BIOMOLECULAR LANGUAGE PROCESSING FOR DRUG-TARGET AFFINITY PREDICTION

by

Rıza Özçelik

B.S., Computer Engineering, Boğaziçi University, 2018

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2022

ACKNOWLEDGEMENTS

"The people that you work with, are just... when you get down to it... your very best friends. They say, on your deathbed, you never wish you spent more time at the office. But I will. Gotta be a lot better than a deathbed! I actually don't understand deathbeds. I mean, who would buy that?"

Michael Scott, The Office (US)

Which chapter of a thesis is to read first? Most skip the preambles and start with the introduction and conclusion. I start with acknowledgements. Because in every thesis worth to read, a story worth to listen is hidden in the acknowledgements. You are here, so you must be alike. Welcome aboard dear visitor! My first thanks are to you – thank you for your interest in my story and my acknowledgements!

My story dates back to a meeting in the summer of 2015. In that meeting, I met someone that I was delighted to meet: I left the meeting room as hopping, not walking. After seven years and countless meetings, that "hoppy" feeling is still as vivid as it was that day. Whom am I talking about? My thesis supervisor, Arzucan Özgür, of course. I thank her for everything she HAS and has NOT done for the last seven years. I would need another 100 pages to properly list and honor her contributions to my academic and personal life. Instead, I write this one paragraph as a placeholder for those pages. She prefers short texts anyway.

The other excellent company I had during the thesis was my co-supervisor Elif Özkırımlı. I thank her for all the energy, joy, and "chemistry" she brought into my life. I am also grateful to her for always believing in me and my potential. I still can't explain how it is possible that we have seen each other only ∼5 times in the physical world, and yet we are this close. I think it is a kind of magic.

One of many things my supervisors excel in is creating teams. That is how I

enjoyed the friendship and fellowship of my lab-mates. First and foremost, I thank Hakime Öztürk for being a friend and sharing her knowledge on drug-target affinity prediction (and everything) with no reservation. Her contributions to this thesis are invaluable. I ended up having similar responsibilities for Berk Atıl and Alperen Bağ, whom I am grateful for beinon DebiasedDTA they would nail every task they overtook.

I also thank Emrah Budur and Arda Çelebi, my two other lab-mates, for perfectly demonstrating that outstanding scientists can be humble. I thank Gönül Aycı, Onur Güngör, İlknur Karadeniz, and Abdullatif Köksal, too, for creating an office I looked forward to coming everyday. TabiLab Rulez!

I came across non-TabiLAB amazing researchers along the way, too. I thank Asu Büşra Temizer for being "the domain expert" in our studies and bringing her inspiring passion into every meeting. I also thank Taha Koulani for his help with drawing chemicals and Nilgün Lütfiye Karalı for helping me with incredible kindness and patience any time I needed. I gratefully acknowledge TUBITAK-BIDEB 2210-A scholarship program and the financial support by TUBITAK ARDEB - 119E133, which enabled our collaboration.

I also thank Öznur Taştan for our joyful scientific discussions in NDAL group and HIBIT, and her contributions to my thesis as a jury member. I am grateful to Hüseyin Birkan Yılmaz as a jury member, too. Plus, I will never forget the CMPE250 we lectured together and I will always keep that section of my life close to my heart.

I had the best set of academic fellows, for sure, but even that wasn't enough at times to survive. Those were the times I turned to my family. I thank my siblings Esra Özçelik, Esma Özçelik, and Sultan Murat Özçelik for facing the challenges in life before me and creating a shield that covered me. I thank my mother Münevver Özçelik for being the living statue of endurance and tenacity. Remembering and seeing her emotional strength pushes me to keep going. I also thank my father Mustafa Özçelik for continuously encouraging me to study as hard as I can and set my own

destiny. I promised him 13 years ago that I would study in the best department in Turkey. Well, I devote this thesis to him and seal the deal.

I have thanked my advisors, colleagues, co-authors, friends, and family. There is one person left, though, and she has been all. Last but definitely not least, I thank Selen Parlar, my high school classmate, friend, deskmate, colleague, labmate, scholarship-fellow, co-worker, co-author, best friend, and eventually, my wife. I thank her for the wonderful person she has been and the person she turned me into. I thank her for the meaning she brings into my life and for being there for me, no matter what. The day I met her is the luckiest day of my life.

ABSTRACT

BIOMOLECULAR LANGUAGE PROCESSING FOR DRUG-TARGET AFFINITY PREDICTION

Finding high-affinity protein-chemical pairs is a prominent stage of the drug discovery pipeline. However, the number of available proteins and chemicals forms an experimentally insurmountable combination space and necessitates computational approaches. Drug-target affinity prediction models come into play here and rapidly highlight the high-affinity pairs. This thesis introduces state-of-the-art drug-target affinity prediction models and training strategies to facilitate drug discovery studies. The introduced approaches leverage biomolecular language processing techniques which interpret the chemicals and proteins as documents formed in biomolecular languages. The units of bimolecular languages, named biomolecular words, are discovered in large corpora and pharmacologically verified as meaningful substructures. The biomolecular words are used to develop a novel drug-target affinity prediction framework: ChemBoost. ChemBoost models leverage the biomolecule word-driven representations and achieve state-of-the-art prediction performance. The experiments also demonstrate that unseen biomolecules challenge all drug-target affinity prediction models and reveal a generalizability problem. A language-inspired model training framework, DebiasedDTA, is introduced to target the problem. The evaluations indicate that DebiasedDTA boosts models on seen and unseen biomolecules, especially when the target pair is dissimilar to training biomolecules. ChemBoost and DebiasedDTA are published as an open-source python package, pydta.

ÖZET

İLAÇ-HEDEF BAĞLILIK İLGİSİ TAHMİNİ İÇİN BİYOMOLEKÜLER DİL İŞLEME

Yüksek bağlılık ilgisi gösteren protein-kimyasal çiftlerinin tespiti ilaç keşfinin önemli bir adımıdır. Ancak, mevcut protein ve kimyasal sayısı deneysel olarak taranamayacak bir kombinasyon uzayı oluşturmakta ve hesaplamalı yöntemler gerektirmektedir. Bu aşamada ilaç-hedef bağlılık ilgisi tahmini modelleri sahne alır ve yüksek bağlılık ilgisi gösteren çiftleri hızla tespit ederler. Bu tez, en üst düzey başarımlı ilaç-hedef bağlılık ilgisi tahmini modelleri ve model eğitim stratejileri önerir. Önerilen yaklaşımlar protein ve kimyasal dizilerini biyomoleküler dildeki dökümanlar olarak gören biyomoleküler dil işleme tekniklerini kullanırlar. Biyomoleküler dilin birimleri, veya biyomoleküler kelimeler, büyük biyomolekül derlemlerinde keşfedilmiştir ve farmakolojik olarak değerli bulunmuştur. Biyomoleküler kelimeler özgün bir ilaç-hedef bağlılık tahmini sistemi, ChemBoost, geliştirmek için kullanılmıştır. ChemBoost biyomoleküler kelime tabanlı vektör temsilleri sayesinde en üst düzey başarıma ulaşmıştır. Deneyler ayrıca eğitim kümesinde olmayan biyomoleküllerin bütün ilaç-hedef bağlılık ilgisi tahmini modellerini zorladığını göstermiştir. Bu probleme çözüm olarak, doğal dil işlemeden ilham alan bir model eğitim stratejisi, DebiasedDTA, geliştirilmiştir. Değelendirmeler DebiasedDTA stratejisinin tahmin modellerini hem eğitim kümesininde bulunan hem de bulunmayan biyomoleküllerde güçlendirdiğini göstermiştir. ChemBoost ve DebiasedDTA pydta adında açık kaynak kodlu bir python kütüphanesi olarak yayımlanmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\vec{i}$	Importance coefficients
$P(L)$	The set of proteins with a reported affinity with ligand L
$SW(x, y)$	Smith-Waterman score of protein x and y
$\bar{y}$	Mean of the values in y
$\vec{w_e}$	Training weights in epoch e
$\in$	Set membership
$\sum$	Summation symbol

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
100D	One Hundred Dimensional
BoW	Bag of Words
BPE	Byte Pair Encoding
CI	Concordance Index
DTA	Drug-Target Affinity
DTI	Drug-Target Interaction
LM	Language Model
MACCS	Molecular Access System
MSE	Mean Squared Error
MSS	Maximum Sequence Similarity
NLP	Natural Language Processing
OOV	Out of Vocabulary
RMSE	Root Mean Squared Error
SMILES	Simplified Molecular Input Entry Specification
SW	Smith-Waterman
ULM	Unigram Language Model

1. INTRODUCTION

1.1. The Perspective

Drugs are chemical substances that interact with a molecular structure in a living organism and demonstrate clinical effects [1]. The targeted molecular structure is often a protein, whose function is linked with the desired clinical results and can be modulated by a "binding" chemical [1, 2]. The binding chemical is called a ligand and the interaction strength between the ligand and the protein is named affinity score. The higher the affinity score between a protein and a ligand, the faster the ligand occupies the proteins in the environment, which is a favorable property for drug candidates. Unfortunately, discovering promising drug candidates, and thus drugs, to remedy the target protein function is time-consuming and expensive. The entire drug discovery pipeline, including the target identification phase, pre-clinical and clinical studies, takes more than a decade and billions of US Dollars [3].

Finding chemicals that bind to the target protein with high affinity is a prominent stage of the drug discovery pipeline. Researchers design chemicals that would bind to the target and measure their binding strength in the lab, until they discover promising protein-ligand pairs. However, the identification of the pairs can take years and stall the entire pipeline. This is where the drug-target affinity (DTA) prediction models come into play. Relying on the known protein-chemical binding affinity measurements, they immediately predict the binding affinity of another protein-chemical pair. Successful predictions can eliminate many unsuccessful wet-lab experiments and speed the drug discovery pipeline.

DTA prediction models often leverage supervised machine learning methods [4–16], because supervised machine learning allows an intuitive formulation of the problem: given a set of protein-chemical pairs as input and their binding affinity as output, what would be the binding affinity of other protein-chemical pairs? In this formulation,

inputs and outputs must be numeric [17–21] and this is already the case for the output, as the affinity score is a scalar value. However, the inputs, proteins and chemicals are physical entities and their effective vectorization is crucial to produce high performance DTA prediction models.

Biomolecules (proteins and chemicals) have different vectorizations with trade-offs, which are linked with their 1D, 2D, and 3D structures. 3D-structure-driven vectorization approaches process the coordinates of atoms and amino-acids in space and are the richest in terms of information. However, 3D structures are frequently unknown and computationally expensive to process [22]. 2D-based vectorization approaches leverage molecular graphs of atoms and bonds and bear less information than in 3D [23]. 1D-based approaches, on the other hand, can encode the same amount of information as in 2D for chemicals [24], carry strong signals regarding the 3D structure of proteins [25], and can outperform other representations in many tasks [26]. Plus, 1D representations are easily available and the simplest to store and compute upon. This is why we rely on 1D representations to develop DTA prediction models in this thesis.

In 1D, chemicals and proteins are strings in which each character is a building block; atoms and bonds for chemicals and amino-acids for proteins. These strings are products of sets of rules that encode structured information about the biomolecule [24]. This is similar to a document in a natural language, where the document is a string of alphanumeric characters and punctuation marks put together through grammar rules. Therefore, we can view chemical and protein strings (or sequences) as documents in chemical and protein languages, biomolecular languages as a whole, and represent them through language processing methodologies. This is what we call "biomolecular language processing".

1.2. Thesis Overview

Biomolecular language processing opens the doors to leverage many successful natural language processing approaches for drug-target affinity prediction, and we ex-

ploit these opportunities in this thesis. First, we examine the language structure of biomolecular sequences to strengthen the foundations of our methodologies. Chapter 3 shows that we can computationally identify chemically meaningful substrings in biomolecular sequences, which we name "biomolecular words". Second, we show in a preliminary study that biomolecular words can empower state-of-the art DTA prediction models. We then computationally and pharmacologically examine the chemical words and find that they present patterns statistically similar to natural languages. The pharmacologic evaluation, on the other hand, reveals that chemical words can be markers of strong binding to protein families.

The language-like attributes and pharmacologic value of chemical words motivate a DTA prediction framework around the chemical language. Chapter 4 introduces ChemBoost, a chemical language-based approach for protein-chemical binding affinity prediction. ChemBoost utilizes distributed chemical word vectors to represent biomolecules and evaluates two chemical word identification methods for DTA prediction. During the efforts to find the limitations of the proposed models in Chapter 4, we find that all DTA models, not only ChemBoost models, struggle to predict affinities between proteins and chemicals when at least one of them is unseen during training. We need DTA prediction models to be successful on these setups too, since novel biomolecules are frequently studied to find alternative treatments to existing ones or cure currently incurable diseases. This motivates Chapter 5.

In Chapter 5, we propose a novel perspective to DTA prediction model training to support models while they predict the affinities between novel biomolecules. We focus on the spurious patterns in the training datasets that misguide the models towards non-generalizable information and propose "DebiasedDTA" to avoid such patterns during training. The experiments show that quantifying and avoiding misguiding information boosts the performance of DTA prediction models on novel biomolecules. We release DebiasedDTA as a python package, which also contains the ChemBoost models, and provide more detail on that in Chapter 6.

1.3. Key Contributions

A summary of the contributions of our work on biomolecular language processing for DTA prediction is provided below.

(i) Novel, simple, and successful DTA prediction models and training strategies are introduced. The methodologies utilize biomolecular language processing methods and therefore are applicable to all biomolecules – even when 2D and 3D structures are unknown.

(ii) A novel chemical language-based DTA prediction framework, ChemBoost, is developed. ChemBoost represents biomolecules through distributed chemical word vectors and performs at the state-of-the-art level (Chapter 4).

(iii) Novel ligand selection approaches are proposed for ligand-centric protein representation. A pipeline is implemented to filter high-affinity protein-ligand pairs in public databases and the protein vectors are bolstered with these pairs. The new approach is more successful than the vanilla approaches (Chapter 4).

(iv) The performance of two different chemical word identification approaches to learn distributed chemical vectors is evaluated for the DTA prediction task (Chapter 4).

(v) A fundamental problem in DTA prediction is demonstrated: current models struggle to predict the affinity score of a protein-chemical pair when at least one of the biomolecules is novel (Chapter 4).

(vi) A novel perspective on DTA prediction model training, DebiasedDTA, is proposed to improve prediction performance on novel protein-chemical pairs. DebiasedDTA is applicable to almost all existing DTA prediction models and boosts prediction performance on seen, unseen, and out-of-dataset biomolecules, regardless of the prediction model architecture (Chapter 5).

(vii) Biomolecule identity- and word-driven dataset biases are studied as two bias sources that challenge DTA prediction models to generalize. Experiments show that eliminating any of the two improves the prediction performance. (Chapter 5).

(viii) The effect of model debiasing on input features is studied. Model debiasing enables learning more from the proteins and decrease the contribution of the phar-

macologically unimportant chemical substructures to model predictions (Chapter 5).

(ix) A python package, pydta, that encapsulates all the proposed methods in the thesis is published. pydta is available in pip repository and is easy to install and use (Chapter 6).

2. BIOMOLECULE REPRESENTATION

2.1. Biomolecular Sequences

2.1.1. Chemicals

Chemicals are substances formed by two or more elements bonded together. They can be represented in 1D, 2D, and 3D, with different rules in each dimension. 3D representation of a chemical specifies the atoms, bonds, stereochemistry and atom coordinates, whereas 2D representation specifies atoms, bonds, and stereochemistry in a molecular graph, omitting the 3D coordinates. Both 2D and 3D representations are information-rich, but introduce complexity for computational processing. 1D expression types aim to overcome the complexity of 2D and 3D representations by encoding the chemical as a sequence while minimizing the information loss.

Simplified Molecular Input Entry Specification (SMILES) [24] syntax is one of the most popular 1D representations for chemicals as it can serialize the 2D molecular graph as a 1D sequence. SMILES syntax defines a set of rules that can encode the chemical's elements, bonds, and stereochemistry as a sequence without information loss. Despite bearing the same amount of information as 2D representations, SMILES strings are simple: the lengths of SMILES sequences for bioactive molecules in the largest chemical database, ChEMBL [27], range from 2 to 2034, 99% of them being shorter than 100 characters. Table 2.1 shows the length distribution of SMILES strings in ChEMBL.

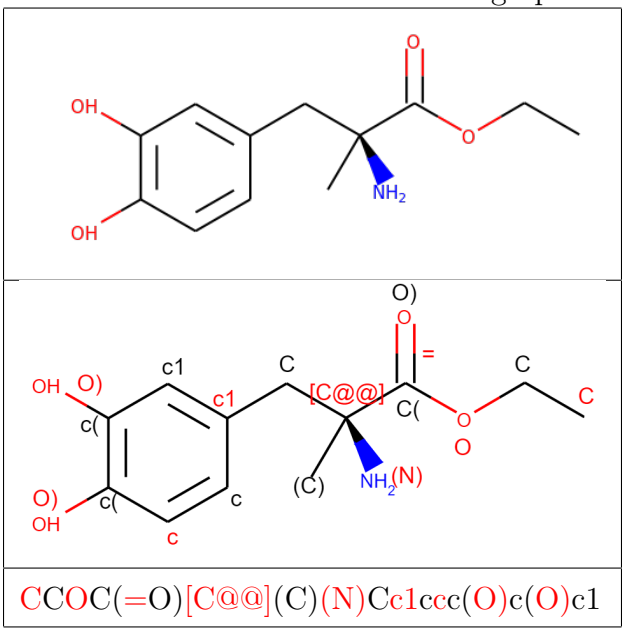
The SMILES syntax encodes the elements by their symbols in the periodic table, while it uses special symbols for bonds: "=" denotes a double bond and "#" denotes a triple bond. Single bonds are implicitly encoded between consecutive elements. SMILES utilizes another symbol to represent stereochemistry. "@" denotes a bond that extends away from the reader while "@@" encodes a bond that extends towards.

Table 2.1. Length distribution of SMILES strings in ChEMBL27.

Length Interval	**Count**	**Cumulative Count**	**Cumulative Ratio**
0 - 100	1823391	1823391	0.93921
100 - 200	92125	1915516	0.98666
200 - 300	15599	1931115	0.99470
300 - 400	4424	1935539	0.99698
400 - 500	2354	1937893	0.99819
500 - 600	1254	1939147	0.99883
600 - 700	1194	1940341	0.99945
700 - 800	605	1940946	0.99976
800 - 900	177	1941123	0.99985
900 - 1000	63	1941186	0.99988
1000 - 1100	46	1941232	0.99991
1100 - 1200	37	1941269	0.99993
1200 - 1300	39	1941308	0.99995
1300 - 1400	59	1941367	0.99998
1400 - 1500	27	1941394	0.99999
1500 - 1600	7	1941401	0.99999
1600 - 1700	3	1941404	1.00000
1700 - 1800	1	1941405	1.00000
1800 - 1900	0	1941405	1.00000
1900 - 2000	2	1941407	1.00000
2000 - 2100	4	1941411	1.00000

A challenge in going from 2D to 1D is representing branches and the SMILES syntax leverages parentheses as solution. In SMILES notation, an opening parenthesis, "(", marks the beginning of a branch and a closing parenthesis, ")", marks the end. Rings are encoded similarly, but with numbers instead of parentheses. Table 2.2 illustrates the 2D structure and SMILES string of *methyldopate* and demonstrates how a SMILES string can be constructed from the molecular graph by starting from its right end. The illustration is annotated and color-coded for demonstration.

Table 2.2. SMILES construction from molecular graph for *methyldopate*.



2.1.2. Proteins

Proteins are the functional units in living organisms that perform vital tasks ranging from respiration to reproduction. Proteins exist in different structures in the cells and their 3D structure affects their function. 3D structure of a protein is observable with X-ray crystallography and nuclear magnetic resonance methods. However, X-ray crystallography of a protein can take up to 5 years and nuclear magnetic resonance is applicable only to small proteins [28]. Consequently, as of December 2021, the largest protein structure database PDB [29] contains 187K structures, whereas there are 225M proteins in UniProt [30].

Proteins can also be represented in 1D as a sequence of their building blocks, amino-acids, which is easily available for all proteins [31]. The amino-acid sequences are accessible through UniProt [30], but they do not contain any explicit information about the 3D structure. However, the 3D structure of a protein can be predicted merely from its sequence in certain cases [25], indicating the value of amino-acid sequences.

Table 2.3. Amino-acid names and symbols

Name	**Symbol**
Alanine	A
Cysteine	C
Aspartic acid	D
Glutamic acid	E
Phenylalanine	F
Glycine	G
Histidine	H
Isoleucine	I
Lysine	K
Leucine	L
Methionine	M
Asparagine	N
Proline	P
Glutamine	Q
Arginine	R
Serine	S
Threonine	T
Valine	V
Tlryptophan	W
Tyrosine	Y

Amino-acid sequences comprise 20 different amino-acids, whose names and symbols, as they appear in the sequences, are displayed in Table 2.3 [31]. Table 2.4 contains further information about the protein sequences and shows that their length distribution has a larger range and greater variance than chemical sequences. This is related to proteins usually being larger molecules than drug-like chemicals.

Table 2.4. Length distribution of amino-acid sequences in UniProt.

