^{+}_{k}(\mathbf{r}) - E^{-}_{l}(\mathbf{r}')\big) \\
&\geqslant \min_{\mathbf{a}\in S} \min_{(k,l)\in\mathcal{P}\times\mathcal{Q}} \Bigg(\Delta c_{kl} + \min_{(r,r')} \big(\sum_{i} \Delta E_{kl}(i_{r,r'}) \\
&+ \sum_{j>i} \Delta E_{kl}(i_{r,r'}, j_{s,s'})\big)\Bigg) \\
&\geqslant \min_{\mathbf{a}\in S} \min_{(k,l)\in\mathcal{P}\times\mathcal{Q}} \Bigg(\Delta c_{kl} + \sum_{i} \min_{(r,r')} \big(\Delta E_{kl}(i_{r,r'}) \\
&+ \sum_{j>i} \min_{(s,s')} \Delta E_{kl}(i_{r,r'}, j_{s,s'})\big)\Bigg)
\end{aligned} \tag{2.11}
$$

$$
\begin{aligned}
&= \min_{(k,l)\in\mathcal{P}\times\mathcal{Q}} \Bigg(\Delta c_{kl} + \min_{\mathbf{a}} \sum_i \min_{(r,r')} \big(\Delta E_{kl}(i_{r,r'}) \\
&+ \sum_{j>i} \min_{(s,s')} \Delta E_{kl}(i_{r,r'}, j_{s,s'})\big)\Bigg) \\
&\geqslant \min_{(k,l)\in\mathcal{P}\times\mathcal{Q}} \Bigg(\Delta c_{kl} + \sum_i \min_{\mathbf{a}\in S(i)} \min_{(r,r')} \big(\Delta E_{kl}(i_{r,r'}) \\
&+ \sum_{j>i} \min_{\mathbf{a}'\in S(j)} \min_{(s,s')} \Delta E_{kl}(i_{r,r'}, j_{s,s'})\big)\Bigg)
\end{aligned}
$$

□

The complexity of evaluating the lower bound for undefined sequences is given as follows:

Theorem 3. *The lower bound in Theorem 2 can be computed in $O\big((nRa)^2 r\big)$, where n is the number of positions, R the average number of rotamers per position, a the average number of substates per state, and r the average number of rotamers per amino acid.*

Proof. we prove the complexity by starting with the most inner minimization:

$$
\begin{aligned}
&\min_{(s,s')} \Delta E_{kl}(i_{r,r'}, j_{s,s'}) \\
&= \min_{(s,s')} \Big(E_k^+(i_r, j_s) - E_l^-(i_{r'}, j_{s'})\Big) \\
&= \min_s E_k^+(i_r, j_s) + \min_{s'} \Big(-E_l^-(i_{r'}, j_{s'})\Big) \\
&= \min_s E_k^+(i_r, j_s) - \max_{s'} E_l^-(i_{r'}, j_{s'})
\end{aligned}
\tag{2.12}
$$

So, we can calculate it in $O(r)$. Since the number of amino acids is known, then

$$
\min_{a'\in S(j)} \min_{(s,s')} \Delta E_{kl}(i_{r,r'}, j_{s,s'}) \tag{2.13}
$$

will be again $O(r)$, so by summing over positions it will be $O(nr)$. For calculating:

$$
\min_{a\in S(i)} \min_{(r_k,r_l)} \Big(\Delta E_{kl}(i_{r,r'}) + \sum_{j>i} \min_{a'\in S(j)} \min_{(s,s')} \Delta E_{kl}(i_{r,r'}, j_{s,s'})\Big) \tag{2.14}
$$

similar to previous version, we can compute it in $O(nR^2r)$ and summing over all positions it will be $O(n^2R^2r)$. Finally, since we are calculating (2.14) for all a^2 pairs of substates across the two states, complexity will be $O(a^2n^2R^2r)$. □

In practice, we only use the first lower bound for partially defined sequences and have derived a tighter one for fully defined sequences:

Theorem 4. *For any defined sequence* s*, a lower bound can be given by*

$$\min_{k\in\mathcal{P}} L_k^{+}(\mathbf{s}) - \min_{l\in\mathcal{Q}} U_l^{-}(\mathbf{s}) \tag{2.15}$$

in which $L_k^{+}(\mathrm{s})$ *is the lower bound on all rotamers for sequence* s *and* k^{th} *substate in positive design and* $U_l^{-}(\mathrm{s})$ *is the upper bound on all rotamers for sequence* s *and* l^{th} *substate in negative design.*

Proof. When sequence is fully defined, a lower bound can be derived by:

$$\begin{aligned}
&\min_{k\in\mathcal{P}} \min_{\mathbf{r_k}\in R_k(\mathbf{s})} E_k^{+}(\mathbf{r_k}) - \min_{l\in\mathcal{Q}} \min_{\mathbf{r_l}\in R_l(\mathbf{s})} E_l^{-}(\mathbf{r_l}) \\
&\geqslant \min_{k\in\mathcal{P}} L_k^{+}(\mathbf{s}) - \min_{l\in\mathcal{Q}} U_l^{-}(\mathbf{s}),
\end{aligned} \tag{2.16}$$

in which $L_k^{+}(\mathrm{s})$ can be any lower bound from single-state protein design (we use existential directed arc consistency a.k.a. EDAC) and $U_l^{-}(\mathrm{s})$ can be any upper bound from single state protein design (we use limited discrepancy search a.k.a. LDS). □

2.2.4.3 State and substate-level search

Once a fully defined sequence s is reached and cannot be pruned, it splits into child nodes of positive and negative states and follows positive substates then negative ones. iCFN repeats DFBB in the rotamer space for side chain packing (SCP) in each substate. We use the following bounding criteria to prune substates.

- Within the same state: Substate k prunes l if they are in the same state (positive/negative) and $U_k^{\cdot}(\mathrm{s}) < L_l^{\cdot}(\mathrm{s}) - \delta$ where the superscript · stands for either + or -. We again use EDAC and LDS for $L_l^{-}(\mathbf{s})$ and $U_k^{-}(\mathbf{s})$, respectively.

- Across the two states: For a negative substate q, all the subsequent negative substates will be skipped if no rotamers both pass the within-state pruning and satisfy $\min\{f^{+}(\mathbf{s})\} - f_q^{-}(\cdot) > S_{\text{best}} + \varepsilon$, where $\min\{f^{+}(\mathbf{s})\}$ denotes the optimal value among all positive substate functions for the sequence s and S_{best} is the lowest specificity score of the best sequence so far. If $q = 1$ (the first negative substate), the sequence s is also pruned.

There might be more substate constraints in practice. For instance, our TCR design formulation has a stability condition for positive substates: $L_p^{+}(\mathbf{s}) > \min\{f^{+}(\text{WT})\} + \tau$. Therefore, a positive substate p is pruned if its stability lower bound is worse than wild type by more than τ. Other user-defined constraints on substates can further speed up the search.

2.2.4.4 Value-level (rotamer) search (side-chain packing)

Once reaching a substate that is not pruned, iCFN again uses binary branching to iteratively split during search the conformational space into a chosen rotamer and all the rest. For pruning conformational subtrees, the search again uses EDAC as lower bounds and LDS as upper bounds in each side chain packing. After a leaf node of the tree (a fully defined conformation for a fully defined sequence) is visited and cannot be pruned, it either becomes the best solution so far or enters the δ-ensemble for the corresponding sequence in the substate. The ensemble size for each sequence in each substate can be limited to K where the K choices can be the first or the best (implemented with a max-heap data structure).

2.2.4.5 Backtracking

When a sequence- or rotamer-level node is pruned with its subtree, our tree search backtracks to its parent node, re-orders variables (positions) and values (amino acids or rotamers) in the tree (see ordering in the next subsection), and repeats the DFBB process.

2.2.4.6 Ordering

The ordering of positions, amino acid types, and rotamers in the search tree also has an impact on the pruning efficiency. We use several ordering heuristics, originally developed for constraint satisfactory problems (CSP) and later extended for weighted CSPs, to boost the speed of iCFN without compromising its guarantee of the global minimum or the gap-free top list.

For variable (position) ordering, the state of the art is the increasing order by the number of amino acids or rotamers over the median of pre-calculated energy terms for global sequence search or side chain packing, respectively. The principle is to visit nodes of higher energies earlier to prune their child nodes more likely and visit nodes of fewer combinations to prune more bigger subtrees.

We improve the efficiency for iCFN by using the median of singleton terms only. The rationale is that singleton terms (e.g. interactions between side chains and backbones) often dominate over pairwise terms in side chain packing problems [75, 76]. Note that this treatment does not affect the accuracy of iCFN. In practice it may lead to slightly increased number of nodes expanded or leaves visited but still saves running time with much less time spent on each node for bound estimation.

For amino acid ordering per position, the wild type is by default the first and the rest is ordered by increasing singleton energy values. And rotamer ordering for each amino acid type is again by the increasing order of singleton energy values. These two orderings are following the principle of increasing the chance to find a good feasible solution early.

2.2.5 Test on T-cell receptor design

We will now introduce the design problem to test our algorithms, with formulation specifics and implementation details.

2.2.5.1 Background

T-cell receptors (TCRs) recognize peptide antigens presented by major histocompatibility complex (MHC) and play a critical role in the immune response. Therefore, TCRs have been actively pursued for cancer immunotherapy. For instance, the first TCRs developed for melanoma are DMF4 and DMF5 which recognize two structurally distinct peptide epitopes of MART-1 (melanoma antigen recognized by T cells 1) bound to MHC class I protein HLA-A*0201 (HLA-A2). Regulation of redesigned TCR with high affinity and specificity toward target peptide-MHC (pMHC) has been a major task to develop effective TCR-based immunotherapies. Whereas improving binding affinity has represented major efforts so far because of TCRs' relatively weak binding to pMHC, such improvements often come at a cost of binding specificity to target peptides and thus bring the risk of off-target effects (for instance, strong affinity to MHC regardless of peptide antigens). In addition, evidence shows that TCR affinity above a certain threshold would cause T cell responsiveness to attenuate. In total, there is a pressing need for the rational design of TCRs of carefully tailored affinity and specificity profiles.

We used the example of TCR DMF5 [77] to design optimal binding specificity while constraining the target-complex folding stability. The target, AAG peptide, is MART-1 nonameric epitope (AAGIGILTV) and the off-target, ELA peptide, is MART-1 decameric epitope (ELAGIGILTV). We modeled global backbone flexibility of bound DMF5, peptides, and MHC with a conformational ensemble sampled by molecular dynamics simulations.

2.2.5.2 Biophysical model

Each conformation of TCR-pMHC in the ensemble was treated as a substate. A hierarchy of energy models is used. (1) During the tree search, folding energy (stability) of TCR-pMHC was used as the substate function. Energy terms included Coulomb electrostatics, van der Waals and internal energies as calculated in a CHARMM22 force field as well as nonpolar contributions of continuum electrostatics based on solvent-accessible surface area (SASA). (2) After iCFN generates the top sequence-conformation ensemble, binding energy difference between the target and the off-target (specificity) was used as the overall objective. Folding energies were re-evaluated with a higher-resolution energy model where implicit-solvent Poisson-Boltzmann electrostatics replaces Coulombic electrostatics [78, 42]. Binding energy for the lowest-energy conformation was reported for each sequence in either state.

2.2.5.3 Substate ensembles

Both peptides were previously crystallized in complex with MHC HLA-A2 or TCR DMF5 and available with PDB accession codes 3QDJ or 3QDG. Both structures were first minimized using a molecular modeling software CHARMM in a CHARMM22 force field with missing residues and atoms added. They were then solvated using VMD with explicit water molecules of 10 Å padding thickness from the molecular boundary and ionized to reach neutral charge and a concentration of 0.145M. Either system was minimized for 5000 steps and a 10-ns molecular dynamics (MD) simulation was performed using a computer program NAMD2. Starting from the beginning of MD simulations, snapshots were retained every nano-second. In total, there are 11 positive and 11 negative substates.

2.2.5.4 Computational mutagenesis

We chose four positions as in an earlier study [77]: residues 26 and 28 on the α chain of DMF5 and residues 98 and 100 on the β chain. Each mutable position is allowed for 26 amino-acid types (some amino acids with multiple protonation states are each counted more than once). Since folding energy was first used, specificity cutoff ε and positive-substate stability cutoff τ were set loose at 1000 Kcal/mol while ensemble cut-off δ per sequence at 2 Kcal/mol. iCFN searched for the best K=1000 conformations for each sequence. To reduce conformational representatives for higher-resolution energy evaluation while maintaining diversity, top conformations of each sequence in each substate were geometrically grouped [39] and only the representatives were evaluated for higher-resolution folding and binding energies.

2.2.6 Results and Discussion

2.2.7 Numerical comparison to COMETS

We first compare iCFN to the only alternative exact method for multistate protein design, COMETS [67], released in OSPREY V2.2 [40]. COMETS uses the weighted sum of substate energies as its objective function, which differs from our formulation in Eq. 2.5. So we resort to compare the methods using the same objective function (without constraints), energy calculations and rotamers (without continuous ones) as in OSPREY, which leads to multistate design with a single substate per state as in Eq. 2.4. Specifically, the task is to minimize the binding energy of a protein complex (XRCC1 N-term domain and DNA polymerase beta; PDB code: 3K75) as the difference between the bound (or positive) and the unbound (or negative) state [67].

Out of five positions in XRCC1 (residues 391, 409, 411, 422, and 424), we incrementally choose the first N_{mut} to be mutable; and for each choice of mutable residues, we set all the N_{flex} residues within $d = 3$Å or 6Å to be flexible, and increasingly demand the top affinity sequence ensemble (ε=0.5, 1.0, and 1.5 Kcal/mol while δ stays 2 Kcal/mol).

					ε = 0.5 Kcal/mol			1 kcal/mol			1.5 Kcal/mol		
N_{mut}	d(Å)	N_{flex}	Pre-DEE Size	Post-DEE Size	COMETS	Reduced iCFN	iCFN	COMETS	Reduced iCFN	iCFN	COMETS	Reduced iCFN	iCFN
1	3	9	2.88×10^{17}	2.04×10^{8}	4.03	0.02	0.02	4.27	0.03	0.3	4.37	0.03	0.03
1	6	16	5.24×10^{30}	2.75×10^{19}	6.84	0.16	0.12	6.97	0.17	0.12	7.72	0.19	0.13
2	3	10	1×10^{22}	1.73×10^{12}	6.85	0.28	0.15	8.36	0.28	0.16	9.29	0.28	0.16
2	6	19	5.88×10^{37}	1.58×10^{25}	19.46	2.63	1.41	29.07	2.67	1.44	29.27	2.79	1.47
3	3	11	7.94×10^{27}	8.31×10^{18}	M	12.64	6.15	M	12.67	6.51	M	12.65	6.54
3	6	20	1.81×10^{43}	1.44×10^{30}	M	62.17	32.9	M	62.22	33.26	M	62.31	33.47
4	3	14	6.54×10^{36}	8.31×10^{26}	M	493.26	268.28	M	493.31	268.3	M	493.95	268.41
4	6	26	7.94×10^{56}	4.36×10^{38}	M	2060	1458	M	2156	1493	M	2161	1498
5	3	15	3.54×10^{42}	2.34×10^{30}	M	9810	5570	M	9943	6005	M	10 046	6040
5	6	26	1.65×10^{61}	1.58×10^{42}	M	49 373	37 198	M	49 978	37 223	M	50 405	39 553

Table 2.1: Search space statistics and running time (in seconds) comparison between COMETS, reduced iCFN, and iCFN over a series of incrementally larger multi-state XRCC1 design problems with a single substate for either positive or negative state. ("M" indicates an out-of-memory error under a 20-Gb limit.)

From Table 2.1 we conclude that our algorithms outperform COMETS in both memory usage and CPU time, which enables large designs in practice. Whereas COMETS couldn't handle designs of more than 2 mutable positions under a 20-Gb memory limit, reduced iCFN and iCFN can design for all 5 positions operating below 80-Mb memory. For the single and double designs where all algorithms produced results, reduced iCFN and iCFN are faster than COMETS by 1$\sim$2 orders of magnitude.

Theoretical reasons underlie the better performance of our algorithms. On the memory demand, the space complexity of our DFBB is $O(K)$ where K is the number of mutable and flexible positions whereas that of A* used in COMETS is $O(e^K)$. On the computational speed, cost function network-based DFBB enjoys stronger lower and upper bounds. Specifically, (1) DFBB uses lower bounds such as EDAC which proved more powerful than PFC-DAC used in DEE/A* [79]. In particular, even when it could not prune the whole subtree for a given position, EDAC can still reduce the subtree size by pruning rotamers of the remaining positions. (2) For larger problems, DFBB often reaches good-quality solutions much faster than A*, which provides tighter upper bounds earlier.

2.2.8 TCR Design: Efficiency

We now apply iCFN to T cell receptor design described in Methods. As shown later, these multistate designs with substate ensembles and dense rotamer libraries are even larger than those with single substates seen in XRCC1 designs. Therefore, we could not apply COMETS under our cluster's memory and CPU-time limits (50Gb and 7 core days). Instead, we compare exhaustive search, reduced iCFN (separate CFNs for individual substates), and iCFN to evaluate the benefit of treating CFNs jointly for powerful substate pruning during search.

2.2.8.1 Reduction in the sequence and conformer spaces

We compare the effective size of the search space after DEE pruning among exhaustive search, reduced iCFN, and iCFN, using the following metrics: 1) the number of fully-defined sequences searched, which is the same for exhaustive search and reduced iCFN but much lower for iCFN; and 2) the number of conformers searched, which, for reduced iCFN and iCFN, include not only leaves (fully defined conformations) visited but also nodes (partially defined conformations) expanded at the rotamer level of side chain packing for those sequences searched. These results are summarized in Table 2.2 for the global optimum-specificity sequence and Table 2.3 for the top ε-Kcal/mol ensemble ($\varepsilon = 3$) of sequences.

As seen in both tables, the TCR design problems feature sizes unprecedented to exact protein-design methods even after type-dependent DEE: up to 10^{56} (10^{82}), 10^{104} (10^{132}), and 10^{134} (10^{162}) for global optimum (ensemble) single, double, and triple designs, respectively.

Reduced iCFN drastically shrinks the conformational space for search. For the global optimum,

			Reduced iCFN				iCFN			
Position(s)	Pre-DEE Size	Post-DEE Size	Nodes Expanded	Leaves Visited	Sequences	Time (s)	Nodes Expanded	Leaves Visited	Sequences	Time (s)
26	10^{61}	10^{27}	2.45×10^3	1.41×10^2	26	1.46	8.55×10^2	36	3	0.56
28	10^{66}	10^{49}	3.75×10^4	1.26×10^3	26	24.5	4.12×10^3	114	4	6.29
98	10^{58}	10^{42}	2.51×10^4	8.16×10^2	25	9.98	2.11×10^3	88	8	3.38
100	10^{84}	10^{56}	4.92×10^4	1.75×10^3	26	19.85	3.78×10^3	101	3	4.44
26,28	10^{87}	10^{62}	1.46×10^6	3.75×10^4	676	1335.95	5.99×10^4	791	40	228
26,98	10^{119}	10^{68}	1.74×10^6	3.88×10^4	650	809.18	9.85×10^3	231	14	182.10
26,100	10^{142}	10^{82}	4.08×10^6	1.02×10^5	676	1510.03	1.01×10^4	234	5	303.64
28,98	10^{126}	10^{92}	5.20×10^6	9.34×10^4	650	3707.04	3.04×10^5	4568	106	745.84
28,100	10^{141}	10^{104}	9.78×10^6	1.80×10^5	676	5603.60	2.00×10^4	349	8	796.96
98,100	10^{112}	10^{88}	6.03×10^6	9.34×10^4	650	4384.48	3.39×10^4	633	19	526.97
26,28,98	10^{146}	10^{106}	1.68×10^8	2.66×10^6	16 900	133 672	7.09×10^5	8171	180	16 879
26,28,100	10^{161}	10^{122}	2.94×10^8	5.11×10^6	17 576	205 865	4.41×10^4	652	14	22 941
26,98,100	10^{169}	10^{117}	3.28×10^8	4.32×10^6	16 900	202 001	1.88×10^5	2425	55	23 430
28,98,100	10^{168}	10^{134}	5.86×10^8	7.35×10^6	16 900	496 051	7.45×10^5	8284	103	46 685

Table 2.2: Comparing search space statistics and running time between reduced iCFN and iCFN for global optimum in multi-state TCR design with substate ensembles.

			Reduced iCFN				iCFN			
Position	Pre-DEE Size	Post-DEE Size	Nodes Expanded	Leaves Visited	Sequences	Time (s)	Nodes Expanded	Leaves Visited	Sequences	Time (s)
26	10^{61}	10^{54}	6.35×10^4	6.20×10^4	26	66.69	6.14×10^3	6.00×10^3	10	21.86
28	10^{66}	10^{61}	5.92×10^4	5.70×10^4	26	114.22	4.09×10^3	4.00×10^3	2	23.55
98	10^{58}	10^{55}	5.15×10^4	5.00×10^4	25	103.29	4.16×10^4	4.00×10^4	20	43.35
100	10^{84}	10^{77}	7.43×10^4	7.11×10^4	26	154.51	5.19×10^3	5.00×10^3	2	23.74
26,28	10^{87}	10^{82}	1.70×10^6	1.62×10^6	676	7454.93	9.44×10^3	9.00×10^3	4	1063.89
26,98	10^{119}	10^{111}	1.82×10^6	1.73×10^6	650	15 449.04	2.51×10^5	2.38×10^5	108	3872.32
26,100	10^{142}	10^{132}	1.89×10^6	1.75×10^6	676	19 780.68	2.62×10^4	2.40×10^4	10	2226.52
28,98	10^{126}	10^{119}	1.45×10^6	1.37×10^6	650	23 378.51	3.13×10^4	3.00×10^4	13	2810.31
28,100	10^{141}	10^{132}	1.77×10^6	1.60×10^6	676	24 631.34	4.22×10^3	4.00×10^3	2	2359.10
98,100	10^{112}	10^{106}	1.60×10^6	1.51×10^6	650	17 303.91	3.98×10^4	3.80×10^4	19	2056.47
26,28,98	10^{146}	10^{141}	—	—	16 900	—	5.86×10^4	5.50×10^4	27	105 343
26,28,100	10^{161}	10^{154}	—	—	17 576	—	1.48×10^4	1.40×10^4	6	99 012
26,98,100	10^{169}	10^{161}	—	—	16 900	—	6.76×10^4	6.00×10^4	27	185 886
28,98,100	10^{168}	10^{162}	—	—	16 900	—	3.73×10^4	3.40×10^4	12	158 995

Table 2.3: Comparing search space statistics and running time between reduced iCFN and iCFN for the top sequence ensemble in multi-state TCR design with a substate ensemble per state. (“—” indicates an out-of-time error under a 7-day limit.) Note that the ensemble versions are run after corresponding global optima are derived to reach a tight sequence-level specificity bound and their statistics do not include those in the global optimum stage reported in Table 2.2.

it only evaluates up to the order of 10^2, 10^5, and 10^6 fully-defined conformations and up to the order of 10^4, 10^6, and 10^8 partially-defined conformations en route for single, double, and triple designs, respectively. This space-reduction power does not weaken significantly with the increase of the space size as type-dependent DEE does. For the top ensemble, it still only evaluates to the order of 10^4 and 10^6 fully-defined conformations or partially-defined conformations en route for single and double designs, respectively.

iCFN shows even more space-reduction power compared to reduced iCFN because, unlike the latter, it reduces the sequence space (and associating substrees) besides the conformational space.

In fact, iCFN visits on average 6.7 (7.4), 58.8 (110.8), and 455.1 (1397.2) times less sequences for the best single (ensemble of) sequence(s) in single, double, and tripe designs, respectively. Therefore, iCFN for global optimum impressively only evaluates to the order of 10^2, 10^3, and 10^3 fully-defined conformations and 10^3, 10^5, and 10^5 partially-defined conformations en route, which translates to 9.2, 214.7, and 2358.6 times more space reduction on average compared to reduced iCFN, for single, double, and triple designs, respectively. For the top ensemble, iCFN has similar space-reduction improvement compared to reduced iCFN and visits only up to 10^5 conformations, fully or partially defined, to solve triple designs that could not be handled by reduced iCFN within 7 CPU days.

2.2.8.2 Acceleration in running time

Since search space reduction can come at a cost of relatively expensive bound calculations, we proceed to compare running time in Table 2.2 for the global-optimum sequence and Table 2.3 for the top sequence ensemble. The design jobs (especially for the top ensemble) are daunting for an exhaustive search or even COMETS. So we only compare between reduced iCFN and iCFN to examine the algorithmic benefits of interconnections among substate CFNs. The pre-search step of sequential reading and pruning of rotamers is the same between both and excluded in running time reported.

iCFN drastically accelerates conformational search even compared to reduced iCFN and the acceleration is observed to improve with the increase of the problem complexity. Specifically, iCFN runs 3.4, 5.9, and 9.0 times faster than reduced iCFN does for global optimum in an average single, double, and tripe design, respectively. For the top ensemble, iCFN runs 4.2 and 7.8 times faster than reduced iCFN does in an average single and double design, respectively; and it solves tripe designs within 1$\sim$2 CPU days whereas reduced iCFN could not within 1 CPU week. The results show that the benefit of pruning power clearly outweighs the burden of bound calculations.

2.2.8.3 Performance improvement vs. problem complexity

In Fig. 2.3 we summarize how much the power of sequence bounding among CFNs grows with the increase of the problem complexity. iCFN manifests its power in reducing sequence space and running time even more with the increase of problem complexity when more positions are mutated or more top solutions are desired.

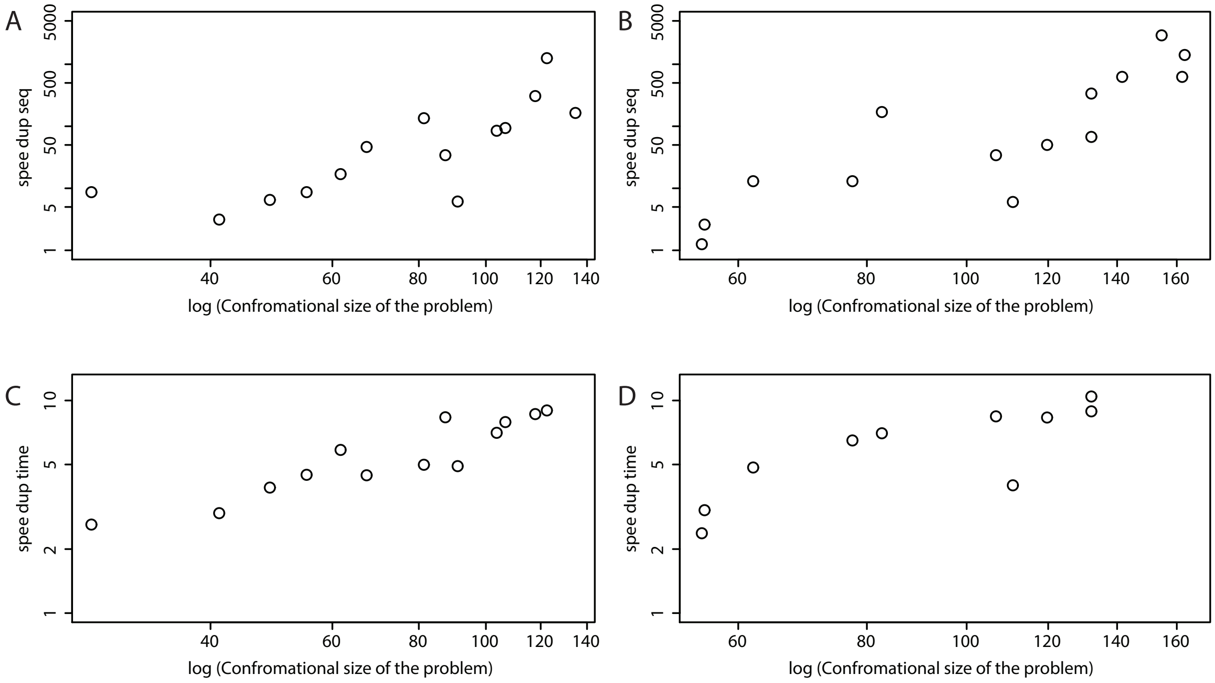


Figure 2.3: Compared to reduced iCFN, the speedup of iCFN in the number of sequences explored for (A) the global optimum and (B) the best ensemble as well as that in running time for (C) the global optimum and (D) the best ensemble.

Beyond substate pruning enabled by interconnected CFNs, three more improvements we made also contribute to the numerical efficiency. The first affects both reduced iCFN and iCFN: (1) variables (positions) are ordered based on the number of rotamers divided by the median of singleton energies only, which affects the nodes and leaves (and ultimately sequences) visited during tree search. The rest two are both for calculating lower bounds of undefined sequences thus only affect iCFN: (2) a lookup table storing intermediate min/max values for each substate reduces calculations in the order of the number of substates, and (3) an upper bounding when minimizing differences over substate pairs can be accomplished with any feasible solution.

To dissect the contributions of these three additional contributions, we start with none and incrementally introduce them into versions 0 (none), 0.1, and 0.2, where the latter two only apply to iCFN. The latest version is regarded version 1. By comparing them, we find that the change of position ordering may lead to slightly increased number of nodes expanded or leaves visited but saves run time due to much less time spent on each node for bound estimation. In addition, the lookup tables are created only once and used multiple times in search, which especially speed up large designs (twice for double designs).

	Reduced iCFN		iCFN			
Position(s)	v0	v1	v0	v0.1	v0.2	v1
26	4.06	1.46	4	0.6	0.62	0.56
28	291	24.5	289	5.88	7.32	6.29
98	32	9.98	26	3.26	4.46	3.38
100	33	19.85	30	4.18	4	4.44
26,28	16 152	1335.95	12 774	676.61	248.26	228
26,98	3540	809.18	2627	283.63	172.89	182.10
26,100	3799	1510.03	3008	686.67	330.68	303.64
28,98	27 252	3707.04	21 809	1522.54	717.25	745.84
28,100	20 521	5603.60	16 605	1785.23	738.04	796.96
98,100	19 808	4384.48	13 726	1257	534.42	526.97

Table 2.4: Comparing run time (in seconds) between different versions of reduced iCFN and iCFN for the best global optimum conformation in multi-state design problems with ensemble of substates per state for TCR.

	Reduced iCFN		iCFN			
Position(s)	v0	v1	v0	v0.1	v0.2	v1
26	6654	66.69	3700	22.94	31.48	21.86
28	4283	114.22	2222	22.32	29.3	23.55
98	10 612	103.29	6318	40.81	55.72	43.35
100	2109	154.51	1102	21.20	20.05	23.74
26,28	384 997	7454.93	205 656	16 120	1705.68	1063.89
26,98	—	15 449.04	—	27 596	4666.22	3872.32
26,100	502 803	19 780.68	265 185	12 162	2689.77	2226.52
28,98	—	23 378.51	487 360	20 629	3561.14	2810.31
28,100	347 872	24 631.34	177 949	11 452	2956.08	2359.10
98,100	—	17 303.91	323 104	11 781	2700.47	2056.47

Table 2.5: Comparing run time (in seconds) between different versions of reduced iCFN and iCFN for the best ensemble conformations in multi-state design problems with ensemble of substates per state for TCR. ("—" indicates an out-of-time error under the 7-day limit.)

2.2.9 TCR Design: Accuracy

2.2.9.1 Comparison to experimental results and Rosetta

When comparing results to experimental results reported earlier [77], we found that iCFN correctly predicted most known AAG-specific TCRs. Specifically, 7 AAG-specific mutants involving 4 residues were previously reported, including α chain D26W, G28L, G28I, G28Y, and G28N as well as β chain F100Y and F100W (βL98W is excluded for its specificity being below experimental error bar). iCFN correctly predicted 6 of the 7 (missing G28N) and produced a false positive D26Y. In contrast, when Pierce et. al. used Rosetta V2.3 [27] for specificity design, they only found 3 (all at residue 28 and missing G28N as well).

To assess iCFN's conformational search and energy models, we also compare computed and measured relative binding affinities ($\Delta\Delta G$) as well as binding specificities ($\Delta\Delta\Delta G$) for the 6 correct designs (G28N not included) and D26Y. For $\Delta\Delta G$, iCFN achieved a Pearson correlation coef-

Method	TP	FP	FN
Rosetta	G28I, G28L, G28Y, F100W	D26Y	D26W, F100Y
Rosetta Min	G28I, G28L, G28Y	N/A	D26W, F100W, F100Y
iCFN	D26W, G28I, G28L, G28Y, F100W, F100Y	D26Y	N/A

Table 2.6: AAG-specificity predictions by Rosetta, Rosetta Min, and iCFN.

ficient of 0.52, which is between that of Rosetta with interface backbone minimization (0.39) and Rosetta without (0.62) [77]. iCFN's $\Delta\Delta G$ values over-estimated weak binding affinities for ELA, possibly due to its lenient DEE-pruning and constraints for negative substates. Using $\Delta\Delta\Delta G < 0$ calculated to predict specificity, we find in Table 2.6 that Rosetta led to 1 false positive (FP) and 2 false negatives (FN) and Rosetta Min did 3 FNs [77] whereas iCFN found the most true positives (TP) with only 1 FP and no FN.

2.2.9.2 The impact of substate ensemble and backbone flexibility

We also perform a multistate design with a single substate for either state (i.e., a fixed backbone conformation). Whereas a flexible-backbone treatment correctly predicted 6 of 7 AAG-specific mutants, a fixed-backbone treatment only did for 3 (G28L, F100W and F100Y). In fact, for the flexible-backbone treatment, various backbone conformations were adopted in iCFN for various successful designs in the AAG- or ELA-bound complex (Table 2.7). These results echo the biophysical concept of protein conformational substates and highlight the importance of backbone flexibility to multistate protein design for more diverse solutions and higher success rates.

Mutation	AAG ($\Delta\Delta$G)	ELA ($\Delta\Delta$G)
$\alpha D26W$	4	8
$\alpha G28L$	10	1
$\alpha G28I$	10	3
$\alpha G28Y$	10	8
$\alpha F100Y$	7	1
$\alpha F100W$	10	2

Table 2.7: TCR designs considering an ensemble of positive or negative substates (flexible backbone conformations here). Reported are indices of various backbone conformations that were adopted in iCFN for various successful designs bound to the AAG peptide (MART-1 nonameric epitope) and the ELA peptide (MART-1 decameric epitope).

2.2.9.3 *Characterization of the design space*

Beyond individual predictions, iCFN effectively produced energetic landscapes in the design space by generating the top sequences and conformation ensembles. Although conformational flexibility is limited, it predicted that position 98 on the α chain, having much less sequence solutions and worse binding affinities, is much less "designable" for AAG-specificity compared to the other three positions, which agrees with previous experimental observations [77]. Other potentially promising designs for AAG-specificity can be found in Table S3.

2.2.9.4 *Molecular mechanisms for AAG-binding specificity*

Our results correctly predicted that introducing bulkier hydrophobic residues properly to position 26/28/100 could improve binding specificity. Moreover, consistent with experimental results, they predicted various patterns for AAG-specificity including 1) strengthening AAG-binding but weakening ELA-binding (G28I); 2) strengthening both binding but more to AAG (D26W); and 3) weakening both binding but less to AAG (G28Y and F100W/Y). They also correctly predicted G28L to have weakened binding to ELA but incorrectly predicted its improved binding to AAG.

Taking G28I as an example, we examine the mutant's energetic and structural consequences in details. Our results show that G28I achieves binding specificity by exploiting the peptide sequence difference, namely the N-terminus being alanine in AAG but glutamate in ELA. Specifically, a bulkier I28 would strengthen interactions with AAG, mainly van der Waals packing with the N-terminal alanine in AAG; but it would weaken interactions with ELA due to both van der Waals clashes with MHC/peptide and worse continuum electrostatics (Fig. 2.4). This explanation agrees with the design rationale from the previous study [77]. And it further suggests the importance of continuum electrostatics for G28I's peptide binding specificity, which has not been raised elsewhere. We note that the negatively charged N-terminal glutamate in ELA was solvent-accessible with TCR wild type but partially blocked from the solvent with the bulky substitution G28I, which leads to increased desolvation penalty.

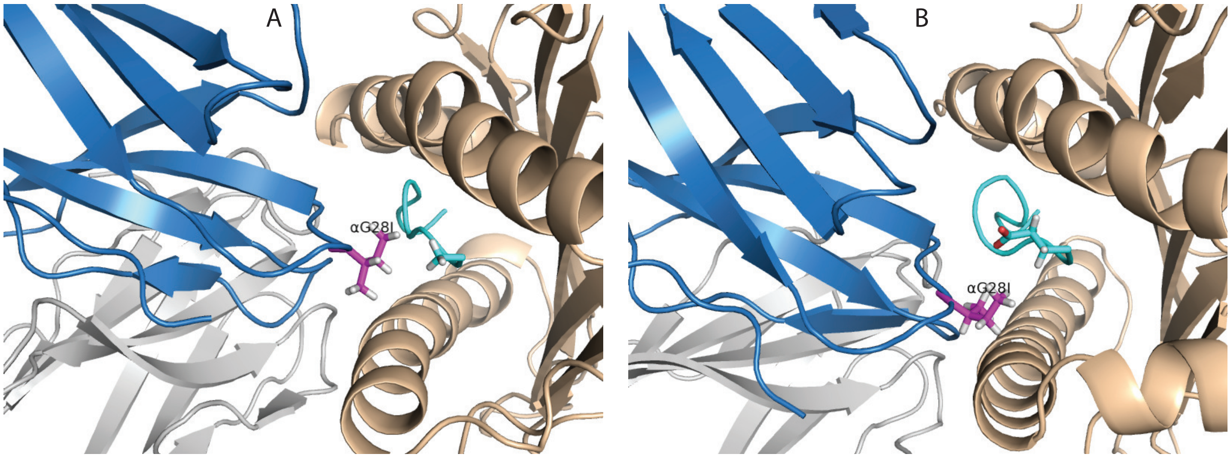


Figure 2.4: Differential effects of G28I to (A) AAG-binding and (B) ELA-binding revealed in iCFN structural models. TCR DMF5 α and β chains are shown in blue and gray cartoons, MHC Class I protein HLA-A2 in wheat cartoon, AAG/ELA peptide in cyan cartoon with N-terminal alanine/glutamate in sticks, and substitution I28 in purple sticks, respectively.

2.2.10 Conclusions

We have developed, for the first time, an exact algorithm that is efficient for generic multistate protein design of unprecedented sizes to previous exact methods. The combinatorial optimization problem is formulated as coupled weighted constraint satisfaction problems (WCSPs) where each

WCSP corresponds to a substate design and is modeled by a cost function network (CFN). The algorithm exploits novel bounds that can be quickly evaluated in the framework of CFN as well as joint consideration of substate CFNs that can quickly prune subtrees at the sequence, substate, and conformer levels with guarantee.

Applications to XRCC1 binding affinity design and TCR specificity design prove that iCFN can be at least 1 to 2 orders of magnitude faster than COMETS with much less memory demand and can solve problems of sizes intangible to COMETS. Also, iCFN shows competitive accuracy compared to Rosetta in replicating experimental results. More importantly, our results suggest that the consideration of backbone global flexibility leads to more diverse solutions and higher sensitivity in protein design. And they provide new mechanistic insights into specificity in protein interactions. Future directions include parallelizing the algorithm and its codes on the architecture of GPU, incorporating more types of constraints seen in applications while allowing for more general objective functions and continuous rotamers, and deriving tighter yet economic bounds under the framework of CFN.

2.3 Scientific discovery through iCFN

2.3.1 Introduction

In the previous section, iCFN is shown to be accurate and efficient (unprecedentedly fast) on the benchmark (XRCC) and known (TCR design) applications. However, in this section, iCFN has been utilized for scientific discovery in real-life applications. iCFN has been utilized for three critically important and challenging applications varying from single-state single substate problem to multi-state ensemble of substates problems (iCFN has been used for other tasks such as predicting cancerous mutation in Frataxin and BRCA1 [80, 81]). More specifically, for a single-state single substate problem, we used iCFN for mutagenesis analysis. Three fundamental pathways and their corresponding genes in DNA damage repair (DDR) processes were studied and analyzed for understanding the effect of mutations across the cancer genome atlas. For the multi-state ensemble of substates formulation, two applications were studied: 1) an "acid-switch" HER2-specific antibody-drug conjugate (ADC) has been engineered to increase the lysosomal delivery and therapeutic efficacy. Further, the potential underlying biological mechanism behind the "acid-switch" ADC was discovered for SG mutations; 2) Cancerous mutations for Estrogen receptors (ERs) in breast cancer have been identified. Also, potential hypotheses for a couple of critical mutations such as Y537S, D538G, and E380Q were discovered to explain their specificity toward the agonist state rather than the antagonist state.

2.3.2 Mutagenesis analysis of Pan-cancer DNA Damage Repair Deficiency across The Cancer Genome Atlas

2.3.2.1 Introduction

DNA damage repair (DDR) genes are crucial for maintaining human genomic stability [82]. Loss of DDR function is an essential factor for the prediction of cancer risk and progression [83]. Through genetic and biochemical criteria, DDR genes can be grouped into functional pathways in which they work together to repair specific types of DNA damage [84]. For example, Base excision repair (BER) pathway repair DNA base damage; Homology-dependent recombination

(HR), and non-homologous end joining (NHEJ) pathways repair DNA strand breaks [84]. In this application, we analyzed the effect of mutations on protein with structural information available in BER, NHEJ, and HR pathways. We utilized the single-state single substate formulation in iCFN for the mutagenesis analysis.

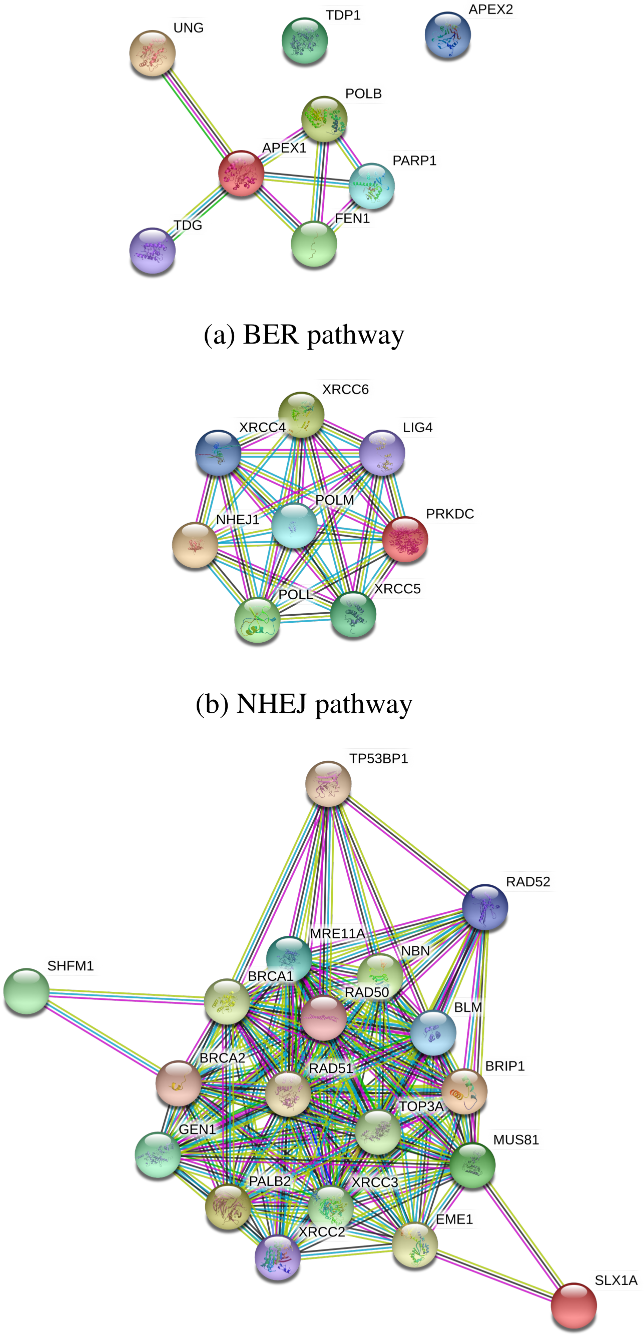
2.3.2.2 Method

With backbone conformation fixed and side-chain conformation discretized as backbone dependent rotamers, each structure-based computational mutagenesis experiment for a given protein-protein complex was performed by solving a combinatorial optimization problem to find the lowest-energy conformations of the complex structure [42] Through iCFN [17]. The mutation site is allowed to adopt wild-type or mutant amino acid type, and residues within 6 Å of a mutation site are allowed to be flexible. iCFN first finds an ensemble of conformations (only differing in side-chain rotamers in this case) in a low-resolution force field: not only the global minimum but also all or 100 conformations within 2 Kcal/mol from the global minimum; then, it re-evaluates energy values of the ensemble of conformations in a high-resolution force field.

The energy to be minimized is the folding energy of a protein complex (“folc” in short), which can be decomposed into contributions from two thermodynamic processes: individual proteins fold into separate unbound structures (“fold”) and then bind to form a protein-protein complex (“bind”): $\Delta G_{\text{folc}} = \Delta G_{\text{fold}} + \Delta G_{\text{bind}}$. The low-resolution force field used to generate conformer ensembles is an all-atom molecular mechanics (MM), including internal energy (bond, angle, dihedral, etc.), Coulomb electrostatics, and van der Waals terms, as implemented in the CHARMM27 force field [85]. The high-resolution force field is all-atom MM/PBSA, including CHARMM27 molecular mechanics terms of internal energy and van der Waals, Poisson-Boltzmann (PB) electrostatics, and nonpolar solvation terms (solvent-accessible surface area, or SA-dependent). More details on the energy calculations can be found in [42, 17]. The energy difference between a mutant and a wild type is annotated as $\Delta\Delta G.$ where $\cdot$ can be folc, fold, or bind.

2.3.2.3 Results

Protein-protein interactions (PPIs) are known to be important to DNA damage and repair (DDR). For instance, a supercomplex BASC (BRCA1-associated genome surveillance complex) plays both a sensory and effector role in DDR, and it involves a group of tumor suppressors and DDR proteins such as BRCA1, MSH2, MSH6, MLH1, ATM, BLM, PMS2, and the MRE11-RAD50-NBN protein (MRN) complex [86]. Protein-protein interaction networks for core genes of three pathways are shown in Fig. 2.5. Moreover, many missense mutations are found in protein regions responsible for PPI (see examples of BRCA1, BRCA2, and RAD50 in Fig. 2.6). Mutational effects and molecular mechanisms of missense SNVs on PPIs are thus examined by structure-based computational mutagenesis experiments. These experiments are for 446 missense mutations of 16 genes (including 15 core genes for DDR) where complex structures formed through PPI are available, and wild-type residue types agree with those in the structures. Considering that a protein can interact with various partners, including homo-oligomerization with itself, there are 547 such mutagenesis experiments in total (Fig. 2.7 for statistics of breakups on genes), including 155 (over one third) mutations at protein-protein interfaces (defined by a 6 Å cutoff between heavy atoms) or elsewhere.



(a) BER pathway

(b) NHEJ pathway

(c) HDR pathway

Figure 2.5: Protein-protein interaction networks for core genes of three pathways: HDR, NHEJ and BER (source: STRING-db with 90% confidence level). Edges are colored according to the type of interaction evidence: cyan and purple for known interactions from curated data and experimental validation, respectively; green, red, and blue for predicted interactions based on gene neighborhood, gene fusion, and gene co-occurrence, respectively; limon for textmining; black for co-expression; and light blue for homology.

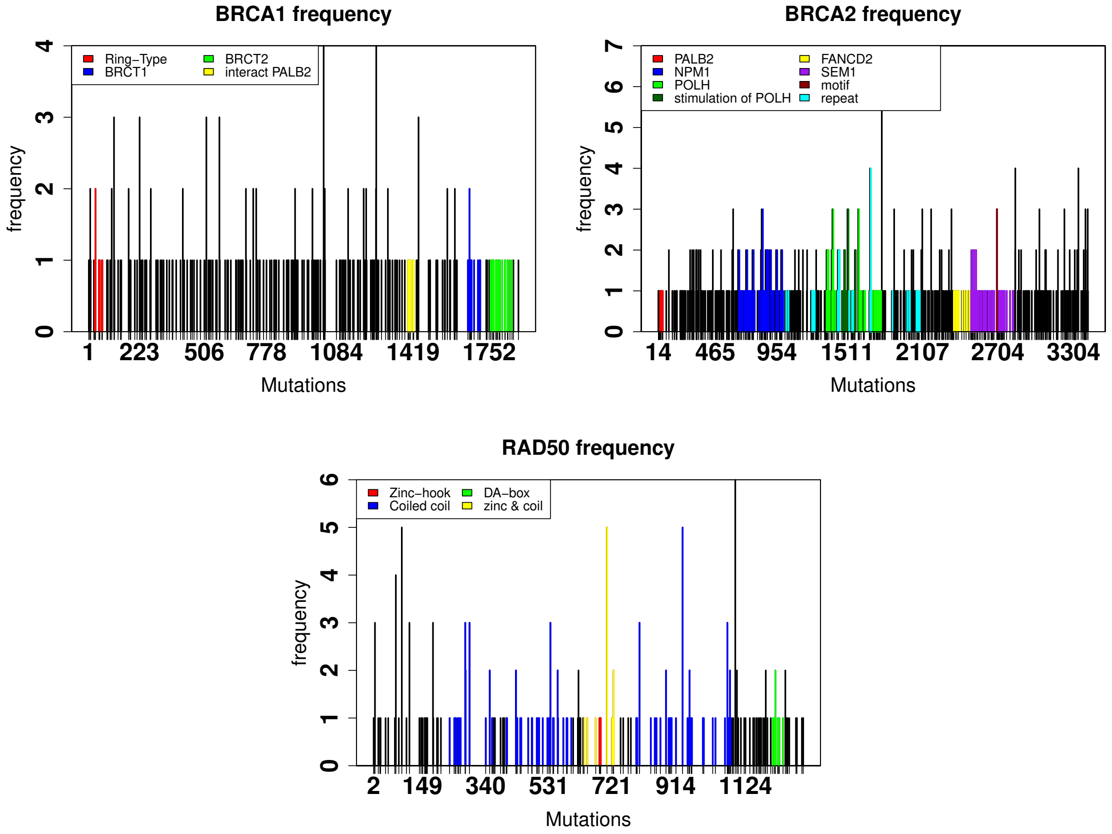


Figure 2.6: The location (protein annotation based on uniprot) and occurrence of SNVs for A. BRCA1, B. BRCA2, and C. RAD50.

Structural analysis indicates that these missense mutations are enriched at protein surfaces as well as at PPI interfaces. Energetic analysis indicates that the majority of these mutations, non-interfacial or interfacial, destabilize protein-protein complexes (Figures 2.8 and 2.9). For interfacial mutations, energy decomposition into contributions from individual protein folding and protein-protein binding indicates that (1) corresponding protein-protein complex destabilizations are dominated by individual proteins being destabilized (Fig 2.8(a)); and (2) most of the corresponding PPIs are also weakened or even disrupted upon mutation (Fig. 2.9), and those few strengthened PPIs rarely overpowered protein destabilization to rescue the stability of the protein-protein complex. Interestingly, although rare, some interfacial mutations are found to have neutral or even positive effects on protein stability but still end up destabilizing the resulting protein-

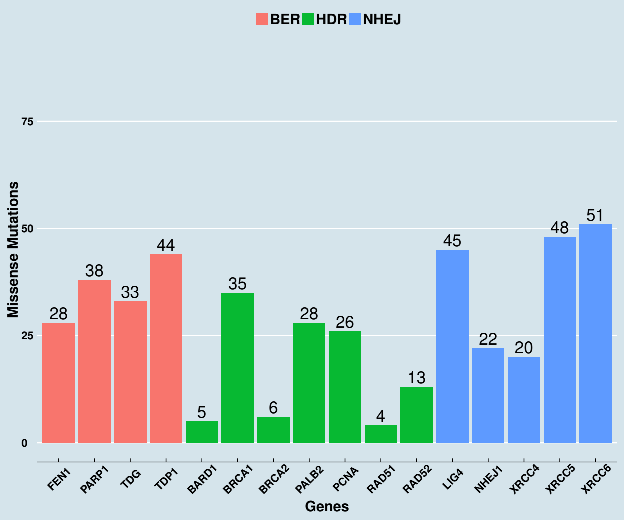


Figure 2.7: The number of missense SVNs examined for 15 core genes and PCNA. PCNA is classified BER according to the interaction partner FEN1.

protein complexes due to much-disrupted binding between proteins. Similar observations were also made experimentally for PALB L939W and T1030I mutants in a previous study [87].

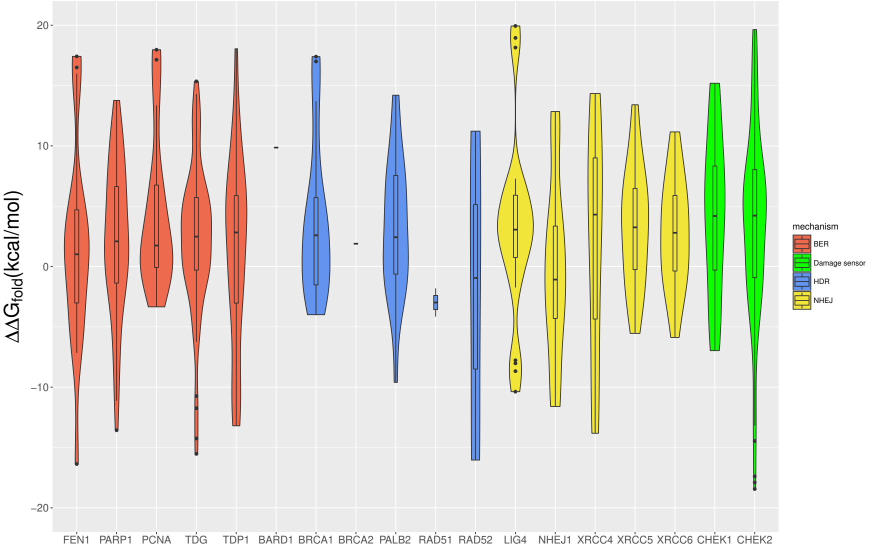
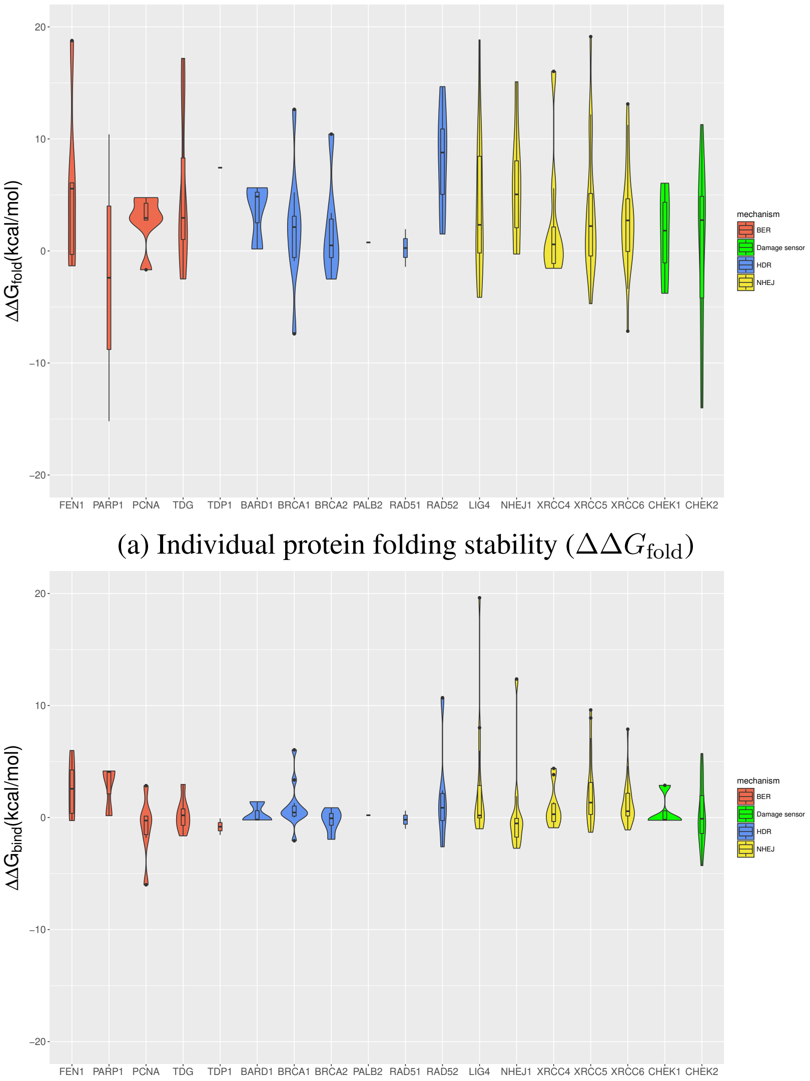


Figure 2.8: Violin plots of energy difference upon non-interfacial mutation for individual protein folding stability ($\Delta\Delta G_{\text{fold}}$).

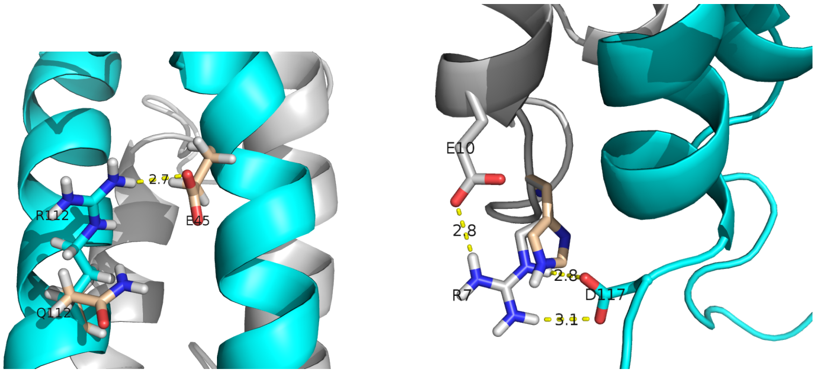


(a) Individual protein folding stability ($\Delta\Delta G_{\mathrm{fold}}$)

(b) Protein-protein binding ($\Delta\Delta G_{\mathrm{bind}}$)

Figure 2.9: Violin plots of energy difference upon interfacial mutation for individual protein folding stability ($\Delta\Delta G_{\mathrm{fold}}$) and protein-protein binding ($\Delta\Delta G_{\mathrm{bind}}$).

A couple of examples of mutations in RING-RING domain interactions between BRCA1 and BARD1 illustrate how a loss of intra- or inter-molecular interactions destabilize a protein or weakens protein-protein interaction. RING domains of BRCA1 and BARD1 form a heterodimer as part of the aforementioned BASC. The heterodimer enables ligase activity, whereas neither monomer is structurally stable or supports ligase activity alone [88]. In the first example (Fig. 2.10(a)), R112Q of BARD1 is far from the interface with BRCA. The wild-type residue is a positively charged, long arginine that forms favorable electrostatic and van der Waals interactions with a negatively charged glutamic acid (E45) within the same molecule BARD1; and such interactions stabilize helix-helix packing within BARD1. The mutation of arginine into a polar and shortened glutamine weakens the intra-molecular helix-helix packing and thus destabilizes BARD1. In the second example (Fig. 2.10(b)), R7H of BRCA1 is right at the interface with BARD1. The wild-type residue is again arginine forming favorable electrostatic and van der Waals interactions with two negatively charged residues: a glutamic acid (E10) of the same molecule BRCA1 and an aspartic acid (D117) of the binding partner BARD1. Its mutation into a polar and smaller histidine weakens both intra- and inter-molecular interactions, thus destabilizing the BRCA1-BARD1 complex by both destabilizing BRCA1 and disrupting BRCA1-BARD1 interaction. We hypothesize that a mutation of either negatively charged residues interacting with R7 would cause similar effects in cancers. Indeed, E10K, the substitution of a negatively charged glutamic acid with a positively charged lysine, was found in patients with ovarian and breast cancers [89]. The reciprocal approach could potentially generalize to obtain functional annotation and mechanistic understanding of more missense mutations in cancers.



(a) R112Q mutation in BARD1

(b) R7H mutation in BRCA1

Figure 2.10: Structural visualization of missense mutations in BRCA1 (gray cartoon) and BARD1 (cyan cartoon) RING-RING domain interaction. a) R112 of BARD1 (cyan sticks) mutates into glutamine (wheat sticks) and destabilizes BARD1 by weakening intramolecular helix-helix interaction especially with E45; (b) R7 of BRCA1 (gray sticks) mutates into histidine (wheat sticks), weakens interactions with both E10 of BRCA1 and D117 of BARD1, and leads to both destabilized BRCA1 and disrupted BRCA1-BARD1 interaction.

