TEXT-BASED MACHINE LEARNING METHODOLOGIES FOR MODELLING DRUG-TARGET INTERACTIONS

by

Hakime Öztürk

B.S., Computer Engineering, Dokuz Eylül University, 2012

M.S., Computer Engineering, Boğaziçi University, 2014

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

Graduate Program in Computer Engineering
Boğaziçi University
2019

TEXT-BASED MACHINE LEARNING METHODOLOGIES FOR MODELLING DRUG-TARGET INTERACTIONS

APPROVED BY:

Assoc. Prof. Arzucan Özgür . . . .
(Thesis Supervisor)

Assoc. Prof. Elif Özkırımlı . . . .
(Thesis Co-supervisor)

Prof. Tunga Güngör . . . .

Prof. Attila Gürsoy . . . .

Prof. Pınar Yolum . . . .

Asst. Prof. Fatma Başak Aydemir . . . .

DATE OF APPROVAL:

to my family

ACKNOWLEDGEMENTS

> "Find a group of people who challenge and inspire you,
> spend a lot of time with them, and it will change your life."
>
> — Amy Poehler

First and foremost, I would like to express my deepest gratitude to my advisor Assoc. Prof. Arzucan Özgür and co-advisor Assoc. Prof. Elif Özkırımlı, who have became a part of my family throughout this journey. Thank you, for your excellent guidance, patience, kindness, the freedom you gave me to be who I am and teaching me how to become a scientist. I will always be grateful to you for the trust you put in me, and pushing me towards my personal and scientific growth. I feel very lucky to have worked with you and couldn't ask for a better team. I believe the future will bring many more Queen lyrics to share together!

I would like to thank my thesis committee members, Prof. Pınar Yolum and Prof Attila Gürsoy, for their valuable recommendations and encouragements in every stage of this thesis, Prof. Tunga Güngör and Asst. Prof. Fatma Başak Aydemir for spending their time on my thesis and their valuable contributions. I am grateful to Asst. Prof. Fatma Başak Aydemir for her guidance through the final stages of my doctoral studies.

I thank the academics of the CMPE department that I had a chance to learn from. I owe special thanks to Suzan Üsküdarlı,PhD, for being an inspiration for a chronic introvert like me, to break through my shell, to be more of myself, and to connect. I thank Prof. Ethem Alpaydın for his valuable recommendations on our works. I would like to thank Prof. Lale Akarun, Prof. Tuna Tuğcu, Prof. Cem Ersoy, and Prof. Ayşın B. Ertüzün, for showing me the importance of working towards the goals for not only individual, but also collective success in mind, with dedication and determination.

I thank the personnel of Technology Transfer Office (TTO); Bülent Üner, Dilek Akgün, Murat Akman, and Hülya Bozkurt for sharing their knowledge and experiences with me, to Didem, Tuğba, Selvi, Öznur, Ari, and Duygu for the pleasant work environment we shared. I thank Feyza Çelebi in TETAM who made us feel at home in Kandilli Campus.

I am very grateful to my lab-mate, room-mate and fellow survivor of Kandilli, Gönül Aycı, who has always been with me through the happy and the difficult times. Thank you, for the vibrant colors you brought into my world and showing me to those new sides to the life. I am very thankful for Gizem Keser, whose crazy existence has always been a welcome escape from the reality of the life. Thank you for the places, flavors and the music we discovered together. I would like to thank Özlem Basmaz, aka Özlemşi, for the joy she brought into my life with her fantastic stories nobody ever heard of and being my otaku comrade. Looking forward to that trip to Japan, my friend! I thank my room-mates over the years for the peaceful living environment we shared. I am also grateful for my long-standing friends who are always with me regardless of the distances.

I owe a huge thank you to all the past and present members of AILAB and the CMPE department. Each taught me invaluable things about life and science, which, in our cases, are mostly quite interwoven. I especially thank Mert Tiftikci, for helping me through many (non/)scientific problems with humor (and too complex sentences!); Şaziye Betül Özateş, Gizem Soğancıoğlu and İlknur Karadeniz Erol for their support from the very beginning; Arda Çelebi for sharing his wisdom; Rıza Özçelik for helpful discussions on discovering drugs; Gökçe Uludoğan and Selen Parlar for the liveliness (and the cookies) they bring to the lab, and Binnur Görer, Melce Hüsünbeyi, Göksu Öztürk, Çağıl Uluşahin, N. Özlem Şimşek, the assistants of CMPE150, CMPE140 and CMPE322 that I had the pleasure of working together, and many others not mentioned by name, for the conversations we shared over coffee and meals.

I also would like to thank the members of KB407 in the CHE department, Begüm Alaybeyoğlu, Kevser Kalyon, Begüm Yağcı, Elif Esvap, Merve Yüce, and Atakan Yüksel for their friendships, support, and keeping the server alive! I thank Efe Erçetin, Berfu Büyüköz, Mehmet Aziz Yırık and Mahmut Karaca for their collaborative effort in developing PLITOOL; Maciej Eder for sharing MCS vocabulary; Teodoro Laino and Philippe Schwaller for their contributions to the review article. I am grateful to Emine Ezel Cilek, for fangirling over fictional characters with me, the postcards she sent from all over the word and her advice on my (non/)scientific questions.

I am deeply grateful to my aunt, Kerime Yılmaz, for making me feel at home when I needed, and I thank my grandfather, my late grandmother, my cousins Erol Yılmaz and Şenol Yılmaz, and their families for their support. Your presence made me feel safe and comfortable all these years.

I would like to my grandpa, Mehmet Öztürk, for his endless support and love for us. Thank you, for being a perfect grandpa! You will always be an inspiration for me to a be better person. I am also grateful to my grandma, Habibe Öztürk, for her support in her own unique ways.

It is difficult to find the words that could fully express how grateful I am for my small family with big hearts; Şükriye Öztürk, Güner Öztürk and Samet Öztürk. You are amazing just the way you are and everyday, I feel overwhelmed by your kindness, generosity and broad-mindedness. Mom, you are my bestest! friend, my mentor, my anchor. I admire your curiosity and the researcher within you. Dad, you are my strength and my unwavering supporter. Thank you for the wings of bravery you passed to me. Samet, thank you for lighting up our lives with your presence, you are perfect, and I just love the way you shine.

I thank Simon & Garfunkel, whose words and melodies have motivated me to push through the hard times. Thank you, because **the fighter still remains!**

I gratefully acknowledge TÜBİTAK-BİDEB 2211-E scholarship program. I would also like to thank BÜVAK, ISMB, and TÜBİTAK 2224A for the financial aid in attending conferences and BAP project (12304) for providing the server I used in my studies. I thank TAM project for letting me use the study area in Kandilli Campus.

ABSTRACT

TEXT-BASED MACHINE LEARNING METHODOLOGIES FOR MODELLING DRUG-TARGET INTERACTIONS

The identification of novel interactions between proteins and drugs with computational methodologies constitutes a significant area of research. Most often, a drug can be re-purposed to target a novel protein which enables machine learning algorithms to learn from existing interactions to predict unknown interactions. The main goal of this thesis is to model the interactions between proteins and ligands (drug candidates) using their textual representations via machine/deep learning techniques. With that aim, we introduce a novel ligand representation approach and a novel protein representation approach as well as two prediction systems for identifying the strengths of the interactions between proteins and compounds (i.e., their binding affinities). The common theme of these studies is the use of textual representations of proteins (i.e., amino-acid sequences) and compounds (i.e., SMILES). A major advantage of text-based representations is that they are experimentally easier to obtain compared to the three-dimensional (3D) representations and therefore there are more protein/ligand text-based representations available than 3D representations. Furthermore, processing text-based representations is computationally less expensive compared to processing two-dimensional (2D) and 3D representations. We hypothesize that, much like natural languages, bio-chemical sequences have their own languages and processing these languages might reveal important insights about their characteristics. The application of Natural Language Processing (NLP) based approaches in tasks such as protein family/super-family clustering and protein-ligand binding affinity prediction achieved state-of-the-art performance. These results indicate that the textual forms of proteins and ligands can be used to formulate effective solutions to address different bioinformatics and cheminformatics problems.

ÖZET

PROTEİN-İLAÇ ETKİLEŞİMLERİNİN METİN TABANLI MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE MODELLENMESİ

Özgün protein-ilaç etkileşimlerinin hesaplamalı metotlar ile saptanması önemli bir araştırma alanıdır. Çoğunlukla, bir ilaç yeni bir proteini hedeflemek için yeniden amaçlandırılabilir. Böylece makine öğrenmesi algoritmaları mevcut protein-ilaç etkileşimlerinden öğrenerek özgün etkileşimleri tahminleyebilir. Bu tezin temel amacı, protein ve ligandların (ilaç adaylarının) aralarındaki ilişkiyi, metinsel gösterimlerini kullanarak makine/derin öğrenme teknikleri ile modellemektir. Bu amaçla, yeni bir ligand gösterim yöntemi ve yeni bir protein gösterim yöntemi ile protein ve kimyasalların aralarındaki bağlanma kuvvetini (bağlanma ilgisini) belirlemek için iki yeni tahminleme sistemi tanıtılmıştır. Bu çalışmaların ortak teması proteinlerin (amino-asit dizileri) ve kimyasalların (SMILES dizileri) metinsel gösterimlerinin kullanılmasıdır. Metinsel gösterim, üç-boyutlu (3D) gösterime göre deneysel olarak daha kolay elde edilebilen bir bilgidir. Bu nedenle, üç-boyutlu bilgiye göre çok daha fazla molekül için metinsel gösterim bulunabilmektedir. Bu durum protein ve kimyasallar ile çalışırken önemli bir avantaj oluşturmaktadır. Ayrıca, metin bazlı gösterimlerin işlenmesi, iki-boyutlu (2D) ve 3D gösterimler ile karşılaştırıldığında hesaplamalı olarak daha ucuzdur. Biz çalışmalarımızda, tıpkı doğal diller gibi, biyo-kimyasal dizilerin kendi dillerinin olduğunu, ve bu dillerin işlenmesinin biyo-kimyasal moleküllerin karakteristikleri hakkında önemli bilgileri ortaya çıkarabileceği varsayımında bulunuyoruz. Protein aile gruplandırılması ve protein-ligand bağlanma ilgisinin tahminlenmesi gibi problemler üzerindeki çalışmalarımız literatürde en iyi performansa ulaşmıştır. Bu sonuçlar, protein ve kimyasalların metinsel gösterimlerinin farklı biyoenformatik ve kimenformatik problemlerine etkili çözümler tasarlanmasında kullanılabileceğini göstermiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

IC_{50}	Half-maximal Inhibitory Constant
K_i	Inhibition Constant
K_d	Dissociation Constant
$\in$	is member of
$\mathbb{R}$	set of real numbers
y	actual value
$\hat{y}$	predicted value
γ	minimum loss reduction value

LIST OF ACRONYMS/ABBREVIATIONS

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
A50	Astral 50 dataset
ANN	Artificial Neural Networks
AUPR	Area Under Precision Recall Curve
BDB	BindingDB dataset
BLAST	Basic Local Alignment Tool
BPE	Byte Pair Encoding
CBOW	Continuous Bag of Words
CI	Concordance Index
CCNN	Combined Convolutional Neural Networks
CNN	Convolutional Neural Networks
D	Dimension
DL	Deep Learning
DNA	Deoxyribonucleic acid
DNN	Deep Neural Network
DT	Drug Target
DTI	Drug Target Interaction
FC	Fully Connected Layer
FFNN	Feed Forward Neural Network
FP	Fingerprint
ID	Identifier
IDF	Inverse Document Frequency
K	Thousand
LCN	Ligand-centric network models
LSTM	Long-short term Memory
M	Million

MCL	Markov Clustering Algorithm
MCS	Maximum Common Substructure
MSE	Mean Squared Error
ML	Machine Learning
NLP	Natural Language Processing
PDB	Protein Data Bank
P-L	Protein Ligand
PLITOOL	Protein-Ligand Interaction Tool
PPI	Protein-Protein Interactions
PubChemFP	PubChem Fingerprint
QSAR	Quantitative structure–activity relationship
ReLU	Rectified Linear Unit
RBM	Rectricted Boltzman Machines
RLS	Regularized Least Squares
RNN	Recurrent Neural Networks
SCOP	Structural Classification of Proteins
sid	SCOP stable domain identifier
SMARTS	SMiles ARbitrary Target Specification
SMILES	Simplified Line Entry Specification
S-W	Smith-Waterman
TF	Term-Frequency
TransClust	Transitive Clustering
t-SNE	t-Distributed Stochastic Neighbor Embedding
UniProt	The Universal Protein Resource
XGBoost	Extreme Gradient Boosting

1. INTRODUCTION

A drug is a chemical that binds to a specific target and modifies its function that has been proven to be related to pathophysiology of a disease [1,2]. A target is considered as a molecular structure, usually a protein, peptide or nucleic acid, with activity that is wanted to be regulated (targeted) by a drug [2]. Drugs bind to specific locations of a protein that are called "binding-sites" forming a protein - ligand complex. Each interaction has a binding affinity value indicated with measures such as half-maximal inhibitory constant (IC_{50}), inhibition constant (K_i) and disasociation constant (K_d) each describing the effectiveness of the binding. Figure 1.1 illustrates the three-dimensional (3D) structure of the *ampicillin* and New Delhi Metallo β-lactamase (NDM-1) complex.

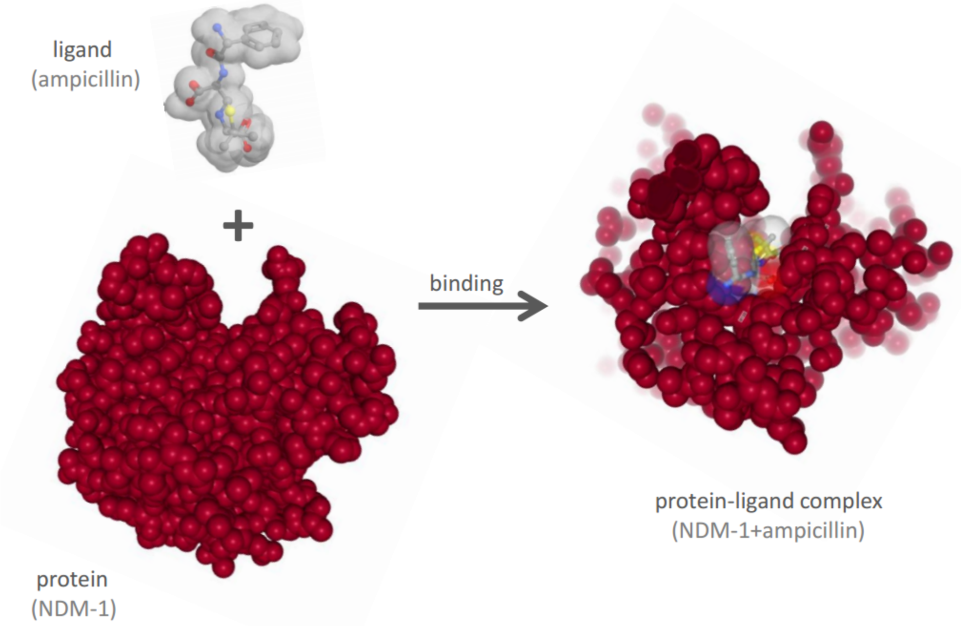


Figure 1.1: Illustration of the 3D-complex of NDM-1 and ampicillin.

Increase in the diversity of target proteins due to selective pressure and evolutionary process results in resistance against existing drugs, therefore causing the need to discover new active compounds. The development of novel drugs is an expensive and resource and time consuming process. Often referred to as *drug re-purposing* or *drug repositioning*, predicting new targets/uses for the existing drugs is an attractive alternative [3]. The polypharmacology approach, that has been coined to name drugs that can bind multiple targets, also supports the re-purposing of drugs [4,5]. Therefore, identification of novel drug-target interactions (DTI) holds a substantial place in drug discovery.

Several public interaction databases, such as ChEMBL [6], BindingDB [7], DrugBank [8], Matador [9] and STITCH [10] are available but the number of known interactions is still limited, since the experimental validation is costly and time consuming. Therefore, the application of computational methods to predict such interactions can limit the search space and suggest possible candidates, which can significantly accelerate the process and minimize *in vitro* efforts.

1.1. Problem Statement

Protein-ligand interactions have three main components: a ligand, a protein and a binding affinity value which indicates the strength of the interaction between a protein-ligand pair. Ligands and proteins can be described in three different representations: (i) one-dimensional (1D) representation refers to the textual representation of molecules, (ii) two-dimensional (2D) representation is a graph based form, and (iii) three-dimensional (3D) representation shows arrangement of the atoms in 3D space. Detailed information about these representation will be given in Section 2.

Representation of the ligands and proteins in the computational space is an important task, since they directly affect the performance of the tasks they contribute to. Fingerprints (FP) are the most widely-adopted representations for compounds, which are either rule-based binary vectors or domain-based hashed fingerprints. Among major drawbacks of rule-based fingerprints are: (i) the pre-defined rules might not cover the important aspects of a chemical, (ii) the rules need to be manually regulated, (iii) they often have high dimensions (e.g. 881, 1024 etc.). 2D graph representation has been used especially in computing similarity of compounds [11], however they are computationally more expensive compared to 1D representations. For proteins, on the other hand, amino-acid sequences are commonly used to build novel representations. Smith-Waterman (S-W) [12] is one of the popular algorithms to determine the similarity between a pair of proteins. However, there might be cases that proteins with low sequence similarity can show similar functional and mechanistic properties.

The modeling of the interactions between proteins and ligands has been mostly approached as a binary classification problem in which the proposed system predicted whether two entities interact(/bind) or not [13–19]. These supervised classification methodologies suffered from the lack of reliable benchmark datasets. Since the interaction databases only store binding information, absence of negative samples influenced the performance of the predictors. However, the available interaction data also contains binding affinity values which can be used to address the unrealistic negative interaction data problem. The binding affinity prediction problem has been first modelled with scoring functions (or parametric models) in which a set of parameters were used to characterize the protein-ligand interaction [20, 21]. A major drawback of this strategy is that there might be pairs that will not conform to these pre-defined formulations. Feature-based machine learning (ML) methods, on the other hand, proposed to learn from data in supervised manner [22–25] while integrating several feature extraction tools and algorithms. More recently, deep learning (DL) architectures have also been applied to the binding affinity prediction problem, which either used engineered features [26, 27] or learned representations directly from data, which was most often three-dimensional (3D) complex form of the interaction [28–30]. A disadvantage of such systems is that 3D form is not available for every possible protein-ligand pair.

In this thesis, we address the ligand representation, protein representation, and protein-ligand binding affinity prediction tasks. Our general aim is to propose solutions that are available for any protein-ligand pair and effective, yet computationally less expensive. Each of these three tasks pose various challenges which are explained in detail in the following section.

1.2. Challenges

The efficient modelling of drug-target interactions is a difficult task with the main challenges summarized as follows:

- **3D information is not available for every molecule.** The Protein Data Bank Bind (PDBBind) database [31] (accession date: October 2018) stores binding affinity values for around only 16K protein-ligand complexes. The ChEMBL database [32], on the other hand, comprises approximately 10M bioactivities for 12K targets and 1.9M compounds (accession date: June 2019). Thus, it is evident that chemical and protein spaces are much larger compared to the pairs that have 3D interaction information.
- **2D and 3D forms are computationally expensive to process.** 2D [11] and 3D [28–30] representations are expressive, however, require much more computational power than needed to process textual data.
- **Binding site is composed of non-consecutive amino acids.** The protein folds into a 3D shape in which the binding site may comprise residues from different regions in the amino acid sequence.
- **Boundaries of bio-chemical text is hard to identify.** Much like binding-sites of proteins, textual representations of proteins and chemicals have sub-sequences that might encode important information about the functionality/characteristics of these entities. Similar to languages such as Japanese and Chinese, however, boundaries of bio-chemical textual semantic units (i.e., words) are not known.
- **Sequence might not always be adequate to describe functional/mechanistic properties.** Even though amino-acids sequences are used to determine many important properties about a protein, sequence itself might not be enough for capturing mechanistic properties of a protein. For instance, proteins with low sequence similarity might share common ligands.
- **Modelling interaction between proteins and compounds are difficult.** The interaction between protein and ligand occurs at the binding-site, which indicates that it is a local event. Thus, the use of similarity information might not be enough to model the interaction. Similarly, the features that contribute to the interaction model should be designed carefully such that over-generalized representations are avoided.

1.3. Motivation

In this thesis, we investigate the representation of ligands and proteins, and the prediction of the interaction strength between these entities in an attempt to understand protein-ligand interactions. Considering the challenges of the field, we propose text-based machine/deep learning approaches to model proteins, ligands and their interactions. Textual representation of proteins and chemicals, unlike 3D-representations, are easier to obtain. Furthermore, processing textual data is less expensive than processing 2D/3D data. Both chemicals and proteins have their own set of characters and rules to construct their textual representations, which in turn, can be considered as biochemical languages. Thus, investigation of these sequences under the linguistic perspective might provide insights about their mechanisms as well as the interactions between them.

In our earlier works, we showed that SMILES text of a compound is not only as powerful as 2D, but also faster in computing compound similarity [33]. Therefore, we were motivated to build a novel ligand representation by utilizing simple, yet computationally less expensive SMILES text. Based on the analogy between a SMILES text and a document, we identified "words" of the chemical space (i.e. chemical words). Words were identified using different techniques which are k-mers (i.e. 8-mers), Byte Pair Encoding (BPE) and Maximum Common Substructures (MCS). A large SMILES corpus was used to learn distributed embeddings for each chemical word, which were then used to build the compound vector, SMILESVec.

A pioneering study by Keiser and co-workers [34] reported that similarity of interacting ligands can be used to detect protein similarity. We also showed that connecting proteins through their interacting ligands in a network resulted in groups of proteins with functional and sequence similarities [35]. Thus, in the light of these works, we proposed a ligand-based protein representation in which proteins are represented via their interacting ligands. A protein was described as the average of the SMILESVecs vectors of its interacting ligands. Unlike amino-acid sequences, such representation has the ability of encoding ligand-binding behaviour of the protein directly.