Length Interval	**Count**	**Cumulative Count**	**Cumulative Ratio**
0 - 1000	545419	545419	0.96710
1000 - 2000	15809	561228	0.99513
2000 - 3000	1835	563063	0.99839
3000 - 4000	514	563577	0.99930
4000 - 5000	207	563784	0.99967
5000 - 6000	86	563870	0.99982
6000 - 7000	32	563902	0.99988
7000 - 8000	37	563939	0.99994
8000 - 9000	10	563949	0.99996
9000 - 10000	5	563954	0.99997
10000 - 11000	8	563962	0.99998
11000 - 12000	3	563965	0.99999
13000 - 14000	2	563967	0.99999
14000 - 15000	1	563968	0.99999
18000 - 19000	2	563970	1.00000
34000 - 35000	1	563971	1.00000
35000 - 36000	1	563972	1.00000

2.2. Biomolecular Language Unit Discovery

A natural language document can be viewed as a sequence of smaller language units, such as prefixes, words, or phrases. These units are already defined in the structure of the document's language and known by the language speakers. In biomolecular sequences, though, the language units are unknown and need identification. Here we describe two methods to discover biomolecular language units, or as we call them, biomolecular words.

2.2.1. k-mers

A *k-mer* of a sequence is a k-element substring. For a natural language unit, it is a k-letter substring of the unit. For instance, 5-mers of the unit "michael_scott" are {"micha", "ichae", "chael", "hael_", "ael_s", "el_sc", "l_sco", "_scot", "scott"}. For amino-acid sequences, a k-mer consists of k amino-acids, whereas for a SMILES string it comprises k SMILES units.

To find the *k-mers* of a language, a sliding window of length k is traversed over all known sequences of the language. Then all k-mers are unioned, constituting the vocabulary of the language, in which each k-mer is a word. *k-mers* have been proven successful to represent both chemicals [32] and proteins [33].

2.2.2. Byte Pair Encoding

Byte Pair Encoding (BPE) is a compression technique [34] that was adopted to the word segmentation task in natural language processing to discover the words or tokens of a language given a large corpus [35–39]. BPE postulates that frequent subsequences in a large corpus are meaningful language units. As such, given a corpus, BPE first extracts the uni-character vocabulary of the corpus and then computes the frequencies of all two-character subsequences. The algorithm expands its vocabulary with the most frequent subsequence and restarts counting by considering all elements in the vocabulary as a single character. The counting and vocabulary expansion continue until the target vocabulary size (V) is reached. When the algorithm terminates, the vocabulary contains the most frequent V subsequences, which are considered to be the words of the language. Biomolecular sequence processing benefited from BPE too [40–43]; a recent study that showed the effectiveness of BPE identified "bio-words" for the task of protein-protein interaction prediction [44]. Table 2.5 provides the chemical words extracted from *ampicilin* via both methods as an example.

Table 2.5. Example chemical words extracted from the SMILES of *ampicilin* using 8-mers and BPE.

Method	**Chemical Words**
k-mer (i.e. 8-mer)	COc1cc2C, Oc1cc2CC, c1cc2CCN, ..., 3)c2cc1C,)c2cc1Cl
BPE	COc1cc2, CCN=C(, c3ccc(Cl)c(Cl)c3), c2cc1Cl

2.3. Biomolecule Vectorization

Throughout the thesis, we train machine learning models that predict the binding affinity of input protein-chemical pairs. Common to all machine learning models, the inputs must be vectorized during training and prediction. The sections below present methods to vectorize chemicals and proteins.

2.3.1. One-hot Encoding

One-hot encoding is a simple method to vectorize any categorical input, such as words, sequences, or biomolecules. In one-hot encoding, each dimension is reserved for a category in a C dimensional space, where C is the number of unique categories. The input category is then vectorized by setting only its reserved dimension to 1 in the vector and setting the rest of the dimensions to 0. For instance, assume a language with only 4 words: {"dunder", "mifflin", "paper", "company"}. The following is a valid one-hot encoding for the language:

$$\begin{aligned} \text{“dunder”} &: \Big[1,0,0,0\Big], \\ \text{“mifflin”} &: \Big[0,1,0,0\Big], \\ \text{“paper”} &: \Big[0,0,1,0\Big], \\ \text{“company”} &: \Big[0,0,0,1\Big]. \end{aligned}$$

This setting cannot represent out-of-vocabulary (OOV) words by default, though. One-hot encoding overcomes this by reserving an additional dimension for OOV words and setting that dimension to 1 when an OOV word is encountered.

One-hot encoding of biomolecules follows the same approach as language vectorization. Each unique chemical is considered as a category and matched with a dimension. $C + 1$ dimensional vectors are used to represent chemicals, where C is the number of unique chemicals in the training set and one dimension is reserved to vectorize chemicals unavailable in the training set. Therefore, all novel chemicals are represented with the same vector during prediction. The same vectorization algorithm applies to proteins, too.

2.3.2. Bag of Biomolecular Words

Bag-of-Words (BoW) is a document vectorization perspective in natural language processing that interprets documents as unordered collections of words. A vectorization approach in BoW perspective is to use word frequencies and represent each document with a V-dimensional vector such that $\vec{v}_i = f(w_i), \forall i \in \{1, 2, ..., V\}$, where $\vec{v}_i$ is the i^{th} element of the vector; $f(w_i)$ is the normalized frequency of the i^{th} word of the vocabulary in the document; and V is the vocabulary size. The normalized frequency of a word is computed by dividing its count in the document by the number of words in the document. For instance, the phrase "dunder mifflin" is vectorized as $\Big[0.5, 0.5, 0, 0\Big]$, given the vocabulary {"dunder", "mifflin", "paper", "company"}.

BoW is applied to biomolecule vectorization through SMILES and amino-acid sequences. The biomolecular words in these sequences are first identified with any identification method, and then the described vectorization algorithm is applied as is.

2.3.3. Distributed Biomolecular Word Vectors

One-hot encoding and bag of biomolecular words are useful vectorization approaches as they are simple and fast. However, they also have major shortcomings. One of them is they cannot embed the similarity between words in the vector space. For instance, assuming a vocabulary of {"dunder", "mifflin","paper", "company", "scranton", "pennsylvania"}, the cosine similarity between "scranton" and "pennsylvania" is equal to 0, just like the cosine similarity between "scranton" and "paper". In other words, despite "scranton" and "pennsylvania" are semantically more related words than "scranton" and "paper", the cosine similarities are the same. Hence, the semantic relations between words are not reflected in the vector space.

Word2Vec [45] is a breakthrough approach in natural language processing that overcomes this limitation by encoding the semantic relations between words in vector space. Word2Vec assumes that words that frequently appear in similar contexts in a large corpus have higher semantic similarity, where context is defined as a set of words within a window frame. By training a single-layered neural network to predict either the target word given the context or the context given a target word, Word2Vec learns a vector for each word in the vocabulary. In the resulting vector set, the words that frequently occur in similar contexts are close to each other in the vector space. Therefore, cosine similarity between "scranton" and "pennsylvania" is higher than the one between "scranton" and "paper", encoding the semantic relations.

Another advantage of Word2Vec over one-hot encoding and BoW representations is that the dimensionality of Word2Vec models is a model hyper-parameter that mostly ranges between 50 and 500. For one-hot encoding and BoW, the vector dimensionality is determined by the number of words in the language, which can go up to hundreds of thousands. This breakthrough approach had implications in biomolecular language processing as well and gave birth to novel biomolecule representation algorithms, SMILESVec [32] and ProtVec [33].

2.3.3.1. SMILESVec. SMILESVec [32] represents chemicals with distributed chemical word vectors of their SMILES representations. The underlying hypothesis is that similar to natural languages where documents and sentences are composed of words; SMILES strings constitute a domain-specific language composed of chemical words. SMILESVec utilizes Word2vec to learn 100-dimensional (100D) distributed chemical word vectors from a large SMILES corpus, while treating SMILES strings of compounds as sentences constituting chemical words. Chemical words are identified via any language unit identification algorithm such as k-mers and BPE.

When the distributed chemical word vectors are in hand, the SMILESVec of a compound is calculated as

$$\vec{compound} = \frac{\sum_{k=1}^{n} (\vec{cw_k})}{n} \tag{2.1}$$

where

n is equal to the number of chemical words (cw) extracted from the SMILES string of a compound

$(\vec{cw_k})$ represents the 100D embedding of the k^{th} chemical word.

The compound is then described as the average of the vectors of the chemical words in its SMILES representation. SMILESVec has been shown to outperform alternative vector representations in several benchmark tasks [32].

2.3.3.2. ProtVec. ProtVec [33] is a protein vectorization approach that relies merely on amino-acid sequences. Similar to SMILESVec, ProtVec assumes that frequently co-occurring amino-acid subsequences have similar "biological meaning" and should have similar vector representations. ProtVec breaks amino-acid sequences into non-overlapping 3-mers and applies Word2Vec algorithm to learn a 100D distributed vector representation for each 3-mer, or biological word, in the corpus. ProtVec utilizes vector summation to produce protein vectors via biological word vectors and shows that consequent vectors bear biological semantics [33].

2.3.3.3. Ligand-centric Protein Representation. The sequence of a protein determines the protein's folding and function. However, the similarity of protein sequences does not necessarily imply a similarity in protein function, and vice verse [46]. Therefore, proteins with similar sequences might have dissimilar binding behavior, whereas proteins with dissimilar sequences might bind to similar ligands. This encourages protein representations that would explicitly encode the binding behavior in the vector space. The ligand-centric protein representation [32] does exactly that.

Ligand-centric protein representation represents proteins through embeddings of the chemical words in their interacting chemicals [32]. The protein vector is computed by averaging the embedding of the chemical words, where chemical word embeddings of the ligands are obtained via any chemical word embedding algorithm.

2.3.4. Biomolecular Language Model Vectors

Word2Vec transformed natural language processing once and for all and had implications in biomolecule representation. Word2Vec owed it popularity to being very easy to use: a vector for each word was produced, which can be re-used in any further natural language processing tasks [47–50]. On the other hand, using the same vector for a word in every context is also a shortcoming of the model, since words are attributed different meanings in different contexts. To exemplify, consider the word "paper" in the following sentences: (i) "Michael Scott runs the best paper company in the world.", (ii) "The last paper from TabiLab is fascinating!". In (i), "paper" means a sheet to write on, whereas in (ii), it denotes a scientific manuscript. Therefore, the word "paper" is used in different meanings in two sentences and should be vectorized differently to encode the difference in the semantics. Word2Vec cannot do that, inspiring studies on contextualized word embeddings.

Encoding the context is a task in language modeling, where the goal is to predict the next word in a sequence based on the previous words. Recent language models (LMs) [38,51,52] are structured as deep sequence models where word vectors are initial-

ized randomly during training and updated alongside the model weights. Being a deep sequence model (Word2Vec is a single layer neural network), the word embeddings in the subsequent layers are affected by the previous words, and thus contextualized. Furthermore, the overall information in the sentence is encoded in a special token ([CLS]) so that its embedding can be used as the sentence embedding. The contextualized word embeddings of LMs empowered state-of-the-art level performance on many other natural language processing tasks [38, 51, 53–56].

The high performance of LMs in natural language processing inspired modeling studies on biomolecular sequences [57–60]. The protein LMs are shown to learn biologically relevant information such as amino-acid locations in 3D [60] and SMILES language models boosted cheminformatics models in various tasks [57]. To vectorize biomolecules with LMs in this thesis, the SMILES of the chemicals are input to ChemBERTa [61], a chemical language, and amino-acid sequences are input to a protein sequence LM, ProtBERT [59]. The models infer the contextualized representations of the input sequences and output one vector per biomolecule, which are used to represent biomolecules in subsequent operations.

2.3.5. Other Representations

Besides language processing-based vectorization approaches, domain knowledge-driven approaches are used to vectorize the biomolecules in parts of this thesis. These are molecular access system (MACCS) keys [62] and Morgan fingerprints [63] for chemicals and Smith-Waterman [64] for proteins.

2.3.5.1. MACCS Keys. MACCS Keys [62] are binary vectors of 166 dimensions in which each dimension specifies the existence of a chemical pattern in the compound. MACCS keys vectorization approach is publicly available in the popular cheminformatics tool `rdkit` [65].

2.3.5.2. Morgan Fingerprints. Morgan fingerprints [63] are summary vectors that encode paths in the molecular graph of a compound in a vector. Morgan fingerprint computation is an iterative process that assigns identifiers to each atom and updates them with the neighboring identifiers in each iteration. As the number of iterations increases, the size of the substructures encoded in the vector space also increases. In other words, larger numbers of iterations encode global structures, whereas smaller numbers focus on local structures. 2 is a popular choice for this parameter.

2.3.5.3. Smith-Waterman. Smith-Waterman (SW) is a dynamic programming algorithm first developed to find the protein sequence alignment that maximizes the local structure similarity [64] between proteins. SW assigns a similarity score to each possible alignment of two sequences and cleverly finds the one with the highest score. This score is also a measure of protein sequence similarity.

SW represents proteins with respect to their similarity to all proteins as

$$\vec{v_i} = \Big[v_j | NSW(p_i, p_j), \forall j \in [1, P] \Big] \tag{2.2}$$

$$NSW(p_i, p_j) = \frac{SW(p_i, p_j)}{\sqrt{SW(p_i, p_i)}\sqrt{SW(p_j, p_j)}} \tag{2.3}$$

where

v_i is the vector of the ith protein (p_i)

P is the number of proteins in consideration

$NSW(p_i, p_j)$ denotes the normalized SW score of p_i and p_j [66]

$SW(p_i, p_j)$ denotes the SW similarity between p_i and p_j.

While SW incorporates protein similarity and domain knowledge in protein vectors, SW computation is quadratic and computing SW score of all protein pairs is not scalable to large protein sets. In such sets, the dimensionality of protein vectors can

also grow too large to have fast models.

3. BIOMOLECULAR LANGUAGES

We build our work on the hypothesis that there are biomolecular languages and biomolecular sequences are documents composed in these languages. Thus, we can process the biomolecules with language processing techniques. Although there is extensive work in the literature [4, 16, 33, 40, 59, 61, 67] that show the merit of the biomolecular languages hypothesis, we further test and strengthen our main claim in this chapter. In order to test the value of biomolecular language processing, we first identify biomolecular words and then utilize them to build drug-target affinity prediction models. Observing the power of biomolecular words, we dive deeper into chemical vocabularies and test if they statistically resemble the words in natural languages. Last, we collaborate with medicinal chemists and investigate the meanings of chemical words from a pharmacological point of view.

3.1. Discovering Biomolecular Words

We introduced two word discovery methods in the previous chapter, k-mers and Byte Pair Encoding (BPE). Both methods previously showed great performance on different biological and chemical tasks and can be used for the discovery of biomolecular words [32, 33, 43, 44]. A distinction between the two approaches is that BPE allows selecting the vocabulary size during training, whereas k-mers includes all k-mers of the training corpus in the final vocabulary, which might grow too large. This flexibility allows extracting different vocabularies from the same corpus encourages us to utilize BPE to explore biomolecular words in this section.

In addition to BPE, we experiment with the unigram language model (ULM) [68], which slightly modifies BPE to introduce a probabilistic perspective. The advantage of ULM over BPE is that it enables training using an initial vocabulary, which allows us to test more ideas for biomolecular word discovery.

Both BPE and UML require a large number of sequences to find language units. To curate protein sequences, we download all ∼520K reviewed protein sequences from UniProt [30]. To curate a chemical sequences corpus, we download the canonical SMILES representations of ∼2M bioactive chemicals in ChEMBL [27].

Having curated large number of biomolecular sequences, or corpora, we discover biomolecular words via BPE and ULM. We select vocabulary sizes of 8000, 16000, and 32000 and first train BPE on ChEMBL and UniProt corpora, separately. We obtain six biomolecular vocabularies in total.

After BPE, we start discovering biomolecular words with ULM. ULM is interesting to us since it enables integrating additional information into biomolecular vocabularies through start vocabularies. For proteins, we find the most frequent 7000 non-overlapping 3-mers in the corpus and create a starting vocabulary, since non-overlapping 3-mers showed promise as biological words in the literature [32, 33]. For the chemicals, we investigate the impact of incorporating domain knowledge into the vocabularies via BRICS [69], which is a rule-based approach that decomposes compounds into fragments based on chemical bonds. We apply BRICS to ChEMBL corpus and obtain the fragments identified by BRICS. Then, we train ULM on BRICS fragments instead of whole SMILES strings. Same as BPE, we obtain six vocabularies with ULM, whose sizes are 8000, 16000, and 32000.

3.2. Exploring Biomolecular Words for Drug-Target Affinity Prediction

We created 12 biomolecular language vocabularies in the previous section. Would they be useful for DTA prediction, though? Here we seek an empirical answer to this question by utilizing the found vocabularies in a DTA prediction model, DeepDTA [4].

Table 3.1. The performance of biomolecular word discovery methods on BDB dataset.

Chemical Vocab Size	**Protein Vocab Size**	**BPE Score**	**ULM Score**
8000	8000	0.319 (0.011)	0.314 (0.006)
8000	16000	0.302 (0.010)	0.347 (0.076)
8000	32000	0.338 (0.025)	0.313 (0.015)
16000	8000	0.324 (0.022)	0.320 (0.009)
16000	16000	0.326 (0.013)	0.314 (0.009)
16000	32000	0.316 (0.009)	0.330 (0.038)
32000	8000	0.311 (0.016)	0.325 (0.016)
32000	16000	0.325 (0.031)	0.305 (0.011)
32000	32000	0.326 (0.010)	0.312 (0.009)

DeepDTA is a convolutional neural network that predicts the affinity between a protein-chemical pair using their amino-acid and SMILES sequence representations. DeepDTA segments these sequences into characters and apply character-level convolutions instead of using biomolecular words. Here, we modify DeepDTA to segment input sequences into biomolecular words found in the previous section and compare the performance with the original model.

To evaluate the performance of DeepDTA, we train all DeepDTA variants on five different training setups of tge BDB dataset [70] and test the models on the corresponding test sets. We compute the average mean squared errors of the models on the test sets and report the results in Table 3.1. The original DeepDTA model obtained a score of 0.345 on the same setup, with a standard deviation of 0.026.

Table 3.1 shows that, except 8K-16K combination of ULM vocabularies, integrating biomolecular words into DeepDTA lowered the average mean squared error, or in other words, improved prediction performance. This evidences that biomolecular words can enable more informative biomolecule representations than character level models.

Table 3.1 also demonstrates that experimented word identification methods have comparable performance, suggesting that the use of start vocabularies did not improve the models. Here, we favor the use of BPE over ULM as it is a simpler approach with fewer components, and thus is easier to maintain.

3.3. Statistical Analysis of Chemical Words

The previous section displayed that biomolecular words are able to produce informative biomolecule representations. Although this exemplifies the merits of biomolecular language processing, it does not necessarily suggest the language-likeness of identified biomolecular words.

A language-likeness indicator for vocabularies is Zipf's Law [71]. Zipf's Law proposes that in all natural languages, the frequency of a word in a sufficiently large corpus is inversely proportional to its frequency ranking in the corpus. In order to test if the identified vocabularies satisfy Zipf's Law, we select the chemical vocabulary of BPE with size 16000, as it consistently produced a strong performance in the previous section. We then compute the word frequency of all words in ChEMBL corpus we previously curated. Finally, we compute pearsonr statistics of the word frequencies with the series proposed by Zipf's Law, which is $\sum_i 1/i$. The pearsonr statistics is calculated as 0.90.

In order to benchmark the pearsonr statistics of chemical vocabulary, we find three natural language corpora, each of which is similar to ChEMBL in size. These corpora are (i) the first 400K lines of the English Wiki-Text dataset [72], (ii) the first 1M lines of the German Deu-News dataset [73], and (iii) the Turkish NLI-TR dataset [74]. We learn BPE vocabularies of size 16000 on each of these corpora and measure pearsonr statistics with Zipf's Law, as previously. We calculate the results as 0.94 for English, 0.88 for German, and 0.99 for Turkish. The figures show that chemical vocabulary satisfies Zipf's Law as much as natural languages, and thus the identified chemical language statistically resembles natural languages.

3.4. Pharmacologic Evaluation of Chemical Words

The previous sections evaluated the biomolecular words computationally and demonstrated their benefits in the DTA prediction task and resemblance to natural words. These results indicate the value of biomolecular language processing and motivate us to interpret the units of these languages. Similar to statistical analysis, we select the BPE vocabulary of size 16000 and collaborate with medicinal chemists to evaluate the chemical words in this vocabulary from a pharmacological perspective.

For pharmacological evaluation, we computationally discover chemical words that might indicate strong binding to a family of proteins. We call these words "strong binding indicators" and take the following steps to identify them:

(i) List the high-affinity protein-chemical pairs in BDB by using a threshold of $K_d > 7$.

(ii) Cluster the proteins with respect to their families. There turned out to be 80 families in the dataset.

(iii) For each family, find the list of chemicals that strongly bind to at least one protein in the family and extract the chemical words of such chemicals.

(iv) Model the protein families as documents composed of chemical words of their high-affinity ligands. In this model, the set of all families forms a corpus.

(v) Vectorize protein families with tf-idf [75]. During vectorization, omit the chemical words that exist in more than 50 families. This resembles stopword removal in natural language processing.

(vi) Based on the intuition that high tf-idf chemical words are distinctive elements for a document; list 10 chemical words that have the highest tf-idf score for each family. Call these 10 words as "strong binding indicators".