2.3.3 Engineering a HER2-specific antibody-drug conjugate to increase lysosomal delivery and therapeutic efficacy

2.3.3.1 Introduction

Conventional cancer treatment, such as chemotherapy and radiotherapy are not always safe and efficacious. Therefore alternative treatments are urgently needed [90]. Antibody-drug conjugates (ADCs) can bring forth a new generation of cancer treatment with the high specificity of antibodies and the potent cytotoxicity of drugs [91, 92]. However, for ADC-based therapeutic to be effective, it needs a high dose of ADCs. Even though a high dose of ADCs will increase the payload delivery, but it will also increase the unacceptable toxicities and side effects towards normal cells [93, 94]. A possible way to overcome this limitation is to have more efficient payload delivery to the targeted cells. "Acid-switched" ADCs with tight binding at neutral pH (pH=7.4) outside of the cell and weak binding in acidic pH (pH=5.8) inside of the cell has been developed [95] for the treatment of HER2-overexpressing tumors in breast cancer. Specifically, An acid-switched ADC with S55H and G57E ('SG') mutations have been found through phage display to have very loose binding at acidic pH inside of the cell and almost the same binding outside of the cell as wild type. We utilized our iCFN [17] model to analyze and discover the mechanism behind the acid-switched property of SG mutations. In this application, we have formulated the problem as a multi-state multi substate optimization problem.

2.3.3.2 Method

The X-ray crystallographic structure of the HER2–pertuzumab complex [96] (Protein Data Bank accession code: 1S78) was minimized using the molecular-modeling software CHARMM v.36 [85] in a CHARMM22 force field, and all missing residues and atoms were added [11]. PROPKA v.3.1[97] was used for empirical pKa prediction and protonation state determination for the minimized structure at pH 7.4 and 5.8, respectively. The resulting structure at either pH value was minimized for 5,000 steps with explicit solvent molecules and then subjected to a 10 ns molecular dynamics simulation using the computer program NAMD2 v.2.10 [11]. Molecular dynamics protocols followed similar details as described previously [11]. Snapshots were retained at 0 ns (the beginning) and every 1 ns from 6 to 10 ns to represent a conformational ensemble of the HER2–pertuzumab complex at either pH.

Structures of HER2 in complex with the SG mutant were modeled using iCFN for multi-state protein design with substate ensembles customized for pH dependence. First, the positive and negative substates were defined as conformational ensembles of the complex at pH 7.4 and 5.8, respectively. Substate energies were folding stabilities of the complex that include Coulomb electrostatics, van der Waals, calculated internal energies, and a nonpolar contribution to the hydration free energy based on solvent-accessible surface area (SASA) [11]. A positive-substate stability cutoff was set at 10 kcal mol^{-1}, and positive-versus-negative substate specificity was essentially not mandated with a cutoff of 1,000 kcal mol^{-1}. Second, pertuzumab VH residues 55 and 57 were allowed to change to any amino acid, including histidines in $\delta-$, $\epsilon-$, and doubly protonated states. Residues within 5 Å of the mutated amino acids were allowed to be flexible. In addition, His245 of HER2 was allowed to be in any of the three possible protonation states, and Asp50 of pertuzumab VH was allowed to be deprotonated or singly protonated at either oxygen atom on the side chain. Top conformations of each sequence-protonation combination in each substate (backbone conformation here) generated from iCFN for either pH were geometrically grouped into representatives and then screened by PROPKA for protonation states. Later, complex-folding stabilities (G) and binding affinities (ΔG) of the top sequence-conformation ensembles retained for either pH were reevaluated and reordered with a higher-resolution energy model where continuum electrostatics replaced Coulombic electrostatics [17]. Lastly, the representative conformation/protonation at either pH was chosen based on the best binding affinity (lowest ΔG) amongst those whose folding energies were within 1 kcal mol^{-1} of the most stable complex.

Each calculated binding energy relative to the wild type, $\Delta\Delta G$, was further decomposed into contributions of van der Waals, continuum electrostatics, SASA-dependent nonpolar solvation interactions, and internal energy. Van der Waals and electrostatics were found to be the most important contributors to the pH-dependent binding. The calculated van der Waals contribution was largely due to the modeled clashes between the bulkier pertuzumab His55 and a HER2 loop at pH 5.8 whereas the flexible HER2 loop could largely ameliorate the clashes. Therefore, long-range electrostatic binding affinity, being less sensitive to conformational details, was further analyzed as an origin of the pH-dependent binding. Specifically, continuum electrostatics for binding were first decomposed into the lost desolvation penalties' contributions, and the gained solvent-screened intermolecular interactions. Desolvations were then decomposed into contributions of all residues, and screened intermolecular electrostatic interactions were decomposed into those of all residue pairs. To sum up, screened intermolecular electrostatic interactions and avoided double-counting for each residue, half of the pairwise interactions for each residue with all other residues were considered. Those residues with insignificant contributions (desolvation plus the half of the summed pairwise interactions) to electrostatic binding specificity $\Delta\Delta\Delta G (\Delta\Delta\Delta G = \Delta\Delta G_{\mathrm{pH\,7.4}} - \Delta\Delta G_{\mathrm{pH\,5.8}})$ were not considered further. Since desolvation penalties dominated over screened electrostatic interactions in contributions to electrostatic binding specificity $\Delta\Delta\Delta G$, the top ten remaining residues with the highest desolvation contributions were reported for $\Delta\Delta G_{\mathrm{pH\,7.4}}$, $\Delta\Delta G_{\mathrm{pH\,5.8}}$ and $\Delta\Delta\Delta G$, respectively.

2.3.3.3 Results

iCFN correctly predicts the sign of binding affinity and specificity at pH 7.4 and 5.8 in comparison to the experimental values. More specifically, it correctly predicted for SG mutations that 1) binding affinity at pH 7.4 ($\Delta\Delta G_{\mathrm{pH\,7.4}}$) would be slightly worse in comparison to the wild type (see Fig. 2.11 A left bar plots); 2) binding affinity at pH 5.8 ($\Delta\Delta G_{\mathrm{pH\,5.8}}$) would be significantly weakened at pH 5.8 (see Fig. 2.11 A middle bar plots); 3) binding specificity would be negative ($\Delta\Delta\Delta G < 0$) which means the binding is more favorable at pH 7.4 (outside of the cell) rather than pH 5.8 (inside of the cell). Also, based on energy type decomposition, we find out that van der Waals and electrostatics are the most important for the specificity. Since van der Waals interactions are short-range and more sensitive to small fluctuations in structural modeling, we focused on electrostatics with longer-range interactions. Further decomposition of electrostatic energies to desolvation penalty and pairwise interactions (see Fig. 2.11 B) indicates that the desolvation penalty is the main contributor to the specificity.

With the further positional decomposition of electrostatic energies, we discover that His55 in CDRH2, Glu57 in CDRH2, and Tyr252 of HER2 are playing a pivotal role in understanding the mechanism behind the increased specificity (see Fig. 2.11 B). These analyses resulted in a model in which protonation of His55 in CDRH2, that replaces Ser55 in wild-type pertuzumab, at Nϵ of the imidazole ring predominates over that of Nδ1 at near-neutral pH, whereas the protonated Nδ1 tautomer is favored at acidic pH (5.8). Protonation of Nδ1 leads to electrostatic interaction with Glu57 in CDRH2 of SG in combination with interactions with Tyr252 of HER2. In contrast to the solvent exposure of Glu57 and Tyr252 in the SG–HER2 complex at pH 7.4, the change in the protonation state of histidine at pH 5.8 results in the reorientation of Glu57 and Tyr252, leading to an unfavorable desolvation penalty that is a primary contributor to the loss of affinity of SG at acidic pH.

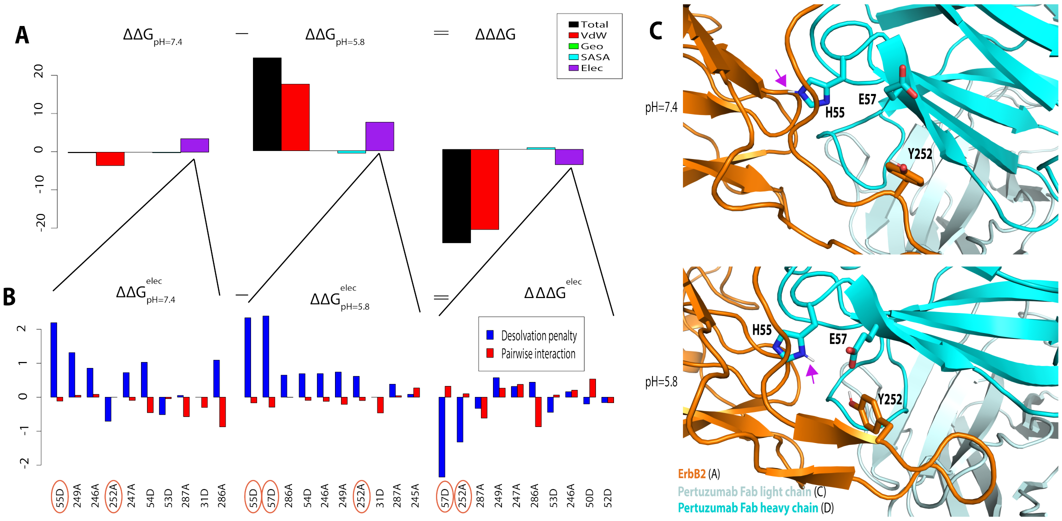


Figure 2.11: The X-ray crystallographic structure of the pertuzumab:HER2 extracellular domain complex (Protein Data Bank: 1S78) is shown, with the pertuzumab VH domain, VL domain, and HER2 domain II shown in cyan, light cyan, and orange, respectively. Residues targeted for mutagenesis by histidine replacement (a) or for the production of phage display libraries (b) are shown in green. (c) Residues Tyr55 (CDRL2) and Ser103 (CDRH3) mutated to histidine to generate the YS mutant are shown in green. (d) Residues Ser55 (CDRH2) and Gly57 (CDRH2) that were mutated to His and Glu, respectively, to generate the SG mutant are displayed in green.

2.3.4 Anticipating Cancer Mutations and unraveling biological mechanism for Estrogen Receptors

2.3.4.1 Introduction

ERα is a member of the nuclear receptor family and plays a key role in the prevention and treatment of the majority of breast cancers [10]. Two main classes of therapeutics are 1) suppression of estrogen production through aromatase inhibitors; 2) Direct inhibition of ERs through selective ER modulators (SERMs) or selective ER degraders (SERDs) [98]. Although these therapeutics are effective in many cases, resistance often occurs after long-term exposure to these drugs [11, 12]. Anticipating resistant mutations and understanding the mechanism underlying are the key steps to developing drugs with lower resistance profiles. In this application, we utilized the multi-state multi substate formulation in iCFN [17] to predict and anticipate the resistant mutations. In addition, we developed energy decomposition approaches to understand and discover the underlying biological mechanism behind them.

2.3.4.2 Method

The X-ray crystallographic structures of dimer ERα were taken from Protein Data Bank [99] based on the list provided in the Uniprot [100] for Estrogen receptor with Uniprot ID: P03372. Twenty structures with the least missing residues have been chosen for active and inactive states of ERα (10 for each state) that are shown in Table 2.8. The selected 3D structures were minimized using the molecular-modeling software CHARMM v.36 [85] in a CHARMM22 force field, and all missing residues and atoms were added [11].

3D Structures of dimer ERα were modeled through iCFN for multi-state protein design with substates ensemble for complex-folding specificity. Positive and negative substates were defined as conformational ensembles ERα in agonist (active) and antagonist (inactive) states (see Table 2.8), respectively. Substate energies were folding stabilities of the dimer ERα complex that include Coulomb electrostatics, van der Waals, calculated internal energies, and a nonpolar contribution to the hydration free energy based on solvent-accessible surface area (SASA) [11]. A positive sub-

Active substrate (+)			Inactive substrate (-)		
PDB	Mut.	Agonist Ligand ID	PDB	Mut.	Antagonist Ligand ID
1GWQ	WT	ZTW	3ERT	WT	OHT
3HLV	Y537S	J2Z	2OUZ	WT	C3D
2QAB	Y537S	EI1	1YIM	WT	CM4
2QGW	Y537S	EES	1XP9	WT	AIJ
2QGT	Y537S	EED	1XP1	WT	AIH
4IWC	Y537S	1GV	1SJ0	WT	I0G
3L03	Y537S	4OH	1YIN	WT	E4D
2FAI	Y537S	459	3DT3	WT	369
2B1V	Y537S	458	3OS8	L372R/L536S	KN0
2Q6J	Y537S	A48	1YIM	WT	CM4

Table 2.8: List of active and inactive Conformational ensemble for ERα.

state stability cutoff and positive-versus-negative substate specificity cutoff were set at 10 and 100 kcal mol^{-1}, respectively. At first, known residues (E380, L469, V534, L536, Y537, and D538 in both chains of ERα) with known cancerous substitution mutations [11] has been considered as mutant and were allowed to be mutated to any amino acids. At the next step, all the residues in helix 12 and its neighboring loop (528-547), excluding the previous ones, have been considered as mutant and were allowed to be mutated to any amino acids. Residues within 5 Å of the mutated amino acids were allowed to be flexible. Top conformations of each sequence in each substate (backbone conformation here) generated from iCFN for either state (active and inactive) were geometrically grouped into representatives. Later, complex-folding stabilities (G) and binding affinities (ΔG) of the top sequence-conformation ensembles retained for either state were reevaluated and reordered with a higher-resolution energy model where continuum electrostatics replaced Coulombic electrostatics [17]. Lastly, the representative conformation at either state was chosen based on the best complex-folding stabilities (lowest G).

Each calculated complex-folding stability energy relative to the wild type, $\Delta\Delta G$, was further decomposed into contributions of van der Waals, continuum electrostatics, SASA-dependent nonpolar solvation interactions, and internal energy. Each of these energy terms further analyzed to discover the origin of resistant mutations (specificity toward agonist state rather than antagonist).

Specifically, 1) continuum electrostatics for complex-folding stability was first decomposed into the lost desolvation penalties' contributions and the gained solvent-screened intermolecular interactions. Desolvations were then decomposed into contributions of all residues, and screened intermolecular electrostatic interactions were decomposed into those of all residue pairs. To sum up, screened intermolecular electrostatic interactions and avoided double-counting for each residue, half of the pairwise interactions for each residue with all other residues were considered. 2) internal energy for complex-folding stability were decomposed into contributions of all residues; 3) van der Waals energy for complex-folding stability were decomposed into contributions residue pairs. To sum up, screened van der Waals interactions and avoided double-counting for each residue, half of the pairwise interactions for each residue with all other residues were considered; 4) SASA-dependent nonpolar solvation interactions for complex-folding stability were decomposed into contributions of all residues.

Several hypotheses have been proposed to discover the biological mechanism underlying several known cancerous mutations through the energy decomposition. Secondary mutations have been introduced to shift the complex-folding stability energies toward antagonist (inactive) state rather than agonist (active) state to validate the proposed hypotheses. iCFN with multi-state protein design and substates ensemble formulation has been used for modeling ERα with secondary mutations. Positive and negative substates were defined as conformational ensembles of ERα in antagonist (inactive) and agonist (active) states, respectively. The same protocol (energy terms, cutoff values, etc.) was utilized for modeling the secondary mutations. We specifically considered mutation E380Q with less knowledge about its biological mechanism for secondary mutation introduction for proof of concept. Based on the pairwise energy decomposition of E380Q mutation on continuum electrostatics, three candidate residues (C381, E542, M543) were identified. The secondary mutation has been introduced on these candidate residues, and they were allowed to be mutated to any amino acids.

For prospective design, a machine learning model is trained to predict whether a mutation is cancerous or not. Since the complex structure of ERα and 17β-estradiol (E2) (known as estrogen response elements (EREs)) plays a critical role in the growth of breast cancer[101], ERE-luciferase experiments have been conducted to measure activation of estrogen genes. ERE-luciferase experiment reports their measure through the Relative light unit ratio (RLU). The high value of RLU is representative of high activation (or high gene expression). We consider $y = -\log(\frac{\mathrm{RLU(mt)}}{\mathrm{RLU(wt)}})$ as a ground through label. Approximated entropy and enthalpy have been considered as features for the machine learning model since they are sufficient to calculate Gibbs free energy. Approximated enthalpy has been calculated through complex-folding stability energy relative to wild type ($\Delta\Delta G$). Entropy (S) of a substate has been approximated through the logarithm of the number of conformation within 1 kcal mol^{-1} of the conformation with the lowest energy. Finally, the relative entropy of a mutation to wild type is approximated as:

$$\Delta\Delta S(mt) = \frac{1}{P}\sum_{i=1}^{P}(S_i(mt) - S_i(wt)) - \frac{1}{Q}\sum_{j=1}^{Q}(S_q(mt) - S_q(wt)) \tag{2.17}$$

where P and Q are the numbers of positive and negative states. There are only 11 mutations with known experimental RLU available. Therefore, we employed linear regression with two features (approximated entropy and enthalpy) to avoid over-fitting. Finally, We trained the linear regression model with all the samples. Since helix 12 plays a crucial role in the estrogen receptor's functionality, we further analyzed all the residues in the helix (res. 528-547) to anticipate novel, unseen, and potential cancerous mutations. Based on the linear regression with two entropy and enthalpy features, which have been explained in the previous section, we predict the $\hat{y}(\cdot)$ for all of the mutations for all the residues in helix 12 and its neighboring loop. $\hat{y}$(E380Q)) ≈ -0.5 and complex-folding stability energy $\Delta\Delta G = 0$ have been as considered as a cutoff to filter mutations with no indication of being cancerous. Therefore, mutations with either $\Delta\Delta G > 0$ or $\hat{y}$(E380Q)) > -0.5 were filtered. Furthermore, to better analyze the remaining mutations, we categorize them in three perspectives:

- **Predicted** $\hat{y}(\cdot)$. In this criteria, remaining mutations have been grouped as 1) strong activating: if their predicted $\hat{y}(\cdot)$ < -1.5 and labeled as 1 2) medium activating: if their predicted

$\hat{y}(\cdot) \in [-1.5, -0.5]$ and labeled as 0.

- **Mechanism**. Let's assume complex-folding stabilities for positive and negative states are represented as ΔG^{+} and ΔG^{-}. Then in these criteria, the remaining mutations have been grouped as 1) $\Delta G^{-} > \Delta G^{+} > 0$: when both antagonist and agonist conformations are becoming unstable but antagonist more. This mechanism is labeled as mechanism 1; 2) $\Delta G^{-} > 0 > \Delta G^{+}$: when the antagonist is unstable, and the agonist is more stable in comparison to wt. This mechanism is labeled as mechanism 2; 3) $0 > \Delta G^{-} > \Delta G^{+}$: when both agonist and antagonist conformations are more stable compared to wild type but agonist more. This mechanism is labeled as mechanism 3.

- **Main contributing energy term.** In this criterion, remaining mutations have been grouped based on which type of energy term is the most contributing one. Then, They are labeled as a) Van der Waals: labeled as 1; b) Continuum electrostatics: this labeled as 2; c) internal energy: labeled as 3; d) solvent accessible surface area (SASA): labeled as 4.

Three categorical features for each mutation have been created based on the categories as mentioned earlier. Then, the k-modes clustering algorithm [102] (with $K = 16$) has been performed for clustering the remaining mutations. k-modes clustering algorithm is an extension of the k-means clustering algorithm [103] for categorical data. The number of mismatches of the features between two mutations has been considered the distance in k-modes clustering.

2.3.4.3 Results

Predicting potential cancerous mutations for known residues.

At first, iCFN has been tested on 6 known cancerous residues E380, L469, V534, L536, Y537, and D538 in Estrogen receptors. iCFN correctly predicted 9 out of 11 known cancerous mutations for these residues, shown in Figure 2.12. In Figure 2.12, if a mutation is above the $y = x$ line (when $\Delta G^{-} > \Delta G^{-}$) then it will be predicted as cancerous mutation. Y537S is the most activating mutation without 17β-estradiol (E2) based on the ERE-luciferase experiment, shown in Figure 2.13 (B). It is also predicted as the most activating mutation among all the Y537 single nucleotide polymorphism (SNP) mutations. Moreover, other high activating mutations Y537D/N/C are following Y537 both in our prediction and in the ERE-luciferase experiment. Similarly, E380Q, L469V, D538G, and V534E are among the activating mutation which has correctly been predicted. However, for residue L536 mutations L536Q, L536R has been misclassified inactivating mutation where the L536H has been correctly classified. The correctly predicted mutations are grouped to mechanism 1 (E380Q, D538G), mechanism 2 (Y537S/N/D/C), and mechanism 3 (L469V, V534E, L536H). Y537S/N/D/C are among the most activating mutations based on the ERE-luciferase experiment, all classified as the strongest activating mechanism (mechanism 2).

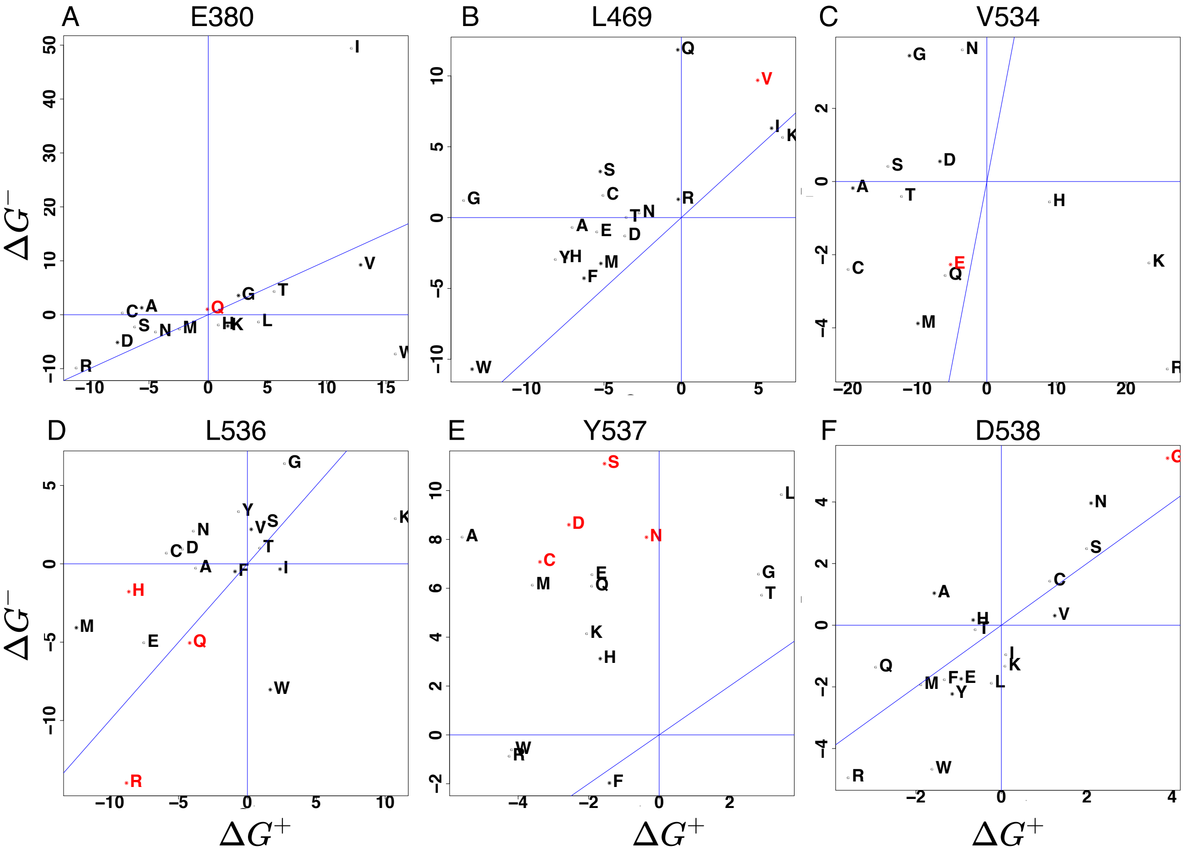


Figure 2.12: Prediction of cancerous mutation for 6 known cancerous residues in ERα through iCFN. The known cancerous mutations are shown in red. SNP mutations are shown as star.

High correlation between in silico prediction and in vitro experiments.

iCFN has shown to have high accuracy in identifying activating mutations (9/11). However, mutations are in a continuous spectrum of activation rather than only being activated or not. Figure 2.13 (B) shows the strength of activation for each of the mutants based on the ERE-luciferase experiment in the presence or absence of the E2 hormone [1]. Figure 2.13 (A) shows the high correlation (66%) between the experimental and predicted RLU based on the trained linear regression with two features (approximated entropy and enthalpy). Therefore, it shows that not only our model can correctly predict a mutation being activating mutation or not but also can predict the strength of its activation. Based on Figure 2.13 (A), we can observe that there are two groups of activating mutations 1) low RLU: L536R/Q, E380Q and V534E which their RLU are above than -1 2) high RLU: Y537C/S/N/D, L536H and D538G which their RLU are below -1. And all the predicted high RLU mutations are the activating mutations with very high strength (experimental RLU is below -1). Y537S and V534E are the most far away from the predicted best linear model based on the predicted RLU. V534E, even though being an activating mutation, has not shown any activation elevation in their ERE-luciferase experiment compared to wild type based on Figure 2.13 (B). V534E is an exception in the previous studies [11]. Y537S has the most strength of activating mutation, which is very different from other activating mutations. It is intuitive for being most far away from the best-fitted line, which represents all the mutations.

Size reduction is one of the main reasons for activating mutation at Y537.

To determine the nature of the energy term deriving the activation for the mutations in Y537, energy decompositions have been performed, shown in Figure 2.14. Energy decomposition has been performed for the agonist state (middle bar plots), antagonist state (right barplot), and their specificity (left barplot). Total energy (black bar plot) is decomposed to Van der Waals (red), Electrostatics (green), SASA (purple), and geometry (cyan). For agonist and antagonist bar plots, each of the bars are relative with respect to wt. Therefore, if the bar is positive, it means its energy is higher than wt. Specificity bar plots are the difference between agonist and antagonist states. Van der Waals energy is the common deriving energy force for all of the Y537 mutations Based on the specificity part of Figure 2.14. Van der Waals energies for all mutations in agonist state are

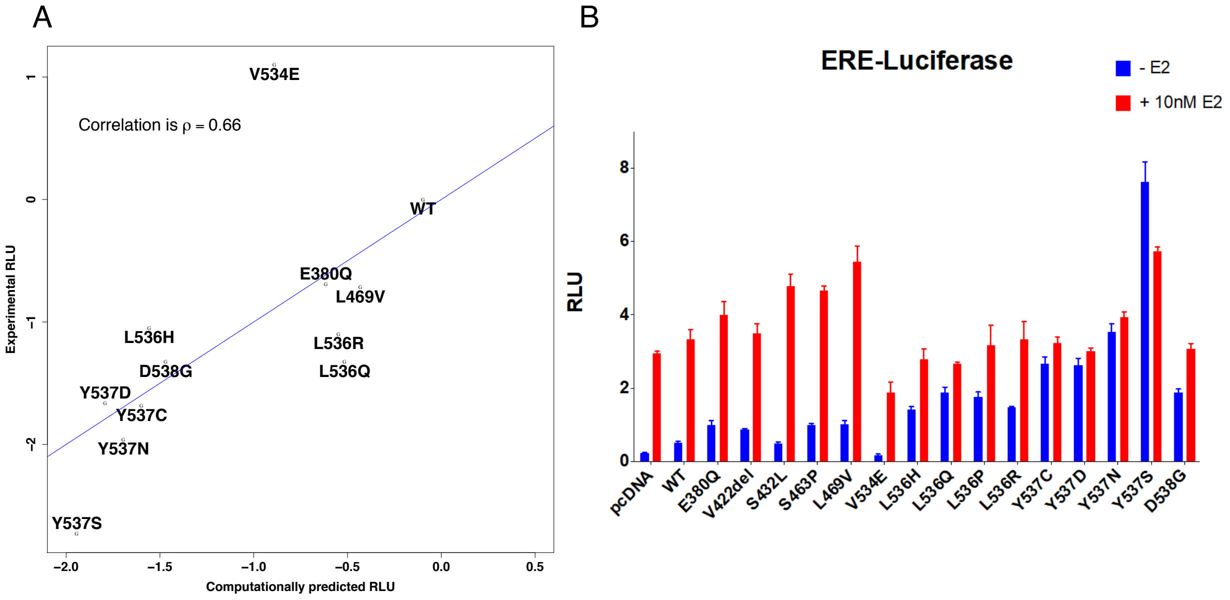


Figure 2.13: A) Predicted RLU (based on linear regression) versus experimental RLU for known activating mutations. B) ERE-luciferase experiment to identify the activation strength of each mutation in presence or absence of E2 hormone [1].

very small. However, their antagonist state values are very high (very unfavorable in comparison to wild type). The main question is why their values are high in the antagonist state and why not in the agonist state. We hypothesize that 1) helix 12 conformational change from agonist to antagonist; 2) the drastic change in the size of mutation compared to wild type are the main factors. More specifically, all four mutations (S/C/N/D) are becoming smaller compared to the wild type (Y) in size. Therefore, Van der Waals packing is affected in the antagonist state (in an unfavorable way). Due to the spatial change of Helix 12 in agonist in comparison to the antagonist (shown in Figure 2.15), in the agonist state, Y537 is in the surface and not buried. Therefore the size reduction does not affect it. But in the antagonist state, the Y537 is buried and faces other residues such as E380 and H377. Therefore, the size reduction from wild type to the mutants is affecting its Van der Waals packing.

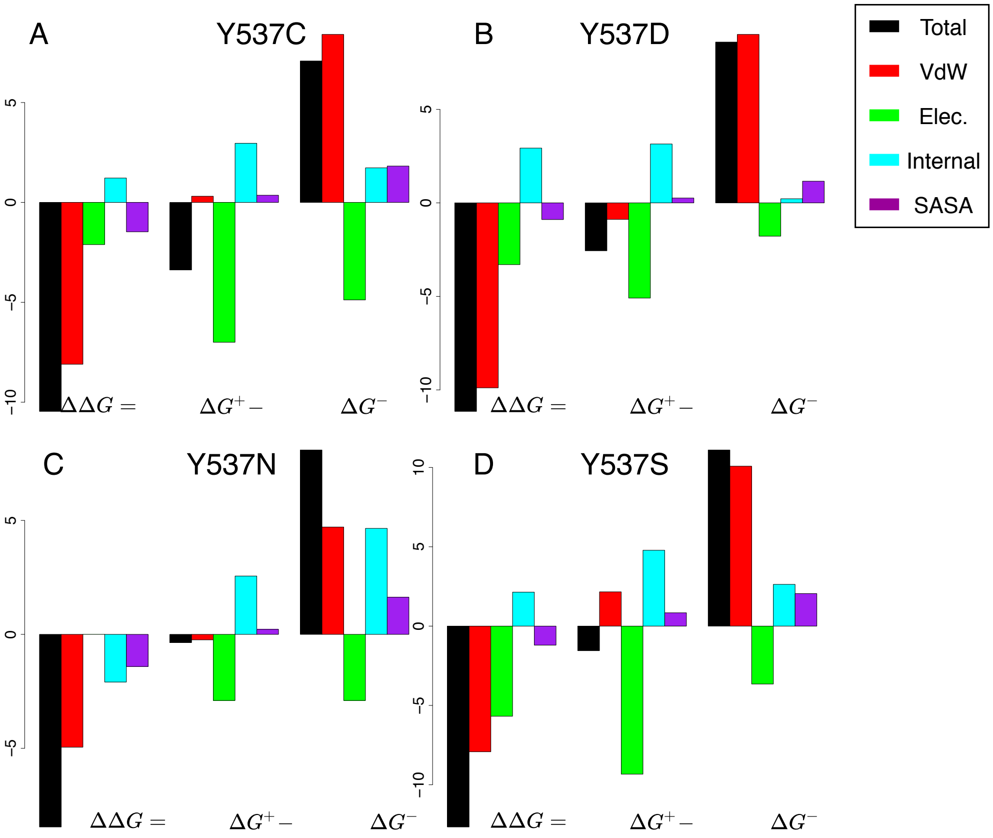


Figure 2.14: Energy decomposition for known activating mutations in Y537 of ERα.

Hydrogen bond in agonist is the driving force for Y537S.

Electrostatics is the second most important energy term for making Y537S cancerous, shown in Figure 2.14. Based on Figure 2.14, Electrostatics is favorable for both agonist and antagonist compared to the wild type. However, it is much more favorable for the agonist. Furthermore, we investigate which type of energy and which residues are the main contributor to the Electrostatics. We find out that by mutating from Y to S, a strong hydrogen bond has been created between S537 and D351 in agonist shown in Figure 2.16 (A). Moreover, in the antagonist state neither in wild type nor in the mutant, there is no hydrogen bond between S537 and D351. The main reason for the existence of hydrogen in the agonist state is the structural change of helix 12 in Estrogen receptors. Helix 12 is in the vicinity of D351 in agonist and is far away from D351 in the antagonist, shown

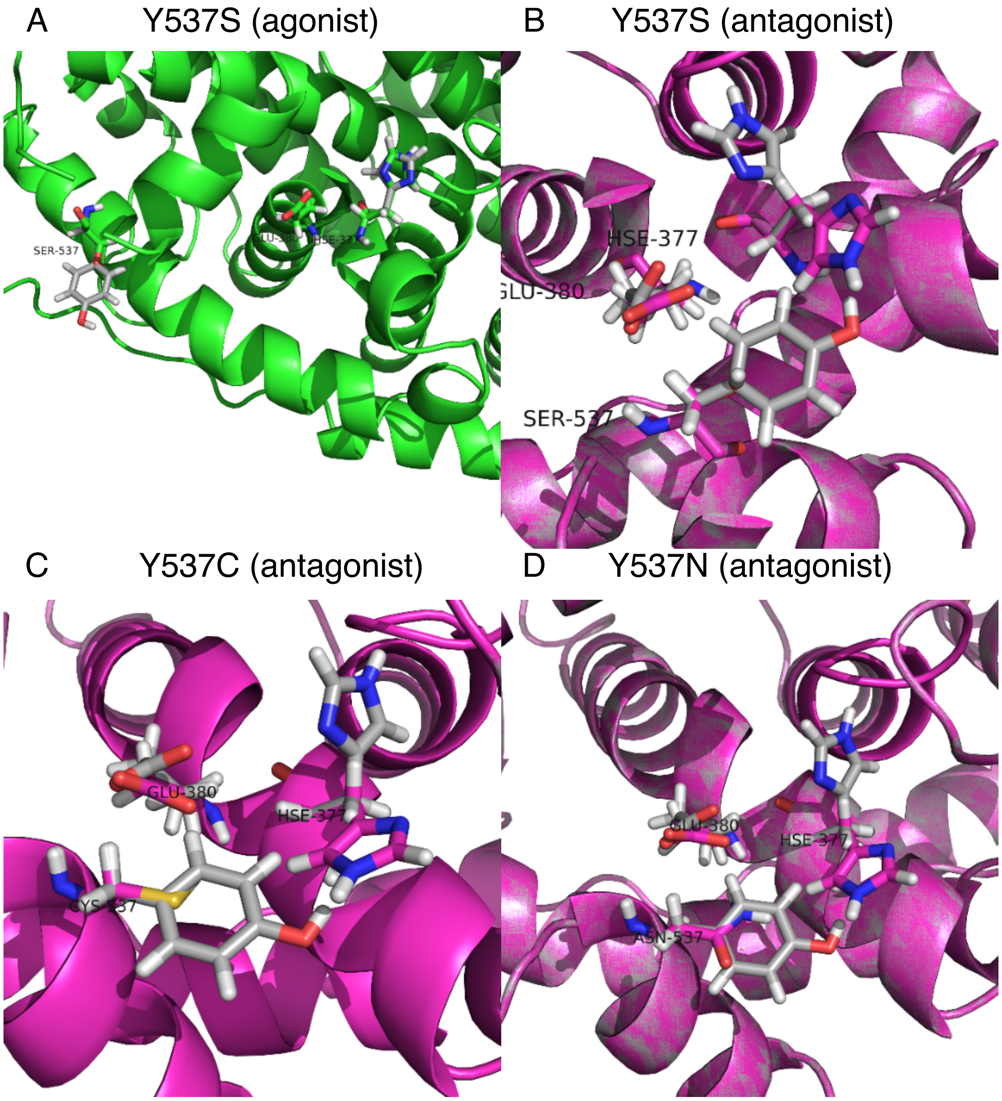


Figure 2.15: Effect of size reduction and spatial change of helix 12 in Van der Waals packing for Y537 mutations.

in Figure 2.16 (B). The hydrogen bond being favorable to the agonist conformation rather than antagonist conformation in Estrogen receptors' Y537S mutation has been validated previously in [11].

Internal energy is the driving force for D538G.

Similarly, to understand the mechanism behind D538G mutation, energy decomposition has been conducted and shown in Figure 2.17 (A). Figure 2.17 (A) shows that internal energy is the main energy term for deriving the D538G mutation. To further understand which residues con-

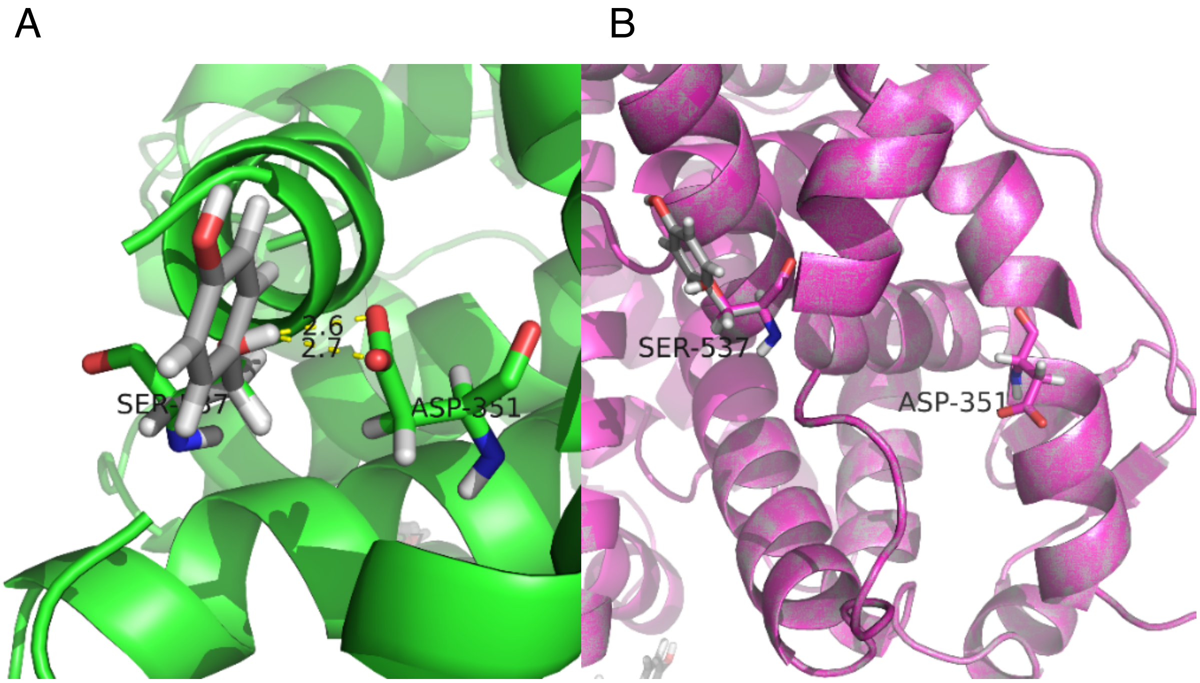


Figure 2.16: Conformation of Y537S mutation in agonist (A) and antagonist (B) states.

tribute most to the internal, we performed residue level energy decomposition and found out G538, Y537, and E380 are contributing the most. The contribution of each residue is colored from blue (no contribution) to red (very high contribution) and shown in figure 7 (b). Interestingly, the three most frequent mutating residues (538, 537, and 380) are chosen to contribute the most to D538G being a cancerous mutation.

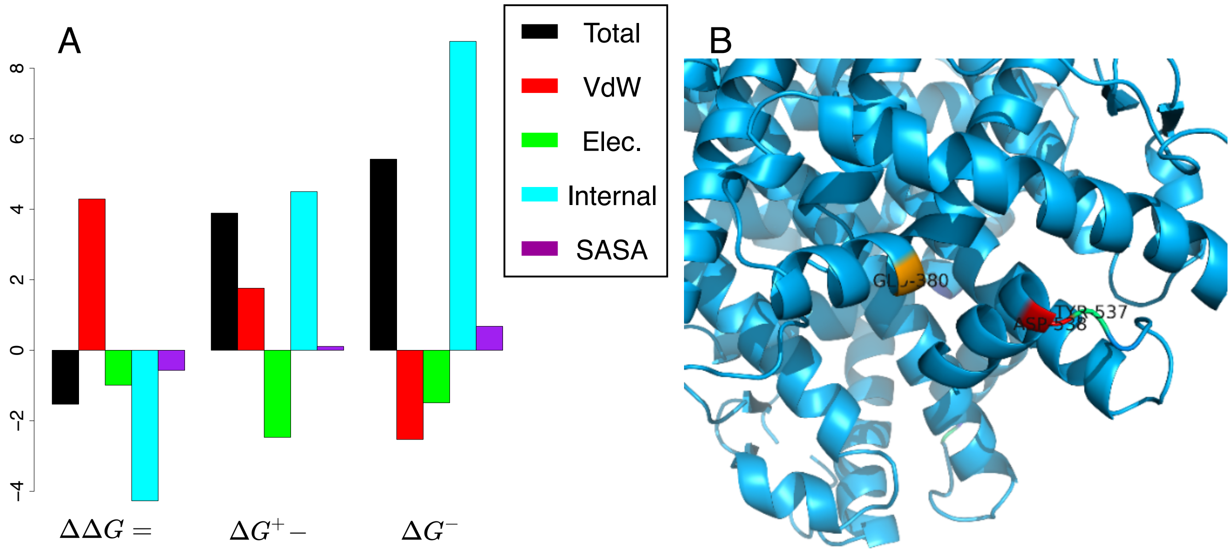


Figure 2.17: (A) Energy decomposition of D538G mutation for ERα. (B) Residue level energy decomposition for internal energy in D538G mutation.

Prospective design: Anticipating unseen cancer mutations in novel positions.

Finally, since helix 12 and its neighboring loop are very important for the activity of ERα [11, 12]. Therefore, we scanned all the potential cancerous mutations for all residues in helix 12 and its neighboring loop. k-modes clustering based on three categorical features and $K = 16$ (predicted RLU range, mechanism, and the most important contributing energy type) has been conducted to group the potential cancerous mutations in different bins, which are shown in Table 2.9. There are 7, 4, and 3 mutations known to be activating mutations, seen in patients with breast cancer [104, 105] and known to be inactivating mutations among the predicted ones, respectively. Some of the model-prioritized mutants are expected to be tested experimentally through collaborations.

Group	RLU	Mechanism	Main	Mutations
1	0	3	1	L536A,L540M,K531R,V533H,V534M,C530Q,V533A, L536E,C530T,V533E,L536M,V533D,E380N,V533S,Y537R,H547M
2	1	2	2	V533K,L541S,E380A,E380C,V533T
3	0	1	1	L541V,N532V,L544R,L536S,N532I,S432C,L536V, L536G,L536T,Y537G,M543H
4	0	1	3	D538G,M543N,D538S,E542K,L541D,D538C,E542V,L541N
5	1	3	3	K529A,K529H,V533N,L536F
6	0	3	3	M543R,N532H,S463A,Y537W,E380R,C530G,E380D, V534E,V533Q,L541Y,L469A,H547Y
7	0	2	1	L541Q,L536C,V534D,Y537H,L536D,C530S,L541M, Y537Q,L541E,L540Q,L540H,E380Q,N532G
8	1	1	1	S432T,E380I,S432M,L541I,L540K
9	1	2	1	M543G,V534G,V534S,L540F,L541A,Y537A,Y537S, L541C,Y537D,V534N,Y537E,Y537N,Y537C,Y537M,L536N
10	0	3	2	E380S,V533M,D538Q,L469D,L469H,L469E,D545R
11	0	2	2	L469C,L536Y,L469S,N532A,L541T,L541G
12	0	1	2	Y537L,C530I,L540R,Y537T,L540D,D538N,L539I, D545V,N532C,E380G
13	0	2	3	D538A,D538H,L540E,M543D,C530A
14	1	2	3	L469G,L469Q
15	1	3	1	V534A,V534C,K529W,V534T,L536H,V533C
16	1	3	2	K529F,K529Y,V533R,C530D

Table 2.9: Predicted cancerous mutations in helix 12 and its neighboring loop of ERα categorizes in 16 groups. Red color mutations show the known to be activating mutations. Pink color mutations show the mutation seen in breast cancer patients. Blue color mutations show the known to be inactivating mutations.

2.4 Exact multistate drug design to overcome resistance

2.4.1 Introduction

Drug resistance is a major obstacle in fighting against many disease such as HIV [2], TB [3], cancers [4, 5] and so on. Drug discovery and design is a costly ($\sim$billions of USD) [6] and lengthy ($\sim$12 years) [7] process with low success rates (3.4% phase-1 oncology compounds make it to approval and market) [8]. It is horrifying if the drug we spent billions of USD and more than a decade of our time lose its potency due to evolution and mutation of the target. It is even more problematic for infective and contagious diseases such as Influenza or maybe newly developed COVID 19.

Chronic myelogenous leukemia (CML) disease and our efforts to overcome and cure it can explain the drug resistance phenomena and its huge problem in a more tangible way. Imatinib has been the first-line treatment for CML [106] since 2001. However, patients with advanced CML might relapse due to drug resistance mutations in the kinase domain of the BCR-ABL target protein [107]. Therefore, the "second generation" of drugs needed to overcome the imatinib resistance CML disease. Fortunately, nilotinib and dasatinib have been discovered and approved as the "second generation" of drugs to overcome advanced CML diseases, especially imatinib resistance ones [108]. However, despite these second-generation drugs' success, they encounter new and additional resistance mutations that reduce their efficacy over time [108]. Therefore, "third-generation" or "fourth generation" of drugs are needed to overcome the resistance due to previous until they will encounter new mutations and be useless. Therefore, the main question is how to be one step ahead of disease and anticipate and overcome target mutations that evade the drug interaction while designing drugs? It is also crucial not only to consider the known resistant mutations but also unknown and potential ones like the case in second and third generations of drugs in CML.

Drug discovery with resistant mutations in consideration can be categorized to four groups 1) Heuristic approaches with considering only known mutations; 2) Heuristic approaches with considering all potential mutations; 3) Exact and with optimality guarantee approaches while considering only known mutations; 4) Exact and with optimality guarantee approaches while considering all potential mutations. The state-of-art methods for each category will be explained in detail in the following paragraphs.

There have been several molecular dynamics (MD) approaches to design a drug for overcoming known resistant mutations in Estrogen receptors (ERs), Androgen receptors (ARs) and HIVs [109, 110, 12]. These approaches are categorized in heuristic approaches since they only consider the effectiveness of several known drugs for a couple of known mutations and their wild-type in molecular dynamics trajectories. Therefore, there is no guarantee of discovering the optimal drug for the desired disease. Two heuristic game theory approaches have been developed for considering all potential resistant mutations while designing drugs. The first one is a Stackelberg game theory algorithm for vaccine design [111]. The process of designing a drug from humans and resistant through mutation from the disease has been modeled as a Stackelberg game. An optimal vaccine is defined as the vaccine with the maximum number of mutations needed for being resistant. Their approach is considered as a heuristic method since 1) they used approximation approaches such as Rosetta [112] for binding affinity calculations instead of exact approaches such DFBB or A^* [17]; 2) They randomly search the space of potential mutations instead of searching the whole space or pruning some region of the space with a guarantee. The second method formulates the design of the drug cocktail as an approximate correlated equilibrium [113]. In the correlated equilibrium framework, a drug cocktail is considered successful if, in expectation it can bind to all mutations. However, the existence of even one worst-case scenario mutation is sufficient for drug-resistant happens, and in expectation binding does not have practical usage. Also, they predicted binding affinity through a very simple sparse linear model.

There is only one exact algorithm for drug design while considering resistant mutations [114]. This approach only considers the known mutations and formulates the problem as a set covering problem. They utilize integer programming and DFBB and A to solve the combinatorial optimization problem. However, Their approach only considers known mutation, and there is no exact with optimality guarantee approach for designing drugs while considering all the potential mutations.

We follow the same principle (energy landscape theory) as iCFN for protein design and formulate the problem as a cost function network. We developed a novel branch and bound algorithm to find the best K drug combination in the drug cocktail, minimizing the weakest binding affinity between the drug cocktail and the mutational space. We provide three different perspectives in overcoming drug resistance in the following section for both single and cocktail designs.

2.4.2 Optimal single drug design

2.4.2.1 Formulation

We will first introduce the mathematical formulation of a drug-protein binding energy function. Then we will introduce a single state optimal drug design. Later we will extend the problem to various cases of optimal single drug design and formulate the formally.

Drug-protein binding energy function: Drugs usually consist of a scaffold (the core structure of drug) and functional groups that are attached to it. To model the flexibility of the drug, for a given fixed combination of scaffold and functional groups we discretize the continuous space of scaffold and functional groups. Suppose there are M positions in the scaffold that functional groups can be added and there are N residues in the protein. Then, the binding energy function is usually assumed to be a sum of constant, singleton, and pairwise terms:

$$g(\mathbf{r}, \mathbf{l}) = c + \sum_{m=1}^{M} E(m_l) + \sum_{m \leq M, i \leq N} E(m_l, i_r) + \sum_{m < n \leq M} E(m_l, n_p), \tag{2.18}$$

where m_l and n_p denote functional group l and p at position i and j, respectively.

Single State optimal drug design

Single state drug design is applicable for the case in which we want to design a drug that tightly binds to "target" protein while considering the resistant mutations. The problem is formulated as follow:

$$\text{Sf}^*, \mathbf{d}^* = arg \min_{\substack{\text{Sf} \in \text{Scaffolds} \\ \mathbf{d} \in \mathcal{D}}} \max_{\mathbf{s} \in \mathcal{S}} \min_{q \in \text{Sf}} \min_{\mathbf{l} \in \mathcal{F}_q(\mathbf{d})} \min_{\mathbf{r} \in \mathcal{R}(\mathbf{s})} g_q(\mathbf{r}, \mathbf{l}), \tag{2.19}$$

where the set $\mathcal{D}$, capturing the functional group design space, is the Cartesian product of the sets of functional group types allowed overall positions; the "Scaffolds" represent the library of available scaffolds, q denote the "q" discretization of the given scaffold, and $\mathcal{F}_q(\cdot)$, capturing the discretization library of the functional group for a given scaffold, is the Cartesian product of the discretization sets overall functional group positions of the drug.

The above formula considers a "min-max" game between the drugs (humans as their creator) and the proteins (cancer, bacteria, etc.). We are considering proteins that are playing optimally with respect to our moves. Therefore, we find the best move (drug), which has the lowest worst-case scenario between all of the protein's move (resistant mutations). We extend the formula to consider more general problems and applications such as specificity with multi-state optimal drug design. We categorize the multi-state drug design problems to three cases 1) specificity over proteins: designing drugs to bind to "targeted" proteins rather than "off-targets" 2) Specificity over states of a protein: designing a drug to bind to "desired" state rather than "undesired" state of a protein 3) specificity over drugs: designing a drug to compete with another compound (such as a natural compound in our body) for binding to a "targeted" protein. The three multi-state drug design perspectives are shown in Fig. 2.18.

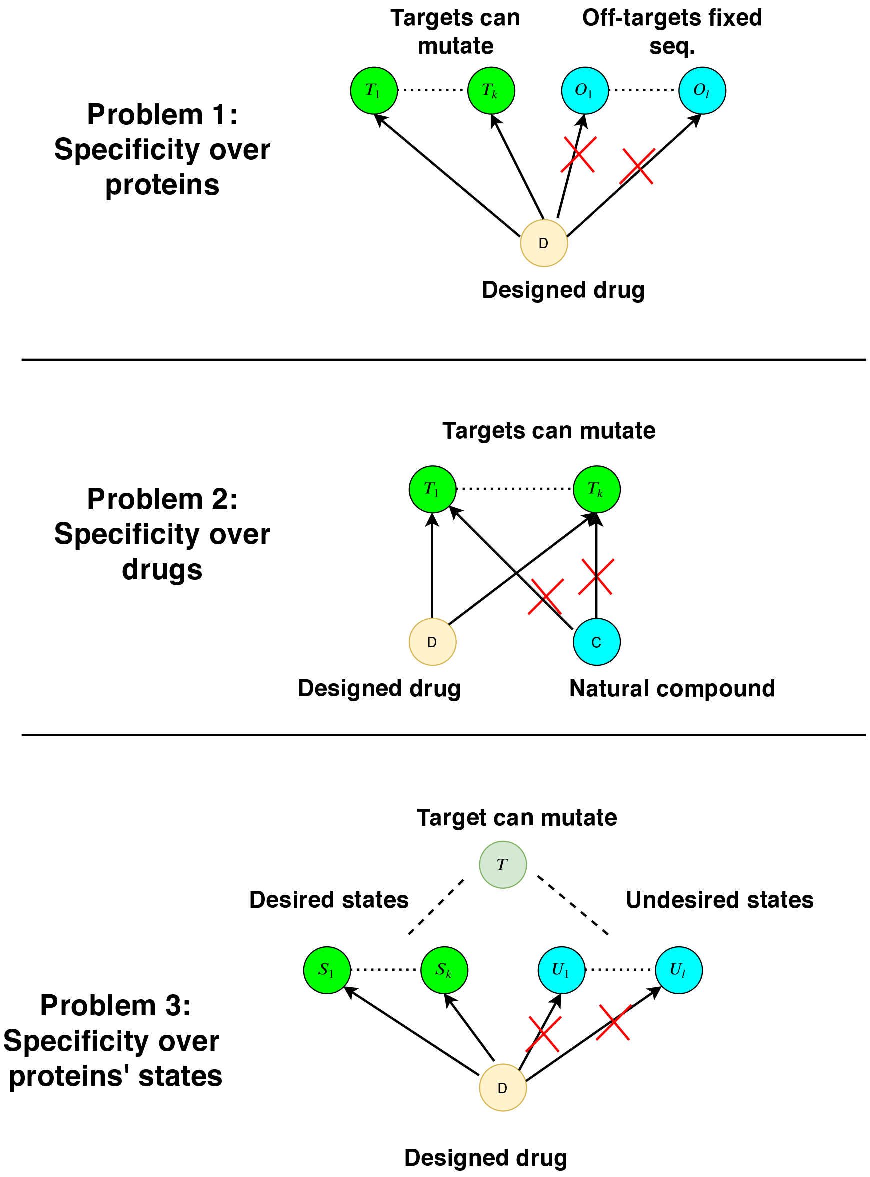


Figure 2.18: Various multi-state drug design perspectives.

Multi-State optimal drug design with proteins specificity

In this category, we are designing an optimal drug to overcome mutational resistance while considering multi-state on multiple proteins. For example, one wants to design an optimal drug to tightly bind to "targeted" proteins rather than "off-targeted" proteins. The "targeted" proteins can overcome the binding with resistance mutations. The problem is formulated as follow:

$$\begin{aligned} \text{Sf}^*, \mathbf{d}^* =& arg \min_{\substack{\text{Sf}\in\text{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \max_{\mathbf{s}\in\mathcal{S}} \min_{q^{+}\in\text{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{+}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{+}(\mathbf{s})} g^{+}_{q^{+}}(\mathbf{r},\mathbf{l}) \\ \text{s.t.} & \min_{o\in\mathcal{O}} \min_{q^{-}\in\text{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{-}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{-}(\text{WT}_o)} g^{-}_{q^{-}}(\mathbf{r},\mathbf{l}) \geq \min_{q^{+}\in\text{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{+}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{+}(\text{WT}_t)} g^{+}_{q^{+}}(\mathbf{r},\mathbf{l}) + \delta, \end{aligned} \tag{2.20}$$

Where positive and negative states are “target” and “off-target” proteins, respectively. WT_t is the wild type amino acid sequence of “targeted” protein and WT_o is o^{th} wild type amino acid sequence of “off-targeted” protein. Moreover, δ is a hyper-parameter with which we restrict the drug to bind to the “off-targeted” proteins in comparison to the “targeted” protein.

Multi-State optimal drug design with protein states’ specificity

In this category, we are designing an optimal drug to overcome mutational resistance while considering multi-state on the protein side. For example, one wants to design an optimal drug to tightly bind to the “desired” state of protein rather “undesired” state of the protein. Or, one wants the drug to bind to the active conformation of the protein rather than inactive conformation of the protein or vice versa. The problem is formulated as follow:

$$\begin{aligned} \text{Sf}^*, \mathbf{d}^* = arg \min_{\substack{\text{Sf}\in\text{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \max_{\mathbf{s}\in\mathcal{S}} \Big(& \min_{q^{+}\in\text{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{+}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{+}(\mathbf{s})} g^{+}_{q^{+}}(\mathbf{r},\mathbf{l}) \\ & - \min_{q^{-}\in\text{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{-}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{-}(\mathbf{s})} g^{-}_{q^{-}}(\mathbf{r},\mathbf{l}) \Big), \end{aligned} \tag{2.21}$$

Where positive and negative states are “desired” state of protein and “undesired” state of the protein, respectively.

Multi-State optimal drug design with competing drug

In this category, we are designing an optimal drug to overcome mutational resistance while considering multi-state on the drug side. For example, one wants to design an optimal drug to in competition with another reference drug or natural substrate, bind tighter to the “target” protein.

The problem is formulated as follow:

$$
\begin{aligned}
\mathrm{Sf}^{*}, \mathbf{d}^{*} = arg \min_{\substack{\mathrm{Sf}^{+} \in \mathrm{Scaffolds}^{+} \\ \mathbf{d}^{+} \in \mathcal{D}^{+}}} \max_{\mathbf{s} \in \mathcal{S}} \Big(& \min_{q^{+} \in \mathrm{Sf}^{+}} \min_{\mathbf{l} \in \mathcal{F}_{q^{+}}(\mathbf{d}^{+})} \min_{\mathbf{r} \in \mathcal{R}^{+}(\mathbf{s})} g^{+}_{q^{+}}(\mathbf{r}, \mathbf{l}) \\
& - \min_{q^{-} \in \mathrm{Sf}^{-}} \min_{\mathbf{l} \in \mathcal{F}_{q^{-}}(\mathbf{d}^{-})} \min_{\mathbf{r} \in \mathcal{R}^{-}(\mathbf{s})} g^{-}_{q^{-}}(\mathbf{r}, \mathbf{l}) \Big),
\end{aligned}
\tag{2.22}
$$

Where positive and negative states are binding of our designed drug with the target protein and binding of reference drug or substrate (denoted by its scaffold Sf^{-} and its functional group combination $\mathbf{d}^{-}$) with the target protein, respectively.

To efficiently and optimally solve the Multi states optimal drug design, We have developed novel lower bounds with theoretical proofs and complexity analysis, presented in the following subsections.