Finally, we proposed two binding affinity prediction systems that depend on textual descriptions of proteins and ligands. In the first system, we adopted a “chemical word” based approach, in which both proteins and compounds are represented via SMILES text. We further investigated the effect of different “chemical word” types on the prediction performance. In the second system, instead of explicitly defining words, we let a Convolutional Neural Network (CNN) based system to learn word boundaries from the sequences themselves. The proposed approach, DeepDTA, aimed to predict binding affinities through learning representations from SMILES and protein sequences.

This thesis introduces a complete system for modelling drug-target interactions, in which protein and ligand representations can be used in any bio/cheminformatics task that requires to describe biochemical data. The proposed prediction systems, although evaluated on benchmark datasets, can be utilized and/or modified for novel datasets.

1.4. Publication Notes

Parts of the work in this thesis have appeared in the following publications:

(i) “A novel methodology on distributed representations of proteins using their interacting ligands.” Öztürk, Hakime, Elif Ozkirimli, and Arzucan Özgür. Bioinformatics 34.13 (2018): i295-i303. (Chapters 4 and 5)

(ii) “DeepDTA: deep drug–target binding affinity prediction.” Öztürk, Hakime, Arzucan Özgür, and Elif Ozkirimli. Bioinformatics 34.17 (2018): i821-i829. (Chapter 6)

(iii) “A chemical language based approach for protein-ligand interaction prediction.” Öztürk, Hakime, Arzucan Özgür, and Elif Ozkirimli. arXiv preprint arXiv:1811.00761 (2018). *in preparation* (Chapter 6)

(iv) “Exploring the Chemical Space using Natural Language Processing Methodologies for Drug Discovery”, Öztürk, Hakime, Arzucan Özgür, Philippe Schwaller, Teodoro Laino and Elif Ozkirimli. *Submitted to Drug Discovery Today* (2019) (Chapter 2)

The other works that are not part of this thesis are:

(i) “WideDTA: prediction of protein-ligand binding affinity.”Öztürk, Hakime, Arzucan Özgür, and Elif Ozkirimli. arXiv preprint, (2019), arXiv:1902.04166.

(ii) “BIOSSES: a semantic sentence similarity estimation system for the biomedical domain.” Soğancıoğlu, Gizem, Hakime Öztürk, and Arzucan Özgür. Bioinformatics 33.14 (2017), i49-i58.

(iii) “CNN based chemical-protein interactions classification.” Yüksel, A., Öztürk, H., Ozkirimli, E., and Özgür, A. In Proceedings of the BioCreative VI Workshop, Bethesda, MD. 201 (2017), pp. 184-186.

(iv) “Construction of miRNA-miRNA networks revealing the complexity of miRNA-mediated mechanisms in trastuzumab treated breast cancer cell lines.” Cilek, E. E., Ozturk, H., and Dedeoglu, B. G. PloS one, 12(10), (2017), e0185558.

(v) “A comparative study of SMILES-based compound similarity functions for drug-target interaction prediction.” Öztürk, Hakime, Elif Ozkirimli, and Arzucan Özgür. BMC bioinformatics, 17(1), (2016), 128.

(vi) “Classification of Beta-lactamases and penicillin binding proteins using ligand-centric network models.” Öztürk, Hakime, Elif Ozkirimli, and Arzucan Özgür. PloS one, 10.2 (2015), e0117874.

1.5. Thesis Overview

In this thesis, we focused on the modelling of drug-target interactions through their textual representations. We first build a text-based representation for ligands utilizing their SMILES, and then propose a ligand-based approach to represent proteins with their interacting ligands. We finally introduced two systems to predict binding affinities for drug-target pairs: (i) a machine-learning based approach which combines the protein and ligand representations we introduced and, (ii) a deep-learning based approach that learns abstract features from the raw textual bio/chemical data (i.e. amino-acid sequences and SMILES for proteins and ligands, respectively). Figure 1.2 demonstrates the brief summary of this thesis. First, a text-based representation that we referred to as SMILESVec is introduced. Then, this representation was utilized to represent interacting ligands of the proteins, which in turn build the protein representation.

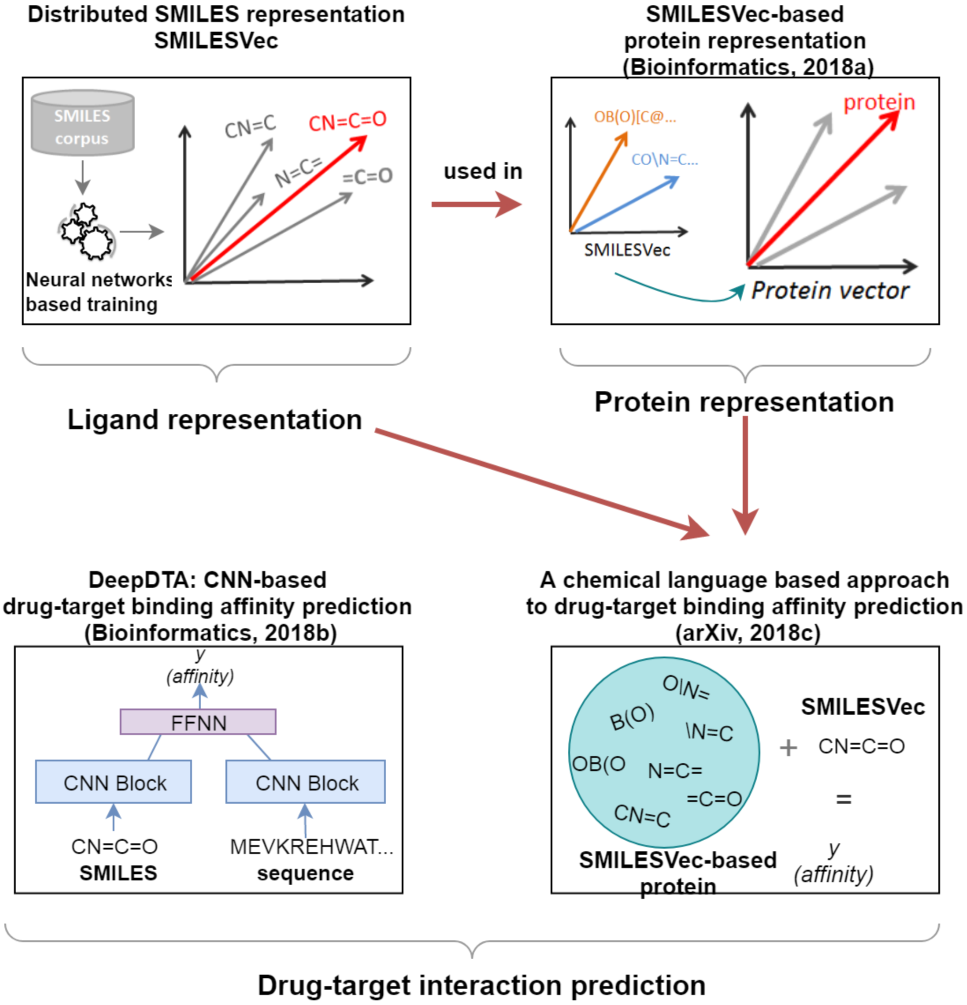


Figure 1.2: Thesis overview. A novel ligand representation, a novel protein representation and two novel binding affinity prediction systems are introduced.

The main contributions of this thesis are summarized as follows:

(i) Four novel completely text-based systems to model ligand representation, protein representation and protein-ligand interaction are introduced. The major advantage of the proposed systems is that the text information is available for every molecule.

(ii) The textual representations of the proteins and ligands are investigated as biochemical languages. The presented test cases in protein family clustering and protein-ligand binding affinity prediction tasks show that these languages are rich in terms of describing these entities.

(iii) A novel data-driven approach to represent ligands using their SMILES, *SMILESVec*, is proposed. Without integrating external rules or expert knowledge, SMILESVec directly learns representations from SMILES text. A data-driven approach has the flexibility of generating task specific representations, unlike universal fingerprint based ligand representations (Chapter 4) [36].

(iv) A novel ligand-based protein representation, which aims to capture functional and mechanistic properties of the proteins, is proposed. To describe the proteins, the SMILESVec representations of their interacting ligands are utilized. Proteins are successfully represented via their interacting ligands without using protein sequence/structure information. The proposed system captures relationships between proteins with similar binding properties, even if they have low sequence similarities (Chapter 5) [36].

(v) A language inspired protein-ligand binding affinity prediction system, *ChemBoost*, is proposed. ChemBoost depends on "chemical words" to describe both ligands and their interacting proteins. The "chemical word" based prediction system provides either similar or better performances when compared to state-of-the-art machine learning systems that utilize protein sequences and other additional features (Chapter 6) [37].

(vi) A novel deep-learning based model named *DeepDTA* to predict drug-target binding affinity, which uses only character representations of proteins and drugs, is introduced. Instead of explicitly describing words, DeepDTA integrates Convolutional Neural Networks (CNN) to extract abstract features from whole sequences of proteins and ligands. DeepDTA outperforms feature-based state-of-the-art machine learning systems (Chapter 6) [38].

(vii) Different "chemical word" identification techniques and their effect on the performance of binding affinity prediction are investigated. (Chapter 6) [37].

(viii) Two Python packages for SMILESVec [36] and DeepDTA [38], and an online tool named PLITOOL to collect protein-ligand interactions to visualize them in a ligand-centric way [35] using SMILESVec are made available to the public. (Chapter 7).

2. BIOLOGICAL BACKGROUND

2.1. Chemicals

A compound is a substance that is formed when two or more chemical elements are bonded together [39]. A drug, on the other hand, is a substance that binds to a specific target (which is usually a protein, peptide or nucleid acid) to modify its function that is proved to be related to pathophysiology of a disease [1,2]. BindingDB describes drugs as small molecules “which are nonpolymer, organic compounds with molecular weights around less than 1000 Da” [39]. To be accepted as a drug, a compound must bind to a target and fullfill some requirements such as being chemically and physically stable, non-toxic etc.

Chemical structures can be represented in different forms including one-dimensional (1D), 2D, and 3D. Figure 2.1 illustrates the different forms of drug *ampicillin*. 1D form of the chemicals often encode information such as atom counts, bond counts, molecular weight in the form of textual representation using characters. 2D forms of compounds are depicted in graph form in which atoms are represented as the nodes of the graph whereas bonds and branches are represented as links (edges) of the graph. 3D description contains information of coordinates of the atoms and bonds [40].

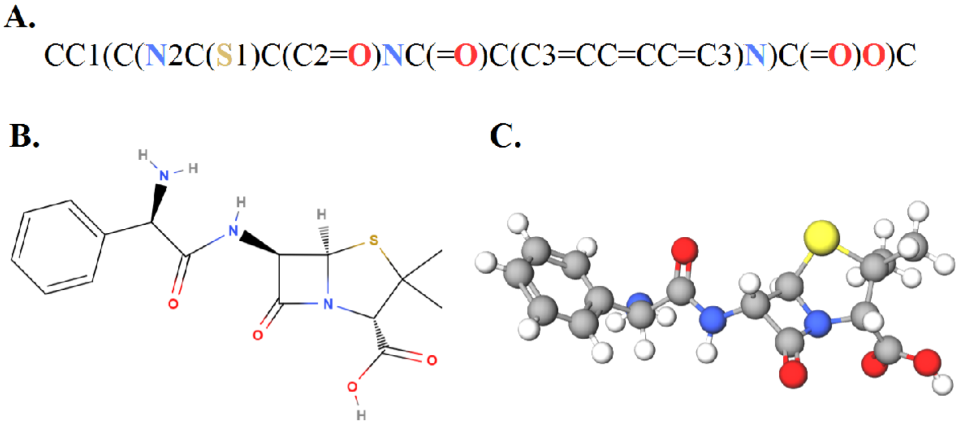


Figure 2.1: Illustration of different forms of ampicillin. (A) SMILES text. (B) 2D graph. (C) 3D form.

While the 2D and 3D representations are routinely used in ML based approaches [40], here we focus on the 1D form. Table 2.1 depicts different textual identifiers and representations of the drug *ampicillin*. The data is collected from PubChem database, and 2D and 3D figures are generated using MolView [41].

Table 2.1: Different textual identifications of drug *ampicillin*.

Identifier	**Representation**
IUPAC name	(2S,5R,6R)-6-[[(2R)-2-amino-2-phenylacetyl]amino]-3,3-dimethyl-7-oxo-4-thia-1-azabicyclo[3.2.0]heptane-2-carboxylic acid
Chemical Formula	$C_{16}H_{19}N_3O_4S$
Canonical SMILES	CC1(C(N2C(S1)C(C2=O)NC(=O)C(C3=CC=CC=C3)N)C(=O)O)C
Isomeric SMILES	CC1([C@@H](N2[C@H](S1)[C@@H](C2=O)NC(=O)[C@@H](C3=CC=CC=C3)N)C(=O)O)C
DeepSMILES (Canonical)	CCCNCS5)CC4=O))NC=O)CC=CC=CC=C6))))))N)))))))C=O)O)))C
InChi	InChI=1S/C16H19N3O4S/c1-16(2)11(15(22)23)19-13(21)10(14(19)24-16)18-12(20)9(17)8-6-4-3-5-7-8/h3-7 9-11,14H,17H2,1-2H3,(H,18,20)(H,22,23)/t9-,10-,11+ 14-/m1/s1
InChi Key	AVKUERGKIZMTKX-NJBDSQKTSA-N

IUPAC name. The International Union of Pure and Applied Chemistry (IUPAC) scheme (i.e. nomenclature) is used to name compounds following pre-defined rules such that the names of the compounds are unique and consistent with each other [42].

Chemical Formula is one of the simplest and most widely-known ways of describing chemicals using letters (i.e. element symbols), numbers, parentheses, and (-/+) signs. This representation gives information about which elements and how many of them are present in the compound.

InChI is the IUPAC International Chemical Identifier, which is a non-proprietary and open-source structural representation [43]. The InChIKey is a character-based representation that is generated by hashing the InChI strings in order to shorten them. Since the software that generates InChi is publicly available, InChi does not suffer from ambiguity problems. InChi representation has several layers (each) separated by the "/" symbol.

In Section 2.1.1, we will discuss Simplified Molecular Input Entry Specification (SMILES) representation in detail since it constitutes one of the main inputs of the approaches proposed in this thesis.

2.1.1. SMILES

Simplified Molecular Input Entry Specification (SMILES) is a text-based form of describing molecular structures and reactions [44, 45]. SMILES is constituted by a set of rules in which atoms, bonds and other components of the molecule are represented with specific symbols. Table 2.2 lists the common specialized characters in a SMILES string.

Table 2.2: Most frequent SMILES symbols [46].

no	**symbol**	**definition**
1	C	nonaromatic carbon atoms
2	c	aromatic carbon atoms
3	N	nonaromatic nitrogen atoms
4	n	aromatic nitrogen atoms
5	O	nonaromatic oxygen atoms
6	o	aromatic oxygen atoms
7	S	nonaromatic sulfur atoms
8	s	aromatic sulfur atoms
9	F	fluorine atoms
10	Cl	chlorine atoms
11	Br	bromine atoms
12	I	iodine atoms
13	P	nonaromatic phosphorus atoms
14	p	aromatic phosphorus atoms
15	B	boron atoms
16	“X”	any other character
17	−	single bonds
18	=	double bonds
19	#	triple bonds
20	[	Nonorganic elements, charges, isotopes, protonation states
21	-	negative charges
22	+	positive charges
23	H	explicit hydrogen atoms
24	(	acyclic branching points
25	1	nonfused ring systems
26	2	bicyclic systems
27	3	tricyclic systems
28	4	tetracyclic systems
29	5	pentacyclic systems
30	6	hexacyclic systems
31	7	heptacyclic systems
32	8	octacyclic systems
33	9	nonacyclic systems
34	%	higher order ring systems

Daylight Chemical Information Inc indicates that with its own vocabulary and the limited set of rules, SMILES notation is indeed a language, rather than simply being a computational data. Being in a textual form, SMILES takes 50% to 70% less space than other representation methods (e.g. an identical connection table). More detail is available at Daylight [47].

Nevertheless, SMILES also provides more complex information than the chemical formula. We can obtain SMILES strings through traversing the 2D graph representation of the compound [48]. Here will provide a brief introduction to the basic properties of SMILES language.

Atoms and bonds. Atoms are represented with their atomic symbols and should be enclosed in square brackets except for the atoms that belong to the organic subset (B, C, N, O, P, S, F, Cl, Br, and I). Upper-case letters are used to represent non-aromatic atoms whereas lower-case letters are used for aromatic atoms (e.g., C and c (Table 2.2). Hydrogen atoms (H) can be omitted.

Bonds between the atoms are described with “-”, “=”, ”#” and “:” symbols denoting the single, double, triple and aromatic bonds, respectively. Single bonds (“-”) and aromatic bonds (“:”) among consecutive atoms are usually omitted since they are the default interaction type [49].

Branches, cycles and disconnected substructures. Parenthesis in SMILES string indicate branches (e.g. triethylamine, *CCN(CC)CC*) and disconnected substructures in a molecule is depicted with “.”. Cyclic structures are designated with matching numbers in ring openings and ring closures (e.g. cyclohexane, *C1CCCCC1*).

Stereochemistry. “@” (anti-clockwise neighbors) and “@@” (clockwise neighbors) symbols are used to indicate the chirality of tetrahedral centers. “\” and “/” symbols are utilized as directional bonds which are placed around double bonds (e.g. trans and cis-difluoroethene, *“F\C=C\F”* and *“F/C=C\F”*, respectively). **Canonicalization.** A molecule can be represented with more than one SMILES because of the different ordering of strings. Though, the arrangement of the string does not affect the structure of the molecule, referring to a molecule with several SMILES might lead to ambiguities in some cases. Canonical SMILES can provide a unique SMILES representation, however, different databases such as PubChem and ChEMBL might use different canonicalization algorithms to generate different unique SMILES. OpenSMILES is a new platform that aims to universalize the SMILES notation [50]. In Isomeric SMILES, isotopism and stereochemistry information of a molecule is encoded using a variety of symbols ("/", "\", "@", "@@").

SMARTS. SMiles ARbitrary Target Specification (SMARTS) is a language introduced by Daylight Chemical Informations Inc. which enables substructure (pattern) search on SMILES string [51].

DeepSMILES is a novel SMILES-like notation that is proposed to address two challenges in SMILES syntax: (i) unbalanced parentheses and (ii) ring closure pairs [52]. DeepSMILES syntax uses a single close parantheses to instead of using open and close parantheses to describe the branch length. For instance, SMILES "C(OF)C" is represented as "COF))C" in DeepSMILES. As for the ring closure numbers, a single symbols denotes the size of the ring unlike the use of two pair ring numbers in SMILES. For instance, SMILES "C1CC(OC)CC1" is expressed as "CCCOC))CC5" in DeepSMILES. DeepSMILES-syntax aims to enhance the effectiveness of the machine/deep-learning based approaches that utilize SMILES data as an input to their systems [53].

2.1.2. Fingerprints

Fingerprints (FPs) are widely adopted representation techniques for chemicals. They are either obtained from SMILES string using pre-defined SMARTS rules or through hash-based approaches. In this section, we will cover three of the most popular fingerprint techniques that will be mentioned throughout the thesis.

PubChem fingerprint (PubChemFP) represents the existence of 881 different features in the form of binary feature vectors (e.g. 1 represents the existence of a certain feature wheres 0 indicates absence) [54]. Pubchem 2D substructure based similarity tool is available online [55].

MACCS is a structural fingerprint where each bit represents a specific substructure [56]. MACCS fingerprints are formed by the binary responses to a set of structural questions such as "Is there a ring size of 4?"". SMARTS representation of the MACCS features can be found at [57].

Extended-Connectivity Fingerprint (ECFP) is a hash-based representation techniques that considers the atoms and their circular neighbors within a radius range to describes the features of substructures [58]. ECFP4 and ECFP6 are two of the most popular molecular fingerprints.

2.1.3. Databases

PubChem [54] stores information for around 96M compounds and 265M substances [59]. PubChem also acts as a cheminformatics tool by providing an interface that enables the computation of 2D/3D similarity of compounds. ChEMBL [32] is another widely accessed database that stores manually curated information about chemical properties and protein targets and bioactivities for 1.9M compounds [60]. DrugBank [8] comprises chemical, pharmacological and pharmaceutical information for 13K drugs and 5K proteins (e.g. drug targets/enzymes) that are associated with these drugs [61]. PDB, BindingDB [39], ChEBI [62], ZINC [63], PDB-Bind [64], KEGG [65] and ChemSpider [66] are also among the important chemical databases/sources that constitute valuable input for drug discovery studies.

2.2. Proteins

Proteins are macromolecules that play key roles in many tasks including catalyzing chemical reactions, transportation of nutrients and forming cellular structures (e.g. tissues, organs) [67, 68]. Proteins can be represented in 1D (protein sequence), 2D and 3D forms. Three-dimensional structure of the protein, which is shaped by the chains folding in the water, is an important determinant of the protein function [67]. Furthermore, the relationship between the structure and the sequence of the protein is verified in past researches [67]. The remaining challenge is, however, that the structure is not available for every protein but sequence is as it can be easily observed from the public databases. This constitutes our main motivation of focusing on the textual data for the proteins.

2.2.1. Protein Sequence

Proteins comprise amino acids which are small organic compounds, connected to each other by forming long chains. There are 20 different amino acids that are represented by unique characters of the English alphabet (Table 2.3). Amino acids are encoded by "codons" which are 3-character nucleotides. Protein size/length is expressed in terms of number of amino acids which can vary between 30 to 30000 [67].

Proteins are composed of 20 different amino acids that can be represented by unique characters of the English alphabet. Protein sequence length usually vary between 30 to 30000 [67]. The shortest sequence in UniProt belongs to *neuropeptide GWa* (P83570) with 2 amino-acids (i.e. GW) while the longest sequence is *Titin* (A2ASS6) with 35,213 aminoacids [69]. The sequence (i.e. text-based form) of the protein can be referred to as 1D representation. The 2D representation refers to inter-residue distances [67] and 3D representation stores the coordinates of the protein structure. Proteins can be subjected to posttranslational modification after their synthesis because of a chemical change [70]. Such modicafications increase the proteome diversity and can be linked to major protein related events such as function and ligand-binding.