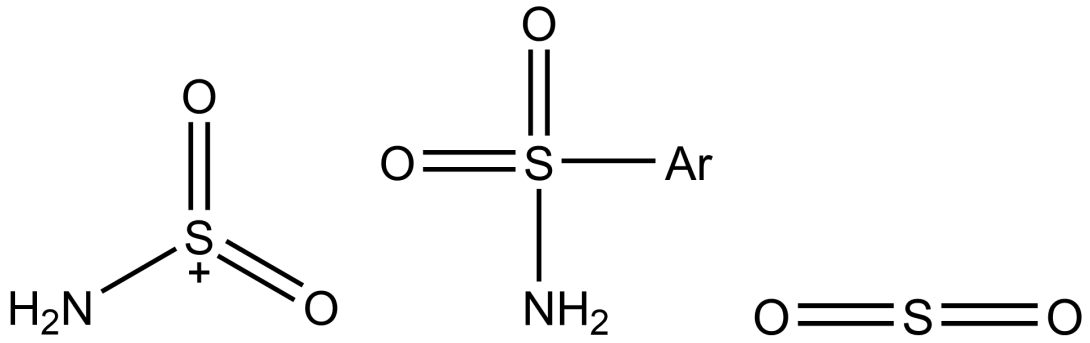
After the computational discovery of chemical words, our medicinal chemist collaborators select carbonic anhydrase enzyme system (PFam ID: PF00194) for pharmacologic evaluation. We provided the following words for the analysis in the decreasing

order of tf-idf score:

1. c1c(F)c(F)
2. NS(=O)(=O)
3. S(N)(=O)=O
4. c(F)c1F
5. CC(=O)c1ccc(
6. c(S
7. NS(=O)(=O)c1ccc(
8. c(Cl)c1)
9. S(=O)(=O)
10. c(F)c1F)

Our collaborators first observe that words 2, 3, 7, and 9 are either contain or already are complete chemical structures and identify these words as *sulfonamide*, *sulfonamide*, *aryl substituent*, and *sulfone*, respectively. Note that chemical words 2 and 3 are the same chemical structures, though their SMILES representations are different. Figure 3.1 visualizes the 2D structures of these compounds and provides their name.

During the literature review, 9 drugs that target carbonic anhydrases stand out. These drugs are *acetazolamide*, *methazolamide*, *ethoxzolamide*, *diclorphenamide*, *dorzolamide*, *brinzolamide*, *sulthiame*, *zonisamide*, and *topiramate* [76–79], whose 2D structures are illustrated in Figure 3.2. Figure 3.2 shows that all of these drugs contain *sulfonamide* (chemical words 2 and 3) as a substructure, which we also propose as a strong binding indicator for carbonic anhydrases. *Dorzolamide*, *brinzolamide*, and *sulthiame* additionally contain standalone *sulfone* structures, which we also listed as a strong binding indicator (chemical word 9). These highlight that biomolecular language processing can be used to reveal binding indicator chemical substructures.



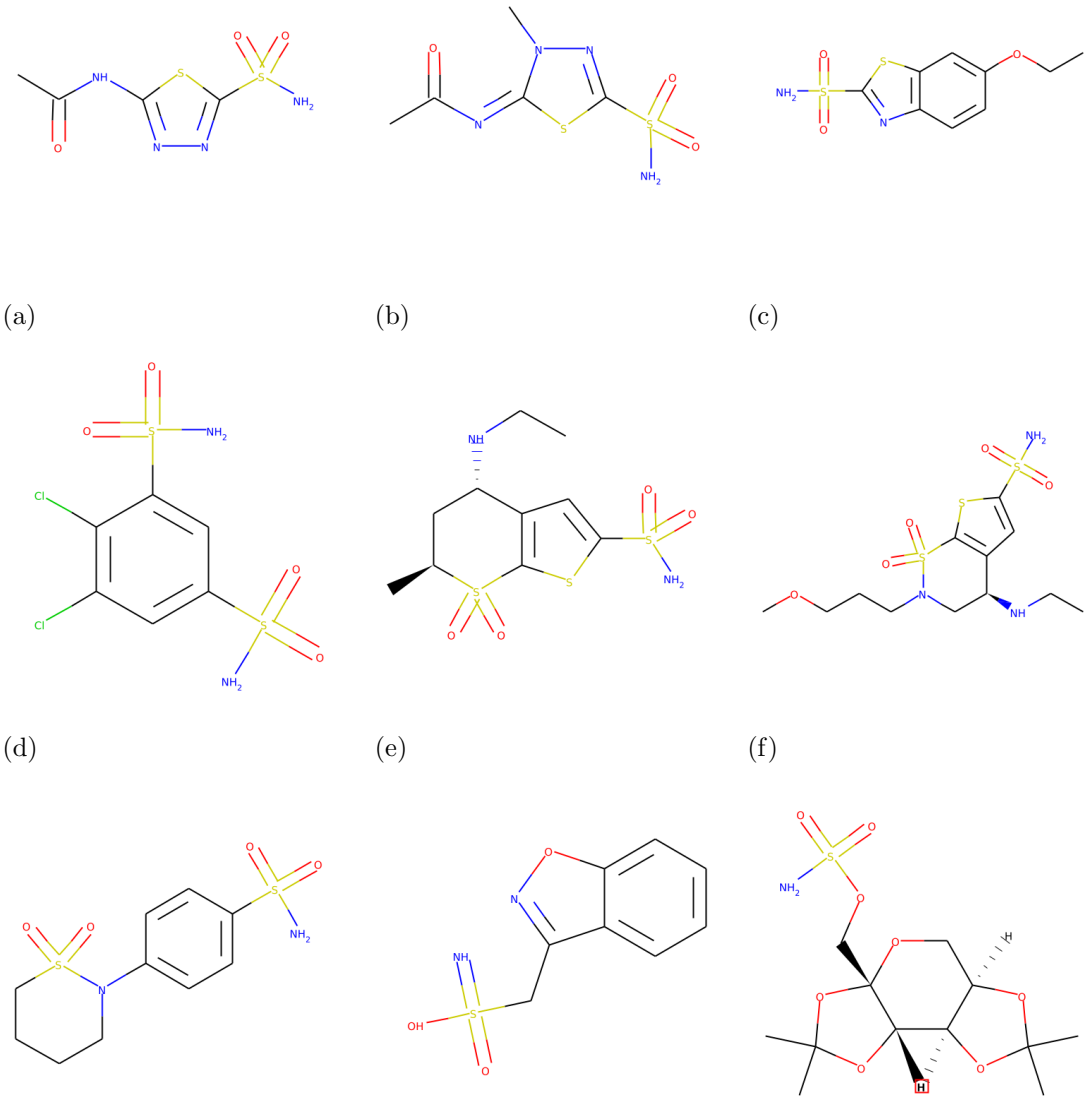
(a) (b) (c)

Figure 3.1. 2D representations of *sulfonamide* (a), *aryl substituent* (b), and *sulfone* (c).

Our collaborators evaluate the rest of the chemical words in the list, too. They conclude that despite having a non-closed branch and being invalid SMILES strings, chemical words 1, 4, and 10 have *fluorine substituent aromatic ring* substructure and chemical word 7 has a *chlorine substituent aromatic ring* substructure, which are meaningful chemical structures. Even though whether these substructures indicate strong binding to carbonic anhydrases is still an open question, these observations demonstrate that chemical words can be used to express compounds in terms of meaningful substructures.

The successful identification of strong binding indicators to carbonic anhydrases motivates applying the same computational analysis to more protein families. Our collaborators select histone deacetylase enzyme system (PFam ID: PF00850), casein kinase 1 gamma enzyme system (PFam ID: PF12605), guanylate kinase enzym system (PFam ID: PF00625), and ATPase enzyme system (PFam ID: PF00004). The analysis predicts ten chemical words as high-affinity indicators per family and the pharmacologic evaluation verifies the following chemical word sets, respectively:

- {NC(=O)CCCC, CCC(=O), c1ccccc1, c1c(F)c(F)c(F)c(F)c1F, CCC(F) },
- {c1ccncc1, c1ccncc1), c2ccc(NC(=O)N, CC(=O)c1c(C), NC), C(F)(F)F)CC11},
- {F)c(F)c1, C2CCN(CC2)},
- {nCCS(=O)(=O), OCC1, OCC2}.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.2. 2D representations of *acetazolamide* (a), *methazolamide* (b), *ethoxzolamide* (c) , *diclorphenamide* (d), *dorzolamide* (e), *brinzolamide* (f), *sulthiame* (g), *zonisamide* (h), and *topiramate* (i).

These strengthen our claim that identified chemical words are chemically meaningful structures.

4. CHEMBOOST: A CHEMICAL LANGUAGE BASED APPROACH FOR PROTEIN - LIGAND BINDING AFFINITY PREDICTION

4.1. Introduction

Identification of high affinity drug-target interactions (DTI) powered by the available knowledgebase of protein-ligand interactions is an important first step in the drug discovery pipeline. Computational tools from structure-based drug design [80] to quantitative structure–activity relationship (QSAR) [81] can accelerate this critical step by narrowing down potential binding partners. The prediction of binding affinity for novel interactions is still a challenging task because (i) representation of proteins and ligands in computational space is complicated by the inherent three-dimensional (3D) nature of the interaction [82], (ii) as of April 2020, there are only around 17680 protein-ligand complex structures in PDBBind [83], (iii) the chemical space sampled by the currently available data (560K proteins in SwissProt [30], 2M compounds in ChEMBL [84]) is limited. Recently, powered by the increase in the quantity, quality, and coverage of protein-ligand interaction data as well as computational advances, machine learning is gaining traction in the DTI prediction task. DTI prediction has been investigated as a binary classification problem [66, 85–92] or a regression problem to predict affinity [4, 93–99].

Similar to structure-based drug design studies [100,101], machine learning methodologies can utilize the 3D structure information of a protein-ligand complex to predict binding affinity [95–97]. Such structure-based methodologies, however, are limited by the available structural information on the complex as stated in point (ii). Two dimensional (2D) graph convolutions can also be used to learn molecule representations [102–104]. However, graph-based methodologies rely on complex and challenging-to-interpret graph convolutional networks for representation learning [105]. An attractive alternative is to use a string-based compound representation, which allows the

application of the recent advances in natural language processing (NLP). The field of chemical linguistics that brings the chemistry and linguistics domains together has been growing since its inception in the 1960s [106]. A recent review highlights the impact of NLP on drug discovery studies [16].

Simplified Molecular Input Line Entry System (SMILES) is a specialised syntax to represent molecules with an alphabet of over sixty characters. The SMILES representation can be used to reconstruct the 2D molecular graph, indicating its potential for encoding valuable molecular information. SMILES has been shown to perform as well as 2D representation-based graph convolutional embeddings in chemical property prediction [102] and drug-target interaction prediction [103]. The SMILES representation has been successfully used in comparison or search algorithms [107, 108], as well as for different problems ranging from information retrieval [109] to the prediction of chemical reaction outcomes [67], and the design of novel scaffolds to expand the chemical space [110]. Here, our aim is to treat the SMILES representation as a language and develop a machine learning and NLP based methodology for the task of protein-ligand binding affinity prediction.

We propose ChemBoost, a novel chemical-language based approach that uses distributed "chemical word" vectors for protein and/or ligand representation to predict interaction strength between targets and compounds. ChemBoost views SMILES strings as documents formed in a chemical language and processes the language units to create ligand and protein representations for affinity prediction. ChemBoost uses "SMILESVec" [32], a compound representation technique that utilizes the SMILES form. Distributed chemical word vectors are learned from a large corpus containing millions of SMILES strings. The chemical language-based nature of ChemBoost allows us to take advantage of the abundant textual data to learn chemical representations unlike structure-based learning algorithms [95–97], which are trained on a limited number of samples.

In the ChemBoost models, ligands are represented with their SMILESVec vectors.

We investigate two different approaches for protein representation. The first approach is the standard approach for protein representation, where proteins are represented with their sequences. We utilize the Smith-Waterman and ProtVec [33] algorithms to obtain sequence-driven protein representations. The second approach is a ligand-centric approach, where proteins are represented with the distributed vectors of their ligands. Biologically and functionally similar proteins often bind to chemically similar ligands and ligand-centric protein similarity calculations have been used in clustering and identifying similar proteins [32, 111–114]. To the best of our knowledge, this is the first study that investigates the effectiveness of distributed chemical word vectors and ligand-centric protein representations for protein-ligand binding affinity prediction. The effect of representing proteins through the chemical words of their known ligands or only high affinity ligands, as well as combining protein sequence based representation with ligand-centric representation are also examined.

We compare the ChemBoost models with three state-of-the-art binding affinity prediction approaches: KronRLS [93], SimBoost [94] and DeepDTA [4], all of which exploit the protein sequence information explicitly. SimBoost further integrates drug-target interaction network statistics to increase prediction accuracy. SimBoost and KronRLS employ traditional machine learning models for prediction, whereas DeepDTA is built upon a multi-layered CNN architecture. Thanks to its novel chemical language based and ligand-centric representations, ChemBoost achieves state-of-the-art performance using a simple predictor, eXtreme Gradient Boosting (XGBoost) [115].

4.2. ChemBoost

4.2.1. Datasets

To benchmark the performance of ChemBoost, we use KIBA [116] bioactivity dataset of proteins from Kinase family and BDB dataset that we extract from the BindingDB database [117], which contains proteins from different families. To compile the BDB dataset, BindingDB is filtered based on the following criteria: (i) proteins and compounds with at least 6 and 3 interactions are kept, respectively, (ii) the experiment with high affinity is selected, if there are multiple instances of the same protein-ligand pair, and (iii) only the interactions with K_d values are included, and then converted into pK_d with

$$pK_d = -\log_{10}(\frac{K_d}{1e9}). \tag{4.1}$$

BDB dataset comprises about 31K interactions between 490 proteins and 924 compounds. The average number of ligands with known binding affinity values for a protein is 53.3, and the average number of ligands with strong binding affinity values (i.e., pK_d value > 7) for a protein is 7.3. The pK_d threshold of strong/weak binding was selected as in the literature [94].

The KIBA dataset comprises KIBA scores for about 118K interactions of 229 proteins and 2111 compounds such that all proteins and compounds have at least 10 interactions [94]. KIBA score is a combination of different bioactivity measurement sources such as K_i, K_d or IC_{50} [116]. The average number of ligands with known binding affinity values for a protein is 516.4, and the average number of ligands with strong binding affinity values (i.e., KIBA score < 12.1) for a protein is 99.2. Protein diversity is lower in KIBA than KIBA as it contains kinase proteins only. Figure 4.1 illustrates the distribution of the binding affinity values of the protein-ligand pairs in BDB and KIBA datasets. We observe a strong peak at $pK_d = 5$ for BDB, since low affinities are frequently reported as $K_d >= 10000$ ($pK_d <= 5$).

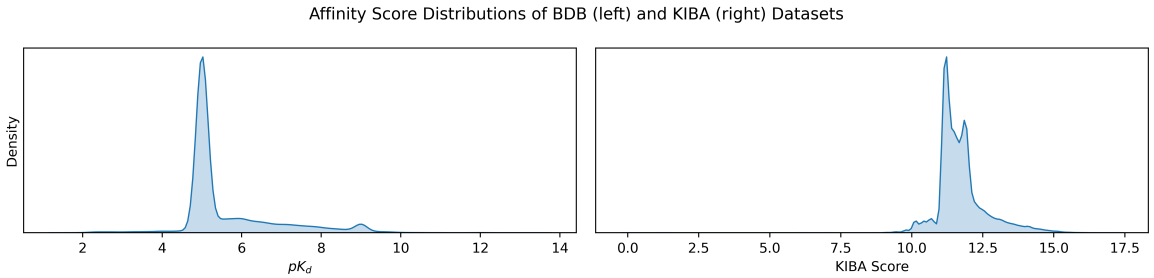


Figure 4.1. Distribution of binding affinity values in BDB and KIBA.

4.2.2. Ligand Representation

ChemBoost adopts SMILESVec to represent ligands through chemical language. As described in Chapter 2.3.3.1, SMILESVec relies on chemical words, which we identify with k-mer and BPE approaches. We set k to 8, since 8-mers were shown to outperform other options ranging from 4-mers to 12-mers [32]. BPE algorithm is trained on approximately 1.7M canonical SMILES strings collected from ChEMBL database (vChEMBL23), with the vocabulary size of 20K, character coverage of 0.99 and maximum word length of 100 characters. The `sentencepiece` library in Python is utilized. In order not to omit the symbols in the rich vocabulary of SMILES strings, `number_split` and `unicode_split` parameters are set to `False`.

Having identified the chemical words, we employ Word2Vec algorithm to learn chemical word embeddings by training the algorithm on vChEMBL23. We use `gensim` implementation [118] of Word2Vec with `Skip-Gram` approach and the size of the vectors is set as the default value of 100.

Besides the chemical language based vectors, we use molecular access system (MACCS) keys [62] and Morgan fingerprints [63] to represent the ligands. We use rdkit [65] to compute both representation vectors and use 166-dimensional MACCS vectors and 2048-dimensional Morgan fingerprints with radius 2. See Chapter 2.3.5 for more details on these methods.

4.2.3. Protein Representation

We represent proteins in a ligand-centric way (Chapter 2.3.3.3), in which the average of the word embeddings of the chemical words in their known ligands are used to construct the protein vector [32] . The chemical words that are used to build the SMILESVec representation are either 8-mers or BPE-based words. Consequently, representation of proteins change according to the word identification technique that is used to describe SMILESVec.

We build on the vanilla ligand-centric protein representation by introducing new ligand selection approaches. In addition to the original approach, in which all of the protein's known ligands in the training set are used to represent a protein, we investigate using only protein's high affinity ligands in the training set, and utilizing a universal high affinity ligand database.

In the first approach we introduce, only the chemical words of the high affinity (in other words strong binding) ligands in the training are used to represent a protein. Here, the intuition is that high affinity ligands of a protein might be more informative about its binding characteristics [119]. For BDB, the pK_d value of 7 is selected as the threshold to divide the ligands into strong-binding and weak-binding classes ($pK_d >$ 7 strong binding), whereas for the KIBA dataset, KIBA score of 12.1 is set as the threshold [94]. If there is no high affinity chemical of a protein in the training set, all known ligands are used. We use only the ligands in the training set for protein representation to prevent information leak from test set.

In the other ligand selection approach we propose, BindingDB is filtered to create a database of high affinity protein-chemical pairs. BindingDB includes experiments whose affinity scores reported in one or more of K_i, IC_{50}, EC_{50} units. A threshold of 100 is selected for each unit to classify experiments as weak and strong binding. Finally, the protein-ligand pairs in the training set and test sets of the models are removed to prevent data leak. The integration of a universal database enriches the protein vectors

with more ligand information and also reduces the number of protein with no known strong binding ligand.

In addition to the proposed ligand-centric protein representation framework, we represent proteins with two existing methods, ProtVec and Smith-Waterman (SW), which we describe in Chapter 2.3.3.2 and Chapter 2.3.5, respectively.

4.2.4. Benchmark Models

We compare the methods presented here with three state of the art DTA prediction models. We adopt Kronecker-Regularized Least Squares (KronRLS) algorithm that predicts binding affinity while representing both proteins and ligands with their pairwise similarity score matrices [93]. In order to compute the similarity between proteins and between compounds, Smith Waterman algorithm (SW) and PubChem structure clustering tool (`http://pubchem.ncbi.nlm.nih.gov`) are utilized, respectively. Second, we employ SimBoost, which is a gradient boosting machine based method that depends on feature engineering of ligands and proteins utilizing information such as similarity and network-inferred statistics [94]. Last, we compare our results with DeepDTA, which is a multi-layered Convolutional Neural Network (CNN) [4] based prediction model. The SMILES representations of ligands and sequences of proteins are provided as inputs to DeepDTA to predict the binding affinity. All of these baselines utilize protein sequence features explicitly.

4.2.5. Evaluation

As protein-chemical binding affinity prediction is a regression problem, the performance of the presented models are measured by calculating the Concordance Index (CI), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 metrics.

CI [120] is described as

$$CI = \frac{1}{Z} \sum_{\delta_x > \delta_y} h(b_x - b_y) \tag{4.2}$$

where

b_x is the prediction value for the larger affinity δ_x

b_y is the prediction value for the smaller affinity δ_y

Z is a normalization constant

$h(m)$ is the step function, which is equal to 0 if $m < 0$, 0.5 if $m = 0$, and 1 if $m > 0$ [93].

MSE measures the average squared difference between the predicted and the actual values with

$$MSE = \frac{1}{n} \sum_{k=1}^{n} (p_k - y_k)^2 \tag{4.3}$$

where

n is the number of samples

p_k is the predicted value for k^{th} sample

y_k is the actual value for k^{th} sample.

We also compute RMSE by taking the square root of Equation (4.3) in order to reduce MSE to the order of the actual affinity scores.

Finally, we compute R^2 which measures how much of the variance in the actual values is explained by the predictions. We calculate R^2 using

$$R^2 = 1 - \frac{\sum_{k=1}^{n} (y_k - p_k)^2}{\sum_{k=1}^{n} (y_k - \bar{y})^2} \tag{4.4}$$

where

p_k is the predicted value for k^{th} sample

y_k is the actual value for k^{th} sample

$\bar{y}$ is the mean of the actual values.

4.2.6. Experimental Settings

We use the same training and test folds that are used in DeepDTA work [4], but use ChEMBL canonical SMILES in ligand representation of the ligands to comply with SMILESVec vectors. In these folds, both datasets are randomly divided into six equal parts and one part is separated as the independent test set. The remaining folds are used to determine the model hyper-parameters, such as learning rate and number of trees via five-fold cross validation. The hyper-parameter combination with which we obtain the best MSE value based on the cross-validation results over the training set is selected to model the test set. To report the performance of the models on the test set, we first train the models on 4 folds of the training set 5 times by masking a different fold at each time. Then, we evaluate each of the 5 models on the test set and report their average performance values alongside the standard deviation.