2.4.2.2 Multi-State optimal drug design with proteins specificity

Theorem 5. ***Lower bounding:*** *For any drug space D and scaffold library, a lower bound of the objective function for multistate drug design with protein specificity (Formulation in Eq. 2.20) is given by ($\times$ denotes Cartesian product):*

$$
\begin{aligned}
\min_{Sf \in Scaffolds} \min_{q^{+} \in Sf} \Bigg(c_{q^{+}} + \sum_{m} \min_{d \in \mathcal{D}(m)} \min_{l} \Big(& E_{q^{+}}(m_l) \\
& + \sum_{n>m} \min_{d' \in \mathcal{D}(n)} \min_{p} E_{q^{+}}(m_l, n_p) \\
& + \sum_{i} \min_{r} E_{q^{+}}(m_l, i_r) \Big)
\end{aligned}
\tag{2.23}
$$

Proof.

$$
\begin{aligned}
&\min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \max_{\mathbf{s}\in\mathcal{S}} \min_{q^{+}\in\mathrm{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{+}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{+}(\mathbf{s})} g^{+}_{q^{+}}(\mathbf{r},\mathbf{l})\\
&\geqslant \min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \min_{q^{-}\in\mathrm{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{-}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{-}(\mathbf{s})} g^{-}_{q^{-}}(\mathbf{r},\mathbf{l}) \quad \forall \mathbf{s}\in\mathcal{S}\\
&\geqslant \min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \min_{q^{+}\in\mathrm{Sf}} \Bigg(c_{q^{+}} + \min_{\mathbf{l}} \min_{\mathbf{r}} \Big(\sum_{m} E_{q^{+}}(m_l)\\
&+ \sum_{n>m} E_{q^{+}}(m_l, n_p) + \sum_{m,i} E_{q^{+}}(m_l, i_r) \Big) \Bigg)\\
&\geqslant \min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \min_{q^{+}\in\mathrm{Sf}} \Bigg(c_{q^{+}} + \sum_{m} \min_{l} \Big(E_{q^{+}}(m_l)\\
&+ \sum_{n>m} \min_{p} E_{q^{+}}(m_l, n_p) + \sum_{i} \min_{r} E_{q^{+}}(m_l, i_r) \Big) \Bigg)\\
&= \min_{\mathrm{Sf}\in\mathrm{Scaffolds}} \min_{q^{+}\in\mathrm{Sf}} \Bigg(c_{q^{+}} + \min_{\mathbf{d}\in\mathcal{D}} \sum_{m} \min_{l} \Big(E_{q^{+}}(m_l)\\
&+ \sum_{n>m} \min_{p} E_{q^{+}}(m_l, n_p) + \sum_{i} \min_{r} E_{q^{+}}(m_l, i_r) \Big) \Bigg)\\
&\geqslant \min_{\mathrm{Sf}\in\mathrm{Scaffolds}} \min_{q^{+}\in\mathrm{Sf}} \Bigg(c_{q^{+}} + \sum_{m} \min_{d\in\mathcal{D}(m)} \min_{l} \Big(E_{q^{+}}(m_l)\\
&+ \sum_{n>m} \min_{d'\in\mathcal{D}(n)} \min_{p} E_{q^{+}}(m_l, n_p) + \sum_{i} \min_{r} E_{q^{+}}(m_l, i_r) \Big) \Bigg)
\end{aligned}
\tag{2.24}
$$

□

Computational complexity: We also provide the time complexity of the lower bound as follows. By using a lookup table of size $O(KM(M+N))$ that contains minimal/maximal energy between all position pairs, I accelerate this lower bound calculation by $O(min(l,r))$.

Theorem 6. *The lower bound for multi-State optimal drug design with protein specificity can be computed in $O(SMK(L)^2(Nr+Ml))$, where N is the number of positions in protein, M is the number of functional group positions in drug, S the number of scaffolds, K the average number of substates per scaffold, L the average number of rotamers per functional group position, l the* *average number of rotamers per functional group and* r *the average number of rotamers per amino acid.*

Proof. The following can be calculated in $O(L^2(Nr + Ml))$:

$$\begin{aligned}\min_{d\in\mathcal{D}(m)} \min_{l} \Big(E_{q^+}(m_l) + \sum_{n>m} \min_{d'\in\mathcal{D}(n)} \min_{p} E_{q^+}(m_l, n_p) \\ + \sum_{i} \min_{r} E_{q^+}(m_l, i_r)\Big)\end{aligned} \tag{2.25}$$

summing over all positions it will be $O(ML^2(Nr + Ml))$. Finally, since we are calculating (2.25) for all K substates (discretization of a scaffold) and S scaffolds, complexity will be $O(SMK(L)^2(Nr + Ml))$. □

2.4.2.3 Multi-State optimal drug design with protein states' specificity

Theorem 7. ***Lower bounding:*** *For any drug space* D *and scaffold library, a lower bound of the objective function for multistate drug design with protein specificity (Formulation in Eq. 2.21) is given by (*× *denotes Cartesian product):*

$$\begin{aligned}\min_{Sf\in Scaffolds} \min_{q^-,q^+\in Sf} \Bigg(\Delta c_{q^-,q^+} + \sum_{m} \min_{d\in\mathcal{D}(m)} \min_{(l,l')} \Big(\Delta E_{q^-,q^+}(m_{l,l'}) \\ + \sum_{n>m} \min_{d'\in\mathcal{D}(n)} \min_{(p,p')} \Delta E_{q^-,q^+}(m_{l,l'}, n_{p,p'}) \\ + \sum_{i} \min_{s\in\mathcal{S}(i)} \min_{(r,r')} \Delta E_{q^-,q^+}(m_{l,l'}, i_{r,r'})\Big)\Bigg), \textit{ where}\end{aligned} \tag{2.26}$$

$$\Delta c_{q^-,q^+} = c_{q^-} - c_{q^+},$$

$$\Delta E_{q^-,q^+}(m_{l,l'}) = E_{q^-}(m_l) - E_{q^+}(m_{l'}),$$

$$\Delta E_{q^-,q^+}(m_{l,l'}, n_{p,p'}) = E_{q^-}(m_l, n_p) - E_{q^+}(m_{l'}, n_{p'}),$$

$$\Delta E_{q^-,q^+}(m_{l,l'}, i_{r,r'}) = E_{q^-}(m_l, i_r) - E_{q^+}(m_{l'}, i_{r'}),$$

i.e., differences in constant, singleton, and pairwise energies between a negative substate k *and a positive substate* l*.*

Proof.

$$
\begin{aligned}
&\min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \max_{\mathbf{s}\in\mathcal{S}} \Bigg(\min_{q^{+}\in\mathrm{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{+}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{+}(\mathbf{s})} g^{+}_{q^{+}}(\mathbf{r},\mathbf{l}) \\
&- \min_{q^{-}\in\mathrm{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{-}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{-}(\mathbf{s})} g^{-}_{q^{-}}(\mathbf{r},\mathbf{l}) \Bigg) \\
&= \min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \min_{\mathbf{s}\in\mathcal{S}} \Bigg(\min_{q^{-}\in\mathrm{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{-}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{-}(\mathbf{s})} g^{-}_{q^{-}}(\mathbf{r},\mathbf{l}) \\
&- \min_{q^{+}\in\mathrm{Sf}} \min_{\mathbf{l}\in\mathcal{F}_{q^{+}}(\mathbf{d})} \min_{\mathbf{r}\in\mathcal{R}^{+}(\mathbf{s})} g^{-}_{q^{-}}(\mathbf{r},\mathbf{l}) \Bigg) \\
&\geqslant \min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \min_{\mathbf{s}\in\mathcal{S}} \min_{q^{-},q^{+}\in\mathrm{Sf}} \min_{(\mathbf{l},\mathbf{l}')\in\mathcal{F}_{q^{-}}(\mathbf{d})\times\mathcal{F}_{q^{+}}(\mathbf{d})} \min_{(\mathbf{r},\mathbf{r}')\in\mathcal{R}^{-}(\mathbf{s})\times\mathcal{R}^{+}(\mathbf{s})} \Big(g^{-}_{q^{-}}(\mathbf{r},\mathbf{l}) - g^{+}_{q^{+}}(\mathbf{r}',\mathbf{l}') \Big) \\
&= \min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \min_{\mathbf{s}\in\mathcal{S}} \min_{q^{-},q^{+}\in\mathrm{Sf}} \Bigg(\Delta c_{q^{-},q^{+}} + \min_{(\mathbf{l},\mathbf{l}')} \min_{(\mathbf{r},\mathbf{r}')} \Big(\sum_{m} \Delta E_{q^{-},q^{+}}(m_{l,l'}) \\
&+ \sum_{n>m} \Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) + \sum_{m,i} \Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'}) \Big) \Bigg) \\
&\geqslant \min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}\in\mathcal{D}}} \min_{\mathbf{s}\in\mathcal{S}} \min_{q^{-},q^{+}\in\mathrm{Sf}} \Bigg(\Delta c_{q^{-},q^{+}} + \sum_{m} \min_{(l,l')} \Big(\Delta E_{q^{-},q^{+}}(m_{l,l'}) \\
&+ \sum_{n>m} \min_{(p,p')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) + \sum_{i} \min_{(r,r')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'}) \Big) \Bigg) \\
&= \min_{\mathrm{Sf}\in\mathrm{Scaffolds}} \min_{q^{-},q^{+}\in\mathrm{Sf}} \Bigg(\Delta c_{q^{-},q^{+}} + \min_{\mathbf{d}\in\mathcal{D}} \min_{\mathbf{s}\in\mathcal{S}} \sum_{m} \min_{(l,l')} \Big(\Delta E_{q^{-},q^{+}}(m_{l,l'}) \\
&+ \sum_{n>m} \min_{(p,p')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) + \sum_{i} \min_{(r,r')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'}) \Big) \Bigg) \\
&\geqslant \min_{\mathrm{Sf}\in\mathrm{Scaffolds}} \min_{q^{-},q^{+}\in\mathrm{Sf}} \Bigg(\Delta c_{q^{-},q^{+}} + \sum_{m} \min_{d\in\mathcal{D}(m)} \min_{(l,l')} \Big(\Delta E_{q^{-},q^{+}}(m_{l,l'}) \\
&+ \sum_{n>m} \min_{d'\in\mathcal{D}(n)} \min_{(p,p')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) + \sum_{i} \min_{s\in\mathcal{S}(i)} \min_{(r,r')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'}) \Big) \Bigg)
\end{aligned}
\tag{2.27}
$$

□

Computational complexity: We provide the time complexity of the lower bound as follows. By using a lookup table of size $O(KM(M+N))$ that contains minimal/maximal energy between all position pairs for each positive/negative substate, I accelerate this lower bound calculation by

$O(min(l, r))$.

Theorem 8. *The lower bound for multi-State optimal drug design with protein specificity can be computed in $O(SM(KL)^2(Nr + Ml))$, where N is the number of positions in protein, M is the number of functional group positions in drug, S the number of scaffolds, K the average number of substates per scaffold, L the average number of rotamers per functional group position, l the average number of rotamers per functional group and r the average number of rotamers per amino acid.*

Proof. we prove the complexity by starting with the most inner minimization:

$$\begin{aligned}
&\min_{(p,p')} \Delta E_{q^-,q^+}(m_{l,l'}, n_{p,p'}) \\
&= \min_{(p,p')} \Big(E_{q^-}(m_l, n_p) - E_{q^+}(m_{l'}, n_{p'}) \Big) \\
&= \min_{p} E_{q^-}(m_l, n_p) + \min_{p'} \Big(- E_{q^+}(m_{l'}, n_{p'}) \Big) \\
&= \min_{p} E_{q^-}(m_l, n_p) - \max_{p'} E_{q^+}(m_{l'}, n_{p'})
\end{aligned} \tag{2.28}$$

So, we can calculate it in $O(l)$. Since the number of functional groups is known, then

$$\min_{d' \in D(n)} \min_{(p,p')} \Delta E_{q^-,q^+}(m_{l,l'}, n_{p,p'}) \tag{2.29}$$

will be again $O(l)$, so by summing over positions it will be $O(Ml)$. For calculating:

$$\begin{aligned}
\min_{d \in \mathcal{D}(m)} \min_{(l,l')} \Big(\Delta E_{q^-,q^+}(m_{l,l'}) + \sum_{n>m} \min_{d' \in \mathcal{D}(n)} \min_{(p,p')} \Delta E_{q^-,q^+}(m_{l,l'}, n_{p,p'}) \\
+ \sum_{i} \min_{s \in \mathcal{S}(i)} \min_{(r,r')} \Delta E_{q^-,q^+}(m_{l,l'}, i_{r,r'}) \Big)
\end{aligned} \tag{2.30}$$

similar to previous version, we can compute it in $O(L^2(Nr + Ml))$ and summing over all positions it will be $O(ML^2(Nr + Ml))$. Finally, since we are calculating (2.30) for all K^2 pairs of substates (discretization of a scaffold) across the two states and S scaffolds, complexity will be $O(SM(KL)^2(Nr + Ml))$. □

2.4.2.4 Multi-State optimal drug design with competing drug

Theorem 9. ***Lower bounding:*** *For any drug space D and scaffold library, a lower bound of the objective function for multistate drug design with drug specificity (Formulation in Eq. 2.22) is given by (× denotes Cartesian product):*

$$
\begin{aligned}
\min_{\mathit{Sf}^{+}\in \mathit{Scaffolds}^{+}} \min_{q^{-},q^{+}\in \mathit{Sf}^{-}\times \mathit{Sf}^{+}} \Bigg(\Delta c_{q^{-},q^{+}} + \sum_{m} \min_{d\in\mathcal{D}(m)} \min_{(l,l')} \Big(& \Delta E_{q^{-},q^{+}}(m_{l,l'}) \\
& + \sum_{n>m} \min_{d'\in\mathcal{D}(n)} \min_{(p,p')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) \\
& + \sum_{i} \min_{s\in\mathcal{S}(i)} \min_{(r,r')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'}) \Big) \Bigg), \textit{ where}
\end{aligned}
\tag{2.31}
$$

$$
\begin{aligned}
&\Delta c_{q^{-},q^{+}} = c_{q^{-}} - c_{q^{+}}, \\
&\Delta E_{q^{-},q^{+}}(m_{l,l'}) = E_{q^{-}}(m_l) - E_{q^{+}}(m_{l'}), \\
&\Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) = E_{q^{-}}(m_l, n_p) - E_{q^{+}}(m_{l'}, n_{p'}), \\
&\Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'}) = E_{q^{-}}(m_l, i_r) - E_{q^{+}}(m_{l'}, i_{r'}),
\end{aligned}
$$

i.e., differences in constant, singleton, and pairwise energies between a negative substate k and a positive substate l.

Proof.

$$
\begin{aligned}
&\min_{\substack{\mathrm{Sf}^{+}\in\mathrm{Scaffolds}^{+}\\ \mathbf{d}^{+}\in\mathcal{D}^{+}}} \max_{\mathbf{s}\in\mathcal{S}} \Big(\min_{q^{+}\in\mathrm{Sf}^{+}} \min_{\mathbf{l}\in\mathcal{F}_{q^{+}}(\mathbf{d}^{+})} \min_{\mathbf{r}\in\mathcal{R}^{+}(\mathbf{s})} g^{+}_{q^{+}}(\mathbf{r},\mathbf{l}) \\
&- \min_{q^{-}\in\mathrm{Sf}^{-}} \min_{\mathbf{l}\in\mathcal{F}_{q^{-}}(\mathbf{d}^{-})} \min_{\mathbf{r}\in\mathcal{R}^{-}(\mathbf{s})} g^{-}_{q^{-}}(\mathbf{r},\mathbf{l})\Big) \\
&= \min_{\substack{\mathrm{Sf}^{+}\in\mathrm{Scaffolds}^{+}\\ \mathbf{d}^{+}\in\mathcal{D}^{+}}} \min_{\mathbf{s}\in\mathcal{S}} \Big(\min_{q^{-}\in\mathrm{Sf}^{-}} \min_{\mathbf{l}\in\mathcal{F}_{q^{-}}(\mathbf{d}^{-})} \min_{\mathbf{r}\in\mathcal{R}^{-}(\mathbf{s})} g^{-}_{q^{-}}(\mathbf{r},\mathbf{l}) \\
&- \min_{q^{+}\in\mathrm{Sf}^{+}} \min_{\mathbf{l}\in\mathcal{F}_{q^{+}}(\mathbf{d}^{+})} \min_{\mathbf{r}\in\mathcal{R}^{+}(\mathbf{s})} g^{+}_{q^{+}}(\mathbf{r},\mathbf{l})\Big) \\
&\geqslant \min_{\substack{\mathrm{Sf}^{+}\in\mathrm{Scaffolds}^{+}\\ \mathbf{d}^{+}\in\mathcal{D}^{+}}} \min_{\mathbf{s}\in\mathcal{S}} \min_{(q^{-},q^{+})\in\mathrm{Sf}^{-}\times\mathrm{Sf}^{+}} \min_{(\mathbf{l},\mathbf{l}')\in\mathcal{F}_{q^{-}}(\mathbf{d}^{-})\times\mathcal{F}_{q^{+}}(\mathbf{d}^{+})} \min_{(\mathbf{r},\mathbf{r}')\in\mathcal{R}^{-}(\mathbf{s})\times\mathcal{R}^{+}(\mathbf{s})} \Big(g^{-}_{q^{-}}(\mathbf{r},\mathbf{l}) - g^{+}_{q^{+}}(\mathbf{r}',\mathbf{l}')\Big) \\
&= \min_{\substack{\mathrm{Sf}^{+}\in\mathrm{Scaffolds}^{+}\\ \mathbf{d}^{+}\in\mathcal{D}^{+}}} \min_{\mathbf{s}\in\mathcal{S}} \min_{q^{-},q^{+}\in\mathrm{Sf}^{-}\times\mathrm{Sf}^{+}} \Bigg(\Delta c_{q^{-},q^{+}} + \min_{(\mathbf{l},\mathbf{l}')} \min_{(\mathbf{r},\mathbf{r}')} \Big(\sum_{m} \Delta E_{q^{-},q^{+}}(m_{l,l'}) \\
&+ \sum_{n>m} \Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) + \sum_{m,i} \Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'})\Big)\Bigg) \\
&\geqslant \min_{\substack{\mathrm{Sf}^{+}\in\mathrm{Scaffolds}^{+}\\ \mathbf{d}^{+}\in\mathcal{D}^{+}}} \min_{\mathbf{s}\in\mathcal{S}} \min_{q^{-},q^{+}\in\mathrm{Sf}^{-}\times\mathrm{Sf}^{+}} \Bigg(\Delta c_{q^{-},q^{+}} + \sum_{m} \min_{(l,l')} \Big(\Delta E_{q^{-},q^{+}}(m_{l,l'}) \\
&+ \sum_{n>m} \min_{(p,p')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) + \sum_{i} \min_{(r,r')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'})\Big)\Bigg) \\
&= \min_{\mathrm{Sf}^{+}\in\mathrm{Scaffolds}^{+}} \min_{q^{-},q^{+}\in\mathrm{Sf}^{-}\times\mathrm{Sf}^{+}} \Bigg(\Delta c_{q^{-},q^{+}} + \min_{\mathbf{d}^{+}\in\mathcal{D}^{+}} \min_{\mathbf{s}\in\mathcal{S}} \sum_{m} \min_{(l,l')} \Big(\Delta E_{q^{-},q^{+}}(m_{l,l'}) \\
&+ \sum_{n>m} \min_{(p,p')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) + \sum_{i} \min_{(r,r')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'})\Big)\Bigg) \\
&\geqslant \min_{\mathrm{Sf}^{+}\in\mathrm{Scaffolds}^{+}} \min_{q^{-},q^{+}\in\mathrm{Sf}^{-}\times\mathrm{Sf}^{+}} \Bigg(\Delta c_{q^{-},q^{+}} + \sum_{m} \min_{d\in\mathcal{D}(m)} \min_{(l,l')} \Big(\Delta E_{q^{-},q^{+}}(m_{l,l'}) \\
&+ \sum_{n>m} \min_{d'\in\mathcal{D}(n)} \min_{(p,p')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, n_{p,p'}) + \sum_{i} \min_{s\in\mathcal{S}(i)} \min_{(r,r')} \Delta E_{q^{-},q^{+}}(m_{l,l'}, i_{r,r'})\Big)\Bigg)
\end{aligned}
\tag{2.32}
$$

□

Computational complexity: We calculate the time complexity of the lower bound as follows. By using a lookup table of size $O(KM(M+N))$ that contains minimal/maximal energy between all position pairs for each positive/negative substate, I accelerate this lower bound calculation by

$O(min(l,r))$.

Theorem 10. *The lower bound for multi-State optimal drug design with competing drug can be computed in $O(SM(KL)^2(Nr+Ml))$, where N is the number of positions in protein, M is the number of functional group positions in drug, S the number of scaffolds, K the average number of substates per scaffold, L the average number of rotamers per functional group position, l the average number of rotamers per functional group and r the average number of rotamers per amino acid.*

Proof. we prove the complexity by starting with the most inner minimization:

$$\begin{aligned}
&\min_{(p,p')} \Delta E_{q^-,q^+}(m_{l,l'}, n_{p,p'}) \\
&= \min_{(p,p')} \Big(E_{q^-}(m_l, n_p) - E_{q^+}(m_{l'}, n_{p'}) \Big) \\
&= \min_{p} E_{q^-}(m_l, n_p) + \min_{p'} \Big(- E_{q^+}(m_{l'}, n_{p'}) \Big) \\
&= \min_{p} E_{q^-}(m_l, n_p) - \max_{p'} E_{q^+}(m_{l'}, n_{p'})
\end{aligned} \tag{2.33}$$

So, we can calculate it in $O(l)$. Since the number of functional groups is known, then

$$\min_{d' \in D(n)} \min_{(p,p')} \Delta E_{q^-,q^+}(m_{l,l'}, n_{p,p'}) \tag{2.34}$$

will be again $O(l)$, so by summing over positions it will be $O(Ml)$. For calculating:

$$\begin{aligned}
\min_{d \in \mathcal{D}(m)} \min_{(l,l')} \Big(\Delta E_{q^-,q^+}(m_{l,l'}) + \sum_{n>m} \min_{d' \in \mathcal{D}(n)} \min_{(p,p')} \Delta E_{q^-,q^+}(m_{l,l'}, n_{p,p'}) \\
+ \sum_{i} \min_{s \in \mathcal{S}(i)} \min_{(r,r')} \Delta E_{q^-,q^+}(m_{l,l'}, i_{r,r'}) \Big)
\end{aligned} \tag{2.35}$$

similar to previous version, we can compute it in $O(L^2(Nr+Ml))$ and summing over all positions it will be $O(ML^2(Nr+Ml))$. Finally, since we are calculating (2.35) for all K^2 pairs of substates (discretization of a scaffold) across the two states and S scaffolds, complexity will be $O(SM(KL)^2(Nr+Ml))$. □

2.4.3 Optimal Drug cocktail design:

One of the biggest concerns with the current formulations is that we are looking for a single drug to overcome all the mutations. In other words, we are looking for a "Superdrug" that have great specificity for worst-case resistant mutations. Therefore, a single drug might not exist to have this property. One remedy for this challenge is to utilize multiple drugs or " drug cocktail" to overcome all the resistant mutations. By utilizing a drug cocktail, we are making the challenge easier on the perspective of the drug. Since every drug needs to cover a subspace of the whole resistant mutation space and overall, their union should add up to the whole space. We will consider the "winner takes all" perspective in drug cocktail: for a given sequence (or resistant mutation), the drug with the highest binding at each state will win the binding among all the drugs. Therefore, we are looking for a drug cocktail in which for each resistant mutation, the best drug binding to the positive state should be better than the best drug in the negative state. For a given sequence, the best drug among the chosen drug cocktail can differ from a positive state to a negative state. Moreover, the best drug among the chosen drug cocktail can be different from one sequence to another. The previous optimal single drug design will be extended to an optimal drug cocktail in the following subsections.

2.4.3.1 Multi-State optimal drug cocktail with proteins specificity

In this category, we are designing an optimal drug cocktail to overcome mutational resistance while considering multi-state on multiple proteins. For example, one wants to design an optimal drug to tightly bind to "targeted" proteins rather than "off-targeted" proteins. The "targeted" proteins can overcome the binding with resistance mutations. The problem is formulated as follow:

$$\begin{aligned}
\mathrm{Sf}^*, \mathbf{d}^* =& arg \min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}_1,\cdots,\mathbf{d}_k\in\mathcal{D}}} \max_{\mathbf{s}\in\mathcal{S}} \min_{q^+\in\mathrm{Sf}} \min_{(\mathbf{l}_1,\cdots,\mathbf{l}_k)\in(\mathcal{F}_{q^+}(\mathbf{d}_1),\cdots,\mathcal{F}_{q^+}(\mathbf{d}_k))} \min_{\mathbf{r}\in\mathcal{R}^+(\mathbf{s})} g^+_{q^+}(\mathbf{r}, \mathbf{l}_k) \\
\text{s.t.} & \min_{o\in\mathcal{O}} \min_{q^-\in\mathrm{Sf}} \min_{(\mathbf{l}_1,\cdots,\mathbf{l}_k)\in(\mathcal{F}_{q^-}(\mathbf{d}_1),\cdots,\mathcal{F}_{q^-}(\mathbf{d}_k))} \min_{\mathbf{r}\in\mathcal{R}^-(\mathrm{WT}_o)} g^-_{q^-}(\mathbf{r}, \mathbf{l}_k) \\
& \geqslant \min_{q^+\in\mathrm{Sf}} \min_{(\mathbf{l}_1,\cdots,\mathbf{l}_k)\in(\mathcal{F}_{q^+}(\mathbf{d}_1),\cdots,\mathcal{F}_{q^+}(\mathbf{d}_k))} \min_{\mathbf{r}\in\mathcal{R}^+(\mathrm{WT}_t)} g^+_{q^+}(\mathbf{r}, \mathbf{l}_k) + \delta,
\end{aligned} \tag{2.36}$$

Where all of the individual drugs in the drug cocktail should have a very low binding with "off-target" proteins in comparison to the WT.

Theorem 11. ***Lower bound for multi-State optimal drug cocktail design with proteins specificity.*** *For any drug D^k of k^{th} drug and scaffold library, a lower bound of the objective function for multistate drug cocktail design with protein specificity (Formulation in Eq. 2.36) is given by (× denotes Cartesian product):*

$$\begin{aligned} \min_{Sf \in Scaffolds} \min_{q^+ \in Sf} \min_{k \in K} \Bigg(c_{q^+} + \sum_{m} \min_{d \in \mathcal{D}^k(m)} \min_{l} \Big(& E_{q^+}(m_l) \\ & + \sum_{n>m} \min_{d' \in \mathcal{D}^k(n)} \min_{p} E_{q^+}(m_l, n_p) \\ & + \sum_{i} \min_{r} E_{q^+}(m_l, i_r) \Big) \end{aligned} \tag{2.37}$$

2.4.3.2 Multi-State optimal drug cocktail design with protein state' specificity

Similar to the optimal single drug design with protein specificity, we want to design optimal drug cocktails to overcome mutational resistance while considering multi-state on the protein side. The problem is formulated as follow:

$$\begin{aligned} \mathrm{Sf}^*, \mathbf{d}^* = arg \min_{\substack{\mathrm{Sf} \in \mathrm{Scaffolds} \\ \mathbf{d}_1, \cdots, \mathbf{d}_k \in \mathcal{D}}} \max_{\mathbf{s} \in \mathcal{S}} \Big(& \min_{q^+ \in \mathrm{Sf}} \min_{(\mathbf{l}_1, \cdots, \mathbf{l}_k) \in (\mathcal{F}_{q^+}(\mathbf{d}_1), \cdots, \mathcal{F}_{q^+}(\mathbf{d}_k))} \min_{\mathbf{r} \in \mathcal{R}^+(\mathbf{s})} g^+_{q^+}(\mathbf{r}, \mathbf{l}_k) \\ & - \min_{q^- \in \mathrm{Sf}} \min_{(\mathbf{l}_1, \cdots, \mathbf{l}_k) \in (\mathcal{F}_{q^-}(\mathbf{d}_1), \cdots, \mathcal{F}_{q^-}(\mathbf{d}_k))} \min_{\mathbf{r} \in \mathcal{R}^-(\mathbf{s})} g^-_{q^-}(\mathbf{r}, \mathbf{l}_k)) \Big), \end{aligned} \tag{2.38}$$

Where positive and negative states are "desired" and "undesired" states of a protein, respectively.

Theorem 12. ***Lower bound for multi-State optimal drug cocktail design with protein states' specificity.*** *For any drug space D^k of k^{th} drug and scaffold library, a lower bound of the objective function for multistate drug cocktail design with protein state's specificity (Formulation in*

Eq. 2.38) is given by (× denotes Cartesian product):

$$\min_{\mathit{Sf}\in\mathit{Scaffolds}} \min_{q^{\text{-}},q^{+}\in\mathit{Sf}} \min_{k\in K} \Bigg(\Delta c_{q^{\text{-}},q^{+}} + \sum_{m} \min_{d\in\mathcal{D}^{k}(m)} \min_{(l,l')} \Big(\Delta E_{q^{\text{-}},q^{+}}(m_{l,l'}) + \sum_{n>m} \min_{d'\in\mathcal{D}^{k}(n)} \min_{(p,p')} \Delta E_{q^{\text{-}},q^{+}}(m_{l,l'}, n_{p,p'}) + \sum_{i} \min_{s\in\mathcal{S}(i)} \min_{(r,r')} \Delta E_{q^{\text{-}},q^{+}}(m_{l,l'}, i_{r,r'}) \Big) \Bigg), \textit{ where} \tag{2.39}$$

$$\begin{aligned}
&\Delta c_{q^{\text{-}},q^{+}} = c_{q^{\text{-}}} - c_{q^{+}},\\
&\Delta E_{q^{\text{-}},q^{+}}(m_{l,l'}) = E_{q^{\text{-}}}(m_l) - E_{q^{+}}(m_{l'}),\\
&\Delta E_{q^{\text{-}},q^{+}}(m_{l,l'}, n_{p,p'}) = E_{q^{\text{-}}}(m_l, n_p) - E_{q^{+}}(m_{l'}, n_{p'}),\\
&\Delta E_{q^{\text{-}},q^{+}}(m_{l,l'}, i_{r,r'}) = E_{q^{\text{-}}}(m_l, i_r) - E_{q^{+}}(m_{l'}, i_{r'}),
\end{aligned}$$

i.e., differences in constant, singleton, and pairwise energies between a negative substate k and a positive substate l.

2.4.3.3 Multi-State optimal drug cocktail design with competing drug

Similar to the optimal single drug design with protein specificity, we want to design optimal drug cocktails to overcome mutational resistance while considering multi-state on the drug side. The problem is formulated as follow:

$$\mathbf{Sf}^{*}, \mathbf{d}^{*} = arg \min_{\substack{\mathrm{Sf}\in\mathrm{Scaffolds}\\ \mathbf{d}_1,\cdots,\mathbf{d}_k\in\mathcal{D}}} \max_{\mathbf{s}\in\mathcal{S}} \Big(\min_{q^{+}\in\mathrm{Sf}} \min_{(\mathbf{l}_1,\cdots,\mathbf{l}_k)\in(\mathcal{F}_{q^{+}}(\mathbf{d}_1),\cdots,\mathcal{F}_{q^{+}}(\mathbf{d}_k))} \min_{\mathbf{r}\in\mathcal{R}^{+}(\mathbf{s})} g^{+}_{q^{+}}(\mathbf{r}, \mathbf{l}_k) - \min_{q^{\text{-}}\in\mathrm{Sf}^{\text{-}}} \min_{\mathbf{l}\in\mathcal{F}_{q^{\text{-}}}(\mathbf{d}^{\text{-}})} \min_{\mathbf{r}\in\mathcal{R}^{\text{-}}(\mathbf{s})} g^{\text{-}}_{q^{\text{-}}}(\mathbf{r}, \mathbf{l})) \Big), \tag{2.40}$$

Where positive and negative states are binding of our designed drug cocktail with the tar- get protein and binding of reference drug or substrate with the target protein, respectively.

Theorem 13. ***Lower bound for multi-State optimal drug cocktail design with competing drug.*** *For any drug space D^k of k^{th} drug and scaffold library, a lower bound of the objective function for multistate drug cocktail design with competing drug (Formulation in Eq. 2.40) is given by (×*

denotes Cartesian product):

$$\begin{aligned}\min_{\mathit{Sf}^+\in \mathit{Scaffolds}^+}\ \min_{q^-,q^+\in \mathit{Sf}^-\times \mathit{Sf}^+}\ \min_{k\in K}\Bigg(\Delta c_{q^-,q^+} &+ \sum_m \min_{d\in\mathcal{D}^k(m)}\ \min_{(l,l')}\Big(\Delta E_{q^-,q^+}(m_{l,l'})\\ &+\sum_{n>m}\min_{d'\in\mathcal{D}^k(n)}\ \min_{(p,p')}\Delta E_{q^-,q^+}(m_{l,l'},n_{p,p'})\\ &+\sum_i \min_{s\in\mathcal{S}(i)}\ \min_{(r,r')}\Delta E_{q^-,q^+}(m_{l,l'},i_{r,r'})\Big)\Bigg),\ \textit{where}\end{aligned}\tag{2.41}$$

$$\begin{aligned}&\Delta c_{q^-,q^+} = c_{q^-} - c_{q^+},\\ &\Delta E_{q^-,q^+}(m_{l,l'}) = E_{q^-}(m_l) - E_{q^+}(m_{l'}),\\ &\Delta E_{q^-,q^+}(m_{l,l'},n_{p,p'}) = E_{q^-}(m_l,n_p) - E_{q^+}(m_{l'},n_{p'}),\\ &\Delta E_{q^-,q^+}(m_{l,l'},i_{r,r'}) = E_{q^-}(m_l,i_r) - E_{q^+}(m_{l'},i_{r'}),\end{aligned}$$

i.e., differences in constant, singleton, and pairwise energies between a negative substate k *and a positive substate* l*.*

2.4.4 Conclusions

For the first time, we have developed an exact algorithm that is efficient for generic multi-state drug design. We have formulated three generic multi-state single drug and drug cocktails problems through the minimax framework. The proposed algorithm exploits novel bounds that can be quickly evaluated in the CFN framework as well as joint consideration of substate CFNs that can quickly prune subtrees at the sequence, substate, and conformer levels with a guarantee. Future directions include parallelizing the algorithm and its codes on the architecture of GPU, incorporating more types of constraints seen in applications while allowing for more general objective functions and continuous rotamers, and deriving tighter yet economic bounds under the framework of CFN.

3. PROTEIN-COMPOUND AFFINITY AND CONTACT PREDICTION*

3.1 Overview

There is a huge need for fast and accurate compound-protein affinity predictor when developing data- and principle-driven approach for overcoming drug resistance in molecular networks. Therefore, in this chapter, we focused on developing fast and accurate protein-compound affinity. These models are critically needed as an oracle for drug combinations generative models. Two models have been developed for very accurate affinity prediction while being interpretable (DeepAffinity) and explainable (DeepRelation). The structure of this chapter is as follows. In section 3.2, DeepAffinity the first interpretable deep learning model, is introduced for interpretable, accurate affinity prediction, and large-scale drug screening. DeepAffinity has shown to be beneficial for binding pocket prediction, explaining protein and drug specificity in a few case studies. In section 3.3, a large dataset of affinity and contact prediction has been curated for large-scale learning and assessment of contact prediction. DeepRelations, as an extension of DeepAffinity, has been developed for explainable affinity with significantly high contact prediction. DeepRelation shows that better interpretability can lead to better generalizability and affinity prediction. In addition, DeepRelation has shown to be beneficial in binding site prediction, SAR, lead optimization, and docking.

3.2 DeepAffinity: Interpretable Deep Learning of Compound–Protein Affinity

3.2.1 Introduction

Drugs are often developed to target proteins that participate in many cellular processes. Among almost 900 FDA-approved drugs as of year 2016, over 80% are small-molecule compounds that act

*Reprinted with permission from "Explainable deep relational networks for predicting compound-protein affinities and contacts" by M. Karimi, D. Wu, Z. Wang, Y. Shen 2020. under revision for Journal of chemical information and modeling, Copyright 2020 American Chemical Society.

*Reprinted with permission from "DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks." by M. Karimi, D. Wu, Z. Wang, Y. Shen 2019. Bioinformatics 35, no. 18, 3329-3338, Copyright 2019 Oxford University Press.

on proteins for drug effects [115]. Clearly, it is of critical importance to characterize compound-protein interaction for drug discovery and development, whether screening compound libraries for given protein targets to achieve desired effects or testing given compounds against possible off-target proteins to avoid undesired effects. However, experimental characterization of every possible compound-protein pair can be daunting, if not impossible, considering the enormous chemical and proteomic spaces. Computational prediction of compound-protein interaction (CPI) has therefore made much progress recently, especially for repurposing and repositioning known drugs for previously unknown but desired new targets [116, 117] and for anticipating compound side-effects or even toxicity due to interactions with off-targets or other drugs [118, 119].

Structure-based methods can predict compound-protein affinity, i.e., how active or tight-binding a compound is to a protein; and their results are highly interpretable. This is enabled by evaluating energy models [120] on 3D structures of protein-compound complexes. As these structures are often unavailable, they often need to be first predicted by "docking" individual structures of proteins and compounds together, which tends to be a bottleneck for computational speed and accuracy [121]. Machine learning has been used to improve scoring accuracy based on energy features [122].

More recently, deep learning has been introduced to predict compound activity or binding-affinity from 3D structures directly. Wallach et al. developed AtomNet, a deep convolutional neural network (CNN), for modeling bioactivity and chemical interactions [123]. Gomes et al. [124] developed atomic convolutional neural network (ACNN) for binding affinity by generating new pooling and convolutional layers specific to atoms. Jimenez et al. [125] also used 3D CNN with molecular representation of 3D voxels assigned to various physicochemical property channels. Besides these 3D CNN methods, Cang and Wei represented 3D structures in novel 1D topology invariants in multiple channels for CNN [126]. These deep learning methods often improve scoring thanks to modeling long-range and multi-body atomic interactions. Nevertheless, they still rely on actual 3D structures of CPI and remain largely untested on lower-quality structures predicted from docking, which prevents large-scale applications.

Sequence-based methods overcome the limited availability of structural data and the costly need of molecular docking. Rather, they exploit rich omics-scale data of protein sequences, compound sequences (e.g. 1D binary substructure fingerprints [127]) and beyond (e.g. biological networks). However, they have been restricted to classifying CPIs [128] mainly into two types (binding or not) and occasionally more (e.g., binding, activating, or inhibiting [129]). More importantly, their interpretablity is rather limited due to high-level features. Earlier sequence-based machine learning methods are based on shallow models for supervised learning, such as support vector machines, logistic regression, random forest, and shallow neural networks [130, 131, 132, 133, 134].

These shallow models are not lack of interpretability *per se*, but the sequence-based high-level features do not provide enough interpretability for mechanistic insights on why a compound–protein pair interacts or not.

Deep learning has been introduced to improve CPI identification from sequence data and shown to outperform shallow models. Wang and Zeng developed a method to predict three types of CPI based on restricted Boltzmann machines, a two-layer probabilistic graphical model and a type of building block for deep neural networks [129]. Tian et al. boosted the performance of traditional shallow-learning methods by a deep learning-based algorithm for CPI [135]. Wan et al. exploited feature embedding algorithm such as latent semantic algorithm [136] and word2vec [137] to automatically learn low-dimensional feature vectors of compounds and proteins from the corresponding large-scale unlabeled data [138]. Later, they trained deep learning to predict the likelihood of their interaction by exploiting the learned low-dimensional feature space. However, these deep-learning methods inherit from sequence-based methods two limitations: simplified task of predicting whether rather than how active CPIs occur as well as low interpretability due to the lack of fine-resolution structures. In addition, interpretability for deep learning models remains a challenge albeit with fast progress especially in a model-agnostic setting [139, 140] .

As has been reviewed, structure-based methods predict quantitative levels of CPI in a realistic setting and are highly interpretable with structural details. But their applicability is restricted by the availability of structure data, and the molecular docking step makes the bottleneck of their efficiency. Meanwhile, sequence-based methods often only predict binary outcomes of CPI in a simplified setting and are less interpretable in lack of mechanism-revealing features or representations; but they are broadly applicable with access to large-scale omics data and generally fast with no need of molecular docking.

Our goal is to, realistically, predict quantitative levels of CPIs (compound-protein affinity measured in IC_{50}, K_i, or K_d) from sequence data alone and to balance the trade-offs of previous structure- or sequence-based methods for broad applicability, high throughput and more interpretability. From the perspective of machine learning, this is a much more challenging regression problem compared to the classification problem seen in previous sequence-based methods.

To tackle the problem, we have designed interpretable yet compact data representations and introduced a novel and interpretable deep learning framework that takes advantage of both unlabeled and labeled data. Specifically, we first have represented compound sequences in the Simplified Molecular-Input Line-Entry System (SMILES) format [141] and protein sequences in novel alphabets of structural and physicochemical properties. These representations are much lower-dimensional and more informative compared to previously-adopted small-molecule substructure fingerprints or protein Pfam domains [135]. We then leverage the wealth of abundant unlabeled data to distill representations capturing long-term, nonlinear dependencies among residues/atoms in proteins/compounds, by pre-training bidirectional recurrent neural networks (RNNs) as part of the seq2seq auto-encoder that finds much success in modeling sequence data in natural language processing [142]. And we develop a novel deep learning model unifying RNNs and convolutional neural networks (CNNs), to be trained from end to end [143] using labeled data for task-specific representations and predictions. Furthermore, we introduce several attention mechanisms to interpret predictions by isolating main contributors of molecular fragments or their pairs, which is further exploited for predicting binding sites and origins of binding specificity. Lastly, we explore alternative representations using protein sequences or compound graphs (structural formulae), develop graph CNN (GCNN) in our unified RNN/GCNN-CNN model, and discuss remaining challenges.

The overall pipeline of our unified RNN-CNN method for semi-supervised learning (data representation, unsupervised learning, and joint supervised learning) is illustrated in Fig. 3.1 with details given next.

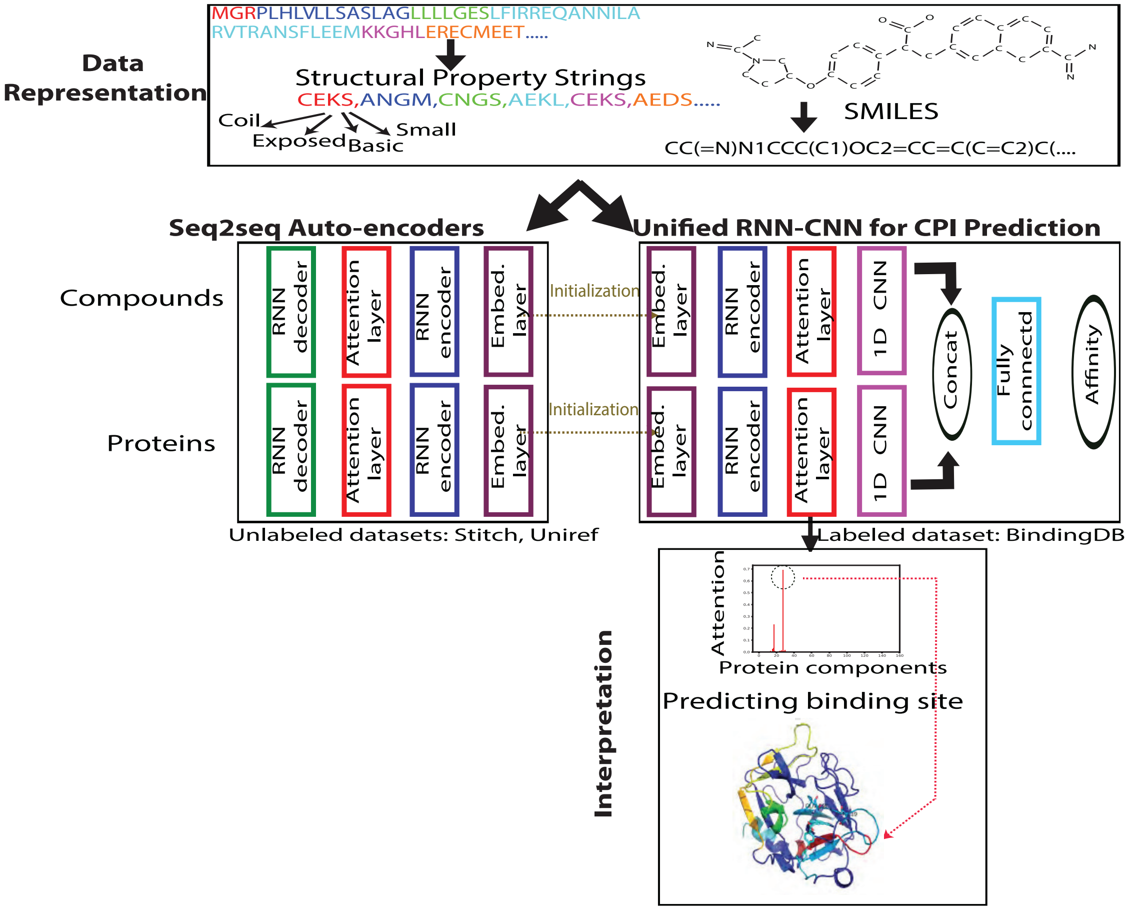


Figure 3.1: Our unified RNN-CNN pipeline to predict and interpret compound-protein affinity.

3.2.2 Materials and Methods

3.2.2.1 Data

We used molecular data from three public datasets: labeled compound-protein binding data from BindingDB [144], compound data in the SMILES format from STITCH [145] and protein amino-acid sequences from UniRef [146].

Starting with 489,280 IC_{50}-labeled samples, we completely excluded four classes of proteins from the training set: nuclear estrogen receptors (ER; 3,374 samples), ion channels (14,599 samples), receptor tyrosine kinases (34,318 samples), and G-protein-coupled receptors (GPCR; 60,238 samples), to test the generalizability of our framework. And we randomly split the rest into the training set (263,583 samples including 10% held out for validation) and the default test set (113,168 samples) without the aforementioned four classes of protein targets.

The label is in the logarithm form: pIC_{50}. We similarly curated datasets for K_i and K_d measurements. For unlabeled compound (protein) data, we used 499,429 (120,000) samples for training and 484,481 (50,525) for validation.

3.2.2.2 Input data representation

Only 1D sequence data are assumed available. 3D structures of proteins, compounds, or their complexes are not used.

3.2.2.2.1 Compound data representation

Baseline representation. A popular compound representation is based on 1D binary substructure fingerprints from PubChem [127]. Mainly, basic substructures of compounds are used as fingerprints by creating binary vectors of 881 dimensions. **SMILES representation.** We used SMILES [141] that are short ASCII strings to represent compound chemical structures based on bonds and rings between atoms. 64 symbols are used for SMILES strings in our data. 4 more special symbols are introduced for the beginning or the end of a sequence, padding (to align sequences in the same batch), or not-used ones. Therefore, we defined a compound "alphabet" of 68 "letters". Compared to the baseline representation which uses k-hot encoding, canonical SMILES strings fully and uniquely determine chemical structures and are yet much more compact.

3.2.2.2.2 Protein data representation

Baseline representation. Previously the most common protein representation for CPI classification was a 1D binary vector whose dimensions correspond to thousands of (5,523 in [135]) Pfam domains [147] (structural units) and 1's are assigned based on k-hot encoding [132, 134]. We considered all types of Pfam entries (family, domain, motif, repeat, disorder, and coiled coil) for better coverage of structural descriptions, which leads to 16,712 entries (Pfam 31.0) as features.

Protein sequences are queried in batches against Pfam using the web server HMMER (hmmscan) [148] with the default gathering threshold.

Structural property sequence (SPS) representation. Although 3D structure data of proteins are often a luxury and their prediction without templates remains a challenge, protein structural properties could be well predicted from sequences [149, 150, 151]. We used SSPro [150] to predict secondary structure for each residue and grouped neighboring residues into secondary structure elements (SSEs). The details and the pseudo-code for SSE are in Algorithm 2.

Algorithm 2 "Smoothing" Secondary Structure Element Sequence

```
Input: Secondary Structures over a Protein Sequence
Output: Smoothed Secondary Structure Element Sequence
Initialize: MeaningfulMinLength as 3 for Coil and 5 for Sheet and Helix
Initialize: MaxGroupLength as 7
for Each line in Secondary Structure do
    Initialize FinalSeq as Empty
    for Each letter in line do
        if Current = Next then
            Store Current in S
            Mark the length of continue same sequence as L
        else if $L > MeaningfulMinLength$ then
            Smooth all letters in S to last letter
            Store S to FinalSeq and clear S
            Continue
        else if Length of $S < MaxGroupLength$ then
            Store to S
            Continue
        else
            Smooth all letters to the most number letter in S
            Store S to FinalSeq and Clear S
            Continue
        end if
    end for
    return FinalSeq
    Clear FinalSeq
end for
```

We defined 4 separate alphabets of 3, 2, 4 and 3 letters, respectively to characterize SSE category, solvent accessibility, physicochemical characteristics, and length (Table 3.1) and combined letters from the 4 alphabets in the order above to create 72 “words” (4-tuples) to describe SSEs.

Secondary Structure			Solvent Exposure		Property				Length		
Alpha	Beta	Coil	Not Exposed	Exposed	Non-polar	Polar	Acidic	Basic	Short	Medium	Long
A	B	C	N	E	G	T	D	K	S	M	L

Table 3.1: 4-tuple of letters in protein structural property sequence (SPS) for creating words.

Pseudo-code for the protein representation is shown as Algorithm 3. Considering the 4 more special symbols similarly introduced for compound SMILES strings, we flattened the 4-tuples and thus defined a protein SPS “alphabet” of 76 “letters”.

The SPS representation overcomes drawbacks of Pfam-based baseline representation: it provides higher resolution of sequence and structural details for more challenging regression tasks, more distinguishability among proteins in the same family, and more interpretability on which protein segments (SSEs here) are responsible for predicted affinity. All these are achieved with a much smaller alphabet of size 76 , which leads to around 100-times more compact representation of a protein sequence than the baseline. In addition, the SPS sequences are much shorter than amino-acid sequences and prevents convergence issues when training RNN and LSTM for sequences longer than 1,000 [152].

3.2.2.3 RNN for unsupervised pre-training

We encode compound SMILES or protein SPS into representations, first by unsupervised deep learning from abundant unlabeled data. We used a recurrent neural network (RNN) model, seq2seq [153], that has seen much success in natural language processing and was recently applied to embedding compound SMILES strings into fingerprints [154].

Algorithm 3 Secondary Structure Element (SSE) Sequence to Structure Property Sequence (SPS)

```
Input: Smoothed Secondary Structure and Exposedness Protein Sequence
Output: Structural Property Sequence (SPS).
Initialize: Exposedness threshold eThres
Initialize: Polarity threshold pThres based on the average of each types
for Each ss in Secondary Structure Element Sequences do
    Initialize FinalSeq as Empty
    Read next Exposedness Protein Sequence to acc
    Read next Protein Sequence to protein
    for Each continuous same letter in ss do
        Store corresponding letter to G
        if the number of ‘e’ in ss $>$ eThres then
            Store ‘E’ to G
        else
            Store ‘N’ to G
        end if
        if percentage of one type polarity in protein $>$ pThres then
            Store corresponding letter to G
        end if
        if Length of group $\leq 7$ then
            Store ‘S’ letter to G
        else if Length of group $\leq 15$ then
            Store ‘M’ letter to G
        else if Length of group $> 15$ then
            Store ‘L’ letter to G
        end if
        Store G to FinalSeq
        Clear G
    end for
    return FinalSeq
    Clear FinalSeq
end for
```

A Seq2seq model is an auto-encoder that consists of two recurrent units known as the encoder and the decoder, respectively (see the corresponding box in Fig. 3.1).

The encoder maps an input sequence (SMILES/SPS in our case) to a fixed-dimension vector known as the thought vector. Then the decoder maps the thought vector to the target sequence (again, SMILES/SPS here). We choose gated recurrent unit (GRU) [155] as our default seq2seq model and treat the thought vectors as the representations learned from the SMILES/SPS inputs.

Our alphabets include 68 and 76 letters (including 4 special symbols such as padding in either alphabet) for compound SMILES and protein SPS strings, respectively. Based on the statistics of 95% CPIs in BindingDB, we set the maximum lengths of SMILES and SPS strings to be 100 and 152, respectively.

Accordingly, we used 2 layers of GRU with both the latent dimension and the embedding layer (discrete letter to continuous vector) dimension being 128 for compounds and 256 for proteins. We used an initial learning rate of 0.5 with a decay rate of 0.99, a dropout rate of 0.2, and a batch size of 64. We also clipped gradients by their global norms. All neural network models for supervised learning were implemented based on TensorFlow [156] and TFLearn [157].

We further test four more seq2seq variants under the combination of the following options:

- **Bucketing** [158] as an optimization trick to put sequences of similar lengths in the same bucket and padded accordingly during training. We used bucketing groups of {(30,30), (60,60), (90,90), (120,120), (152,152)} for SPS and {(20,20), (40,40), (60,60), (80,80), (100,100)} for SMILES.

- **Bidirectional GRU** that shares parameters to capture both forward and backward dependencies [159].

- **Attention** mechanism [160] that allows encoders to "focus" in each encoding step on selected previous time points that are deemed important to predict target sequences.

Both bidirectional RNN and attention mechanism can help address computational challenges from long input sequences.

To address the challenge from long input sequences, the attention mechanism provides a way to "focus" for encoders. It only allows each encoding step to be affected by selected previous time points deemed important, thus saving its memory burden. Suppose that the maximum length of both the encoder and the decoder is L, the output of RNN (the input to the attention model) is $(s_1, \cdots, s_t, \cdots, s_L)$, the hidden state of the decoder $(h_1, \cdots, h_j, \cdots, h_L)$, and the output of attention model is C_j at each time step j of the decoder. The attention model is parametrized by two matrices, U_a and W_a, and a vector $\boldsymbol{v}_\mathrm{a}$ ('a' stands for attention) and formulated as:

$$
\begin{aligned}
e_{j,t} &= \boldsymbol{v}_\mathrm{a} \tanh(U_\mathrm{a} h_{j-1} + W_\mathrm{a} s_t) \quad \forall j = 1, \cdots, L \quad \text{and} \quad \forall t = 1, \cdots, L \\
\alpha_{j,t} &= \frac{exp(e_{j,t})}{\sum_{k=1}^{L} exp(e_{j,k})} \quad \forall j = 1 \cdots L \quad and \quad t = 1 \cdots L \\
C_j &= \sum_{t=1}^{L} \alpha_{j,t} s_t \quad \forall j = 1 \cdots L
\end{aligned}
\tag{3.1}
$$

All seq2seq models were implemented based on the seq2seq code released by Google [161].

Through unsupervised pre-training, the learned representations capture nonlinear joint dependencies among protein residues or compound atoms that are far from each other in sequence. Such "long-term" dependencies are very important to CPIs since corresponding residues or atoms can be close in 3D structures and jointly contribute to intermolecular interactions.

3.2.2.4 Unified RNN-CNN for supervised learning

With compound and protein representations learned from the above unsupervised learning, we solve the regression problem of compound-protein affinity prediction using supervised learning. For either proteins or compounds, we append a CNN after the RNN (encoders and attention models only) that we just trained. The CNN model consists of a one-dimensional (1D) convolution layer followed by a max-pooling layer. The outputs of the two CNNs (one for proteins and the other for compounds) are concatenated and fed into two more fully connected layers.

The entire RNN-CNN pipeline is trained from end to end [143], with the pre-trained RNNs serving as warm initializations, for improved performance over two-step training. The pre-trained RNN initializations prove to be very important for the non-convex training process [162]. In comparison to such a "unified" model, we also include the "separate" RNN-CNN baseline for comparison, in which we fixed the learned RNN part and train CNN on top of its outputs.

3.2.2.5 Attention mechanisms in unified RNN-CNN

We have also introduced three attention mechanisms to unified RNN-CNN models. The goal is to both improve predictive performances and enable model interpretability at the level of "letters"

(SSEs in proteins and atoms in compounds) and their pairs.

1. **Separate attention**. This default attention mechanism is applied to the compound and the protein separately so the attention learned on each side is non-specific to a compound-protein pair. However, it has the least parameters among the three mechanisms.

2. **Marginalized attention**. To introduce pair-specific attentions, we first use a pairwise "interaction" matrix for a pair and then marginalize it based on maximization over rows or columns for separate compound or protein attention models, which is motivated by [163].

3. **Joint attention**. We have developed this novel attention model to fully explain the pairwise interactions between components (compound atoms and protein SSEs). Specifically, we use the same pairwise interaction matrix but learn to represent the pairwise space and consider attentions on pairwise interactions rather than "interfaces" on each side. Among the three attention mechanisms, joint attention provides the best interpretability albeit with the most parameters.

These attention models (for proteins, compounds, or their pairs) are jointly trained with the RNN/CNN parts. Their learned parameters include attention weights on all "letters" for a given string (or those on all letter-pairs for a given string-pair). Compared to that in unsupervised learning, each attention model here outputs a single vector as the input to its corresponding subsequent 1D-CNN. These attention mechanisms are formally formulated in the next subsections.

3.2.2.5.1 Separate attention We have also introduced protein and compound attention models in supervised learning to both improve predictive performances and enable model interpretability at the level of "letters" (SSEs in proteins and atoms in compounds). In the supervised model we just have the encoder and its attention α_t on each letter t for a given string $\mathbf{x}$ (protein or compound). And the output of the attention model, A, will be the input to the subsequent 1D-CNN model. Suppose that the length of protein encoder is T and $(s_1, \cdots, s_t, \cdots, s_T)$ are the output of protein encoder and similarly the length of compound encoder is D and $(m_1, \cdots, m_d, \cdots, m_D)$ are the output of compound encoder. We parametrize the attention model of unified model with matrix U_a and the vector $\boldsymbol{v}_\mathrm{a}$. Then, The attention model for the protein encoder is formulated as:

$$\begin{aligned}
e_t^P &= \boldsymbol{v}_\mathrm{a}^P \tanh(W_\mathrm{a}^P s_t) \quad \forall t = 1, \cdots, T \\
\alpha_t^P &= \frac{exp(e_t^P)}{\sum_{k=1}^{T} exp(e_k^P)} \quad \forall t = 1 \cdots T \\
A^P &= \sum_{t=1}^{T} \alpha_t^P s_t
\end{aligned} \tag{3.2}$$

Similarly for the compound encoder:

$$\begin{aligned}
e_d^C &= \boldsymbol{v}_\mathrm{a}^C \tanh(W_\mathrm{a}^C m_d) \quad \forall d = 1, \cdots, D \\
\alpha_d^C &= \frac{exp(e_d^C)}{\sum_{k=1}^{D} exp(e_k^C)} \quad \forall d = 1 \cdots D \\
A^C &= \sum_{d=1}^{D} \alpha_d^C m_d
\end{aligned} \tag{3.3}$$

The attention weights (scores) $\alpha_t^{\cdot}$ suggest the importance of the t^{th} “letter” (secondary structure element in proteins and atom or connectivity in compounds) and thus predict the binding sites relevant to the predicted binding affinity.

3.2.2.5.2 Marginalized attention Considering that the separate attention model does not address compound-protein pair specificity, we have exploited a co-attention mechanism similar to [163] which has been widely used in visual question answering. We name this attention mechanism “marginalized attention” because we marginalize over rows or columns of a pair-specific interaction matrix for compound or protein attention in this pair. Specifically, a pairwise “interaction” matrix N of size $T \times D$ is defined with each element as:

$$N_{td} = \tanh(s_t^T W_\mathrm{a} m_d) \quad \forall t = 1, \cdots, T, \quad \forall d = 1, \cdots, D, \tag{3.4}$$

where the parameter matrix W_{a} is now of size $T \times D$.

By considering max column marginalization, we have the marginalized attention model for the protein:

$$\begin{aligned} e^P_t &= \max_{d=1:D}(N_{td}) \quad \forall t = 1, \cdots, T \\ \alpha^P_t &= \frac{exp(e^P_t)}{\sum_{k=1}^{T} exp(e^P_k)} \quad \forall t = 1 \cdots T \\ A^P &= \sum_{t=1}^{T} \alpha^P_t s_t \end{aligned} \tag{3.5}$$

Similarly, by max row marginalization, we have the marginalized attention model for compounds:

$$\begin{aligned} e^C_d &= \max_{t=1:T}(N_{td}) \quad \forall d = 1, \cdots, D \\ \alpha^C_d &= \frac{exp(e^C_d)}{\sum_{k=1}^{D} exp(e^C_k)} \quad \forall d = 1 \cdots D \\ A^C &= \sum_{d=1}^{D} \alpha^C_d m_d \end{aligned} \tag{3.6}$$

3.2.2.5.3 Joint attention We have further developed a novel "joint attention" mechanism that removes the need of marginalization and thus pays attention directly on interacting "letter" pairs rather than individual interfaces. Specifically, for the same pairwise interaction matrix N of size $T \times D$ as defined in Eq. (3.6), we would learn the space for the joint model by another layer of neural network:

$$B_{td} = \tanh(V_{\mathrm{b}} s_t + W_{\mathrm{b}} m_d + b) \quad \forall d = 1 \cdots D, \quad \forall t = 1, \cdots, T \tag{3.7}$$

Then we can derive the attention score α_{td} for each (t, d) pair and the final output A as:

$$\begin{aligned} \alpha_{td} &= \frac{exp(N_{td})}{\sum_{k=1}^{T} \sum_{k'=1}^{D} exp(N_{kk'})} \quad \forall d = 1 \cdots D, \quad \forall t = 1, \cdots, T \\ A &= \sum_{t=1}^{T} \sum_{d=1}^{D} \alpha_{td} B_{td} \end{aligned} \tag{3.8}$$

Throughout the study, we disregarded attention scores on special symbols and re-normalized the rest. For compound SMILES, we further disregarded SMILES symbols that are not English letters when interpreting compound attentions on atoms, which awaits to be improved.