Table 2.3: List of 20 amino-acids.

no	**definition**	**abbreviation**	**symbol**
1	Arginine	Arg	R
2	Histidine	His	H
3	Lysine	Lys	K
4	Aspartic acid	Asp	D
5	Glutamic acid	Glu	E
6	Asparagine	Asn	N
7	Cysteine	Cys	C
8	Glutamine	Gln	Q
9	Glycine	Gly	G
10	Serine	Ser	S
11	Threonine	Thr	T
12	Tyrosine	Tyr	Y
13	Alanine	Ala	A
14	Isoleucine	Ile	I
15	Leucine	Leu	L
16	Methionine	Met	M
17	Phenylalanine	Phe	F
18	Proline	Pro	P
19	T1ryptophan	Trp	W
20	Valine	Val	V

2.2.2. Protein Representation/Similarity

Smith-Waterman (S-W). Smith-Waterman [12] is a local alignment algorithm which aims to capture the similarity between two sequences (e.g. proteins, DNA). Instead of comparing whole sequences, S-W algorithm matches local patterns in sequences.

Basic Local Alignment Search Tool. Basic Local Alignment Search Tool (BLAST) [71] is a similarity computation tool that is based local similarities. It performs sequence similarity for a given sequence against a database of sequences and reports significant similarity matches. For instance, BLAST can be used to detect the family of a protein.

2.2.3. Databases

The Universal Protein Resource (UniProt) [69] is one of the main public databases for proteins that stores sequence and function information for over 158M proteins (including the automatically annotated proteins, 550K of which are reviewed) [72]. The Protein Data Bank (PDB) [68, 73] is the other main source for proteins which comprises available protein crystal structures and structural information for around 152K macromolecular structures [74]. Aside from these, databases such as STRING that contains protein-protein interactions [75], Pfam which comprises protein family information [76], CATH [77] and PROSITE [78] that store protein domain information are among the widely accessed resources for proteins.

3. THEORETICAL BACKGROUND

3.1. Related Methodologies

3.1.1. Identification of chemical words/tokens

Similar to words in natural languages, we can assume that the "words" of biochemical sequences are able to convey significant information (e.g. folding, function etc) about the entities. In this regard, each compound/protein is analogous to a sentence, and each compound/protein unit is analogous to a word. Therefore, if we can decipher the grammar of biochemical languages, it would be easier to model bio/cheminformatics problems. However, protein and chemical words are not explicitly known and different approaches are needed to extract syntactically and semantically meaningful biochemical word units from these textual information sources (i.e. sequences). Here, we investigate the tokenization approaches that are used in this thesis to determine the words of the chemicals and proteins.

3.1.1.1. *k*-mers. One of the simplest approaches in NLP to extract a small language unit is to use n-grams. N-grams indicate n consecutive overlapping characters that are extracted from the sequence using a sliding window approach. For example, the 3-grams of the word "happiness" can be listed as "hap", "app", "ppi", . . . , "nes", "ess" }. From a sequence of length L, total $(L - n) + 1$ n-grams can be extracted. N-grams, often in bioinformatics domain, are also referred to as k-mers indicating the same approach of identifying words.

k-mers (*n*-grams) are frequently used in bio/cheminformatics domain to represent the "words" of the biological and chemical languages. Vidal and co-workers' study on processing SMILES strings to extract fragment-like units that are essentially overlapping *k*-mers (i.e. 4-mers, or 4 consecutive SMILES characters, or LINGO) utilize these "words" to compute inter-molecular similarities [79]. The LINGOs that can be identified from SMILES string of *ampicillin* “CC1(C(N2C(S1)C(C2=O)NC(=O)C(C3=CC =CC=C3)N)C(=O)O)C” are { “CC0(”, “C0(C”, “0(C(”, . . . , “)O)C” } (All ring numbers in SMILES are replaced with 0s before extracting LINGOs). LINGO profiles were as good at differentiating between bioisosteric and random molecular pairs, without requiring 2D or 3D information. LINGOs were successfully employed in drug-target interaction prediction task by our team [33]. The results suggested that SMILES-based approach to compute similarity of chemicals is not only as good as a 2D-based similarity measurement, but also efficiently faster.

3.1.1.2. Fragments. Fragments represent the small molecular structures such as functional groups (e.g. carbonyl group) that are created by the decomposition of the SMILES string. There has been an ongoing interest in the use of fragmentation in drug discovery studies for the last decades. MolBlocks [80] is a tool to partition SMILES notated compounds into fragments. MolBlocks depends on SMARTS, a rule-centric language, to find the substructures and property patterns (e.g. for carbonyl group, the following SMARTS is used ([CX3]=[OX1]). There are three most widely used rule sets for fragmentation, namely [81], BRICS [82] and CCQ [83].

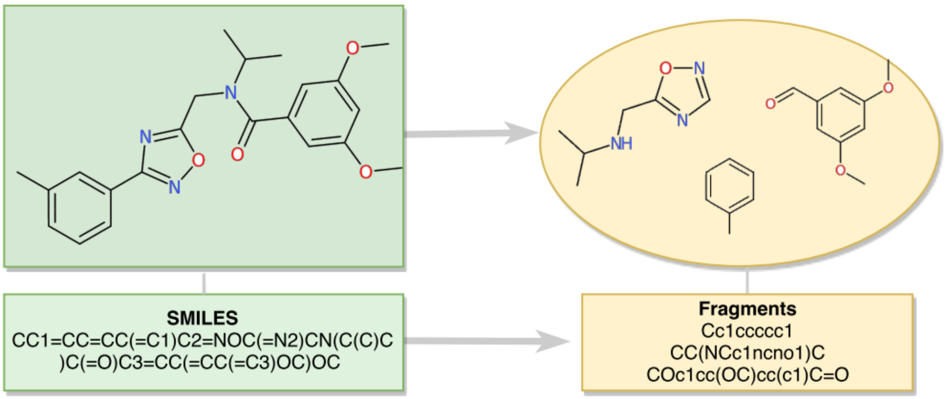


Figure 3.1: Extraction of substructures from 2D molecule graph (top) and the extraction of chemical words as fragments using BRICS (bottom).

3.1.1.3. Byte-Pair Encoding (BPE). Byte-Pair Encoding (BPE) is a compression technique [84] that inspired Senrich and co-workers to adopt it to the word segmentation task [85]. BPE generates words based on high frequency sub-sequences starting from frequent characters. The system is initialized with a character vocabulary, which is extracted from a large corpus that the model is trained on. The training process includes iteratively creating new symbols by merging the most frequent ones together. For instance, frequent symbols “o” and “n” together creates the word “on”. Then, combination of “on” and “e” leads to the word “one”.

A recent study adopted a linguistic-inspired approach to predict protein-protein interactions (PPIs) [86]. To determine the “words” (i.e. bio-words) of the protein language, they utilized a uni-gram model based on a data-driven word segmentation algorithm and used BPE and expectation maximization [87] algorithms to build the bio-word vocabulary. Wang and co-workers [86] suggested that segmented words indicate a language-like behavior for the protein sequences.

3.1.1.4. Maximum Common Substructures. Cadeddu and co-workers [88] investigated organic chemistry as a language in an interesting study that extracts maximum common substructures (MCS) from the 2D structures of pairs of compounds to build a vocabulary of the molecule corpus. Contrary to the common idea of functional groups (e.g. methyl, ethyl etc.) being “words” of the chemical language, the authors argued that MCSs (i.e. fragments) can be described as the words of the chemical language [88]. A recent work investigated the distribution of these words in different molecule subsets [89]. The “words” followed *Zipf”s Law*, which indicates the relationship between the frequency of a word and its rank (based on the frequency) [90], similar to most natural languages. Their results also showed that drug “words” are shorter compared to natural product “words”.

3.1.2. Word/Sentence Embeddings

3.1.2.1. Vector Space Model. The Vector Space Model is used in information retrieval to estimate the relevance of each document in a corpus to a user query [91]. A document is represented by a vector of either weighted or un-weighted terms (usually words). The document vector represents the document in the form of a *bag-of-words* [92]. For instance, the SMILES of ampicillin "CC1(C(N2C(S1)C(C2=O)NC(=O)C(C3=CC=CC=C3)N)C(=O)O)C" can be represented as a bag-of 8-mers as follows: {"CC1(C(N2", "C1(C(N2C", "1(C(N2C(", "(C(N2C(S",...,"N)C(=O)O" ,")C(=O)O)" ,"C(=O)O)C" }. The *bag* {"1(C(N2C(", "O)NC(=O)", "=CC=C3)N", "C3=CC=CC", ...,"=O)C(C3=", "N2C(S1)C", ")C(=O)O)"} is equal to the previous *bag*. We can vectorize this SMILES as $S = [1, 1, 1, 1, ..., 1, 1, 1]$ in which each number refers to the frequency of the 8-mers, *"1(C(N2C(", "O)NC(=O)", "=CC=C3)N", "C3=CC=CC", ..., "=O)C(C3=", "N2C(S1)C", ")C(=O)O)"*, respectively.

3.1.2.2. Term Frequency-Inverse Document Frequency (TF-IDF). Approaches, such as vector-space models, that are based on counting the terms of the sentence and/or document might prioritize insignificant but frequent words. For instance, compared to the $8-mer$ "C3=CC=CC", the existence of "(C(N2C(S" in a SMILES string might give more information about the compound. To overcome this issue, a weighting scheme can be integrated into the vector representation in order to give more importance to the rare terms that might play a key role in detecting similarity between two documents.

One of the most popular weighting approach is to use term frequency-inverse document frequency (TF-IDF). TF can be computed as described in Equation 3.1 where $tf_{t,d}$ refers to the number of occurrences of term t in document d .

$$tf_{t,d} = \begin{cases} 1 + log_{10}(tf_{t,d}), & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

Then TF-IDF weighting is computed as follows [93]:

$$tf\text{-}idf_{t,d} = tf_{t,d} * idf_t \tag{3.2}$$

in which idf_t indicates the inverse document frequency of term t. idf_t can be computed as $idf_t = log(D/df_t)$ in which D is the number of documents in the corpus, and df_t

indicates the frequency of term t appearing in document d. *idf* of a rare term is higher whereas informative words such as "the" or "a" have lower *idf* values. For instance, the IDF of "C3=CC=CC" is lower than that of "(C(N2C(S" because the former appears in more compounds than the latter.

3.1.2.3. Distributional Word Embeddings. The distributional word embeddings models have gained popularity with the introduction of Word2Vec that is proposed by Mikolov and co-workers [94]. The main motivation behind the Word2Vec model is to build real-valued high-dimensional vectors for each word in the vocabulary based on its neighboring words. The power of distributed word embeddings comes from this feature, which encodes semantic relatedness of the words. Thus, two words that appear in the same context have similar vector representations.

Word2Vec relies on a simple ANN structure with a single hidden layer in which number of the nodes in the hidden layer decides the size of the embedding vector. The model trains on a large corpus, such as Wikipedia [95] for English, to learn efficient embeddings. Figure 3.2 shows an example of how words that are represented through the embeddings learned via Word2Vec look when mapped into 2D-space. The relationships between Thor and Mjolnir, and between Merlin and Excalibur are similar to each other in which two memorable weapons are often remembered with two famous heroes of fiction.

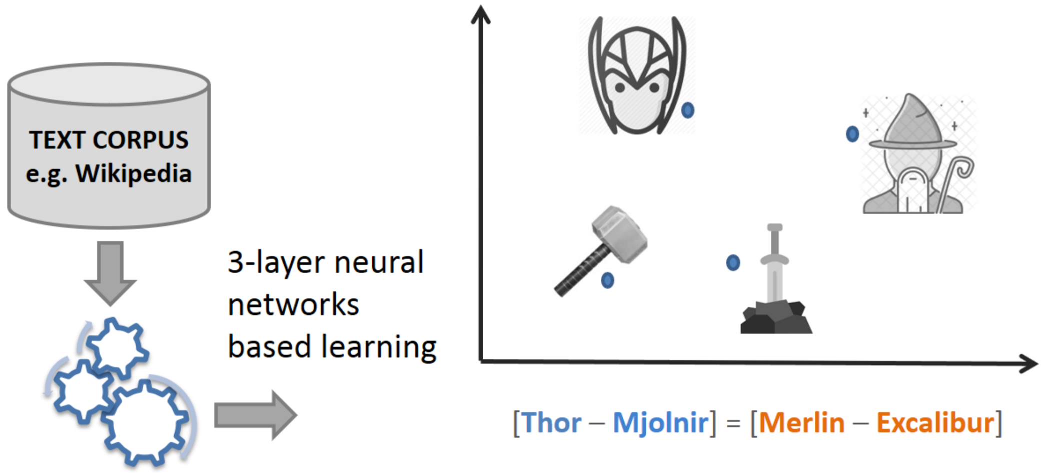

TEXT CORPUS
e.g. Wikipedia
3-layer neural networks based learning
[Thor – Mjolnir] = [Merlin – Excalibur]

Figure 3.2: Each dot (image) represents the position of the word in 2D space.

There are two main approaches in Word2Vec: (i) Skip-Gram and (ii) Continuous Bag of Words (CBOW). In the Skip-Gram model the aim of the model is to predict context word given the center word, whereas in CBOW the objective of the model is to predict the target word given the context words. The weight matrix between the input layer and the hidden layer stores the embeddings of the vocabulary words.

The words are represented as one-hot encoded vectors in the input layer of the Word2Vec algorithm. One-hot encoding means that for a vocabulary size of V, each word w_i is assigned to 1 in the corresponding position, and the remaining words are represented as 0. For instance, 43 unique 8-mers can be extracted from the SMILES of ampicilin: {“CC1(C(N2”, “C1(C(N2C”, “1(C(N2C(”, “(C(N2C(S”,...,“N)C(=O)O” ,“)C(=O)O)” ,“C(=O)O)C” }. Thus, the size of the one-hot vector for each word becomes 43 in which only the positions that represent the corresponding 8-mer is set to 1.

$$\begin{bmatrix} \text{CC1(C(N2} \\ \text{C1(C(N2C} \\ \text{1(C(N2C(} \\ \cdots \\ \text{)C(=O)O)} \\ \text{C(=O)O)C} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3.3 illustrates the Skip-Gram network architecture in which for a given word c number of context words are predicted.

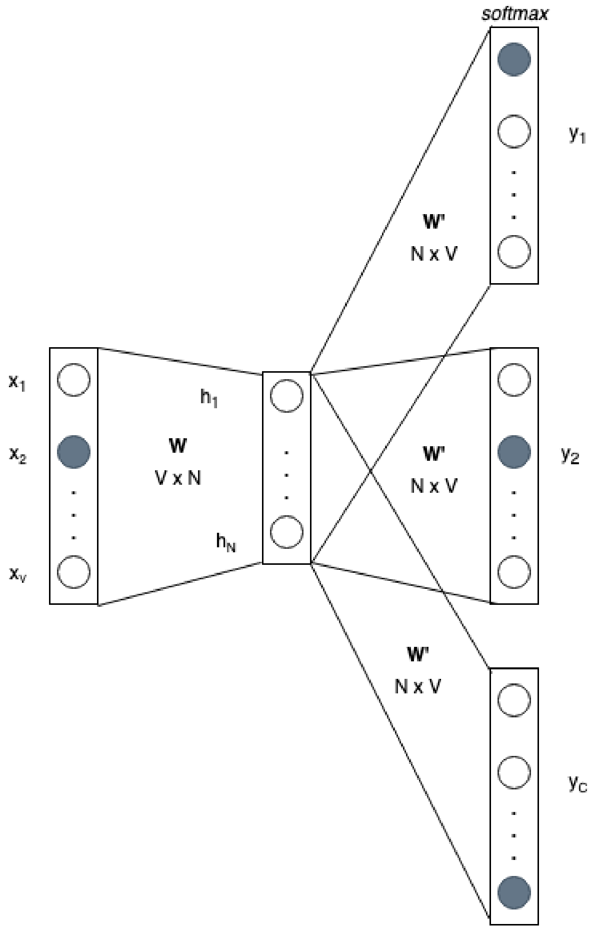


Figure 3.3: For a target word its neighbor words are predicted.

The Skip-Gram architecture can be explained as described in Equations 3.3, 3.4 and 3.5:

$$h = W^T x \tag{3.3}$$

$$u_c = {W'}^T h = {W'}^T W^T x \text{ c=1,2,...,C} \tag{3.4}$$

$$y_c = Softmax(u) = Softmax({W'}^T W^T x) \text{ c=1,2,...,C} \tag{3.5}$$

where x is the one-hot encoded input vector, V is the size of the vocabulary, N represents the number of the nodes in the hidden layer (i.e. the size of the embedding vector, W is the weight matrix that stores the embeddings and y is the output vector. ($x \in \mathbb{R}^V, W \in \mathbb{R}^{VxN}, W' \in \mathbb{R}^{NxV}$)

3.1.3. Deep Learning

3.1.3.1. Artificial Neural Networks. An artificial neural network (ANN), which is inspired by the nervous system in the brain, is a model formed by interconnected layers in a non-linear way [96,97]. Figure 3.4 depicts a simple ANN with a single hidden layer between the input and output layer. The increase in the computational power of the machines has enabled the design of neural networks with large number of hidden layers and neurons, thus giving rise to deep neural networks (DNN).

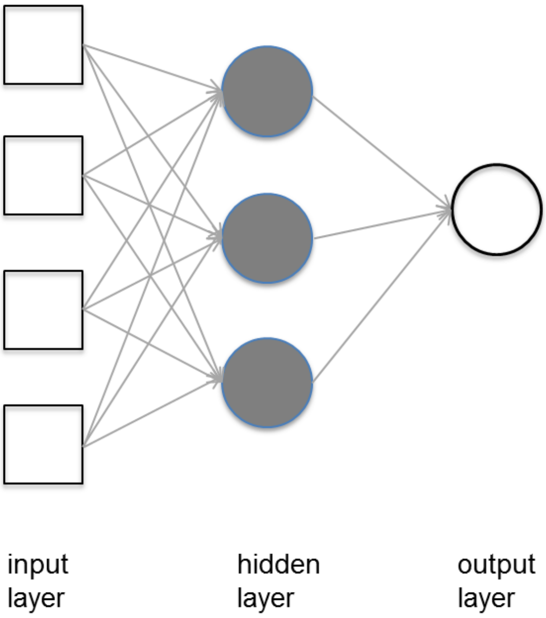


Figure 3.4: An ANN that contains single hidden layer.

The DNN architecture can be described as in Equation 3.6 where x represent the input data, z^l is the input of the l_{th} layer, W^l is the weight matrix and b denotes the bias term [26].

$$z^{l+1} = W^l a^l + b^l \tag{3.6}$$

$a = f(x)$, where $f(x)$ indicates the activation function. Following, we will shortly describe the main parameters of general deep neural networks.

Activation function There are many activation functions widely used in different studies such as sigmoid and tanh. However, recent studies showed that Rectified Linear Unit (ReLU) [98], $f(x) = max(0, x)$, is a better choice for deep learning studies [99]. DNN tries to minimize the difference between the expected (real) value and the prediction during training.

Loss function evaluates how well the candidate solution performs. In regression problems Mean Squared Error (MSE), in binary classification problems cross-entropy are popular choices for loss functions.

Batch-size indicates the number of patterns you keep in memory before updating the weights in order to reach the desired output.

Epoch describes the number of times the model sees the whole dataset.

Learning rate determines how much the weights should be updated after the end of each batch. It is one of the important hyper-parameters of DNN architecture design since it has an important effect on the speed and the performance of the model.

Dropout is a regularization technique that is used to avoid the over-fitting problem [100]. With dropout, some of the neurons are "dropped-out" meaning their activation is set to 0.

3.1.3.2. Convolutional Neural Networks. Convolutional Neural Network (CNN) is a special type of an ANN. CNN architecture comprises one or more convolutional layers usually followed by a pooling layer. A pooling layer down-samples the output of the previous layer and provides a way of generalization of the features that are learned by the filters. On top of the convolutional and pooling layers, the model is completed with one or more fully connected layers of a feed-forward neural network (FFNN). Figure 3.5 illustrates an image classification system designed with CNN with a FFNN on top.

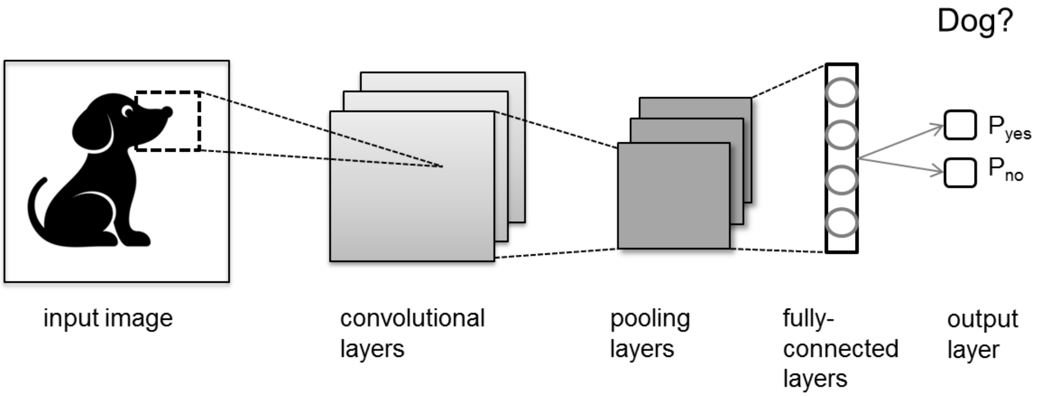


Figure 3.5: An example of an image classification task.

The ability to capture the local dependencies is the most powerful feature of CNN models which is achieved with the help of filters. Point-wise multiplication of the kernel matrix with the values of the input constitutes the new output feature. The kernel moves through the input to create new features with the given stride size.

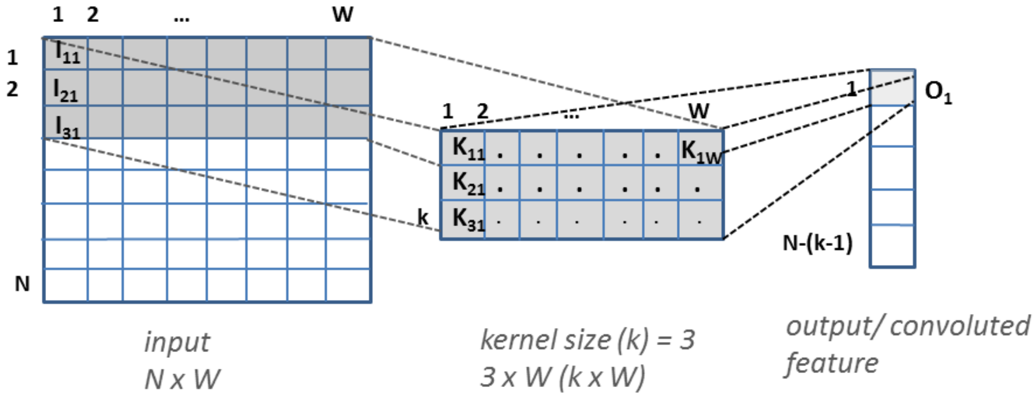


Figure 3.6: Pairwise multplication of kernel and input matrix creates the output feature.