4.3. Results

We introduced ChemBoost, a novel protein-ligand binding affinity prediction approach in which both ligands and chemicals are represented through distributed chemical word vectors. We now investigate chemical language based biomolecule representations for affinity prediction. We further compare the impact of three approaches in ligand-centric representation of proteins: (i) using all ligands with a reported affinity, (ii) using only strong binding (SB) ligands and (iii) utilizing an additional, non-redundant high affinity protein-ligand interaction database. SW and ProtVec are used as alternative protein representations to assess the effectiveness of ligand-centric protein representation. In addition, the effect of the chemical word discovery approach is examined by comparing 8-mers with BPE words. We evaluate different predictive models on BDB and KIBA using CI, MSE, RMSE and R^2 as metrics and compare ChemBoost

models with three state of the art affinity prediction models, two traditional machine learning based systems, namely KronRLS and SimBoost, and a deep-learning based approach, DeepDTA. We report the significance levels of comparisons using paired t-test for models with close scores.

4.3.1. Investigation of Chemical Language Based Biomolecule Representations

In this subsection, we inspect the effectiveness of chemical language based ligand and protein representations in the affinity prediction problem. We train each model five times with different folds of the training set. We measure the performance on the test set for each trained model and report the average results on BDB and KIBA datasets utilizing XGBoost as the prediction algorithm for all cases. We compute CI, MSE, (Table 4.1 and Table 4.2) RMSE, and R^2 scores (Table 4.3 and Table 4.4), but compare the models based only on CI and MSE for readability, because MSE is highly parallel with RMSE and R^2.

Ligand representation. We first test the impact of ligand representation by creating two baselines. Model (S1) does not utilize any ligand information, whereas Model (R1) represents each ligand with a 100-dimensional random vector sampled from a uniform distribution over [0, 1). In both models, proteins are represented with SW vectors. Model (R1) outperforms Model (S1) on both datasets, with respect to both evaluation metrics, emphasizing the requirement for ligand information to build a successful predictive model. As each random vector is associated with a particular ligand, their success in prediction performance can be linked to ligand-specificity. We suggest that these random vectors encode the identity of the ligands and thus, improve model performance, even though the content of the representation vector is pure noise.

Table 4.1. CI and MSE scores of ChemBoost models on BDB.

Name	**Protein Representation**	**Ligand Representation**	**CI**	**MSE**
Model (S1)	SW	-	0.687 (0.002)	1.037 (0.006)
Model (S2)	-	SMILESVec (8mer)	0.773 (0.002)	0.876 (0.005)
Model (R1)	SW	Random	0.859 (0.002)	0.512 (0.005)
Model (R2)	Random	SMILESVec (8mer)	0.849 (0.002)	0.537 (0.009)
Model (F1)	SW	MACCS	0.811 (0.003)	0.817 (0.016)
Model (F2)	SW	Morgan	0.819 (0.002)	0.767 (0.016)
Model (1)	SW	SMILESVec (8mer)	0.873 (0.001)	0.439 (0.008)
Model (2)	ProtVec	SMILESVec (8mer)	0.854 (0.002)	0.512 (0.004)
Model (3)	ProtVec	SMILESVec (BPE)	0.849 (0.002)	0.548 (0.008)
Model (4)	SMILESVec (all, 8mer)	SMILESVec (8mer)	0.847 (0.001)	0.524 (0.006)
Model (5)	SMILESVec (SB, 8mer)	SMILESVec (8mer)	0.845 (0.002)	0.478 (0.005)
Model (6)	SMILESVec (SB, BPE)	SMILESVec (BPE)	0.842 (0.001)	0.497 (0.007)
Model (7)	SMILESVec (BindingDB, SB, 8mer)	SMILESVec (8mer)	0.856 (0.001)	0.454 (0.007)
Model (8)	SW & SMILESVec (SB, 8mer)	SMILESVec (8mer)	0.873 (0.001)	0.420 (0.004)
Model (9)	SW & SMILESVec (BindingDB, SB, 8mer)	SMILESVec (8mer)	0.871 (0.002)	0.420 (0.007)

Table 4.2. CI and MSE scores of ChemBoost models on KIBA.

Name	**Protein Representation**	**Ligand Representation**	**CI**	**MSE**
Model (S1)	SW	-	0.683 (0.000)	0.585 (0.000)
Model (S2)	-	SMILESVec (8mer)	0.699 (0.000)	0.425 (0.001)
Model (R1)	SW	Random	0.803 (0.001)	0.276 (0.002)
Model (R2)	Random	SMILESVec (8mer)	0.815 (0.001)	0.258 (0.002)
Model (F1)	SW	MACCS	0.829 (0.001)	0.222 (0.002)
Model (F2)	SW	Morgan	0.847 (0.001)	0.186 (0.002)
Model (1)	SW	SMILESVec (8mer)	0.837 (0.001)	0.203 (0.002)
Model (2)	ProtVec	SMILESVec (8mer)	0.818 (0.001)	0.244 (0.001)
Model (3)	ProtVec	SMILESVec (BPE)	0.814 (0.001)	0.252 (0.002)
Model (4)	SMILESVec (all, 8mer)	SMILESVec (8mer)	0.823 (0.001)	0.243 (0.003)
Model (5)	SMILESVec (SB, 8mer)	SMILESVec (8mer)	0.829 (0.001)	0.221 (0.001)
Model (6)	SMILESVec (SB, BPE)	SMILESVec (BPE)	0.825 (0.001)	0.227 (0.001)
Model (7)	SMILESVec (BindingDB, SB, 8mer)	SMILESVec (8mer)	0.829 (0.001)	0.223 (0.001)
Model (8)	SW & SMILESVec (SB, 8mer)	SMILESVec (8mer)	0.837 (0.001)	0.206 (0.001)
Model (9)	SW & SMILESVec (BindingDB, SB, 8mer)	SMILESVec (8mer)	0.836 (0.001)	0.207 (0.002)

Table 4.3. RMSE and R^2 scores of ChemBoost models on BDB.

Name	**Protein Representation**	**Ligand Representation**	**RMSE**	**R^2**
Model (S1)	SW	-	1.018 (0.003)	0.265 (0.004)
Model (S2)	-	SMILESVec (8mer)	0.936 (0.002)	0.379 (0.003)
Model (R1)	SW	Random	0.716 (0.003)	0.637 (0.003)
Model (R2)	Random	SMILESVec (8mer)	0.733 (0.006)	0.619 (0.006)
Model (F1)	SW	MACCS	0.904 (0.009)	0.421 (0.012)
Model (F2)	SW	Morgan	0.874 (0.009)	0.458 (0.011)
Model (1)	SW	SMILESVec (8mer)	0.662 (0.006)	0.689 (0.006)
Model (2)	ProtVec	SMILESVec (8mer)	0.716 (0.003)	0.637 (0.003)
Model (3)	ProtVec	SMILESVec (BPE)	0.740 (0.006)	0.611 (0.006)
Model (4)	SMILESVec (all, 8mer)	SMILESVec (8mer)	0.724 (0.004)	0.628 (0.004)
Model (5)	SMILESVec (SB, 8mer)	SMILESVec (8mer)	0.692 (0.004)	0.661 (0.003)
Model (6)	SMILESVec (SB, BPE)	SMILESVec (BPE)	0.705 (0.005)	0.647 (0.005)
Model (7)	SMILESVec (BindingDB, SB, 8mer)	SMILESVec (8mer)	0.674 (0.006)	0.678 (0.005)
Model (8)	SW & SMILESVec (SB, 8mer)	SMILESVec (8mer)	0.648 (0.003)	0.702 (0.003)
Model (9)	SW & SMILESVec (BindingDB, SB, 8mer)	SMILESVec (8mer)	0.648 (0.005)	0.702 (0.005)

Table 4.4. RMSE and R^2 scores of ChemBoost models on KIBA.

Name	**Protein Representation**	**Ligand Representation**	**RMSE**	**R^2**
Model (S1)	SW	-	0.765 (0.000)	0.139 (0.001)
Model (S2)	-	SMILESVec (8mer)	0.652 (0.000)	0.374 (0.001)
Model (R1)	SW	Random	0.525 (0.002)	0.594 (0.003)
Model (R2)	Random	SMILESVec (8mer)	0.508 (0.002)	0.621 (0.002)
Model (F1)	SW	MACCS	0.471 (0.002)	0.674 (0.002)
Model (F2)	SW	Morgan	0.431 (0.002)	0.727 (0.003)
Model (1)	SW	SMILESVec (8mer)	0.450 (0.002)	0.702 (0.003)
Model (2)	ProtVec	SMILESVec (8mer)	0.494 (0.001)	0.641 (0.001)
Model (3)	ProtVec	SMILESVec (BPE)	0.502 (0.002)	0.630 (0.003)
Model (4)	SMILESVec (all, 8mer)	SMILESVec (8mer)	0.493 (0.003)	0.642 (0.004)
Model (5)	SMILESVec (SB, 8mer)	SMILESVec (8mer)	0.470 (0.001)	0.675 (0.001)
Model (6)	SMILESVec (SB, BPE)	SMILESVec (BPE)	0.477 (0.002)	0.665 (0.002)
Model (7)	SMILESVec (BindingDB, SB, 8mer)	SMILESVec (8mer)	0.472 (0.001)	0.672 (0.001)
Model (8)	SW & SMILESVec (SB, 8mer)	SMILESVec (8mer)	0.454 (0.002)	0.697 (0.002)
Model (9)	SW & SMILESVec (BindingDB, SB, 8mer)	SMILESVec (8mer)	0.455 (0.002)	0.696 (0.002)

We then assess the performance of MACCS keys by comparing Model (R1) and Model (F1). Both models represent proteins with SW vectors, whereas Model (R1) uses

random vectors for ligand representation and Model (F1) uses MACCS keys. Model (F1) outperforms Model (R1) on KIBA in terms of both MSE and CI, whereas on BDB, Model (R1) achieves better scores. We then evaluate the effectiveness of Morgan fingerprints by comparing Model (F1) and Model (F2) and observe that Morgan fingerprints are superior to MACCS keys for affinity prediction, since they yield higher scores on both datasets. However, similar to MACCS keys, the performance of Morgan fingerprints is lower than random vectors on BDB, suggesting that fingerprints are not sufficiently distinctive for the ligands in BDB.

In order to analyze the performance of distributed chemical word vectors based approach in ligand representation, we compare Model (1) and Model (F2). While both models represent proteins using SW, Model (1) represents ligands with SMILESVec and Model (F2) represents ligands with Morgan fingerprints. Morgan vectors yield the best performance among all models on KIBA in terms of both metrics, although they underperformed random vectors on BDB. SMILESVecs, on the other hand, achieve high performance on both datasets, suggesting that they are more consistent representations for the binding affinity prediction task.

We design Models (2), (3), (5) and (6) to investigate the effect of different chemical word discovery techniques on prediction performance. Models (2) and (3) adopt ProtVec for protein representation, whereas Models (5) and (6) use the average of chemical word vectors of proteins with high affinity ligands. Comparing Model (2) with (3) and Model (5) with (6), we observe that 8-mer based representations achieves higher performance than their BPE counterparts, indicating that 8-mers are stronger language units for the affinity prediction task. Consequently, we decide to utilize 8-mers as chemical words in the remaining experiments.

Protein representation. We design two models to show the impact of protein representation on the affinity prediction problem. We construct Model (S2) which does not utilize any protein information and Model (R2) that represents each protein with

a random vector that is sampled from a uniform distribution over [0, 1). Both models represent the ligands with SMILESVecs. Model (R2) outperforms Model (S2) with respect to both MSE and CI, on both datasets. Similar to our experiment with ligand representation, we suggest that random vectors are interpreted as unique fingerprints for each protein by the prediction algorithm. This information boosts the performance of the system, validating the necessity of protein information for affinity prediction modeling.

To test the effectiveness of different protein representation techniques, we first compare protein-specific random vectors (Model (R2)) with SW (Model (1)) in which both models describe ligands with SMILESVecs. The use of SW representation improves prediction performance on both datasets for both metrics, demonstrating the advantage of SW vectors over random vectors. Then we compare SW (Model (1)) with ProtVec (Model (2)) in which SW outperforms ProtVec on both datasets for both evaluation metrics. This result shows that SW, which includes sequence similarity and amino acid physicochemical information, is a better alternative for representing proteins in affinity prediction, when combined with SMILESVec.

Although each binding affinity measurement provides a valuable data point for learning algorithms, most often, the mechanism of bimolecular interaction is accurately described by high affinity interactions [119]. Therefore, to verify that the contribution of high affinity ligands is more informative for ligand-centric protein representation, we compare the use of chemical words of all ligands with a reported affinity value (Model (4)) to the use of chemical words of high affinity ligands (Model (5)). In our experiments, all known ligands of a protein are the ones for which the affinity value to the protein is reported. In both cases, 8-mer based SMILESVecs are used to represent the ligands. The results illustrate that using high affinity ligands in protein representation outperforms the model in which all known ligands are utilized in terms of all metrics on both datasets. These results emphasize that considering strong binding ligands in protein representation improves the prediction performance, motivating us to construct Model (7).

In Model (7), the number of high affinity ligands incorporated in protein representation is increased by including high affinity protein-ligand pairs obtained from different experimental measurements in BindingDB (e.g. K_i, IC_{50} etc.). These pairs are used in the protein representation in addition to the ones that are already in the training set. We compare Model (7) with Model (5) using a paired t-test at 99% significance and observe significant improvement in MSE and CI on BDB dataset, whereas the same test indicates that their performance are on par with each other on KIBA. The improvement for BDB may be due to the higher increase in the number of high affinity ligands of a protein. The number of strong binding ligands of a protein increases 17.1 times in BDB but only 2.3 times in KIBA with the inclusion of the additional data.

On the other hand, Model (1), which utilizes SW in protein representation, outperforms Model (7) with respect to both metrics for both datasets, suggesting the merit of SW over ligand-centric protein representation. Then, we construct Model (8) and Model (9) where we concatenate SW vectors with ligand-centric representation to incorporate amino-acid sequence and ligand binding information. A paired t-test with 99% confidence does not indicate a distinction between Models (8) and (9), despite Model (9) utilizing an external database of high affinity protein-ligand pairs.

Model (8), however, provides an improvement over Model (6) and Model (9) outperforms Model (7), emphasizing that integration of SW brings in complementary information to ligand-centric representation on both datasets. Last, we compare the hybrid protein representation (Model (9)) with SW (Model (1)). Though the models perform similarly in terms of CI, MSE indicates a distinction between two representations. On BDB dataset, concatenating SW and ligand-centric vectors (Model (9)) yields a better MSE than Model (1) with 95% confidence, whereas on KIBA, the concatenation results in a performance decrease. This can be due to higher physicochemical similarity of Kinases in KIBA, compared to the heterogeneous structure of BDB.

We conclude that both protein and ligand information is indispensable for predicting binding affinities of protein-ligand pairs, and distributed chemical word vectors

(SMILESVecs) can be successfully utilized in representing these biomolecular entities. Distributed vectors for ligand representation combined with SW or ligand-centric protein vectors capture the necessary information for the targets. The results indicate that the combination of strong affinity ligands-based protein representation with SW can improve the predictive performance for datasets with high protein diversity (i.e. proteins from different families). On the other hand, SW vectors are suitable for protein representation in datasets with low protein diversity (i.e. proteins from the same family). We also suggest that 8-mers can be used as chemical words in affinity prediction, since it is a simple and effective technique that performs better than the more complicated BPE approach.

4.3.2. Comparing ChemBoost with the State of the Art Models

We compare one of the best ChemBoost models (Model (9)) with three state of the art drug-target affinity prediction models: (i) KronRLS [93], a regularized linear regression model that uses SW similarity of protein sequences and PubChem structure similarity of ligands, (ii) SimBoost [94], a gradient boosting tree based prediction algorithm that utilizes network statistics of protein-ligand interactions, in addition to the similarity scores used by KronRLS, (iii) DeepDTA [4], a multi-layered CNN that learns features through raw protein sequences and SMILES strings. Table 4.5 reports the metrics for these models and ChemBoost. Here ChemBoost refers to Model (9) in Table 4.1 and describes ligands with SMILESVec and proteins by concatenating SW and ligand-centric representations.

Table 4.5. CI, MSE, RMSE, and R^2 scores of the state of the art affinity prediction models and ChemBoost on BDB and KIBA.

	Model	**CI**	**MSE**	**RMSE**	**R^2**
BDB	KronRLS	0.815 (0.003)	0.939 (0.005)	0.969 (0.002)	0.334 (0.003)
	SimBoost	0.855 (0.004)	0.501 (0.026)	0.707 (0.018)	0.645 (0.018)
	DeepDTA	0.863 (0.007)	0.397 (0.011)	0.630 (0.009)	0.719 (0.008)
	ChemBoost	0.871 (0.002)	0.420 (0.007)	0.648 (0.005)	0.702 (0.005)
KIBA	KronRLS	0.785 (0.001)	0.411 (0.001)	0.641 (0.001)	0.395 (0.002)
	SimBoost	0.836 (0.001)	0.224 (0.001)	0.473 (0.001)	0.671 (0.001)
	DeepDTA	0.846 (0.002)	0.215 (0.005)	0.464 (0.006)	0.683 (0.008)
	ChemBoost	0.836 (0.001)	0.207 (0.002)	0.455 (0.002)	0.696 (0.002)

SimBoost performs better than KronRLS in terms of all evaluation metrics on both datasets. This is an expected outcome given that SimBoost relies on network-based features as well as the features KronRLS utilized, namely the SW and PubChem similarity scores. ChemBoost, on the other hand, obtains either similar to or higher scores than SimBoost on both datasets. Although both ChemBoost and SimBoost depend on the XGBoost algorithm and utilize SW vectors for protein representation, ChemBoost incorporates chemical word vectors for both ligand and protein representations, indicating the effectiveness of the information they bring in. On BDB dataset, DeepDTA obtains a better MSE than ChemBoost, but a similar CI (significance level 95%), whereas on KIBA, ChemBoost achieves a better MSE but worse CI than DeepDTA with the 99% significance level, indicating their prediction strengths are comparable. Hence, we suggest that ChemBoost achieves state-of-the-art level performance by exploiting distributed chemical word vectors and protein sequence information.

4.3.3. ChemBoost can Capture Functional Similarity of Proteins with Low Sequence Similarity

We investigate different ChemBoost models to observe the impact of using amino-acid sequence based protein representation in comparison to ligand-centric protein

representation in predictive performance. Our focus is on proteins with low sequence similarity that bind to similar ligands. This case, where unrelated proteins bind to identical or similar ligands is complicated even in 3D space and no clear patterns emerge [121]. However, ligand binding is known to capture mechanistic information about the protein [112, 119] and a ligand centric approach is expected to boost performance in the protein-ligand interaction prediction task, especially for proteins of low sequence similarity.

Therefore, we investigate the performance of ChemBoost models as a function of protein sequence similarity. For each protein-ligand pair (P-L) in the test set, we compute the normalized SW similarity score (see Chapter 2.3.5 for more details) of P to the other interacting proteins of L in the training set. Then, we calculate the maximum score, which we refer to as Maximum Sequence Similarity (MSS_{PL}), for a P-L pair. We formulate MSS_{PLL} as

$$MSS_{PL} = max\{SW(P, p) \forall p \in P(L)\} \tag{4.5}$$

where $P(L)$ is the set of proteins with a reported affinity with ligand L in the training set.

We divide the P-L pairs in the test set into 4 categories with respect to their MSS, such that each category comprises approximately the same number of interactions (around 5000 P-L pairs for KIBA and 1250 for BDB). Here, an interaction pair P-L being in a high MSS category, such as (60.00, 100], indicates that the models already observed an interaction of ligand L with a protein that is sequence-wise similar to P. Likewise, we can consider low MSS pairs in test set as pairs with no similar interactions in the training set.

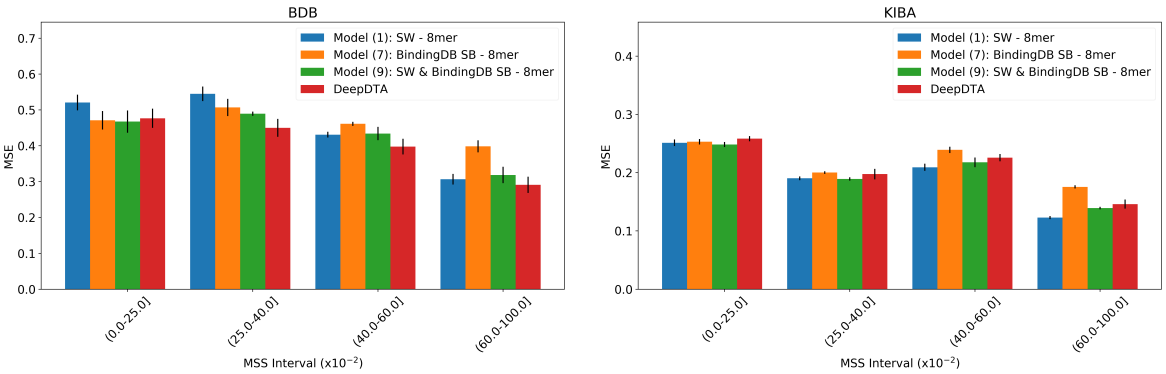


Figure 4.2. Test set performance of ChemBoost and DeepDTA on BDB (left) and KIBA (right) with respect to MSS of interactions.

For each MSS category, we compute MSE value with Model (1), Model (7), Model (9) of ChemBoost and DeepDTA. All ChemBoost models use SMILESVec to describe ligands. For protein representation, however, Model (1) utilizes SW vectors, Model (7) uses a strictly ligand-centric approach with BindingDB integration and Model (9) combines both. Last, DeepDTA uses raw protein and ligand sequences and shown as the best performing model among the existing benchmarks in Chapter 4.3.2. We repeate the computations 5 times for each model using different training fold configurations and Figure 4.2 illustrates the mean and standard deviation of MSE scores for the given MSS intervals for BDB (left) and KIBA (right) datasets.