All neural network models for supervised learning were implemented based on TensorFlow [156] and TFLearn [157].

3.2.3 Results

3.2.3.1 Compound and protein representations

We compared the auto-encoding performances of our vanilla seq2seq model and 4 variants: bucketing, bi-directional GRU ("fw+bw"), attention mechanism, and attention mechanism with fw+bw, respectively, in Tables 3.2 and 3.3. We used the common assessment metric in language models, perplexity, which is related to the entropy H of modeled probability distribution P ($\mathrm{Perp}(P) = 2^{H(P)} \geqslant 1$). First, the vanilla seq2seq model had lower test-set perplexity for compound SMILES than protein SPS (7.07 versus 41.03), which echoes the fact that, compared to protein SPS strings, compound SMILES strings are defined in an alphabet of less letters (68 versus 76) and are of shorter lengths (100 versus 152), thus their RNN models are easier to learn. Second, bucketing, the most ad-hoc option among all, did not improve the results much. Third, whereas bi-directional GRUs lowered perplexity by about 2$\sim$3.5 folds and the default attention mechanism did much more for compounds or proteins, they together achieved the best performances (perplexity being 1.0002 for compound SMILES and 1.001 for protein SPS).

	seq2seq	+bucketing	+fw/bw	+attention	+attention+fw/bw
Number of iterations	400K	400K	400K	400K	400K
Training error (perplexity)	7.07	7.09	2.02	1.25	**1.001**
Testing error (perplexity)	6.5	6.91	1.84	1.13	**1.0002**
Time (h)	40.2	49.52	62.93	84.75	82.52

Table 3.2: Performance comparison among 5 variants of seq2seq for compound representation based on perplexity under the limit of 4-day running time and 400K iterations.

	seq2seq	+bucketing	+fw/bw	+attention	+attention+fw/bw
Number of iterations	400K	400K	400K	153K	153K
Training error (perplexity)	40.85	40.52	16.77	1.007	**1.003**
Testing error (perplexity)	41.03	43.19	19.62	1.001	**1.001**
Time (h)	80.7	83.75	79.89	96	96

Table 3.3: Performance comparison among 5 variants of seq2seq for protein representations based on perplexity under the limit of 4-day running time and 400K iterations.

Therefore, the last seq2seq variant, bidirectional GRUs with attention mechanism, is regarded the most appropriate one for learning compound/protein representations and adopted thereinafter.

3.2.3.2 Compound-protein affinity prediction

3.2.3.2.1 Comparing novel representations to baseline ones To assess how useful the learned/encoded protein and compound representations are for predicting compound-protein affinity, we compared the novel and baseline representations in affinity regression using the labeled datasets. The representations were compared under the same shallow machine learning models – ridge regression, lasso regression and random forest (RF). As shown in Table 3.4 our novel representations learned from SMILES/SPS strings by seq2seq outperform baseline representations of k-hot encoding of molecular/Pfam features. For the best performing RF models, using 46% less training time and 24% less memory, the novel representations achieved the same performance over the default test set as the baseline ones and lowered root mean squared errors (RMSE) for two of the four generalization sets whose target protein classes (nuclear estrogen receptors / ER and ion channels) are not included in the training set. Similar improvements were observed on pK_i, pK_d, and pEC$_{50}$ predictions in Tables 3.5, 3.6, and 3.7 respectively.

These results show that learning protein or compound representations even from unlabeled datasets alone could improve their context-relevance for various labels. We note that, unlike Pfam-based protein representations that exploit curated information only available to some proteins and their homologs, our SPS representations do not assume such information and can apply to uncharacterized proteins lacking annotated homologs.

	Baseline representations			Novel representations		
	Ridge	Lasso	RF	Ridge	Lasso	RF
Training	1.16 (0.60)	1.16 (0.60)	0.76 (0.86)	1.23 (0.54)	1.22 (0.55)	**0.63** (0.91)
Testing	1.16 (0.60)	1.16 (0.60)	**0.91** (0.78)	1.23 (0.54)	1.22 (0.55)	**0.91** (0.78)
ER	1.43 (0.30)	1.43 (0.30)	1.44 (0.37)	1.46 (0.18)	1.48 (0.18)	**1.41** (0.26)
Ion Channel	1.32 (0.22)	1.34 (0.20)	1.30 (0.22)	1.26 (0.23)	1.32 (0.17)	**1.24** (0.30)
GPCR	**1.28** (0.22)	1.30 (0.22)	1.32 (0.28)	1.34 (0.20)	1.37 (0.17)	1.40 (0.25)
Tyrosine Kinase	**1.16** (0.38)	1.16 (0.38)	1.18 (0.42)	1.50 (0.11)	1.51 (0.10)	1.58 (0.11)
Time (core hours)	3.5	7.4	1239.8	0.47	2.78	668.7
Memory (GB)	7.6	7.6	8.3	7.3	7.3	6.3

Table 3.4: Comparing the novel representations to the baseline based on RMSE (and Pearson correlation coefficient r) of pIC$_{50}$ shallow regression.

	Baseline representations			Novel representations		
	Ridge	Lasso	RF	Ridge	Lasso	RF
Training	1.23 (0.60)	1.21 (0.62)	0.83 (0.84)	1.26 (0.58)	1.26 (0.58)	**0.67** (0.91)
Testing	1.24 (0.60)	1.22 (0.61)	**0.97** (0.78)	1.27 (0.58)	1.27 (0.58)	**0.97** (0.78)
ER	1.37 (0.10)	1.38 (0.10)	1.35 (0.24)	1.50 (0.17)	**1.34** (0.14)	1.48 (0.14)
Ion Channel	1.44 (0.14)	1.44 (0.14)	**1.38** (0.24)	1.52 (0.10)	1.66 (0.10)	1.46 (0.21)
GPCR	1.27 (0.20)	1.28 (0.17)	1.23 (0.30)	1.44 (0.10)	1.44 (0.10)	**1.20** (0.19)
Tyrosine Kinase	1.41 (0.41)	1.44 (0.38)	**1.43** (0.52)	1.73 (0.20)	1.77 (0.16)	1.75 (0.10)
Time (core hours)	2	2.22	333.2	0.22	1.42	236.21
Memory (GB)	3.2	3.3	3.6	3.2	3.1	2.8

Table 3.5: Comparing the baseline and the novel representations based on RMSE (and Pearson correlation coefficient r) of pK_i shallow regression.

3.2.3.2.2 Comparing shallow and deep models Using the novel representations we next compared the performances of affinity regression between the best shallow model (RF) and various deep models. For both separate and unified RNN-CNN models, we tested results from a single

	Baseline representations			Novel representations		
	Ridge	Lasso	RF	Ridge	Lasso	RF
Training	1.04 (0.72)	1.10 (0.70)	0.97 (0.78)	1.17 (0.64)	1.21 (0.61)	**0.76** (0.88)
Testing	1.33 (0.67)	1.19 (0.65)	1.11 (0.70)	1.24 (0.60)	1.28 (0.55)	**1.10** (0.70)
Ion Channel	**1.53** (0.46)	1.62 (0.37)	1.74 (0.30)	1.81 (0.18)	1.77 (0.16)	1.79 (0.09)
GPCR	1.75 (0.07)	1.74 (0.07)	1.58 (0.09)	1.55 (0.10)	**1.53** (0.13)	1.57 (0.07)
Tyrosine Kinase	1.35 (0.45)	1.36 (0.45)	**1.32** (0.50)	1.56 (0.18)	1.59 (0.13)	1.46 (0.31)
Time (core hours)	0.1	0.16	4.78	0.01	0.06	10.34
Memory (Gb)	0.4	0.4	0.4	0.4	0.4	0.4

Table 3.6: Comparing the baseline and the novel representations based on RMSE (and Pearson correlation coefficient r) of pK_d shallow regression.

	Baseline representations			Novel representations		
	Ridge	Lasso	RF	Ridge	Lasso	RF
Training	1.02 (0.72)	0.98 (0.74)	0.74 (0.86)	0.92 (0.78)	0.97 (0.74)	**0.64** (0.90)
Testing	1.04 (0.70)	1.02 (0.72)	0.90 (0.79)	0.93 (0.77)	0.98 (0.74)	**0.80** (0.84)
ER	1.55 (0.20)	1.60 (0.17)	1.71 (0.17)	1.72 (0.14)	1.55 (0.17)	**1.51** (0.13)
Ion Channel	1.25 (0.44)	1.26 (0.43)	1.21 (0.48)	1.30 (0.40)	1.39 (0.30)	**1.20** (0.33)
GPCR	1.34 (0.20)	1.38 (0.14)	1.39 (0.14)	1.31 (0.24)	1.34 (0.22)	**1.31** (0.24)
Tyrosine Kinase	1.25 (0.09)	1.30 (0.08)	1.21 (0.09)	1.33 (0.25)	1.63 (0.26)	**1.07** (0.10)
Time (core hours)	0.34	0.42	38.46	0.05	0.26	31.67
Memory (GB)	1	1	1.1	0.9	0.9	0.8

Table 3.7: Comparing the baseline and the novel representations based on RMSE (and Pearson correlation coefficient r) of pEC$_{50}$ shallow regression.

model with (hyper)parameters optimized over the training/validation set, averaging a "parameter ensemble" of 10 models derived in the last 10 epochs, and averaging a "parameter+NN" ensemble of models with varying number of neurons in the fully connected layers ((300,100), (400,200) and (600,300)) trained in the last 10 epochs. The attention mechanism used here is the default, separate attention.

From Table 3.8 we noticed that unified RNN-CNN models outperform both random forest and separate RNN-CNN models (the similar performances between RF and separate RNN-CNN indicated a potential to further improve RNN-CNN models with deeper models). By using a relatively small amount of labeled data that are usually expensive and limited, protein and compound representations learned from abundant unlabeled data can be tuned to be more task-specific. We also noticed that averaging an ensemble of unified RNN-CNN models further improves the performances especially for some generalization sets of ion channels and GPCRs. As anticipated, averaging ensembles of models reduces the variance originating from network architecture and parameter optimization thus reduces expected generalization errors. Similar observations were made for pK_i predictions as well (Table 3.9) even when their hyper-parameters were not particularly optimized and simply borrowed from pIC$_{50}$ models. Impressively, unified RNN-CNN models without very deep architecture could predict IC$_{50}$ values with relative errors below $10^{0.7}$=5 fold (or 1.0 kcal/mol) for the test set and even around $10^{1.3} = 20$ fold (or 1.8 kcal/mol) on average for protein classes not seen in the training set. Interestingly, GPCRs and ion channels had similar RMSE but more different Pearson's r, which is further described by the distributions of predicted versus measured pIC$_{50}$ values for various sets (Fig. 3.2).

	RF	Separate RNN-CNN Models			Unified RNN-CNN Models		
		single	parameter ensemble	parameter+NN ensemble	single	parameter ensemble	parameter+NN ensemble
Training	0.63 (0.91)	0.68 (0.88)	0.67 (0.90)	0.68 (0.89)	0.47 (0.94)	0.45 (0.95)	**0.44** (0.95)
Testing	0.91 (0.78)	0.94 (0.76)	0.92 (0.77)	0.90 (0.79)	0.78 (0.84)	0.77 (0.84)	**0.73** (0.86)
Generalization – ER	**1.41** (0.26)	1.45 (0.24)	1.44 (0.26)	1.43 (0.28)	1.53 (0.16)	1.52 (0.19)	1.46 (0.30)
Generalization – Ion Channel	**1.24** (0.30)	1.36 (0.18)	1.33 (0.18)	1.29 (0.25)	1.34 (0.17)	1.33 (0.18)	1.30 (0.18)
Generalization – GPCR	1.40 (0.25)	1.44 (0.19)	1.41 (0.20)	1.37 (0.23)	1.40 (0.24)	1.40 (0.24)	**1.36** (0.30)
Generalization – Tyrosine Kinase	1.58 (0.11)	1.66 (0.09)	1.62 (0.10)	1.54 (0.12)	1.24 (0.39)	1.25 (0.38)	**1.23** (0.42)

Table 3.8: Under novel representations learned from seq2seq, comparing random forest and variants of separate RNN-CNN and unified RNN-CNN models based on RMSE (and Pearson's r) for pIC$_{50}$ prediction.

3.2.3.2.3 Comparing attention mechanisms in prediction To assess the predictive powers of the three attention mechanisms, we compared their pIC$_{50}$ predictions in Table 3.10 using the same dataset and the same unified RNN-CNN models as before. All attention mechanisms had similar performances on the training and test sets. However, as we anticipated, separate attention with the least parameters edged joint attention in generalization (especially for receptor tyrosine kinases).

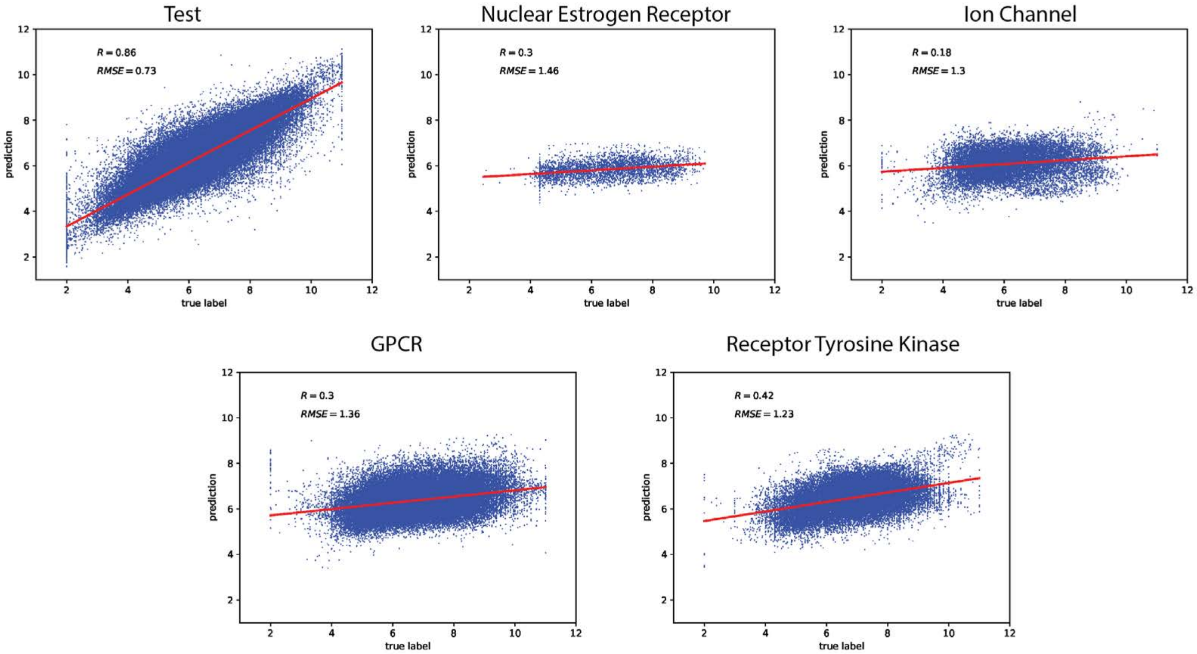


Figure 3.2: Comparing predictions vs real labels for test and generalization tests for the unified RNN-CNN model (separate attention).

	RF	Unified RNN-CNN Models (separate attention)			Unified RNN-CNN Models (joint attention)		
		single	parameter ensemble	parameter+NN ensemble	single	parameter ensemble	parameter+NN ensemble
Training	0.67 (0.91)	0.44 (0.95)	0.42 (0.95)	0.41 (0.96)	0.48 (0.94)	0.44 (0.95)	**0.40** (0.96)
Testing	0.97 (0.78)	0.84 (0.84)	0.83 (0.84)	**0.79** (0.86)	0.91 (0.81)	0.88 (0.82)	0.80 (0.85)
ER	**1.48** (0.14)	1.76 (0.17)	1.74 (0.17)	1.62 (0.07)	1.76 (0.09)	1.78 (0.07)	1.63 (0.10)
Ion Channel	1.46 (0.21)	1.50 (0.21)	1.49 (0.20)	**1.41** (0.29)	1.79 (0.23)	1.78 (0.24)	1.51 (0.32)
GPCR	**1.20** (0.19)	1.35 (0.28)	1.33 (0.29)	1.26 (0.36)	1.50 (0.21)	1.48 (0.23)	1.35 (0.28)
Tyrosine Kinase	**1.75** (0.10)	1.83 (0.27)	1.81 (0.27)	1.85 (0.25)	2.10 (0.16)	2.08 (0.15)	1.95 (0.17)

Table 3.9: Under novel representations learned from seq2seq, comparing random forest and variants of separate RNN-CNN and unified RNN-CNN models based on RMSE (and Pearson correlation coefficient r) for pK_i prediction.

Meanwhile, joint attention had similar predictive performances and much better interpretability, thus will be further examined in case studies of selective drugs.

	Separate attention			Marginalized attention			Joint attention		
	single	parameter ensemble	parameter+NN ensemble	single	parameter ensemble	parameter+NN ensemble	single	parameter ensemble	parameter+NN ensemble
Training	0.47 (0.94)	0.45 (0.95)	0.44 (0.95)	0.50 (0.94)	0.47 (0.95)	0.42 (0.96)	0.48 (0.94)	0.44 (0.94)	**0.40** (0.95)
Testing	0.78 (0.84)	0.77 (0.84)	**0.73** (0.86)	0.81 (0.83)	0.79 (0.84)	**0.73** (0.86)	0.84 (0.82)	0.80 (0.83)	**0.73** (0.86)
Generalization – ER	1.53 (0.16)	1.52 (0.19)	1.46 (0.30)	1.69 (0.20)	1.67 (0.20)	1.53 (0.30)	1.78 (0.03)	1.68 (0.04)	**1.37** (0.23)
Generalization – Ion Channel	1.34 (0.17)	1.33 (0.18)	**1.30** (0.18)	1.63 (0.01)	1.64 (0.06)	1.41 (0.13)	1.54 (0.25)	1.53 (0.26)	1.42 (0.26)
Generalization – GPCR	1.40 (0.24)	1.40 (0.24)	**1.36** (0.30)	1.59 (0.17)	1.57 (0.18)	1.42 (0.24)	1.53 (0.19)	1.53 (0.19)	1.38 (0.25)
Generalization – Tyrosine Kinase	1.24 (0.39)	1.25 (0.38)	**1.23** (0.42)	1.69 (0.22)	1.62 (0.25)	1.50 (0.32)	2.22 (0.18)	2.17 (0.21)	2.04 (0.17)

Table 3.10: Under novel representations learned from seq2seq, comparing different attention mechanisms of unified RNN-CNN models based on RMSE (and Pearson correlation coefficient r for pIC$_{50}$ prediction.

3.2.3.2.4 Deep transfer learning for new classes of protein targets Using the generalization sets, we proceed to explain and address our models' relatively worse performances for new classes of protein targets without any training data. We chose to analyze separate attention models with the best generalization results and first noticed that proteins in various sets have different distributions in the SPS alphabet (4-tuples). In particular, the test set, ion channels/GPCRs/tyrosine kinases, and estrogen receptors are increasingly different from the training set (measured by Jensen-Shannon distances in SPS letter or SPS length distribution), which correlated with increasingly deteriorating performance relative to the training set (measured by the relative difference in RMSE) with a Pearson's r of 0.68 (SPS letter distribution) or 0.96 (SPS length distribution) (Fig. 3.3).

To improve the performances for new classes of proteins, we compare two strategies: re-training shallow models (random forest) from scratch based on new training data alone and "transferring" original deep models (unified parameter+NN ensemble with the default separate attention) to fit new data. The reason is that new classes of targets often have few labeled data that might be adequate for re-training class-specific shallow models from scratch but not for deep models with much more parameters.

As shown in Fig. 3.4, deep transfer learning models increasingly improved the predictive performance compared to the original deep learning models, when increasing amount of labeled data are made available for new protein classes. The improvement was significant even with 1% training coverage for each new protein class. Notably, deep transfer learning models outperformed random forest models that were re-trained specifically for each new protein class.

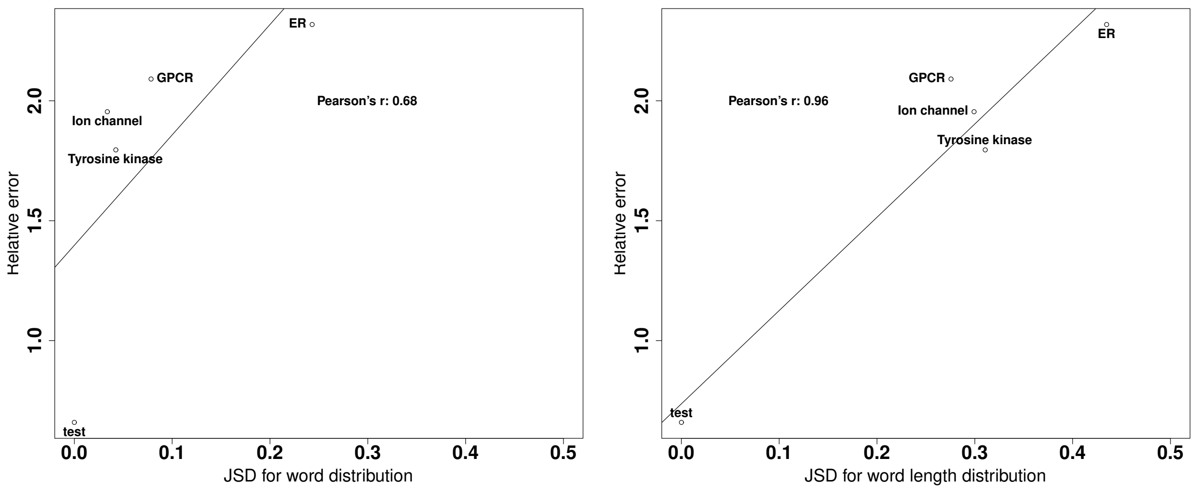


Figure 3.3: Relative errors to the training set (y axis) versus Jensen-Shannon distances from the training-set protein SPS letter distribution (x axis: left) or SPS length distribution (x axis: right) for various sets of protein targets.

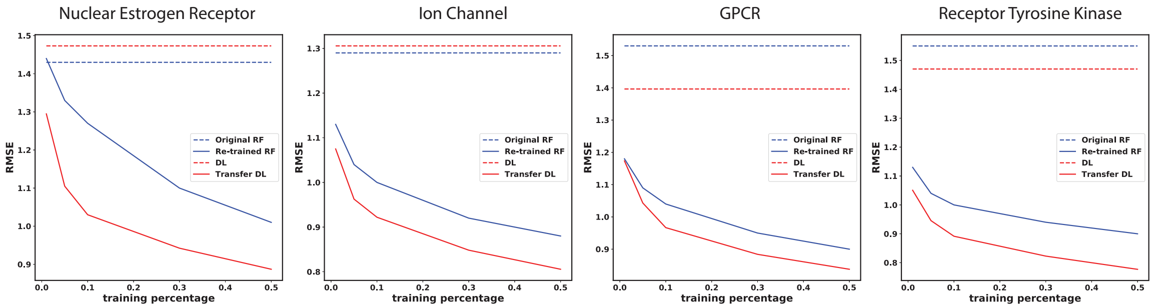


Figure 3.4: Comparing strategies to generalize predictions for four sets of new protein classes: original random forest (RF), original param.+NN ensemble of unified RNN-CNN models (DL for deep learning with the default attention), and re-trained RF or transfer DL using incremental amounts of labeled data in each set.

3.2.3.3 Predicting target selectivity of drugs

We went on to test how well our unified RNN-CNN models could predict certain drugs' target selectivity, using 3 sets of drug-target interactions of increasing prediction difficulty. Our novel representations and models successfully predicted target selectivity for 6 of 7 drugs whereas baseline representations and shallow models (random forest) failed for most drugs.

3.2.3.3.1 Factor Xa versus Thrombin Thrombin and factor X (Xa) are important proteins in the blood coagulation cascade. Antithrombotics, inhibitors for such proteins, have been developed to treat cardiovascular diseases. Due to thrombin's other significant roles in cellular functions and neurological processes, it is desirable to develop inhibitors specifically for factor Xa. Compound DX-9065a is such a selective inhibitor (pK_i value being 7.39 for Xa and <2.70 for thrombin) [164].

We used the learned pK_i models in this study. Both proteins were included in the K_i training set with 2,294 and 2,331 samples, respectively, but their interactions with DX-9065a were not. Table 3.11 suggested that random forest correctly predicted the target selectivity (albeit with less than 0.5-unit margin) using baseline representations but failed using novel representations. In contrast, our models with separate and joint attention mechanisms both correctly predicted the compound's favoring Xa. Moreover, our models predicted selectivity levels being 2.4 (separate attention) and 3.9 (joint attention) in pK_i difference (ΔpK_i), where the joint attention model produced predictions very close to the known selectivity margin (ΔpK_i $\geqslant$4.7).

	Baseline rep. + RF	Novel rep. + RF	Novel rep. + DL (sep. attn.)	Novel rep. + DL (joint attn.)
Thrombin	6.36	**6.71**	5.68	4.77
Factor Xa	**6.87**	6.54	**8.08**	**8.64**

Table 3.11: Predicted pK_i values and target specificity for compound DX-9065a interacting with human factor Xa and thrombin.

3.2.3.3.2 Cyclooxygenase (COX) protein family COX protein family represents an important class of drug targets for inflammatory diseases. These enzymes responsible for prostaglandin biosynthesis include COX-1 and COX-2 in human, both of which can be inhibited by nonsteroidal anti-inflammatory drugs (NSAIDs). We chose three common NSAIDs known for human COX-1/2 selectivity: celecoxib/CEL (pIC$_{50}$ for COX-1: 4.09; COX-2: 5.17), ibuprofen/IBU (COX-1: 4.92, COX-2: 4.10) and rofecoxib/ROF (COX-1: <4; COX-2: 4.6) [165]. This is a very challenging case for selectivity prediction because selectivity levels of all NSAIDs are close to or within 1 unit of pIC_{50}. We used the learned pIC_{50} ensemble models in this study. COX-1 and COX-2 both exist in our IC_{50} training set with 959 and 2,006 binding examples, respectively, including 2 of the 6 compound-protein pairs (CEL and IBU with COX-1 individually).

	Baseline rep. + RF			Novel rep. + RF			Novel rep. + DL (sep. attn.)			Novel rep. + DL (joint attn.)		
	CEL	IBU	ROF	CEL	IBU	ROF	CEL	IBU	ROF	CEL	IBU	ROF
COX-1	6.06	5.32	5.71	6.41	6.12	6.13	5.11	**6.06**	5.67	5.18	**5.94**	6.00
COX-2	6.06	5.32	5.71	**6.57**	**6.19**	**6.21**	**7.60**	5.96	**6.51**	**7.46**	5.62	**6.03**

Table 3.12: pIC_{50} predictions and target specificity for three NSAIDs interacting with human COX-1 and COX-2.

From Table 3.12, we noticed that, using the baseline representations, random forest incorrectly predicted COX-1 and COX-2 to be equally favorable targets for each drug. This is because the two proteins are from the same family and their representations in Pfam domains are indistinguishable. Using the novel representations, random forest correctly predicted target selectivity for two of the three drugs (CEL and ROF), whereas our unified RNN-CNN models (both attention mechanisms) did so for all three. Even though the selectivity levels of the NSAIDs are very challenging to predict, our models were able to predict all selectivities correctly with the caveat that few predicted differences might not be statistically significant.

3.2.3.3.3 Protein-tyrosine phosphatase (PTP) family Protein-tyrosine kinases and protein-tyrosine phosphatases (PTPs) are controlling reversible tyrosine phosphorylation reactions which are critical for regulating metabolic and mitogenic signal transduction processes. Selective PTP inhibitors are sought for the treatment of various diseases including cancer, autoimmunity, and diabetes. Compound 1 [2-(oxalyl-amino)-benzoic acid or OBA] and its derivatives, compounds 2 and 3 (PubChem CID: 44359299 and 90765696), are highly selective toward PTP1B rather than other proteins in the family such as PTPRA, PTPRE, PTPRC and SHP1 [166]. Specifically, the pK_i values of OBA, compound 2, and compound 3 against PTP1B are 4.63, 4.25, and 6.69, respectively; and their pK_i differences to the closest PTP family protein are 0.75, 0.7, and 2.47, respectively [166].

We used the learned pK_i ensemble models in this study. PTP1B, PTPRA, PTPRC, PTPRE and SHP1 were included in the K_i training set with 343, 33, 16, 6 and 5 samples respectively. These examples just included OBA binding to all but SHP1 and compound 2 binding to PTPRC.

	Baseline rep. + RF			Novel rep. + RF			Novel rep. + DL (sep. attn.)			Novel rep. + DL (joint attn.)		
Protein	Comp1	Comp2	Comp3	Comp1	Comp2	Comp3	Comp1	Comp2	Comp3	Comp1	Comp2	Comp3
PTP1B	4.15	3.87	5.17	6.70	6.55	6.71	3.76	**3.84**	**3.92**	2.84	**4.10**	**4.04**
PTPRA	4.15	3.87	5.17	6.29	6.59	6.27	2.73	2.90	3.44	2.39	2.62	2.12
PTPRC	4.15	3.87	5.17	**6.86**	6.73	**6.87**	3.37	3.25	3.19	3.36	3.49	2.97
PTPRE	4.15	3.87	5.17	6.79	6.68	6.81	**3.83**	3.75	3.85	2.75	2.93	2.61
SHP1	4.15	3.87	5.17	6.71	**6.74**	6.73	3.37	3.38	3.89	**3.42**	3.52	3.22

Table 3.13: Predicted pK_i values and target specificity for three PTP1B-selective compounds interacting with five proteins in the human PTP family.

Results in Table 3.13 showed that random forest using baseline representations cannot tell binding affinity differences within the PTP family as the proteins' Pfam descriptions are almost indistinguishable. Using novel representations, random forest incorrectly predicted target selectivity for all 3 compounds, whereas unified RNN-CNN models with both attention mechanisms correctly did so for all but one (compound 1 – OBA). We also noticed that, although the separate attention model predicted likely insignificant selectivity levels for compounds 2 (Δp$K_i = 0.09$) and 3 (Δp$K_i = 0.03$), the joint attention model much improved the prediction of selectivity margins (Δp$K_i = 0.58$ and 0.82 for compounds 2 and 3, respectively) and their statistical significances.

3.2.3.4 Explaining target selectivity of drugs

After successfully predicting target selectivity for some drugs, we proceed to explain using attention scores how our deep learning models did so and what they reveal about those compound-protein interactions.

3.2.3.4.1 How do the compound-protein pairs interact? Given that SPS and SMILES strings are interpretable and attention models between RNN encoders and 1D convolution layers can report their focus, we pinpoint SSEs in proteins and atoms in compounds with high attention scores, which are potentially responsible for CPIs. We chose 3 compound-protein pairs that have 3D crystal complex structures from the Protein Data Bank; and extracted residues in direct contacts with ligands (their SSEs are regarded ground truth for binding site) for each protein from ligplot diagrams provided through PDBsum [167].

After correcting and marginalizing joint attention scores, we picked the top 10% (4) SSEs in resulting β_i as predicted binding sites. More specifically, we first corrected joint attention scores α_{ij} on pairs of protein SSE i and compound atom j in the single unified RNN-CNN model to be $\beta_{ij} = \alpha_{ij} - \left(\sum_{k=1}^{I} \alpha_{kj}\right)/I \quad (\forall i = 1, \ldots, I, \quad j = 1, \ldots, J)$ to offset the contribution of any compound atom j with promiscuous attentions over all protein SSEs. We then calculated the attention score β_i for protein SSE i by max-marginalization ($\beta_i = \max_j \beta_{ij}$). No negative β_i was found in this case thus no further treatment was adopted..

Table 3.14 shows that, compared to randomly ranking the SSEs, our approach can enrich binding site prediction by 1.7∼5.8 fold for the three CPIs. Consistent with the case of target selectivity prediction, joint attention performed better than separate attention did. One-sided paired t-tests suggested that binding sites enjoyed higher attention scores than non-binding sites in a statistically significant way. When the strict definition of binding sites is relaxed to residues within 5Å of any heavy atom of the ligand, results were further improved with all top 10% SSEs of factor Xa being at the binding site (Table 3.15).

		Number of SSEs		Top 10% (4) SSEs predicted as binding site by sep. attn.				Top 10% (4) SSEs predicted as binding site by joint attn.			
Target–Drug	PDB ID	total	binding site	# of TP	Enrichment	Highest rank	P value	# of TP	Enrichment	Highest rank	P value
Human COX2–rofecoxib	5KIR	40	6	2	2.22	1	1.18e-9	1	1.68	4	0.0107
Human PTP1B–OBA	1C85	34	5	0	0	20	1	1	1.7	1	1.12e-10
Human factor Xa–DX9065	1FAX	31	4	0	0	7	0.898	3	5.81	2	2.2e-16

Table 3.14: Interpreting deep learning models: predicting binding sites based on joint attentions. The binding site here is defined as SSEs making direct contacts with compounds (according to the LIGPLOT service from PDBsum).

		Number of SSEs		Top 10% (4) SSEs predicted as binding site by joint attn.			
Target–Drug	PDB ID	total	binding site	# of TP	Enrichment	Highest rank	P value
Human COX2–rofecoxib	5KIR	40	9	1	1.11	4	1.3e-1
Human PTP1B–OBA	1C85	34	6	2	3.77	1	<2.2e-16
Human factor Xa–DX9065	1FAX	31	6	4	5.16	1	<2.2e-16

Table 3.15: Interpreting deep learning models: predicting binding sites based on joint attentions. The binding site here is defined as SSEs falling within 5Å from compound heavy atoms.

We delved into the predictions for factor Xa–DX-9065a interaction in Fig. 3.5A. Warmer colors (higher attentions) are clearly focused near the ligand. The red loops connected through a β strand (resi. 171–196) were correctly predicted to be at the binding site with a high rank 2, thus a true positive (TP). The SSE ranked first, a false positive, is its immediate neighbor in sequence (resi. 162-170; red helix at the bottom) and is near the ligand. In fact, as mentioned before, when the binding site definition is relaxed, all top 10% SSEs were at the binding site. Therefore, in the current unified RNN-CNN model with attention mechanism, wrong attention could be paid to sequence neighbors of ground truth; and additional information (for instance, 2D contact maps or 3D structures of proteins, if available) could be used as additional inputs to reduce false negatives.

We also examined attention scores β_j on compound atoms (j). Many high attention scores were observed (Fig. 3.6), which is somewhat intuitive as small-molecule compounds usually fit in protein pockets or grooves almost entirely. The top-ranked atom happened to be a nitrogen atom forming a hydrogen bond with an aspartate (Asp189) of factor Xa, although more cases need to be studied more thoroughly for a conclusion.

3.2.3.4.2 How are targets selectively interacted? To predictively explain the selectivity origin of compounds, we designed an approach to compare attention scores between pairs of CPIs and tested it using factor Xa-selective DX-9065a with known selectivity origin. Specifically, position 192 is a charge-neutral polar glutamine (Gln192) in Xa but a negatively-charged glutamate (Glu192) in thrombin [168]. DX-9065a exploited this difference with a carboxylate group forming unfavorable electrostatic repulsion with Glu192 in thrombin but favorable hydrogen bond with Gln192 in Xa.

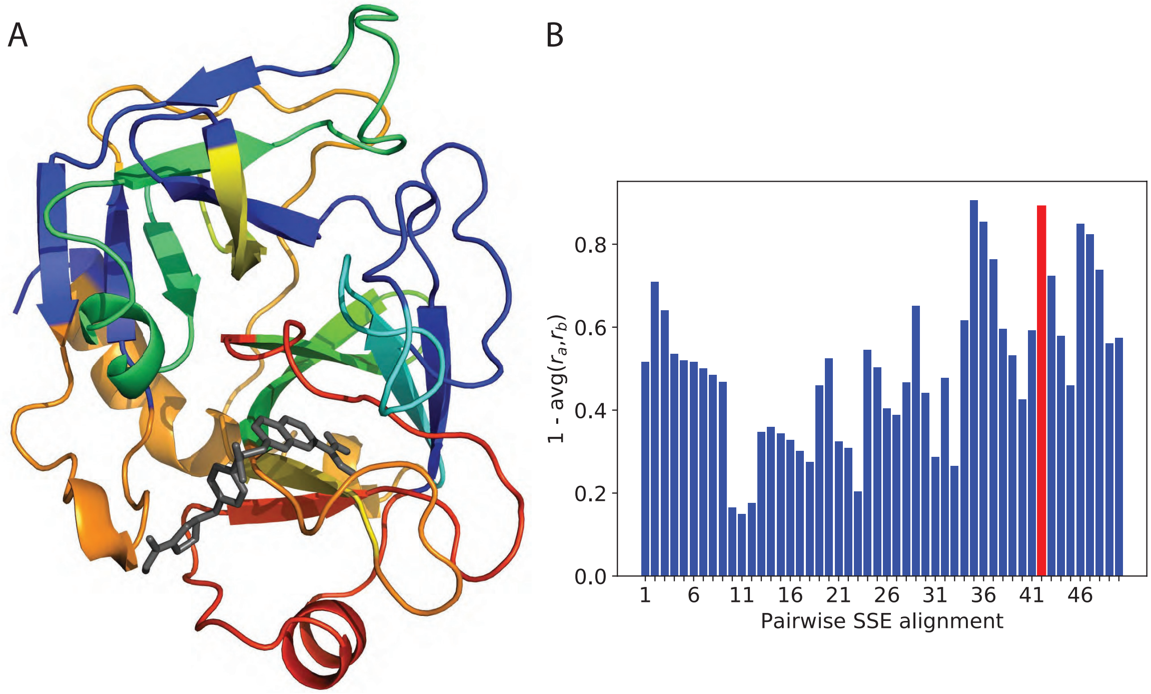


Figure 3.5: Interpreting deep learning models for predicting factor Xa (A) binding site and (B) selectivity origin based on joint attention. (A) 3D structure of factor Xa (colored cartoon representation) in complex with DX-9065a (black sticks) (PDB ID:1FAX) where protein SSEs are color-coded by attention scores (β_i), warmer colors indicating higher attentions. (B) Segments of factor Xa are scored by one less the average of the β_i rank ratios for the two compound-protein interactions where the ground truth of the selectivity origin is in red.

To compare DX-9065a interacting with the two proteins, we performed amino-acid sequence alignment between the proteins and split two sequences of mis-matched SSEs (count: 31 and 38) into those of perfectly matched segments (count: 50 and 50). In the end, segment 42, where SSE 26 of Xa and SSE 31 of thrombin align, is the ground truth containing position 192 for target selectivity.

For DX-9065a interacting with either factor Xa or thrombin, we ranked the SSEs based on the attention scores β_i and assigned each segment the same rank as its parent SSE. Due to the different SSE counts in the two proteins, we normalized each rank for segment i by the corresponding SSE count for a rank ratio r^i. For each segment we then substracted from 1 the average of rank ratios between factor Xa and thrombin interactions so that highly attended segments in both proteins can

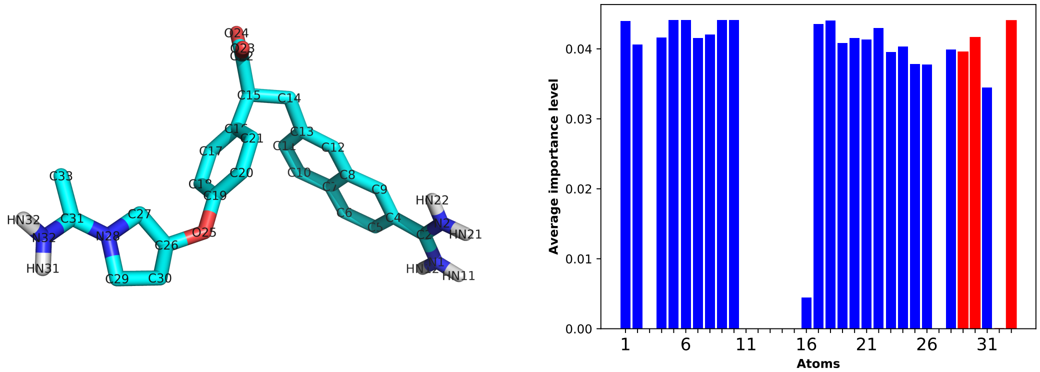


Figure 3.6: (Left figure) The chemical structure and atom names for compound DX-9065a, a selective ligand for factor Xa. (Right figure) Max-marginalized attention scores β_j's for compound DX-9065a interacting with factor Xa.

be scored higher. Fig. 3.5B shows that the ground-truth segment in red was ranked the 2nd among 50 segments albeit with narrow margins over the next 3 segments.

3.2.4 Discussion

We lastly explore alternative representations of proteins and compounds and discuss remaining challenges.

3.2.4.1 Protein representations using amino acid sequences

As shown earlier, our SPS representations integrate both sequence and structure information of proteins and are much more compact compared to the original amino acid sequences. That being said, there is a value to consider a protein sequence representation with the resolution of residues rather than SSEs: potentially higher-resolution precision and interpretability. We started with unsupervised learning to encode the protein sequence representation with seq2seq.

	SPS rep. +attention+fw/bw	seq. rep. +attention+fw/bw
Training error (Perplexity)	**1.003**	11.46
Testing error (Perplexity)	**1.001**	12.69
Time (h)	**96**	192

Table 3.16: Comparing the auto-encoding performance between amino acid and SPS sequences using the best seq2seq model (bidirectional GRU with attention).

Compared to SPS representations, protein sequences are 10-times longer and demanded 10-times more GRUs in seq2seq, which suggests much more expensive training. Under the limited computational budget, we trained the protein sequence seq2seq models using twice the time limit on the SPS ones. The perplexity for the test set turned out to be over 12, which is much worse than 1.001 in the SPS case and deemed inadequate for subsequent (semi-)supervised learning. Learning very long sequences is still challenging and calls for advanced architectures of sequence models.

3.2.4.2 Compound representation using chemical graphs

We have chosen SMILES representations for compounds partly due to recent advancements of sequence models especially in the field of natural language processing. Meanwhile, the descriptive power of SMILES strings can have limitations. For instance, some syntactically invalid SMILES strings can still correspond to valid chemical structures. Therefore, we also explore chemical formulae (2D graphs) for compound representation.

We replaced RNN layers for compound sequences with graph CNN (GCNN) in our unified model (separate attention) and kept the rest of the architecture. This new architecture is named unified RNN/GCNN-CNN. The GCNN part is adopting a very recently-developed method [169] for compound-protein interactions. Atom types have been used for node features and no edge features have been utilized.

	SMILES rep.			Graph rep.		
	single	parameter ensemble	parameter+NN ensemble	single	parameter ensemble	parameter+NN ensemble
Training	0.47 (0.94)	0.45 (0.95)	**0.44** (0.95)	0.55 (0.92)	0.54 (0.92)	0.55 (0.92)
Testing	0.78 (0.84)	0.77 (0.84)	**0.73** (0.86)	1.50 (0.35)	1.50 (0.35)	1.34 (0.45)
Generalization – ER	1.53 (0.16)	1.52 (0.19)	**1.46** (0.30)	1.68 (0.05)	1.67 (0.03)	1.67 (0.07)
Generalization – Ion Channel	1.34 (0.17)	1.33 (0.18)	**1.30** (0.18)	1.43 (0.10)	1.41 (0.13)	1.35 (0.12)
Generalization – GPCR	1.40 (0.24)	1.40 (0.24)	**1.36** (0.30)	1.63 (0.04)	1.61 (0.04)	1.49 (0.07)
Generalization – Tyrosine Kinase	1.24 (0.39)	1.25 (0.38)	**1.23** (0.42)	1.74 (0.01)	1.71 (0.03)	1.70 (0.03)

Table 3.17: Comparing unified RNN-CNN and unified RNN/GCNN-CNN based on RMSE (and Pearson's r) for pIC$_{50}$ prediction.

Results in Table 3.17 indicate that the unified RNN/GCNN-CNN model using compound graphs did not outperform the unified RNN-CNN model using compound SMILES in RMSE and did a lot worse in Pearson's r. These results did not show the superiority of SMILES versus graphs for compound representations *per se*. Rather, they show that graph models need new architectures and further developments to address the challenge. We have further advanced GCN models for CPI problems in the next sections (see 3.3) by utilizing more informative features from both atoms and their edge types.

3.2.5 Conclusion

We have developed accurate and interpretable deep learning models for predicting compound-protein affinity using only compound identities and protein sequences. By taking advantage of massive unlabeled compound and protein data besides labeled data in semi-supervised learning, we have jointly trained unified RNN-CNN models from end to end for learning context- and task-specific protein/compound representations and predicting compound-protein affinity. These models outperform baseline machine-learning models. Impressively, they achieve the relative error of IC$_{50}$ within 5-fold for a comprehensive test set and even that within 20-fold for generalization sets of protein classes unknown to training. Deeper models would further improve the results.

Moreover, for the generalization sets, we have devised transfer-learning strategies to significantly improve model performance using as few as about 30 labeled samples.

Compared to conventional compound or protein representations using molecular descriptors or Pfam domains, the encoded representations learned from novel structurally-annotated SPS sequences and SMILES strings improve both predictive power and training efficiency for various machine learning models. Given the novel representations with better interpretability, we have included attention mechanism in the unified RNN-CNN models to quantify how much each part of proteins, compounds, or their pairs are focused while the models are making the specific prediction for each compound-protein pair.

When applied to case studies on drugs of known target-selectivity, our models have successfully predicted target selectivity in all cases whereas conventional compound/protein representations and machine learning models have failed some. Furthermore, our analyses on attention weights have shown promising results for predicting protein binding sites as well as the origins of binding selectivity, thus calling for further method development for better interpretability.

For protein representation, we have chosen SSE as the resolution for interpretability due to the known sequence-size limitation of RNN models [152]. One can easily increase the resolution to residue-level by simply feeding to our models amino-acid sequences instead of SPS sequences, but needs to be aware of the much increased computational burden and much worse convergence when training RNNs. For compound representation, we have started with 1D SMILES strings and have also explored 2D graph representations using graph CNN (GCNN). Although the resulting unified RNN/GCNN-CNN model did not improve against unified RNN-CNN, graphs are more descriptive for compounds and more developments in graph models are needed to address remaining challenges.

3.3 Explainable Prediction of Compound-Protein Interactions

3.3.1 Introduction

As explained in section 3.2, drug discovery calls for efficient characterization of compound efficacy and toxicity, and computational prediction of compound-protein interactions (CPI) addresses the need. Also, classical physics-driven methods model atomic-level energetics using co-crystallized or docked 3D structures of compound-protein pairs [170, 120], such as molecular mechanical and quantum mechanical force fields, potentials of mean force, and empirical and statistical scoring. Moreover, their affinity predictions are intrinsically interpretable toward revealing mechanistic principles, with the consideration of atomic contacts, dynamics, and energetics as well as solvent effects. Recently, thanks to increasingly abundant molecular data and advanced computing power, data-driven machine learning (especially deep learning) methods are also developed using the input structures of compound-protein complexes [123, 124, 171] or proteins alone (see a related task of classifying binding [172, 173]), albeit with less focus on interpretability. However, these structure-based methods, physics- or data-driven, are limited by the availability of structure data. Indeed, 3D structures are often not available for compound-protein pairs or even proteins alone and their prediction through docking is still a computationally demanding and challenging task.

To overcome the data limitation of structure-based affinity-prediction methods and broaden the applicability to more chemical-proteomic pairs without structures, our focus of the study is structure-free prediction of compound-protein affinities. Recent developments only use identities of compounds (SMILES [174, 19] or graphs[19, 175]) and proteins (amino acid sequences [174, 175] or shorter, predicted structural property sequences[19]) as inputs. Compared to these recent work, our goals are two folds: improved generalizability to “new” molecules unseen in training data as well as improved interpretability to a level that data supports (not yet the level of mechanical principles that can be revealed by physics-driven structure-based methods). In particular, interpretability remains a major gap between the capability of current structure-free machine-learning models and the demand for rational drug discovery. The central question about interpretability is whether and how methods (including machine learning models) could explain *why* they make certain predictions (affinity level for any compound-protein pair in our context). This important topic is rarely addressed in structure-free machine learning models. DeepAffinity [19] has embedded joint attentions over compound-protein component pairs and uses such joint attentions to assess origins of affinities (binding sites) or specificities. Additionally, attention mechanisms have been used for predictions of CPI [169], chemical stability [176] and protein secondary structures [177]. Assessment of interpretability for all these studies was either lacking or limited to a few case studies. We note a recent work proposing *post-hoc* attribution-based test to determine whether a model learns binding mechanisms [178].

We raise reasonable concerns on how much attention mechanisms can reproduce native contacts in compound-protein interactions. Attention mechanisms were originally developed to boost the performance of seq2seq models for neural machine translations [160]. And they have gained popularity for interpreting deep learning models in visual question answering [163], natural language processing [179], and healthcare [180]. However, they were also found to work differently from human attentions in visual question answering [181].

Representing the first effort dedicated to the interpretability of structure-free compound-protein affinity predictors (in particular, deep-learning models), our study is focused on how to define, assess, and enhance interpretability for these methods as follows.

How to define interpretability for affinity prediction. Interpretable machine learning is increasingly becoming a necessity [182] for fields beyond drug discovery. Unlike interpretability in a generic case [182], what interpretability actually means and how it should be evaluated is much less ambiguous for compound-protein affinity prediction. So that explanations conform with scientific knowledge, human understanding, and drug-discovery needs, we define interpretability of affinity prediction as to the ability to explain predicted affinity through underlying atomic interactions (or contacts). Specifically, atomic contacts of various types are known to constitute the physical basis of intermolecular interactions [183], modeled in force fields to estimate interaction energies [120], needed to explain mechanisms of actions for drugs [184, 185], and relied upon to guide structure-activity research in drug discovery [186, 187]. Therefore we use the ability to replicate such corresponding contacts *while* predicting affinities as a vehicle for interpretability. The current definition of interpretability (residue-atom pairs in contact) is primitive compared to mechanistic principles in structure-based classical methods. But it is expected to serve as a vehicle to help fill the mechanistic void in structure-free affinity predictors (especially deep-learning models). We emphasize that simultaneous prediction of affinity and contacts does not necessarily make the affinity predictors intrinsically interpretable unless predicted contacts form the basis for predicted affinities.

How to assess interpretability for affinity prediction. Once interpretability of affinity predictors is defined first through atomic contacts, it can be readily assessed against ground truth known in compound-protein structures, which overcomes the barrier for interpretable machine learning without ground truth [188]. In our study, we have curated a dataset of compound-protein pairs, all of which are labeled with K_d values and some of which with contact details; and we have split them into training, test, compound-unique, protein-unique, and both-unique (or double-unique) sets. We measure the accuracy of contact prediction over various sets using area under the precision-recall curve (AUPRC) which is suitable for binary classification (contacts/non-contacts) with highly imbalanced classes (far fewer contacts than non-contacts). We have performed large-scale assessments of attention mechanisms in various molecular data representations (protein amino-acid sequences and structure-property annotated sequences[19] as well as compound SMILES and graphs) and corresponding neural network architectures (convolutional and recurrent neural networks [CNN and RNN] as well as graph convolutional and isomorphism networks [GCN and GIN]). And we have found that current attention mechanisms inadequate for interpretable affinity prediction, as their AUPRCs were merely slightly more than chance (0.004).

How to enhance interpretability for affinity prediction. We have made three main contributions to enhance interpretability for structure-free deep-learning models.

The first contribution is to incorporate physical constraints into data representations, model architectures, and model training. (1) To respect the sequence nature of protein inputs and to overcome the computational bottlenecks of RNNs, inspired by protein folding principles, we represent protein sequences as hierarchical k-mers and model them with hierarchical attention networks (HANs). (2) To respect the structural contexts of proteins, we predict from protein sequences solvent exposure over residues and contact maps over residue pairs; and we introduce novel structure-aware regularizations for structured sparsity of model attentions.

The second contribution is to supervise attentions with native intermolecular contacts available to training data and to accordingly teach models how to pay attention to pairs of compound atoms and protein residues while making affinity predictions. We have formulated a hierarchical multi-objective optimization problem where contact predictions form the basis for affinity prediction. We utilize contact data available to training compound-protein pairs and design hierarchical training strategies accordingly.

The last contribution is to design intrinsic explainability into the architecture of a deep "relational" network. Inspired by physics, we explicitly model and learn various types of atomic interactions (or "relations") through deep neural networks with joint attentions embedded. This was motivated by relational neural networks first introduced to learn to reason in computer vision [189, 190] and subsequent interaction networks to learn the relations and interactions of complex objects and their dynamics [191, 192]. Moreover, we combine such deep relational modules in a hierarchy to progressively focus attention from putative protein surfaces, binding-site k-mers and residues, to putative residue-atom binding pairs.

The rest of the paper is organized as follows. The aforementioned contributions in defining, measuring, and enhancing interpretable affinity prediction will be detailed in Methods. In Results, we first show over established affinity-benchmark datasets that the original DeepAffinity[19] and its variants (with various molecular representations and neural networks) have comparable or better accuracy in affinity prediction, compared to current non-interpretable structure-free methods. We then describe a dataset newly curated for both affinity and contact prediction. The dataset is designed to be diverse and challenging with the generalizability test in mind. Using this dataset, we incrementally introduce the three contributions to DeepAffinity and compare the resulting DeepAffinity+ (using the first two contributions) and DeepRelations (using all three contributions) to a competing interpretable method. Both methods produce remarkably improved interpretability (now defined as accuracy of contacts predicted by joint attentions) while maintaining accurate and generalizable affinity prediction. Importantly, compared to the competing method and their reduced version without supervising attentions, they show that sufficiently better interpretability (much more accurate contact predictions) can help improve accuracy in affinity prediction. Lastly, we use various focused studies to show the spatial patterns of top-10 predicted contacts, the benefit of these predictions to contact-assisted protein-ligand docking, and the additional utilities of aggregating attentions and decomposing predicted affinities for binding site prediction and QSAR.

3.3.2 Methods

Toward genome-wide prediction of compound-protein interactions (CPI), we assume that proteins are only available in 1D amino-acid sequences, whereas compounds are available in 1D SMILES or 2D chemical graphs. We start the subsection with the curation of a dataset of compound-protein pairs with known pK_d/pK_i values, which is also of known intermolecular contacts. We will introduce the state-of-the-art and our newly-adopted neural networks to predict from such molecular data. These neural networks will be first adopted in our previous framework of DeepAffinity [19] (supervised learning with joint attention) so that the interpretability of attention mechanisms can be systematically assessed in CPI prediction. We will then describe our physics-inspired, intrinsically explainable architecture of deep relational networks where aforementioned neural networks are used as basis models. With carefully designed regularization terms, we will explain multi-stage deep relational networks that increasingly focus attention on putative binding-site k-mers, binding-site residues, and residue-atom interactions, for the prediction and interpretation of compound-protein affinity. We will also explain how the resulting model can be trained strategically.

3.3.2.1 Benchmark Set with Compound-Protein Affinities and Contacts

We have previously curated affinity-labeled compound-protein pairs [19] based on BindingDB [144]. In this study, we used those pK_i/pK_d-labeled data with amino-acid sequence length no more than 1,000 and curated a subset with known complex-protein co-crystal structures. We further merge the data with the refined set of PDBbind (v. 2019) [193], leading to 4,446 pairs between 3,672 compounds and 1,287 proteins.

The compound data are in the format of canonical SMILES as provided in PubChem [194] and the protein data are in the format of FASTA sequences (UniProt canonical). Compound SMILES were also converted to graphs with RDKit [195]. Ionization states of compounds defined in PubChem were validated using the software OpenBabel and the compounds were further sanitized and standardized using "chem.SanitizeMol( )" in the software RDKit. Atomic-level intermolecular contacts (or "relations") were derived from compound-protein co-crystal structures in PDB [99], as ground truth for the interpretablity of affinity prediction. Specifically, we cross-referenced aforementioned compound-protein pairs in PDBsum[196] and used its LigPlot service to collect high-resolution atomic contacts or relations. These direct, first-shell contacts are given in the form of contact types (hydrogen bond or hydrophobic contact), atomic pairs, and atomic distances.

The dataset was randomly split into four folds where fold 1 did not overlap with fold 2 in compounds, did not do so with fold 3 in proteins, and did not do so with fold 4 in either compounds or proteins. Folds 2, 3, and 4 are referred to as new-compound, new-protein, and both-new sets for generalizability tests; and they contain 521, 795 and 205 pairs, respectively. Fold 1 was randomly split into training (2,334) and test (591) sets. The split of the whole dataset is illustrated in Figure 3.7 below. And the similarity profiles between training molecules and those in the test and generalization sets are analyzed in Results later.

Although monomer structures of proteins are often unavailable, their structural features can be predicted from protein sequences alone with reasonable accuracy. We have predicted the secondary structure and solvent accessibility of each residue using the latest SCRATCH [149, 150] and contact maps for residue pairs using RaptorX-contact [197]. These data provide additional structural

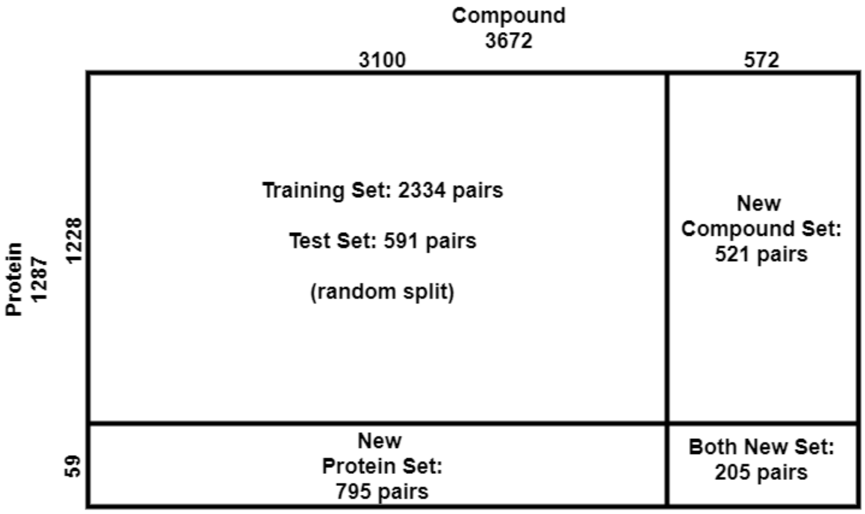


Figure 3.7: The complete data set consists of training, test, compound-unique, protein-unique, and double unique sets with compound-protein counts provided.

information to regularize our machine learning models. If protein structures are available, actual rather than predicted such data can be used instead.

3.3.2.2 Data Representation and Corresponding Basis Neural Networks

Baseline: CNN and RNN for 1D protein and compound sequences.

When molecular data are given in 1D sequences, these inputs are often processed by convolutional neural networks (CNN) [174, 198] and by recurrent neural networks (RNN) that are more suitable for sequence data with long-term interactions [19].

Challenges remain in RNN for compound strings or protein sequences. For compounds in SMILES strings, the descriptive power of such strings can be limited. In this study, we overcome the challenge by representing compounds in chemical formulae (2D graphs) and using two types of graph neural networks (GNN). For proteins in amino-acid sequences, the often-large lengths demand deep RNNs that are hard to be trained effectively (gradient vanishing or exploding and non-parallel training) [199]. We previously overcame the second challenge by predicting structure properties from amino-acid sequences and representing proteins as a much shorter structure property sequences where each 4-letter tuple corresponds to a secondary structure [19]. This treatment however limits the resolution of interpretability to be at the level of protein secondary structures (multiple neighboring residues) rather than individual residues. In this study, we overcome the second challenge while achieving residue-level interpretability by using biologically-motivated hierarchical RNN (HRNN).

Notation summary. Scalars, vectors, and matrices are denoted in normal lower-case, bold-faced lower-case, and upper-case characters, respectively. Subscripts i, t, and j are for the i^{th} protein residue, t^{th} protein k-mer and j^{th} compound atom, respectively. And subscript it represent the i^{th} residue in the t^{th} k-mer (where i can be regarded as a global residue index). Therefore, the j^{th} atom of compound $\mathcal{X}$ described in d_g features is denoted $\mathbf{x}_j$ and its learned representation (embedded through GNN) is denoted $\mathbf{z}_j$. The i^{th} residue of protein $\mathcal{Y}$ with d_p features is denoted by $\mathbf{y}_i$ and its learned representation (embedded through HRNN) is denoted $\mathbf{h}_{it}$ where t is the index of the k-mer containing residue i. These residue representations $\mathbf{h}_{it}$ within the k-mer are then aggregated to obtain the k-mer representation $\mathbf{h}_t$ and all k-mer representations are concatenated to reach the protein representation. Superscript r, (l), and $[s]$ indicate the r^{th} relation about molecular features, the l^{th} layer of graph neural networks, and the s^{th} stage of DeepRelations, respectively.