Figure 3.6 depicts the construction of an output vector from an input matrix of size NxW in 1D convolution. The size of filter is 3, which corresponds to the number of the rows. In 1D convolutional architecture the number of the features of the kernel is always equal to the ones in the input matrix (i.e. W). With stride equal to 1, the first feature of the output vector is computed as, $O_1 = (I_{11} * K_{11}) + (I_{12} * K_{12}) + ... +$

$(I_{3W} * K_{3W})$. The number and size of the filters in a CNN directly affects the type of features the model learns from the input. The increase in the number of filters is positively correlated with the increase in the performance of the model at recognizing patterns [101].

3.1.4. eXtreme Gradient Boosting

eXtreme Gradient Boosting (XGBoost) has became a popular choice for many researchers even surpassing popular deep neural networks as the statistics from Kaggle challenges indicate [102]. XGBoost has also recently been employed in different bioinformatics tasks such as QSAR studies [103] and prediction of physical chemistry properties [104].

Gradient boosting tree (or gradient boosting machine) is one of the most popular algorithms in machine learning that is an ensemble of sequential trees in which a given tree *t* aims to learn from the misclassified samples of the previous *t-1* trees by assigning them higher weights [105]. eXtreme Gradient Boosting (XGBoost), proposed by Chen and co-workers [102], is built on gradient boosting tree algorithm and is a regularized and scalable version of the original algorithm to avoid over-fitting. A tree ensemble can be formulated as follows (3.7) [106]:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{3.7}$$

where K is the number of trees, f is a function in the functional space F, and F is the set of all possible classification and regression trees (CARTs). And the objective function can be expressed as in Equation 3.8:

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.8}$$

where l is the loss function that measures the difference between the actual value y_i and the predicted value $\hat{y}_i$, while Ω is the tuning parameter that controls the complexity of the model. The details are described in [102, 107].

3.1.5. Transitive Clustering

Transitivity Clustering (TransClust) is a clustering method that is based on the weighted transitive graph projection problem [108]. TransClust uses a weighted cost function to construct transitive graphs by adding or removing edges from an intransitive graph. The weighted cost function is expressed as the distance between a pre-defined (e.g. user-defined) threshold and a pairwise similarity function.

In a protein clustering task, TransClust can be used to connect proteins on the network if their similarity is greater than the user-defined threshold. The graph is expanded by adding or removing edges until it becomes a disjoint union of clusters.

3.1.6. Markov Clustering Algorithm

Markov Clustering (MCL) is a network clustering algorithm that works on flows of the network [109], which means that the edge weights of the network are considered for the clustering process. The algorithm is implemented for given number of iterations. For each iteration, first, the matrix is expanded by algebraic matrix multiplication to itself. Then, each non-zero elements of the new matrix are raised to a power which is an input of the algorithm called granularity inflation. Increasing the inflation value causes the emerging of new clusters of the network.

3.1.7. Kronecker Regularized Least Squares

Kronecker Regularized Least Squares (KronRLS) approach aims to minimize the following f function [24] (Equation 3.9):

$$J(f) = \sum_{i=1}^{m} (y_i - f(x_i))^2 + \lambda ||f||_k^2 \tag{3.9}$$

where x_i is the training inputs and y_i is the real value. λ acts as a regularization parameter that negotiates between the model complexity and and the prediction error, whereas $||f||_k^2$ is the norm of f function associated with the kernel k [24].

4. SMILESVec: DISTRIBUTED REPRESENTATION OF LIGANDS

4.1. Introduction

The description of a ligand is a significant task for many bio/cheminformatics problems. Ligands can be described in various different forms including knowledge-based fingerprints, graphs, or strings. SMILES, which is a character-based representation of ligands (more details in Section 2.1.1), has been used for QSAR studies [110,111] and protein-ligand interaction prediction [33,112]. Even though it is a string based representation form, use of SMILES yielded close performance to powerful graph-based representation methods in protein-ligand interaction prediction [33]. Our team also showed that SMILES-based representation is computationally cheaper in terms of running time [33]. Besides, as a character-based form, SMILES provides a promising environment for the adoption of Natural Language Processing (NLP) methodologies.

Distributed word representation models have been widely adopted in recent studies of NLP problems, in particular with the introduction of the Word2Vec algorithm [94]. The neural-network based model requires a large amount of text data to learn the representations of words. Then, the model describes the words in low-dimensional space as real valued vectors. These vectors comprise the syntactic and semantic features of the words, since they are created by considering the neighbor words, e.g., the vectors of words with close meanings are also similar.

Here, we introduce an approach called SMILESVec to represent ligands using their SMILES strings. SMILESVec is built upon the distributed word embeddings, in which a large SMILES corpus was used to train Word2Vec model to learn features for the “chemical words”. And thus, instead of using manually constructed ligand features, the words of the SMILES language are defined by a data-driven approach. This chapter is based on the work published as [36] and introduces the SMILESVec approach that is utilized in [36].

4.2. Related Work

Learning an efficient representation for molecules has been attempted by many studies in the recent years. The SMILES2Vec approach proposed a recurrent neural network (RNN) based architecture for training on SMILES to predict chemical properties [113] whereas CheMixNet employed CNN and RNN based architectures to learn both from SMILES and MACCS fingerprints for the same task. Mol2Vec, which adopts the Word2Vec algorithm [94], on the other hand, extracted substructures from SMILES using Morgan algorithm fingerprints, and then learned an embedding for each substructure fingerprint to predict chemical properties [114].

Our method is different than these studies on this problem because of its NLP-inspired nature. Bridging an analogy between NLP and chemistry, the SMILES string is treated as a sentence and thus, "chemical words" are extracted from it. The words of the sentence convey a sentiment or a topic in which some of the words are more descriptive while the others are not. Thus, we can also use the analogy of sentence similarity to design the similarity between the chemicals.

4.3. Methods

4.3.1. Distributed Representations of Chemical Words

Word embeddings are ways of representing words as low-dimensional real-valued vectors using their context to integrate syntactical and semantic information. Word embeddings were introduced over a decade ago [115] and followed by several studies [116], and finally became popular with the success of Mikolov's Word2Vec [94].

In the Word2Vec model, a word is embedded into a n-dimensional space dependent of the context that the word occurs together after training on a large corpus. The vector learning is based on the context of each word (e.g. its surrounding word) and can detect some important words that most often occur in the same contexts. Thus, two words have similar vectors if they often appear together in similar contexts, such as “Thor” and “Mjolnir”.

In this work, we adopt this methodology into the chemical/biological domain in order to represent compounds. If the words of the chemical language (SMILES) is determined, then these words and their relationships with each other can give away important information about the chemical such as its functionality. Therefore every word of a ligand SMILES was described in a semantically meaningful way with the help of the neural-network based nature of Word2Vec.

SMILESVec is a NLP-inspired approach that combines: (i) identification of chemical words, and (ii) representation of these words in the distributed space. The Word2Vec model has been adopted to represent proteins using their sequences in an earlier study [117]. The approach, that we will refer to as ProtVec throughout the article, enhanced the performance for the protein classification problem.

Asgari and co-workers defined 3-mers as the words of the protein sequences in ProtVec [117]. For SMILESVec, on the other hand, we performed several experiments in which word size (k) varied in the range of 4-12 characters and 8-mers as the chemical words yielded more meaningful results (The results are reported in Appendix A). Figure 4.1 depicts the Zipf’s plot in which the distribution of 4-, 6-, 8-, and 10-mers are compared to the distribution of the words in English, specifically all works of Sir Arthur Conan Doyle.

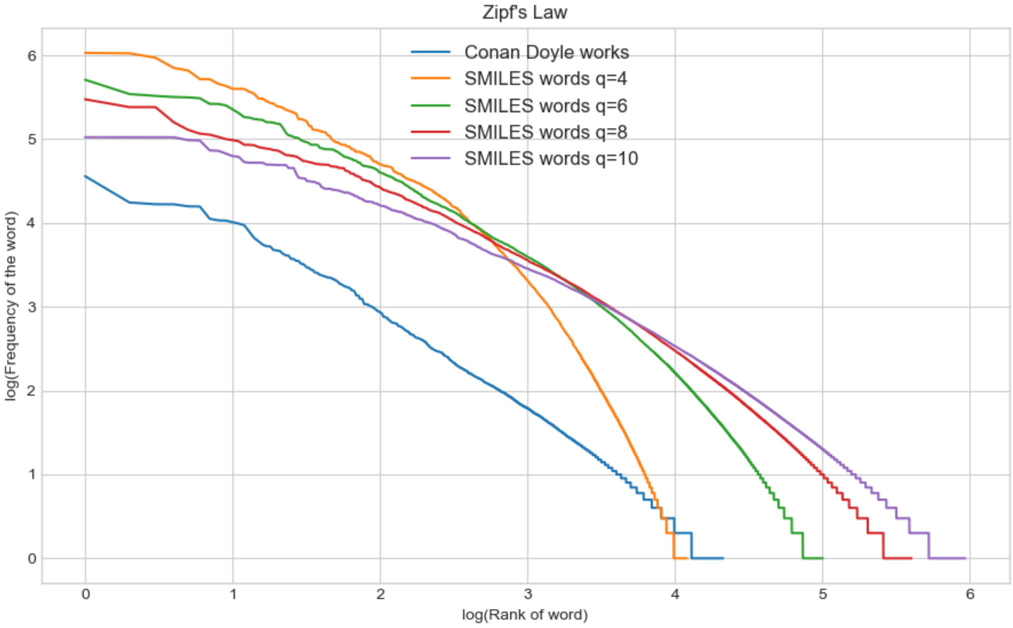


Figure 4.1: Zipf's Law distribution of different k-mers (k=4,6,8,10).

Zipf's Law indicates the relationship between the frequency of a word and its rank (based on the frequency) [90] and it is often used to understand the distribution of the words. Natural languages usually hold Zipf's Law in which the frequency of a word is inversely proportional to its rank. 8-mers in a randomly sampled 500K SMILES corpus (chemical language) follows the Zipf's law much similar to the distribution of the words in the works of Sir Arthur Conan Doyle (English).

Figure 4.2 demonstrates an example protein sequence and its sequence list (biological words) as well as an example ligand SMILES and its corresponding sub-sequences (chemical words). The protein is represented as sequence-lists which comprise biological words, a set of three characters (3-mers) of non-overlapping sub-sequences. Total three sequence lists are generated where each list stores the biological words that starts from the character indices 1,2, and 3, respectively [117]. The chemical words, on the other hand, are created as 8-character long overlapping substrings (8-mers) of SMILES with sliding window approach. The SMILES string “C(C1CCCCC1)N2CCCC2” is divided into the following chemical words: “C(C1CCCC”, “(C1CCCCC”, “C1CCCCC1”, “1CCCCC1), “CCCCC1)N”, “CCCC1)N2”, ... , “)N2CCCC2” in Figure 4.2.

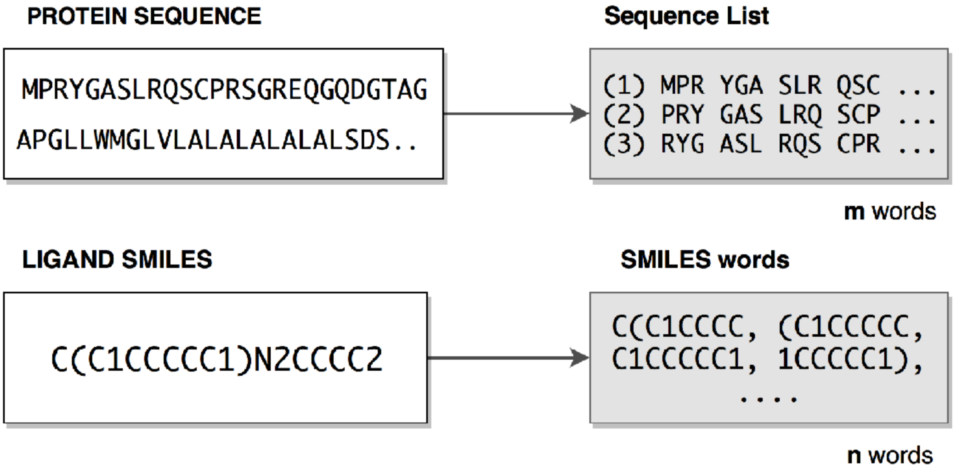


Figure 4.2: Chemical and biological words extracted from compounds and proteins.

4.3.2. SMILES Corpora

In order to train the Word2Vec algorithm for representing chemicals, we built two different SMILES corpora: (i) PubChem canonical [54] and (ii) ChEMBL canonical [6]. Even though both SMILES databases are referred to as canonical, the encoded information changes because of the use of different canonicalization algorithms. ChEMBL considers isomeric information in canonical SMILES whereas PubChem does not.

While building the PubChem SMILES corpus, we followed two rules in selecting compounds:

- Molecular weight should be less than 1000, (MW< 1000). (This eliminates the large molecules such as peptides.)
- PubChem bioassay status should be Tested. (PubChem assigns bioassay status to the compounds, representing whether it is experimented with a target or not. A tested compound could either be active or inactive.)

These criteria resulted in SMILES information for approximately 2M compounds.

ChEMBL SMILES corpus, on the other hand, is created with 1.7M canonical SMILES from CHEMBL23 database [118]. No further filter is applied. Before training, the elements represented with more than one character such as “Cl” and “Na” are converted into a single character. As such, these elements are considered as a single symbol while creating 8-mers.

4.3.3. SMILESVec

Once the words of the SMILES are determined, the next step is to learn distributed vector representations for these words. We used the Word2Vec model with the Skip-gram approach to consider the order of the surrounding words. With the use of the Word2Vec model, we were able to describe complex structures using their simplified representations. For each sub-sequence (word) that was extracted from protein ligand SMILES, Word2Vec produced a real-valued vector that is learned from a large training set. As for training sets, we either used ChEMBL23 or PubChem corpora.

Figure 4.3 illustrates SMILESVec pipeline such that, first, Word2Vec is trained on a large SMILES corpus to learn embeddings for the words (i.e. 8-mers) that are extracted from these SMILES strings. Then, the learned word embedding vectors are taken average of to construct the SMILES (i.e. compound) vector. SMILESVec can be described in Equation 4.1:

$$SMILESVec = vector(ligand) = \frac{\sum_{k=1}^{n} vector(word_k)}{n} \tag{4.1}$$

in which $vector(word_k)$ represents the Word2Vec output vector for the k_{th} word (i.e. 8-mer) of the SMILES string and n indicates the total number of these chemical words. We will refer to ligand vectors as SMILESVec throughout the article.

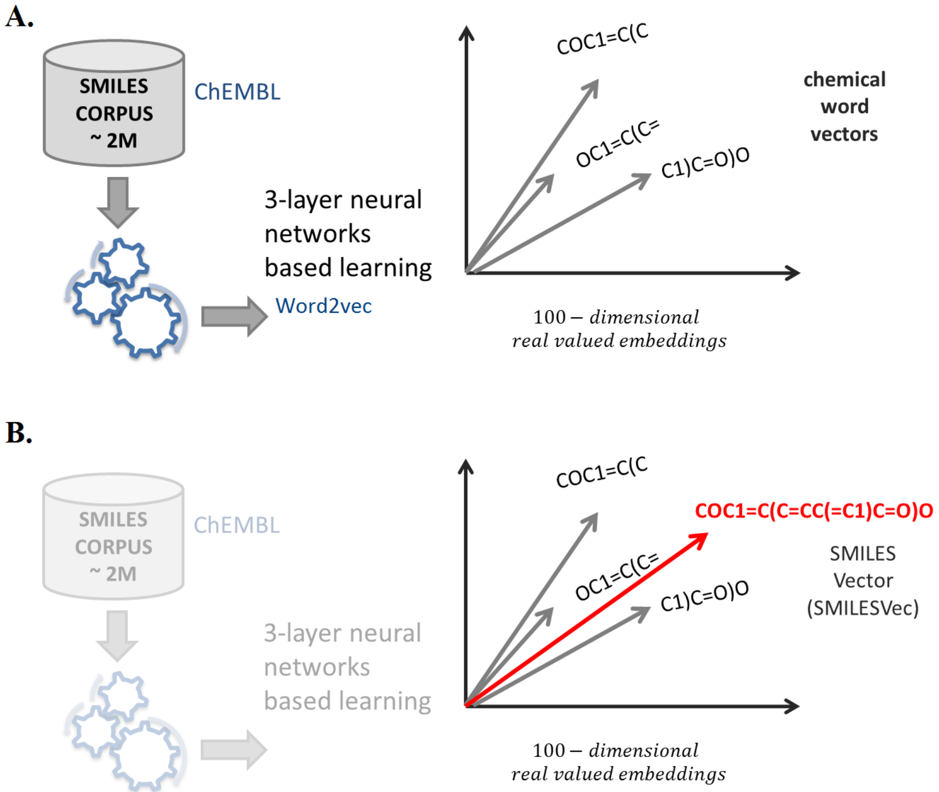


Figure 4.3: (A) Learning embeddings for chemical words from a large corpus. (B) Combining word embeddings into a SMILES embedding.

We used the Gensim implementation [119] of Word2Vec and the size of the vectors was set to the default value of 100. We also experimented with CBOW implementation of Word2Vec, since it does not consider the order of the words. Our preliminary experiments for CBOW and Skip-Gram implementations yielded similar results.

4.4. Conclusion

We have presented a novel approach, SMILESVec, to represent ligands using their textual forms (i.e. SMILES). SMILESVec adopts the word-embeddings approach to define ligands by utilizing the chemical words that are extracted from their SMILES strings. Ligands are represented by learning features from a large SMILES corpus via Word2Vec [94], instead of using manually constructed ligand features.

SMILESVec can be used as alternative representation technique in different tasks involving drug discovery such as prediction of molecular properties or binding affinities. We will introduce a novel use of SMILESVec in the next chapter (Chapter 5) and compare the performance of SMILESVec to popular fingerprint-based representations. In Chapter 6, we will also investigate the use of SMILESVec in the drug-target binding affinity prediction task. The design of chemical words is important in constructing the SMILESVec. The work we introduced in this chapter provides examples of the words that are determined as k-mers. Different word extraction techniques both from NLP and chemical domains will be evaluated in Chapter 6 as well.

5. SMILESVec-BASED PROTEIN REPRESENTATION

5.1. Introduction

Reliable representation of proteins plays a significant role in the performance of myriad of bioinformatics tasks such as protein family classification and clustering, prediction of protein functions and the interactions between protein-protein and protein-ligand pairs. Proteins are usually represented based on their sequences [120–122]. However, even though the structure of a protein is determined by its sequence, sequence alone is usually not sufficient to thoroughly interpret its mechanism. Moreover, the relationship between fold or architecture and function was shown to be weak, while a strong correlation was reported for architecture and bound ligand [123].

As a consequence, a novel approach that describes proteins by integrating functional characterizations can provide major information toward understanding and predicting protein structure, function and mechanism. Ligand-centric approaches are based on the chemical similarity of compounds that interact with similar proteins [124]. The pioneering works that proposed to measure protein similarity using their ligands [34, 125] inspired the following works that successfully adopted this approach for tasks such as target fishing, off-target effect prediction and protein-clustering [126,127]. The use of chemical similarity of the interacting ligands of proteins to group them resulted in both biologically and functionally related protein clusters [34, 35, 128].

Motivated by these studies and their results, we propose an alternative approach to describe proteins using their interacting ligands instead of using their sequences. With this, we aim to integrate the functional information of the proteins that make them the components of the interactions between a specific set of ligands.

In order to define a protein with a ligand-centric approach, the definition of the ligands is vital. In this work, we utilize the ligand representation methodology SMILESVec, that we introduced in Chapter 4. We first build SMILESVec for each ligand the proteins binds to. We then describe the protein using the average of its interacting ligand vectors (ie. SMILESVec). The proposed ligand-centric protein representation is evaluated in protein family/superfamily clustering task. We practised an identical pipeline for evaluation that is presented in the work of Bernardes and co-workers [129]. The authors compared the performances of different clustering algorithms on the task of detecting remote homologous protein families/super-families. We measured how well SMILESVec-based protein representation describes proteins within a protein clustering task by utilizing two state-of-the-art clustering algorithms; transitive clustering (TransClust) [108] and Markov clustering algorithm (MCL) [109]. We also compare the performance SMILESVec-based protein representation to MACCS and ECFP6 based protein representations as well as sequence based approaches. This chapter is based on the work published as [36].

5.2. Related Work

Proteins are most often described using their sequences [120–122]. The earliest works of classifying proteins into families are the global alignment [130] and local alignment [12] algorithms, and the local alignment model is carried to today with BLAST [71]. Semantic features such as functional categories, annotations and gene ontology classes [131–134] have been proposed to integrate the functional understanding of proteins, yet the description of these features usually in the form of binary vectors prevents the direct use of the available information.

A recent study adapted Word2Vec [94], which has been a popular word-embeddings model in NLP tasks, into the genomic space to describe proteins as low-dimensional continuous vectors using their sequences, and used these vectors to classify proteins [117]. Unlike the explicit integration of the evolutionary information among the amino-acids via Position Specific Scoring Matrix [135] or grouping amino acids into six exchange groups [136], provided a large sequence corpora, a neural-network based model might learn this information implicitly from the raw data. However, modelling the segments/sub-sequences/words that are directly or indirectly associated with functional properties of a protein is still a difficult task.

5.3. Methods

5.3.1. Ligand-centric Protein Representation

Ligand-centric protein representation is based on representing a protein using its interacting ligands. Thus, for each interacting ligand, a vector representation should be constructed. In order to represent proteins we utilized SMILESVec approach. SMILESVec is based on the average of the embeddings of the chemical words (i.e. 8-mers) that are learned via the Word2Vec algorithm [94]. Figure 5.1 illustrates the construction of the ligand-based vector for protein *NDM-1* from the SMILESVec of its interacting ligands, *ampicillin* and *L-captopril*.