For BDB, Model (1) that uses only SW vectors performs significantly worse than all three models (paired t-test, significance=95%) in the lowest MSS category, proposing that SW vectors are insufficient to capture functional similarity when sequence similarity is low. The same statistical test did not suggest a distinction for KIBA. On the other hand, we observe a clear performance increase of Model (1) in the highest MSS category for both datasets.

Unlike Model (1), Model (7) is unaware of protein sequence information since it depends on SMILESVecs based ligand-centric representations to describe proteins. Model (7) is on par with the rest of the models in the lowest MSS category of both datasets and second lowest category of KIBA, although its overall performance is signif-

icantly worse than all. This highlights the merit of using ligand-centric representations when functional similarity cannot be captured from protein sequence.

Model (9) combines SW with chemical language based ligand-centric representations and performs comparably to DeepDTA as shown in Section Chapter 4.3.2. Here, we observe that Model (9) achieves relatively consistent scores across all MSS categories. For all categories except the highest MSS category of KIBA, Model (9) is on par with (99% significance) the best model of the category. On the other hand, DeepDTA is negatively affected by low sequence similarity and presents an unstable prediction performance with higher standard deviation. These results show that, Model (9) can exploit the advantages of both SW and ligand-centric protein representations and is more consistent than DeepDTA.

In conclusion, we show that the information encoded through sequence similarity and ligand-centric approach emphasizes different characteristics in protein representation. While ligand-centric models are able to capture functional similarity without using protein sequence information, SW-based models can exploit high sequence similarity. Combining these two complementary approaches, ChemBoost achieves state of the art performance with robustness to the changes in protein sequence similarity.

4.3.4. Evaluating ChemBoost on Novel Biomolecules

In order to estimate the performance of ChemBoost for novel proteins and ligands, we perform experiments using new train test splits. We randomly group the proteins and ligands in BDB and KIBA into two categories as *known* and *unknown* in order to create one training split and four test splits, per dataset. For BDB and KIBA, respectively, we divide the interactions of the molecules in the known group into two sets and obtain the training set and the first test set (warm). Afterwards, we identify the interactions of the known proteins with unknown ligands (cold ligand), unknown proteins with known ligands (cold protein), and unknown proteins with unknown ligands to form the second, third, and fourth test sets, respectively [122].

Table 4.6. Performance of three ChemBoost models and DeepDTA on warm and cold ligand test sets of BDB and KIBA.

		Warm		**Cold Ligand**	
	Model	**MSE**	**CI**	**MSE**	**CI**
BDB	Model (1)	0.373 (0.016)	0.885 (0.010)	1.178 (0.079)	0.736 (0.036)
	Model (7)	0.404 (0.010)	0.863 (0.010)	1.185 (0.143)	0.700 (0.044)
	Model (9)	0.361 (0.010)	0.880 (0.008)	1.157 (0.285)	0.730 (0.044)
	DeepDTA	0.345 (0.026)	0.879 (0.007)	1.350 (0.306)	0.672 (0.024)
KIBA	Model (1)	0.185 (0.010)	0.845 (0.006)	0.450 (0.040)	0.732 (0.009)
	Model (7)	0.202 (0.008)	0.839 (0.005)	0.445 (0.038)	0.736 (0.009)
	Model (9)	0.183 (0.006)	0.847 (0.005)	0.442 (0.034)	0.735 (0.011)
	DeepDTA	0.199 (0.014)	0.853 (0.005)	0.456 (0.068)	0.754 (0.012)

We train Model (1), Model (7), and Model (9), on the new training sets and test them on each test set. Here, the cold protein representation is especially challenging for Model (7) and Model (9), since they utilize the high affinity ligands of proteins for protein representation, but the proteins are absent in the training set. We alleviate this problem by using the ligand-centric vector of the sequence-wise most similar (based on 3-mer based Jaccard similarity) protein in the training set to compute ligand-centric vectors for cold proteins. We also evaluate the performance of DeepDTA as a strong benchmark using the same train-test configuration. We repeat the random splitting and model training 5 times and report the results in Table 4.6 and Table 4.7, for both datasets.

For the warm test set, we observe that the ranking of the models is similar to our previous results, validating that the new experimental setup is also reliable. On the other hand, the results for cold ligand split does not indicate an apparent ordering of the models, suggesting that SMILESVec representations, learned from an external corpus, are as good as the CNN-based representations learned from the training sets to represent novel ligands.

Table 4.7. Performance of three ChemBoost models and DeepDTA on the cold protein and cold both test sets of BDB and KIBA.

		Cold Protein		**Cold**	
	Model	**MSE**	**CI**	**MSE**	**CI**
BDB	Model (1)	0.720 (0.094)	0.799 (0.012)	1.393 (0.145)	0.657 (0.044)
	Model (7)	1.156 (0.251)	0.749 (0.023)	1.576 (0.185)	0.596 (0.055)
	Model (9)	0.800 (0.145)	0.786 (0.023)	1.358 (0.324)	0.665 (0.053)
	DeepDTA	0.810 (0.147)	0.778 (0.015)	1.522 (0.300)	0.614 (0.039)
KIBA	Model (1)	0.298 (0.024)	0.762 (0.031)	0.588 (0.058)	0.646 (0.043)
	Model (7)	0.453 (0.051)	0.734 (0.018)	0.667 (0.060)	0.638 (0.027)
	Model (9)	0.340 (0.011)	0.748 (0.021)	0.614 (0.047)	0.640 (0.033)
	DeepDTA	0.400 (0.054)	0.747 (0.020)	0.655 (0.080)	0.652 (0.045)

We also analyze the results on the cold protein split and observe that Model (1) and Model (9), which utilize SW vectors for protein representation, obtain higher results than Model (7). This shows that SW vectors are more robust to represent novel proteins, compared to ligand-centric representations. We then analyze the scores on the cold split and observe that all models perform worse than the first three test sets. This indicates that the interactions where the protein and ligand are both novel are challenging for all affinity prediction models.

5. DEBIASEDDTA: MODEL DEBIASING TO BOOST DRUG-TARGET AFFINITY PREDICTION

5.1. Introduction

The first step toward drug discovery is to identify high affinity protein-chemical pairs. However, the number of possible protein-chemical combinations makes this task a "needle in the haystack" problem (∼560K proteins in UniProt [30] and ∼2.1M chemicals in ChEMBL [84]). This is where drug-target affinity (DTA) prediction models come into play; they can rapidly identify high-affinity protein-chemical pairs in the combination space after learning generalizable affinity prediction rules from large interaction datasets.

The interaction datasets report affinity measurements for millions of protein-chemical pairs and stand as invaluable resources to learn rules of affinity prediction. However, they also contain spurious patterns that can misguide the learning [123–127]. For instance, a single atom may be separating actives and inactives of a target [128] and the prediction models can learn to predict interaction strength through that atom exclusively, instead of learning generalizable affinity prediction rules. Consequently, models struggle to estimate the binding affinity between unseen biomolecules, for which the learned shortcuts are unavailable [70, 126, 129–131]. These dataset shortcuts are the dataset biases and form a major problem to discover drugs for rare diseases or to identify novel chemical moieties to which proteins have not yet acquired resistance.

To the best of our knowledge, there is no study with a focus on boosting drug-target affinity prediction on novel biomolecules. Recent works studied the generalizability problem in a similar task, drug-target interaction prediction. They focused on the datasets and designed train-test splits with dissimilar biomolecules so that the training set biases are less rewarding on the test set [124, 130]. However, counter to the aim, these "dataset-oriented" approaches introduced the risk of degrading model generaliz-

ability and inaccurate estimation of dissimilar test set performance [132]. Furthermore, their use in the affinity prediction task would require non-trivial adaptations, as they exploit the two-class structure (active or inactive) in the drug-target interaction task.

An alternative perspective to cope with biases and improve model generalizability is to focus on the prediction models instead of datasets. This "model-oriented" perspective is free from the limitations of the task and has been recently successfully used in natural language processing [133–139], computer vision [140, 141], as well as for structure-based virtual screening [127]. Unfortunately, the impact of the model-oriented perspective on computational drug discovery was limited by the number of available 3D structures [127].

In this chapter, we propose DebiasedDTA, a novel model training framework to address dataset biases and boost the generalizability of DTA prediction models. DebiasedDTA adopts the model-oriented perspective and, unlike the dataset-oriented approaches proposed for drug-target interaction prediction, it is applicable to datasets with continuous and discrete labels without requiring modifications. In addition, DebiasedDTA can be used to debias DTA prediction models with any biomolecule representation and finds a wider application range than 3D-structure based approaches.

DebiasedDTA ensembles a "guide" and a "predictor" to train debiased DTA prediction models. The guide quantifies a particular type of training set bias and prepares a debiasing roadmap for the predictor. The predictor utilizes the roadmap in order to adapt the sample weights during training to avoid biases and to achieve higher generalizability on novel biomolecules.

We test DebiasedDTA with two guides on different bias sources and with three predictors to evaluate across biomolecule representations. Experiments on two datasets and ten test sets show that the proposed approach is robust to different bias sources and can boost prediction performance of DTA models with different drug-target representations. Noteworthy, the improvement is not only observed for novel biomolecules

but also for the seen ones.

DebiasedDTA is a novel approach that boosts the generalizability of DTA prediction models. Using a model-oriented perspective and a biomolecule representation independent sample weight adaptation strategy, DebiasedDTA can be adopted to enhance the prediction performance of any DTA prediction model that allows sample weighting.

5.2. DebiasedDTA

DebiasedDTA is a model debiasing framework to boost drug-target affinity prediction on novel biomolecules and consists of two DTA prediction models, the guide and the predictor. The guide aims to identify dataset biases only, and thus uses biomolecule representations that target a specific bias source in the dataset. When the guide is trained on the training set, it prepares a training roadmap for the predictor and the predictor follows the roadmap to drive its training away from dataset biases; towards generalizable information. The debiased model is used standalone to predict the affinity between target protein-chemical pairs. Figure 5.1 illustrates the DebiasedDTA training framework.

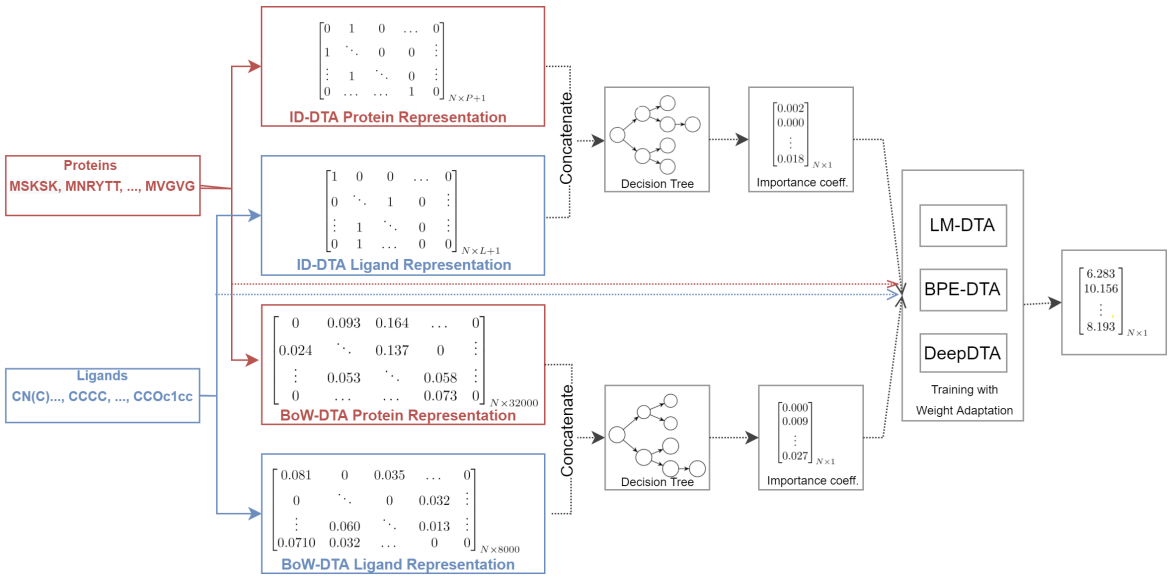


Figure 5.1. DebiasedDTA.

5.2.1. The Guide

The guides in DebiasedDTA are designed to learn merely dataset biases and should have limited learning capacity. So, we design two weak learners with simple biomolecule representations to identify different bias sources: an identifier-based model (ID-DTA) and a biomolecule word-based model (BoW-DTA). ID-DTA is motivated by the fact that mere use of random biomolecule identifiers can produce high-achieving models for similar test sets [70], and thus, can be a strong bias source. ID-DTA featurizes the interactions by concatenating the one-hot encoded vectors of chemicals and proteins. BoW-DTA, on the other hand, bases on natural language inference studies in which the use of certain words in a sentence produces a strong bias with its semantic label [142, 143]. Here, we investigate a similar bias in biomolecular sequences and create BoW-DTA. BoW-DTA represents the proteins and chemicals with bag-of-words vectors and concatenates their vectors to represent the interaction.

BoW-DTA segments the biomolecule sequences into their words via BPE vocabularies and this might create an inconsistency between BoW-DTA and the predictor, if the predictor uses different vocabularies. So, we fork BoW-DTA and create BoW-LM-DTA to use with LM-DTA, a predictor introduced in Section 5.2.2, which has different

vocabularies. BoW-LM-DTA adopts the same word segmentation strategy as LM-DTA and same vectorization method as BoW-DTA. ID-DTA, BoW-DTA, BoW-LM-DTA use decision tree regression for prediction, as decision trees have limited learning capacity and yet are effective to learn spurious patterns.

We adopt 5-fold cross-validation to quantify dataset biases with the guides. First, we randomly divide the training set into five folds and construct five different mini-training and mini-validation sets. We train the guide on each mini-training set and compute the squared errors of its predictions on the corresponding mini-validation set. One run of cross-validation yields one squared-error measurement per protein-chemical pair as each pair is placed in the mini-validation set exactly once. In order to better estimate the performance on each sample, we run the 5-fold cross-validation 10 times and obtain 10 error measurements per sample. We compute the median of the 10 squared errors and name it as the "importance coefficient" of a protein-chemical pair. If the affinity of pair is easily predictable via exploiting dataset biases, i.e. the guide has a low prediction error, then the pair might contain biasing patterns for DTA prediction models and has a low importance coefficient. Otherwise, the pair is more likely to contain generalizable information about binding affinity and has a high importance coefficient. The importance coefficients guide the training of the predictor.

5.2.2. The Predictor

In DebiasedDTA training framework, the predictor is the model to debias and use on the target protein-chemical pairs. The predictor is free to adopt any biomolecule representation, but have to be able to weight the training samples during training to comply with the weight adaptation strategy proposed in DebiasedDTA.

The proposed strategy initializes the training sample weights to 1 and updates them at each epoch such that the weight of each training sample converges to its importance coefficient at the last epoch. When trained with this strategy, the predictor attributes more importance to samples with less biasing patterns as the learning con-

tinues, that is the bias in the model decays over time. Our weight adaptation strategy is formulated as

$$\vec{w}_e = (1 - \frac{e}{E}) + \vec{i} \times \frac{e}{E} \tag{5.1}$$

where

w_e is the vector of training sample weights at epoch e

E is the number of training epochs

$\vec{i}$ is the importance coefficients vector.

In Equation (5.1), e/E increases as the training continues, and so does the impact of $\vec{i}$ on the sample weights. This ensures that the importance of samples with less biasing patterns is increased towards the end of training.

We implement three drug-target affinity prediction models to observe the performance of DebiasedDTA training framework with different predictors. The first one is DeepDTA [4], an influential affinity prediction model that uses SMILES strings of chemicals and amino-acid sequences of proteins to represent biomolecules. DeepDTA applies three layers of character-level convolutions over input sequences and uses a three-layered fully-connected neural network for prediction. Here, we slightly modify DeepDTA and treat chemical groups in the SMILES strings ([OH], [COH], [COOH] etc.) as a single token, while the original DeepDTA processes these groups as character-by-character, too.

In the second model, we alter DeepDTA to use biomolecular word-level convolutions, where the words are identified via BPE algorithm and name the resulting model BPE-DTA. We experiment with BPE vocabulary sizes of 8K, 16K, and 32K for SMILES and protein sequences and pick the combination of 8K-32K as it yields the highest scores on datasets of our previous studies [70]. We report the results for all vocabulary combinations in our GitHub repository for completeness.

Table 5.1. Average number of proteins, chemicals, and interactions per dataset split.

	Fold	# Proteins	# Chemicals	# Interactions
BDB	Train	403.4 ± 2.8	740.8 ± 19.46	17988.2 ± 646.45
	Validation	355.0 ± 5.62	170.0 ± 11.05	1494.2 ± 56.17
	Warm	354.4 ± 3.44	179.6 ± 5.28	1494.4 ± 56.32
	Cold Chemical	376.0 ± 4.38	84.8 ± 5.53	2448.8 ± 373.48
	Cold Protein	43.6 ± 2.15	264.8 ± 90.17	2360.0 ± 216.02
	Cold Both	41.4 ± 3.07	30.8 ± 11.92	274.6 ± 36.19
KIBA	Train	200.6 ± 1.36	1834.6 ± 6.41	77264.4 ± 814.94
	Validation	193.0 ± 1.67	1467.2 ± 23.75	6650.2 ± 69.53
	Warm	192.0 ± 3.16	1476.2 ± 17.7	6650.6 ± 69.1
	Cold Chemical	193.0 ± 2.45	140.0 ± 5.59	6810.0 ± 570.52
	Cold Protein	14.6 ± 0.8	1296.0 ± 179.09	6259.6 ± 1024.25
	Cold Both	14.0 ± 1.1	100.2 ± 14.55	468.6 ± 37.89

Third, we utilize ChemBERTa [61] and ProtBERT [59] to create another drug-target affinity prediction model, LM-DTA. LM-DTA vectorizes SMILES and amino-acid sequences via the language models and concatenates their vectors to represent the interaction. Finally, LM-DTA uses a two-layered fully connected neural network for prediction.

5.2.3. Experimental Settings

We use BDB and KIBA datasets (Chapter 4.2.1) and create five distinct train-test setups per dataset to evaluate the models. To create different setups, we cluster the proteins and chemicals in the datasets and randomly divide the clusters into two as "warm" and "cold". We interpret the warm clusters as already known biomolecules and the cold clusters as novel biomolecules. The dissimilarity of known and novel biomolecules is enforced by the clustering-based split.

To produce training and test sets from warm and cold biomolecule clusters, we first filter the interactions between proteins and chemicals in the warm clusters. We

use these interactions mainly as the training set, but also separate small subsets as "validation" and "warm test" sets. The validation fold is used to tune model hyper-parameters, whereas the warm test set is to evaluate models on the interactions between known biomolecules.

We create two more test sets called "cold chemical" and "cold protein", where the cold chemical test set consists of the interactions between chemicals in the cold cluster and proteins in the warm cluster. This test set is used to measure model performance in the scenarios in which new drugs are searched to target existing proteins. The cold protein test set is created similarly and used to evaluate models in the scenarios where existing drugs are searched to target a novel protein.

Last, we create a "cold both" test set, that is the set of interactions between the proteins and chemicals in the cold clusters. This is the most challenging test set of every setup, as both the proteins and the chemicals are unavailable in the training set. The average number of proteins, chemicals, and interactions in the training and test sets are reported in Table 5.1, alongside standard deviations.

To tune the hyper-parameters, we train models on the training set of each setup and measure the performance on the corresponding validation set. We pick the hyper-parameter combination that scores the lowest validation average mean squared error to predict the test set interactions.

5.3. Results

5.3.1. DebiasedDTA Boosts Drug-Target Affinity Prediction Models

We debias three DTA prediction models, namely DeepDTA, BPE-DTA, and LM-DTA with two debiasing approaches, BoW-DTA and ID-DTA, on BDB and KIBA datasets in DebiasedDTA training framework and report mean CI and R^2 on the test sets in Table 5.2 - Table 5.9, alongside standard deviations in parentheses.

Table 5.2. Model debiasing results on warm test set of BDB.

The Predictor	**The Guide**	**CI**	**R^2**
DeepDTA	None	0.888 (0.009)	0.781 (0.028)
	BoW-DTA	0.899 (0.004)	0.799 (0.013)
	ID-DTA	0.898 (0.005)	0.804 (0.011)
BPE-DTA	None	0.883 (0.006)	0.774 (0.013)
	BoW-DTA	0.888 (0.008)	0.781 (0.016)
	ID-DTA	0.891 (0.005)	0.777 (0.019)
LM-DTA	None	0.876 (0.005)	0.745 (0.011)
	BoW-DTA	0.882 (0.006)	0.762 (0.003)
	ID-DTA	0.883 (0.006)	0.758 (0.003)
	BoW-LM-DTA	0.884 (0.009)	0.761 (0.008)

The Overall Gain of Debiasing. We first examine the performance boost due to DebiasedDTA and compare the best DebiasedDTA score on each setup with no debiasing score. Table 5.10 reports the percent increase in CI and absolute increase in R^2 thanks to debiasing.

Table 5.10 demonstrates that in 17 of 24 ($\sim$71%) evaluation setups, at least one model trained in DebiasedDTA outperforms the non-debiased counterpart, highlighting the strength of the proposed training framework to boost DTA prediction performance. To show that the performance increase due to DebiasedDTA is statistically significant, we conduct a one-sided one-sample t-tests with the null hypotheses that mean CI and R^2 gains are 0. The statistical tests result in the rejection of the null hypothesis with p-value < 0.01, suggesting that DebiasedDTA boosts prediction performance in general, with 99% significance.