Proposed: GCN and GIN for 2D compound graphs.

Compared to 1D SMILES strings, chemical formulae (2D graphs) of compounds have more descriptive power and are increasingly used as inputs to predictive models [169, 175, 19, 176, 200]. In this study, compounds are represented as 2D graphs in which vertices are atoms and edges are covalent bonds between atoms. Suppose that n is the maximum number of atoms in our compound set (compounds with smaller number of atoms are padded to reach size n). Let's consider a graph $G = (\mathcal{V}, \mathcal{X}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{v_j\}_{j=1}^{n}$ is the set of n vertices (each with d_g features), $\mathcal{X} \in R^{n \times d_g}$ that of vertex features ($\mathcal{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_j, \cdots, \mathbf{x}_n]$), $\mathcal{E}$ that of edges, and $\mathcal{A} \in \{0,1\}^{n \times n}$ is unweighted symmetric adjacency matrix. Let $\hat{\mathcal{A}} = \mathcal{A} + \mathcal{I}$ and $\hat{\mathcal{D}}$ be the degree matrix (the diagonals of $\hat{\mathcal{A}}$).

We used Graph Convolutional Network (GCN) [201] and Graph Isomorphism Network (GIN) [202] which are the state of the art for graph embedding and inference. GCN consists of multiple layers and at layer l the model can be written as:

$$\mathcal{H}^{(l)} = \mathrm{ReLU}(\hat{\mathcal{D}}^{-\frac{1}{2}}\hat{\mathcal{A}}\hat{\mathcal{D}}^{-\frac{1}{2}}\mathcal{H}^{(l-1)}\Theta^{(l)}), \tag{3.9}$$

where $\mathcal{H}^{(l)} \in R^{n\times d_g^{(l)}}$ is the output, $\Theta^{(l)} \in R^{d_g^{(l-1)}\times d_g^{(l)}}$ the trainable parameters, and $d_g^{(l)}$ the number of features, all at layer l. Initial conditions (when $l = 0$) are $\mathcal{H}^{(0)} = \mathcal{X}$ and $d_g^{(0)} = d_g$.

GIN is the most powerful graph neural network in theory: its discriminative or representational power is equal to that of the Weisfeiler-Lehman graph isomorphism test [203]. Similar to GCN, GIN consists of multiple layers and at layer l the model can be written as a multi-layer perceptron (MLP):

$$\mathcal{H}^{(l)} = \mathrm{MLP}^{(l)}(\bar{\mathcal{A}}^{(l)}\ \mathcal{H}^{(l-1)}), \tag{3.10}$$

where $\bar{\mathcal{A}}^{(l)} = \mathcal{A} + \epsilon^{(l)}\mathcal{I}$, $\epsilon^{(l)}$ can be either a trainable parameter or a fixed hyper-parameter. Each GIN layer has several nonlinear layers compared to GCN layer with just a ReLU per layer, which might improve predictions but suffer in interpretability.

The final representation for a compound is $\mathcal{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_j. \cdots, \mathbf{z}_n] = \mathcal{H}^{(L)}$ if GCN or GIN has L layers. In this study, vertex features are as in [176], with few additional features detailed later in physics-inspired relational modules.

Proposed: HRNN for 1D protein sequences.

We aim to keep the use of RNN that respects the sequence nature of protein data and mitigate the difficulty of training RNN for long sequences. To that end, inspired by the hierarchy of protein structures, we model protein sequences using hierarchical attention networks (HANs). Specifically, during protein folding, sequence segments may fold separately into secondary structures and the secondary structures can then collectively pack into a tertiary structure needed for protein functions. We exploit such hierarchical nature by representing a protein sequence of length easily in thousands as tens or hundreds of k-mers (consecutive sequence segments) of length k (hyper-parameter in this study). Accordingly we process the hierarchical data with hierarchical attention networks (HANs) [204] which have been proposed for natural language processing. We also refer to it as hierarchical RNN (HRNN). Although the inter-k-mer attentions might overcome potential issues brought by k-mer definition as they do in natural language processing [204], it would be interesting to examine the potential benefit of using other domain-relevant definition of k-mers, such as (predicted or actual) secondary structure elements.

Given $\mathcal{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_i, \cdots, \mathbf{y}_m]$, a protein sequence described with d_p features for each residue i ($\mathcal{Y} \in R^{n \times d_p}$), we partition it into T consecutive, non-overlapping k-mers. We use two types of RNNs in hierarchy for modeling within and across k-mers. We first use an embedding layer to represent the i^{th} residue in the t^{th} k-mer as a vector $\mathbf{e}_{it}$. And we use a shared RNN for all k-mers for the latent representation of the residue: $\mathbf{h}_{it} = \mathrm{RNN}(\mathbf{e}_{it})$ $(t = 1, \ldots, T)$. We then summarize each k-mer as $\mathbf{k}_t$ with an intra-k-mer attention mechanism:

$$\begin{aligned}
u_{it} &= \mathbf{v_1}\tanh(\Theta_1 \mathbf{h}_{it} + \mathbf{b_1}) \ \forall \mathrm{i, t} \\
u'_{it} &= \frac{\exp(u_{it})}{\sum_{i'} \exp(u_{i't})} \ \forall \mathrm{i, t} \\
\mathbf{k}_t &= \sum_i u'_{it} \mathbf{h}_{it} \ \forall \mathrm{t}
\end{aligned} \tag{3.11}$$

With another RNN for $\mathbf{k}_t$ we reach the representation of the t^{th} k-mer: $\mathbf{h}_t = \mathrm{RNN}(\mathbf{k_t})$ $(t = 1, \ldots, T)$.

The final representation for a protein sequence is the collection of $\mathbf{h}_t$.

Joint attention over protein-compound atomic pairs for interpretability. Once the learned representation of protein sequences ($\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_t, \ldots, \mathbf{h}_T]$ where t is the index of protein k-mer) and that of compound sequences or graphs ($\mathcal{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_j, \ldots, \mathbf{z}_n]$ where j is the index of compound atom) are defined, they are processed with a joint k-mer–atom attention mechanism to interpret any downstream prediction:

$$\begin{aligned}
N_{tj} &= \tanh(\mathbf{h}_t \Theta_2 \mathbf{z}_j) \ \forall t, j \\
\mathcal{W}'_{tj} &= \frac{\exp(\mathbf{N}_{tj})}{\sum_{t', j'} \exp(\mathbf{N}_{t'j'})} \ \forall t, j
\end{aligned} \tag{3.12}$$

With $\mathcal{W}'_{tj}$, the joint attention between the t^{th} k-mer and the j^{th} atom, we can combine it with the intra-k-mer attention over each residue i in the t^{th} k-mer and reach $\mathcal{W}_{ij}$, the joint attention between the i^{th} protein residue and the j^{th} compound atom:

$$\mathcal{W}_{ij} = u'_{it}\mathcal{W}'_{tj} \ \ \forall\, i, j \tag{3.13}$$

This joint attention mechanism is an extension of our previous work [19] where a protein sequence was represented as a single, "flat" RNN rather than multiple, hierarchical RNNs.

Given learned representations $\mathbf{h}_i$ for protein residue i (the k-mer index is ignored for simplicity) and $\mathbf{z}_j$ for compound atom j as well as the joint attention $\mathcal{W}_{ij}$ over the pair, we further jointly embed the pair and aggregate over all pairs to reach $\mathbf{f}$ — the joint embedding of protein $\mathcal{Y}$, compound $\mathcal{X}$, and their residue-atom "interactions" captured by $\mathcal{W}$:

$$\begin{aligned} \mathbf{f}_{ij} &= \tanh(\Theta_3\mathbf{h}_i + \Theta_4\mathbf{z}_j + \mathbf{b}_2) \\ \mathbf{f} &= \sum_{i,j} \mathbf{f}_{ij}\mathcal{W}_{ij} \end{aligned} \tag{3.14}$$

where Θ_3, Θ_4 and $\mathbf{b}_2$ are learnable parameters. The joint embedding $\mathbf{f}$ is fed to a CNN and two multi-layer perceptrons (MLP) to make affinity prediction as before [19]. In other words, $\mathcal{W}$ for contact prediction directly forms the basis of $\mathbf{f}$ for affinity prediction.

In comparison, Gao et al.'s method [169] also uses joint attention for contact prediction. But the joint attention matrix is marginalized for either the compound or the protein; and the separately processed compound or protein representatons were used for affinity prediction. More specifically,

$$\begin{aligned} \mathcal{W}_{ij} &= \tanh(\mathbf{h}_i\Theta\mathbf{z}_j) \\ \mathbf{u}_i &= \max_j \mathcal{W}_{ij},\ \alpha_i = \frac{\exp(\mathbf{u}_i)}{\sum_{i'}\exp(\mathbf{u}_{i'})} \ \ \forall \mathrm{i} \\ \mathbf{u_j} &= \max_i \mathcal{W}_{ij},\ \alpha_j = \frac{\exp(\mathbf{u}_j)}{\sum_{j'}\exp(\mathbf{u}_{j'})} \ \ \forall \mathrm{i} \\ \mathbf{o}_p &= \sum_i \alpha_i\mathbf{h_i},\ \mathbf{o}_d = \sum_j \alpha_j\mathbf{z_j}. \end{aligned} \tag{3.15}$$

The separate final representations for the compound ($\mathbf{o}_d$) and the protein ($\mathbf{o}_p$) were fed to downstream layers for affinity prediction, with much of information lost on the joint attention (the basis of contact prediction).

3.3.2.3 DeepRelations

Overall architecture. We have developed an end-to-end "by-design" interpretable architecture named DeepRelations for joint prediction and interpretation of compound-protein affinity. The overall architecture is shown in Figure 3.8.

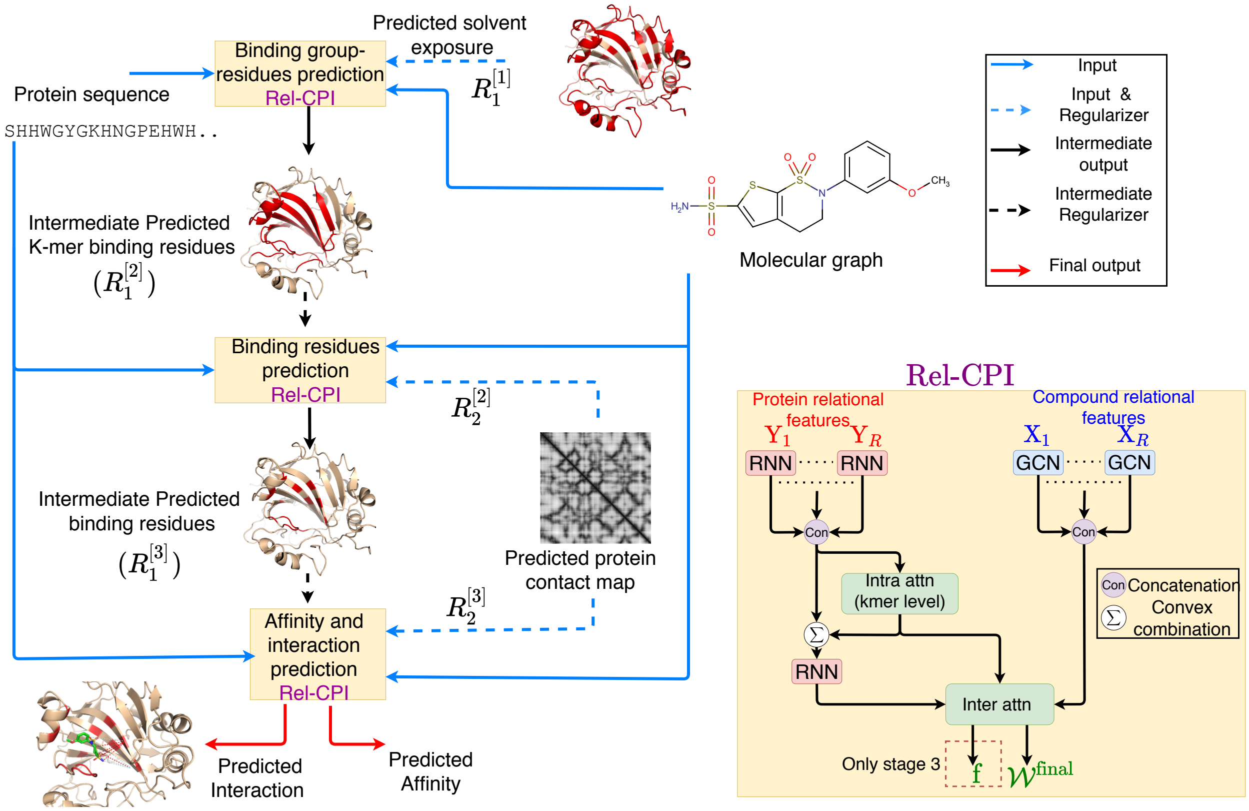


Figure 3.8: Schematic illustration of DeepRelations, an intrinsically explainable neural network architecture for predicting compound-protein interactions. Three linked relational modules (Rel-CPI in the small yellow boxes) correspond to three stages of attention focusing. Each module embeds relational features with joint attentions over pairs of protein residues and compound atoms (details on the right). In comparison, DeepAffinity+ has a single module with all relational features lumped together. Both methods are structure-free and protein structures are just for illustration.

There are three relational modules (Rel-CPI) corresponding to three stages. Their attentions are trained to progressively focus on putative binding k-mers, residues, and pairs; and earlier-stage attentions guide those in the next stage through regularization. In each Rel-CPI module, there are $K = 10$ types of atomic "relational" features for proteins or compounds (9 relation (sub)types are described next and the last is the union of all 9 types of features). All types of relational features are individually fed to aforementioned neural network pairs (for instance, HRNN for protein sequences and GCN for compound graphs, or HRNN-GCN in short), concatenated, and jointly embedded for proteins and compounds with attentions over residue-compound pairs. The embedding output (based on joint attentions for contact prediction) of the last module is fed to CNN and MLP layers for affinity prediction. All three modules are trained end-to-end as a single model. In contrast, DeepAffinity+ only has one module without multi-stage focusing; and its module only uses the last type of relational features (the union of the first 9 types).

Physics-inspired relational modules

The relational modules are inspired by physics. Specifically, atomic "relations" or interactions constitute the physical bases and explanations of compound-protein interaction affinities and are often explicitly modelled in force fields. We have considered the following six types of relations with attentions paid on and additional input data defined for.

- *Electrostatic interactions*: The ion feature of a protein residue is its net charge as in the force field CHARMM36 and that of a compound atom is its formal charge. The dipole feature of a protein residue is 1 for polar residues (S, T, C, Y, N, Q and H [205]) or 0 for others; and that of a compound atom is its Gasteiger partial charge. The electrostatics thus include all four combinations (subtypes) of residue-atom relations: ion-ion, ion-dipole, dipole-ion, and dipole-dipole.

- *Hydrogen bond*: Non-covalent interaction (A$\cdots$H–D) between an electronegative atom as a hydrogen "acceptor" ("A") and a hydrogen atom that is covalently bonded to an electronegative atom called a hydrogen "donor" ("D"). Therefore, if a protein residue or compound atom could provide a hydrogen acceptor/donor, its hydrogen-bond feature is -1/+1; otherwise the feature value is 0. A protein residue is allowed to be both hydrogen-bond donor and acceptor. Specifically, for protein residues, amino acids of hydrogen-bond acceptors are N, D, Q, E, H, S, T, and Y; and those of hydrogen-bond donors are Y, W, T, S, K, H, Q, N, and R [206]. For compound atoms, hydrogen-bond acceptor or donor is defined as in the base features factory file (atom types "SingleAtomAcceptor" and "SingleAtomDonor" in the file "BaseFeatures.fdef") of the software RDKit v. 2018.03.4.

- *Halogen bond*: A halogen bond ($A\cdots X$–D) is very similar to hydrogen bond except that a halogen "X" (rather than hydrogen) atom (often found in drug compounds) is involved in such interactions. As standard amino acids do not contain halogen atoms, a protein residue can only be a halogen bond acceptor ("A" in $A\cdots X$–D) and assigned a nonzero halogen-bond feature of -1, only if it is amino acid S, T, Y, D, E, H, C, M, F, W [207], N or Q. On the compound side, only a halogen atom is assigned a nonzero feature value. Specifically, halogen-bond features of iodine, bromine, chlorine and fluorine atoms are assigned at +4, +3, +2 and +1, respectively, for decreasing halogen-bonding strengths [207].

- *Hydrophobic interactions*: The interactions between hydrophobic protein residues and compound atoms contribute significantly to the binding energy between them. This feature is only nonzero and set at 1 for hydrophobic residues of proteins or non-polar atoms of compounds (atoms whose absolute values of partial atomic charges are less than 0.2 units [208, 209]).

- *Aromatic interactions*: Aromatic rings in histidines, tryptophans, phenylalanines, and tyrosines participate in "stacking" interactions with aromatic moieties of a compound (π-π stacking). Therefore, if a protein residue has an aromatic ring, its aromatic feature is set at 1 and otherwise at 0. Similarly, if a compound atom is part of an aromatic ring, the feature is set at 1 and otherwise at 0.

- *VdW interactions*: Van der Waals are weaker interactions compared to others. But the large amount of these interactions contribute significantly to the overall binding energy between a protein and a compound. We consider the amino-acid type and the atom element as their features and use an embedding layer to derive their continuous representations.

For each (sub)type of atomic relations, corresponding protein and compound features are fed into basis neural network models such as HRNN for protein sequences and GNN for compound graphs. The embeddings over all types are concatenated for protein residues or compound atoms and then jointly embedded with joint attentions over residue-atom pairs.

Physical constraints as attention regularization The joint attention matrices $\mathcal{W}$ in each Rel-CPI module, for individual relations or overall, are regularized with the following two types of physical constraints. We note that, aiming at the general case where protein structures may not be available, we use sequence-predicted rather than actual structure properties (solvent exposure and residue contacts) when introducing these physical constraints.

3.3.2.3.1 Focusing regularization In the first regularization, a constraint input is given as a matrix $\mathcal{T} \in [0,1]^{m\times n}$ to penalize the attention matrix $\mathcal{W}$ if it is focused on undesired regions of proteins. In addition, an L1 sparsity regularization is on the attention matrix $\mathcal{W}$ to promote interpretability as a small portion of protein residues interact with compounds. Therefore, this "focusing" penalty can be formalized as:

$$R_1(\mathcal{W}) = \lambda_{\text{relation}}||(\mathbf{1} - \mathcal{T}) \odot \mathcal{W}||_2 + \lambda_{\text{L1}}||\mathcal{W}||_1, \tag{3.16}$$

where the $\mathcal{T}$ term, a parameter, can be considered as soft thresholding and the matrix norms are element-wise. The L1 regularization term in $R_1(\cdot)$ is only included in the first module (stage 1) where $R_1(\cdot)$ is the only regularization term. It is then moved to another term in the second and the last modules where multiple regularization terms are used together.

The first regularization is used for all three Rel-CPI modules or stages with increasingly focusing $\mathcal{T}$. Let $\mathcal{T}^{[s]}$ be the constraint matrix and $\mathcal{W}^{[s]}$ the learned attention matrix in the s^{th} stage. In the first stage, $\mathcal{T}^{[1]}_{ij}$, being binary, is one only for any residue i predicted to be solvent-exposed (relative solvent-accessible area predicted above 0.25 by SCRATCH[149, 150]) in order to focus on potential surfaces. In the second stage, $\mathcal{T}_{ij}^{[2]} = \max_{j'} \mathcal{W}_{ij'}^{[1]}$ to focus on putative binding residues hierarchically learned for k-mers and residues at module/stage 1. In the third and last stage, $\mathcal{T}_{ij}^{[3]} = \max_{j'} \mathcal{W}_{ij'}^{[2]}$ focuses on putative contacts between protein residues and compound atoms based on the learned binding residues at module/stage 2.

3.3.2.3.2 Structure-aware sparsity regularization over protein contact maps We further develop a structure aware sparsity constraints based on known or RaptorX-predicted contact maps of the unbound protein. As sequentially distant residues might be close in 3D and form binding sites for compounds, we define overlapping groups of residues where each group consists of a residue and its spatially close neighboring residues. Just in the second stage, we introduce Group Lasso for spatial groups and the Fused Sparse Group Lasso (FSGL) for sequential groups on the overall, joint attention matrix $\mathcal{W}$:

$$R_2(\mathcal{W}) = \lambda_{\text{group}}||\mathcal{W}||_{\text{group}} + \lambda_{\text{fused}}||\mathcal{W}||_{\text{fused}} + \lambda_{\text{L1}-\text{overall}}||\mathcal{W}||_1. \tag{3.17}$$

The group Lasso penalty will encourage a structured group-level sparsity so that few clusters of spatially close residues share similar attentions within individual clusters. The fused sparsity will encourage local smoothness of the attention matrix so that sequentially close residues share similar attentions with compound atoms. The L1 term again maintains the sparsity of the attention matrix $\mathcal{W}$. This regularization is only introduced in the second and third stages for $\mathcal{W}^{[2]}$ and $\mathcal{W}^{[3]}$, after the first-stage attention matrix $\mathcal{W}^{[1]}$ is supposedly focused on protein surfaces. The attention matrix in the last stage, $\mathcal{W}^{[3]}$, is used for predicting residue-atom contacts.

Supervised attentions

It has been shown in visual question answering that attention mechanisms in deep learning can differ from human attentions[181]. As will be revealed in our results, they do not necessarily focus on actual atomic contacts (relations) in compound-protein interactions either. We have thus curated a relational subset of our compound-protein pairs with affinities, for which known ground-truth atomic contacts or relations are available. We summarize actual contacts of a pair in a matrix $\mathcal{W}^{\text{native}}$ of length $m \times n$, which is a binary pairwise interaction matrix padded with 0 to reach the maximum number of protein residues or compound atoms and then normalized by the total number of nonzero entries. We have accordingly introduced an additional third regularization term to supervise attention matrix $\mathcal{W}$ in the second and third stages:

$$R_3(\mathcal{W}) = \lambda_{\text{bind}} ||\mathcal{W} - \mathcal{W}^{\text{native}}||_F. \tag{3.18}$$

In the case of DeepAffinity+ with a single module, all three regularization terms are included as in the last module of DeepRelations.

Training strategy for hierarchical multi-objectives Accuracy and interpretability are the two objectives we pursue at the same time. In our case, the two objectives are hierarchical: compound-protein affinity originates from atomic-level interactions (or "relations") and better interpretation in the latter potentially contributes to better prediction of the former.

Challenges remain in solving the hierarchical multi-objective optimization problem. Optimizing for both objectives simultaneously (for instance, through weighted sum of them) does not respect that the two objectives do no perfectly align with each other and are of different sensitivities to model parameters. Therefore, we consider the problem as multi-label machine learning. And we design hierarchical training strategies to solve the corresponding hierarchical multi-objective optimization problem, which is detailed next.

Take DeepAffinity+ as an example. We first "pre-trained" it to minimize mean squared error (MSE) of pK_i/pK_d regression alone, with physical constraints turned on; in other words, attentions were regularized (through $R_1(\cdot)$ and $R_2(\cdot)$) but not supervised in this stage. We tuned combinations of all hyperparameters except λ_{bind} in the discrete set of $\{10^{-4}, 10^{-3}, 10^{-2}\}$, with 200 epochs at the learning rate of 0.001. Over the validation set, we recorded the lowest RMSE for affinity prediction and chose the hyperparameter combination with the highest AUPRC for contact prediction subjective to that the corresponding affinity RMSE (root mean square error) does not deteriorate from the lowest by more than 10%.

With the optimal values of all hyperparameters but λ_{bind} fixed, we then loaded the corresponding optimized model in the first stage and "fine-tuned" the model to minimize MSE additionally regularized by supervised attentions (through $R_1(\cdot)$, $R_2(\cdot)$, and $R_3(\cdot)$). We used the same learning rate (0.001) and training epochs (200) in fine-tuning; and we tuned λ_{bind} in the set of $\{10^0, \ldots, 10^5\}$ following the same strategy as in pre-training.

The tuned hyperparameters for all DeepAffinity+ variants are summarized as following. For HRNN-GCN_cstr (modeling protein sequences with HRNN and compound graphs with GCN, regularized by physical constraints in $R_2(\cdot)$),

we chose $\lambda_{\text{group}} = 10^{-4}$, $\lambda_{\text{fused}} = 10^{-3}$, and $\lambda_{\text{L1-overall}} = 10^{-2}$; and for its supervised version HRNN-GCN_cstr_sup, the additional $\lambda_{\text{bind}} = 10^4$. For HRNN-GIN_cstr (modeling protein sequences with HRNN and compound graph with GIN, regularized by physical constraints in $R_2(\cdot)$), we chose $\lambda_{\text{group}} = 10^{-4}$,$\lambda_{\text{fused}} = 10^{-3}$, and $\lambda_{\text{L1-overall}} = 10^{-4}$; and for its supervised version HRNN-GIN_cstr_sup, the additional $\lambda_{\text{bind}} = 10^3$. $R_1(\cdot)$ was for attentions on individual relations in DeepRelations and not applicable for DeepAffinity+ variants, although a surface-focusing regularization on overall attentions could be introduced.

We did similarly for hyper-parameter tuning for DeepRelations while constraining (and supervising) attentions. The whole DeepRelations model, including the three Rel-CPI modules, is trained end-to-end [143]. To save computational resources, we used the same hyperparameters in $R_2(\cdot)$ ($\lambda_{\text{L1-overall}}$, λ_{fused}, and λ_{group}) as those optimally tuned in HRNN-GCN_cstr_sup. We then tuned the rest of the hyper-parameters (λ_{L1}, $\lambda_{\text{relation}}$, and$\lambda_{\text{bind}}$) following the aforementioned process of pre-training and fine-tuning. In the end, we chose $\lambda_{\text{relation}} = 10^{-4}$, $\lambda_{\text{L1}} = 10^{-5}$, $\lambda_{\text{group}} = 10^{-4}$, $\lambda_{\text{fused}} = 10^{-3}$, $\lambda_{\text{L1-overall}} = 10^{-2}$ and $\lambda_{\text{bind}} = 10^3$ for DeepRelations. λ_{bind} is usually larger because it is multiplied to the attention-supervision term that can be orders of magnitude smaller than other terms.

3.3.3 Results

We first assess the accuracy of compound-protein affinity predictions made by state-of-the-art non-interpretable methods and our interpretable DeepAffinity framework [19] (with new variants), using three established benchmark sets. After establishing that DeepAffinity achieves the state of the art in the accuracy of affinity prediction, we then describe a newly-curated dataset with both affinities and contacts of compound-protein interactions and assess the interpretability of various DeepAffinity versions and a competing interpretable method adapted to affinity prediction. We find that current attention-based interpretable models are not adequate for interpreting affinity (i.e., predicting contacts). Thus we proceed to regularize and supervise attentions in DeepAffinity to make DeepAffinity+ models. And we additionally use a novel, physics-inspired and intrinsically-interpretable deep relational architecture to make DeepRelations models.

Over the curated dataset, we compare our methods with a competing, structure-free interpretable method in accuracy, generalizability, and interpretability. Using a series of case studies, we also analyze the accuracy levels and spatial patterns of their top-predicted contacts, which are shown to benefit protein-ligand docking. We end the subsection by introducing analytics to aggregate joint attentions and decompose predicted affinity; and by demonstrating their potential utilities toward binding site prediction for proteins and SAR for compounds (scoring and lead optimization).

3.3.3.1 DeepAffinity with interpretable attentions achieves the state-of-the-art accuracy in compound-protein affinity prediction.

As the starting point of interpretability assessment and improvement, our previous interpretable DeepAffinity framework [19] is first compared to current methods based on prediction accuracy for established benchmark sets.

For affinity benchmark datasets, we adopt three established ones of increasing difficulty, the Davis[210], the Kinase Inhibitor BioActivity (KIBA)[211] and the refined set of PDBbind (v. 2019) [193]. We filtered and partitioned the first two datasets consistently with earlier studies [212, 211, 174, 213]. The Davis dataset [212] contains all 30,056 K_d-labeled pairs between 68 kinase inhibitors (including FDA-approved drugs) and 442 kinases, randomly split into 25,046 for training and 5,010 for testing (the widely-used "S1" setting [212]). The filtered KIBA dataset [211, 212] contains 118,254 pairs between 2,111 kinase inhibitors and 229 kinases, including

98,545 for training and 19,709 for testing (S1 split again). Other split settings were not pursued because published performances in such settings are not always available and comparable. The KIBA scores combine k_i, k_d, and IC$_{50}$ sources for consistency and are further processed [212, 174]. As to the refined PDBbind dataset (v. 2019), we filtered and processed it to reach 3,505 pairs with k_i or k_d labeled between 1,149 proteins and 2,870 compounds. Compared to Davis and KIBA, the PDBbind dataset contains more diverse protein classes: 2,157 interactions with enzymes including 72 with kinases, 62 with nuclear receptors, 33 with G protein-coupled receptors (GPCRs), and 106 with ion channels. The portion of labeled compound-protein pairs is much lower than that of Davis and KIBA. We randomly split the PDBbind dataset into 2,921 pairs for training and 584 for testing.

For our framework of DeepAffinity[19], we adopt various data representations and corresponding state-of-the-art neural network architectures as detailed in Methods. To model proteins, we have adopted RNN using protein SPS [19] as input data as well as CNN and newly developed HRNN using protein amino-acid sequences. To model compounds, we have adopted RNN using SMILES as input data as well as GCN and GIN using compound graphs with node features and edge adjacency [176]. In the end, we have tested five DeepAffinity variants (including four new) for protein-compound pairs, including RNN-RNN[19], RNN-GCN, CNN-GCN, HRNN-GCN, and HRNN-GIN. Names before and after hyphens indicate models to embed proteins and compounds, respectively; and embeddings of a pair of protein and compound are passed through joint attentions in Eq 3.14 before being fed to a convolutional neural network (CNN) and multi-layer perceptrons (MLP)[19]. For instance, the first one, RNN-RNN indicates that protein SPS sequences are modeled by RNN and compound SMILES or graphs are modeled by RNN. This is essentially our previous method [19] except that no unsupervised pretraining or ensemble averaging is used here. We have tuned hyper-parameters for DeepAffinity variants including learning rate ($\{10^{-3}$,$10^{-4}\}$), batch size ({64, 128} (16 for CNN-GCN due to the limit of GPU memory) and dropout rate ({0.1, 0.2}) using random 10% of training data as validation sets. When HRNN was used to model protein sequences, we have also tuned k-mer lengths and group sizes in pairs ({(40,30), (48,25), (30,40), (25,48), (15,80), (80,15)} for Davis and {(40,25), (50,20), (25,40), (20,50)} for KIBA and

PDBbind) using the validation sets.

For comparison, we use published current methods that are not structure-based, including DeepDTA[174], KronRLS[214], and WideDTA[215], all of which are non-interpretable. Their results for the Davis and KIBA sets were self reported in individual studies and summarized in a comparison study[213]. And their results for the PDBbind set are derived by re-training released source codes with published hyper-parameter grids and individual training sets (except wideDTA whose codes are not available). In addition, we compare to structure-free methods that are interpretable. Except DeepAffinity, the only other interpretable method published so far (Gao et al.) was for predicting binary compound-protein interaction [169]. As its codes are not publicly available, we have implemented the method, revised its model's last layer (sigmoid) and retrained the model for affinity prediction using each training set. To ensure fair comparison, all deep-learning models including our DeepAffinity variants here are trained for 100 epochs or until convergence (the validation loss does not improve within 15 epochs), as competing methods previously did [213].

We compare aforementioned competing methods and DeepAffinity variants in accuracy using two assessment metrics: RMSE (root mean squared error; see Table 3.18) and CI (concordance index; see Table 3.19). Whereas RMSE evaluates the proximity between predictions are to corresponding native values, CI [216], often used for virtual screening, measures the probability of correctly ordering non-equal pairs. We summarize the results in Tables 3.18 and 3.19.

RMSE	DeepDTA	KronRLS	WideDTA	Gao et al.[b]	DeepAffinity				
					RNN-RNN[19]	RNN-GCN	CNN-GCN	HRNN-GCN	HRNN-GIN
Davis	0.5109[a]	0.6080[a]	0.5119[a]	0.7864	0.5032	0.5095	0.8106	**0.5019**	0.6604
KIBA	0.4405[a]	0.6200[a]	**0.4230**[a]	0.7368	0.4335	0.5367	0.8244	0.4480	0.6669
PDBbind	2.0631	1.8005	-	1.8071	1.4524	**1.4277**	1.5580	1.4743	1.4858

[a] Self-reported and published results as summarized in Thafar et al. [213].

[b] Originally a binary classifier, it was implemented and revised by us for affinity prediction.

Table 3.18: Comparing current methods (non-interpretable except Gao et al.) and interpretable DeepAffinity variants in prediction accuracy (measured by RMSE, the lower the better) for the Davis, KIBA and PDBbind benchmark sets. The best performance in each dataset is bold-faced.

CI	DeepDTA	KronRLS	WideDTA	Gao et al.[b]	DeepAffinity				
					RNN-RNN[19]	RNN-GCN	CNN-GCN	HRNN-GCN	HRNN-GIN
Davis	0.8780[a]	0.8830[a]	0.8860[a]	0.7824	**0.9000**	0.8808	0.7373	0.8814	0.8224
KIBA	0.8630[a]	0.7820[a]	**0.8750**[a]	0.7335	0.8423	0.7968	0.5761	0.8420	0.6893
PDBbind	0.7125	0.7197	-	0.7610	**0.8042**	0.7543	0.7119	0.7544	0.7398

[a] Self-reported and published results as summarized in Thafar et al. [213].

[b] Originally a binary classifier, it was implemented and revised by us for affinity prediction.

Table 3.19: Comparing current methods (non-interpretable except Gao et al.) and interpretable DeepAffinity variants in prediction accuracy (measured by concordance index or CI, the larger the better) for the Davis, KIBA and PDBbind benchmark sets. The best performance in each dataset is bold-faced.

From both tables we conclude that the original DeepAffinity method [19] (RNN-RNN; RNN for protein SPS and RNN for compound SMILES) and its variants compared favorably to the state of the art. Specifically, the DeepAffinity variants achieved the best performances in RMSE and CI for both the Davis dataset and the most diverse and sparse dataset of PDBbind. And it closely followed the best performances (WideDTA) for the KIBA dataset.

In particular, the newly introduced HRNN models for protein sequences (higher-resolution than SPS) and graph models GCN & GIN for compound graphs achieved the best or close-to-the-best performances, which enables interpreting affinity prediction at the level of protein residues and compound atoms without sacrificing the accuracy. Considering that other methods are not interpretable and the only exception Gao et al. did not perform as well, the performancs of interpretable DeepAffinity variants are particularly impressive.

3.3.3.2 *Our new dataset for both affinity and contact prediction is diverse and challenging.*

To support systematic assessment and development of explainable affinity prediction, we have constructed a dataset of 4,446 compound-protein pairs (between 1,287 proteins and 3,672 compounds) with both affinity values (pK_i or pK_d) and atomic contacts (available in co-crystal structures).

The dataset contains diverse proteins and compounds. Among the 4,446 pairs, there are 2,913 interactions with enzymes including 114 with kinases, 105 with nuclear receptors, 89 with GPCRs, and 111 with ion channels. The enzymes are across all seven enzyme commission classes. The

3,672 compounds cover wide ranges of physicochemical properties (logP, molecular weight, and affinity values) as seen in Figure 3.9.

The dataset is split into training including validation (2,334), test (591), new-protein (795), new-compound (521), and both-new sets (205), as illustrated in Figure 3.7. Compared to the test set, the three generalization sets not only contain new proteins or/and compounds but also mainly consist of very dissimilar proteins or/and compounds compared to the training set, which suggest their challenges for machine learning. For instance, the new-protein set only contains proteins not present in the training set; and 454 (57.1%) pairs in the set involve new proteins whose sequence identities to the closest training proteins are below 30%. Similarly, 414 (79.5%) new-compound pairs involve new compounds whose Tanimoto scores to the closest training compounds are below 0.5. The both-new set only contains pairs of new proteins and new compounds with similarly low resemblance to the training set. 98 (47.8%) pairs involve new proteins with sequence identity below 30% *and* new compounds with Tanimoto scores below 0.5. So the both-new set is expected to be the most challenging set among the four for the generalizability of machine learning models. Pair breakdowns are visualized in part of Figure 3.13 (counts). In addition, Jensen-Shannon distances between compound properties of training and those of the other sets are given in the Table 3.20, similarly revealing the most challenging both-new set.

Training set compare to	Test	New Protein	New Compound	Both New
LogP	0.1707	0.1781	0.2350	0.3077
Exact_MW	0.1350	0.2304	0.2146	0.2906
Label (Affinity)	0.1721	0.2118	0.1581	0.3065

Table 3.20: Jensen-Shannon distances between the training and the other sets in various property distributions.

3.3.3.3 Attentions alone are inadequate for interpreting compound-protein affinity prediction.

Now that we have established the accuracy of attention-embedded DeepAffinity and constructed a suitable dataset, our first task for interpretability is to systematically assess the adequacy

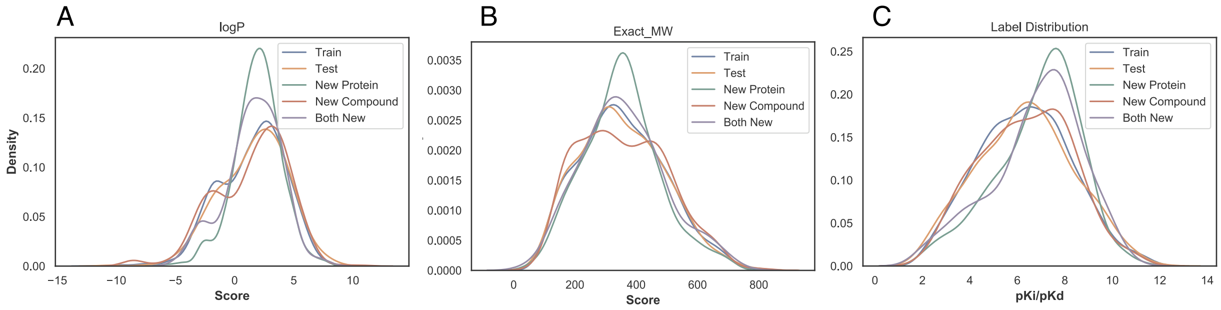


Figure 3.9: The distributions of compound properties across various subsets: A. logP; B. exact molecule weight; and C. pK_i/pK_d labels.

of attention mechanisms for interpreting model-predicted compound-protein affinities. To that end, using our newly curated benchmark set for both affinity and contact prediction, we have tested six DeepAffinity variants for protein-compound pairs (including RNN-RNN, RNN-GCN, CNN-GCN, HRNN-RNN, HRNN-GCN, and HRNN-GIN) as well as the only other interpretable method (Gao et al.) that is also attention-based and adapted by us from a classifier to a regressor. All models are re-trained using the new training set with details in Methods.

The first two DeepAffinity (RNN-RNN and RNN-GCN) models' attentions on proteins are at the secondary structure levels. Their joint attentions were thus converted to residue-atom matrices, using equal weights across all residues within a secondary structure, in the post-analysis of interpretability. The rest have joint attentions at the level of pairs of protein residues and compound atoms.

The accuracy of affinity prediction, measured by RMSE in pK_i/pK_d, is summarized for the DeepAffinity variants in the top panel of Figure 3.10. Overall, all variants have shown affinity error around 1.5, 1.6, 1.4, and 1.7 for the default test, new-protein, new-compound, and both-new sets, respectively. In particular, the HRNN-GCN version achieved an RMSE of 1.47, 1.46, 1.34, and 1.49 for the four sets, respectively, showing a robust accuracy profile. In contrast, the competing method (Gao et al.) has worse RMSE values between 1.72 and 1.87.

The interpretability of affinity prediction is assessed against ground truth of contacts, as in the bottom panel of Figure 3.10. Specifically, we use joint attention scores to classify all pos-

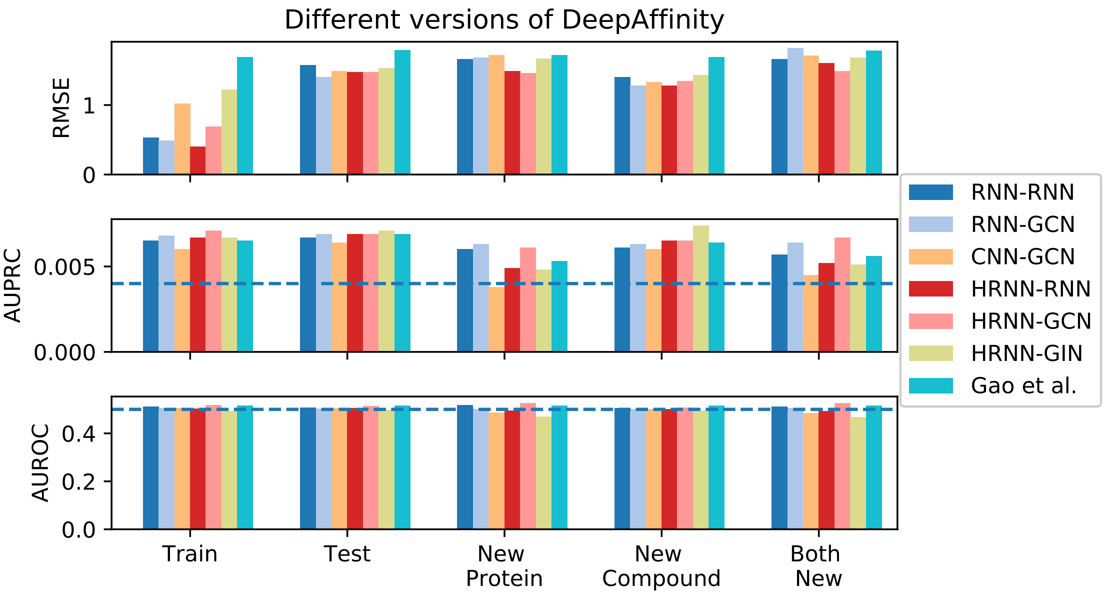


Figure 3.10: Comparing accuracy and interpretability among various versions of DeepAffinity with unsupervised joint attention mechanisms as well as another interpretable method (Gao et al.). Separated by hyphens in legends are neural network models for proteins and compounds respectively. A horizontal dashed line indicates the performances of a random predictor

.

sible residue-atom pairs into contacts or non-contacts. As contacts only represent a tiny portion 0.0040±0.0029 in our dataset) of all possible pairs, we use the the area under the precision-recall curve (AUPRC) as the major metric and the area under the receiver operating characteristic curve (AUROC) as a reference, to assess such binary classification. Here AUPRC/AUROC is averaged over all pairs involved in the corresponding set. Interestingly, compared to chance (AUPRC=0.004 and AUROC=0.5), all attention-based models including DeepAffinity variants and Gao et al. only had slightly better AUPRC (around 0.006 albeit a 50% improvement) except CNN-GCN for the new-protein set. The best DeepAffinity variant, HRNN-GCN, did improve against Gao et al..

From the results above, we conclude that attention mechanisms alone are inadequate for the interpretability of compound-protein affinity predictors, regardless of the choice of commonly used, generic neural network architectures.

3.3.3.4 Regularizing attentions with physical constraints modestly improves interpretability.

Our next task is to enhance the interpretability of compound-protein affinity prediction beyond the level achieved by attention mechanisms alone. The first idea is to incorporate domain-specific physical constraints into model training. The rationale is that, by bringing in the (predicted) structural contexts of proteins and protein-compound interactions, attentions can be guided in their sparsity patterns accordingly for better interpretability.

We start with the two best-performing DeepAffinity variants so far (HRNN-GCN and HRNN-GIN) where protein amino-acid sequences are modeled by hierarchical RNN and compound graphs by various GNNs (including GCN and GIN). And we introduce structure-aware sparsity regularization $R_2(\cdot)$ to the two models to make "DeepAffinity+" variants. The resulting HRNN-GCN_cstr and HRNN-GIN_cstr models with physical constraints are assessed in Figure 3.11. Compared to the the non-regularized counterparts in Figure 3.10, both models achieved similar accuracy levels across various test sets for affinity prediction. As to their interpretability, HRNN-GCN_cstr had similar AUPRC as before regularization (0.006) and HRNN-GIN_cstr slightly improved AUPRC to around 0.008, although both were still close to the baseline (0.004).

These results suggest that incorporating physical constraints to structurally regularize the sparsity of attentions is useful for improving interpretability but may not be enough.

3.3.3.5 Supervising attentions significantly improves interpretability.

As regularizing attentions with physical constraints was not enough to enhance interpretability, our next idea is to additionally supervise attentions with ground-truth contact data available to some but not all training examples. Again we introduce "DeepAffinity+" models starting with HRNN-GCN and HRNN-GIN, by both regularizing and supervising attentions (using $R_2(\cdot)$ and $R_3(\cdot)$).

The performances of resulting HRNN-GCN_cstr_sup and HRNN-GIN_cstr_sup models are shown in Figure 3.11. Importantly, HRNN-GCN_cstr_sup (light blue) significantly improved interpretability of affinity prediction without the sacrifice of accuracy. The average AUPRC im-

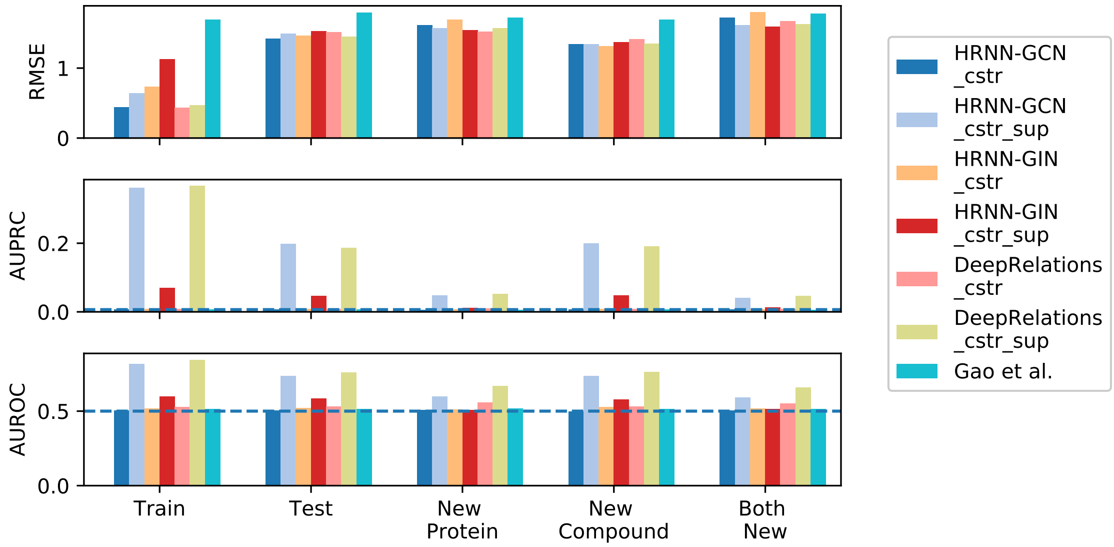


Figure 3.11: Comparing accuracy and interpretability among various versions of DeepAffinity+ (DeepAffinity with regularized and supervised attentions) and DeepRelations. "cstr" in legends indicates physical constraints imposed on attentions through regularization term $R_2(\cdot)$, whereas "sup" indicates supervised attentions through regularization term $R_3(\cdot)$. Horizontal dashed lines suggest the performances of a random contact predictor.

proved to 0.197, 0.048, 0.200, and 0.041 for the default test, new-protein, new-compound, and both-new sets, representing a 30.4, 9.2, 31.2, and 6.3-fold increase, respectively, compared to the version with just regularization but not supervision of attentions (HRNN-GCN_cstr). The performances also represented a 32.9, 9.9, 35.1, and 8.6-fold increase, respectively, compared to Gao et al.. Interestingly, supervising attentions in HRNN-GIN did not see as significant improvement in interpretability.

3.3.3.6 Building explainability into DeepRelations architecture further improves interpretability.

Toward better interpretability, besides regularizing and supervising attentions, we have further developed an explainable, deep relational neural network named DeepRelations. Here atomic "relations" constituting physical bases and explanations of compound-protein affinities are explicitly modeled in the architecture with multi-stage gradual "zoom-in" to focus attentions. In other words, the model architecture itself is intrinsically explainable by design.

The performances of the resulting DeepRelations (with both regularized and supervised attentions) are shown in Figure 3.11 (yellow-green "DeepRelations_cstr_sup"). With equally competitive accuracy in affinity prediction as all previous models, DeepRelations achieved further improvements in interpretability. The AUPRC values were similar to the best DeepAffinity+ model (HRNN-GCN_cstr_sup): 0.187, 0.052, 0.191, and 0.047 for the default test, new-protein, new-compound, and both-new sets, respectively. The AUROC values improved to 0.76, 0.67, 0.76, and 0.66 for the four sets, representing an increase of 0.03, 0.07, 0.03, and 0.07 compared to those of the best DeepAffinity+, respectively.

To disentangle various components of DeepRelations and understand their relative contributions to DeepRelations' improved interpretability, we removed components from DeepRelations for ablation study. Besides regularized and supervised attentions, we believe that the main contributions in the architecture itself are (1) the multi-stage "zoom-in" mechanisms that progressively focus attentions from surface, binding k-mers, binding residues to binding residue-atom pairs; and (2) the explicit modeling of atomic relations that can explain the structure feature-affinity mappings consistently with physics principles.

We thus made three DeepRelations- variants: DeepRelations without multi-stage focusing, without explicit atomic relations, or without both. And we compare them with DeepRelations in the Figure 3.12. Consistent with our conjecture, we found that, the explicit modeling of atomic relations was the main contributor as its removal led to worse affinity and contact predictions in new-protein and both-new sets. The multi-stage focusing also contributes as its removal led to worse affinity prediction for both new-compound and both-new sets.

3.3.3.7 Better interpretability helps better accuracy and generalizability of affinity prediction

To examine whether the more interpretable affinity predictors are also more accurate in affinity prediction, we compare our two final models HRNN-GCN_cstr_sup (DeepAffinity+ hereinafter) and DeepRelations_cstr_sup (DeepRelations hereinafter) to the competing interpretable affinity predictor Gao et al. Re-examining earlier results (Figure 3.11) shows that DeepAffinity+ and DeepRelations with much better interpretability (AUPRC increase between 8.6 and 59-fold) than Gao et al. are also more accurate in affinity prediction (RMSE drop between 0.15 and 0.42) over

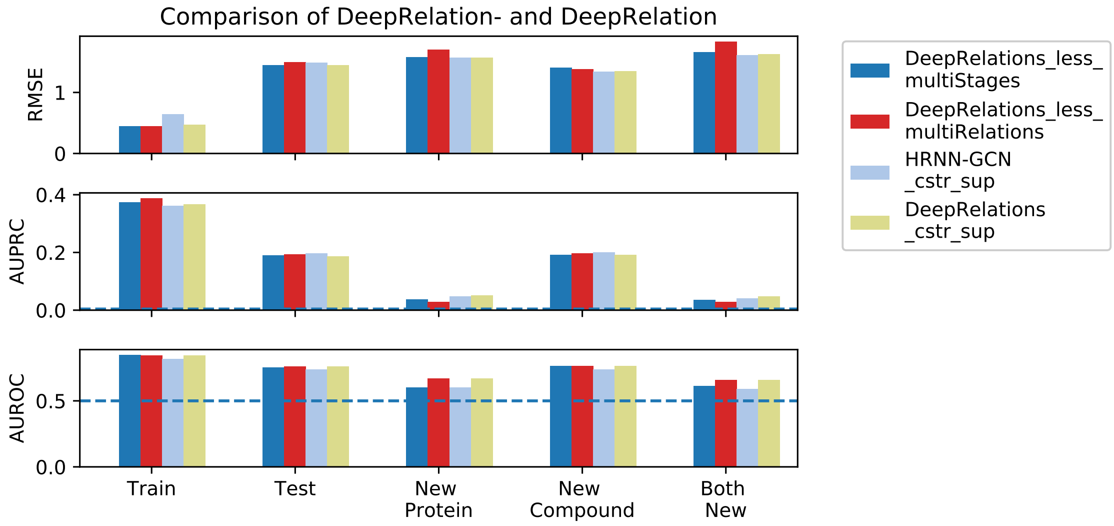


Figure 3.12: Comparing interpretability between DeepRelations and DeepRelations- (DeepRelations without multi-stage focusing, explicitly-modeled relations, or both).

all sets considered. Even when we compare DeepAffiity+ and DeepRelations to their attention-unsupervised counterparts (HRNN-GCN_cstr and DeepRelations_cstr), we find that better interpretability (contact prediction) leading to better accuracy (affinity prediction) in 6 of 8 cases where the only exceptions occurred when AUPRC values were low.

Here we further compare DeepAffinity+ and DeepRelations to Gao et al. in affinity and contact prediction over multiple difficulty ranges (measured by protein sequence identity or compound Tanimoto scores) of the new-compound, new-protein, and both-new sets. The results are reported in Figure 3.13. We find that the same conclusion (better interpretability leads to better accuracy) also applies where model generalizability is needed the most: pairs involving very dissimilar proteins (sequence identity below 30%) or/and compounds (Tanimoto scores below 0.5) compared to training molecules. Importantly, in those cases demanding generalizability the most, DeepAffinity+ and DeepRelations have much better accuracy (affinity-prediction RMSE decrease between 0.14 and 0.40) as well as significantly improved interpretability (contact-prediction AUPRC increase between 5.9 and 33.3-fold) compared to Gao et al..

DeepAffinity+ and DeepRelations also showed competitive generalizability in both affinity and contact prediction. From the most similar proteins (sequence identity above 60%) to the least (sequence identity below 30%), affinity-prediction RMSE values of DeepAffinity+ (DeepRelations) only increased 0.13 (0.08) for the new-compound set and increased 0.00 (0.16) for the most challenging both-new set. From the most similar compounds (Tanimoto scores above 0.8) to the least (Tanimoto scores below 0.5), affinity-prediction RMSE values of DeepAffinity+ (DeepRelations) only increased 0.14 (0.08) for the new-compound set and increased 0.43 (0.48) for the most challenging both-new set. Similar conclusions can be made about their generalizability in contact prediction.

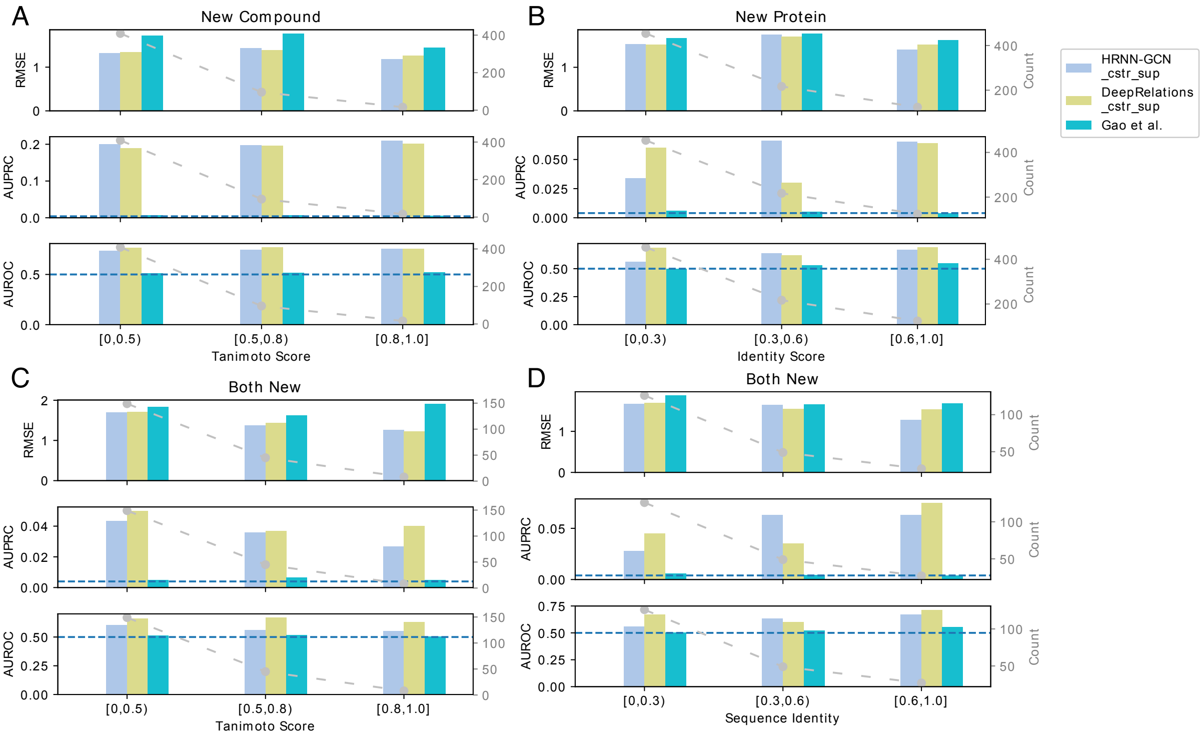


Figure 3.13: Comparing DeepAffinity+, DeepRelations, and Gao's method in the generalizability of affinity prediction (RMSE) and contact prediction (AUPRC and AUROC) to molecules unlike training data.

3.3.3.8 Case studies

Now that we have established and explained how DeepAffinity+ and DeepRelations significantly improve the interpretability of compound-protein affinity prediction, we went on to delve into their affinity and contact predictions in comparison to Gao et al. using a series of cases studies of increasing difficulty. Summary performances of the five cases are reported in Table 3.21. DeepAffinity+ and DeepRelations had better affinity and contact prediction in all cases compared to the competing method whose top-10 predicted contacts failed to produce any native contacts. In order to understand model behaviors, our analysis next would focus on the patterns of top-10 contacts predicted by DeepAffinity+ and DeepRelations compared to Gao et al..

		DeepAffinity+				DeepRelations				Gao et al.			
Protein	Ligand	Affinity Error	Contact AUROC	Contact AUPRC	Top-10 Contact Precision	Affinity Error	Contact AUROC	Contact AUPRC	Top-10 Contact Precision	Affinity Error	Contact AUROC	Contact AUPRC	Top-10 Contact Precision
Two compounds bind to the same pocket of a new, non-homologous protein (different affinity-prediction quality)													
CA2	AL1	1.89	0.658	0.284	0.5	2.70	0.828	0.075	0.6	3.28	0.500	0.006	0.0
	IT2	2.92	0.601	0.034	0.3	3.03	0.780	0.309	0.5	3.09	0.630	0.009	0.0
Two new compounds bind to distinct pockets of a protein													
PYGM	CPB	0.10	0.552	0.006	0.1	0.39	0.513	0.005	0.0	0.61	0.522	0.001	0.0
	T68	0.68	0.944	0.675	1.0	0.66	0.908	0.610	1.0	1.80	0.635	0.006	0.0
A new compound very dissimilar to training compounds binds to a new protein non-homologous to training proteins													
LCK	LHL	2.12	0.500	0.053	0.4	1.30	0.702	0.053	0.4	2.89	0.540	0.005	0.0

Table 3.21: Performance summary of three interpretable methods for five case studies.

Two compounds bind to the same pocket of a new protein non-homologous to training examples

Our first case study involves a protein from the new-protein set, human carbonic anhydrase II (CA2, UniProt ID: P00918), that has no close homolog in the training set. Specifically, the closest training protein would be human carbonic anhydrase IV (CA4, UniProt ID: P22748) with a sequence identity below the 30% threshold (29%). We choose two compounds (HET IDs: AL1 and IT2) that bind to the same pocket of CA2 with distinct sizes (AL1 is larger by 14 heavy atoms) and affinity-prediction quality (see Table 3.21).

We compare in Figure 3.14 the top-10 contacts between protein residues and compound atoms that are predicted by three methods. Top-predicted contacts by Gao et al. were scattered across protein residues that are far from the binding site, failing to match any native contact. In contrast, those top-10 contacts predicted by DeepAffinity+ and DeepRelations were more focused in or near the binding site, containing 3 to 6 native contacts that are direct, first-shell contacts. Between our two models, DeepRelations showed better contact prediction in these two cases: its top-10 predictions were more focused in the binding site and contained 60% and 50% native contacts for compounds AL1 and IT2, respectively. The more focused contact prediction of our methods could be attributed to structure-aware regularization using protein residue-residue contact maps. DeepRelations had better focus than DeepAffinity+, possibly due to the multi-stage focusing strategy.

Even the incorrect predictions of DeepRelations can correspond to residue-atom pairs that are close (but above the 4 Å-cutoff used in the first-shell contact definition). For instance, in the case of compound AL1, the 4 incorrect predictions all corresponded to correct binding-residues that were paired to wrong compound atoms. In the case of compound IT2, the 5 incorrect predictions included 2 that paired correct binding-site residues to wrong atoms and 3 that included (the very next) sequence neighbors of correct binding-site residues.