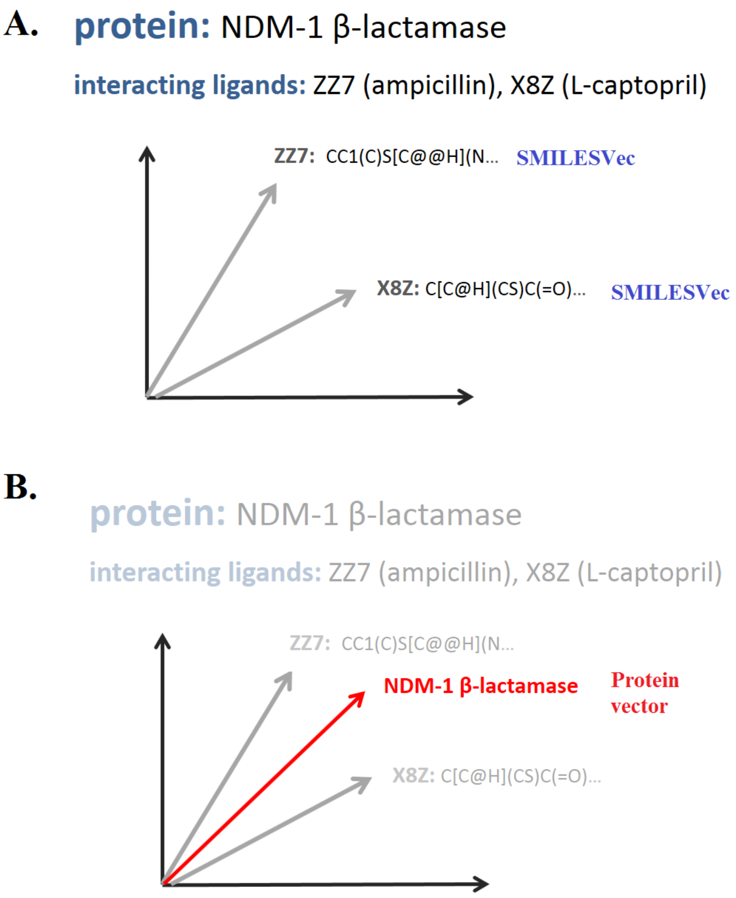

A.
protein: NDM-1 β-lactamase
interacting ligands: ZZ7 (ampicillin), X8Z (L-captopril)
ZZ7: CC1(C)S[C@@H](N...
SMILESVec
X8Z: C[C@H](CS)C(=O)...
SMILESVec
B.
protein: NDM-1 β-lactamase
interacting ligands: ZZ7 (ampicillin), X8Z (L-captopril)
ZZ7: CC1(C)S[C@@H](N...
NDM-1 β-lactamase
Protein vector
X8Z: C[C@H](CS)C(=O)...

Figure 5.1: (A) SMILESVec for each interacting ligand of NDM-1 is constructed. (B) Vector for NDM-1 is built by taking the average of the SMILESVec of the ligands it binds to.

Equation 5.1 describes the construction of a protein vector from its binding ligands, where SMILESVec represents the ligand vector and n_l represents the total number of ligands that the protein interacts with.

$$vector(protein) = \frac{\sum_{k=1}^{n_l} vector(SMILESVec_k)}{n_l} \tag{5.1}$$

While constructing SMILESVec-based protein vectors, we investigated the effect of three factors on SMILESVecs: (i) different types of canonical SMILES, (ii) word or character based learning in ligand representation, and (iii) different techniques for combining word/character embeddings to molecule embedding.

5.3.1.1. Canonical SMILES type. We first examined an important feature of SMILES representation which is the canonical form. Because of the possibility of representing a single molecule with several valid SMILES, canonicalization algorithms were originated for generating a unique SMILES for a molecule. However they couldn't obviate the diverseness that came with different canonicalization algorithms. Therefore, it is no surprise that canonical SMILES definition can change from database to database. ChEMBL utilizes Accelryss̈ Pipeline Pilot that is based on an algorithm derived from Daylight's [137], whereas PubChem uses OpenEye software [138] for canonical SMILES generation [139]. The most apparent difference between the canonical SMILES of two databases is that ChEMBL comprises isomeric and aromatic information, whereas PubChem does not. Thus, although we collected the SMILES of the interacting ligands from the ChEMBL database, we both investigated the cases in which ChEMBL and PubChem canonical SMILES corpora both separately and together (combined) are used to learn embeddings for chemical words and characters.

5.3.1.2. Word versus Character Embeddings. We also investigated whether the use of SMILES characters as the words of the SMILES sequence can provide improvement over 8-mers.

Word embedding. SMILES sequences are divided into 8-mers, and for each chemical word a continuous vector is learned. These vectors, are then combined to represent complete ligands as described in Equation 4.1 in Chapter 4.

Char embedding. Each character of the SMILES string is treated as a word, and thus the Word2Vec model is trained on characters. We created char-level embeddings for 59 and 61 unique characters that occur in SMILES strings in ChEMBL23 and PubChem datasets, respectively.

Equation 5.2 describes $SMILESVec_{char}$ where n in this case represents the total number of the characters in a SMILES.

$$SMILESVec_{char} = vector(ligand) = \frac{\sum_{k=1}^{n} vector(char_k)}{n} \tag{5.2}$$

5.3.1.3. Vector Combination. After obtaining word embedding vectors, we need to combine these into a SMILES (i.e. compound) vector. In Chapter 4 while describing SMILESVec, the word vectors are taken average of to obtain the compound vector. However, we can also describe a protein/ligand vector as the output of the maximum or minimum functions, where m is the total number of the words that are created from the protein/ligand sequence and d is the dimensionality of the vector (i.e. the number of features). MIN_i is equal to the minimum value of the i^{th} feature among m sub-sequences (i.e. words) (Equation 5.3). To build a ligand vector of minimum, MIN_i is selected for each feature as defined in Equation 5.4. Similarly, MAX_i define the maximum value of the i^{th} feature among m sub-sequences (Equation 5.5) and ligand vector of maximum is created as in Equation 5.6 for d number of features. The concatenation of these minimum and maximum protein vectors results in a vector with twice the dimensionality of the original vectors [140]. The min/max representation is explained in Equation 5.7.

$$MIN_i = min([sub\text{-}sequence_{0i}, sub\text{-}sequence_{mi}]) \tag{5.3}$$

$$vector_{min}(protein) = [MIN_0 MIN_1 ... MIN_i ... MIN_d] \tag{5.4}$$

$$MAX_i = max([sub\text{-}sequence_{0i}, sub\text{-}sequence_{mi}]) \tag{5.5}$$

$$vector_{max}(protein) = [MAX_0 MAX_1 ... MAX_i ... MAX_d] \tag{5.6}$$

$$vector_{minmax}(protein) = [vector_{min}(protein)][vector_{max}(protein)] \tag{5.7}$$

5.3.2. Ligand-based Protein Similarity Computation

Ligand-based protein representation model can be utilized to compute protein similarity without using protein sequence. Here, other than SMILESVec, we used fingerprints and bag-of-words based ligand representations to create protein embeddings. Then, we used cosine similarity to compute similarity between two proteins.

5.3.2.1. SMILESVec-based Protein Similarity. Proteins were represented as SMILESVec-based vectors (Equation 5.1) and cosine similarity was used to compute similarity.

5.3.2.2. Fingerprint-based Protein Similarity. We used MACCS and ECFP, two fingerprint-based compound representation methods, to compare against SMILESVec.
MACCS and ECFP are represented with 166 and 1024 bit vectors, respectively. The default settings of Chemical Development Kit [141] was utilized to obtain the MACCS

and ECFP representations of the ligands. For ECFP, we employed ECFP6 which assumes the value of 6 as the maximum diameter. The proteins were represented as explained in Equation 5.8 where fingerprint vectors are utilized to represent each interacting ligand.

$$vector(protein) = \frac{\sum_{k=1}^{n_l} vector(FingerprintMethod_k)}{n_l} \tag{5.8}$$

Fingerprints were used in order to evaluate how competitory a text-based data-driven approach (SMILESVec) is against the popular chemical descriptors.

5.3.2.3. SMILES word frequency-based Protein Similarity. For each interacting ligand of a protein, 8-mers that are extracted from SMILES were utilized as words. Then, the similarity between two proteins was computed as explained in Equation 5.9 using the set of chemical words of their respective interacting ligands. In order to compute the similarity between these two proteins, we adopted the formula depicted in Equation 5.9 [142]:

$$WordFrequency_{sim}(P_1, P_2) = \frac{\sum_{i=1}^{m} 1 - \frac{\left|N_{P_1,i} - N_{P_2,i}\right|}{\left|N_{P_1,i} + N_{P_2,i}\right|}}{m} \tag{5.9}$$

where m is the total number of unique chemical words that are extracted from the interacting ligands of P_1 and P_2, $N_{P_1,i}$ is the frequency of chemical words of type i in protein P_1 and $N_{P_2,i}$ is the frequency of chemical words of type i in protein P_2.

5.3.3. Sequence-based Protein similarity computation

We also utilized protein sequence based approaches to compute protein similarity to be able to compare ligand-based protein representation approach. We used BLAST and ProtVec methods as baseline.

5.3.3.1. BLAST. Basic Local Alignment Tool (BLAST) reveals the similarity between protein sequences using the local alignment algorithm [71]. We used both BLAST sequence identity values and BLAST e-values that were previously obtained by Bernardes and co-workers [129] with all-versus-all BLAST with e-value threshold of 100, for the benchmark dataset.

5.3.3.2. Word Frequency-based Protein Similarity. Word frequency-based protein similarity method utilizes 3-mers as protein words that are created following the procedure explained in Section 3.1.2.3. However, instead of a employing a learning process, the occurrences of the protein words in a protein sequence was counted. The similarity between two proteins was then computed as explained in Equation 5.10:

$$WordFrequency_{sim}(L_1, L_2) = \frac{\sum_{i=1}^{m} 1 - \frac{\left|N_{P_1,i} - N_{P_2,i}\right|}{\left|N_{P_1,i} + N_{P_2,i}\right|}}{m} \tag{5.10}$$

in which m represents the total number of unique words created from protein sequences P_1 and P_2, $N_{P_1,i}$ is the frequency of words of type i in protein P_1 and $N_{P_2,i}$ is the frequency of words of type i in protein P_2.

5.3.3.3. ProtVec-based Protein Similarity. ProtVec is a Word2Vec based model that constructs protein vectors by taking average of the 3-mer sub-sequence vectors that are extracted from the sequence [117]. ProtVec can be formulated as in Equation 5.11:

$$ProtVec = vector(protein) = \frac{\sum_{k=1}^{m} vector(sub\text{-}sequence_k)}{m} \tag{5.11}$$

where $vector(sub\text{-}sequence_k)$ refers to the 100-dimensional continuous vector for the k_{th} sub-sequence and m corresponds to the total number of sub-sequences that can be extracted from a protein sequence. The cosine similarity function was used then, to compute the similarity between two protein vectors $P1$ and $P2$ as in Equation 5.12:

$$CosSim(P1, P2) = \frac{\sum_{i=1}^{d} P1_i P2_i}{\|P1\| \|P2\|} \tag{5.12}$$

where d is the size (dimensionality) of the vectors. Even though the original method used averaging to build protein vectors from 3-mer vectors, in this work we also experimented with minmax combination method as well. To learn 3-mer sub-sequence embeddings, 550K protein sequences from UniProt were used in training Word2Vec.

5.4. Dataset

The ASTRAL datasets are part of Structural Classification of Proteins (SCOP) collection and classified under folds, families and super-families [143]. A family indicates a group of proteins with conventionally distinct functionalities but also with high sequence similarities. A super-family, on the other hand, is a group of protein families with structural and functional similarities amongst families.

The minimum sequence similarity of the proteins that they contain determines the name of the ASTRAL datasets. For instance, ASTRAL50 (A-50) dataset includes proteins with at most 50% sequence similarity. In this work, A-50 dataset from SCOP 1.75 version was utilized as benchmark to demonstrate the performance of the protein representation methods [144]. The proteins were clustered into families and super-families for evaluation. We utilized the same protein pairs that Bernardes and co-workers [129] used for A-50 to compute similarity scores [145] and thus removed the families and super-families with a single protein.

5.5. Evaluation

We utilized the F-measure, precision and recall metrics to assess the performance of the proposed methods which are commonly used in the evaluation of classification methods. We followed the formulation explained by Bernardes and co-workers [129] to adapt these metrics for the evaluation of the clustering task.

For a dataset of n proteins, let us assume n_f represents the number of proteins that belong to the f^{th} family or class, n_g is the number of proteins that are placed in the g^{th} cluster and n_{fg} represents the number of proteins that belong to the f^{th} family and are placed in the g^{th} cluster. Precision of cluster g with respect to the f^{th} family is computed as $precision_{fg} = n_{fg}/n_g$, whereas recall is defined as $recall_{fg} = n_{fg}/n_f$. Finally we can define F-measure as in Equation 5.13:

$$F\text{-}measure = \frac{1}{n}\sum_{f} n_f max_g \frac{2precision_{fg}recall_{fg}}{precision_{fg} + recall_{fg}} \tag{5.13}$$

max_g indicates that for each family f, we compute precision and recall values for each cluster g, and choose the maximum resulting F-score. The weighted mean precision and recall are described in Equations 5.14 and 5.15, respectively [129].

$$Precision = \frac{1}{n}\sum_{f} n_f max_g precision_{fg} \tag{5.14}$$

$$Recall = \frac{1}{n}\sum_{f} n_f max_g recall_{fg} \tag{5.15}$$

We also utilized Pearson Correlation [146] to measure the linear correlation between similarity methods X and Y, which can be described as in Equation 5.16:

$$P(X,Y) = \frac{cov(X,Y)}{sd(X)sd(Y)} \tag{5.16}$$

in which *cov* indicates covariance function and *sd* indicates standard deviation.

5.6. Results

We evaluated the performance of five different protein similarity computation approaches in clustering of the A-50 dataset. These were BLAST, ProtVec, SMILESVec, MACCS, and ECFP, the first two of which are protein sequence based similarity methods, whereas the latter three utilize the ligands to which proteins bind. We accepted word-frequency based protein similarity methods that use protein sequences and compound SMILES strings, respectively, as the baseline. Average (avg) and minimum/maximum (min/max) of the vectors were taken to build combined vectors for ProtVec and SMILESVec from their word vectors.

We performed our experiments on the A-50 dataset using two different clustering algorithms, TransClust and MCL. The ligand-based (SMILESVec, MACCS and ECFP) protein representation approaches require a protein to bind to at least one ligand in order to define a ligand-based vector for that protein. Therefore, we removed the proteins with no binding ligands from both datasets. Table 5.1 provides a summary of the A-50 dataset before and after filtering.

Table 5.1: Distribution of families and super-families in A-50 dataset before and after filtering.

dataset	Num. Sequences	Super-families	Families
Before filtering	10816	1080	2109
After filtering	1639	425	652

When the set of proteins that remain in our dataset are examined, we observed that some of the superfamilies/families that were initially in the top-10 most frequent family and super-family lists are replaced by others. Among the superfamilies that are no longer in the most frequent list are “Winged helix” DNA - binding domain and thioredoxin - like superfamilies because the number of known ligands is lower. On the other hand, super-families and families that weren’t initially in the top-10 list such as Protein-kinase like (d.144.1) super-family and nuclear-receptor binding domain (a.123.1) and their respective descendant families make it to the frequent set of proteins when ligand interactions are taken into account. Table 5.2 summarizes the top-10 most frequent family and super-families with known ligand interactions.

Table 5.2: Distribution of the top-10 most frequent super-families and families with known ligand interactions.

	Super-family	**# prots.**	**Family**	**# prots.**
1	Protein kinase-like (d.144.1)	47	Protein kinases, catalytic subunit (d.144.1.7)	39
2	P-loop containing nucleoside triphosphate hydrolases (c.37.1)	43	Fibronectin type III (b.1.2.1)	28
3	Immunoglobulin (b.1.1)	41	Eukaryotic proteases (b.47.1.2)	25
4	NAD(P)-binding Rossmann-fold domain (c.2.1)	32	EGF-type module (g.3.11.1)	24
5	Trypsin-like serine proteases (b.47.1)	31	Immunoglobulin I set (b.1.1.4)	23
6	Fibronectin type III (b.1.2)	28	SH2 domain (d.93.1.1)	22
7	EGF/Laminin (g.3.11)	27	Nuclear receptor ligand-binding domain (a.123.1.1)	18
8	SH2 domain (d.93.1)	22	Cyclin (a.74.1.1)	15
9	Cysteine proteinases (d.3.1)	20	Pleckstrin-homology domain (b.55.1.1)	15
10	Nuclear receptor ligand-binding domain (a.123.1)	19	Tyrosine-dependent oxidoreductases (c.2.1.2)	15

In the filtered dataset where all proteins have an interacting ligand, there are 1057 proteins with fewer than 200 ligands (64% of all proteins) and 101 proteins with single ligands (0.6% of all proteins). There are 67 proteins with more than 10000 interacting ligands (0.4%), thus increasing the mean number of the interacting ligands to 1791. The protein with the highest number of interacting ligands is d2dpia2 (DNA polymerase iota), a protein involved in DNA repair [147] and implicated in esophageal squamous cell cancer [148] and breast cancer [149], with 115018 ligands.

We assessed the performance of the clustering algorithms with F-measure values for two different clustering scenarios, family and super-family clustering. We also provided Precision and Recall values for each of the methods. In clustering, high recall indicates that the method assigns a high number of proteins from the same family/super-family to the same cluster. High precision, on the other hand, means the assigned clusters contain high percentage of proteins that belong to the same family/super-family. Higher precision values indicate that the clusters are more homogeneous, i.e., mostly contain proteins from the same families/supefamilies.

Tables 5.3 and 5.4 report the Precision, Recall and F-measure values for family and super-family clustering and the number of clusters that are detected with the TransClust algorithm, and Tables 5.5 and 5.6 report the same metrics for family and super-family clustering with the MCL algoritm, respectively. Between TransClust and MCL, TransClust produced better F-measure values in all representation methods on the A-50 dataset. The results obtained by both clustering algorithms were better in family clustering than in super-family clustering, which was an expected outcome, since detection of relationships between distantly related proteins is a much harder task.

To group proteins, both clustering algorithms utilized their similarity scores. Among the protein sequence-based similarity methods, the poorest clustering performance with F-measure metric in super-family/family (0.350/0.500) belonged to BLAST with e-value, the baseline. Protein word frequency obtained the best performance on the A-50 dataset in super-family and family clustering (0.686/0.744). The performance of the ProtVec Avg (0.681/0.739) and the ligand-based protein representation methods followed the best result closely. Bringing in a semantic aspect with learning through the Word2Vec model, ProtVec-based similarity (avg and minmax), was outperformed

Table 5.3: Precision, Recall and F-measure values for all protein similarity computation methods in super-family clustering with TransClust algorithm. The values indicated in bold shows the best F-measure for the Protein sequence and ligand based methods.

		Super-family			
		No.Clusters	Precision	Recall	F-measure
Protein sequence based					
Blast (e-val)	A-50	1596	0.997	0.261	0.350
Blast (identity)	A-50	606	0.861	0.550	0.595
Protein Word frequency	A-50	708	0.952	0.621	**0.686**
ProtVec Avg (word)	A-50	655	0.927	0.620	0.681
ProtVec Avg (char)	A-50	707	0.940	0.603	0.674
ProtVec MinMax (word)	A-50	586	0.891	0.623	0.667
Ligand based					
SMILES Word frequency	A-50	801	0.951	0.548	0.624
SMILESVec (word, chembl)	A-50	621	0.921	0.621	0.677
SMILESVec (word, PubChem)	A-50	573	0.888	0.627	0.668
SMILESVec (word, combined)	A-50	617	0.923	0.627	0.675
SMILESVec (char, chembl)	A-50	636	0.920	0.621	0.678
SMILESVec (char, PubChem)	A-50	714	0.941	0.600	0.671
SMILESVec (char, combined)	A-50	712	0.949	0.602	0.675
MACCS	A-50	589	0.909	0.629	**0.679**
ECFP6	A-50	611	0. 917	0.627	**0.679**

by the straightforward word-frequency based approach.

The results also showed that the average-based combination method (ProtVec avg) was better than the min/max-based combination method (ProtVec minmax) to build a single protein vector from sub-sequence vectors in the protein clustering task. Since min/max-based combination method did not perform well in sequence-based protein similarity, we did not test the technique for SMILES-based protein similarity approaches.

Among the ligand based representation methods, we examined the performance of the word-based embeddings and character-based embeddings as well as the effect of the source of the training dataset on the embeddings. We collected canonical SMILES

Table 5.4: Precision, Recall and F-measure values for all protein similarity computation methods in family clustering with TransClust algorithm. The values indicated in bold shows the best F-measure for the Protein sequence and ligand based methods.

		Family			
		No.Clusters	Precision	Recall	F-measure
Protein sequence based					
Blast (e-val)	A-50	1636	1.0	0.399	0.500
Blast (identity)	A-50	660	0.781	0.668	0.631
Protein Word frequency	A-50	688	0.844	0.777	**0.744**
ProtVec Avg (word)	A-50	704	0.845	0.757	0.739
ProtVec Avg (char)	A-50	707	0.842	0.746	0.729
ProtVec MinMax (word)	A-50	704	0.829	0.741	0.718
Ligand based					
SMILES Word frequency	A-50	957	0.934	0.658	0.704
SMILESVec (word, chembl)	A-50	730	0.855	0.744	0.735
SMILESVec (word, PubChem)	A-50	692	0.839	0.751	0.730
SMILESVec (word, combined)	A-50	764	0.873	0.732	0.735
SMILESVec (char, chembl)	A-50	710	0.844	0.743	0.729
SMILESVec (char, PubChem)	A-50	715	0.845	0.744	0.729
SMILESVec (char, combined)	A-50	712	0.850	0.749	**0.739**
MACCS	A-50	683	0.839	0.757	0.736
ECFP6	A-50	725	0.860	0.746	0.733

from both ChEMBL (∼1.7M) and PubChem (∼2.3M) databases. The SMILES strings of the interacting ligands were only collected from ChEMBL. The main difference between these two databases is that ChEMBL allows the isomeric information of the molecule to be encoded within SMILES. The results indicated that the choice of the SMILES corpus in which the word-embeddings are trained on should be considered carefully, since even slight changes in the notation of SMILES, affects the formation of the chemical words directly. In our case, since the SMILES of the interacting ligands of the A-50 dataset were collected from the ChEMBL database, the performance of SMILESVec in which embeddings were learned from training with ChEMBL SMILES rather than PubChem SMILES was notably better. We also investigated whether using the combination of the SMILES corpus of ChEMBL and PubChem can improve the performance of SMILESVec embeddings. We indeed reported an improvement on

Table 5.5: Precision, Recall and F-measure values for all protein similarity computation methods in super-family clustering with Markov Clustering (MCL) algorithm. The values indicated in bold shows the best F-measure for the Protein sequence and ligand based methods.