Table 5.3. Model debiasing results on cold chemical test set of BDB.

The Predictor	**The Guide**	**CI**	**R^2**
DeepDTA	None	0.687 (0.096)	0.039 (0.243)
	BoW-DTA	0.698 (0.037)	0.043 (0.108)
	ID-DTA	0.693 (0.058)	0.026 (0.109)
BPE-DTA	None	0.657 (0.083)	-0.143 (0.202)
	BoW-DTA	0.687 (0.082)	-0.091 (0.302)
	ID-DTA	0.692 (0.065)	-0.045 (0.252)
LM-DTA	None	0.688 (0.046)	-0.027 (0.175)
	BoW-DTA	0.688 (0.069)	-0.005 (0.169)
	ID-DTA	0.683 (0.067)	-0.016 (0.270)
	BoW-LM-DTA	0.662 (0.074)	-0.096 (0.227)

The improvement in performance due to debiasing is more evident in the cold test sets of BDB, because BDB is a more diverse dataset than KIBA. Since the BDB biomolecules are more diverse, the training biases are less applicable to the unknown test biomolecules and their elimination boosts the DTA prediction performance more than KIBA.

Table 5.10 also highlights that, training models in DebiasedDTA improves the performance on every warm test set, though it is mainly designed to boost DTA prediction on novel biomolecules. This shows that eliminating the training set biases helps models to better represent the known biomolecules, too.

Finally, Table 5.10 shows that debiasing improves the performance of all affinity prediction models in the study on at least one test setup. This emphasizes that DTA prediction models are susceptible to dataset biases irrespective of their input representation and the proposed training framework is powerful and abstract enough to eliminate biases in different biomolecule representation settings.

Table 5.4. Model debiasing results on cold protein test set of BDB.

The Predictor	**The Guide**	**CI**	**R^2**
DeepDTA	None	0.759 (0.006)	0.315 (0.049)
	BoW-DTA	0.777 (0.014)	0.351 (0.090)
	ID-DTA	0.771 (0.007)	0.339 (0.067)
BPE-DTA	None	0.653 (0.060)	-0.256 (0.411)
	BoW-DTA	0.664 (0.067)	-0.386 (0.593)
	ID-DTA	0.650 (0.039)	-0.689 (0.476)
LM-DTA	None	0.780 (0.016)	0.384 (0.083)
	BoW-DTA	0.781 (0.017)	0.386 (0.081)
	ID-DTA	0.782 (0.017)	0.387 (0.080)
	BoW-LM-DTA	0.784 (0.016)	0.395 (0.078)

Effect of The Guides. We investigate the effect of the guide selection in DebiasedDTA on the affinity prediction performance by comparing BoW-DTA models with ID-DTA. For BDB, models debiased with BoW-DTA yield higher scores in both metrics in 5 cases and ID-DTA based models outperform BoW-DTA 2 times in terms of CI and R^2. 5 out of 12 times, no guide can outscore the other in both metrics.

On KIBA, ID-DTA achieves higher CI and R^2 than BoW-DTA in 7 cases whereas BoW-DTA outperforms ID-DTA 4 times in terms of both metrics. The higher performance of ID-DTA on KIBA compared to BDB (7 wins vs. 2 wins) suggests that biomolecule identities cause more bias in this dataset. We relate this with the fact that KIBA contains more interactions per biomolecule and thus the models can infer more information about the biomolecule identities from the training set. In total, both guides outperform the other 9 times, indicating that the performance of ID-DTA and BoW-DTA is comparable to each other and both chemical word based and identity based biases are prevalent in the datasets.

Table 5.5. Model debiasing results on cold test set of BDB.

The Predictor	**The Guide**	**CI**	**R^2**
DeepDTA	None	0.554 (0.047)	-0.154 (0.164)
	BoW-DTA	0.568 (0.044)	-0.092 (0.132)
	ID-DTA	0.585 (0.040)	-0.128 (0.056)
BPE-DTA	None	0.522 (0.054)	-0.442 (0.349)
	BoW-DTA	0.568 (0.084)	-0.334 (0.347)
	ID-DTA	0.565 (0.090)	-0.426 (0.231)
LM-DTA	None	0.572 (0.028)	-0.226 (0.205)
	BoW-DTA	0.563 (0.032)	-0.182 (0.136)
	ID-DTA	0.581 (0.017)	-0.198 (0.174)
	BoW-LM-DTA	0.548 (0.033)	-0.244 (0.137)

Last, we examine the effect of using the same biomolecule vocabularies in the guide and predictor by comparing BoW-DTA and BoW-LM-DTA. BoW-LM-DTA, which uses the same vocabulary as LM-DTA, outperforms BoW-DTA based models on only 2 of 8 setups in Table 5.2 - Table 5.9, whereas BoW-DTA outscores BoW-LM-DTA on 4 setups. This shows that the guide and the predictor architectures do not have to be similar for a cohesive learning. Because, once the guides quantify the dataset biases, the predictors acquire all the information they need for weight adaptation – they become indifferent to the underlying computation.

5.3.2. DebiasedDTA Facilitates Out-of-Dataset Generalization

Having observed the strong prediction performance of DebiasedDTA on almost every test sets of BDB and KIBA, we further challenge the proposed methodology by out-of-dataset interactions. For out-of-dataset evaluation, we use the models trained on BDB to predict the affinity of all protein-chemical pairs in KIBA, and vice verse. Prior to prediction, we remove the SMILES-aminoacid sequence pairs shared between the datasets to eliminate risk of information leak from test set to training set.

Table 5.6. Model debiasing results on warm test set of KIBA.

The Predictor	**The Guide**	**CI**	**R^2**
DeepDTA	None	0.873 (0.005)	0.756 (0.021)
	BoW-DTA	0.888 (0.005)	0.775 (0.019)
	ID-DTA	0.887 (0.006)	**0.775 (0.018)**
BPE-DTA	None	0.881 (0.005)	0.760 (0.016)
	BoW-DTA	0.891 (0.003)	0.774 (0.016)
	ID-DTA	0.893 (0.003)	0.776 (0.012)
LM-DTA	None	0.858 (0.005)	0.756 (0.012)
	BoW-DTA	0.865 (0.005)	0.769 (0.013)
	ID-DTA	0.864 (0.006)	0.767 (0.014)
	BoW-LM-DTA	0.864 (0.005)	0.768 (0.012)

A remark for cross-evaluation is that BDB and KIBA report the affinity scores in terms of inconvertible metrics, and thus regression performance of the models on the cross-dataset cannot be evaluated. We convert both the model predictions and the affinity scores reported in the datasets to binary classes of strong- and weak-binding to overcome the inconsistency. $pK_d > 7$ in BDB and KIBA Score > 12.1 in KIBA are selected as the high-affinity threshold [70].

We utilize the previously trained models to predict cross-dataset interactions and use F1-score as the evaluation metric since labels are unevenly distributed (Figure 4.1). Table 5.11 reports the mean and standard deviation of the non-debiased and debiased models on cross-dataset and also presents in-dataset cold-both test set results as benchmark. For brevity, the best performing DebiasedDTA models are shown in Table 5.11 and the statistics for all DebiasedDTA models are presented in GitHub repository.

Table 5.7. Model debiasing results on cold chemical test set of KIBA.

The Predictor	**The Guide**	**CI**	**R^2**
DeepDTA	No Debiasing	0.753 (0.018)	0.337 (0.081)
	BoW-DTA	0.761 (0.004)	0.349 (0.046)
	ID-DTA	0.761 (0.020)	0.350 (0.101)
BPE-DTA	No Debiasing	0.735 (0.025)	0.274 (0.105)
	BoW-DTA	0.736 (0.018)	0.231 (0.093)
	ID-DTA	0.736 (0.021)	0.229 (0.099)
LM-DTA	No Debiasing	0.749 (0.012)	0.409 (0.067)
	BoW-DTA	0.756 (0.013)	0.435 (0.064)
	ID-DTA	0.759 (0.011)	0.436 (0.056)
	BoW-LM-DTA	0.758 (0.010)	0.441 (0.055)

Table 5.11 demonstrates that DebiasedDTA achieves a higher mean cross-dataset F1-score than the non-debiased models, except for the DeepDTA model trained on KIBA. The difference is the most significant for BPE-DTA trained on BDB, where a student's t-test also supports the superiority of DebiasedDTA with 0.99 significance. These suggest that DebiasedDTA can boost out-of-dataset generalization of the DTA prediction models.

Table 5.11 also displays the higher generalization capability of LM-DTA, as it achieves the highest performance on both datasets. We explain this with the pre-trained biomolecule embeddings in LM-DTA, which encompass information about millions of biomolecules through language modeling.

Another result in Table 5.11 is that, models trained on BDB perform better on KIBA than their in-dataset cold-both test sets. This is a consequence of BDB and KIBA sharing 201 proteins and BDB having a challenging cold-both test set due to its higher biomolecule diversity. This aligns with the finding in the previous sections that DebiasedDTA boosted BDB cold-both performance more than KIBA, again due to higher diversity.

Table 5.8. Model debiasing results on cold protein test set of KIBA.

The Predictor	**The Guide**	**CI**	**R^2**
DeepDTA	None	0.719 (0.029)	0.330 (0.109)
	BoW-DTA	0.713 (0.036)	0.308 (0.115)
	ID-DTA	0.725 (0.038)	0.333 (0.124)
BPE-DTA	None	0.680 (0.020)	0.185 (0.077)
	BoW-DTA	0.679 (0.030)	0.174 (0.103)
	ID-DTA	0.684 (0.023)	0.179 (0.060)
LM-DTA	None	0.713 (0.049)	0.366 (0.137)
	BoW-DTA	0.717 (0.051)	0.382 (0.139)
	ID-DTA	0.718 (0.053)	0.385 (0.143)
	BoW-LM-DTA	0.719 (0.054)	0.382 (0.145)

Overall, we observe that DebiasedDTA can boost performance not only on in-dataset test sets but also on other datasets. We also show that pre-trained language models can help to predict the affinities of novel biomolecules and the interactions between biomolecules dissimilar to the training set challenge the models the most.

5.3.3. Demonstrating the Effect of Model Debiasing on Input Features

The experiments show that DebiasedDTA can improve DTA prediction models with different biomolecule representations on similar and distant test sets. In this section, we focus on the underlying mechanisms to demonstrate the effect of model debiasing on input features. The debiasing setup of BoW-DTA and BPE-DTA is selected, since both models use biomolecule words and this can help to demonstrate the debiasing on word level. These models are re-trained on an arbitrary setup of BDB dataset and their predictions on the warm test set are examined.

Table 5.9. Model debiasing results on cold test set of KIBA.

The Predictor	**The Guide**	**CI**	**R^2**
DeepDTA	None	0.654 (0.019)	0.087 (0.099)
	BoW-DTA	0.639 (0.028)	0.045 (0.147)
	ID-DTA	0.660 (0.034)	0.084 (0.195)
BPE-DTA	None	0.605 (0.033)	-0.006 (0.117)
	BoW-DTA	0.604 (0.017)	-0.046 (0.082)
	ID-DTA	0.590 (0.014)	-0.037 (0.079)
LM-DTA	None	0.650 (0.041)	0.107 (0.122)
	BoW-DTA	0.653 (0.028)	0.159 (0.121)
	ID-DTA	0.652 (0.036)	0.151 (0.126)
	BoW-LM-DTA	0.646 (0.032)	0.139 (0.115)

In the described setup, the models predict binding affinities using biomolecule words as input features. Therefore, the effect of model debiasing on each biomolecule word's contribution to the predictions is studied. To quantify and study this effect, we use Gradient-weighted Class Activation Mapping (Grad-CAM) method, which was designed to measure the importance of each feature in object classification models [144]. Grad-CAM uses the gradient flow in the model to output the "attention coefficient" of each feature for each prediction. The attention coefficient reflects the feature's contribution to the prediction. We run Grad-CAM on debiased and non-debiased BPE-DTA models and acquire attention coefficients of biomolecular words for each prediction on the warm test set.

Table 5.10. The gains of debiasing on each test set of (top) BDB and KIBA (bottom).

	Warm		**Cold Chemical**		**Cold Protein**		**Cold Both**	
Model	**CI**	**R^2**	**CI**	**R^2**	**CI**	**R^2**	**CI**	**R^2**
DeepDTA	1.239%	0.023	1.601%	0.004	2.372%	0.036	5.596%	0.062
BPE-DTA	0.906%	0.007	5.327%	0.098	1.685%	-0.141	8.812%	0.108
LM-DTA	0.913%	0.017	0.000%	0.022	0.513%	0.011	1.573%	0.044
DeepDTA	1.718%	0.019	1.062%	0.013	0.834%	0.003	0.917%	-0.003
BPE-DTA	1.362%	0.017	0.136%	-0.045	0.588%	-0.006	-1.157%	-0.031
LM-DTA	0.816%	0.013	1.335%	0.032	0.842%	0.019	0.462%	0.052

First, we compare the maximum attention coefficients of protein and chemical words for each test set interaction, since dataset biases are likely to be attended more by presenting prediction shortcuts. The comparison shows for non-debiased BPE-DTA that in 85% of test set interactions, the most attended feature is a chemical word. This statistic decreases to 78% for debiased BPE-DTA, indicating that debiasing pushes models to attribute more importance to protein words. Therefore, DebiasedDTA is a step to learn more from the proteins, which is a known barrier to produce DTA prediction models successful on novel biomolecules [124, 125, 129, 145]. We also visualize the maximum attention coefficient distribution of protein and chemical words in Figure 5.2 in order to illustrate the higher attendance to the chemical words than protein words in both models and the push of model debiasing towards learning more from the proteins.

As model training and debiasing continues, models weights and thus the attention coefficients are updated. In order to observe the effect of model debiasing on attention coefficients, we study the protein-chemical pairs that utilized debiasing the most, with the intuition that the effect of debiasing should be more apparent in these instances. We visualize the attention coefficient of chemical words (since they are attended more) over training epochs and observe that attention coefficients of the most attended words tend to decrease with debiasing. The change in squared error and attention coefficients for a protein-chemical pair (UniProtID: Q6ZN16-PubChemCID: 9869779) is demonstrated in Figure 5.3.

Table 5.11. Binary evaluation of model debiasing on cross-datasets.

Training Dataset	**Model**	**Cold Both**	**Cross Dataset**
BDB	DeepDTA	0.122 (0.029)	0.146 (0.025)
	DebiasedDTA	0.126 (0.046)	0.152 (0.011)
	BPE-DTA	0.072 (0.059)	0.168 (0.040)
	DebiasedDTA	0.124 (0.099)	0.186 (0.042)
	LM-DTA	0.217 (0.107)	0.520 (0.031)
	DebiasedDTA	0.246 (0.103)	0.522 (0.021)
KIBA	DeepDTA	0.361 (0.141)	0.246 (0.021)
	DebiasedDTA	0.337 (0.137)	0.243 (0.037)
	BPE-DTA	0.291 (0.123)	0.190 (0.040)
	DebiasedDTA	0.225 (0.083)	0.217 (0.018)
	LM-DTA	0.384 (0.101)	0.286 (0.019)
	DebiasedDTA	0.391 (0.106)	0.289 (0.016)

Figure 5.3 displays that the non-debiased model pays attention to a pharmacologically unimportant word ("C(C)") during training, whereas the debiased model learns to reduce its importance and eventually achieves a lower error on the target pair. In this sense, the guide teaches the predictor to regularize certain biomolecular words as the training continues. Figure 5.3 also shows that the debiased model starts outputting stable predictions earlier in the training than the non-debiased model. We welcome this as a positive side-effect of debiasing, since it can allow to stop training in earlier steps and save compute time.

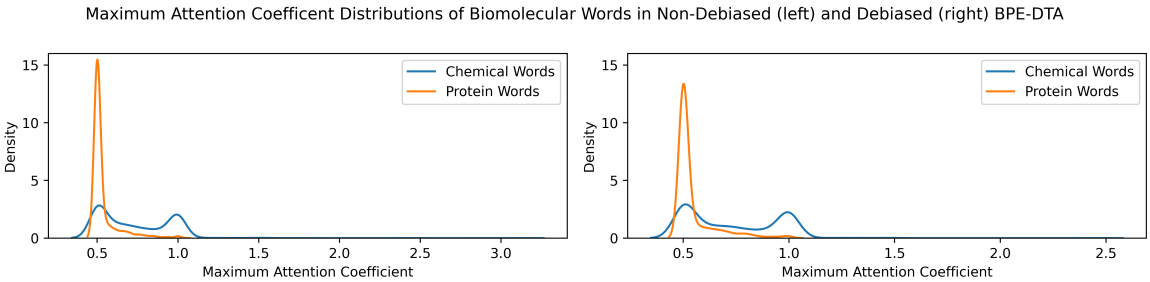


Figure 5.2. Distributions of maximum attention coefficients.

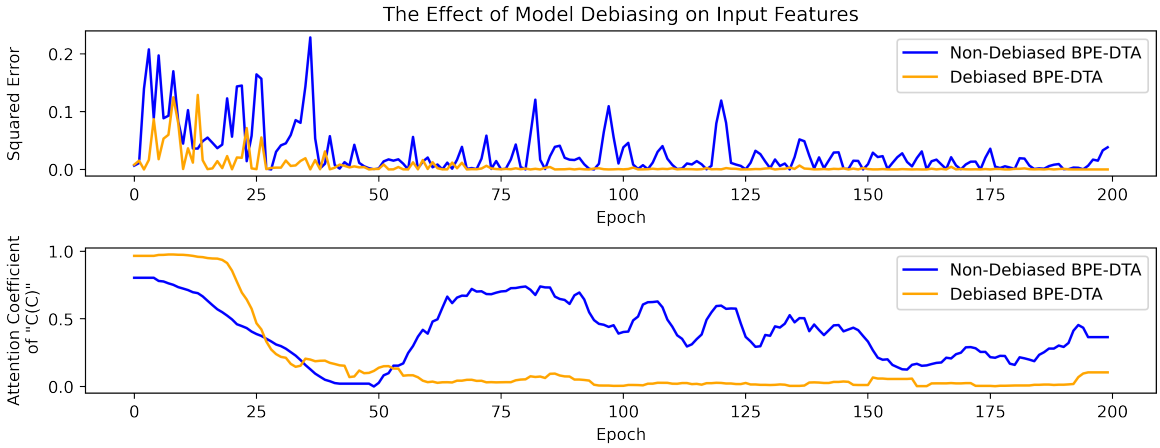


Figure 5.3. The effect of model debiasing on input features.

6. PYDTA: A PYTHON LIBRARY FOR DRUG-TARGET AFFINITY PREDICTION IN BIOMOLECULAR LANGUAGE

6.1. Motivation

Cheminformatics is an excellent meeting point for researchers of versatile backgrounds. The versatility in the field is a double-edged sword, though. On one end, it brings different perspectives to the table to solve the very same problem, while on the other end, it brings different tool usage habits that are not always compatible with each other. The difference in these habits can create high barriers while evaluating other perspectives/tools and hinder the interaction between disciplines. This makes it critical to communicate perspectives in accessible languages, such that researchers of any discipline can test, evaluate, and appreciate the wit of the ideas.

For researchers of computational background, like us, the versatility challenge motivates creating tools whose entry barriers are as low as possible. Such tools are downloaded and used tens of thousands of times [65, 146, 147]. In order to make our contribution, here we present a easy-to-use python library, pydta. pydta not only presents the DTA models developed in this thesis with an intuitive programming interface, but also contains building blocks to produce models from scratch. We aim pydta to be an ever-growing effort that eventually becomes the go-to library for biomolecular language processing. We open-source pydta to summon community support in our quest: `https://github.com/boun-tabi/pydta`.

6.2. The Library

pydta implements DTA models developed in this thesis; provides biomolecular language processing tools, and evaluation metrics. pydta consists of four modules: `data`, `evaluation`, `models`, and `utils`.

```
data
├── dta_sample_data
│   ├── dta_sample_data.json
│   └── dta_sample_data_mini.json
├── representation
│   └── smilesvec_chembl_8mer.kv
└── vocabs
    ├── chemical
    │   ├── chembl27_enc_94.json
    │   ├── chembl27_enc_bpe_8000.json
    │   └── chembl27_encoding.json
    └── protein
        ├── uniprot_26.json
        └── uniprot_bpe_32000.json
```

Figure 6.1. Directory tree of `data` module in pydta.

6.2.1. `data` Module

We proposed ChemBoost and BPE-DTA in Chapter 4 and Chapter 5, respectively. The former requires SMILESVec vectors to create chemical and protein vectors, while the latter relies on biomolecular vocabularies. `data` module contains these files and allows creating SMILESVec based biomolecular vectors and segmenting biomolecular sequences with pre-trained vocabularies. `data` module also contains uni-character vocabularies for biomolecular sequences and a dictionary that encodes SMILES tokens to allow respecting multi-character units during SMILES tokenization.

Finally, `data` module contains a sample DTA dataset for easy experimentation with the models. The directory structure of this module is shown in Figure 6.1

6.2.2. `evaluation` Module

This module contains only one file, `evaluation.py`, and provides standardized implementations for regression metrics for DTA. Concordance index, mean squared error, root mean squared error, and R^2 metrics are all implemented in this module and a generic evaluation function is provided, that is `evaluate_predictions(y_true, y_preds, metrics)`. This function returns a mapping from metric names to scores, given expected and predicted affinity values. The accepted metric names are `ci` for concordance index, `mse` for mean squared error, `rmse` for root mean squared error, and `r2` for R^2 score. All names are case insensitive.