These two cases also provided examples to interpret the values of AURPC and top-10 contact precision. A seemingly "low" AUPRC value of 0.075 can lead to 5 of 10 top predictions being correct. The reason is that native contacts represent a rare minority (0.004) among all possible residue-atom pairs and an AUPRC value of 0.075 actually represents over 18-fold increase compared to the baseline AUPRC by chance. Meanwhile, a top-10 contact precision of 0.4 predicted by our structure-free methods is close to the average level (0.44) achieved by a popular structure-based protein-ligand docking program, AutoDock Vina [217], under default settings. [218]

Two new compounds bind to distinct pockets of a protein

Our next case study involves two compound-protein pairs from the new-compound set, where two compounds (HET ID: CPB and T68) not present in the training set bind to two distinct pockets of the rabbit glycogen phosphorylase (PYGM, UniProt ID: P00489). The protein is present in the training set with 38 ligands (all but one are occupying the same pocket as T68). In addition, the compound CPB does not resemble its closest training example interacting with the same protein

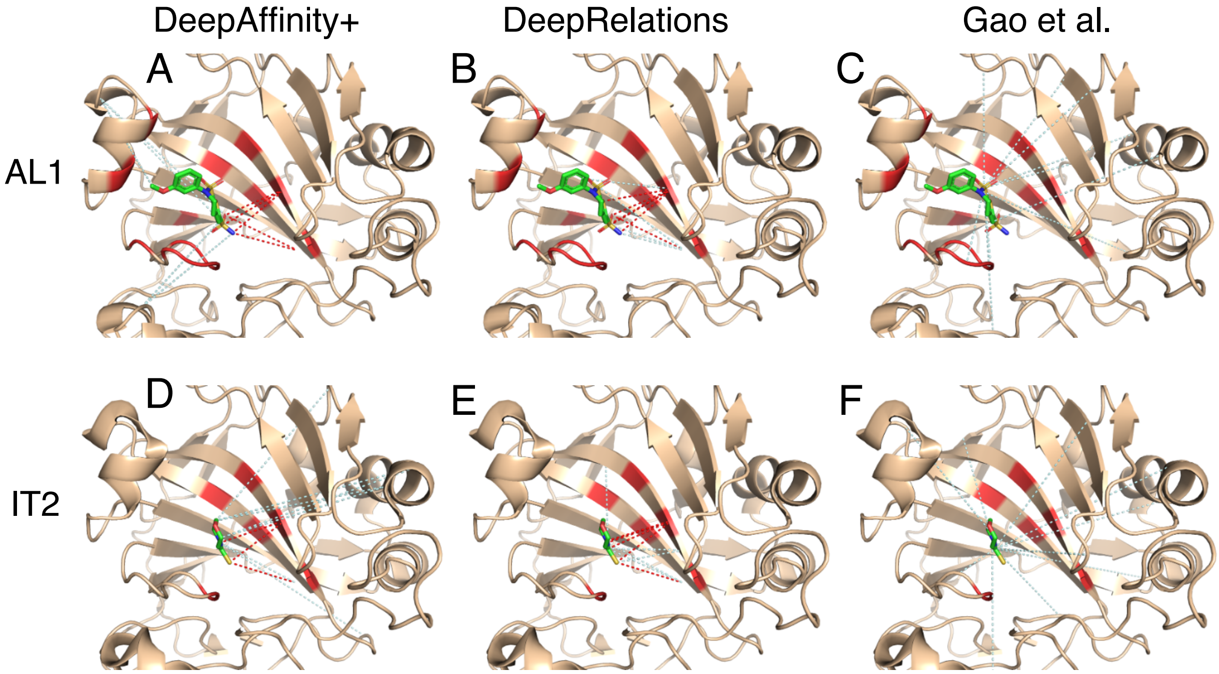


Figure 3.14: Structural visualization of top-10 intermolecular contacts predicted by DeepAffinity+ (left), DeepRelations (middle) and Gao et al. (right) for two test cases. Here two compounds (AL1: top panels and IT2: bottom panels; stick representations) bind to the same pocket of the human carbonic anhydrase II that is new and non-homologous to training data (wheat cartoons where binding residues are highlighted in red). Shown in dashed lines are top-10 predicted contacts (interactions between protein residues and compound atoms). The dashed lines in red and pale cyan highlight correct and incorrect predictions, respectively, according to native, direct contacts retrieved by LigPlot.

(HET ID: 62N) when 62N rather occupies the same pocket as T68. Therefore, contact prediction for the CPB case would be much more challenging. Indeed, our results supported the conjecture (Tabe 3.21). In their top-10 contact predictions our both models achieved 100% native contacts for T68 but just 10% (DeepAffinity+) or even 0% (DeepRelations) for CPB. They had good estimation of binding affinity for both cases.

A closer look into their contact predictions reveal more insights. As seen in Figure 3.15, consistent with our earlier observations, Gao et al.'s contact predictions are dispersed across the whole protein whereas ours are focused. In the case of T68, our predictions are focused in the correct binding site (and even the correct binding residues). However, in the case of CPB, our predictions are actually still focused in the same site as they did for T68, only being wrong this time. Interestingly many falsely-predicted contacts for CPB were not only in the other binding site (circled area) but also with the T68 binding residues. This model behavior is understandable when almost all training examples, including a very similar compound, are indicating a different site. It also reveals a situation that would challenge more generalizability and demand more explainability from machine learning methods. Intriguingly, DeepAffinity+ still managed to make one correct contact prediction (pointed at by a red arrow).

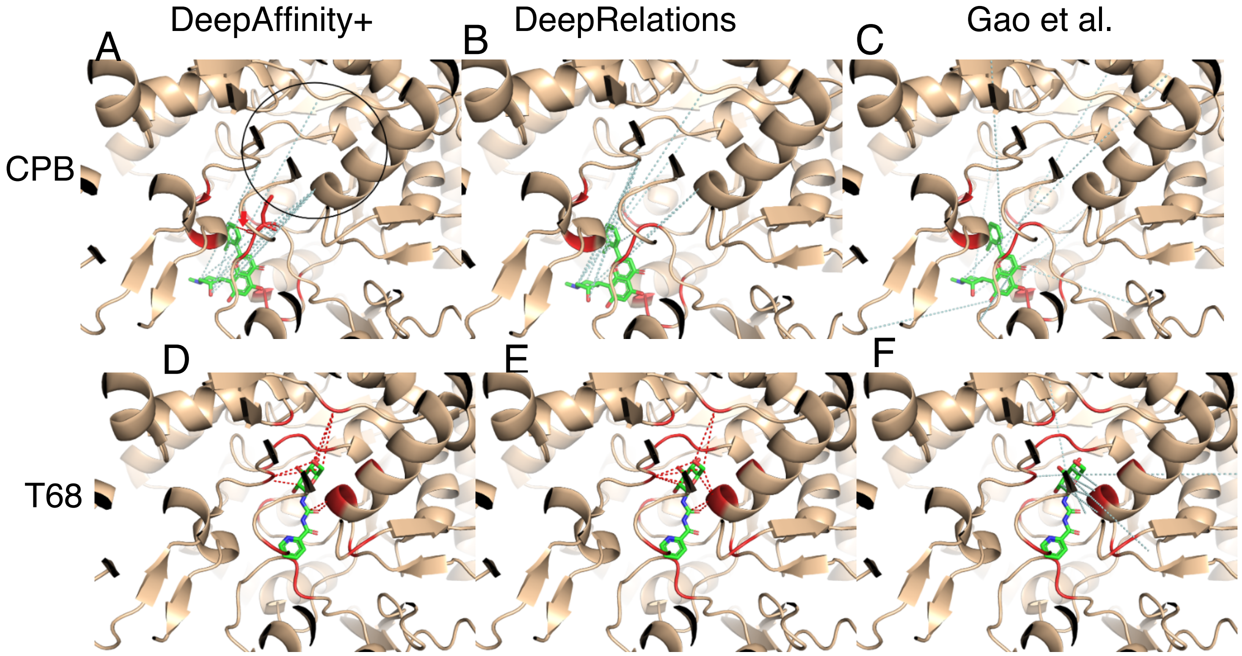


Figure 3.15: Structural visualization of top-10 intermolecular contacts predicted by DeepAffinity+ (left), DeepRelations (middle) and Gao et al. (right) for another two test cases. Here two compounds that are new to training data (CPB: top panels and T68: bottom panels; stick representations) bind to distinct pockets of the human glycogen phosphorylase(wheat cartoons where binding residues are highlighted in red). Shown in dashed lines are top-10 predicted contacts (interactions between protein residues and compound atoms), including correct (red) and incorrect (pale cyan) ones according to LigPlot's definition of native, direct contacts. The red arrow in the top-left CPB panel points the only correct prediction by DeepAffinity+ and the black circle there indicates the binding site for T68. Interestingly, many incorrect predictions by DeepAffinity+ and DeepRelations for CPB were with binding residues to T68.

A pair of new protein and new compound very dissimilar to training examples

Our last case study is even more challenging in that both the protein (the human tyrosine-protein kinase Lck, LCK in short, UniProt ID: P06239) and the compound (HET ID: LHL) are new and they don't even resemble training examples. Specifically, the most similar training protein would be the human tyrosine-protein kinase BTK, BTK in short, UniProt ID: Q06239) with sequence identity at 28%. And the most similar training compound would be K60 (HET ID) with Tanimoto score at 0.12. Indeed our results (Table 3.21) showed that contact AUPRC is just around 0.053. Given the explanation to interpret AUPRC and top-10 contact precision in the first case study, one would notice that the AUPRC value is 14-fold of the baseline (0.004) and 40% of our top-10 contact predictions were true positives (a level close to average protein-ligand docking performances).

As seen in Figure 3.16, again, our contact predictions are more focused in or near the binding site compared to the competing methods, which can be attributed to our structure-aware attention regularization (and supervision). A closer look into the false positives reveal more into our methods. Take DeepAffinity+ as an example. Among the 6 false-positive contact predictions, 4 were pairing correct binding residues with wrong compound atoms, 1 was paired to a protein residue that is a close sequence neighbor (2 residues away) of a correct binding residue, and 1 was paired to a protein residue that is not present in the co-crystal structure but predicted to be spatially close to a correct binding residue. In other words, the origins of false positives in contact prediction include (but are not limited to) pairing with other (nearby) compound atoms and pairing with sequential or predicted spatial neighbors of protein binding-residues. When the criterion of native contacts is relaxed from direct, first-shell contacts within 4Å to more contacts within longer distance cutoffs, the precision level would further increase, which is detailed next.

Global patterns of top-10 predicted contacts

We extended the analysis of the patterns of predicted contacts to all test cases. Considering that the native contacts are defined strictly as direct, first-shell contacts within 4Å, we assess 4∼10-Å distance-distributions of residue-atom pairs predicted by DeepAffinity+ (HRNN-GCN_cstr_sup) and DeepRelations in comparison with Gao et al.. As seen in the global analysis in Figure 3.17,

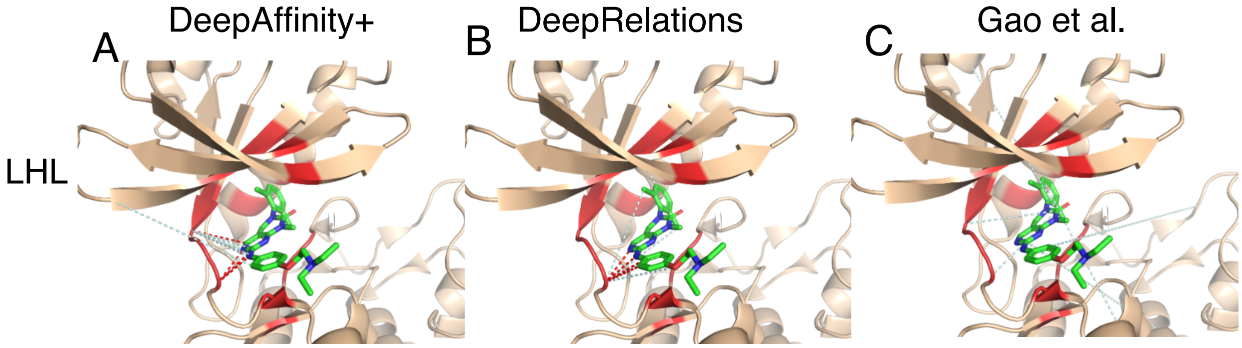


Figure 3.16: Structural visualization of top-10 intermolecular contacts predicted by DeepAffinity+ (left), DeepRelations (middle) and Gao et al. (right) for a difficult test case. Here both the compound (LHL, in sticks) and the protein (tyrosine-protein kinase Lck, in wheat cartoons with binding residues highlighted in red) are new and very dissimilar to training data. The red and pale cyan dashed lines represent correct and incorrect top-10 predicted contacts. DeepAffinity+ and DeepRelations still managed to achieve the precision of 40% in their top-10 contact predictions.

DeepAffinity+ and DeepRelations significantly outperform the competing method in all distance ranges over all test sets. Specifically, among their top-10 contact predictions, around 40% for the default test and new-compound sets were first-shell contacts within 4Å and the ratios increased to about 70% when considering contacts within 10Å. For the more challenging cases of new-protein and both-new sets, the ratios were around 20% and 50%, respectively. These results significantly outperformed the competing method whose such ratios were merely 4∼6% over all sets. Between our two models, DeepRelations behaved similarly as DeepAffinity+ and had more top-10 predictions falling in the long range of 8∼10Å.

Predicted contacts assist and improve protein-ligand docking

From the case studies and the global analysis above, we have concluded that top-10 contact predictions by our methods are enriched with native contacts within 4 Å (20∼40%) as well as dominated by longer-range "contacts" within 10 Å (50∼70%). We therefore test how much the top-10 contact predictions, including false positives, could make a positive impact in the drug discovery process. Picking a typical task – protein-ligand docking and a popular tool – AutoDock Vina [219], we assess how our contact predictions could assist the task by reducing the search space.

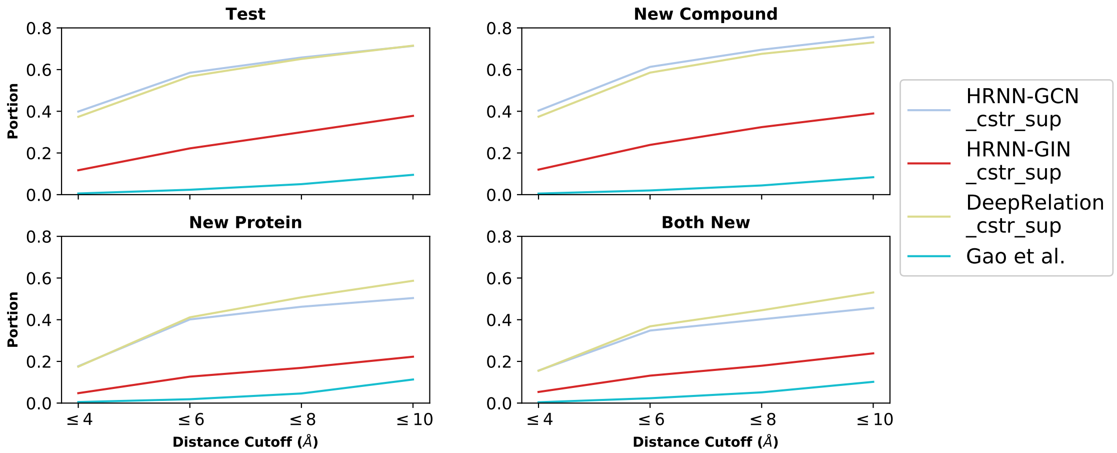


Figure 3.17: Distributions of top-10 contacts, predicted by DeepAffinity+, DeepRelations, and Gao's method, in various distance ranges.

Specifically, we chose the five case studies (except the case where DeepRelations made no correct contact prediction) and performed unbound protein-ligand docking (all protein structures are unbound except PYGM whose structure is co-crystalized with its cognate phosphate AMP). Each pair (rigid protein and flexible ligand) is docked twice: one with the default procedure to define a search "box" covering the entire protein, and the other using a restricted box that barely covers *all* residues in the top-10 DeepRelations contact predictions (including false positives) and then has 20 Å-padding. All the other docking parameters in AutoDock Vina are default, including a total of 9 protein-ligand complex models ordered and reported at the end. Docking performances were evaluated by ligand RMSD of the top few models using the software DOCKRMSD [220].

Protein (UniProt, PDB)	Ligand	Complex PDB	Top-10 Contact Precision	RMSD (Å) – Vina				RMSD (Å) – Contact-Assisted Vina			
				Top 1	Top 3	Top 5	Best	Top 1	Top 3	Top 5	Best
LCK (P06239, 3LCK)	LHL	3KMM	40%	3.02	3.02	2.77	2.77	4.65	2.98	2.45	**2.45**
CA2 (P00918, 2CBA)	AL1	1BNN	50%	18.55	16.62	16.62	16.62	4.78	4.78	4.73	**4.73**
CA2 (P00918, 2CBA)	IT2	3P5A	50%	15.98	15.98	4.01	4.01	3.65	3.65	3.65	**1.59**
PYGM (P00489, 8GPB)	T68	3ZCU	100%	36.40	18.75	18.75	18.75	9.08	2.23	2.23	**1.88**

Table 3.22: Ligand docking performances for case studies. The default Autodock Vina is compared with that assisted by DeepRelations top-10 contact predictions.

Results in Table 3.22 show that AutoDock Vina assisted by DeepRelations top-10 contact predictions had much improved docking performance compared to otherwise. When the top-10 contact precision was 40%, 50%, 50%, and 100%, respectively, the best ligand RMSD (among all 9 complex models) reduced from 2.77 Å, 4.01 Å, 16.62 Å, and 18.75 Ådown to 2.45 Å, 1.59 Å, 4.73 Å, and 1.88 Å, respectively. The quality of the top-1 models also drastically improved in 3 of 4 cases. Although the way to incorporate predicted contacts into protein-ligand docking remains to be optimized, these results have proved that the precision and spatial pattern of our structure-free contact prediction is at a level useful to assist and improve structure-based protein-ligand docking for pose prediction.

3.3.3.9 More utilities from explainable affinity prediction

In the last part of the results, we explore additional utilities of our methods toward facilitating drug discovery: binding-site prediction for proteins and structure-activity relationship (SAR) for compounds. Our methods do not demand protein structures or protein-ligand docking to make these predictions. Rather, they simply aggregate predicted attentions (or predicted weights of residue-atom contacts) or/and decompose predicted affinities. Although not directly designed or optimized for these tasks, our explainable models have shown promising potentials in the tasks toward rational drug discovery.

Binding site prediction The first extended utility we aim at is structure-free and ligand-specific binding-site prediction for proteins. To this end, we feed an arbitrary pair of protein and compound to the trained DeepAffinity+ and DeepRelations models and predict the weights of residue-atom pairs ($\mathcal{W}_{ij}$ where i and j are the indices of a protein residue and a compound atom, respectively). We then calculate the max-marginal attention ($\max_j \mathcal{W}_{ij}$) for each residue i as a weight for ranking. The performances of the residue weights toward ligand-specific binding site prediction are summarized in Figure 3.18. Here binding-site residues of a protein are strictly defined as those making direct, first-shell contacts with a paired compound. Without the help of protein structures, predicted residue-contact maps, or protein-ligand docking, our methods on average achieved AUPRC (AUROC) of around 0.43 (0.77) for the default test and new-compound sets as well as

AUPRC (AUROC) of around 0.18 (0.69) for the more challenging new-protein and both-new sets. In contrast, the competing method had AUPRC and AUROC close to the random performances of 0.004 and 0.50, respectively.

Between DeepAffinity+ and DeepRelations, we noticed that the latter had better performance in predicting binding sites for new proteins. Specifically, the AUPRC (AUROC) increased from 0.17 (0.65) to 0.21 (0.73) for the new-protein set and did from 0.16 (0.65) to 0.20 (0.72) for the both-new set.

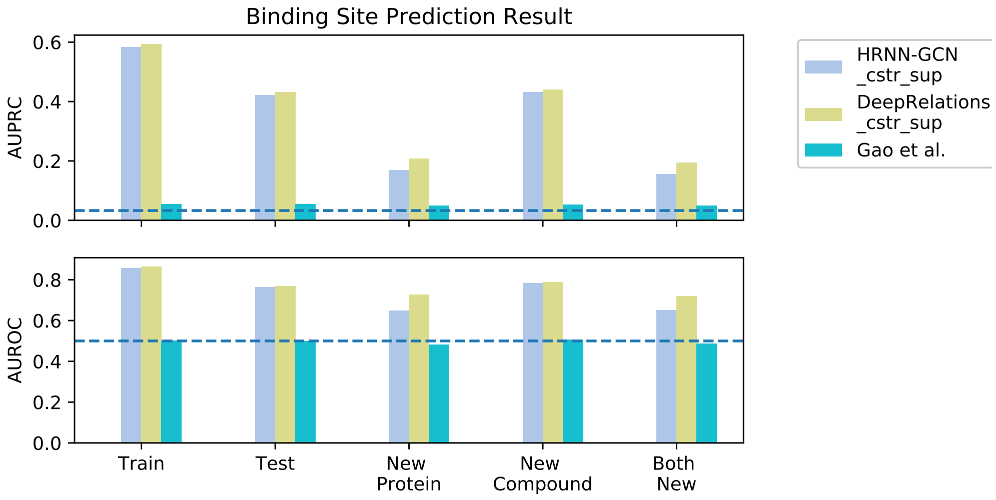
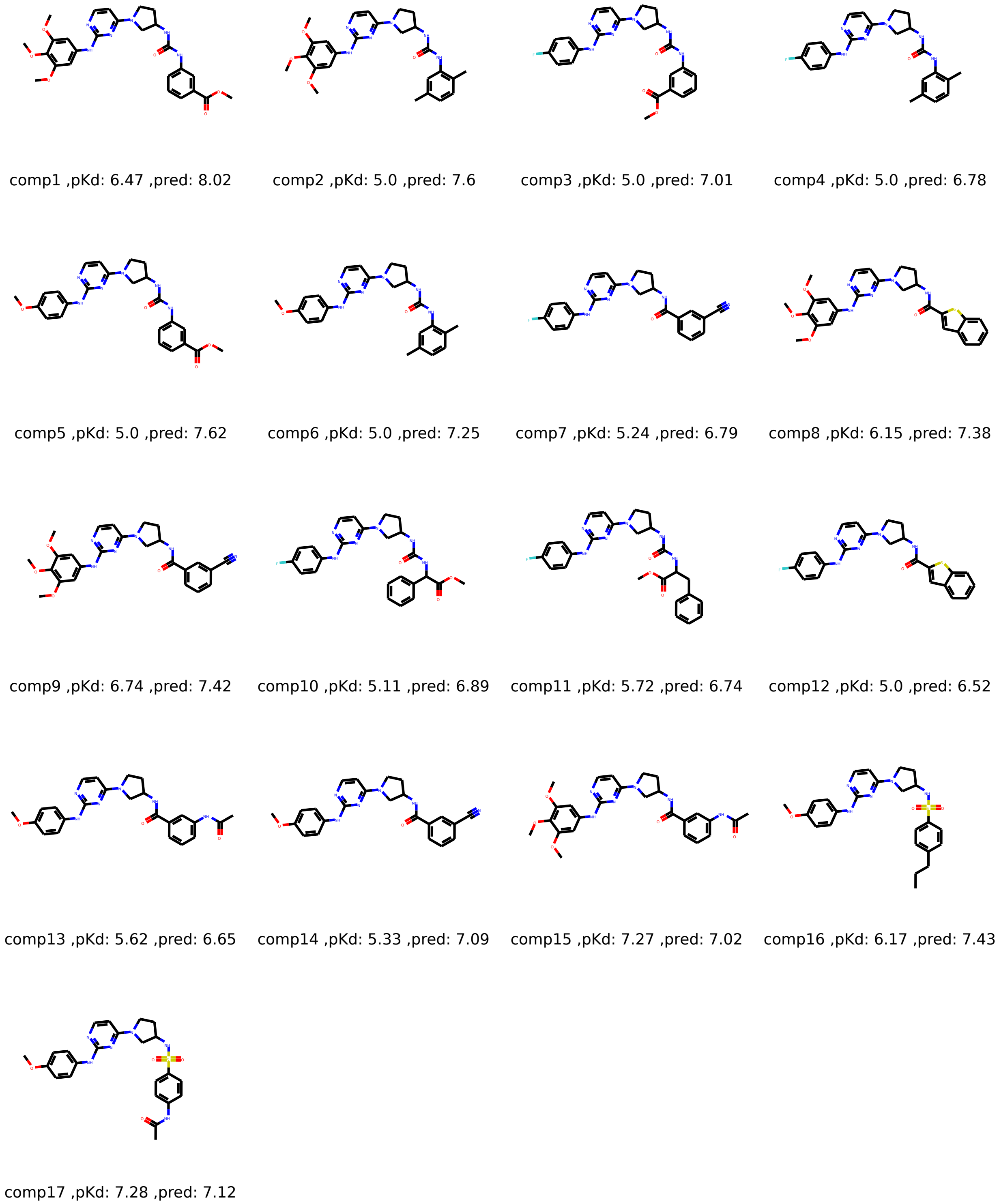


Figure 3.18: Comparing three interpretable methods (DeepAffinity+, DeepRelations, and Gao et al.) in binding-site prediction.

Structure activity relationship (SAR)

The second extended utility we aim at is SAR for compounds. To test the utility we choose two subchallenges (SC3 and SC4) from Grand Challenge 3 of D3R [221], Janus kinase 2 (JAK2) and Angiopoietin-1 receptor (TIE2), that were excluded in our training set (thus new proteins). The most similar proteins to JAK2 and TIE2 in our training set are calcium/calmodulin-dependent protein kinase kinase 2 (CAMKK2, sequence identity 48%) and cyclin-dependent kinase 2 (CDK2, sequence identity 39%), respectively. The two datasets include 17 and 18 congeneric compounds, respectively, with K_d values measured. They were meant to “detect large changes in affinity due to small changes in chemical structure” (`https://drugdesigndata.org/about/grand-challenge-3`). In other words, the datasets focus on the sensitivity of methods targeting SAR. Chemical graphs, actual pK_d, and DeepRelations-predicted pK_d of the JAK2 and TIE2 compounds are in Figure 3.19 and 3.20, respectively.



comp1 ,pKd: 6.47 ,pred: 8.02

comp2 ,pKd: 5.0 ,pred: 7.6

comp3 ,pKd: 5.0 ,pred: 7.01

comp4 ,pKd: 5.0 ,pred: 6.78

comp5 ,pKd: 5.0 ,pred: 7.62

comp6 ,pKd: 5.0 ,pred: 7.25

comp7 ,pKd: 5.24 ,pred: 6.79

comp8 ,pKd: 6.15 ,pred: 7.38

comp9 ,pKd: 6.74 ,pred: 7.42

comp10 ,pKd: 5.11 ,pred: 6.89

comp11 ,pKd: 5.72 ,pred: 6.74

comp12 ,pKd: 5.0 ,pred: 6.52

comp13 ,pKd: 5.62 ,pred: 6.65

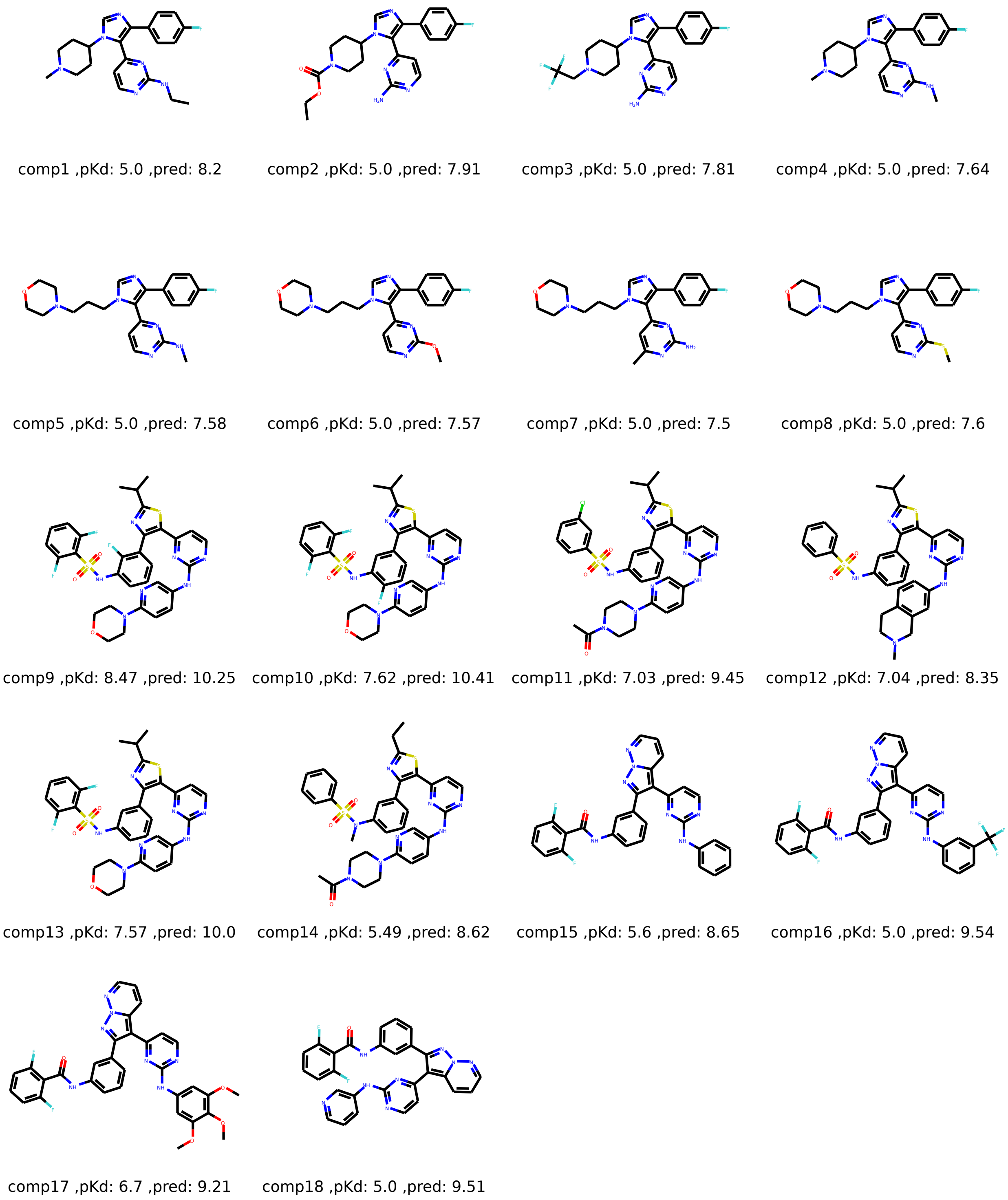
comp14 ,pKd: 5.33 ,pred: 7.09

comp15 ,pKd: 7.27 ,pred: 7.02

comp16 ,pKd: 6.17 ,pred: 7.43

comp17 ,pKd: 7.28 ,pred: 7.12

Figure 3.19: Compounds in JAK2's SAR with actual and DeepRelations-predicted affinities.



comp1 ,pKd: 5.0 ,pred: 8.2

comp2 ,pKd: 5.0 ,pred: 7.91

comp3 ,pKd: 5.0 ,pred: 7.81

comp4 ,pKd: 5.0 ,pred: 7.64

comp5 ,pKd: 5.0 ,pred: 7.58

comp6 ,pKd: 5.0 ,pred: 7.57

comp7 ,pKd: 5.0 ,pred: 7.5

comp8 ,pKd: 5.0 ,pred: 7.6

comp9 ,pKd: 8.47 ,pred: 10.25

comp10 ,pKd: 7.62 ,pred: 10.41

comp11 ,pKd: 7.03 ,pred: 9.45

comp12 ,pKd: 7.04 ,pred: 8.35

comp13 ,pKd: 7.57 ,pred: 10.0

comp14 ,pKd: 5.49 ,pred: 8.62

comp15 ,pKd: 5.6 ,pred: 8.65

comp16 ,pKd: 5.0 ,pred: 9.54

comp17 ,pKd: 6.7 ,pred: 9.21

comp18 ,pKd: 5.0 ,pred: 9.51

Figure 3.20: Compounds in TIE2's SAR with actual and DeepRelations-predicted affinities.

Here we compare our DeepAffinity+ and DeepRelations not only to structure-free Gao et al. but also to 18 structure-based methods from the community that participated in the subchallenges. The assessment metrics for affinity ranking are Kendall's τ and Spearman's ρ as in D3R. A summary of the performances is in Table 3.23. In the case of JAK2, the 17 structure-based methods had τ ranging from 0.71 to -0.56 and ρ ranging from 0.86 to -0.70, including 8 methods with negative τ and ρ. As to the structure-free affinity predictors, Gao et al. had $\tau = -0.42$ and $\rho = -0.54$ whereas our DeepAffinity+ had slightly better $\tau = -0.36$ and $\rho = -0.47$, both outperforming just 1 structure-based method. However, our DeepRelations achieved $\tau = 0.15$ and $\rho = 0.21$, outperforming 12 (two-thirds) of structure-based methods. In the case of TIE2, the 18 structure-based methods had τ ranging from 0.57 to -0.57 and ρ ranging from 0.76 to -0.69, including 8 methods with negative τ and ρ. Interestingly, the best structure-based method for JAK2 only placed 12th among 18 with slightly negative τ and ρ for TIE2. In contrast, all the structure-free affinity predictors performed well for TIE2: Gao et al., DeepAffinity+, and DeepRelations had τ (ρ) reaching 0.60 (0.74), 0.65 (0.79), and 0.61 (0.72), respectively; and they all outperformed the best structure-based method. We note that all 18 structure-based methods used crystal structures of proteins and often-expensive ligand docking whereas structure-free methods did not. Our methods only cost a fraction of a second when making quality predictions for tens to hundreds of compound-protein pairs, thus a useful complement to structure/docking-based methods toward virtual screening.

JAK2 (Subchallenge 3)				TIE2 (Subchallenge 4)			
Ranking[a]	Method(s)	τ	ρ	Ranking	Method(s)	τ	ρ
1–5	5 structure-based methods in D3R	$0.16 \sim 0.71$	$0.25 \sim 0.86$	1	Structure-free DeepAffinity+ (ours)	0.65	0.79
6	Structure-free DeepRelations (ours)	0.15	0.21	2	Structure-free DeepRelations (ours)	0.61	0.72
6	1 structure-based method in D3R	0.13	0.32	2	Structure-free Gao et al.	0.60	0.74
8–18	11 structure-based methods in D3R	$-0.31 \sim 0.05$	$-0.50 \sim 0.05$	2	2 structure-based methods in D3R	0.57	$0.74 \sim 0.76$
19	Structure-free DeepAffinity+	-0.36	-0.47	6–21	16 structure-based methods in D3R	$-0.57 \sim 0.50$	$-0.69 \sim 0.67$
20	Structure-free Gao et al.	-0.42	-0.54				
21	1 structure-based method in D3R	-0.56	-0.70				

[a] The 18 structure-based methods participated in the D3R subchallenges and were assessed officially. The 3 structure-free methods were assessed *post hoc*.

Table 3.23: Summary of scoring performances among three structure-free methods (including our DeepAffinity+ and DeepRelations and eighteen structure-based methods.

Beyond affinity scoring we further examine DeepRelations in extracting SAR knowledge toward drug discovery. A central question in lead optimization is where and how to modify a lead compound to improve its property (affinity here). As a stepping stone, we construct predictors from our DeepRelations in order to anticipate the affinity changes when a functional-group substituent is introduced to a lead. Specifically, we regard our predicted pK_d (p$\hat{K_d}$) as estimated binding energy for a compound-protein pair and our predicted joint-attention $\mathcal{W}_{ij}$ as the fraction of contribution between protein residue i and compound atom j. Borrowing the idea of energy decomposition, we calculate the binding-energy contribution of a functional group R as the product of the predicted binding-energy and the sum-marginals of joint attention: $p\hat{K}_d^R = p\hat{K}_d \cdot (\sum_{j \in R} \sum_i \mathcal{W}_{ij})$. In this way the difference of this R-group contribution, $\Delta p\hat{K}_d^R$, can be a predictor of affinity change when introducing a substituent R-group to a compound.

To test our predictor for lead optimization, we use the JAK2 dataset involving 17 compounds that share a common scaffold and have distinct combinations of two functional groups (3 choices for R_1 and 10 for R_2, see Figure 3.21A and Figure 3.19). We construct 121 pairs of compounds between a weaker binder (origin) and a stronger binder (end). 7, 36, and 78 of the structural changes from the origin to the end compound involve R_1, R_2, and both-R substitutions, respectively. And we compare three methods in predicting these 121 affinity changes with assessment metrics including Pearson's r (main assessment), Spearman's ρ and Kendall's τ (Figure 3.21B–D). A straightforward predictor using DeepRelations' ΔpK_d without decomposition had $r = 0.218$; whereas the decomposed affinity-change predictor ΔpK_d^R improved r to 0.361. If one has access to the protein in complex with a previously discovered compound and can have an accurate estimate of the binding residues, the summation of protein residue i in $p\hat{K}_d^R$ can be just over binding residues rather than all residues. In that case, the new ΔpK_d^R can slightly improve r further to 0.363. Our decomposed affinity-change predictor ΔpK_d^R similarly improved ρ and τ. When we split the analysis into 3 series involving R_1, R_2, and both-R separately, we observed that ΔpK_d^R improved r from 0.267 to 0.753, -0.081 to 0.137, and 0.244 to 0.377, respectively (Figure 3.22). Using the binding-residue information could slightly improve the correlation further. Interestingly, when both R-groups are substituted (78 cases), ΔpK_d^{R1} had a better Pearson's correlation (0.405) with the actual affinity changes than ΔpK_d^{R2} (-0.121) and even ΔpK_d^{R1+R2} (0.375) did (Figure 3.23), potentially suggesting that R_1 group could be explored first for affinity optimization. Once a functional group R is chosen, affinity changes upon any proposed substitution can be predicted using our group-decomposed ΔpK_d^R.

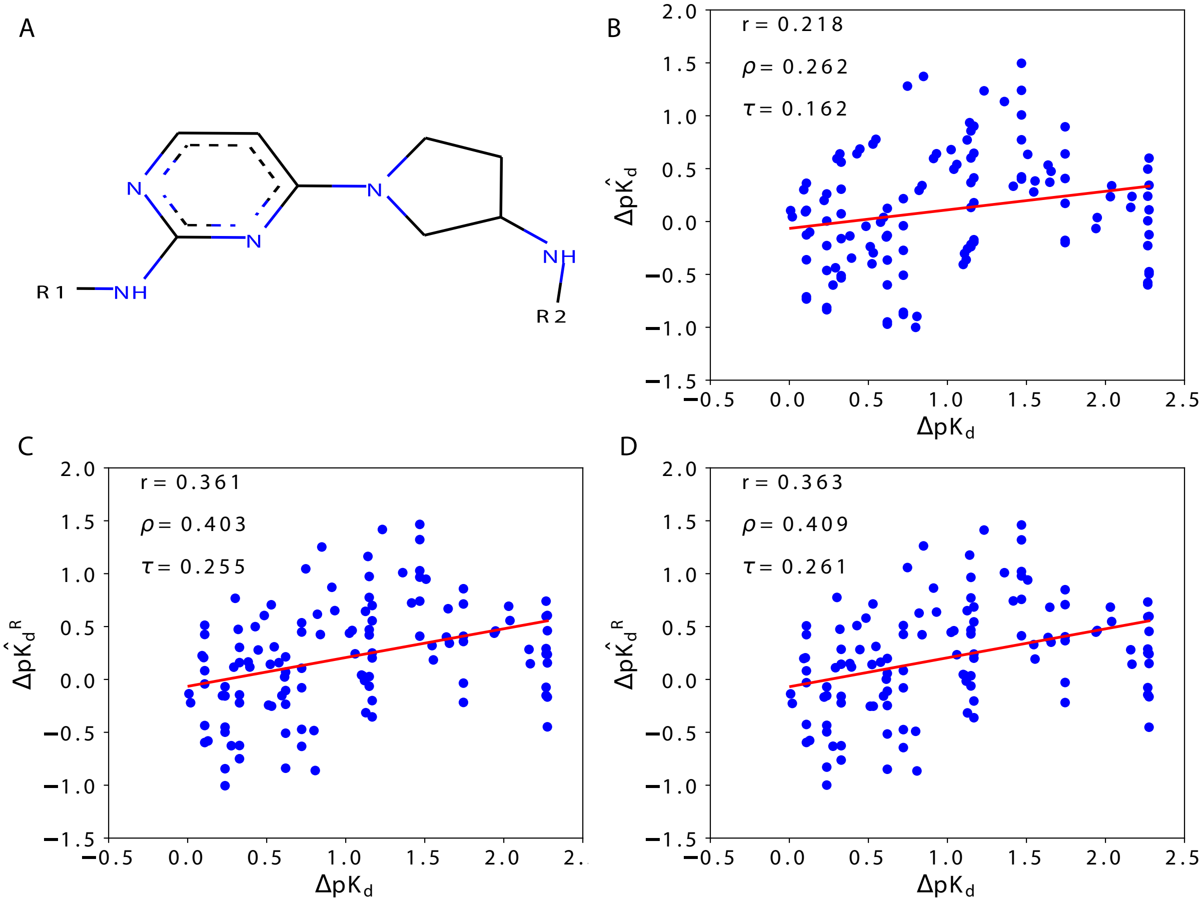


Figure 3.21: Actual (x-axis) *versus* DeepRelations-predicted (y-axis) affinity changes when introducing functional-group substitutions (R_1, R_2 or both in A) to lead compounds for JAK2. The three predictors are: B. predicted affinity change $\Delta p\hat{K}_d$; C. group-decomposed affinity change $\Delta p\hat{K}_d^R$ using all protein residues and the substituent group R alone; and D. group-decomposed affinity change $\Delta p\hat{K}_d^R$ using estimated protein binding residues and the substituent group R alone.

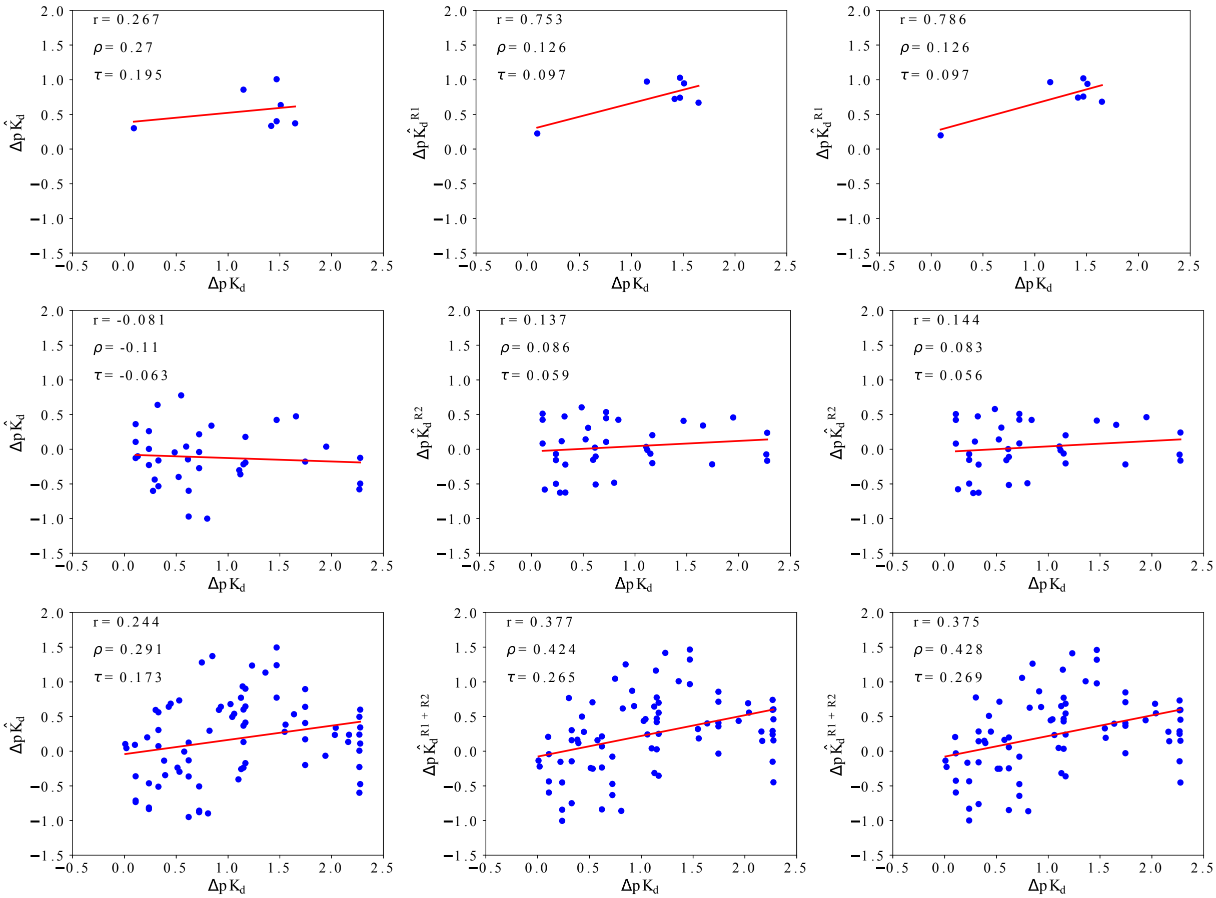


Figure 3.22: Comparison of true affinity changes versus various DeepRelations-based predictions, when JAK2 compounds are changed by substituting functional groups. The first row is for R1 substitution only, second for R2 substitution only and third for both substitutions. The first column is for prediction based on predicted affinity only, the second is based on decomposition (substituent group for the compound and all residues for the protein), and the third column is based on decomposition (substituent group for the compound and binding-site residues for the protein).

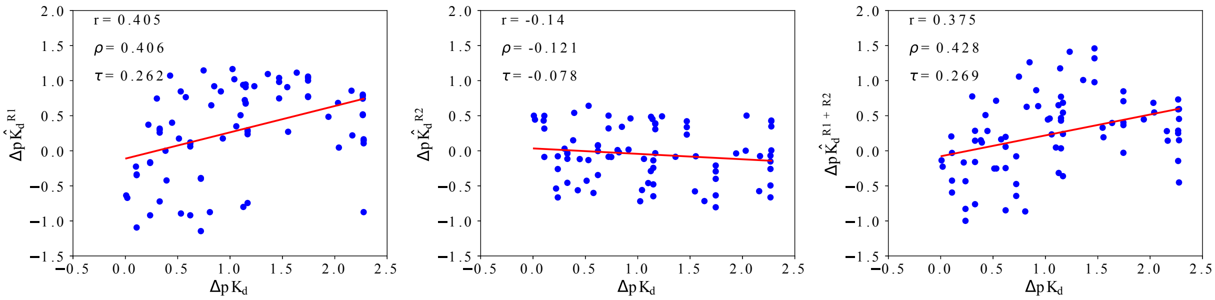


Figure 3.23: For both R1+R2 substitutions, we compare the contribution of R1 (left), R2 (middle) and R1+R2 (right).

3.3.4 Conclusions

Toward accurate and interpretable machine learning for structure-free prediction of compound-protein interactions, we have curated compound-protein interaction dataset annotated with both affinities and intermolecular atom-contacts, assessed the adequacy of current attention-based deep learning models for both accuracy and interpretability, and developed novel machine-learning models (in particular, DeepAffinity+ and DeepRelations) to remarkably enhance interpretability without sacrificing accuracy. We have also shown that our methods' accuracy for affinity prediction is comparable or better than competing (non-interpretable) methods using established benchmark datasets. This is the first study with dedicated model development and systematic model assessment for interpretability in affinity prediction.

Our study has found that commonly-used attention mechanisms alone, although better than chance in most cases, are not satisfying in interpretability. The most attended protein-ligand contacts in affinity prediction do not reveal native contacts underlying affinities at a useful level. The conclusion maintains regardless of the representation of molecules (sequences/strings or graphs) the architecture of neural networks. We have tackled the challenge with three innovative, methodological advances. First, we introduce structure-aware constraints to regularize attentions (or guide their sparsity patterns), using sequence-predicted structural contexts such as protein surfaces and protein residue-residue contact maps. Second, we exploit available native contacts to supervise novel joint attentions, i.e., to teach neural network how to weigh residue-atom pairs when making affinity predictions. Lastly, we build intrinsically explainable model architecture where various atomic relations, reflecting physics laws, are explicitly modeled and aggregated for affinity prediction. Joint attentions are embedded over residue-atom pairs for individual and overall relations. And a multi-stage hierarchy, trained end-to-end, progressively focuses attentions on protein surfaces, binding k-mers and residues, and residue-atom contact pairs. The first two advances are introduced in both DeepAffinity+ and DeepRelations; and the last is additionally introduced in DeepRelations. Their best versions involve hierarchical recurrent neural networks (HRNN) to embed protein sequences and graph convolutional networks (GCN) to embed compound graphs.

Empirical results demonstrate the superiority of DeepAffinity+ and DeepRelations in interpretable and accurate prediction of compound-protein interactions. Their affinity prediction shows generalizability to compounds or/and proteins that are new or even dissimilar to training data. Compared to a competing interpretable method, they boosted the AUPRC for contact prediction (a measure of interpretability) by around 33, 10, 35, and 9-fold for the default test, new-protein, new-compound, and both-new sets, respectively. Importantly,improved model interpretability has shown to contribute to improved model accuracy and generalizability.

Case studies suggest that DeepAffinity+ and DeepRelations predict not only more correct but also more well-patterned contacts that are focused in or near binding sites, which is thanks to the structure-aware regularization and supervision of joint attentions. A global analysis indicates that around 40% (20%) of our top-10 predicted contacts are native contacts that are direct and first-shell for the test and the new-compound set (the new-protein and both-new set). Many “incorrect” predictions due to the strict definition of native contacts were within reasonable ranges — in fact, around 70% (50%) of the top-10 predicted contacts correspond to residue-atom pairs within 10 Å when the set does not (does) involve a new protein. With the precision level and the focused pattern, our top-10 contact predictions (including false positives) have demonstrated their value in assisting and improving protein-ligand docking, while the protocol to incorporate the predictions into docking remains to be optimized.

By aggregating joint attentions and decomposing predicted affinities, we also demonstrate additional utilities of our explainable affinity and contact predictor, toward drug-discovery tasks such as binding site prediction, SAR (scoring) and SAR (lead optimization). Although not directly designed nor optimized for these tasks, our methods and analyses have shown great potentials in these tasks toward facilitating drug discovery.

An additional benefit of our structure-free methods is their broad applicability toward the vast chemical and proteomic spaces. They do not rely on 3D structures of compound-protein complexes or even proteins alone when such structures are often unavailable. The only inputs needed are protein sequences and compound graphs. Meanwhile, they adopt the latest technology to predict structural contexts from protein sequences (such as surfaces, secondary structures, and residue-residue contact maps). And they introduce structure-aware regularization to incorporate the predicted structural contexts into affinity and contact predictions. When structure data are available, DeepRelations can readily integrate such data by using actual rather than predicted structural contexts. We tested the use of actual versus predicted protein residue-residue contact maps and did not observe significant performance differences in our cases.

Our study demonstrates that, it is much more effective to directly teach explainability to machine learning models (such as our structure-aware regularization and supervision of joint attentions) and build explainability into model architectures (such as our explicit modeling of atomic relations in DeepRelations) than to demand explainability from general-purpose models

(such as seeking contact-interpretation from unsupervised, generic attention mechanisms). In other words, designing intrinsically interpretable machine learning models incorporated with domain knowledge, although more difficult, can be much more desired than pursuing interpretability in a *post hoc* manner.

4. DEEP GENERATIVE MODELS FOR DESIGNING DRUG COMBINATIONS FOR OVERCOMING RESISTANCE IN MOLECULAR NETWORKS*

4.1 Overview

In chapter 2, we developed a principle-driven approach for overcoming mutational resistance in a single protein target. The proposed model is guaranteed to provide an optimal solution. However, it cannot scale to large-scale molecular networks such as the human genome due to its exactness criteria. Therefore, in this chapter, we shifted toward data- and principle-driven approach, which is scalable to large scale molecular networks. A data- and principle-driven drug combinations generative model has been developed for overcoming drug resistance through alternative pathway activation in molecular networks. This is the first generative model for designing drug combinations for molecular networks. Firstly, in chapter 3 we developed fast and accurate oracles for compound-protein affinity prediction. These models have been utilized for large-scale drug screening in molecular networks. Secondly, hierarchical variation graph auto-encoders (HGVAE) have been developed to jointly embed domain knowledge such as gene-gene, disease-disease, gene-disease networks, and learn disease representations. Thirdly, drug combinations generative model condition on a disease representation has been formulated as a graph set generation problem and solved through reinforcement learning (RL) and generative adversarial networks (GANs). Our model utilized complementary exposure criteria (learned from FDA-approved drug combinations) for efficient exploration of drug combination space.

4.2 Introduction

Drug resistance is a fundamental barrier to developing robust antimicrobial and anticancer therapies [222, 223]. Its first sign was observed in the 1940s, soon after discovering penicillin [224], the first modern antibiotic. Since then, drug resistance has surfaced and progressed in infectious

*Reprinted with permission from "Network-principled deep generative models for designing drug combinations as graph sets." by M. Karimi, A. Hasanzadeh, Y. Shen 2020. Bioinformatics 36, Supplement_1, i445-i454, Copyright 2020 Oxford University Press.

diseases such as HIV [225], tuberculosis (TB) [226] and hepatitis [227] as well as cancers [228]. Mechanistically, it can emerge through drug efflux [229], activation of alternative pathways [230] and protein mutations [11, 110] while decreasing the efficacy of drugs.

Combination therapy is a resistance-overcoming strategy that has found success in combating HIV [2], TB [3], cancers [4, 5] and so on. Considering that most diseases and their resistances are multifactorial [15, 16], multiple drugs targeting multiple components simultaneously could confer less resistance than individual drugs targeting components separately. Examples include targeting both MEK and BRAF in patients with BRAF V600-mutant melanoma rather than targeting MEK or BRAF alone [13, 14]. The effect of drug combination is usually categorized as synergistic, additive, or antagonistic depending on whether it is greater than, equal to, or less than the sum of individual drug effects [231]. Synergistic combinations are effective at delaying the beginning of the resistance. However, antagonistic combinations are effective at suppressing the expansion of resistance [232, 233], representing offensive and defensive strategies to overcome drug resistance. In particular, offensive strategies cause huge early causalities, but defensive ones anticipate and develop protection against future threats. [232].

Discovering a drug combination to overcome resistance is, however, extremely challenging, even more so than discovering a drug which is already a costly ($\sim$billions of USD) [6] and lengthy ($\sim$12 years) [7] process with low success rates (3.4% phase-1 oncology compounds make it to approval and market) [8]. An apparent challenge, a combinatorial one, is in the scale of chemical space, which is estimated to be 10^{60} for single compounds [234] and can "explode" to 10^{60K} for K-compound combinations. Even if space is restricted to around 10^3 FDA-approved human drugs, there are 10^5–10^6 pairwise combinations. Another challenge, a conceptual one, is in the complexity of systems biology. On top of on-target efficacy and off-target side effects or even toxicity that need to be considered for individual drugs, network-based design principles are much needed for drug combinations that effectively target multiple proteins in a disease module and have low toxicity or even resistance profiles [235, 236].

Current computational models in drug discovery, especially those for predicting pharmacokinetic and pharmacodynamic properties of individual drugs/compounds, can be categorized into discriminative and generative models. Discriminative models predict the distribution of a property for a given molecule, whereas generative models would learn the joint distribution on the property and molecules. For instance, discriminative models have been developed for predicting single compounds' toxicities, based on support vector machines [237], random forest [238], and deep learning [239]. Whereas discriminative models are useful for evaluating given compounds or even searching compound libraries, generative models can effectively design compounds of desired properties in chemical space. Recent advance in inverse molecular design has seen deep generative models such as SMILES representation-based reinforcement learning [240] or recurrent neural networks (RNNs) as well as graph representation-based generative adversarial network (GANs), reinforcement learning [241], and generative tensorial reinforcement learning (GENTRL) [242].

Unlike single drug design, current computational efforts for drug combinations are exclusively focused on discriminative models and lack generative models. The main focus of drug combination is to use discriminate models to identify synergistic or antagonistic drugs for a given specific disease. Examples include the Chou-Talalay method [243], integer linear programming [244], and deep learning [245, 246]. However, it is daunting, if not infeasible, to enumerate all cases in the enormous chemical combinatorial space and evaluate their combination effects using a discriminative model. Not to mention that such methods often lack explainability.

Directly addressing the aforementioned combinatorial and conceptual challenges and filling the void of generative models for drug combinations, in this study, we develop network-based representation learning for diseases and deep generative models for accelerated and principled drug combination design (the general case of K drugs). Recently, by analyzing the network-based relationships between disease proteins and drug targets in the human protein-protein interactome, [247] proposed an elegant principle for FDA-approved drug combinations that targets of two drugs both hit the disease module but cover different neighborhoods. Our methods allow for examining and following the proposed network-based principle [247] to efficiently generate disease-specific drug combinations in a vast chemical combinatorial space. They will also help meet a critical need for computational tools in a battle against quickly evolving bacterial, viral, and tumor populations with accumulating resistance.

To tackle the problem, we have developed a network principle-based deep generative model for faster, broader, and more in-depth exploration of drug combination space by following the principle underlying FDA approved drug combinations. First, we have developed Hierarchical Variational Graph Auto-Encoders (HVGAE) for jointly embedding disease-disease network and gene-gene networks.

Through end-to-end training, we embed genes in a way that they can represent the human interactome. Then, we utilize their embeddings with novel attentional pooling to create features for each disease so that we can embed diseases more accurately. Second, we have also developed a reinforcement-learning based graph-set generator for drug combination design by utilizing both gene/disease embedding and network principles. Besides those for chemical validity and properties, our rewards also include 1) a novel adversarial reward, generalized sliced Wasserstein distance, that fosters generated molecules to be diverse yet similar in distribution to known compounds (ZINC database and FDA-approved drugs) and 2) a network principle-based reward for drug combinations that are feasible for online calculations.

The overall schematics are shown in Fig. 4.1 and details in Sec. 4.4.

4.3 Data

4.3.1 Human interactome and its features

We used the human interactome data (a gene-gene network) from [248] that feature 13,460 proteins interconnected by 141,296 interactions.

We introduced edge features for the human interactome based on the biological nature of edges (interactions). The interactome was compiled by combining experimental support from various sources/databases including 1) regulatory interactions from TRANSFAC [249]; 2) binary interactions from high-throughput (including [250]) and literature-curated datasets (including IntAct [251] and MINT [252]) as well as literature-curated interactions from low-throughput experiments

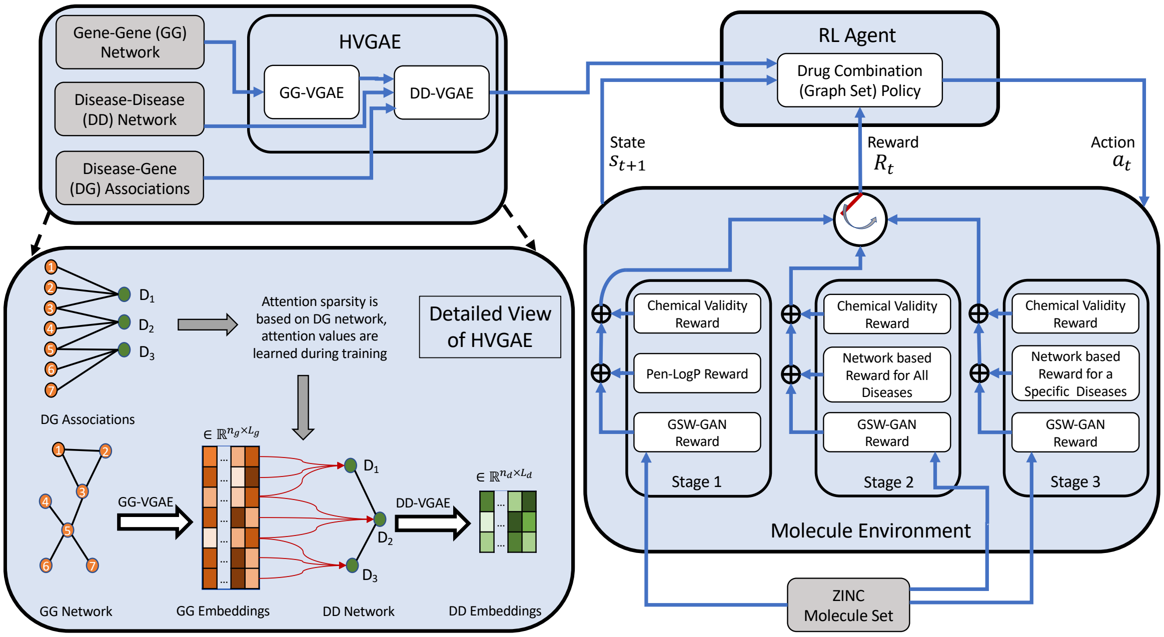


Figure 4.1: Overall schematics of the proposed approach for generating disease-specific drug combinations.