		Super-family			
		No.Clusters	Precision	Recall	F-measure
Protein sequence based					
Blast (e-val)	A-50	728	0.792	0.271	0.290
Blast (identity)	A-50	783	0.882	0.496	0.540
Protein Word frequency	A-50	411	0.769	0.625	0.590
ProtVec Avg (word)	A-50	1001	0.964	0.514	**0.596**
ProtVec Avg (char)	A-50	1017	0.964	0.508	0.590
ProtVec MinMax (word)	A-50	1014	0.964	0.508	0.590
Ligand based					
SMILES Word frequency	A-50	312	0630	0.550	0.470
SMILESVec (word, chembl)	A-50	867	0.937	0.544	**0.608**
SMILESVec (word, PubChem)	A-50	857	0.931	0.544	0.604
SMILESVec (word, combined)	A-50	894	0.940	0.540	0.607
SMILESVec (char, chembl)	A-50	999	0.962	0.514	0.596
SMILESVec (char, PubChem)	A-50	977	0.958	0.514	0.595
SMILESVec (char, combined)	A-50	1006	0.963	0.514	0.595
MACCS	A-50	874	0.936	0.540	0.606
ECFP6	A-50	618	0.863	0.582	0.599

character-based embedding in family clustering (0.739 F-measure) whereas word-based embedding produced F-measure values higher than the PubChem-based learning and lower than the ChEMBL-based learning. We can suggest that the increase in the performance of the character-based learning with the combination of two different SMILES corpora might be positively correlated with the increase in SMILES samples, while the number of unique letters that appear in the SMILES did not significantly change between databases (e.g. absence/presence of the few characters that represent isometry information). However, with the word-based learning, we observed that there was significant increase in the variety of the chemical words, thus the combined SMILES corpus model did not work as well as it did in character-based learning. This result suggests that the size of the learning corpus may affect the representation of the embed-

Table 5.6: Precision, Recall and F-measure values for all protein similarity computation methods in family clustering with Markov Clustering (MCL) algorithm. The values indicated in bold shows the best F-measure for the Protein sequence and ligand based methods.

		Family			
		No.Clusters	Precision	Recall	F-measure
Protein sequence based					
Blast (e-val)	A-50	728	0.687	0.406	0.379
Blast (identity)	A-50	783	0.803	0.622	0.592
Protein Word frequency	A-50	411	0.643	0.767	0.606
ProtVec Avg (word)	A-50	1001	0.909	0.639	**0.665**
ProtVec Avg (char)	A-50	1017	0.910	0.633	0.662
ProtVec MinMax (word)	A-50	1014	0.909	0.634	0.662
Ligand based					
SMILES Word frequency	A-50	312	0.497	0.686	0.475
SMILESVec (word, chembl)	A-50	867	0.870	0.672	0.667
SMILESVec (word, PubChem)	A-50	857	0.861	0.673	0.664
SMILESVec (word, combined)	A-50	894	0.877	0.666	0.668
SMILESVec (char, chembl)	A-50	999	0.908	0.641	0.668
SMILESVec (char, PubChem)	A-50	977	0.900	0.643	0.667
SMILESVec (char, combined)	A-50	1006	0.909	0.641	**0.669**
MACCS	A-50	874	0.866	0.668	0.667
ECFP6	A-50	618	0.762	0.710	0.631

dings, and a larger SMILES corpus could lead to better character-based embeddings for SMILESVec.

Considering only ChEMBL trained SMILESVec, word-based approach was slightly better than character-based SMILESVec in terms of F-measure in family clustering. In super-family clustering however, character-based approach performs as well as word-based SMILESVec. Similarly, ProtVec is also better represented in word-level rather than character-level.

The ligand-based protein representation methods, SMILESVec and MACCS-based approach performed almost as well as ProtVec in family and super-family clustering with TransClust algorithm, even though no protein sequence information was used.

A lower clustering performance was obtained with MCL than with to TransClust, and both SMILESVec and MACCS-based method produced slightly better F-measure than ProtVec Avg in both super-family and family clustering. Since ligand-based protein representation methods capture indirect function information through ligand binding, they were recognizably better at detecting super-families than families compared to sequence-based ProtVec on a relatively distant dataset. Furthermore, SMILESVec, a text-based unsupervised learning model, produced comparable F-measure values to MACCS and Extended-Connectivity Fingerprints, which are binary vectors based on human-engineered and hash-based feature descriptions, respectively.

Table 5.7 reports the Pearson correlations [146] among the protein similarity computation methods. Comparison with BLAST e-value resulted in a negative correlation, as expected, since e-values closer to zero indicate high match (similarity). Ligand based protein representation methods had higher correlation values with BLAST e-value than protein-sequence based methods. We also observed strong correlation among the ligand-based protein representation methods, suggesting that, regardless of the ligand representation approach, the use of interacting ligands to represent proteins provides similar information.

We further investigated a case in which similar super-family clusters were produced with SMILESVec-based protein similarity and ProtVec protein similarity using the TransClust algorithm. We chose one of the medium-sized clusters for manual inspection. We observed that Fibronectin Type III proteins (7 proteins) were clustered together when SMILESVec was used, whereas using ProtVec placed them into four different clusters; one cluster contained four of those proteins, another cluster contained a single protein and the other two proteins were part of other clusters. The protein that was clustered by itself (SCOP ID:d1n26a3, Human Interleukin-6 Receptor alpha chain) had two interacting ligands (CHEMBL81;Raloxifene and CHEMBL46740;Bazedoxifene) that were also shared by a protein (SCOP ID:d1bqua2,Cytokine-binding region of GP130) clustered separately with ProtVec. Thus, we can suggest that using information on common interacting ligands, SMILESVec achieved to combine these seven proteins into a single cluster, while ProtVec failed to do so with a sequence-based approach.

Table 5.7: Pearson correlation between protein similarity methods.

Method	Method	Pearson correlation
BLAST (e-value)	BLAST (identity)	-0.109
BLAST (e-value)	Protein word frequency	-0.250
BLAST (e-value)	ProtVec (avg)	-0.291
BLAST (e-value)	SMILESVec (word, chembl)	-0.335
BLAST (e-value)	SMILESVec (char, chembl)	-0.207
BLAST (e-value)	MACCS	-0.336
SMILESVec (word, chembl)	MACCS	0.895
SMILESVec (char, PubChem)	MACCS	0.590
SMILESVec (word, chembl)	SMILESVec (char, PubChem)	0.682
SMILESVec (word, chembl)	ECFP6	0.933
ECFP6	MACCS	0.898

We would like to mention that ASTRAL datasets contain domains rather than full length proteins, while CHEMBL collects protein - ligand interaction information based on the whole protein sequence from UniProt. A multidomain protein may have multiple and diverse chemotypes of ligands binding to each domain and retrieving ligand information based on the full length protein may lump this disparate information together, leading to loss of information on domain specific ligand interactions. The performance of domain sequence based methods is therefore at an advantage because family/superfamily assignment in SCOP is also based on domain sequence, while the ligand based approach we use in SMILESVec uses more noisy data. Despite this disadvantage, ligand based approach performs as well as the sequence based approaches.

Due to the domain based nature of the ASTRAL datasets, clustering based on the full protein sequence can lead to a reduction in performance because of the presence of multidomain proteins. Similarly, we hypothesized that the ligand-based methods might not show their true performance, since the interactions collected from ChEMBL are based on protein-ligand interactions and not domain-ligand interactions. For instance, the domains d2nxyb1 and d2nxyb2 belong to different families, b.1.1.1 and b.1.1.3, respectively. If the ligands that bind to each of these domains were known, the performance of the ligand-based models might have improved. However, in our current setting, for each of these domains, we collected the same interacting ligands from ChEMBL, since their target identifiers are the same. Therefore, as expected we observed that these two domains were clustered together with ligand-based protein representation methods leading to a decrease in F-measure.

To test our methodology on single domain proteins of the A-50 dataset, we created a subset that contains only single domains and another that contains the rest of the sequences. SCOP stable domain identifier (sid) uses 7-charactered system in which the last character defines the domains uniquely (e.g., d2sqca**1**, d2sqca**2** for several domain or d1n4qb_ when there is no need for domain specification). The single domain subset comprised sequences with sid ending with the "_" character. Using the predicted clusters, we measured how accurately proteins of the single-domain were assigned by computing the percentage of True-Positives (TP) (N_{TP}/N) where N is the number of the samples in the subset and N_{TP} is equal to the number of the correctly clustered samples of the subset. As expected, when only single domains were considered, we observed that both Protvec and SMILESVec had higher percentage of TPs. The performance of SMILESVec was increased from 0.743 for all proteins to 0.82 for single domain proteins. ProtVec had a slightly less pronounced increase from 0.757 to 0.829. On the other hand, when multidomain proteins were taken into account, the TP percentage reduced to 0.671 (SMILESVec) and 0.689 (ProtVec). These results suggest that taking domain information into account can enhance the performance of these representation methods.

5.7. Conclusion

In this study, we first propose to represent proteins using their interacting ligands. In this approach, the interacting ligands of each protein in the dataset are collected. Then, the SMILES string of each ligand is divided into fixed-length overlapping substrings (i.e. words, 8-mers). These created substrings are then used to build real-valued vectors using the Word2vec model and then the vectors are combined into a single vector to represent the whole SMILES string (i.e. SMILESVec). Finally, protein vectors are constructed by taking the average of the vectors of their ligands.

The effectiveness of the proposed method in describing the proteins was measured by performing clustering on the ASTRAL 50 (A-50) dataset from the SCOP database using two different clustering algorithms, TransClust and MCL. Both of these clustering algorithms use protein similarity scores to identify cliques. SMILESVec-based protein representation was compared with other protein representation methods, namely BLAST and ProtVec, both of which depend on protein sequence to measure protein similarity, and the MACCS and ECFP binary fingerprint based ligand-centric protein representation approaches. The performance of the clustering algorithms, as reported by F-measure, showed that protein word-frequency based similarity model was a better alternative to BLAST e-value or sequence identity to measure protein similarity. Furthermore, ligand-based protein representation methods also produced comparable F-measure scores to ProtVec.

Using SMILESVec, we were able to define proteins based on their interacting ligands even in the absence of sequence or structure information. SMILESVec-based protein representation had better clustering performance than BLAST and comparable clustering performance to protein word-frequency based method, both of which use protein sequences. We should emphasize that SCOP datasets were constructed based on protein similarity, thus high performance with the protein sequence-based models in family/super-family clustering is no surprise. However, the fact that ligand-based protein representation methods, either learning from SMILES or represented with binary compound features, perform as well as protein sequence-based models is quite intriguing and promising.

SMILESVec, MACCS and ECFP representations performed similarly in the task of protein clustering, suggesting that the word-embeddings approach that learns representations from a large SMILES corpus in an unsupervised manner is as accurate as widely adopted Fingerprint models. We propose that the ligand-based representation of proteins might reveal important clues especially in protein-ligand interaction related tasks like drug specificity or identification of proteins for drug targeting. The similarity between a candidate ligand and the SMILESVec for a protein can be used as an indicator for a possible interaction.

The study we conducted here also showed that SMILES description is sensitive to the database definition conventions, therefore the use of SMILES strings requires careful consideration. Since the protein-ligand interaction and ligand SMILES information are obtained from ChEMBL database to represent proteins, building SMILESVec vectors from the chemical words trained in ChEMBL SMILES corpus yielded better F-measure than the model in which the PubChem SMILES corpus was used for training of the chemical words.

6. DRUG-TARGET BINDING AFFINITY PREDICTION

6.1. Introduction

Identification of high affinity drug-target interactions is a significant first step in the drug discovery pipeline. The development of novel drugs is an expensive and resource-consuming process and the repurposing/repositioning of existing approved drugs is a major alternative [3]. Thus, exploiting the available protein - drug interaction knowledge can provide a good starting point in drug repurposing studies. Furthermore, understanding bimolecular recognition between proteins and drugs can also provide valuable information for generation of novel drugs using generative models [28,150,151].

Drug-target interaction prediction has often been investigated as a binary classification problem [13, 14, 33, 152–155], but recent studies have also been focusing on the prediction of the strength of the interaction between the drug and its target [24, 25]. Binding affinity is expressed in dissociation constant (K_d), inhibition constant (K_i), or the half maximal inhibitory concentration (IC50) values. The prediction of binding affinity for novel interactions is still a challenging task because (i) representation of proteins and ligands in the computational space is complicated by the inherent three-dimensional nature of the binding, (ii) there are only 14,761 protein - ligand complex structures in PDBBind [31] in which the interaction mode is reported, (iii) the chemical space sampled by the currently available data is limited, and (iv) the prediction algorithm needs to take the level of noise in experimental measurements into account. As the number and reproducibility of the available protein - ligand interaction data increases, utilizing this large dataset provides access to a larger chemical space, and a reduction in signal to noise ratio.

In this part of the thesis, we introduce two text-based approaches to address the drug-target binding affinity prediction problem: (i) a machine learning approach that combines the protein and ligand representations that we have presented (Chapters 4 and 5) so far, under a linguistic perspective, and (ii) a deep learning approach that simply utilizes raw textual data of proteins and compounds. The field of chemical linguistics that brings the chemistry and linguistics domains together is growing since its initial inception in the 1960s [156] with efforts that focus on identifying information rich patterns and important keywords. The works presented here aim to benefit from the rich textual information that the biochemical data has in order to enhance the available drug-target binding affinity prediction models.

In the first work, motivated by the promising results of SMILESVec in the protein clustering task, we introduce a strictly SMILES-based methodology to predict drug-target binding affinity. The novelty of this approach is that we combine the ligand-centric protein representation based on SMILESVec with SMILESVec in predicting binding affinity. SMILESVec combines the embedding vectors of the chemical words extracted from SMILES to build a vector representation for the whole SMILES and the proteins are represented as the average of the SMILESVec vector of their interacting ligands. We also introduce two modifications to further analyze the drug-target binding system. First, instead of using all interacting ligands (with either high or low binding affinity) we only choose the ligands that bind to the protein with high affinity to represent the protein. In this way, novel ligands with high possibility of binding to the target might be highlighted. Second, we investigated different techniques to extract “chemical words” from the SMILES strings. The original SMILESVec algorithm was based on 8-mers, but we also performed experiments using Byte-Pair Encoding (BPE) and Maximum Common Substructure (MCS) techniques to extract words.

We also adopted a recent approach, namely DeepSMILES, which introduces a new syntax for SMILES representation [52]. DeepSMILES transforms the regular SMILES syntax by updating the use of ring closure digits and paired parantheses that are used to represent branching. We investigated the effect of using DeepSMILES syntax in prediction task. Finally, the IDF weighting was utilized in ligand-based protein representation to emphasize rare chemical words which might be important for the binding functionality of the protein.

In the second work, instead of defining the words of the proteins and ligands explicitly, we benefited from a deep-learning based approach to learn the important patterns (i.e. words) of the biochemical data from their raw representations. We adopted Convolutional Neural Networks (CNN) [157], which are good at capturing local patterns, to learn abstract representation from the protein sequences and SMILES strings. The high-level representations were then fed into a feed-forward neural network to predict drug-target binding affinity with a promising performance. The model, which we named DeepDTA, contained two CNN blocks, each for protein and compound representation, and a feed-forward neural network on top to perform the prediction from the CNN-based representations.

We reported the performance of SMILES-based drug-target binding affinity prediction on two different datasets: Kinase KIBA dataset [158] and BindingDB(BDB) dataset [7] that we collected and filtered. For DeepDTA, other than these two datasets, a smaller Kinase dataset Davis [159] was also utilized as a benchmark. We compared our results with two recent state-of-the-art studies that employ traditional machine learning methods to predict binding affinity of drug-target pairs, namely KronRLS [24] and SimBoost [25] using Concordance Index (CI) and Mean Squared Error (MSE) metrics.

This chapter is based on the works published as DeepDTA [38] and the updated version of the arXiv paper on chemical language processing based approach for binding affinity prediction [37].

6.2. Related Work

Molecular docking, which is a computational technique to identify two main features of the target and the small molecule: (i) the binding pose and (ii) the binding affinity [22], is an essential part of drug discovery. The determination of the binding affinity of a drug-target pair is an important component of many tasks such as virtual screening, target druggability, and protein function prediction [160].

Scoring functions rank and score protein-ligand complexes in order to predict binding affinity and classify between active and inactive compounds (virtual screening) [161]. Existing studies focused on scoring are usually categorized under three main classes: force-field, knowledge-based and empirical. Using different features of the protein-ligand complexes, these approaches have been dependent on a scoring function in which the parameters are estimated from experimental and computational data [22]. Furthermore, these functions are also reported to have low correlation with experimental binding along with the drawback of taking too much time and being hard to implement [162].

Non-parametric machine learning methods, which enables flexibility by learning the required parameters from data, have been used as an alternative to scoring functions as of last decade [22, 163]. Among several regression methods such as Linear Regression and Support Vector Regression, Random Forest (RF) algorithm has been widely adopted by the studies that pursue machine-learning based scoring prediction. RF is dessigned as an ensemble of decision trees in which RF-score is measured by the vote of all trees. The first study to employ RF to protein-ligand scoring used the frequency of interacting protein-ligand atom pairs within a special distance as feature and reported a success over traditional scoring functions including widely-used X-Score and SYBYL [22].

The first applications of machine learning in predicting the drug-target interactions were classification models that employed either similarity-based [13, 15, 164–166] or feature-based [26, 27, 167–171] representation of compounds and proteins by employing mostly 1D or 2D based representations, since 3D structure of molecules is not always available. Even though similarity-based methods performed well, several studies showed that predicting targets for compounds based on similarity is a lot more complicated process since similar drugs might have different mechanisms [172, 173]. Feature-based approaches, on the other hand, either utilized binary feature vectors that are encoded based on the existence of different structural properties or were dependent of different tools. Pahikkala and co-workers were first to propose the prediction of binding affinity values and approached DTI identification as a regression problem. They employed Kronecker Regularized Least Squares (KronRLS) algorithm on a similarity-based model [24]. The results showed that there is much to improve, especially for predicting S4 interactions. SimBoost method was then proposed to predict binding affinity scores and prediction intervals for a DT pair using KronRLS [25].

Gabel and co-workers, however, speculated machine-learning based methods that use features such as co-occurence of atom-pairs etc. might over-simplify the description and lead to the loss of information that the raw interaction complex could provide [174]. Around the same time this study published, deep learning has become a very popular architecture powered by the big data and high capacity computing machines challenging machine learning methods. Several studies have been already proposed to use deep learning architectures, which are better at handling the raw data and learning hidden patterns from it, for protein-ligand interaction scoring.

More recently, there has been a significant increase in the number of studies that employed deep learning architectures such as Convolutional Neural Networks (CNNs) to predict binding affinity [28, 29, 161, 175, 176]. These studies utilized 3D-based information of drug-target complex to address point (i). The major drawback of these approaches is that the available information on 3D structure of the protein - compound complex is limited compared to the sequence information of proteins and compounds as stated in point (ii). Therefore, a sequence based approach can take advantage of the increasing wealth of information on protein - drug recognition. String based approaches take advantage of the tools and algorithms developed in the natural language processing (NLP) domain.

6.3. Datasets

Binding affinity describes the strength of the intermolecular interaction between the small molecule and its target. It is usually measured by disassociation constant (K_d), inhibition constant (K_i) and the half maximal inhibitory concentration (IC50). Most studies prefer using K_i/K_d data since IC50 depends on the concentration of target and ligand along with other conditions [177]. Low K_i/K_d means high binding affinity (e.g. good inhibitors have around K_i $1n$M or better.) and is usually represented in terms of pK_d or pK_i, the negative logarithm of binding or inhibition constants. Most studies choose to preprocess their protein-ligand interaction data to build high-quality data based on pre-defined conditions such as resolution, elements in the ligands etc, using only K_i/K_d data. The final aim of a model is to predict binding affinity values. Recent studies pointed out that inclusion of low quality data in the training set (i.e. IC_{50}) improves the performance of the test set [23, 160].

We evaluated our proposed models on three different datasets: Davis Kinase dataset [159], KIBA Kinase dataset [158], and BindingDB (BDB) dataset we collected from BindingDB database [7]. Both Davis and KIBA were previously used as benchmark dataset for binding affinity prediction evaluation [24]. Table 6.1 reports the dataset statistics for these datasets.

Table 6.1: Binding affinity dataset statistics.

	Proteins	Compounds	Interactions
Davis (K_d)	442	68	30056
KIBA	229	2111	118254
BindingDB (BDB)	490	924	31368

Davis. The Davis dataset contains selectivity assays of the kinase protein family and the relevant inhibitors with their respective disassociation constant (K_d) values. It comprises interactions of 442 proteins and 68 ligands. While [24] used the K_d values of the Davis dataset directly, we used the values transformed into log space pK_d similarly to [25] as explained in Equation 6.1.

$$pK_d = -log10(\frac{K_d}{1e9}) \tag{6.1}$$

KIBA. KIBA dataset is originated from an approach called KIBA, in which kinase inhibitor bioactivities from different sources such as K_i, K_d, or IC_{50} were combined [158]. KIBA scores were constructed to optimize the consistency between K_i,

K_d, or IC_{50} by using statistical information they held within. KIBA dataset originally comprised 467 targets and 52498 drugs, but filtered by [25] to contain only drugs and targets with at least 10 interaction observations yielding to total 229 unique proteins and 2111 unique drugs.

BindingDB (BDB). BDB dataset is collected from BindingDB database and filtered based on several criteria:

- only K_d values are kept and Equaton 6.1 is applied to convert K_d values to pK_d
- high affinity experiment is chosen if there were multiple instances of the same pair,
- proteins with at least 6 interactions, and
- compounds with at least 3 interactions are kept.