6.2.3. `models` Module

`models` includes all strong and weak DTA models proposed throughout the thesis, namely: BoW-DTA, BPE-DTA, ChemBoost, DebiasedDTA, ID-DTA, and LM-DTA. The module also contains the official implementation of DeepDTA, as it inspired several models in this thesis and served as a benchmark several times. We illustrate the directory tree of this module in Figure 6.2.

BoW-DTA, LM-DTA, and DeepDTA are all deep learning models that are implemented in `tensorflow` [148]. Thus, they are trained and stored similarly and share large code parts. To avoid code duplication, we use the inheritance concept in object-oriented programming, and implement a base class named `TFModel`. `TFModel` implements functions to train, save, and load these models and predict the affinity of protein-chemical pairs. `TFModel` also provides abstract functions to build the model architecture, vectorize chemicals, and vectorize proteins, which are implemented by the child models. We aim `TFModel` to serve as a base class to custom DTA models in future releases, too.

```
models
├── bowdta.py
├── bpedta.py
├── chemboost.py
├── debiaseddta.py
├── deepdta.py
├── dta_model.py
├── iddta.py
└── lmdta.py
```

Figure 6.2. Python files under `models` module in pydta.

6.2.4. `utils` Module

Similar to `evaluation`, `utils` module also comprises a single file, `utils.py`. However, `utils.py` is as versatile and useful as a Swiss army knife. Besides containing generic utility functions that strip paths, finding the absolute path of a file, or casting a `numpy` array to a list, `utils.py` comprises functions to find units of a SMILES and a class named `HfWordIdentifier` that enables segmenting biomolecular sequences into biomolecular words in a vocabulary. Finally, `utils.py` allows loading the sample DTA dataset embedded in the library and creating uniform training weights for Debiased-DTA model. We plan to focus on developing these utility functions in future releases and create standalone modules to emphasize their importance.

6.2.5. Dependencies

pydta rises on the shoulders of the giants (existing libraries) in order to have reliable implementations. pydta is implemented in python and relies on `numpy` [149] for matrix operations. It leverages `tensorflow` [148] to implement deep models and `transformers` and `tokenizers` [150] for biomolecular word identification with BPE. It also uses `scikit-learn` [151] to implement evaluation metrics and `pytorch` [152] to obtain language model based biomolecule embeddings. Finally, `xgboost` [153] is used to implement XGBoost algorithm in ChemBoost. We hope to minimize these dependencies in future releases to shorten the installation time of the library. We list the current dependencies below with their versions.

```
from pydta.utils import load_sample_dta_data
from pydta.models import LMDTA

train_chemicals, train_proteins, train_labels =
    load_sample_dta_data(mini=True)['train']
lmdta = LMDTA(n_epochs=100)
lmdta.train(train_chemicals, train_proteins, train_labels)
```

Figure 6.3. Training LM-DTA in pydta.

- `python==3.7.9`
- `numpy==1.19.2`
- `scikit_learn==0.24.2`
- `xgboost==1.4.0`
- `tensorflow==2.3.0`
- `pytorch==1.8.1`
- `tokenizers==0.10.3`
- `transformers==4.10.0`

6.3. Installation and Code Examples

pydta is designed to have low entry barriers. So, it is very easy to install and use. pydta is already published as a python package in pip repositories and can be installed by a single command, assuming that python3 and pip3 are already installed:

```
pip3 install pydta
```

This command installs all dependencies and the library becomes ready to use in several minutes. After the installation, the models can be trained immediately. For instance, training an LM-DTA model on the toy dataset of pydta for 100 epochs is done with five lines displayed in Figure 6.3.

```
from pydta.utils import load_sample_dta_data
from pydta.models import BoWDTA, BPEDTA, DebiasedDTA

train_chemicals, train_proteins, train_labels =
    load_sample_dta_data(mini=True)['train']
debiaseddta = DebiasedDTA(BoWDTA, BPEDTA, predictor_params={'
    n_epochs': 100})
debiaseddta.train(train_chemicals, train_proteins,
    train_labels)
```

Figure 6.4. Debiasing BPE-DTA using BoW-DTA in pydta.

Debiasing is also as easy as it gets. The same amount of lines is required in order to train a debiased BPE-DTA using BoW-DTA. The code is displayed in Figure 6.4.

Training a ChemBoost model follows a similar pipeline but also requires a mapping from protein sequences to high affinity SMILES strings during initialization, due to the nature of the algorithm. Assuming such a mapping is stored in a variable already (`prot_to_sb_chemicals`), a ChemBoost model is created and evaluated with the script in Figure 6.5. Here we note that ChemBoost model in pydta concatenates SW and ligand-centric vectors to represent proteins and can correspond to Model (8) and Model (9) in Chapter 4, depending on the list of high affinity protein-chemical pairs.

Finally, pydta integrates with custom DTA prediction models, too, and offers a DebiasedDTA interface. To use DebiasedDTA, pydta enforces the custom prediction model to be implemented as a class that has an n_epochs attribute and a train method with arguments training_chemicals, training_proteins, training_labels, and sample_weights_by_epoch. DebiasedDTA imposes no restriction on the inner-workings of the train function and the content of the arguments. Figure 6.6 displays the template to debias a custom DTA prediction model with ID-DTA and more examples are available in the pydta repository.

```
from pydta.models import ChemBoost
chemboost = ChemBoost(prot_to_sb_chemicals=
    prot_to_sb_chemicals, n_estimators=1000)
chemboost.train(train_chemicals, train_proteins, train_labels)
preds = chemboost.predict(train_chemicals, train_proteins)
evaluate_predictions(train_labels, preds, ['r2'])
```

Figure 6.5. Training ChemBoost in pydta.

```
from pydta.models import IDDTA, DebiasedDTA
class CustomDTAModel:
    # The constructor can have other arguments and/or the
      class have other attributes.
    def __init__(self, n_epochs):
        self.n_epochs = n_epochs

    # The last argument will be filled by DebiasedDTA.
    def train(self, train_chemicals, train_proteins,
      train_labels, sample_weights_by_epoch):
        pass

train_chemicals, train_proteins, train_labels = [...], [...],
  [...]
debiaseddta = DebiasedDTA(IDDTA, CustomDTAModel,
  predictor_params={'n_epochs': 100})
debiaseddta.train(train_chemicals, train_proteins,
  train_labels)
```

Figure 6.6. Debiasing a custom model in pydta.

7. CONCLUSION

7.1. Contributions

The drug discovery pipeline takes more than a decade, and so does the discovery of new treatments. An important step in the drug discovery pipeline is to find high-affinity protein-chemical pairs to experiment in pre-clinical trials. However, the large number of possible protein-chemical combinations is experimentally invincible and challenges the search process. DTA prediction models help in this step by immediately highlighting the promising pairs *in silico* and speeding the pipeline. This thesis proposes novel DTA prediction models and training strategies in this quest, with the ultimate goal of discovering new drugs more quickly.

In this thesis, we exploit 1D representations of biomolecules. The major advantage of using sequence-based representations is that they are easier to obtain, unlike 3D structures, and easy to store and process, unlike 2D molecular graphs. In addition to being simpler and accessible, 1D sequences are information-rich and can empower models on par with 2D and 3D structure based models, if not superior [26].

We leverage biomolecular language processing to process 1D representations. Biomolecular language processing views biomolecular sequences as documents coded in biomolecular languages and adopts language processing methods. In order to demonstrate the validity of this perspective, we statistically evidence the existence of biomolecular languages and examine their components computationally and pharmacologically (Chapter 3).

A language consists of units such as letters, syllables, and words. However, such units are undefined in biomolecular languages and need identification. Chapter 3 shows that Byte Pair Encoding (BPE) and Unigram Language Model (ULM) can identify meaningful language components, "biomolecular words", that empower state-of-the-

art DTA prediction models. BPE and ULM perform comparably in the experiments, although we provide chemical information to ULM through start vocabularies. We recommend BPE instead of ULM for the sake of simplicity and further investigate its chemical words.

Statistical analysis on BPE words demonstrates a language-like structure and motivates pharmacologic evaluation. We collaborate with domain experts for this study and they find that BPE words can be markers of high affinity to protein families. The success of chemical words in computational and pharmacologic evaluation motivates our chemical language-based drug-target affinity prediction framework, ChemBoost.

Chapter 4 introduces ChemBoost, a chemical-word based affinity prediction framework. ChemBoost achieves state-of-the-art prediction performance, when ligands are represented through chemical words and proteins with the combination of SW and the chemical words of their high-affinity ligands. As chemical words are at the core of ChemBoost, we compare 8-mers with BPE words. The experiments show that 8-mers create better ligand and protein representations for affinity prediction and we recommend their usage as they are also simpler.

Rost proposes the presence of a "twilight zone" in sequence similarity, for which information about the protein can be predicted only in the presence of additional information [46]. We show that, SW is not able to accurately capture binding information when sequence similarity is low and a ligand-centric approach can improve model performance, especially for proteins with low sequence similarity. When used as a stand-alone representation, ligand-centric protein representation is more successful than SW at capturing functional similarities for low MSS interactions. We also observe that ligand-centric representations are more powerful when only high-affinity ligands are used and when the number of high affinity ligands per protein is higher.

The ligand-centric can be used in combination with orthogonal pieces of information for tasks ranging from fold prediction [154] to function annotation [155]. Model (9)

of ChemBoost, which uses both SW and ligand-centric vectors achieves state-of-the-art performance and is more robust to the changes in sequence similarity than both Model (1) (SW only protein representation) and the current state-of-the-art model, DeepDTA. However, ChemBoost models and DeepDTA struggle when the target pair contains at least one novel biomolecule. This encourages methods to boost the generalizability of DTA models.

Dataset bias is a major hurdle on the path to develop robust and generalizable machine learning models and one approach is to obtain a sampling from all knowledge space. However, protein-chemical interaction space is not sampled evenly, either because some protein targets are privileged due to their association with certain disease states, or because some chemicals or chemical moieties are privileged due to their relatively easier synthesis, or because the study of some interactions is experimentally infeasible. As some proteins or chemicals are over-represented, machine learning models tend to overfit and memorize these patterns and perform well when the training and test sets are similar to each other. However, it is difficult to learn generalizable patterns about protein-chemical interactions and machine learning methodologies fail when they are tasked with predictions about unseen biomolecules. In Chapter 5, we propose DebiasedDTA, a novel training framework that boosts the performance of DTA prediction methods both on known and unknown biomolecules. The performance improvement is observed for similar and distant test sets and underlines the value of DebiasedDTA.

DebiasedDTA owes the performance boost to the guides that are designed to identify specific types of bias sources. Here, we experiment with biomolecule word- and identity-driven biases and find that the elimination of either of the two can improve prediction performance. We also find that DebiasedDTA does not require a similarity in biomolecule representations of guides and predictors and can improve predictors of diverse architectures.

The predictors weight training samples for debiasing, which tunes the contribution of input features to the predictions. We show that elimination of biomolecule word biases pushes the models to learn more from the proteins and can reduce the effect of pharmacologically unimportant substructures to the predictions.

We investigate the consequences of debiasing on unknown biomolecules, too, and find that the more dissimilar the unknown biomolecules to the known ones, the more beneficial model debiasing is. We further challenge model debiasing with a cross-dataset evaluation setup, whose results underpins the merit of debiasing one more time.

The success and usability of the models in this thesis encouraged publishing an accessible tool. Chapter 6 introduces pydta, a python package that wraps up the methods presented in this thesis. pydta is available in pip repository and presents an intuitive interface for model development.

7.2. Future Directions

Biomolecular language processing is a promising perspective for DTA prediction task: it presents simple methods that yield high performance. We present further studies to overcome the limitations of the models in this thesis and explore the untapped potentials of language processing for drug discovery.

We need tools to ease developing biomolecular language processing pipelines. pydta is a step in this direction, which we will continue in the future. With the support of the community, we will turn pydta into a library that contains all building blocks to develop biomolecular language processing models for any drug discovery task, not only for DTA.

One attractive technique biomolecular language processing enables is to represent proteins through SMILES strings of their known ligands. What about the proteins with no known ligands, though? How can we represent them? In this thesis, we test

using the ligands of the most similar protein with a known ligand but the question is still unanswered. Answering this question would enable functionally more informative representations for novel proteins and support drug discovery studies on new targets and diseases.

Developing robust prediction models for novel biomolecules remains as one of the major problems in drug discovery. We relate this with dataset biases and introduce a new research direction, in which models are debiased to have stronger generalizability to novel biomolecules. DebiasedDTA is the first model debiasing approach and shows potential in the experiments. We view DebiasedDTA as a technique to prioritize informative training samples and believe that it will have implications on *de novo* molecule generation task, where out-of-distribution generalization is also an essential problem. This would allow us to traverse the unexplored regions of the chemical space and discover novel drugs.

REFERENCES

1. Imming, P., C. Sinning and A. Meyer, "Drugs, Their Targets and the Nature and Number of Drug Targets", *Nature Reviews Drug Discovery*, Vol. 5, No. 10, pp. 821–834, 2006.

2. Gashaw, I., P. Ellinghaus, A. Sommer and K. Asadullah, "What Makes a Good Drug Target?", *Drug Discovery Today*, Vol. 17, pp. S24–S30, 2012.

3. Vamathevan, J., D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer and S. Zhao, "Applications of Machine Learning in Drug Discovery and Development", *Nature Reviews Drug Discovery*, Vol. 18, No. 6, pp. 463–477, 2019.

4. Öztürk, H., A. Özgür and E. Ozkirimli, "DeepDTA: Deep Drug–Target Binding Affinity Prediction", *Bioinformatics*, Vol. 34, No. 17, pp. i821–i829, 2018.

5. Öztürk, H., E. Ozkirimli and A. Özgür, "WideDTA: Prediction of Drug-Target Binding Affinity", *arXiv preprint arXiv:1902.04166*, 2019.

6. Jiang, M., Z. Li, S. Zhang, S. Wang, X. Wang, Q. Yuan and Z. Wei, "Drug–Target Affinity Prediction using Graph Neural Network and Contact Maps", *RSC Advances*, Vol. 10, No. 35, pp. 20701–20712, 2020.

7. Abdel-Basset, M., H. Hawash, M. Elhoseny, R. K. Chakrabortty and M. Ryan, "DeepH-DTA: Deep Learning for Predicting Drug-Target Interactions: A Case Study of COVID-19 Drug Repurposing", *IEEE Access*, Vol. 8, pp. 170433–170451, 2020.

8. Lin, X., "DeepGS: Deep Representation Learning of Graphs and Sequences for Drug-Target Binding Affinity Prediction", *arXiv preprint arXiv:2003.13902*, 2020.

9. Zhao, L., J. Wang, L. Pang, Y. Liu and J. Zhang, "GANsDTA: Predicting Drug-Target Binding Affinity using GANs", *Frontiers in Genetics*, Vol. 10, p. 1243, 2020.

10. Nguyen, T. M., T. Nguyen, T. M. Le and T. Tran, "GEFA: Early Fusion Approach in Drug-Target Affinity Prediction", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

11. Shim, J., Z.-Y. Hong, I. Sohn and C. Hwang, "Prediction of Drug–Target Binding Affinity using Similarity-based Convolutional Neural Network", *Scientific Reports*, Vol. 11, No. 1, pp. 1–9, 2021.

12. Nguyen, T., H. Le, T. P. Quinn, T. Nguyen, T. D. Le and S. Venkatesh, "GraphDTA: Predicting Drug–Target Binding Affinity with Graph Neural Networks", *Bioinformatics*, Vol. 37, No. 8, pp. 1140–1147, 2021.

13. Sánchez-Cruz, N., J. L. Medina-Franco, J. Mestres and X. Barril, "Extended Connectivity Interaction Features: Improving Binding Affinity Prediction through Chemical Description", *Bioinformatics*, Vol. 37, No. 10, pp. 1376–1382, 2021.

14. Rifaioglu, A. S., R. Cetin Atalay, D. Cansen Kahraman, T. Doğan, M. Martin and V. Atalay, "MDeePred: Novel Multi-Channel Protein Featurization for Deep Learning-based Binding Affinity Prediction in Drug Discovery", *Bioinformatics*, Vol. 37, No. 5, pp. 693–704, 2021.

15. Abbasi, K., P. Razzaghi, A. Poso, S. Ghanbari-Ara and A. Masoudi-Nejad, "Deep Learning in Drug Target Interaction Prediction: Current and Future Perspectives", *Current Medicinal Chemistry*, Vol. 28, No. 11, pp. 2100–2113, 2021.

16. Öztürk, H., A. Özgür, P. Schwaller, T. Laino and E. Ozkirimli, "Exploring Chemical Space using Natural Language Processing Methodologies for Drug Discovery", *Drug Discovery Today*, Vol. 25, No. 4, pp. 689–705, 2020.

17. McCulloch, W. S. and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity", *The Bulletin of Mathematical Biophysics*, Vol. 5, No. 4, pp. 115–133, 1943.

18. Breiman, L., "Random Forests", *Machine Learning*, Vol. 45, No. 1, pp. 5–32, 2001.

19. LeCun, Y., Y. Bengio and G. Hinton, "Deep Learning", *Nature*, Vol. 521, No. 7553, pp. 436–444, 2015.

20. Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

21. Alpaydin, E., *Introduction to Machine Learning*, MIT Press, 2020.

22. Berman, H. M., 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.

23. Morgan, H. L., "The Generation of a Unique Machine Description for Chemical Structures-a Technique Developed at Chemical Abstracts Service.", *Journal of Chemical Documentation*, Vol. 5, No. 2, pp. 107–113, 1965.

24. Weininger, D., "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.

25. Jumper, J., R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, K. Pushmeet and D. Hassabis, "Highly Accurate Protein Structure Prediction with AlphaFold", *Nature*, Vol. 596, No. 7873, pp. 583–589, 2021.

26. Flam-Shepherd, D., K. Zhu and A. Aspuru-Guzik, "Keeping It Simple: Language Models can Learn Complex Molecular Distributions", *arXiv preprint arXiv:2112.03041*, 2021.

27. Gaulton, A., A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, M. Davies, N. Dedman, A. Karlsson, M. P. Magariños, J. P. Overington, G. Papadatos, I. Smit and A. R. Leach, "The ChEMBL Database in 2017", *Nucleic Acids Research*, Vol. 45, No. D1, pp. D945–D954, 11 2016.

28. Bhasin, M. and G. Raghava, "Computational methods in Genome Research", *Applied Mycology and Biotechnology*, Vol. 6, pp. 179–207, Elsevier, 2006.

29. Berman, H. M., 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, 01 2000.

30. Apweiler, R., A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi and L.-S. L. Yeh, "UniProt: The Universal Protein Knowledgebase", *Nucleic Acids Research*, Vol. 32, No. suppl_1, pp. D115–D119, 2004.

31. Berg, J. M., J. L. Tymoczko and L. Stryer, *Biochemistry*, New York: WH Freeman, 2002.

32. Öztürk, H., E. Ozkirimli and A. Özgür, "A Novel Methodology on Distributed Representations of Proteins using Their Interacting ligands", *Bioinformatics*, Vol. 34, No. 13, pp. i295–i303, 2018.

33. Asgari, E. and M. R. Mofrad, "Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics", *PloS One*, Vol. 10, No. 11,

p. e0141287, 2015.

34. Gage, P., "A New Algorithm for Data Compression", *The C Users Journal*, Vol. 12, No. 2, pp. 23–38, 1994.

35. Sennrich, R., B. Haddow and A. Birch, "Neural Machine Translation of Rare Words with Subword Units", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Association for Computational Linguistics, Berlin, Germany, Aug. 2016.

36. Nguyen, T. Q. and D. Chiang, "Transfer Learning across Low-Resource, Related Languages for Neural Machine Translation", *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 296–301, 2017.

37. Heinzerling, B. and M. Strube, "BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages", *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

38. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

39. Liu, H., Z. Dai, D. R. So and Q. V. Le, "Pay Attention to MLPs", *arXiv preprint arXiv:2105.08050*, 2021.

40. Asgari, E., A. C. McHardy and M. R. Mofrad, "Probabilistic Variable-Length Segmentation of Protein Sequences for Discriminative Motif Discovery (DiMotif) and Sequence Embedding (ProtVecX)", *Scientific Reports*, Vol. 9, No. 1, pp. 1–16, 2019.

41. Kawano, K., S. Koide and C. Imamura, "Seq2Seq Fingerprint with Byte Pair Encoding for Predicting Changes in Protein Stability upon Single Point Mutation", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 17, No. 5, pp. 1762–1772, 2019.

42. Huang, K., C. Xiao, L. M. Glass and J. Sun, "MolTrans: Molecular Interaction Transformer for Drug–Target Interaction Prediction", *Bioinformatics*, Vol. 37, No. 6, pp. 830–836, 10 2020.

43. Li, X. and D. Fourches, "SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning", *Journal of Chemical Information and Modeling*, Vol. 61, No. 4, pp. 1560–1569, 2021.

44. Wang, Y., Z.-H. You, S. Yang, X. Li, T.-H. Jiang and X. Zhou, "A High Efficient Biological Language Model for Predicting Protein–Protein Interactions", *Cells*, Vol. 8, No. 2, p. 122, 2019.

45. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality", *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

46. Rost, B., "Twilight Zone of Protein Sequence Alignments", *Protein Engineering, Design and Selection*, Vol. 12, No. 2, pp. 85–94, 02 1999.

47. Pennington, J., R. Socher and C. D. Manning, "Glove: Global Vectors for Word Representation", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

48. Le, Q. and T. Mikolov, "Distributed Representations of Sentences and Documents", *International Conference on Machine Learning*, pp. 1188–1196, PMLR, 2014.

49. Zhang, Y. and B. C. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide

to) Convolutional Neural Networks for Sentence Classification", *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 253–263, 2017.