(IntAct, MINT, BioGRID [253], and HPRD [254]); 3) metabolic enzyme-coupled interactions from [255]; 4) protein complexes from CORUM [256]; 5) kinase-substrate pairs from PhosphositePlus [257]; and 6) signaling interactions. In summary, an edge could correspond to one or multiple physical interaction types. So we used a 6-hot encoding for edge features, based on whether an edge corresponds to regulatory, binary, metabolic, complex, kinase and signaling interactions.

We also introduced features for nodes (genes) in the human interactome based on 1) KEGG pathways [258] (336 features) queried through Biopython [259]; 2) Gene Ontology (GO) terms [260] including biological process (30,769 features), molecular function (12,183 features), and cellular component (4,451 features), mapped using the NCBI Gene2Go dataset; 3) disease-gene associations from the database OMIM (Mendelian Inheritance in Man) [261] and the results from Genome-Wide Association Studies (GWAS) [262, 263] (299 features). The last 299 features correspond to 299 diseases represented by the Medical Subject Headings (MeSH) vocabulary [264].

After removing those genes without KEGG pathway information, the human interactome used in this study has 13,119 genes and 352,464 physical interactions.

4.3.2 Disease-disease network

We used a disease-disease network from [248] with 299 nodes (diseases), created based on human interactome data (as detailed earlier), gene expression data [265], disease-gene associations [262, 263, 261], Gene Ontology [260], symptom similarity [266] and comorbidity [267]. The original disease-disease network is a complete graph with real-valued edges. The edge value between the two diseases shows how much they are topologically separated from each other. A positive/negative edge weight indicates that two disease modules are topologically separated/overlapped. Therefore, we used zero-weight as the threshold and pruned positive-valued edges, which results in a disease-disease network of 299 nodes and 5,986 edges (without weights).

4.3.3 Disease-gene associations

We used disease-gene associations from the database OMIM [261]. These associations bridge the aforementioned gene-gene and disease-disease networks into a hierarchical graph of genes and diseases, based on which gene and disease representations will be learned.

4.3.4 Disease classification

For the purpose of assessment, we used the Comparative Toxicogenomics Database (CTD) [268] to classify diseases into 8 classes based on their Disease Ontology (DO) terms [269] where diseases are represented in the MeSH vocabulary [264]. In the CTD database, only 201 of the 299 diseases have a corresponding DO term. Therefore, for the 98 diseases with missing DO terms, we considered the majority of their parents' DO terms, if applicable, as their DO terms. With this approach, we assigned DO terms to 66 such diseases and classified 267 of the 299 diseases. The 32 diseases with DO terms still missing are usually at the top layers of the MeSH tree.

4.3.5 FDA-approved drugs and drug combinations

To assess our deep generative model for drug combination design (to be detailed in Sec. 4.4.2), we consider a comprehensive list of US FDA-approved combination drugs (1940–2018.9) [270]. The dataset contains 419 drug combinations consisting of 328 unique drugs, including 341 (81%), 67 (16%), and 11 (3%) of double, triple, and quadruple drug combinations.

We also utilized the curated drug-disease association from the CTD database [268].

4.4 Methods

We have developed a network-based drug combination generator that can be utilized in overcoming drug resistance. Representing drugs through their molecular graphs, we recast the problem of drug combination generation into network-principled, graph-set generation by incorporating prior knowledge such as human interactome (gene-gene), disease-gene, disease-disease, gene pathway, and gene-GO relationships. Furthermore, we formulate the graph-set generation problem as learning a Reinforcement Learning (RL) agent that iteratively adds substructures and edges to each molecular graph in chemistry- and system-aware environment.

To that end, the RL model is trained to maximize a desired property Q (for example, therapeutic efficacy for drug combinations) while following the valency (chemical validity) rules and being similar in distribution to the prior set of graphs.

As shown in Fig. 4.1, the proposed approach consists of 1) embedding prior knowledge (different network relationships) through Hierarchical Variational Graph Auto-Encoders (HVGAE); and 2) generating drug combinations as graph sets through a reinforcement learning algorithm, which will be detailed next.

Notations: As both gene-gene and disease-disease networks can be represented as graphs, notations are differentiated by superscripts ‘g’ and ’d’ to indicate gene-gene and disease-disease networks, respectively. Drugs (compounds) are also represented as graphs and notations with ‘k’ in the superscript indicates the k-th drug (graph) in the drug combination (graph set).

4.4.1 Hierarchical Variational Graph Auto-Encoders (HVGAE) for representation learning

Suppose that a gene-gene network is represented as a graph $G^{(\mathrm{g})} = (A^{(\mathrm{g})}, \{F^{(\mathrm{g},m)}\}_{m=1}^{M})$, where $A^{(\mathrm{g})} = [A^{(\mathrm{g},1)}, \cdots, A^{(\mathrm{g},n_e)}] \in \{0,1\}^{n_\mathrm{g} \times n_\mathrm{g} \times n_e}$ is the adjacency tensor of the gene-gene network with n_g nodes and n_e edge types (k-hot encoding of 6 types of aforementioned physical interactions such as regulatory, binary, metabolic, complex, kinase and signaling interactions).

We also define $\tilde{A}^{(\mathrm{g})} \in \{0,1\}^{n_\mathrm{g} \times n_\mathrm{g}}$ to be elemenwise OR of $\{A^{(\mathrm{g},1)}, \cdots, A^{(\mathrm{g},n_e)}\}$. Furthermore, $F^{(\mathrm{g},m)}$ denotes the m^{th} set of node features for gene-gene network where M (5 in the study) represents different types of node features such as pathways, 3 GO terms and gene-disease relationship. We also suppose the disease-disease network is represented as graph $G^{(\mathrm{d})} = (A^{(\mathrm{d})}, F^{(\mathrm{d})})$, where $A^{(\mathrm{d})} \in \{0,1\}^{n_\mathrm{d} \times n_\mathrm{d}}$ is the adjacency matrix of the disease-disease network with n_d nodes; and $F^{(\mathrm{d})}$ represents the set of node features for the disease-disease network.

We have developed a hierarchical embedding with 2 levels. In the first level, we embed the gene-gene network to get the features related to each disease, and then we incorporate the disease features within the disease-disease network to embed their relationship. We infer the embedding for each gene and disease jointly through end-to-end training. The proposed HVGAE performs probabilistic auto-encoding to capture the uncertainty of representations, which is in the same spirit as the variational graph auto-encoder models introduced in [271, 272, 273].

4.4.1.1 First level: Gene-Gene embedding

The inference model for variational embedding of the gene-gene network is formulated as follows. We first use M graph neural networks (GNNs) to transform individual nodes' features in M types and then concatenate the M sets of results $\hat{F}^{(\mathrm{g},m)}$ ($m = 1, \ldots, M$) into $\hat{F}^{(\mathrm{g})}$:

$$\begin{gathered}
\hat{F}^{(\mathrm{g},m)} = \mathrm{AGG}\left(\{\mathrm{GNN}_j(A^{(\mathrm{g},j)}, F^{(\mathrm{g},m)})\},\ j = 1, \cdots, n_e\right) \\
\hat{F}^{(\mathrm{g},m)} \in \mathbb{R}^{n_\mathrm{g} \times L_\mathrm{g}}, \quad m = 1, \cdots, M \\
\hat{F}^{(\mathrm{g})} = \mathrm{CONCAT}(\{\hat{F}^{(\mathrm{g},m)}\}_{m=1}^{M}) \in \mathbb{R}^{n_\mathrm{g} \times M L_\mathrm{g}}, \qquad (4.1)
\end{gathered}$$

where AGG is an aggregation function combining output features of GNN_j's for each node. We used a two-layer fully connected neural network with ReLU activation functions followed by a single linear layer in our implementation. We then approximate the posterior distribution of stochastic latent variables $\boldsymbol{Z}^{(\mathrm{g})}$ (containing $\boldsymbol{z}_i^{(\mathrm{g})} \in \mathbb{R}^{L_\mathrm{g}}$ for $i = 1, \cdots, n_\mathrm{g}$ where L_g (32 in this study) is the latent space dimensionality for the i^{th} gene), with a multivariate Gaussian distribution $q(\cdot)$ given the gene-gene network's aggregated node features $\hat{F}^{(\mathrm{g})}$ and adjacency tensor $A^{(\mathrm{g})}$:

$$\begin{gathered}
q(\boldsymbol{Z}^{(\mathrm{g})}|\hat{F}^{(\mathrm{g})}, A^{(\mathrm{g})}) = \prod_{i=1}^{n_\mathrm{g}} q(\boldsymbol{z}_i^{(\mathrm{g})}|\hat{F}^{(\mathrm{g})}, A^{(\mathrm{g})}), \text{where} \\
q(\boldsymbol{z}_i^{(\mathrm{g})}|\hat{F}^{(\mathrm{g})}, A^{(\mathrm{g})}) = \mathcal{N}(\boldsymbol{\mu}_i^{(\mathrm{g})}, \mathit{diag}(\boldsymbol{\sigma}_i^{2,(g)})), \\
\boldsymbol{\mu}^{(\mathrm{g})} = \mathrm{AGG}\left(\{\mathrm{GNN}_{\boldsymbol{\mu},g,j}(A^{(\mathrm{g},j)}, \hat{F}^{(\mathrm{g})})\},\, j = 1, \cdots, n_e\right), \\
\log(\boldsymbol{\sigma}^{(\mathrm{g})}) = \mathrm{AGG}\left(\{\mathrm{GNN}_{\boldsymbol{\sigma},g,j}(A^{(\mathrm{g},j)}, \hat{F}^{(\mathrm{g})})\},\, j = 1, \cdots, n_e\right), \\
\boldsymbol{\mu}^{(\mathrm{g})} \in \mathbb{R}^{n_\mathrm{g} \times L_\mathrm{g}}, \quad \log(\boldsymbol{\sigma}^{(\mathrm{g})}) \in \mathbb{R}^{n_\mathrm{g} \times L_\mathrm{g}}.
\end{gathered} \tag{4.2}$$

where $\boldsymbol{Z}^{(\mathrm{g})} \in \mathbb{R}^{n_\mathrm{g} \times L_\mathrm{g}}$; $\boldsymbol{\mu}^{(\mathrm{g})}$ is the matrix of mean vectors $\boldsymbol{\mu}_i^{(\mathrm{g})}$; and $\boldsymbol{\sigma}^{(\mathrm{g})}$ the matrix of standard deviation vectors $\boldsymbol{\sigma}_i^{(\mathrm{g})}$ ($i = 1, \ldots, n_\mathrm{g}$).

The generative model for the gene-gene network is formulated as:

$$\begin{gathered}
p(\tilde{A}^{(\mathrm{g})}|\boldsymbol{Z}^{(\mathrm{g})}) = \prod_{i=1}^{n} \prod_{j=1}^{n} p(\tilde{A}_{ij}^{(\mathrm{g})}|\boldsymbol{z}_i^{(\mathrm{g})}, \boldsymbol{z}_j^{(\mathrm{g})}), \text{where} \\
p(\tilde{A}_{ij}^{(\mathrm{g})}|\boldsymbol{z}_i^{(\mathrm{g})}, \boldsymbol{z}_j^{(\mathrm{g})}) = \sigma(\boldsymbol{z}_i^{(\mathrm{g})} \boldsymbol{z}_j^{(\mathrm{g})T}),
\end{gathered} \tag{4.3}$$

and $\sigma(\cdot)$ is the logistic sigmoid function. The loss for gene-gene variational embedding is represented as a variational lower bound (ELBO):

$$\begin{aligned}
\mathcal{L}^{(\mathrm{g})} = &\mathbb{E}_{q(\boldsymbol{Z}^{(\mathrm{g})}|\hat{F}^{(\mathrm{g})}, A^{(\mathrm{g})})}[\log p(\tilde{A}^{(\mathrm{g})}|\boldsymbol{Z}^{(\mathrm{g})})] \\
&- \mathrm{KL}\big(q(\boldsymbol{Z}^{(\mathrm{g})}|\hat{F}^{(\mathrm{g})}, A^{(\mathrm{g})})||p(\boldsymbol{Z}^{(\mathrm{g})})\big),
\end{aligned} \tag{4.4}$$

where $\mathrm{KL}\big(q(\cdot)||p(\cdot)\big)$ is the Kullback-Leibler divergence between $q(\cdot)$ and $p(\cdot)$. We take the Gaussian prior for $p(\boldsymbol{Z}^{(\mathrm{g})})$ and make use of the reparameterization trick [271] for training.

4.4.1.2 Second level: disease-disease embedding

The inference model for variational embedding of the disease-disease network is similar to that of the gene-gene network except that the disease-disease network's aggregated node features, $\hat{F}^{(\mathrm{d})}$, are derived through parameterized attentional pooling of $\hat{\boldsymbol{Z}}_r^{(\mathrm{g})}$, latent variables of genes associated with the r^{th} disease (a subset of $\boldsymbol{Z}^{(\mathrm{g})}$):

$$
\begin{aligned}
\boldsymbol{e}_r &= \boldsymbol{v}\ \tanh(\hat{\boldsymbol{Z}}_r^{(\mathrm{g})} W + \boldsymbol{b}), \quad r = 1, \cdots, n_{\mathrm{d}} \\
\boldsymbol{\alpha}_r &= \mathrm{softmax}(\boldsymbol{e}_r), \quad r = 1, \cdots, n_{\mathrm{d}} \\
\hat{F}_r^{(\mathrm{d})} &= \sum_i \boldsymbol{\alpha}_{r,i} \hat{\boldsymbol{Z}}_{r,i}^{(\mathrm{g})}, \quad r = 1, \cdots, n_{\mathrm{d}} \\
\hat{F}^{(\mathrm{d})} &= \mathrm{CONCAT}(\{\hat{F}_r^{(\mathrm{d})}\}_{r=1}^{n_{\mathrm{d}}}) \in \mathbb{R}^{n_{\mathrm{d}} \times L_{\mathrm{d}}},
\end{aligned}
\tag{4.5}
$$

where $\boldsymbol{\alpha}_m$ capture the importance of genes related to the r^{th} disease for calculating its latent representations and L_{d} is the latent space dimensionality of a disease.

Once $\hat{F}^{(\mathrm{d})}$, the disease-disease network's aggregated node features for all diseases, are derived; we again define $q(\boldsymbol{Z}^{(\mathrm{d})}|\hat{F}^{(\mathrm{d})}, A^{(\mathrm{d})})$ for the posterior distribution of stochastic latent variables $\boldsymbol{Z}^{(\mathrm{d})}$ similarly to what we did in Eq. (4.2) except that AGG functions are removed since disease-disease network has one binary adjacency matrix; give the generative decoder $p(A^{(\mathrm{d})}|\boldsymbol{Z}^{(\mathrm{d})})$ for embedding the disease-disease network similarly to what we did in Eq. (4.3); and calculate the variational lowerbound (ELBO) loss $\mathcal{L}^{(\mathrm{d})}$ for the disease-disease network similarly to what we did in Eq. (4.4). More specifically, given the adjacency matrix $A^{(\mathrm{d})}$, and the node attributes for diseas-disease network, which is derived by attentional pooling and denoted by $\hat{F}^{(\mathrm{d})}$, the encoder for this level of HVGAE is defined as:

$$q(\boldsymbol{Z}^{(\mathrm{d})}\,|\,\hat{F}^{(\mathrm{d})},A^{(\mathrm{d})})=\prod_{i=1}^{n_{\mathrm{d}}}q(\boldsymbol{z}_i^{(\mathrm{d})}\,|\,\hat{F}^{(\mathrm{d})},A^{(\mathrm{d})}),\quad \text{where}\quad q(\boldsymbol{z}_i^{(\mathrm{d})}\,|\,\hat{F}^{(\mathrm{d})},A^{(\mathrm{d})})=\mathcal{N}(\boldsymbol{\mu}_i^{(\mathrm{d})},diag(\boldsymbol{\sigma}_i^{2,(d)})),$$

$$\boldsymbol{\mu}^{(\mathrm{d})}=\mathrm{GNN}_{\boldsymbol{\mu},d}(A^{(\mathrm{d})},\hat{F}^{(\mathrm{d})})\quad\in\mathbb{R}^{n_{\mathrm{d}}\times L_{\mathrm{d}}},$$
$$\log(\boldsymbol{\sigma}^{(\mathrm{d})})=\mathrm{GNN}_{\boldsymbol{\sigma},d}(A^{(\mathrm{d})},\hat{F}^{(\mathrm{d})})\quad\in\mathbb{R}^{n_{\mathrm{d}}\times L_{\mathrm{d}}}.\tag{4.6}$$

In the equation above, $\mathrm{GNN}_{\boldsymbol{\mu},d}$ and $\mathrm{GNN}_{\boldsymbol{\sigma},d}$ are multi-layer graph neural networks; n_{d} is the number of nodes; L_{d} is the dimension of latent variables; and $\boldsymbol{\mu}^{(\mathrm{d})}$ and $\boldsymbol{\sigma}^{(\mathrm{d})}$ are matrices of mean vectors and standard deviation vectors, respectively.

The generative model is given by inner product decoder between latent variable. More specifically,

$$p(A^{(\mathrm{d})}\,|\,\boldsymbol{Z}^{(\mathrm{d})})=\prod_{i=1}^{n}\prod_{j=1}^{n}p(A_{ij}^{(\mathrm{d})}\,|\,\boldsymbol{z}_i^{(\mathrm{d})},\boldsymbol{z}_j^{(\mathrm{d})}),\quad \text{where}\quad p(A_{ij}^{(\mathrm{d})}\,|\,\boldsymbol{z}_i^{(\mathrm{d})},\boldsymbol{z}_j^{(\mathrm{d})})=\sigma(\boldsymbol{z}_i^{(\mathrm{d})}\boldsymbol{z}_j^{(\mathrm{d})T}),\tag{4.7}$$

with $\sigma(\cdot)$ as the logistic sigmoid function. Finally, the evidence lower bound (ELBO) for this level of HVGAE is defined as follows:

$$\mathcal{L}^{(\mathrm{d})}=\mathbb{E}_{q(\boldsymbol{Z}^{(\mathrm{d})}\,|\,\hat{F}^{(\mathrm{d})},A^{(\mathrm{d})})}[\log p(A^{(\mathrm{d})}\,|\,\boldsymbol{Z}^{(\mathrm{d})})]-\mathrm{KL}\big(q(\boldsymbol{Z}^{(\mathrm{d})}\,|\,\hat{F}^{(\mathrm{d})},A^{(\mathrm{d})})\,||\,p(\boldsymbol{Z}^{(\mathrm{d})})\big),\tag{4.8}$$

Both levels of our proposed HVGAE, i.e. gene-gene and disease-disease variational graph representation learning, are jointly trained in an end-to-end fashion using the following overall loss:

$$\mathcal{L}^{\mathrm{HVGAE}}=\mathcal{L}^{(\mathrm{d})}+\mathcal{L}^{(\mathrm{g})}.\tag{4.9}$$

4.4.2 Reinforcement learning-based graph-set generator for drug combinations

In this section, we introduce the reinforcement learning-based drug combination generator. We will detail 1) the state space of graph sets (K compounds) and the action space of graph-set growth; 2) multi-objective rewards including chemical validity and our generalized sliced Wasserstein reward for individual drugs as well as our newly designed network principle-based reward for drug combinations; 3) policy network that learns to take actions in the rewarding environment.

4.4.2.1 State and action space

We represent a graph set (drug combination) with K graphs as $\mathcal{G} = \{G^{(k)}\}_{k=1}^{K}$. Each graph $G^{(k)} = (A^{(k)}, E^{(k)}, F^{(k)})$ where $A^{(k)} \in \{0,1\}^{n_k \times n_k}$ is the adjacency matrix, $F^{(k)} \in \mathbb{R}^{n_k \times \phi}$ the node feature matrix, $E^{(k)} \in \{0,1\}^{\epsilon \times n_k \times n_k}$ the edge-conditioned adjacency tensor, and n_k the number of vertices for the k^{th} graph, respectively; and ϕ is the number of features per nodes and ϵ the number of edge types.

The state space $\mathcal{G}$ is the set of all K graphs with different numbers and types of nodes or edges. Specifically, the state of the environment s_t at iteration t is defined as the intermediate graph set $\mathcal{G}_t = \{G_t^{(k)}\}_{k=1}^{K}$ generated so far which is fully observable by the RL agent.

The action space is the set of edges that can be added to the graph set. An action a_t at iteration t is analogous to link prediction in each graph in the set. More specifically, a link can either connect a new subgraph (a single node/atom or a subgraph/drug-substructure) to a node in $G_t^{(k)}$ or connect existing nodes within graph $G_t^{(k)}$. The actions can be interpreted as connecting the current graph with a member of scaffold subgraphs set C. Mathematically, for $G_t^{(k)}$, graph k at step t, the action $a_t^{(k)}$ is the quadruple of $a_t^{(k)} = \text{CONCAT}(a_{\text{first,t}}^{(k)}, a_{\text{second,t}}^{(k)}, a_{\text{edge,t}}^{(k)}, a_{\text{stop,t}}^{(k)})$.

4.4.2.2 Multi-objective reward

We have defined a multi-objective reward R_t to satisfy certain requirements in drug combination therapy. First, a chemical validity reward maintains that individual compounds are chemically valid. Second, a novel adversarial reward, generalized sliced Wasserstein GAN (GS-WGAN), enforces generated compounds that are synthesizable and “drug-like” by following the distribution of synthesizable compounds in the ZINC database [274] or FDA-approved drugs. Third, a network principle-based award would encourage individual drugs to target the desired disease module but not to overlap in their target sets. Toxicity due to drug-drug interactions can also be included as a reward. It is intentionally left out in this study so that toxicity can be evaluated for drug combinations designed to follow the network principle.

When training the RL agent, we use different reward combinations in different stages. We first only use the weighted combination of chemical validity and GS-WGAN awards learning over drug combinations for all diseases; then we remove the penalized logP (Pen-logP) portion of chemical validity and add adversarial loss again while learning over drug combinations for all diseases; finally use the combination of the three rewards as in the second stage but focusing on a target disease and possibly on restricted actions/scaffolds (in a spirit similar to transfer learning). The three types of rewards are detailed as follows.

Chemical validity reward for individual drugs. A small positive reward is assigned if the action does not violate valency rules. Otherwise, a small negative reward is assigned. This is an intermediate reward added at each step. Another reward is on penalized logP (lipophilicity where P is the octanol-water partition coefficient) or Pen-logP values. The design and the parameters of this reward are adopted from [241] without optimization.

Adversarial reward using generalized sliced Wasserstein distance (GSWD). To ensure that the generated molecules resemble a given set of molecules (such as those in ZINC or FDA-approved), we deploy Generative Adversarial Networks (GAN). GANs are very successful at modeling high-dimensional distributions from given samples. However, they are known to suffer from training unsuitability and cannot generate diverse samples (a phenomenon known as *mode collapse*).

Wasserstein GANs (WGAN) have shown to improve stability and mode collapse by replacing the Jenson-Shannon divergence in original GAN formulation with the Wasserstein Distance (WD) [275, 276, 277]. More specifically, the objective function in WGAN with gradient penalty [278] is defined as follows:

$$\min_{\theta} \max_{\phi} V_{\mathrm{W}}(\pi_\theta, D_\phi) + \lambda R(D_\phi), \tag{4.10}$$

$$\text{with} \quad V_W(\pi_\theta, D_\phi) = \mathbb{E}_{\mathbf{x}\sim p_r}[\log D_\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{y}\sim\pi_\theta}[\log D_\phi(\mathbf{y})],$$

where p_r is the data distribution, λ is a hyper-parameter, R is the Lipschitz continuity regularization term, D_ϕ is the critic with parameters ϕ, and π_θ is the policy (generator) with parameters θ.

Despite the theoretical advantages of WGANs, solving equation (4.10) is computationally expensive and intractable for high dimensional data. To overcome this problem, we propose and formulate a novel Generalized Sliced WGAN (GS-WGAN) which deploys Generalized Sliced Wasserstein Distance (GSWD) [279]. GSWD, first, factorizes high-dimensional probabilities into multiple marginal 1D distributions with generalized Radon transform. Then, by taking advantage of the closed-form solution of Wasserstein distance in 1D, the distance between two distributions is approximated by the sum of Wasserstein distances of marginal 1D distributions. More specifically, let $\mathcal{R}$ represent generalized Radon transform operator. The generalized Radon transform (GRT) of a probability distribution $\mathbb{P}(\cdot)$ which is defined as follows:

$$\mathcal{R}P(t, \psi) = \int_{\mathbb{R}^d} \mathbb{P}(\mathbf{x})\, \delta(t - f(\mathbf{x}, \psi))\, d\mathbf{x}, \tag{4.11}$$

where $\delta(\cdot)$ is the one-dimensional Dirac delta function, $t \in \mathbb{R}$ is a scalar, ψ is a unit vector in the unit hyper-sphere in a d-dimensional space ($\mathbb{S}^{d-1}$), and f is a projection function whose parameters will be learned in training. Injectivity of the GRT [280] is the requirement for the GSWD to be a valid distance.

We use linear project $f(x, \psi)$ here and can easily extend to two nonlinear cases that maintain the GRT-injectivity (circular nonlinear projections or homogeneous polynomials with an odd degree).

GSWD between two d-dimensional distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$ is therefore defined as:

$$\text{GSWD}(\mathbb{P}_X,\, \mathbb{P}_Y) = \int_{\mathbb{S}^{d-1}} \text{WD}(\mathcal{R}P_X(\cdot, \psi),\, \mathcal{R}P_Y(\cdot, \psi))\, d\psi\,. \tag{4.12}$$

The integral in the above equation can be approximated with a Riemann sum. Knowing the definition of GSWD, we define the objective function of GS-WGAN as follows:

$$\min_{\theta} \max_{\phi} V_{GSW}(\pi_\theta, D_\phi) + \lambda R(D_\phi), \tag{4.13}$$

$$\text{s.t.} \quad V_{\text{GSW}}(\pi_\theta, D_\phi) = \int_{\psi \in \mathbb{S}^{d-1}} \mathbb{E}_{\mathbf{x} \sim p_{\text{r}}}[\log D_\phi(\mathbf{x})] - \tag{4.14}$$

$$\mathbb{E}_{\mathbf{y} \sim \pi_\theta}[\log D_\phi(\mathbf{y})] \, d\psi \,,$$

where the parameters and notations are the same as defined in Eq. (4.10).

We note that $\mathbf{x}$ and $\mathbf{y}$ in Eq. (4.13) are random variables in $\mathbb{R}^d$, which is not a reasonable assumption for graphs. To that end, we use an embedding function g that maps each graph to a vector in $\mathbb{R}^d$. We use graph convolutional layers, followed by fully connected layers, to implement g. We deploy the same type of neural network architecture for D_ϕ. We use $\text{R}_{\text{advers}} = -V_{\text{GSW}}(\pi_\theta, D_\phi)$ as the adversarial reward used together with other rewards, and optimize the total rewards with a policy gradient method (Sec. 4.4.2.3).

Network principle-based reward for drug combinations. Proteins or genes associated with a disease tend to form a localized neighborhood disease module rather than scattering randomly in the interactome [247]. A network-based score has been introduced [248], to efficiently capture the network proximity of a drug (X) and disease (Y) based on the shortest-path length $d(x, y)$ between a drug target (x) and a disease protein (y):

$$\begin{aligned} Z &= \frac{d(X,Y) - \bar{d}}{\sigma_d} \\ d(X,Y) &= \frac{1}{||Y||} \sum_{y \in Y} \min_{x \in X} d(x,y), \end{aligned} \tag{4.15}$$

where $d(\cdot, \cdot)$ is the shortest path distance; $\bar{d}$ and σ_d are the mean and standard deviation of the reference distribution which is corresponding to the expected network topological distance between two randomly selected groups of proteins matched to size and degree (connectivity) distribution as the original disease proteins and drug targets in the human interactome. Z-score being negative ($Z < 0$) implies network proximity of disease module and drug targets which is desirable. From the drug combination perspective, it has been shown that the complementary exposed drug-drug relationship has the least side drug side affect and the most drug combination efficacy [247]. Complementary exposed drug-drug (X_1 and X_2) relationship means that the drug targets (x_1) and drug targets (x_2) are not in the same neighborhood and has the least overlapping. Therefore, [247] have proposed a network-separation score which is formulated as follow:

$$s_{X_1,X_2} = d(X_1, X_2) - \frac{d(X_1, X_1) + d(X_2, X_2)}{2}, \tag{4.16}$$

where $d(X_1, X_2)$ is the mean shortest path distance between drugs X_1 and X_2; $d(X_1, X_1)$ and $d(X_2, X_2)$ are the mean shortest path distance within drug targets X_1 and X_2 respectively [247]. The separation score being positive ($s > 0$) implies to network are separated from each other which is desirable. We have extended and combined these scores for general drug combination therapy where we have a set of k drugs $\{X_1, \cdots, X_k\}$ and disease Y:

$$\mathrm{R}_{\text{network}} = \lambda_1 \sum_{i=1}^{k} \sum_{j>i} s(X_i, X_j) - \lambda_2 \sum_{i=1}^{k} Z(X_i, Y) \tag{4.17}$$

However, the exact online calculation of the reward $\mathrm{R}_{\text{network}}$ is infeasible while training across all the diseases and the whole human interactome with more than 13K nodes and 352K edges. Therefore, we have developed a relaxed version of the reward, which is feasible for online calculation and correlates with the actual reward. Specifically, we consider the normalized exclusive or (XOR) of intersections of disease modules with drug targets:

$$\hat{\mathrm{R}}_{\text{network}} = \frac{Y \cap (X_1 \oplus \cdots \oplus X_k)}{|Y|} = \frac{(X_1 \cap Y) \oplus \cdots \oplus (X_k \cap Y)}{|Y|}. \tag{4.18}$$

The relaxed network principle-based reward is penalizing a drug combination if the overlap between drug targets in the disease module is high. Therefore it will prevent adverse drug-drug interactions. We scaled the network score by a constant (equals 10) such that the score would be in the same range as Pen-logP and can use the same weight in the total reward as Pen-logP did in

[241].

For a generated compound, we predict its protein targets by DeepAffinity [19], judging by whether the predicted IC_{50} is below 1μM.

4.4.2.3 Policy Network

Having explained the graph generation environment (various rewards), we outline the architecture of our proposed policy network. Our method takes the intermediate graph set $\mathcal{G}_t$ and the collection of scaffold subgraphs C as inputs and outputs the action a_t, which predicts a new link for each of the graphs in $\mathcal{G}_t$ [241].

Since the input to our policy network is a set of K compounds or graphs $\{G_t^{(k)} \cup C\}_{k=1}^K$, we first deploy some layers of graph neural network to process each of the graphs. More specifically,

$$X^{(k)} = \text{GNN}^{(k)}(G_t^{(k)} \cup C), \quad \text{for } k = 1, \dots, K, \tag{4.19}$$

where $\text{GNN}^{(k)}$ is a multilayer graph neural network. The link prediction based action at iteration t is a concatenation of four components for each of the K graphs: selection of two nodes, prediction of edge type, and prediction of termination. Each component is sampled according to a predicted distribution [241]. Specifically, each component is sampled according to a predicted distribution governed by the following equations [241]:

$$\begin{aligned}
a_t^{(k)} &= \text{CONCAT}\big(a_{\text{fr},t}^{(k)}, a_{\text{sc},t}^{(k)}, a_{\text{et},t}^{(k)}, a_{\text{tr},t}^{(k)}\big), \\
a_t &= \{a_t^{(k)}\}_{k=1}^K, \\
X_t &= \{X_t^{(k)}\}_{k=1}^K,
\end{aligned}$$

where

$$\begin{aligned}
f_1(s_t) &= \textsc{softmax}(\mathrm{FC}_{\mathrm{fr}}(X_t, Y)),\\
a^{(k)}_{\mathrm{fr,t}} &\sim f_1(s_t) \in \{0,1\}^{n_k};\\
f_2(s_t) &= \textsc{softmax}(\mathrm{FC}_{\mathrm{sc}}(X_{a_{\mathrm{fr},t}}, X_t, Y)),\\
a^{(k)}_{\mathrm{sc,t}} &\sim f_2(s_t) \in \{0,1\}^{(n_k+c)};\\
f_3(s_t) &= \textsc{softmax}(\mathrm{FC}_{\mathrm{et}}(X_{a_{\mathrm{fr},t}}, X_{a_{\mathrm{sc},t}}, Y)),\\
a^{(k)}_{\mathrm{et,t}} &\sim f_3(s_t) \in \{0,1\}^{b};\\
f_4(s_t) &= \textsc{softmax}(\mathrm{FC}_{\mathrm{tr}}(X_t, Y)),\\
a^{(k)}_{\mathrm{tr,t}} &\sim f_4(s_t) \in \{0,1\};
\end{aligned} \tag{4.20}$$

where FC's are fully connected neural networks, subscripts 'fr' and 'sc' indicate the first and the second drug, and c is the cardinality of C. Also, Y is the targeted disease we are generating drug combination for.

We note that the first node is always chosen from $\mathcal{G}_t$ while the next node is chosen from $\{G^{(k)}_t \cup C\}^K_{k=1}$. We also note that infeasible actions (i.e., actions that do not pass valency check) proposed by the policy network are rejected, and the state remains unchanged. We adopt Proximal Policy Optimization (PPO) [281], one of the state-of-the-art policy gradient methods, to train the model.

4.5 Results

To assess the performance of our proposed model, we have designed a series of experiments. In section 4.5.1, we first compare HVGAE to state-of-art graph embedding methods in disease-disease network representation learning and further include several variants of HVGAE for ablation studies. We then assess the performance of the proposed reinforcement learning method in two aspects. In a landscape assessment in Section 4.5.2, we examine designed pairwise compound-combinations for 299 diseases in quantitative scores of following a network-based principle [247]. In Section 4.5.3, we focus on four case studies involving multiple diseases of various systems-pharmacology strategies. Our method is capable of generating higher-order combinations of K drugs. As FDA-approved drug combinations are often pairs, here we design compound pairs from the scaffolds of FDA-approved drug pairs. We further delve into designed compound pairs to understand the benefit of following network principles in lowering toxicity from drug-drug interactions. We also do so to understand their systems pharmacology strategies in comparison to the FDA-approved drug combinations.

4.5.1 HVGAE representation compares favorably to baselines

4.5.1.1 Experiment setup

To assess the performance of our proposed embedding method HVGAE, we compare its performance in (disease-disease) network reconstruction with Node2Vec [282], DeepWalk [283], and VGAE [271], as well as some variants of our own model for ablation study. Node2Vec and DeepWalk are random walk based models that do not capture node attributes. Hence we only used the disease-disease graph structure. For VGAE, we used the identity matrix as node attributes, as suggested by the authors.

For our HVGAE described in Sec. 4.4.1, we also considered two variants for ablation study: HVGAE-disjoint does not jointly embed gene-gene and disease-disease networks and does not use attentional pooling for disease embedding; whereas HVGAE-noAtt just does not use attentional pooling. Specifically, in HVGAE-disjoint, we first learned an embedding for the gene-gene network, then used the sum of the mean of the node representations of genes affected by the disease as its node attributes. In HVGAE-noAtt, we jointly learned the representations while using the sum of the mean of the node representations of genes as node attributes for the disease-disease network.

In node2vec and DeepWalk, the walk length was set to 80, the number of walks starting at each node was set to 10, and the nodes were embedded in a 16-dimensional space. The window size was 10 for node2vec while it is set to 10 in DeepWalk. All models were trained using Adam optimizer. In VGAE, a 32-dimensional graph convolutional (GC) layer followed by two 16-dimensional layers was used for mean and variance inference. The learning rate was set to 0.01.

For HVGAE and its variants (for ablation study), we embed gene networks in 32-dimensional space using a single GC layer with 32 filters for each of the 5 types of input followed by a 64-

dimensional GC layer and two 32-dimensional GC layer to infer mean and variance of the representation. We used a single 32-dimensional fully connected (FC) layer for the attention layer. For disease-disease network embedding, we deployed a single 32-dimensional GC layer followed by two 16-dimensional layers for mean and variance inference resulting in 16-dimensional embedding for the disease-disease network.

Learning rates were set to 0.001. The models were trained for 1,000 epochs choosing the best representation based on their reconstruction performance at each epoch.

4.5.1.2 Numerical analysis and ablation study for network embedding

Table 4.1 summarizes the reconstruction performance of the aforementioned methods. Compared to all baselines, our HVGAE showed the best performance in all metrics considered. Node2Vec and DeepWalk showed the worst performance as they only use the graph structure. The performance of VGAE was very close to DeepWalk. This is due to the fact that no attributes have been provided to VGAE despite having the capability of capturing attributes.

Table 4.1: Graph reconstruction performances (unit: %) in the disease-disease network using our proposed HVGAE and baselines. F-1 scores are based on 50% threshold.

Method	AUC-ROC	AP	F1-Macro	F1-Micro
Node2Vec	79.01	72.82	35.73	51.10
DeepWalk	79.32	73.77	40.28	53.30
VGAE	88.12	85.71	60.19	64.98
HVGAE-disjoint	91.45	90.72	73.45	74.77
HVGAE-noAtt	92.83	92.34	73.81	75.14
HVGAE	**96.11**	**95.89**	**79.77**	**80.45**

Compared to VGAE, HVGAE-disjoint without joint embedding or attentional pooling still saw a better performance, which suggests that the attributes generated by the gene-gene network contain meaningful features about the disease-disease network. The slight performance gain from HVGAE-disjoint to HVGAE-noAtt shows that joint learning of both networks hierarchically helps to render more informative features for the disease-disease network. Finally, HVGAE had another performance boost compared to HVGAE-noAtt and outperformed all competing methods, which shows the benefit of attentional pooling. Specifically, the attention layer of HVGAE allows the model to produce features that are specifically informative for the disease-disease network representation learning.

4.5.2 Our model generates drug combinations following network principles across diseases

4.5.2.1 Experiment setup

We have trained the proposed reinforcement model in 3 stages using different rewards, disease sets, and action spaces to increasingly focus on a target disease while exploiting all diseases whose representations already jointly embed gene-gene, disease-disease, and gene-disease networks. In the first stage, we train the model only to generate drug-like small-molecules that follow the chemistry valency reward, lipophilicity reward (logP where P is the octanol-water partition coefficient) [241], and our novel adversarial reward for individual compounds. In this study, we trained the model for 3 days (4,800 iterations) to learn to follow the valency conditions and promote high logP for generated compounds.

In the second stage, we start from the trained model at the end of the first stage ("warm-start" or "pre-training"). Moreover, we continue to train the model to generate good drug combinations across all diseases. We do so by adding the network principle-based reward for compound combinations and sequentially generating drug combinations for each disease one by one. Then, we calculate the network-based score for the generated drug combinations at the last epoch across disease ontologies and compare them with the FDA-approved melanoma drug combinations' network-based score. In this study, we trained the model for 1,500 iterations to generate drug combinations across all 299 diseases. In each iteration, we generated 8 drug combinations for a given disease. We adopted PPO [281] with a learning rate of 0.001 to train the proposed RL for both stages.

The last stage is disease-specific and will be detailed in Sec. 4.5.3.

4.5.2.2 Numerical analysis

Across disease ontologies, we quantify the performance of the proposed RL (stage 2 model first) using quantitative scores of compound-combinations following a network-based principle [247]. We consider the generated combinations in the last epoch (the last 299 iterations) and calculate the network score $\hat{\mathrm{R}}_{\mathrm{network}}$ based on disease ontologies. We assess our model based on two versions of disease classification, original disease ontology, and its extension, explained in Sec. 4.3.4. Table 4.2 summarizes the network-based scores for our model. Specifically, suppose that the set of targets for drug 1 and 2 are represented by A and B whereas the disease module is the universal set Ω, we report the portion exclusively covered by drug 1 (η_{A-B}), exclusively covered by drug 2 (η_{B-A}), overlapped by both ($\eta_{A\cap B}$), and collectively by both ($\eta_{A\cup B}$). As a reference, we calculated the corresponding network scores for 3 FDA-approved drug combinations for melanoma.

Based on the results shown in Table 4.2, we note that across all disease classes, the designed compound combinations learned in an environment where the network principle[247] was rewarded did achieve the desired performances. Specifically, their overlaps in disease modules were low as $\eta_{A\cap B}$ fractions are around 0.1; whereas their joint coverage in disease modules was high as $\eta_{A\cup B}$ fractions were in the range of 0.4–0.5 for all diseases.

Table 4.2: Network-based score for the generated drug combinations based on disease ontology classifications.

	Disease Ontology				**Disease Ontology extended**			
	η_{A-B}	η_{B-A}	$\eta_{A\cap B}$	$\eta_{A\cup B}$	η_{A-B}	η_{B-A}	$\eta_{A\cap B}$	$\eta_{A\cup B}$
infectious disease	0.25	0.10	0.06	0.41	0.20	0.07	0.05	0.33
disease of anatomical entity	0.27	0.12	0.10	0.49	0.26	0.11	0.09	0.48
disease of cellular proliferation	0.25	0.09	0.07	0.42	0.25	0.10	0.08	0.44
disease of mental health	0.22	0.11	0.10	0.43	0.22	0.11	0.10	0.43
disease of metabolism	0.22	0.13	0.10	0.46	0.23	0.14	0.11	0.48
genetic disease	0.23	0.15	0.11	0.4	0.23	0.15	0.11	0.49
syndrome	0.22	0.11	0.11	0.44	0.22	0.11	0.11	0.44

Compared to a few FDA-approved drugs for melanoma in Table 3, we notice that the designed compound combinations had similar exclusive coverage (η_{A-B} and η_{B-A}) as the drug combinations. However, the overlapping and overall coverage ($\eta_{A\cap B}$ and $\eta_{A\cup B}$) were both much higher in FDA-approved drug combinations than the designed. Improvements could be made by training the RL agent longer, as these scores had already been improving during the limited training process under computational restrictions. More improvement can be made by adjusting the network-based reward as well.

Table 4.3: Network-based scores for FDA-approved melanoma drug-combinations.

	η_{A-B}	η_{B-A}	$\eta_{A\cap B}$	$\eta_{A\cup B}$
Dabrafenib + Trametinib	0.05	0.21	0.55	0.81
Encorafenib + Binimetinib	0.21	0.05	0.53	0.86
Vemurafenib + Cobimetinib	0.05	0.27	0.36	0.68

4.5.3 Case studies for specific diseases

4.5.3.1 Experiment Setup

In the third and last stage of RL model training, we start from the stage 2 model and generate drug combinations for a fixed target disease and can choose scaffold libraries specific to the disease. In parallel, we trained the model for 500 iterations (roughly 1 day) to generate 4,000 drug combinations specifically for each of 4 diseases featuring various drug-combination strategies: melanoma, lung cancer, ovarian cancer, and breast cancer. In all cases, we started with the Murcko scaffolds of specific FDA-approved drug combinations to be detailed next.

Melanoma: Different targets in the same pathway. Resistance to BRAF kinase inhibitors is associated with reactivation of the mitogen-activated protein kinase (MAPK) pathway. There is, thus, a phase 1 and 2 trial of combined treatment with Dabrafenib, a selective BRAF inhibitor, and Trametinib, a selective MAPK kinase (MEK) inhibitor. As melanoma is not one of the 299 diseases, we chose a broader neoplasm as an alternative. To compensate for the loss of focus on target disease, we design compound pairs from Murcko scaffolds of Dabrafenib + Trametinib.

Lung and ovarian cancers: Targeting parallel pathways. MAPK and PI3K signaling pathways are parallels important for treating many cancers, including lung and ovarian cancers [284, 285]. Clinical data suggest that dual blockade of these parallel pathways has synergistic effects. Buparlisib (BKM120) and Trametinib (GSK1120212; Mekinist) are as a drug combination therapy are used for the purpose. Specifically, Buparlisib is a potent and highly specific PI3K inhibitor, whereas Trametinib is a highly selective, allosteric inhibitor of MEK1/MEK2 activation and kinase activity [285].

Breast cancer: Reverse resistance. Endocrine therapies, including Fulvestrant, are the primary treatment for hormone receptor-positive breast cancers (80% of breast cancers) [286]. However, they could confer resistance to patients during or after the treatment. A phase 3 study is using Fulvestrant and Palbociclib as a combination therapy to reverse the resistance. Fulvestrant and Palbociclib are targeting different genes in different pathways. Specifically, Fulvestrant targets the estrogen receptor (ER) α in the estrogen signaling pathway, and Palbociclib targets cyclin-dependent kinases 4 and 6 (CDK4 and CDK6) in cell cycle pathway [286].

4.5.3.2 Baseline methods for drug pair combination

Since our proposed method is the first to generate drug combinations for specific diseases, we consider the following baseline methods to compare with: 1) random selection of 1,000 pairs from 8,724 small-molecule drugs in Drugbank [287]; 2) 628 FDA-approved drug combinations curated by [247] for hypertension and cancers (our case studies are on 4 types of cancers); 3) random selection of 1,000 pairs of FDA-approved drugs for the given disease, based on the drug-disease dataset "SCMFDD-L" [288].

4.5.3.3 Designed pairs follow network principles and improve toxicity

We first compare the compound combinations designed by our model and those from the baselines using the network score that reflects the network-based principle. Fig. 4.2(a)–(d) shows that our designed combinations in all 4 cases, with higher network scores in distribution, respected the network principle more than the baselines (including the FDA-approved pairs not necessarily specific for the target disease). The observation is statistically significant with P-values ranging from 6E-74 to 7E-7 (one-sided Kolmogorov-Smirnov [KS] test; Tables 4.4 and 4.5). Such a result is thanks to the network-principled reward we introduced.

Table 4.4: One-sided KS test statistics for comparison of network score distributions

P-values	drugbank	FDA drug comb.	cancer spec.
Melanoma	1.55 e-37	6.05 e-74	1.46 e-15
Breast cancer	7.05 e-7	6.47 e-59	1.02 e-17
Lung cancer	1.48 e-27	2.22 e-43	6.48 e-24
Ovarian cancer	5.29 e-68	1.73 e-90	8.28 e-30

Table 4.5: Comparison of percentage of low and high network score

	Low network score (< 0.2)				High network score (> 0.5)			
	drugbank	drug pair	Cancer	RL	drugbank	drug pair	Cancer	RL
Melanoma	28%	31.6%	23.8%	0.2%	4.6%	0.4%	7.8%	6.2%
Breast cancer	18%	22.6%	15%	2.2%	6%	2%	9.4%	11.6%
Lung cancer	20.6%	21.6%	17.6%	6.4%	15.4%	8%	19.4%	24.6%
Ovarian cancer	50.2%	54.2%	37.6%	11.2%	3.6%	0.4%	10.6%	7.2%

We also examine whether drug combinations designed to follow the network principle could reduce toxicity from drug-drug interactions (DDIs). DDIs are crucial when using drug combinations since they may trigger unexpected pharmacological effects, including adverse drug events (ADEs). We used a deep-learning model DeepDDI [289] with a mean accuracy of 92.4% to predict for each combination the probabilities of 86 types of DDIs (we manually split them into 16 positive and 70 negatives). To summarize the DDIs, we considered both maximum and mean probabilities

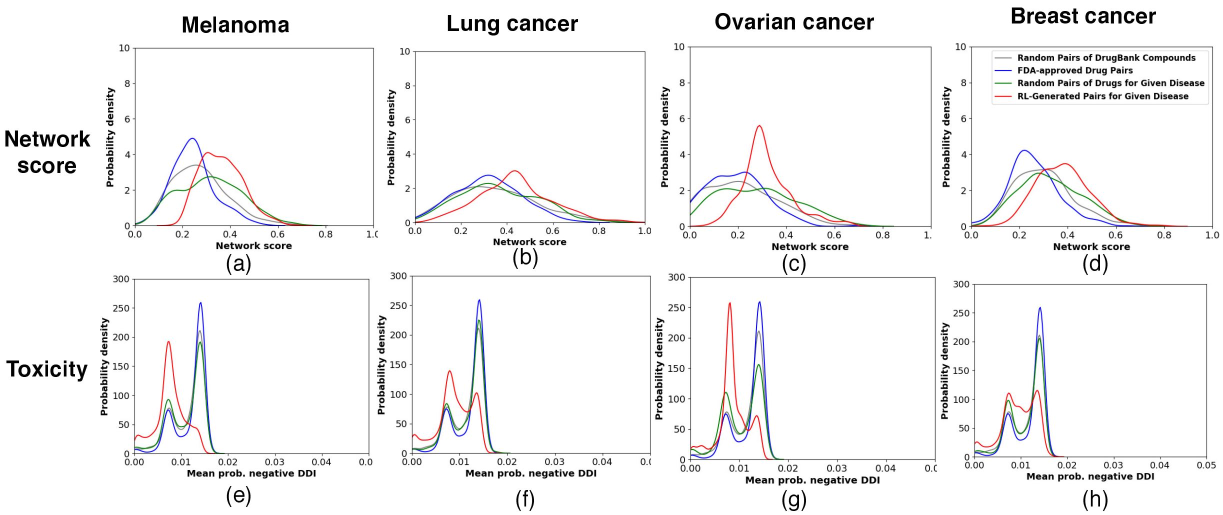


Figure 4.2: Comparison of network score and toxicity of RL-generated pairs of compounds (our proposed method) with three baselines, i.e. random pairs of DrugBank compounds, FDA-approved drug pairs, and random pairs of FDA-approved drugs for four case-study diseases.

of positive or negative ones. Furthermore, we compared those distributions between our designed pairs and baselines in each disease.

Fig. 4.2(e)–(h), using the mean probability among negative DDIs, shows that our compound pairs designed for all 4 diseases were predicted to have fewer chances of toxicity compared to the baselines. One-sided KS tests attested to the statistical significance of the observation as P-values ranged between 2E-166 and 2E-53.

Taken together, Fig. 4.2 suggested that following the network principle in designing drug combinations would help reduce toxicity due to DDIs.

4.5.3.4 Designed pairs reproduce approved polypharmacology strategies

We next examine the DeepAffinity-predicted target genes of our designed pairs and compare them to the polypharmacology strategies outlined in Sec. 4.5.3.1 for each disease. Since improved network scores have been shown to correlate with lower toxicity, we used the scores to filter the 4,000 combinations designed for each disease. Specifically, we retained combinations with network scores above 0.5 and $\eta_{A\cap B}$ below 0.1. These designs are shared along with the codes.

For melanoma, out of 69 combination designs retained, 26% were predicted to jointly cover

BRAF and MEK genes in a complementary way. In other words, one molecule only targets BRAF, and the other only targets MEK, according to our DeepAffinity[19]-predicted IC_{50}, echoing the systems pharmacology strategy of the drug combination of Dabrafenib and Trametinib. There were also other designs that demand further examination and potentially contain novel strategies. All retained designs were predicted to target the MAPK pathway to which BRAF and MEK belong.

For lung and ovarian cancers, the same filtering criteria retained 204 (896) compound combinations designed for ovarian (lung) cancer. As disease modules can be limited, MEK1/2 does not exist in the used modules for ovarian or lung cancer, and a gene-level analysis cannot be performed as the melanoma case. Instead, we performed the pathway-level analysis and found that 50.9% (45.2%) of combination designs for ovarian (lung) cancer were predicted to jointly and complementarily cover the MAPK and PI3K signaling pathways, which echoes the combination of Buparlisib and Trametinib. Moreover, 99.5% (100%) of these retained designs were predicted to jointly target both pathways for ovarian (lung) cancer.

For breast cancer, 77 designed compound-combinations passed the filters. As CDK4/6 does not belong to the breast-cancer module due to the limitation of disease modules used, we again only performed a pathway-level analysis. 9% of the combinations were predicted to jointly and complementarily cover ER-signaling and cell-cycle pathways, as Fulvestrant and Palbociclib do. Also, 74% of the retained combinations jointly cover these pathways. These two portions suggest that many designed combinations were predicted to simultaneously target both pathways (with possible overlapping genes). If we consider PI3K signaling rather than cell cycle pathway for CDK4/6, 15.5% of retained drug combinations were predicted to jointly and complementarily cover estrogen and PI3K signaling pathways, and all of them did jointly.

4.5.3.5 Ablation study for RL-based drug-combination generation

Besides HVGAE for network and disease embedding, two of our novel contributions in RL-based drug set generations were network-principled reward and adversarial reward through GS-WGAN. To assess the effects of these contributions to our model, we performed an ablation study for stage 3 using the case of melanoma. We ablated the originally proposed model in two ways:

removing the network-principled reward or replacing the GS-WGAN adversarial reward with the previously-used GAN reward based on Jenson-Shannon (JS) divergence. Results in Fig. 4.3 suggested that both rewards led to faster initial growth and higher saturation values in network-based scores.

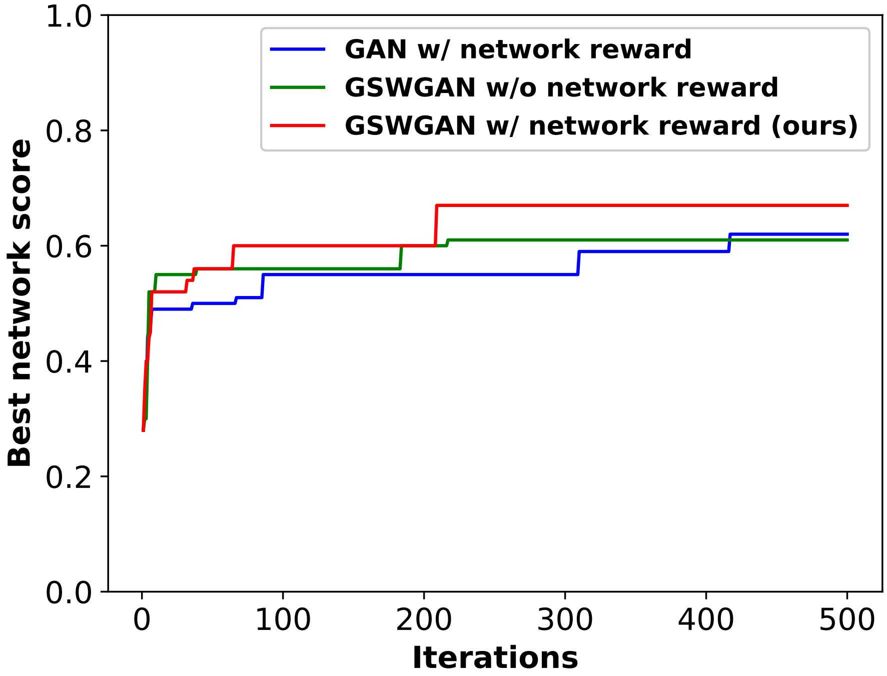


Figure 4.3: Ablation study for RL: Best network scores achieved by three variants of the proposed method over training iterations.

4.6 Conclusion

In response to the need for accelerated and principled drug-combination design, we have recast the problem as graph set generation in a chemically and net-biologically valid environment and developed the first deep generative model with the novel adversarial award and drug-combination award in reinforcement learning for the purpose. We have also designed hierarchical variation graph auto-encoders (HGVAE) to jointly embed domain knowledge such as gene-gene, disease-

disease, gene-disease networks and learn disease representations to be conditioned on in the generative model for disease-specific drug combination. Our results indicate that HGVAE learns integrative gene and disease representations that are much more generalizable and informative than state-of-the-art graph unsupervised-learning methods. The results also indicate that the reinforcement learning model learns to generate drug combinations following a network-based principle thanks to our adversarial and drug-combination rewards. Case studies involving four diseases indicate that drug combinations designed to follow network principles tend to have low toxicity from drug-drug interactions. These designs also encode systems pharmacology strategies echoing FDA-approved drug combinations as well as other potentially promising strategies. As the first generative model for disease-specific drug combination design, our study allows for assessing and following network-based mechanistic hypotheses for efficiently searching the chemical combinatorial space and effectively designing drug combinations.

5. SUMMARY AND CONCLUSIONS

In this dissertation, several approaches have been developed for overcoming drug resistance due to mutation in single protein target and alternative (same) pathway (re-)activation in the molecular network.

In chapter 2, Various computational protein design problems have been formalized. iCFN, an exact and efficient algorithm, has been developed for generic multi-state computational protein design through combinatorial optimization. iCFN has been employed for Mutagenesis analysis, Engineering “acid-switched” antibody-drug conjugate, anticipating resistant mutation, and discovering the underlying biological mechanism. Then, building upon iCFN various multi-state optimal drug design problems for overcoming drug resistance in single protein target have been introduced. Novel lower bounds with theoretical proofs and computational complexity analysis have been developed for each drug design problem. Furthermore, formulations and lower bounds have been extended to optimal drug cocktail design.

In chapter 3, at first, an interpretable semi-supervised deep learning model, named DeepAffinity, has been developed for fast and accurate compound-protein affinity and contact predictions. DeepAffinity has been utilized for binding pocket prediction, transfer learning paradigm for affinity prediction of protein targets with a scarce amount of data, specificity prediction over drugs and proteins. Then, DeepAffinity has been extended to DeepRelations for explainable prediction of binding affinity through contact prediction between compounds and proteins. DeepRelations significantly improved both binding affinity and contact predictions in comparison to the state-of-art model. DeepRelations shows good generalizability over completely new drugs and protein. Its accurate contact prediction has been beneficial for assisting ligand-protein docking. Furthermore, DeepRelation has been utilized in real-life problems such as binding site prediction, structure-activity relations (SAR), and lead optimization.

In chapter 4, a data- and principle-driven deep generative model has been developed for faster, broader, and more in-depth exploration of drug combination space by following the principle underlying FDA approved drug combinations. Firstly, a Hierarchical Variational Graph Auto-Encoders (HVGAE) has been developed for jointly and end-to-end embedding of disease-disease, disease-gene, and gene-gene networks. The novel, attentional pooling mechanism has been developed for taking into account the contribution of each gene for a given disease. HVGAE has shown to embed the disease-disease and gene-gene networks with a significant improvement compared to deterministic and stochastic state-of-art approaches. Secondly, the drug combinations generation problem has been formulated as a graph set generator, and a reinforcement-learning approach has been developed. More specifically, a novel network-based reward based on complementary exposure and generalized sliced Wasserstein generative adversarial networks (GS-WGAN)-based reward has been developed and utilized for more principle and stabler drug combinations generation. The proposed DeepAffinity model has been used as a fast and accurate oracle for compound-protein prediction in calculating network-based reward.

Future directions for multi-state protein and drug design include 1) parallelizing the algorithm and its codes on the architecture of GPU; 2) incorporating more types of constraints seen in applications while allowing for more general objective functions; 3) extending rigid rotamers to continuous rotamers; 4) deriving tighter yet economic bounds under the framework of CFN; 5) exploiting reinforcement learning approaches such as Monte Carlo tree search (MCTS) method for solving protein folding problems efficiently and accurately with probabilistic guarantees; 6) developing decentralized multi-agent solvers for large scale protein and drug design problems.

Future directions for compound-protein affinity prediction include 1) exploiting learning-to-rank framework for more accurate SAR prediction; 2) extending sequential deep learning models (such as RNN and HAN) to graphical deep learning models (such as GCN and GIN) or even sequential-graphical deep learning models (such as graphical transformers) for utilizing the 3D structure of proteins; 3) developing probabilistic deep learning models with built-in uncertainty quantification ability for affinity prediction with probabilistic guarantees; 4) curating specific datasets for high throughput assessment of current state-of-art models and developing novel methods for more accurate lead optimization; 5) incorporating supervising of specific energy terms such as electrostatics and hydrogen bonding for more explainable models; 6) utilizing and expanding current state-of-art models for mutational effect prediction.

Future directions for network-based drug combinations include 1) extending the complementary exposure reward to higher-order interactions in k-drug combinations; 2) utilizing FDA-approved drug combinations in the training process through the GAN framework; 3) extending drug combinations generative model from host diseases only to viruses by attacking host-pathogen network; 4) developing drug combination generative model for disease network reversal.

REFERENCES

[1] W. Toy, H. Weir, P. Razavi, M. Lawson, A. U. Goeppert, A. M. Mazzola, A. Smith, J. Wilson, C. Morrow, W. L. Wong, *et al.*, “Activating esr1 mutations differentially affect the efficacy of er antagonists,” *Cancer discovery*, vol. 7, no. 3, pp. 277–287, 2017.