Figure 6.1 top histogram illustrates the distribution of the binding affinity values in pK_d form for Davis dataset. We can clearly observe the peak at pK_d value 5 ($10000nM$) which constitutes more than half of the dataset (20931 out of 30056). These values correspond to the negative pairs that either have very weak binding affinities ($K_d > 10000nM$) or are not observed in the primary screen [24]. The distribution of the KIBA scores was depicted in the middle histogram of Figure 6.1. We used pre-processed KIBA scores which were updated by first, taking the negative of each value and then adding to minimum value to all scores. Finally, the bottom histogram in 6.1 illustrates the distribution of the pK_d values in BindingDB dataset.

The protein sequences of the Davis dataset were extracted from the UniProt protein database based on gene names/RefSeq accession numbers [69]. Similarly, UniProt IDs of target in KIBA and BDB datasets were used to collected protein sequences.

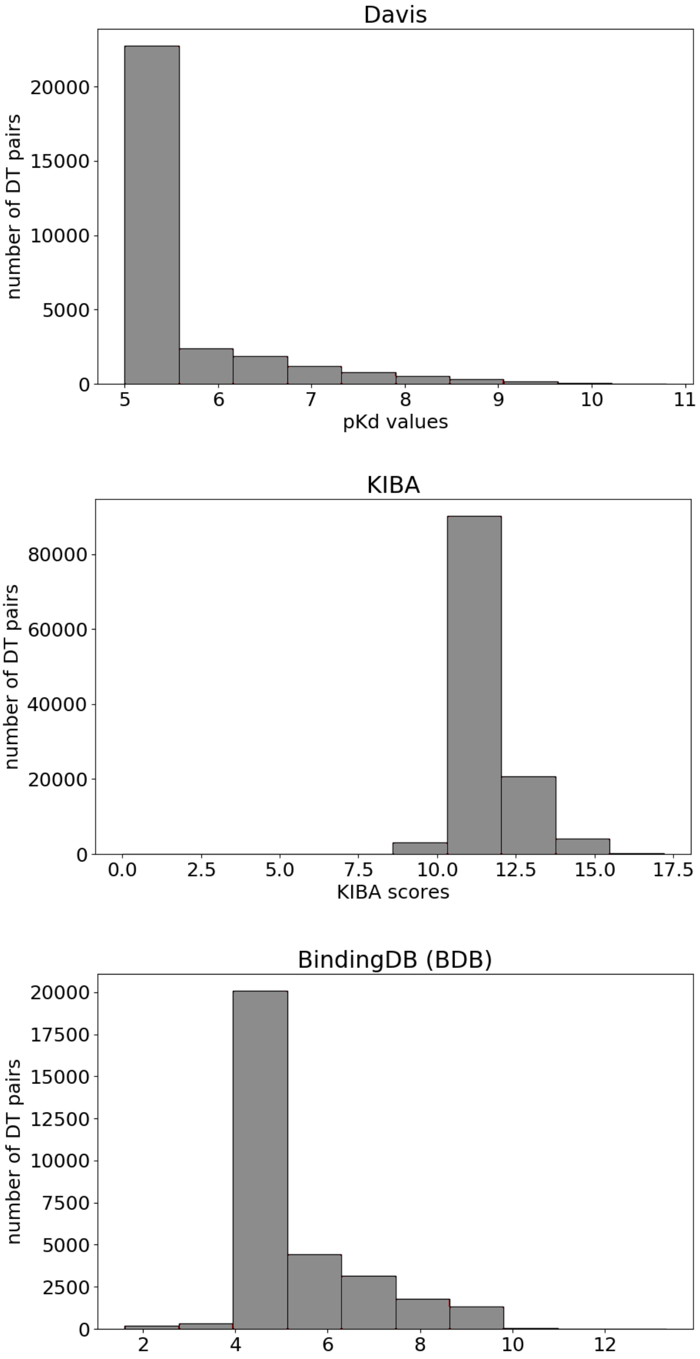


Figure 6.1: Distribution of binding affinity values of Davis (top), KIBA (center) and BindingDB (bottom) datasets.

Figure 6.2 (top) right side shows the lengths of the sequences of the proteins in the Davis dataset. The maximum length of a protein sequence is 2549 and the average length is 788 characters. Figure 6.2 right side depicts the distribution of protein sequence length in KIBA targets. The maximum length of a protein sequence is 4128 and the average length is 728 characters.

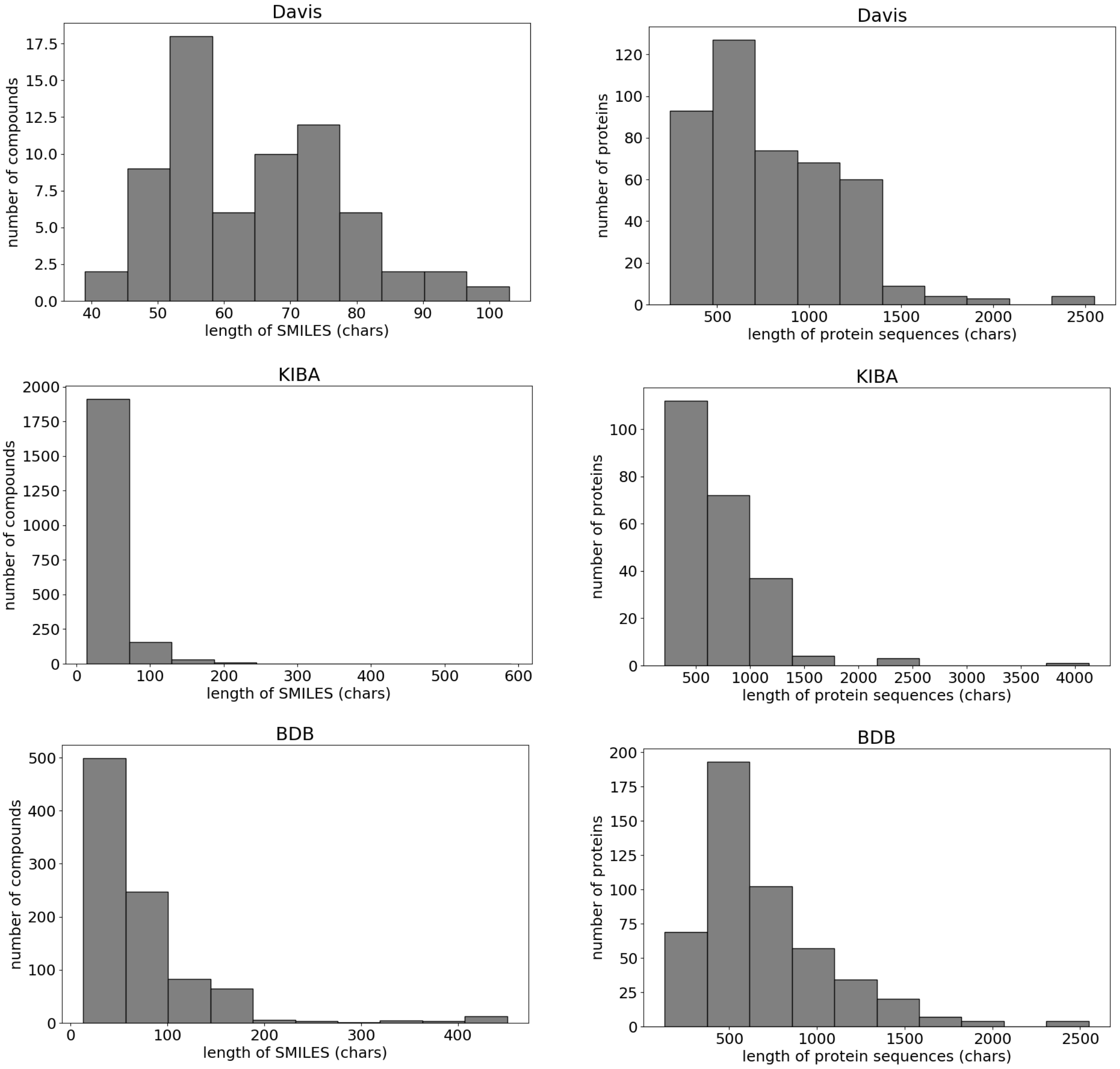


Figure 6.2: Distribution of SMILES and protein sequences of Davis (top), KIBA (center) and BindingDB (bottom) datasets.

The compound SMILES strings were extracted from the PubChem [54] and ChEMBL [6] databases. Figure 6.2 (left side) illustrates the distribution of the lengths of the SMILES strings of the compounds in the Davis, KIBA and BDB datasets, respectively. Among the compounds of Davis, the maximum length of a SMILES is 103, while the average length is equal to 64. The distribution of SMILES character length of KIBA drugs is illustrated in Figure 6.2 (middle, right side). the maximum length of a SMILES is 590, while the average length is equal to 58.

Figure 6.3, on the other, hand depicts the heatmaps for the S-W based protein similarity and PubChem structure based compound similarity matrices for KIBA, Davis and BindingDB datasets, respectively.

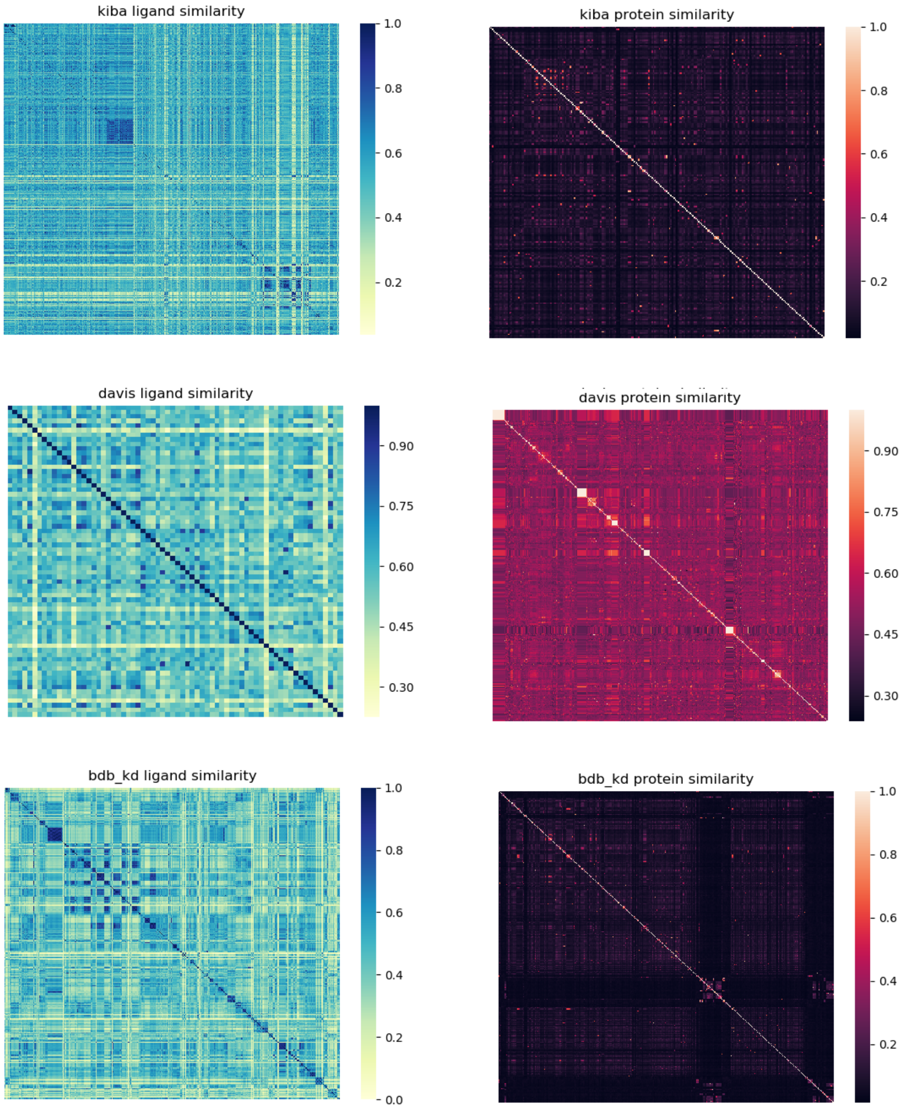


Figure 6.3: Illustration of a protein and ligand similarity matrices for three benchmark datasets.

In order to learn a generalized model, we randomly divided each dataset into six equal parts in which one part is selected as the independent test set. The remaining parts of the dataset were used to determine the hyper-parameters via five-fold cross validation. The same setting was run for KronRLS [24] and SimBoost [25] for a fair comparison.

6.4. Evaluation

Gabel and co-workers define two evaluation criteria that machine-learning based scoring models should pass: scoring power test and docking power test [174]. Scoring power test measures the performance based on the correlation between predicted and experimental binding affinity values. Existing studies mostly report scoring power test by using Pearson correlation coefficient, the Spearman correlation coefficient, and the root-mean-square error as evaluation metrics. The evaluation metrics that we adopted for the proposed approaches are summarized.

Concordance Index (CI) is adopted to measure the performance of a regression model in binding affinity prediction task [24]. To evaluate the performance of a model that outputs continuous values, Concordance Index (CI) can be used [178]:

$$CI = \frac{1}{Z} \sum_{\delta_i > \delta_j} h(f_i - f_j) \tag{6.2}$$

where f_i is the prediction value for the larger affinity δ_i, f_j is the prediction value for the smaller affinity δ_j, Z is a normalization constant, $h(m)$ is the step function [24]:

$$h(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0.5, & \text{if } x = 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{6.3}$$

r_m^2 **index** can be used to evaluate the external predictive performance of QSAR models where r_m^2 values greater than 0.5 for the test set was determined as an acceptable model (Equation 6.4) [179, 180].

$$r_m^2 = r^2 * (1 - \sqrt{r^2 - r_0^2}) \tag{6.4}$$

Mean Squared Error (MSE) is used as an evaluation metric and as a loss function in DeepDTA. MSE can be desribed as in Equation 6.5:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(P_i - Y_i)^2 \tag{6.5}$$

in which P is the prediction vector whereas Y corresponds to the vector of actual outputs.

The Area Under Precision Recall (AUPR) score AUPR metric is usually used to evaluate binary classification performance. AUPR is especially useful in cases where the training data is unbalanced. It is calculated based on the Precision (Equation 6.6) and Recall (Equation 6.7) values.

$$Precision = \frac{TP}{TP + FP} \tag{6.6}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.7}$$

in which TP, FP, and FN correspond to the number of True Positive, False Positive and False Negative predictions, respectively.

6.5. Baseline methods

The following two machine learning methods were employed as the baselines, in order to compare the performances of the proposed methodologies.

KronRLS algorithm was utilized by Pahikkala and co-workers (Section 3.1.7) in predicting binding strength of the proteins and ligands. [24]. Their model used protein and compound similarity matrices as kernel functions. The similarity of the proteins were computed using Smith-Waterman (S-W) algorithm and the compound similarities are computed via PubChem structural similarity.

SimBoost is a gradient boosting machine based method that depends on the features constructed from drugs, targets and drug-target pairs [25]. The proposed methodology uses feature engineering to build three types of features: (i) object-based features (for both drug and target) that utilize occurrence statistics and pairwise similarity information (SW and PubChem structure similarity), (ii) network-based features (for both drug and target) which were collected from two separate networks (each drug/protein represented as a node and connected to another node if above a user-defined similarity threshold) such as neighbor statistics, network metrics (betweenness, closeness etc.), PageRank [181] score etc., and (iii) network-based features which were collected from a heterogeneous network where either drugs or targets are nodes and connected to each other via binding affinity value. In addition to the network metrics, neighbor statistics and PageRank score, latent vectors from matrix factorization were also included.

These features were fed into a supervised learning method named gradient boosting regression trees [102,107] which was derived from gradient boosting machine model [105]. With gradient boosting regression trees, for a given drug-target pair dt_i, the binding affinity score is $\bar{y}$ predicted as follows [25]:

$$\bar{y_i} = \theta(dt_i) = \sum_{m=1}^{M} f_m(dt_i), f_m \in F \tag{6.8}$$

in which M denotes the number of regression trees and F represents the space of all possible trees.

6.6. ChemBoost: Chemical Language-based Drug-Target Binding Affinity Prediction

In this section, we will describe our first approach that employs XGBoost to solve the binding affinity prediction problem. The proposed system combines SMILESVec and SMILESVec-based protein representation while investigating the effect of chemical word design.

6.6.1. Chemical Word Design

SMILESVec is ligand representation technique that is built upon the Word2Vec algorithm which learns abstract features for words from a large corpus. The model is successful at capturing the semantic similarity between words that appear in similar contexts, since it considers the neighboring words of each word within a window frame during training. Thus, we hypothesize that the choice of chemical word extraction technique might have an important effect on describing ligands with SMILESVec.

6.6.1.1. k-mer. SMILESVec is originally built upon k-mer approach to determine the chemical words in which k is chosen as 8 [36]. More details are available in Chapter 4. 8-mers were used to train embeddings from the ChEMBL SMILES corpus.

6.6.1.2. Maximum Common Substructures (MCS). MCS were accepted as the words of the chemical language in recent studies [88,89]. The authors extracted a vocabulary of MCS from a large molecule corpus in which pairwise 2D representations are compared pairwise. These substructures were then converted into SMILES sub-sequences (i.e. words) via RDKit [182], a chemistry-specific development package.

The MCS vocabulary, which comprised 100K MCS (in SMILES form), was kindly provided to us by the authors of [89]. The SMILES sequences, however, were expressed in RDKit format. For instance an aromatic carbon “c” was represented as “C:” in the vocabulary. In order to prevent the confusion that might be caused because of this syntax, we used RDKit to convert these SMILES sub-sequences, first into a mol file, and then into SMILES which were compatible with ChEMBL canonical SMILES convention. We further filtered out the MCS that has at most one or two characters. These processes, resulted in vocabulary size reduced to around 68K SMILES words, combined with the failure of RDKit to perform conversion for some of the sub-sequences. The final vocabulary was used to extract words from the ChEMBL SMILES corpus and to train the Word2Vec algorithm.

6.6.1.3. Byte-Pair Encoding (BPE). BPE segmentation algorithm is used to extract words from the SMILES sequence in order to construct SMILESVec [85]. The algorithm is trained on ChEMBL SMILES corpus with the maximum vocabulary size of 20K, character coverage of 0.99, maximum word length of 100 characters. The SentencePiece library was utilized to train the model [183]. In order to parse the SMILES string efficiently, "number_split" and "unicode_split" features of the SentencePiece library were set to "False".

Table 6.2 provides examples of the words that are extracted with these word segmentation techniques from the SMILES of ampicillin, "CC1(C(N2C(S1)C(C2=O)NC(=O)C(C3=CC=CC=C3)N)C(=O)O)C".

Table 6.2: Example words extracted from the SMILES of *ampicillin* using different techniques.

Method	Words
k-mer (i.e. 8-mer)	CC1(C(N2, C1(C(N2C, 1(C(N2C(, ...,)C(=O)O), C(=O)O)C
MCS	CC=CC, C=C, C=CC=C, C=CC
BPE	CC1(, C(N2, C(S1), C(C2=O), NC(=O)C(, C3=CC=CC=C3,)N), C(=O)O)C

We also utilized DeepSMILES syntax for the representation of SMILES strings [52]. DeepSMILES hypothesizes that the proposed syntax will improve the performance of machine learning models that deal with SMILES strings with the help of the modifications on the branch and ring representations in the SMILES. Therefore, in order to investigate the effectiveness of the DeepSMILES syntax, we first converted our training data corpora into DeepSMILES. Then, we created 8-charactered sub-sequences from each DeepSMILES to train the Word2Vec algorithm. We will refer to DeepSMILES-based embeddings as DeepSMILESVec in the rest of the thesis.

6.6.2. SMILESVec-based Protein Representation

In order to represent proteins, we adopted a ligand-centric approach, where a protein is represented as a vector that is the average of the SMILESVec vectors of its interacting ligands [36]. Different from our recent work in [36], where all chemical words are considered to be equally important, proteins are represented using the ligands that they bind to with strong binding affinity values. For the Davis and BindingDB datasets, we selected the pK_d value of 7 as threshold to divide the ligands into strong-binding and weak-binding classes ($pK_d \geq 7$ strong binding) [25], whereas for KIBA dataset, KIBA value of 12.1 was chosen as threshold to choose between weak and strong binding ligands.

If a protein interacts with at least one ligand from the high-binding class, then that ligand is used to represent the protein. If not, the protein is represented with all of its interacting ligands.

$$protein = \begin{cases} \text{high affinity ligands,} & \text{if } pK_d \vee \text{KIBA} \geq \text{threshold} \\ \text{all interacting ligands,} & \text{otherwise} \end{cases}$$

Eventually, a protein is represented using its strong binding (SB) ligands.

6.6.3. Finding Important Chemical Words

Not every word in the chemical language has the same importance. For instance, some words might be the key players of the action of binding to a target. Since the protein is represented with the chemical words of its (high) binding ligands, it is be crucial to detect the important words which might provide insights about the binding mechanism of the protein. Thus, we propose to weight the chemical words using Inverse Document Frequency (IDF). The TF weight is implicitly considered in the protein representation because of the consideration of the same chemical word based on its occurence. IDF weight, on the other hand, assigns higher importance to the rare words in a corpus. It is described as in Equation 6.9:

$$IDF(cw, D) = log\frac{N}{|s \in S : cw \in s|} \tag{6.9}$$

where cw, S and N denote the chemical word, SMILES corpus, and number of SMILES in the corpus, respectively [184]. SMILES corpus in this work is designed as the protein universe that is constructed by the chemical words. In other words, a chemical word that appears in many proteins has a low IDF weight, whereas another that appears in rare proteins has a weight.

6.6.4. Experiment Settings

We evaluated the performance of the presented models on the benchmark datasets, KIBA [158] and BindingDB (BDB). XGBoost algorithm was utilized for prediction. The hyper-parameters C and γ were determined via five-fold cross validation. We chose the values for C and γ among $0.01, 1.0, 100.0$ and $0.1, 1.0, 10.0$, respectively. The parameter combination with which we obtained the best CI value on the training set was selected to model the test set. We performed statistical significance tests with paired t-test with the 95% confidence interval.

Proteins and ligands are both described with 100-dimensional real-valued word embeddings. Therefore, the input for the prediction model is a 200-dimensional vector, which is equal to the concatenation of the protein and ligand vectors for each protein-ligand pair.