50. Yang, Z., D. Yang, C. Dyer, X. He, A. Smola and E. Hovy, "Hierarchical Attention Networks for Document Classification", *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, 2016.

51. Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep Contextualized Word Representations", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, 2018.

52. Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "Roberta: A Robustly Optimized BERT Pretraining Approach", *arXiv preprint arXiv:1907.11692*, 2019.

53. McCann, B., J. Bradbury, C. Xiong and R. Socher, "Learned in Translation: Contextualized Word Vectors", *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6297–6308, 2017.

54. Sun, C., X. Qiu, Y. Xu and X. Huang, "How to Fine-Tune BERT for Text Classification?", *China National Conference on Chinese Computational Linguistics*, pp. 194–206, Springer, 2019.

55. Yamada, I., A. Asai, H. Shindo, H. Takeda and Y. Matsumoto, "LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6442–6454, 2020.

56. Jiang, H., P. He, W. Chen, X. Liu, J. Gao and T. Zhao, "SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2177–2190, 2020.

57. Wang, S., Y. Guo, Y. Wang, H. Sun and J. Huang, "SMILES-BERT: Large Scale Unsupervised Pre-training for Molecular Property Prediction", *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 429–436, 2019.

58. Fabian, B., T. Edlich, H. Gaspar, M. Segler, J. Meyers, M. Fiscato and M. Ahmed, "Molecular Representation Learning with Language Models and Domain-Relevant Auxiliary Tasks", *arXiv preprint arXiv:2011.13230*, 2020.

59. Elnaggar, A., M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik and B. Rost, "ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing", *bioRxiv*, 2020.

60. Vig, J., A. Madani, L. R. Varshney, C. Xiong, R. Socher and N. F. Rajani, "Bertology Meets Biology: Interpreting Attention in Protein Language Models", *arXiv preprint arXiv:2006.15222*, 2020.

61. Chithrananda, S., G. Grand and B. Ramsundar, "ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction", *arXiv preprint arXiv:2010.09885*, 2020.

62. Durant, J. L., B. A. Leland, D. R. Henry and J. G. Nourse, "Reoptimization of MDL Keys for Use in Drug Discovery", *Journal of Chemical Information and Computer Sciences*, Vol. 42, No. 6, pp. 1273–1280, 2002.

63. Rogers, D. and M. Hahn, "Extended-Connectivity Fingerprints", *Journal of*

Chemical Information and Modeling, Vol. 50, No. 5, pp. 742–754, 2010.

64. Smith, T. F. and M. S. Waterman, "Identification of Common Molecular Subsequences", *Journal of Molecular Biology*, Vol. 147, No. 1, pp. 195–197, 1981.

65. Landrum, G., *RDKit: Open-Source Cheminformatics*, 2006, `https://www.rdkit.org`, accessed in January 2022.

66. Yamanishi, Y., M. Araki, A. Gutteridge, W. Honda and M. Kanehisa, "Prediction of Drug–Target Interaction Networks from the Integration of Chemical and Genomic Spaces", *Bioinformatics*, Vol. 24, No. 13, pp. i232–i240, 2008.

67. Schwaller, P., T. Gaudin, D. Lanyi, C. Bekas and T. Laino, ""Found in Translation": Predicting Outcomes of Complex Organic Chemistry Reactions using Neural Sequence-to-Sequence Models", *Chemical Science*, Vol. 9, No. 28, pp. 6091–6098, 2018.

68. Kudo, T., "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates", *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, 2018.

69. Degen, J., C. Wegscheid-Gerlach, A. Zaliani and M. Rarey, "On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces", *ChemMedChem*, Vol. 3, No. 10, pp. 1503–1507, 2008.

70. Özçelik, R., H. Öztürk, A. Özgür and E. Ozkirimli, "ChemBoost: A Chemical Language Based Approach for Protein–Ligand Binding Affinity Prediction", *Molecular Informatics*, Vol. 40, No. 5, p. 2000212, 2021.

71. Zipf, G. K., "Human Behavior and the Principle of Least Effort", *Cambridge, Massachusetts: Addison-Wesley*, p. 1, 1949.

72. Merity, S., C. Xiong, J. Bradbury and R. Socher, "Pointer Sentinel Mixture Models", *arXiv preprint arXiv:1609.07843*, 2016.

73. Goldhahn, D., T. Eckart and U. Quasthoff, "Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages", *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pp. 759–765, 2012.

74. Budur, E., R. Özçelik, T. Gungor and C. Potts, "Data and Representation for Turkish Natural Language Inference", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8253–8267, 2020.

75. Schütze, H., C. D. Manning and P. Raghavan, *Introduction to Information Retrieval*, Vol. 39, Cambridge University Press Cambridge, 2008.

76. Sugrue, M. F., "Pharmacological and Ocular Hypotensive Properties of Topical Carbonic Anhydrase inhibitors", *Progress in Retinal and Eye Research*, Vol. 19, No. 1, pp. 87–112, 2000.

77. Supuran, C. T., A. Scozzafava and A. Casini, "Carbonic Anhydrase Inhibitors", *Medicinal Research Reviews*, Vol. 23, No. 2, pp. 146–189, 2003.

78. Supuran, C. T., "Carbonic Anhydrases: Novel Therapeutic Applications for Inhibitors and Activators", *Nature Reviews Drug Discovery*, Vol. 7, No. 2, pp. 168–181, 2008.

79. George, M., M. Rajaram and E. Shanmugam, "New and Emerging Drug Molecules against Obesity", *Journal of Cardiovascular Pharmacology and Therapeutics*, Vol. 19, No. 1, pp. 65–76, 2014.

80. Śledź, P. and A. Caflisch, "Protein Structure-based Drug Design: From Docking to Molecular Dynamics", *Current Opinion in Structural Biology*, Vol. 48, pp. 93–102, 2018.

81. Bosc, N., F. Atkinson, E. Felix, A. Gaulton, A. Hersey and A. R. Leach, "Large Scale Comparison of QSAR and Conformal Prediction Methods and Their Applications in Drug Discovery", *Journal of Cheminformatics*, Vol. 11, No. 1, p. 4, 2019.

82. Limongelli, V., "Ligand Binding Free Energy and Kinetics Calculation in 2020", *Wiley Interdisciplinary Reviews: Computational Molecular Science*, p. e1455, 2020.

83. Wang, R., X. Fang, Y. Lu and S. Wang, "The PDBbind Database: Collection of Binding Affinities for Protein-Ligand Complexes with Known Three-Dimensional Structures", *Journal of Medicinal Chemistry*, Vol. 47, No. 12, pp. 2977–2980, 2004.

84. Davies, M., M. Nowotka, G. Papadatos, N. Dedman, A. Gaulton, F. Atkinson, L. Bellis and J. P. Overington, "ChEMBL Web Services: Streamlining Access to Drug Discovery Data and Utilities", *Nucleic Acids Research*, Vol. 43, No. W1, pp. W612–W620, 2015.

85. Bleakley, K. and Y. Yamanishi, "Supervised Prediction of Drug–Target Interactions using Bipartite Local Models", *Bioinformatics*, Vol. 25, No. 18, pp. 2397–2403, 2009.

86. Öztürk, H., E. Ozkirimli and A. Özgür, "A Comparative Study of SMILES-based Compound Similarity Functions for Drug-Target Interaction Prediction", *BMC Bioinformatics*, Vol. 17, No. 1, p. 128, 2016.

87. Zhang, X., L. Li, M. K. Ng and S. Zhang, "Drug–Target Interaction Prediction by Integrating Multiview Network Data", *Computational Biology and Chemistry*, Vol. 69, pp. 185–193, 2017.

88. Peng, L., B. Liao, W. Zhu, Z. Li and K. Li, "Predicting Drug–Target Interac-

tions with Multi-Information Fusion", *IEEE Journal of Biomedical and Health Informatics*, Vol. 21, No. 2, pp. 561–572, 2017.

89. Wen, M., Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun and H. Lu, "Deep-Learning-Based Drug–Target Interaction Prediction", *Journal of Proteome Research*, Vol. 16, No. 4, pp. 1401–1409, 2017.

90. Gao, K. Y., A. Fokoue, H. Luo, A. Iyengar, S. Dey and P. Zhang, "Interpretable Drug Target Prediction Using Deep Neural Representation.", *IJCAI*, pp. 3371–3377, 2018.

91. Chu, Y., A. C. Kaushik, X. Wang, W. Wang, Y. Zhang, X. Shan, D. R. Salahub, Y. Xiong and D.-Q. Wei, "DTI-CDF: A Cascade Deep Forest Model towards the Prediction of Drug-Target Interactions based on Hybrid Features", *Briefings in Bioinformatics*, 2019.

92. Lee, I., 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.

93. Pahikkala, T., A. Airola, S. Pietilä, S. Shakyawar, A. Szwajda, J. Tang and T. Aittokallio, "Toward More Realistic Drug–Target Interaction Predictions", *Briefings in Bioinformatics*, p. bbu010, 2014.

94. He, T., 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, p. 24, 2017.

95. Gomes, J., B. Ramsundar, E. N. Feinberg and V. S. Pande, "Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity", *arXiv preprint arXiv:1703.10603*, 2017.

96. Jiménez Luna, J., M. Skalic, G. Martinez-Rosell and G. De Fabritiis, "K DEEP:

Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks.", *Journal of Chemical Information and Modeling*, 2018.

97. Stepniewska-Dziubinska, M. M., P. Zielenkiewicz and P. Siedlecki, "Development and Evaluation of a Deep Learning Model for Protein-Ligand Binding Affinity Prediction", *Bioinformatics*, Vol. 1, p. 9, 2018.

98. Zhang, H., L. Liao, K. M. Saravanan, P. Yin and Y. Wei, "DeepBindRG: A Deep Learning based Method for Estimating Effective Protein–Ligand Affinity", *PeerJ*, Vol. 7, p. e7362, 2019.

99. Karimi, M., 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, 02 2019.

100. Barcellos, M. P., C. B. Santos, L. B. Federico, P. F. d. Almeida, C. H. d. P. da Silva and C. A. Taft, "Pharmacophore and Structure-based Drug Design, Molecular Dynamics and Admet/Tox Studies to Design Novel Potential PAD4 Inhibitors", *Journal of Biomolecular Structure and Dynamics*, Vol. 37, No. 4, pp. 966–981, 2019.

101. Xue, W., J. Tian, X. S. Wang, J. Xia and S. Wu, "Discovery of Potent PTP1B Inhibitors via Structure-based Drug Design, Synthesis and in vitro Bioassay of Norathyriol Derivatives", *Bioorganic Chemistry*, Vol. 86, pp. 224–234, 2019.

102. Jaeger, S., S. Fulle and S. Turk, "Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition", *Journal of Chemical Information and Modeling*, 2017.

103. Mayr, A., G. Klambauer, T. Unterthiner, M. Steijaert, J. K. Wegner, H. Ceulemans, D.-A. Clevert and S. Hochreiter, "Large-Scale Comparison of Machine

Learning Methods for Drug Target Prediction on ChEMBL", *Chemical Science*, 2018.

104. Wu, Z., B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing and V. Pande, "MoleculeNet: A Benchmark for Molecular Machine Learning", *Chemical Science*, Vol. 9, No. 2, pp. 513–530, 2018.

105. Ying, Z., D. Bourgeois, J. You, M. Zitnik and J. Leskovec, "GNNExplainer: Generating Explanations for Graph Neural Networks", H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett (Editors), *Advances in Neural Information Processing Systems 32*, pp. 9240–9251, Curran Associates, Inc., 2019.

106. Garfield, E., "Chemico-linguistics: Computer Translation of Chemical Nomenclature", *Nature*, Vol. 192, No. 4798, pp. 192–192, 1961.

107. Cadeddu, A., E. K. Wylie, J. Jurczak, M. Wampler-Doty and B. A. Grzybowski, "Organic Chemistry as a Language and the Implications of Chemical Linguistics for Structural and Retrosynthetic Analyses", *Angewandte Chemie International Edition*, Vol. 53, No. 31, pp. 8108–8112, 2014.

108. Woźniak, M., A. Wołos, U. Modrzyk, R. L. Górski, J. Winkowski, M. Bajczyk, S. Szymkuć, B. A. Grzybowski and M. Eder, "Linguistic Measures of Chemical Diversity and the "Keywords" of Molecular Collections", *Scientific Reports*, Vol. 8, 2018.

109. Krallinger, M., O. Rabal, A. Lourenco, J. Oyarzabal and A. Valencia, "Information Retrieval and Text Mining Technologies for Chemistry", *Chemical Reviews*, Vol. 117, No. 12, pp. 7673–7761, 2017.

110. Boström, J., D. G. Brown, R. J. Young and G. M. Keserü, "Expanding the Medicinal Chemistry Synthetic Toolbox", *Nature Reviews Drug Discovery*, 2018.

111. Keiser, M. J., B. L. Roth, B. N. Armbruster, P. Ernsberger, J. J. Irwin and B. K. Shoichet, "Relating Protein Pharmacology by Ligand Chemistry", *Nature Biotechnology*, Vol. 25, No. 2, p. 197, 2007.

112. Hert, J., M. J. Keiser, J. J. Irwin, T. I. Oprea and B. K. Shoichet, "Quantifying the Relationships among Drug Classes", *Journal of Chemical Information and Modeling*, Vol. 48, No. 4, pp. 755–765, 2008.

113. Keiser, M. J., V. Setola, J. J. Irwin, C. Laggner, A. I. Abbas, S. J. Hufeisen, N. H. Jensen, M. B. Kuijer, R. C. Matos, T. B. Tran, R. Whaley, R. A. Glennon, J. Hert, K. L. H. Thomas, D. D. Edwards, B. K. Shoichet and B. L. Roth, "Predicting New Molecular Targets for Known Drugs", *Nature*, Vol. 462, No. 7270, pp. 175–181, 2009.

114. Öztürk, H., E. Ozkirimli and A. Özgür, "Classification of Beta-Lactamases and Penicillin Binding Proteins using Ligand-Centric Network Models", *PloS One*, Vol. 10, No. 2, p. e0117874, 2015.

115. Chen, T. and C. Guestrin, "Xgboost: A Scalable Tree Boosting System", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

116. Tang, J., 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.

117. Liu, T., 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, 2007.

118. Rehurek, R. and P. Sojka, "Gensim–Python Framework for Vector Space Modelling", *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, Vol. 3, No. 2, 2011.

119. Martin, A. C., C. A. Orengo, E. G. Hutchinson, S. Jones, M. Karmirantzou, R. A. Laskowski, J. B. Mitchell, C. Taroni and J. M. Thornton, "Protein Folds and Functions", *Structure*, Vol. 6, No. 7, pp. 875–884, 1998.

120. Gönen, M. and G. Heller, "Concordance Probability and Discriminatory Power in Proportional Hazards Regression", *Biometrika*, Vol. 92, No. 4, pp. 965–970, 2005.

121. Barelier, S., T. Sterling, M. J. O'Meara and B. K. Shoichet, "The Recognition of Identical Ligands by Unrelated Proteins", *ACS Chemical Biology*, Vol. 10, No. 12, pp. 2772–2784, 2015.

122. Feng, Q., E. Dueva, A. Cherkasov and M. Ester, "Padme: A Deep Learning-based Framework for Drug-Target Interaction Prediction", *arXiv preprint arXiv:1807.09741*, 2018.

123. Chaput, L., J. Martinez-Sanz, N. Saettel and L. Mouawad, "Benchmark of Four Popular Virtual Screening Programs: Construction of the Active/Decoy Dataset Remains a Major Determinant of Measured Performance", *Journal of Cheminformatics*, Vol. 8, No. 1, pp. 1–17, 2016.

124. Wallach, I. and A. Heifets, "Most Ligand-based Classification Benchmarks Reward Memorization rather than Generalization", *Journal of Chemical Information and Modeling*, Vol. 58, No. 5, pp. 916–932, 2018.

125. Sieg, J., F. Flachsenberg and M. Rarey, "In Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-based Virtual Screening", *Journal of Chemical Information and Modeling*, Vol. 59, No. 3, pp. 947–961, 2019.

126. Yang, J., C. Shen and N. Huang, "Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets", *Frontiers in Pharmacology*, Vol. 11, p. 69, 2020.

127. Scantlebury, J., N. Brown, F. Von Delft and C. M. Deane, "Data Set Augmentation Allows Deep Learning-Based Virtual Screening to Better Generalize to Unseen Target Classes and Highlight Important Binding Interactions", *Journal of Chemical Information and Modeling*, Vol. 60, No. 8, pp. 3722–3730, 2020.

128. Bietz, S., K. T. Schomburg, M. Hilbig and M. Rarey, "Discriminative Chemical Patterns: Automatic and Interactive Design", *Journal of Chemical Information and Modeling*, Vol. 55, No. 8, pp. 1535–1546, 2015.

129. Chen, L., A. Cruz, S. Ramsey, C. J. Dickson, J. S. Duca, V. Hornak, D. R. Koes and T. Kurtzman, "Hidden Bias in the DUD-E Dataset Leads to Misleading Performance of Deep Learning in Structure-based Virtual Screening", *PloS One*, Vol. 14, No. 8, p. e0220113, 2019.

130. Tran-Nguyen, V.-K., C. Jacquemard and D. Rognan, "LIT-PCBA: An Unbiased Data Set for Machine Learning and Virtual Screening", *Journal of Chemical Information and Modeling*, 2020.

131. Boyles, F., C. M. Deane and G. M. Morris, "Learning from the Ligand: Using Ligand-based Features to Improve Binding Affinity Prediction", *Bioinformatics*, Vol. 36, No. 3, pp. 758–764, 2020.

132. Sundar, V. and L. Colwell, "The Effect of Debiasing Protein–Ligand Binding Data on Generalization", *Journal of Chemical Information and Modeling*, Vol. 60, No. 1, pp. 56–62, 2019.

133. He, H., S. Zha and H. Wang, "Unlearn Dataset Bias in Natural Language Inference by Fitting the Residual", *EMNLP-IJCNLP 2019*, p. 132, 2019.

134. Clark, C., M. Yatskar and L. Zettlemoyer, "Don't Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases", *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4060–4073, 2019.

135. Sakaguchi, K., R. Le Bras, C. Bhagavatula and Y. Choi, "Winogrande: An Adversarial Winograd Schema Challenge at Scale", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp. 8732–8740, 2020.

136. Clark, C., M. Yatskar and L. Zettlemoyer, "Learning to Model and Ignore Dataset Bias with Mixed Capacity Ensembles", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 3031–3045, 2020.

137. Sanh, V., T. Wolf, Y. Belinkov and A. M. Rush, "Learning from Others' Mistakes: Avoiding Dataset Biases without Modeling Them", *International Conference on Learning Representations*, 2021.

138. Utama, P. A., N. S. Moosavi and I. Gurevych, "Mind the Trade-off: Debiasing NLU Models without Degrading the In-distribution Performance", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8717–8729, 2020.

139. Utama, P. A., N. S. Moosavi and I. Gurevych, "Towards Debiasing NLU Models from Unknown Biases", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7597–7610, 2020.

140. Bissoto, A., E. Valle and S. Avila, "Debiasing Skin Lesion Datasets and Models? not so fast", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 740–741, 2020.

141. Majumdar, P., R. Singh and M. Vatsa, "Attention Aware Debiasing for Unbiased Model Prediction", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4133–4141, 2021.

142. Gururangan, S., S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman and N. A. Smith, "Annotation Artifacts in Natural Language Inference Data", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112, 2018.

143. Poliak, A., J. Naradowsky, A. Haldar, R. Rudinger and B. Van Durme, "Hypothesis Only Baselines in Natural Language Inference", *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pp. 180–191, Association for Computational Linguistics, New Orleans, Louisiana, Jun. 2018.

144. Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization", *International Journal of Computer Vision*, Vol. 128, No. 2, p. 336–359, Oct 2019.

145. Scantlebury, J., N. Brown, F. Von Delft and C. M. Deane, "Data Set Augmentation Allows Deep Learning-Based Virtual Screening to Better Generalize to Unseen Target Classes and Highlight Important Binding Interactions", *Journal of Chemical Information and Modeling*, Vol. 60, No. 8, pp. 3722–3730, 2020.

146. Ramsundar, B., P. Eastman, P. Walters, V. Pande, K. Leswing and Z. Wu, *Deep Learning for the Life Sciences*, O'Reilly Media, 2019.

147. Huang, K., T. Fu, L. M. Glass, M. Zitnik, C. Xiao and J. Sun, "DeepPurpose: A Deep Learning Library for Drug-Target Interaction Prediction", *Bioinformatics*, 2020.

148. Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng, "TensorFlow: A System for Large-Scale Machine Learning", *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, p. 265–283, USENIX Association, USA, 2016.

149. Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T. E. Oliphant, "Array Programming with NumPy", *Nature*, Vol. 585, No. 7825, pp. 357–362, Sep. 2020.

150. Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest and A. M. Rush, "Transformers: State-of-the-Art Natural Language Processing", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Association for Computational Linguistics, Online, Oct. 2020.

151. Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine learning in Python", *the Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, 2011.

152. Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library",

H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett (Editors), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

153. Chen, T. and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, 2016.

154. Khor, B. Y., G. J. Tye, T. S. Lim and Y. S. Choong, "General Overview on Structure Prediction of Twilight-Zone Proteins", *Theoretical Biology and Medical Modelling*, Vol. 12, No. 1, p. 15, 2015.

155. Gana, R., S. Rao, H. Huang, C. Wu and S. Vasudevan, "Structural and Functional Studies of S-Adenosyl-L-Methionine Binding Proteins: A Ligand-Centric Approach", *BMC Structural Biology*, Vol. 13, No. 1, p. 6, 2013.