[2] R. Shafer and D. Vuitton, “Highly active antiretroviral therapy (haart) for the treatment of infection with human immunodeficiency virus type 1,” *Biomedicine & pharmacotherapy*, vol. 53, no. 2, pp. 73–86, 1999.

[3] S. Ramón-García, C. Ng, H. Anderson, J. D. Chao, X. Zheng, T. Pfeifer, Y. Av-Gay, M. Roberge, and C. J. Thompson, “Synergistic drug combinations for tuberculosis therapy identified by a novel high-throughput screen,” *Antimicrobial agents and chemotherapy*, vol. 55, no. 8, pp. 3861–3869, 2011.

[4] P. Sharma and J. P. Allison, “Immune checkpoint targeting in cancer therapy: toward combination strategies with curative potential,” *Cell*, vol. 161, no. 2, pp. 205–214, 2015.

[5] I. Bozic, J. G. Reiter, B. Allen, T. Antal, K. Chatterjee, P. Shah, Y. S. Moon, A. Yaqubie, N. Kelly, D. T. Le, *et al.*, “Evolutionary dynamics of cancer in response to targeted combination therapy,” *elife*, vol. 2, p. e00747, 2013.

[6] J. A. DiMasi, H. G. Grabowski, and R. W. Hansen, “Innovation in the pharmaceutical industry: new estimates of r&d costs,” *Journal of health economics*, vol. 47, pp. 20–33, 2016.

[7] G. A. Van Norman, “Drugs, devices, and the fda: Part 1: an overview of approval processes for drugs,” *JACC: Basic to Translational Science*, vol. 1, no. 3, pp. 170–179, 2016.

[8] C. H. Wong, K. W. Siah, and A. W. Lo, “Estimation of clinical trial success rates and related parameters,” *Biostatistics*, vol. 20, no. 2, pp. 273–286, 2019.

[9] C. L. Ventola, “The antibiotic resistance crisis: part 1: causes and threats,” *Pharmacy and therapeutics*, vol. 40, no. 4, p. 277, 2015.

[10] B. L. Riggs and L. C. Hartmann, "Selective estrogen-receptor modulators—mechanisms of action and application to clinical practice," *New England Journal of Medicine*, vol. 348, no. 7, pp. 618–629, 2003.

[11] W. Toy, Y. Shen, H. Won, B. Green, R. A. Sakr, M. Will, Z. Li, K. Gala, S. Fanning, T. A. King, *et al.*, "Esr1 ligand-binding domain mutations in hormone-resistant breast cancer," *Nature genetics*, vol. 45, no. 12, p. 1439, 2013.

[12] S. W. Fanning, R. Jeselsohn, V. Dharmarajan, C. G. Mayne, M. Karimi, G. Buchwalter, R. Houtman, W. Toy, C. E. Fowler, R. Han, *et al.*, "The serm/serd bazedoxifene disrupts esr1 helix 12 to overcome acquired hormone resistance in breast cancer cells," *Elife*, vol. 7, p. e37161, 2018.

[13] S. A. Madani Tonekaboni, L. Soltan Ghoraie, V. S. K. Manem, and B. Haibe-Kains, "Predictive approaches for drug combination discovery in cancer," *Briefings in bioinformatics*, vol. 19, no. 2, pp. 263–276, 2018.

[14] K. T. Flaherty, J. R. Infante, A. Daud, R. Gonzalez, R. F. Kefford, J. Sosman, O. Hamid, L. Schuchter, J. Cebon, N. Ibrahim, *et al.*, "Combined braf and mek inhibition in melanoma with braf v600 mutations," *New England Journal of Medicine*, vol. 367, no. 18, pp. 1694–1703, 2012.

[15] J.-C. Kaplan and C. Junien, "Genomics and medicine: an anticipation.," *Comptes Rendus de l'Académie des Sciences-Series III-Sciences de la Vie*, vol. 323, no. 12, pp. 1167–1174, 2000.

[16] C. T. Keith, A. A. Borisy, and B. R. Stockwell, "Multicomponent therapeutics for networked systems," *Nature reviews Drug discovery*, vol. 4, no. 1, pp. 71–78, 2005.

[17] M. Karimi and Y. Shen, "icfn: an efficient exact algorithm for multistate protein design," *Bioinformatics*, vol. 34, no. 17, pp. i811–i820, 2018.

[18] M. Karimi, A. Hasanzadeh, and Y. Shen, “Network-principled deep generative models for designing drug combinations as graph sets,” *Bioinformatics*, vol. 36, no. Supplement_1, pp. i445–i454, 2020.

[19] 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.

[20] M. Karimi, D. Wu, Z. Wang, and Y. Shen, “Explainable deep relational networks for predicting compound-protein affinities and contacts,” *arXiv preprint arXiv:1912.12553*, 2019.

[21] H. Hartmann, F. Parak, W. Steigemann, G. A. Petsko, D. R. Ponzi, and H. Frauenfelder, “Conformational substates in a protein: structure and dynamics of metmyoglobin at 80 K,” *Proceedings of the National Academy of Sciences*, vol. 79, pp. 4967–4971, Aug 1982.

[22] H. Frauenfelder, F. Parak, and R. D. Young, “Conformational substates in proteins,” *Annual review of biophysics and biophysical chemistry*, vol. 17, pp. 451–479, 1988.

[23] P. B. Harbury, J. J. Plecs, B. Tidor, T. Alber, and P. S. Kim, “High-resolution protein design with backbone freedom,” *Science*, vol. 282, pp. 1462–1467, Nov 1998.

[24] R. L. Dunbrack and M. Karplus, “Backbone-dependent rotamer library for proteins. Application to side-chain prediction,” *J. Mol. Biol.*, vol. 230, pp. 543–574, Mar 1993.

[25] N. A. Pierce and E. Winfree, “Protein design is np-hard,” *Protein engineering*, vol. 15, no. 10, pp. 779–782, 2002.

[26] D. T. Jones, “De novo protein design using pairwise potentials and a genetic algorithm,” *Protein Science*, vol. 3, no. 4, pp. 567–574, 1994.

[27] A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak, K. Kaufman, P. D. Renfrew, C. A. Smith, W. Sheffler, I. W. Davis, S. Cooper, A. Treuille, D. J. Mandell, F. Richter, Y. E. Ban, S. J. Fleishman, J. E. Corn, D. E. Kim, S. Lyskov, M. Berrondo, S. Mentzer, Z. Popović, J. J. Havranek, J. Karanicolas, R. Das, J. Meiler, T. Kortemme, J. J. Gray, B. Kuhlman, D. Baker, and P. Bradley, “ROSETTA3: an object-oriented software

suite for the simulation and design of macromolecules," *Methods in Enzymology*, vol. 487, pp. 545–574, 2011.

[28] B. Kuhlman, G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard, and D. Baker, "Design of a novel globular protein fold with atomic-level accuracy," *Science*, vol. 302, pp. 1364–1368, Nov 2003.

[29] T. Kortemme, L. A. Joachimiak, A. N. Bullock, A. D. Schuler, B. L. Stoddard, and D. Baker, "Computational redesign of protein-protein interaction specificity," *Nature Structural & Molecular Biology*, vol. 11, pp. 371–379, Apr 2004.

[30] X. I. Ambroggio and B. Kuhlman, "Computational design of a single amino acid sequence that can switch between two distinct protein folds," *Journal of the American Chemical Society*, vol. 128, pp. 1154–1161, Feb 2006.

[31] D. Rothlisberger, O. Khersonsky, A. M. Wollacott, L. Jiang, J. DeChancie, J. Betker, J. L. Gallaher, E. A. Althoff, A. Zanghellini, O. Dym, S. Albeck, K. N. Houk, D. S. Tawfik, and D. Baker, "Kemp elimination catalysts by computational enzyme design," *Nature*, vol. 453, pp. 190–195, May 2008.

[32] L. Jiang, E. A. Althoff, F. R. Clemente, L. Doyle, D. Rothlisberger, A. Zanghellini, J. L. Gallaher, J. L. Betker, F. Tanaka, C. F. Barbas, D. Hilvert, K. N. Houk, B. L. Stoddard, and D. Baker, "De novo computational design of retro-aldol enzymes," *Science*, vol. 319, pp. 1387–1391, Mar 2008.

[33] J. B. Bale, S. Gonen, Y. Liu, W. Sheffler, D. Ellis, C. Thomas, D. Cascio, T. O. Yeates, T. Gonen, N. P. King, and D. Baker, "Accurate design of megadalton-scale two-component icosahedral protein complexes," *Science*, vol. 353, pp. 389–394, Jul 2016.

[34] C. L. Kingsford, B. Chazelle, and M. Singh, "Solving and analyzing side-chain positioning problems using linear and integer programming," *Bioinformatics*, vol. 21, pp. 1028–1039, Apr. 2005.

[35] C. Yanover and Y. Weiss, "Approximate inference and protein-folding," in *Advances in neural information processing systems*, pp. 1457–1464, 2002.

[36] M. Fromer and C. Yanover, "A computational framework to empower probabilistic protein design," *Bioinformatics*, vol. 24, no. 13, pp. i214–i222, 2008.

[37] D. Simoncini, D. Allouche, S. de Givry, C. Delmas, S. Barbe, and T. Schiex, "Guaranteed Discrete Energy Optimization on Large Protein Design Problems," *Journal of chemical theory and computation*, vol. 11, pp. 5980–5989, Dec 2015.

[38] A. R. Leach and A. P. Lemon, "Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm," *Proteins*, vol. 33, pp. 227–239, Nov 1998.

[39] S. M. Lippow, K. D. Wittrup, and B. Tidor, "Computational design of antibody-affinity improvement beyond in vivo maturation," *Nature biotechnology*, vol. 25, pp. 1171–1176, Oct 2007.

[40] 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 in Enzymology*, vol. 523, pp. 87–107, 2013.

[41] Y. Shen, M. D. Altman, A. Ali, M. N. Nalam, H. Cao, T. M. Rana, C. A. Schiffer, and B. Tidor, "Testing the substrate-envelope hypothesis with designed pairs of compounds," *ACS chemical biology*, vol. 8, pp. 2433–2441, Nov 2013.

[42] Y. Shen, M. L. Radhakrishnan, and B. Tidor, "Molecular mechanisms and design principles for promiscuous inhibitors to avoid drug resistance: lessons learned from HIV-1 protease inhibition," *Proteins*, vol. 83, pp. 351–372, Feb 2015.

[43] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[44] J. Desmet, M. D. Maeyer, B. Hazes, and I. Lasters, "The dead-end elimination theorem and its use in protein side-chain positioning," *Nature*, vol. 356, pp. 539–542, Apr. 1992.

[45] J. Desmet, M. D. Maeyer, and I. Lasters, "The "Dead-End Elimination" Theorem: A New Approach to the Side-Chain Packing Problem," in *The Protein Folding Problem and Tertiary Structure Prediction* (K. M. M. Jr and S. M. L. Grand, eds.), pp. 307–337, Birkhäuser Boston, 1994.

[46] R. F. Goldstein, "Efficient rotamer elimination applied to protein side-chains and related spin glasses.," *Biophysical journal*, vol. 66, no. 5, p. 1335, 1994.

[47] D. B. Gordon and S. L. Mayo, "Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem," *Journal of Computational Chemistry*, vol. 19, pp. 1505–1514, Oct. 1998.

[48] N. A. Pierce, J. A. Spriet, J. Desmet, and S. L. Mayo, "Conformational splitting: A more powerful criterion for dead-end elimination," *Journal of Computational Chemistry*, vol. 21, pp. 999–1009, Aug. 2000.

[49] I. Georgiev, R. H. Lilien, and B. R. Donald, "A Novel Minimized Dead-End Elimination Criterion and Its Application to Protein Redesign in a Hybrid Scoring and Search Algorithm for Computing Partition Functions over Molecular Ensembles," in *Research in Computational Molecular Biology* (A. Apostolico, C. Guerra, S. Istrail, P. A. Pevzner, and M. Waterman, eds.), no. 3909 in Lecture Notes in Computer Science, pp. 530–545, Springer Berlin Heidelberg, Apr. 2006.

[50] P. Gainza, K. E. Roberts, and B. R. Donald, "Protein Design Using Continuous Rotamers," *PLoS Computational Biology*, vol. 8, Jan. 2012.

[51] I. Georgiev and B. R. Donald, "Dead-End Elimination with Backbone Flexibility," *Bioinformatics*, vol. 23, pp. i185–i194, July 2007.

[52] M. A. Hallen, D. A. Keedy, and B. R. Donald, “Dead-End Elimination with Perturbations (“DEEPer”): A provable protein design algorithm with continuous sidechain and backbone flexibility,” *Proteins*, vol. 81, pp. 18–39, Jan. 2013.

[53] E. J. Hong, S. M. Lippow, B. Tidor, and T. Lozano-Perez, “Rotamer optimization for protein design through MAP estimation and problem-size reduction,” *Journal of computational chemistry*, vol. 30, pp. 1923–1945, Sep 2009.

[54] K. E. Roberts, P. Gainza, M. A. Hallen, and B. R. Donald, “Fast gap-free enumeration of conformations and sequences for protein design,” *Proteins*, vol. 83, pp. 1859–1877, Oct 2015.

[55] Y. Zhou, Y. Wu, and J. Zeng, “Computational Protein Design Using AND/OR Branch-and-Bound Search,” *Journal of computational biology*, vol. 23, pp. 439–451, Jun 2016.

[56] T. Schiex, H. Fargier, and G. Verfaillie, “Valued Constraint Satisfaction Problems: Hard and Easy Problems,” in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI’95, (San Francisco, CA, USA), pp. 631–637, Morgan Kaufmann Publishers Inc., 1995.

[57] J. Larrosa, “Node and Arc Consistency in Weighted CSP.,” *Proceedings of the National Conference on Artificial Intelligence*, pp. 48–53, 2002.

[58] D. Allouche, S. Traoré, I. André, S. d. Givry, G. Katsirelos, S. Barbe, and T. Schiex, “Computational Protein Design as a Cost Function Network Optimization Problem,” in *Principles and Practice of Constraint Programming* (M. Milano, ed.), no. 7514 in Lecture Notes in Computer Science, pp. 840–849, Springer Berlin Heidelberg, 2012.

[59] S. Traoré, D. Allouche, I. André, S. d. Givry, G. Katsirelos, T. Schiex, and S. Barbe, “A new framework for computational protein design through cost function network optimization,” *Bioinformatics*, vol. 29, pp. 2129–2136, Sept. 2013.

[60] C. Viricel, S. de Givry, T. Schiex, and S. Barbe, "Cost function network-based design of protein–protein interactions: predicting changes in binding affinity," *Bioinformatics*, vol. 34, no. 15, pp. 2581–2589, 2018.

[61] J. Larrosa and T. Schiex, "In the Quest of the Best Form of Local Consistency for Weighted CSP," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, (San Francisco, CA, USA), pp. 239–244, Morgan Kaufmann Publishers Inc., 2003.

[62] J. Larrosa and T. Schiex, "Solving weighted CSP by maintaining arc consistency," *Artificial Intelligence*, vol. 159, pp. 1–26, Nov. 2004.

[63] S. D. Givry and M. Zytnicki, "Existential arc consistency: Getting closer to full arc consistency in weighted csps," in *In Proc. of the 19 th IJCAI*, pp. 84–89, 2005.

[64] M. C. Cooper, S. de Givry, and T. Schiex, "Optimal soft arc consistency.," in *IJCAI*, vol. 7, pp. 68–73, 2007.

[65] M. Cooper, S. De Givry, M. Sanchez, T. Schiex, and M. Zytnicki, "Virtual Arc Consistency for Weighted CSP," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*, AAAI'08, (Chicago, Illinois), pp. 253–258, AAAI Press, 2008.

[66] H. Nguyen, C. Bessiere, S. De Givry, and T. Schiex, "Triangle-based consistencies for cost function networks," *Constraints*, vol. 22, no. 2, pp. 230–264, 2017.

[67] M. A. Hallen and B. R. Donald, "Comets (Constrained Optimization of Multistate Energies by Tree Search): A Provable and Efficient Algorithm to Optimize Binding Affinity and Specificity with Respect to Sequence," in *Research in Computational Molecular Biology* (T. M. Przytycka, ed.), no. 9029 in Lecture Notes in Computer Science, pp. 122–135, Springer International Publishing, Apr. 2015.

[68] J. J. Havranek and P. B. Harbury, "Automated design of specificity in molecular recognition," *Nature structural biology*, vol. 10, pp. 45–52, Jan 2003.

[69] G. Grigoryan, A. W. Reinke, and A. E. Keating, “Design of protein-interaction specificity gives selective bZIP-binding peptides,” *Nature*, vol. 458, pp. 859–864, Apr 2009.

[70] C. Negron and A. E. Keating, “Multistate protein design using CLEVER and CLASSY,” *Methods in Enzymology*, vol. 523, pp. 171–190, 2013.

[71] A. Leaver-Fay, R. Jacak, P. B. Stranges, and B. Kuhlman, “A generic program for multistate protein design,” *PLoS ONE*, vol. 6, no. 7, p. e20937, 2011.

[72] A. M. Sevy, T. M. Jacobs, J. E. Crowe, and J. Meiler, “Design of Protein Multi-specificity Using an Independent Sequence Search Reduces the Barrier to Low Energy Sequences,” *PLoS computational biology*, vol. 11, p. e1004300, Jul 2015.

[73] P. Loffler, S. Schmitz, E. Hupfeld, R. Sterner, and R. Merkl, “Rosetta:MSF: a modular framework for multi-state computational protein design,” *PLoS computational biology*, vol. 13, p. e1005600, Jun 2017.

[74] C. Yanover, M. Fromer, and J. M. Shifman, “Dead-end elimination for multistate protein design,” *Journal of Computational Chemistry*, vol. 28, pp. 2122–2129, Oct. 2007.

[75] J. Desmet, J. Spriet, and I. Lasters, “Fast and accurate side-chain topology and energy refinement (faster) as a new method for protein structure optimization,” *Proteins: Structure, Function, and Bioinformatics*, vol. 48, no. 1, pp. 31–43, 2002.

[76] F. Eisenmenger, P. Argos, and R. Abagyan, “A method to configure protein side-chains from the main-chain trace in homology modelling,” *Journal of molecular biology*, vol. 231, no. 3, pp. 849–860, 1993.

[77] B. G. Pierce, L. M. Hellman, M. Hossain, N. K. Singh, C. W. Vander Kooi, Z. Weng, and B. M. Baker, “Computational design of the affinity and specificity of a therapeutic T cell receptor,” *PLoS Comput Biol*, vol. 10, no. 2, p. e1003478, 2014.

[78] Y. Shen, “Improved flexible refinement of protein docking in CAPRI rounds 22-27,” *Proteins*, vol. 81, pp. 2129–2136, Dec 2013.

[79] D. Allouche, I. André, S. Barbe, J. Davies, S. De Givry, G. Katsirelos, B. O'Sullivan, S. Prestwich, T. Schiex, and S. Traoré, "Computational protein design as an optimization problem," *Artificial Intelligence*, vol. 212, pp. 59–79, 2014.

[80] C. Savojardo, M. Petrosino, G. Babbi, S. Bovo, C. Corbi-Verge, R. Casadio, P. Fariselli, L. Folkman, A. Garg, M. Karimi, *et al.*, "Evaluating the predictions of the protein stability change upon single amino acid substitutions for the fxn cagi5 challenge," *Human mutation*, 2019.

[81] Y. Cao, Y. Sun, M. Karimi, H. Chen, O. Moronfoye, and Y. Shen, "Predicting pathogenicity of missense variants with weakly supervised regression," *Human mutation*, vol. 40, no. 9, pp. 1579–1592, 2019.

[82] T. A. Knijnenburg, L. Wang, M. T. Zimmermann, N. Chambwe, G. F. Gao, A. D. Cherniack, H. Fan, H. Shen, G. P. Way, C. S. Greene, *et al.*, "Genomic and molecular landscape of dna damage repair deficiency across the cancer genome atlas," *Cell reports*, vol. 23, no. 1, pp. 239–254, 2018.

[83] P. A. Jeggo, L. H. Pearl, and A. M. Carr, "Dna repair, genome stability and cancer: a historical perspective," *Nature Reviews Cancer*, vol. 16, no. 1, p. 35, 2016.

[84] E. C. Friedberg, G. C. Walker, W. Siede, and R. D. Wood, *DNA repair and mutagenesis*. American Society for Microbiology Press, 2005.

[85] B. R. Brooks, C. L. Brooks III, A. D. Mackerell Jr, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, *et al.*, "Charmm: the biomolecular simulation program," *Journal of computational chemistry*, vol. 30, no. 10, pp. 1545–1614, 2009.

[86] Y. Wang, D. Cortez, P. Yazdi, N. Neff, S. J. Elledge, and J. Qin, "Basc, a super complex of brca1-associated proteins involved in the recognition and repair of aberrant dna structures," *Genes & development*, vol. 14, no. 8, pp. 927–939, 2000.

[87] J.-Y. Park, T. R. Singh, N. Nassar, F. Zhang, M. Freund, H. Hanenberg, A. R. Meetei, and P. R. Andreassen, "Breast cancer-associated missense mutants of the palb2 wd40 domain,

which directly binds rad51c, rad51 and brca2, disrupt dna repair," *Oncogene*, vol. 33, no. 40, p. 4803, 2014.

[88] J. E. Meza, P. S. Brzovic, M.-C. King, and R. E. Klevit, "Mapping the functional domains of brca1 interaction of the ring finger domains of brca1 and bard1," *Journal of Biological Chemistry*, vol. 274, no. 9, pp. 5659–5665, 1999.

[89] M. T. Valarmathi, M. Sawhney, S. S. Deo, N. K. Shukla, and S. N. Das, "Novel germline mutations in the brca1 and brca2 genes in indian breast and breast-ovarian cancer families," *Human mutation*, vol. 23, no. 2, pp. 205–205, 2004.

[90] M.-R. Nejadmoghaddam, A. Minai-Tehrani, R. Ghahremanzadeh, M. Mahmoudi, R. Dinarvand, and A.-H. Zarnani, "Antibody-drug conjugates: possibilities and challenges," *Avicenna journal of medical biotechnology*, vol. 11, no. 1, p. 3, 2019.

[91] E. L. Sievers and P. D. Senter, "Antibody-drug conjugates in cancer therapy," *Annual review of medicine*, vol. 64, 2013.

[92] J. M. Lambert and A. Berkenblit, "Antibody–drug conjugates for cancer treatment," *Annual review of medicine*, vol. 69, pp. 191–207, 2018.

[93] M. J. M. Hinrichs and R. Dixit, "Antibody drug conjugates: nonclinical safety considerations," *The AAPS journal*, vol. 17, no. 5, pp. 1055–1064, 2015.

[94] B. E. de Goeij and J. M. Lambert, "New developments for antibody-drug conjugate-based therapeutic approaches," *Current opinion in immunology*, vol. 40, pp. 14–23, 2016.

[95] J. C. Kang, W. Sun, P. Khare, M. Karimi, X. Wang, Y. Shen, R. J. Ober, and E. S. Ward, "Engineering a her2-specific antibody–drug conjugate to increase lysosomal delivery and therapeutic efficacy," *Nature biotechnology*, vol. 37, no. 5, pp. 523–526, 2019.

[96] M. C. Franklin, K. D. Carey, F. F. Vajdos, D. J. Leahy, A. M. De Vos, and M. X. Sliwkowski, "Insights into erbb signaling from the structure of the erbb2-pertuzumab complex," *Cancer cell*, vol. 5, no. 4, pp. 317–328, 2004.

[97] M. H. Olsson, C. R. Søndergaard, M. Rostkowski, and J. H. Jensen, “Propka3: consistent treatment of internal and surface residues in empirical p k a predictions,” *Journal of chemical theory and computation*, vol. 7, no. 2, pp. 525–537, 2011.

[98] E. A. Ariazi, J. L. Ariazi, F. Cordera, and V. C. Jordan, “Estrogen receptors as therapeutic targets in breast cancer,” *Current topics in medicinal chemistry*, vol. 6, no. 3, pp. 181–202, 2006.

[99] 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.

[100] “Uniprot: the universal protein knowledgebase,” *Nucleic acids research*, vol. 45, no. D1, pp. D158–D169, 2017.

[101] N. Andruska, C. Mao, M. Cherian, C. Zhang, and D. J. Shapiro, “Evaluation of a luciferase-based reporter assay as a screen for inhibitors of estrogen-erα-induced proliferation of breast cancer cells,” *Journal of biomolecular screening*, vol. 17, no. 7, pp. 921–932, 2012.

[102] Z. Huang, “A fast clustering algorithm to cluster very large categorical data sets in data mining.,” *DMKD*, vol. 3, no. 8, pp. 34–39, 1997.

[103] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.

[104] A. B. Schrock, D. Pavlick, S. J. Klempner, J. H. Chung, B. Forcier, A. Welsh, L. Young, B. Leyland-Jones, R. Bordoni, R. D. Carvajal, *et al.*, “Hybrid capture–based genomic profiling of circulating tumor dna from patients with advanced cancers of the gastrointestinal tract or anus,” *Clinical Cancer Research*, vol. 24, no. 8, pp. 1881–1890, 2018.

[105] J. Niu, G. Andres, K. Kramer, M. N. Kundranda, R. H. Alvarez, E. Klimant, A. R. Parikh, B. Tan, E. D. Staren, and M. Markman, “Incidence and clinical significance of esr1 mu-

tations in heavily pretreated metastatic breast cancer patients," *OncoTargets and therapy*, vol. 8, p. 3323, 2015.

[106] T. Sacha, "Imatinib in chronic myeloid leukemia: an overview," *Mediterranean journal of hematology and infectious diseases*, vol. 6, no. 1, 2014.

[107] O. G. Ottmann, B. J. Druker, C. L. Sawyers, J. M. Goldman, J. Reiffers, R. T. Silver, S. Tura, T. Fischer, M. W. Deininger, C. A. Schiffer, *et al.*, "A phase 2 study of imatinib in patients with relapsed or refractory philadelphia chromosome–positive acute lymphoid leukemias," *Blood, The Journal of the American Society of Hematology*, vol. 100, no. 6, pp. 1965–1971, 2002.

[108] E. Jabbour, H. Kantarjian, and J. Cortes, "Use of second-and third-generation tyrosine kinase inhibitors in the treatment of chronic myeloid leukemia: an evolving treatment paradigm," *Clinical Lymphoma Myeloma and Leukemia*, vol. 15, no. 6, pp. 323–334, 2015.

[109] T. Hou and R. Yu, "Molecular dynamics and free energy studies on the wild-type and double mutant hiv-1 protease complexed with amprenavir and two amprenavir-related inhibitors: mechanism for binding and drug resistance," *Journal of medicinal chemistry*, vol. 50, no. 6, pp. 1177–1188, 2007.

[110] M. D. Balbas, M. J. Evans, D. J. Hosfield, J. Wongvipat, V. K. Arora, P. A. Watson, Y. Chen, G. L. Greene, Y. Shen, and C. L. Sawyers, "Overcoming mutation-based resistance to antiandrogens with rational drug design," *Elife*, vol. 2, p. e00499, 2013.

[111] S. Panda and Y. Vorobeychik, "Stackelberg games for vaccine design," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1391–1399, International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[112] J. J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. A. Rohl, and D. Baker, "Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations," *Journal of molecular biology*, vol. 331, no. 1, pp. 281–299, 2003.

[113] H. Kamisetty, E. P. Xing, and C. J. Langmead, "Approximating correlated equilibria using relaxations on the marginal polytope.," in *International Conference on Machine Learning*, pp. 1153–1160, 2011.

[114] M. L. Radhakrishnan and B. Tidor, "Optimal drug cocktail design: methods for targeting molecular ensembles and insights from theoretical model systems," *Journal of chemical information and modeling*, vol. 48, no. 5, pp. 1055–1073, 2008.

[115] R. Santos, O. Ursu, A. Gaulton, A. P. Bento, R. S. Donadi, C. G. Bologa, A. Karlsson, B. Al-Lazikani, A. Hersey, T. I. Oprea, and J. P. Overington, "A comprehensive map of molecular drug targets," *Nat Rev Drug Discov*, vol. 16, pp. 19–34, 01 2017.

[116] M. J. Keiser, V. Setola, J. J. Irwin, C. Laggner, A. Abbas, S. J. Hufeisen, N. H. Jensen, M. B. Kuijer, R. C. Matos, T. B. Tran, *et al.*, "Predicting new molecular targets for known drugs," *Nature*, vol. 462, no. 7270, p. 175, 2009.

[117] A. Power, A. C. Berger, and G. S. Ginsburg, "Genomics-enabled drug repositioning and repurposing: insights from an IOM Roundtable activity," *JAMA*, vol. 311, pp. 2063–2064, May 2014.

[118] R. L. Chang, L. Xie, L. Xie, P. E. Bourne, and B. P. Palsson, "Drug off-target effects predicted using structural analysis in the context of a metabolic network model," *PLoS computational biology*, vol. 6, p. e1000938, Sep 2010.

[119] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, "Deeptox: Toxicity prediction using deep learning," *Frontiers in Environmental Science*, vol. 3, p. 80, 2016.

[120] M. K. Gilson and H.-X. Zhou, "Calculation of protein-ligand binding affinities," *Annual review of biophysics and biomolecular structure*, vol. 36, 2007.

[121] A. R. Leach, B. K. Shoichet, and C. E. Peishoff, "Prediction of protein-ligand interactions. Docking and scoring: successes and gaps," *Journal of medicinal chemistry*, vol. 49, pp. 5851–5855, Oct 2006.

[122] Q. U. Ain, A. Aleksandrova, F. D. Roessler, and P. J. Ballester, “Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening,” *Wiley Interdiscip Rev Comput Mol Sci*, vol. 5, no. 6, pp. 405–424, 2015.

[123] I. Wallach, M. Dzamba, and A. Heifets, “Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery,” *arXiv preprint arXiv:1510.02855*, 2015.

[124] J. Gomes, B. Ramsundar, E. N. Feinberg, and V. S. Pande, “Atomic convolutional networks for predicting protein-ligand binding affinity,” *arXiv preprint arXiv:1703.10603*, 2017.

[125] J. Jimenez, M. Skalic, G. Martinez-Rosell, and G. De Fabritiis, “KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks,” *J Chem Inf Model*, vol. 58, pp. 287–296, Feb 2018.

[126] Z. Cang and G. W. Wei, “TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions,” *PLoS computational biology*, vol. 13, p. e1005690, Jul 2017.

[127] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant, “Pubchem: a public information system for analyzing bioactivities of small molecules,” *Nucleic acids research*, vol. 37, no. suppl_2, pp. W623–W633, 2009.

[128] X. Chen, C. C. Yan, X. Zhang, X. Zhang, F. Dai, J. Yin, and Y. Zhang, “Drug-target interaction prediction: databases, web servers and computational models,” *Brief. Bioinformatics*, vol. 17, pp. 696–712, 07 2016.

[129] Y. Wang and J. Zeng, “Predicting drug-target interactions using restricted Boltzmann machines,” *Bioinformatics*, vol. 29, pp. i126–134, Jul 2013.

[130] F. Cheng, Y. Zhou, J. Li, W. Li, G. Liu, and Y. Tang, “Prediction of chemical–protein interactions: multitarget-qsar versus computational chemogenomic methods,” *Molecular BioSystems*, vol. 8, no. 9, pp. 2373–2384, 2012.

[131] H. Yu, J. Chen, X. Xu, Y. Li, H. Zhao, Y. Fang, X. Li, W. Zhou, W. Wang, and Y. Wang, “A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data,” *PloS one*, vol. 7, no. 5, p. e37608, 2012.

[132] Y. Tabei and Y. Yamanishi, “Scalable prediction of compound-protein interactions using minwise hashing,” *BMC systems biology*, vol. 7, no. 6, p. S3, 2013.

[133] Y. Shi, X. Zhang, X. Liao, G. Lin, and D. Schuurmans, “Protein-chemical interaction prediction via kernelized sparse learning svm.,” in *Pacific Symposium on Biocomputing*, pp. 41–52, 2013.

[134] Z. Cheng, S. Zhou, Y. Wang, H. Liu, J. Guan, and Y.-P. P. Chen, “Effectively identifying compound-protein interactions by learning from positive and unlabeled examples,” *IEEE/ACM transactions on computational biology and bioinformatics*, 2016.

[135] K. Tian, M. Shao, Y. Wang, J. Guan, and S. Zhou, “Boosting compound-protein interaction prediction by deep learning,” *Methods*, vol. 110, pp. 64–72, 2016.

[136] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

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

[138] F. Wan and J. Zeng, “Deep learning with feature embedding for compound-protein interaction prediction,” *bioRxiv*, p. 086033, 2016.

[139] M. T. Ribeiro, S. Singh, and C. Guestrin, “"why should i trust you?": Explaining the predictions of any classifier,” in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, (New York, NY, USA), pp. 1135–1144, ACM, 2016.

[140] P. W. Koh and P. Liang, “Understanding black-box predictions via influence functions,” in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and

Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 1885–1894, PMLR, 06–11 Aug 2017.

[141] D. Weininger, “Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules,” *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[142] N. Kalchbrenner and P. Blunsom, “Recurrent continuous translation models,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709, 2013.

[143] Z. Wang, S. Chang, Y. Yang, D. Liu, and T. S. Huang, “Studying very low resolution recognition using deep networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4792–4800, 2016.

[144] T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson, “Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities,” *Nucleic acids research*, vol. 35, no. suppl_1, pp. D198–D201, 2006.

[145] M. Kuhn, C. von Mering, M. Campillos, L. J. Jensen, and P. Bork, “Stitch: interaction networks of chemicals and proteins,” *Nucleic acids research*, vol. 36, no. suppl_1, pp. D684–D688, 2007.

[146] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium, “Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches,” *Bioinformatics*, vol. 31, no. 6, pp. 926–932, 2014.

[147] R. D. Finn, A. Bateman, J. Clements, P. Coggill, R. Y. Eberhardt, S. R. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, E. L. L. Sonnhammer, J. Tate, and M. Punta, “Pfam: the protein families database,” *Nucleic Acids Research*, vol. 42, no. D1, pp. D222–D230, 2014.

[148] R. D. Finn, J. Clements, W. Arndt, B. L. Miller, T. J. Wheeler, F. Schreiber, A. Bateman, and S. R. Eddy, "Hmmer web server: 2015 update," *Nucleic acids research*, vol. 43, no. W1, pp. W30–W38, 2015.

[149] J. Cheng, A. Z. Randall, M. J. Sweredoski, and P. Baldi, "Scratch: a protein structure and structural feature prediction server," *Nucleic acids research*, vol. 33, no. suppl_2, pp. W72–W76, 2005.

[150] C. N. Magnan and P. Baldi, "Sspro/accpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity," *Bioinformatics*, vol. 30, no. 18, pp. 2592–2597, 2014.

[151] S. Wang, W. Li, S. Liu, and J. Xu, "Raptorx-property: a web server for protein structure property prediction," *Nucleic Acids Research*, vol. 44, no. W1, pp. W430–W435, 2016.

[152] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (indrnn): Building A longer and deeper RNN," *CoRR*, vol. abs/1803.04831, 2018.

[153] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," *Advances in NIPS*, 2014.

[154] Z. Xu, S. Wang, F. Zhu, and J. Huang, "Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 285–294, ACM, 2017.

[155] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[156] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning.," in *OSDI*, vol. 16, pp. 265–283, 2016.

[157] Y. Tang, "Tf. learn: Tensorflow's high-level module for distributed machine learning," *arXiv preprint arXiv:1612.04251*, 2016.

[158] V. Khomenko, O. Shyshkov, O. Radyvonenko, and K. Bokhan, "Accelerating recurrent neural network training using sequence bucketing and multi-gpu data parallelization," in *Data Stream Mining & Processing (DSMP), IEEE First International Conference on*, pp. 100–103, IEEE, 2016.

[159] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[160] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[161] D. Britz, A. Goldie, T. Luong, and Q. Le, "Massive Exploration of Neural Machine Translation Architectures," *ArXiv e-prints*, Mar. 2017.

[162] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, pp. 1139–1147, 2013.

[163] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Advances In Neural Information Processing Systems*, pp. 289–297, 2016.

[164] H. Brandstetter, A. Kühne, W. Bode, R. Huber, W. von der Saal, K. Wirthensohn, and R. A. Engh, "X-ray structure of active site-inhibited clotting factor xa implications for drug design and substrate recognition," *Journal of Biological Chemistry*, vol. 271, no. 47, pp. 29988–29992, 1996.

[165] 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, p. 573, 2017.

[166] L. F. Iversen, H. S. Andersen, S. Branner, S. B. Mortensen, G. H. Peters, K. Norris, O. H. Olsen, C. B. Jeppesen, B. F. Lundt, W. Ripka, *et al.*, "Structure-based design of a low

molecular weight, nonphosphorus, nonpeptide, and highly selective inhibitor of protein-tyrosine phosphatase 1b," *Journal of Biological Chemistry*, vol. 275, no. 14, pp. 10300–10307, 2000.

[167] T. A. De Beer, K. Berka, J. M. Thornton, and R. A. Laskowski, "Pdbsum additions," *Nucleic acids research*, vol. 42, no. D1, pp. D292–D296, 2013.

[168] D. J. Huggins, W. Sherman, and B. Tidor, "Rational approaches to improving selectivity in drug design," *Journal of medicinal chemistry*, vol. 55, no. 4, pp. 1424–1444, 2012.

[169] K. Y. Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, and P. Zhang, "Interpretable drug target prediction using deep neural representation.," in *IJCAI*, pp. 3371–3377, 2018.

[170] I. Muegge and M. Rarey, "Small molecule docking and scoring," *Reviews in computational chemistry*, vol. 17, pp. 1–60, 2001.

[171] J. Jiménez, M. Škalič, G. Martínez-Rosell, and G. De Fabritiis, "Kdeep: Protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks," *Journal of chemical information and modeling*, vol. 58, no. 2, pp. 287–296, 2018. PMID: 29309725.

[172] W. Torng and R. B. Altman, "Graph convolutional neural networks for predicting drug-target interactions," *Journal of chemical information and modeling*, vol. 59, no. 10, pp. 4131–4149, 2019.

[173] 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.

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

[175] Q. Feng, E. V. Dueva, A. Cherkasov, and M. Ester, "PADME: A deep learning-based framework for drug-target interaction prediction," *CoRR*, vol. abs/1807.09741, 2018.

[176] X. Li, X. Yan, Q. Gu, H. Zhou, D. Wu, and J. Xu, "Deepchemstable: Chemical stability prediction with an attention-based graph convolution network," *Journal of chemical information and modeling*, vol. 59, no. 3, pp. 1044–1049, 2019.

[177] M. R. Uddin, S. Mahbub, M. S. Rahman, and M. S. Bayzid, "Saint: Self-attention augmented inception-inside-inception network improves protein secondary structure prediction," *bioRxiv*, p. 786921, 2019.

[178] K. McCloskey, A. Taly, F. Monti, M. P. Brenner, and L. J. Colwell, "Using attribution to decode binding mechanism in neural network models for chemistry," *Proceedings of the National Academy of Sciences*, vol. 116, no. 24, pp. 11624–11629, 2019.

[179] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, pp. 2048–2057, 2015.

[180] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Advances in Neural Information Processing Systems*, pp. 3504–3512, 2016.

[181] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?," *Computer Vision and Image Understanding*, vol. 163, pp. 90–100, 2017.

[182] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017.

[183] K. Dill and S. Bromberg, *Molecular driving forces: statistical thermodynamics in biology, chemistry, physics, and nanoscience*. Garland Science, 2012.

[184] A. M. Brzozowski, A. C. Pike, Z. Dauter, R. E. Hubbard, T. Bonn, O. Engström, L. Öhman, G. L. Greene, J.-Å. Gustafsson, and M. Carlquist, "Molecular basis of agonism and antagonism in the oestrogen receptor," *Nature*, vol. 389, no. 6652, p. 753, 1997.

[185] M. Congreve, C. W. Murray, and T. L. Blundell, "Keynote review: Structural biology and drug discovery," *Drug discovery today*, vol. 10, no. 13, pp. 895–907, 2005.

[186] A. Wlodawer and J. Vondrasek, "Inhibitors of hiv-1 protease: a major success of structure-assisted drug design," *Annual review of biophysics and biomolecular structure*, vol. 27, no. 1, pp. 249–284, 1998.

[187] A. Brik and C.-H. Wong, "Hiv-1 protease: mechanism and drug discovery," *Organic & biomolecular chemistry*, vol. 1, no. 1, pp. 5–14, 2003.

[188] F. Yang, M. Du, and X. Hu, "Evaluating explanation without ground truth in interpretable machine learning," *arXiv preprint arXiv:1907.06831*, 2019.

[189] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Advances in Neural Information Processing Systems*, pp. 4967–4976, 2017.

[190] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *European Conference on Computer Vision*, pp. 852–869, Springer, 2016.

[191] P. Battaglia, R. Pascanu, M. Lai, and D. J. Rezende, "Interaction networks for learning about objects, relations and physics," in *Advances in Neural Information Processing Systems*, pp. 4502–4510, 2016.

[192] Y. Hoshen, "Vain: Attentional multi-agent predictive modeling," in *Advances in Neural Information Processing Systems*, pp. 2701–2711, 2017.

[193] 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.

[194] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, and L. Zaslavsky, "Pubchem 2019 update: improved access to chemical data," *Nucleic acids research*, vol. 47, no. D1, pp. D1102–D1109, 2018.

[195] "RDKit: Open-source cheminformatics." `http://www.rdkit.org`. [Online; accessed: april 2019].

[196] R. A. Laskowski, J. Jabłońska, L. Pravda, R. S. Vařeková, and J. M. Thornton, "Pdbsum: Structural summaries of pdb entries," *Protein science*, vol. 27, no. 1, pp. 129–134, 2018.

[197] S. Wang, S. Sun, Z. Li, R. Zhang, and J. Xu, "Accurate de novo prediction of protein contact map by ultra-deep learning model," *PLoS computational biology*, vol. 13, no. 1, p. e1005324, 2017.

[198] I. Lee, J. Keum, and H. Nam, "Deepconv-dti: Prediction of drug-target interactions via deep learning with convolution on protein sequences," *PLoS computational biology*, vol. 15, no. 6, p. e1007129, 2019.

[199] T. H. Trinh, A. M. Dai, M.-T. Luong, and Q. V. Le, "Learning longer-term dependencies in rnns with auxiliary losses," *arXiv preprint arXiv:1803.00144*, 2018.

[200] I. Cortes-Ciriano and A. Bender, "KekuleScope: prediction of cancer cell line sensitivity and compound potency using convolutional neural networks trained on compound images," *Journal of cheminformatics*, vol. 11, p. 41, Jun 2019.

[201] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[202] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.

[203] B. Weisfeiler and A. A. Lehman, "A reduction of a graph to a canonical form and an algebra arising during this reduction," *Nauchno-Technicheskaya Informatsia*, vol. 2, no. 9, pp. 12–16, 1968.

[204] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, 2016.

[205] C. I. Branden and J. Tooze, *Introduction to protein structure*. Garland Science, 2012.

[206] C. Pommié, S. Levadoux, R. Sabatier, G. Lefranc, and M. P. Lefranc, “IMGT standardized criteria for statistical analysis of immunoglobulin V-REGION amino acid properties,” *Journal of molecular recognition*, vol. 17, no. 1, pp. 17–32, 2004.

[207] S. Sirimulla, J. B. Bailey, R. Vegesna, and M. Narayan, “Halogen interactions in protein–ligand complexes: implications of halogen bonding for rational drug design,” *Journal of chemical information and modeling*, vol. 53, no. 11, pp. 2781–2791, 2013.

[208] K. Roy, S. Kar, and R. N. Das, *Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment*. Academic press, 2015.

[209] R. Todeschini and V. Consonni, *Handbook of molecular descriptors*, vol. 11. John Wiley & Sons, 2008.

[210] 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, p. 1046, 2011.

[211] 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.

[212] T. He, M. Heidemeyer, F. Ban, A. Cherkasov, and M. Ester, “Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines,” *Journal of cheminformatics*, vol. 9, no. 1, pp. 1–14, 2017.

[213] M. Thafar, A. B. Raies, S. Albaradei, M. Essack, and V. B. Bajic, “Comparison study of computational prediction tools for drug-target binding affinities,” *Frontiers in Chemistry*, vol. 7, 2019.

[214] T. Pahikkala, S. Okser, A. Airola, T. Salakoski, and T. Aittokallio, “Wrapper-based selection of genetic features in genome-wide association studies through fast matrix operations,” *Algorithms for Molecular Biology*, vol. 7, no. 1, p. 11, 2012.

[215] H. Öztürk, E. Ozkirimli, and A. Özgür, “Widedta: prediction of drug-target binding affinity,” *arXiv preprint arXiv:1902.04166*, 2019.

[216] M. Gönen and G. Heller, “Concordance probability and discriminatory power in proportional hazards regression,” *Biometrika*, vol. 92, no. 4, pp. 965–970, 2005.

[217] O. Trott and A. J. Olson, “AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading,” *Journal of computational chemistry*, vol. 31, pp. 455–461, Jan 2010.

[218] W. P. Feinstein and M. Brylinski, “Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets,” *Journal of cheminformatics*, vol. 7, p. 18, 2015.

[219] O. Trott and A. J. Olson, “Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading,” *Journal of computational chemistry*, vol. 31, no. 2, pp. 455–461, 2010.

[220] E. W. Bell and Y. Zhang, “Dockrmsd: an open-source tool for atom mapping and rmsd calculation of symmetric molecules through graph isomorphism,” *Journal of cheminformatics*, vol. 11, no. 1, pp. 1–9, 2019.

[221] Z. Gaieb, C. D. Parks, M. Chiu, H. Yang, C. Shao, W. P. Walters, M. H. Lambert, N. Nevins, S. D. Bembenek, M. K. Ameriks, and T. Mirzadegan, “D3r grand challenge 3: blind prediction of protein–ligand poses and affinity rankings,” *Journal of computer-aided molecular design*, vol. 33, no. 1, pp. 1–18, 2019.

[222] G. Taubes, “The bacteria fight back,” 2008.

[223] G. Housman, S. Byler, S. Heerboth, K. Lapinska, M. Longacre, N. Snyder, and S. Sarkar, “Drug resistance in cancer: an overview,” *Cancers*, vol. 6, no. 3, pp. 1769–1792, 2014.

[224] E. P. Abraham and E. Chain, “An enzyme from bacteria able to destroy penicillin,” *Nature*, vol. 146, no. 3713, pp. 837–837, 1940.

[225] F. Clavel and A. J. Hance, “Hiv drug resistance,” *New England Journal of Medicine*, vol. 350, no. 10, pp. 1023–1035, 2004.

[226] S. W. Dooley, W. R. Jarvis, W. J. Marione, and D. E. Snider, “Multidrug-resistant tuberculosis,” *Annals of internal medicine*, vol. 117, no. 3, pp. 257–259, 1992.

[227] M. Ghany and T. J. Liang, “Drug targets and molecular mechanisms of drug resistance in chronic hepatitis b,” *Gastroenterology*, vol. 132, no. 4, pp. 1574–1585, 2007.

[228] C. Holohan, S. Van Schaeybroeck, D. B. Longley, and P. G. Johnston, “Cancer drug resistance: an evolving paradigm,” *Nature Reviews Cancer*, vol. 13, no. 10, pp. 714–726, 2013.

[229] G. Chang and C. B. Roth, “Structure of msba from e. coli: a homolog of the multidrug resistance atp binding cassette (abc) transporters,” *Science*, vol. 293, no. 5536, pp. 1793–1800, 2001.

[230] C. M. Lovly and A. T. Shaw, “Molecular pathways: resistance to kinase inhibitors and implications for therapeutic strategies,” *Clinical Cancer Research*, vol. 20, no. 9, pp. 2249–2256, 2014.

[231] T.-C. Chou, “Theoretical basis, experimental design, and computerized simulation of synergism and antagonism in drug combination studies,” *Pharmacological reviews*, vol. 58, no. 3, pp. 621–681, 2006.

[232] E. C. Saputra, L. Huang, Y. Chen, and L. Tucker-Kellogg, “Combination therapy and the evolution of resistance: the theoretical merits of synergism and antagonism in cancer,” *Cancer research*, vol. 78, no. 9, pp. 2419–2431, 2018.

[233] N. Singh and P. J. Yeh, “Suppressive drug combinations and their potential to combat antibiotic resistance,” *The Journal of antibiotics*, vol. 70, no. 11, p. 1033, 2017.

[234] R. S. Bohacek, C. McMartin, and W. C. Guida, “The art and practice of structure-based drug design: A molecular modeling perspective,” *Medicinal Research Reviews*, vol. 16, no. 1, pp. 3–50, 1996.

[235] F. Martínez-Jiménez and M. A. Marti-Renom, “Should network biology be used for drug discovery?,” 2016.

[236] H. Billur Engin, A. Gursoy, R. Nussinov, and O. Keskin, “Network-based strategies can help mono-and poly-pharmacology drug discovery: a systems biology view,” *Current pharmaceutical design*, vol. 20, no. 8, pp. 1201–1207, 2014.

[237] R. Darnag, E. M. Mazouz, A. Schmitzer, D. Villemin, A. Jarid, and D. Cherqaoui, “Support vector machines: development of qsar models for predicting anti-hiv-1 activity of tibo derivatives,” *European journal of medicinal chemistry*, vol. 45, no. 4, pp. 1590–1597, 2010.

[238] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, “Random forest: a classification and regression tool for compound classification and qsar modeling,” *Journal of chemical information and computer sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[239] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, “Deeptox: toxicity prediction using deep learning,” *Frontiers in Environmental Science*, vol. 3, p. 80, 2016.

[240] M. Popova, O. Isayev, and A. Tropsha, “Deep reinforcement learning for de novo drug design,” *Science advances*, vol. 4, no. 7, p. eaap7885, 2018.

[241] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec, “Graph convolutional policy network for goal-directed molecular graph generation,” in *Advances in Neural Information Processing Systems*, pp. 6410–6421, 2018.

[242] A. Zhavoronkov, Y. A. Ivanenkov, A. Aliper, M. S. Veselov, V. A. Aladinskiy, A. V. Aladinskaya, V. A. Terentiev, D. A. Polykovskiy, M. D. Kuznetsov, A. Asadulaev, *et al.*, “Deep learning enables rapid identification of potent ddr1 kinase inhibitors,” *Nature biotechnology*, vol. 37, no. 9, pp. 1038–1040, 2019.

[243] T.-C. Chou, “Drug combination studies and their synergy quantification using the chou-talalay method,” *Cancer research*, vol. 70, no. 2, pp. 440–446, 2010.

[244] K. Pang, Y.-W. Wan, W. T. Choi, L. A. Donehower, J. Sun, D. Pant, and Z. Liu, “Combinatorial therapy discovery using mixed integer linear programming,” *Bioinformatics*, vol. 30, no. 10, pp. 1456–1463, 2014.

[245] K. Preuer, R. P. Lewis, S. Hochreiter, A. Bender, K. C. Bulusu, and G. Klambauer, “Deep-synergy: predicting anti-cancer drug synergy with deep learning,” *Bioinformatics*, vol. 34, no. 9, pp. 1538–1546, 2017.

[246] M. P. Menden, D. Wang, M. J. Mason, B. Szalai, K. C. Bulusu, Y. Guan, T. Yu, J. Kang, M. Jeon, R. Wolfinger, *et al.*, “Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen,” *Nature communications*, vol. 10, no. 1, pp. 1–17, 2019.

[247] F. Cheng, I. A. Kovács, and A.-L. Barabási, “Network-based prediction of drug combinations,” *Nature communications*, vol. 10, no. 1, p. 1197, 2019.

[248] J. Menche, A. Sharma, M. Kitsak, S. D. Ghiassian, M. Vidal, J. Loscalzo, and A.-L. Barabási, “Uncovering disease-disease relationships through the incomplete interactome,” *Science*, vol. 347, no. 6224, p. 1257601, 2015.

[249] V. Matys, E. Fricke, R. Geffers, E. Gößling, M. Haubrock, R. Hehl, K. Hornischer, D. Karas, A. E. Kel, O. V. Kel-Margoulis, *et al.*, “Transfac®: transcriptional regulation, from patterns to profiles,” *Nucleic acids research*, vol. 31, no. 1, pp. 374–378, 2003.

[250] T. Rolland, M. Taşan, B. Charloteaux, S. J. Pevzner, Q. Zhong, N. Sahni, S. Yi, I. Lemmens, C. Fontanillo, R. Mosca, *et al.*, “A proteome-scale map of the human interactome network,” *Cell*, vol. 159, no. 5, pp. 1212–1226, 2014.

[251] B. Aranda, P. Achuthan, Y. Alam-Faruque, I. Armean, A. Bridge, C. Derow, M. Feuermann, A. Ghanbarian, S. Kerrien, J. Khadake, *et al.*, “The intact molecular interaction database in 2010,” *Nucleic acids research*, vol. 38, no. suppl_1, pp. D525–D531, 2010.

[252] A. Ceol, A. Chatr Aryamontri, L. Licata, D. Peluso, L. Briganti, L. Perfetto, L. Castagnoli, and G. Cesareni, “Mint, the molecular interaction database: 2009 update,” *Nucleic acids research*, vol. 38, no. suppl_1, pp. D532–D539, 2010.

[253] C. Stark, B.-J. Breitkreutz, A. Chatr-Aryamontri, L. Boucher, R. Oughtred, M. S. Livstone, J. Nixon, K. Van Auken, X. Wang, X. Shi, *et al.*, “The biogrid interaction database: 2011 update,” *Nucleic acids research*, vol. 39, no. suppl_1, pp. D698–D704, 2010.

[254] T. Keshava Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, *et al.*, “Human protein reference database—2009 update,” *Nucleic acids research*, vol. 37, no. suppl_1, pp. D767–D772, 2009.

[255] D.-S. Lee, J. Park, K. Kay, N. A. Christakis, Z. N. Oltvai, and A.-L. Barabási, “The implications of human metabolic network topology for disease comorbidity,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 29, pp. 9880–9885, 2008.

[256] A. Ruepp, B. Waegele, M. Lechner, B. Brauner, I. Dunger-Kaltenbach, G. Fobo, G. Frishman, C. Montrone, and H.-W. Mewes, “Corum: the comprehensive resource of mammalian protein complexes—2009,” *Nucleic acids research*, vol. 38, no. suppl_1, pp. D497–D501, 2010.

[257] P. V. Hornbeck, J. M. Kornhauser, S. Tkachev, B. Zhang, E. Skrzypek, B. Murray, V. Latham, and M. Sullivan, “Phosphositeplus: a comprehensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse,” *Nucleic acids research*, vol. 40, no. D1, pp. D261–D270, 2012.

[258] M. Kanehisa *et al.*, “The kegg database,” in *Novartis Foundation Symposium*, pp. 91–100, Wiley Online Library, 2002.

[259] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, *et al.*, “Biopython: freely available python tools

for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, 2009.

[260] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[261] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, "Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders," *Nucleic acids research*, vol. 33, no. suppl_1, pp. D514–D517, 2005.

[262] A. Mottaz, Y. L. Yip, P. Ruch, and A.-L. Veuthey, "Mapping proteins to disease terminologies: from uniprot to mesh," in *BMC bioinformatics*, vol. 9, p. S3, BioMed Central, 2008.

[263] E. M. Ramos, D. Hoffman, H. A. Junkins, D. Maglott, L. Phan, S. T. Sherry, M. Feolo, and L. A. Hindorff, "Phenotype–genotype integrator (phegeni): synthesizing genome-wide association study (gwas) data with existing genomic resources," *European Journal of Human Genetics*, vol. 22, no. 1, pp. 144–147, 2014.

[264] F. B. Rogers, "Medical subject headings.," *Bulletin of the Medical Library Association*, vol. 51, no. 1, pp. 114–116, 1963.

[265] A. I. Su, T. Wiltshire, S. Batalov, H. Lapp, K. A. Ching, D. Block, J. Zhang, R. Soden, M. Hayakawa, G. Kreiman, *et al.*, "A gene atlas of the mouse and human protein-encoding transcriptomes," *Proceedings of the National Academy of Sciences*, vol. 101, no. 16, pp. 6062–6067, 2004.

[266] X. Zhou, J. Menche, A.-L. Barabási, and A. Sharma, "Human symptoms–disease network," *Nature communications*, vol. 5, no. 1, pp. 1–10, 2014.

[267] C. A. Hidalgo, N. Blumm, A.-L. Barabási, and N. A. Christakis, "A dynamic network approach for the study of human phenotypes," *PLoS computational biology*, vol. 5, no. 4, 2009.

[268] A. P. Davis, C. J. Grondin, R. J. Johnson, D. Sciaky, R. McMorran, J. Wiegers, T. C. Wiegers, and C. J. Mattingly, "The comparative toxicogenomics database: update 2019," *Nucleic acids research*, vol. 47, no. D1, pp. D948–D954, 2019.

[269] L. M. Schriml, C. Arze, S. Nadendla, Y.-W. W. Chang, M. Mazaitis, V. Felix, G. Feng, and W. A. Kibbe, "Disease ontology: a backbone for disease semantic integration," *Nucleic acids research*, vol. 40, no. D1, pp. D940–D946, 2012.

[270] P. Das, M. D. Delost, M. H. Qureshi, D. T. Smith, and J. T. Njardarson, "A survey of the structures of us fda approved combination drugs," *Journal of medicinal chemistry*, vol. 62, no. 9, pp. 4265–4311, 2018.

[271] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[272] A. Hasanzadeh, E. Hajiramezanali, K. Narayanan, N. Duffield, M. Zhou, and X. Qian, "Semi-implicit graph variational auto-encoders," in *Advances in Neural Information Processing Systems*, pp. 10711–10722, 2019.

[273] E. Hajiramezanali, A. Hasanzadeh, K. Narayanan, N. Duffield, M. Zhou, and X. Qian, "Variational graph recurrent neural networks," in *Advances in Neural Information Processing Systems*, pp. 10700–10710, 2019.

[274] J. J. Irwin and B. K. Shoichet, "Zinc- a free database of commercially available compounds for virtual screening," *Journal of chemical information and modeling*, vol. 45, no. 1, pp. 177–182, 2005.

[275] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[276] M. Karimi, G. Veni, and Y.-Y. Yu, "Illegible text to readable text: An image-to-image transformation using conditional sliced wasserstein adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 552–553, 2020.

[277] M. Karimi, S. Zhu, Y. Cao, and Y. Shen, "De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks (gcwgan)," *bioRxiv*, p. 769919, 2019.

[278] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, pp. 5767–5777, 2017.

[279] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. K. Rohde, "Generalized sliced wasserstein distances," *arXiv preprint arXiv:1902.00434*, 2019.

[280] G. Beylkin, "The inversion problem and applications of the generalized radon transform," *Communications on pure and applied mathematics*, vol. 37, no. 5, pp. 579–599, 1984.

[281] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[282] 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*, pp. 855–864, 2016.

[283] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

[284] D. Day and L. L. Siu, "Approaches to modernize the combination drug development paradigm," *Genome medicine*, vol. 8, no. 1, p. 115, 2016.

[285] P. L. Bedard, J. Tabernero, F. Janku, Z. A. Wainberg, L. Paz-Ares, J. Vansteenkiste, E. Van Cutsem, J. Pérez-García, A. Stathis, C. D. Britten, *et al.*, "A phase ib dose-escalation study of the oral pan-pi3k inhibitor buparlisib (bkm120) in combination with the oral mek1/2 inhibitor trametinib (gsk1120212) in patients with selected advanced solid tumors," *Clinical Cancer Research*, vol. 21, no. 4, pp. 730–738, 2015.

[286] N. C. Turner, J. Ro, F. André, S. Loi, S. Verma, H. Iwata, N. Harbeck, S. Loibl, C. Huang Bartlett, K. Zhang, *et al.*, "Palbociclib in hormone-receptor–positive advanced breast cancer," *New England Journal of Medicine*, vol. 373, no. 3, pp. 209–219, 2015.

[287] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, *et al.*, "Drugbank 5.0: a major update to the drugbank database for 2018," *Nucleic acids research*, vol. 46, no. D1, pp. D1074–D1082, 2018.

[288] W. Zhang, X. Yue, W. Lin, W. Wu, R. Liu, F. Huang, and F. Liu, "Predicting drug-disease associations by using similarity constrained matrix factorization," *BMC bioinformatics*, vol. 19, no. 1, pp. 1–12, 2018.

[289] J. Y. Ryu, H. U. Kim, and S. Y. Lee, "Deep learning improves prediction of drug–drug and drug–food interactions," *Proceedings of the National Academy of Sciences*, vol. 115, no. 18, pp. E4304–E4311, 2018.