6.6.5. Results

With this work, we introduce a novel drug - target binding affinity prediction method based only on SMILES string representation with which ligands and their target proteins are represented. We adopted the eXtreme Gradient Boosting (XGBoost) algorithm as the prediction algorithm and performed our experiments on the KIBA Kinase and BDB datasets. Proteins were represented with their strong affinity ligands and the results were compared with protein sequence based methodologies. We refer to the proposed prediction system in which both proteins and ligands are represented through chemical words as ChemBoost.

Tables 6.3 and 6.4 report the performance of the prediction systems on the respective test sets of KIBA and BDB, respectively, in terms of CI, MSE, and AUPR metrics. To compute AUPR, the threshold values are set to 12.1 and 7.0 for KIBA and BDB, respectively, to divide the datasets into positive (binding) and negative (non-binding) classes.

Table 6.3: CI and MSE values for KIBA dataset on the independent test set using XGBoost algorithm. ChEMBL canonical SMILES were used. Standard deviations are given in the parenthesis. (SB: refers to strong-binding ligands)

Method	Proteins	Compounds	CI	MSE	AUPR
KronRLS [24]	S-W	Pubchem Sim	0.782 (0.0009)	0.411	0.635 (0.004)
SimBoost [25]	S-W	Pubchem Sim	0.836 (0.001)	0.222	0.760 (0.003)
DeepDTA	CNN	CNN	**0.864** (0.003)	**0.197** (0.003)	**0.787** (0.007)
XGBoost (1)	S-W	Pubchem Sim	0.824 (0.0003)	0.248 (0.001)	0.735 (0.001)
XGBoost (2)	S-W	SMILESVec	0.837 (0.0004)	0.221 (0.001)	0.761 (0.0008)
XGBoost (3)	Protvec	SMILESVec	0.826 (0.0004)	0.245 (0.001)	0.735 (0.002)
ChemBoost (4)	ProtVec	SMILESVec (BPE)	0.828 (0.0004)	0.238 (0.001)	0.743 (0.002)
XGBoost (5)	ProtVec	SMILESVec (MCS)	0.716 (0.0005)	0.508 (0.001)	0.454 (0.003)
ChemBoost (6)	SMILESVec	SMILESVec	0.826 (0.0008)	0.249 (0.001)	0.728 (0.001)
ChemBoost (7)	SMILESVec (SB)	SMILESVec	0.838 (0.0006)	0.221 (0.0006)	0.770 (0.001)
ChemBoost (8)	SMILESVec (BPE, SB)	SMILESVec (BPE)	0.838 (0.0006)	0.220 (0.001)	0.773 (0.002)

In both datasets, SimBoost [25] performed better than KronRLS [24] in three of the evaluation metrics. This outcome might be expected since SimBoost relies on network-based features as well as the features KronRLS utilized, S-W and PubChem similarity scores. Model (1) reports the performance with XGBoost algorithm when only S-W and PubChem similarity scores are used as features, which as expected, performed worse than the SimBoost method (MSE values of 0.222 and 0.485 for KIBA and BDB, respectively), with MSE values of 0.248 for KIBA and 0.698 for BDB. However, when PubChem similarity scores were replaced with SMILESVec in ligand representation in Model (2), we observed an increase in the performance compared to Model (1). Furthermore, Model 2 achieved a similar performance to SimBoost in Davis dataset in terms of CI, whereas in BDB dataset Model (2) over-performed SimBoost in CI and MSE metrics.

Model (3) utilized ProtVec, a Word2Vec-based method that is built upon 3-mers, as the protein representation technique, unlike Model (2) which used S-W. The results indicated that, when combined with SMILESVec ligand representation, S-W (value of 0.221 in KIBA/ value of 0.437 in BDB) is better than ProtVec (value of 0.245 in KIBA/ value of 0.493 in BDB) in protein description in terms of MSE metric.

The effect of different chemical word identification methods in ligand representation was investigated by comparing the Models (3), (4), (5) which correspond to , 8-mer, BPE, and MCS, respectively. In both datasets, 8-mer based SMILESVec and BPE-based SMILESVec yielded to close performances whereas MCS-based SMILESVec provide the worst results. This might be due to, even though comprising a 68K vocabulary, MCS words being rare in the compounds of KIBA and BDB datasets. Furthermore, most compounds were represented with common words such as “NCCO”, “C=CC”, and “CCO” thus leading to a simplified and similar representations.

In all three metrics, Model (7) performed better than Model (6) in both datasets, indicating that using strong binding ligands instead of all interacting ligands improved the protein representation for binding affinity prediction task. Model (7) and Model (8) in which 8-mers and BPE segments were used as chemical words yielded to similar performances in BDB and in KIBA, except for AUPR value in which BPE-based system was better than 8-mers based system. For KIBA dataset, ChemBoost systems, when strong ligands are used in protein representation (Models (7) and (8)), were better than KronRLS and almost as well as SimBoost, both of which utilized protein sequence information. For BDB dataset, Models (7) and (8) overperformed the KronRLS and SimBoost systems, especially in terms MSE.

Table 6.4: CI and MSE values for BindingDB (pK_d) dataset on the independent test set using XGBoost algorithm. ChEMBL canonical SMILES were used. Standard deviations are given in the parenthesis. (SB: refers to strong-binding ligands)

Method	Proteins	Compounds	CI	MSE	AUPR
KronRLS [24]	S-W	Pubchem Sim	0.814 (0.002)	0.939 (0.004)	0.709 (0.006)
SimBoost [25]	S-W	Pubchem Sim	0.853 (0.003)	0.485 (0.038)	0.827 (0.010)
DeepDTA [38]	CNN	CNN	0.873 (0.009)	**0.409** (0.046)	**0.836** (0.009)
XGBoost (1)	S-W	Pubchem Sim	0.818 (0.001)	0.698 (0.006)	0.699 (0.003)
XGBoost (2)	S-W	SMILESVec	**0.878** (0.001)	0.437 (0.002)	0.812 (0.003)
XGBoost (3)	Protvec	SMILESVec	0.866 (0.001)	0.493 (0.004)	0.787 (0.003)
XGBoost (4)	ProtVec	SMILESVec (MCS)	0.750 (0.002)	0.925 (0.008)	0.560 (0.005)
XGBoost (5)	ProtVec	SMILESVec (BPE)	0.863 (0.001)	0.508 (0.002)	0.784 (0.004)
ChemBoost (6)	SMILESVec	SMILESVec	0.854 (0.001)	0.503 (0.005)	0.803 (0.004)
ChemBoost (7)	SMILESVec (SB)	SMILESVec	0.867 (0.0009)	0.422 (0.003)	0.830 (0.002)
ChemBoost (8)	SMILESVec (BPE, SB)	SMILESVec (BPE)	0.866 (0.001)	0.425 (0.002)	0.830 (0.002)

We also investigated a new syntax, DeepSMILES [52], that modified the original SMILES syntax by updating branch and ring number syntax rules. When, ChemBoost, in which proteins are represented with strong binding ligands, was updated to utilize DeepSMILES instead of SMILES, in KIBA dataset there was a significant increase in the performance considering the MSE metric (from 0.422 to 0.213), whereas the CI (from 0.867 to 0.843) and AUPR (from 0.830 to 0.783) values dropped. In the BDB datasets, SMILES-based ChemBoost and DeepSMILES-based ChemBoost yielded similar values in both metrics. When investigated, we observed that the number of unique chemical words dropped when DeepSMILES was used instead of SMILES, from 21K to 14K in KIBA dataset, and from 10K to 8K in BDB. Therefore, the reason why the effect of DeepSMILES was more prominent in KIBA might be the number of unique words changing dramatically between two forms.

Figures 6.4 and 6.5 illustrate the distribution of chemical words identified with different techniques in 2D space for KIBA and BDB datasets, respectively. The 100D embeding of each word is mapped into 2D space via t-Distributed Stochastic Neighbor Embedding (t-SNE) [185] which is a dimensionality reduction technique. Python sklearn [186] implementation of t-SNE was utilized.

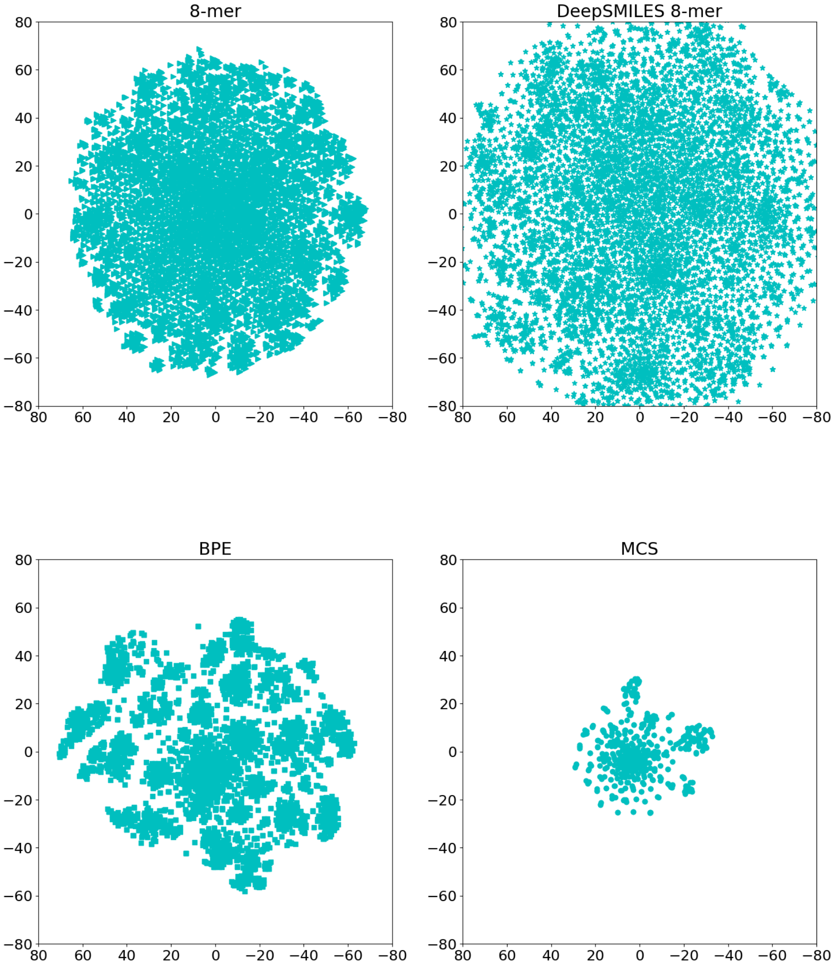


Figure 6.4: Representation of chemical words of KIBA dataset.

While 8-mers cover a larger space in both datasets, we can observe slight differences between SMILES-based 8-mers and DeepSMILES-based 8-mers in two datasets. While SMILES-based 8-mers look more compact than DeepSMILES-based 8-mers in KIBA, the distribution of the SMILES-, and DeepSMILES-based 8-mers look similar in BDB dataset. MCS words constitute a quite small area, whereas BPE words contain small clusters.

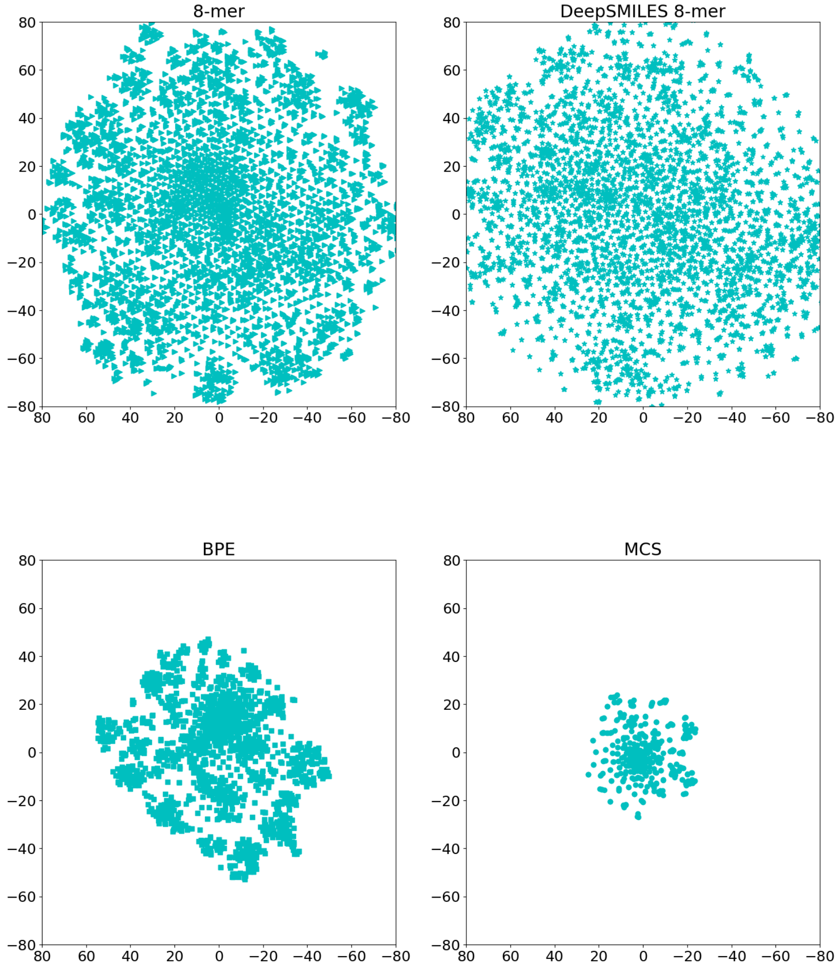


Figure 6.5: Representation of chemical words of BDB dataset.

In natural languages, not every word has the same importance for a given text. We hypothesized that the proteins that are represented by the chemical words of its interacting ligands, should also have some words that are important for that particular protein. For that, we used IDF weighting to find the rare words of the protein. We computed IDF values with two different approaches: global and local. The global

IDF of a word is computed based on its occurences in a large SMILES corpus (i.e. ChEMBL23), whereas the local IDF of each word is based on the number of proteins it appears. However, integration of either global or local IDF did not result in an improvement in the prediction performance for both datasets.

6.7. DeepDTA: Deep Drug-Target Binding Affinity Prediction

In this section, we will introduce the second approach that we proposed to model drug-target binding affinity. Unlike the previous model, DeepDTA depends on CNN networks to extract features from chemical and protein sequences implicitly.

6.7.1. CNN-based Prediction Module

In this work, we adopted a popular deep learning architecture, Convolutional Neural Network (CNN) as the prediction model. CNN is an architecture that contains one or more convolutional layers often followed by a pooling layer. A pooling layer down-samples the output of the previous layer and provides a way of generalization of the features that are learned by the filters. On top of the convolutional and pooling layers, the model is completed with one or more fully connected (FC) layers. The most powerful feature of CNN models is their ability to capture the local dependencies with the help of filters. Therefore, the number and size of the filters in a CNN directly affects the type of features the model learns from the input. It is often reported that as the number of filters increases, the model becomes better at recognizing patterns [101].

We proposed a CNN-based prediction model that comprises two separate CNN blocks, each of which aims to learn representations from SMILES strings and protein sequences. For each CNN block, we used three consecutive 1D-convolutional layers with increasing number of filters. The second layer had double and the third convolutional layer had triple the number of filters in the first one. The convolutional layers were then followed by the max-pooling layer. The final features of the max-pooling layers were concatenated and fed into three FC layers, which we named as DeepDTA. We used 1024 nodes in the first two FC layers, each followed by a dropout layer of rate

0.1. The third layer consisted of 512 nodes and was followed by the output layer. The proposed model that combines two CNN blocks is illustrated in Figure 6.6.

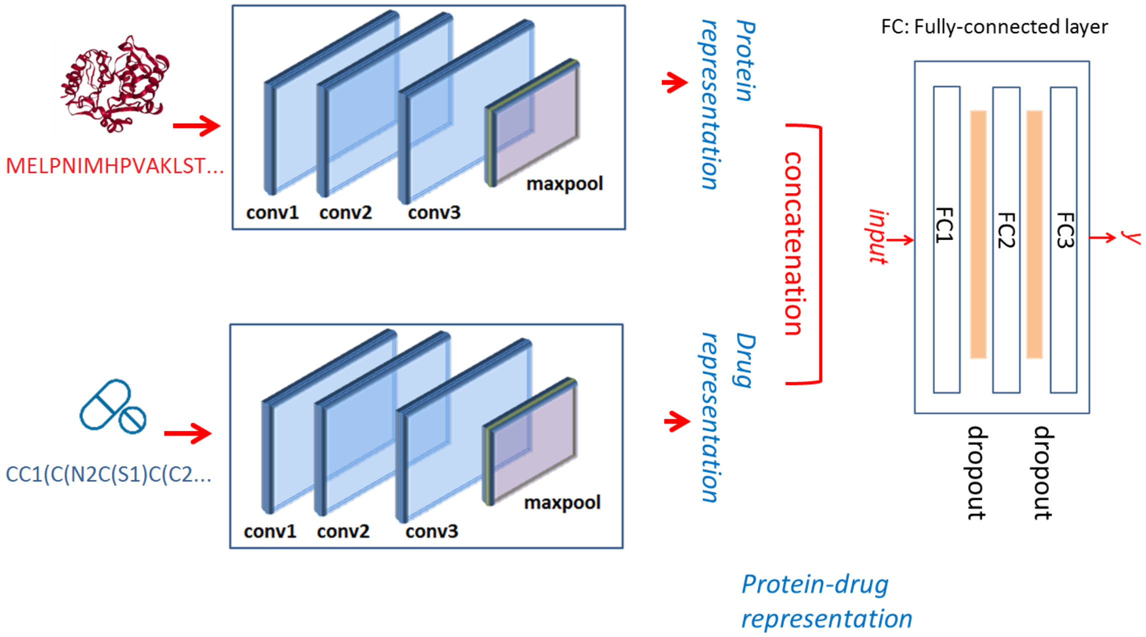


Figure 6.6: DeepDTA pipeline.

As activation function, we used Rectified Linear Unit (ReLU) [98], $g(x) = max(0, x)$, which has been widely used in deep learning studies [99]. A learning model tries to minimize the difference between the expected (real) value and the prediction during training. Since we work on a regression task, we used mean squared error (MSE) as the loss function (Equation 6.5). The learning was completed with 100 epochs and mini-batch size of 256 was used to update the weights of the network. Adam was used as the optimization algorithm to train the networks [187] with the default learning rate of 0.001. We used Keras [188] and the available embedding layer to represent characters with 128-dimensional dense vectors. The input for Davis dataset consisted of (85, 128) and (1200, 128) dimensional matrices for the compounds and proteins, respectively. We represented KIBA dataset with a (100, 128) dimensional matrix for the compounds and a (1000, 128) dimensional matrix for the proteins.

6.7.1.1. Representation of Proteins and Ligands. We used integer/label encoding which uses integers for the categories (label/integer encoding) to represent inputs. We scanned through approximately 2M SMILES sequences that we collected from PubChem and compiled 64 labels (unique letters). For protein sequences, we scanned 550K protein sequences from UniProt and 25 categories (unique letters) were extracted.

Here we simply represent each label with a corresponding integer (e.g. “C”:1, “H”:2, “N”:3 etc.). Label encoding for the example SMILES, “CN=C=O”, is given below.

$$\begin{bmatrix} C & N & = & C & = & O \end{bmatrix} = \begin{bmatrix} 1 & 3 & 63 & 1 & 63 & 5 \end{bmatrix}$$

Similar to the SMILES, protein sequences are encoded in the same fashion using label encodings. Both SMILES and protein sequences have varying lengths. Hence, in order to create an effective representation form, we decided on fixed maximum lengths of 85 for SMILES and 1200 for protein sequences. We chose these maximum lengths based on the distributions of the sequence lengths (illustrated in Figure 6.2) so that the maximum lengths cover most of the dataset. The sequences that are longer than the maximum length are truncated, whereas shorter sequences are 0-padded.

Both SMILES and protein sequences have varying lengths. Hence, in order to create an effective representation form, we fixed the length of SMILES and protein sequences to 85 and 1200, respectively. The sequences that are longer than maximum length are truncated whereas shorter sequences are 0-padded.

6.7.2. Experiment Settings

We decided on three hyper-parameters for our model, the number of the filters (same for proteins and compounds), the length of the filter size for compounds, and the length of the filter size for proteins. We chose to experiment with different filter lengths for compounds and proteins instead of a common one, due to the fact that they have different alphabets in terms of characters. The hyper-parameter combination that provided the best average CI score over the five-folds was chosen as the best combination in order to model the test set. We first experimented with hyper-parameters chosen from a wide range and then fine-tuned the model. For example, to determine the number of filters we performed a search over [16, 32, 64, 128, 512]. As explained in the Proposed Model subsection, the second convolution layer was set to contain twice the number of filters of the first layer, and the third one was set to contain three times the number of filters of the first layer. 32 filters obtained the best results over the cross-validation experiments. Therefore, in the final model, each CNN block consisted of three 1D convolutions of 32, 64, 96 filters, respectively. For all test results reported in Table 6.6 we used the same structure summarized in Table 6.5 except for the lengths of the filters that were used for the compound CNN-block and protein CNN-block.

Table 6.5: Parameters setting for DTA model.

Parameters	**Range**
Number of filters	32*1; 32*2; 32*3
Filter length (compounds)	[4,5,6,8]
Filter length (proteins)	[4,6,8,12]
epoch	100
hidden neurons	1024; 1024; 512
batch size	256
dropout	0.1
optimizer	Adam
learning rate (lr)	0.001

In order to provide a more robust performance measure, we evaluated the performance over the independent test set, when the model was trained with the learned parameters in Table 6.5 on the five training sets that we used in five-fold cross validation (note that the validation sets were not used).

The final CI score was reported as the average of these five results. Keras with Tensorflow [189] back-end was used as development framework. Our experiments were run on OpenSuse 13.2 (3.50GHz Intel(R) Xeon(R) and GeForce GTX 1070 (8GB)). The work was accelerated by running on GPU with cuDNN [190].

6.7.3. Results

In this study, we propose a deep-learning model that uses two CNN-blocks to learn representations for drugs and targets based on their sequences. As a baseline for comparison, the KronRLS algorithm and SimBoost methods that use similarity matrices for proteins and compounds as input were used. The Smith-Waterman (S-W) and PubChem Sim algorithms were used to compute the pairwise similarities for the proteins and ligands, respectively. We then used these S-W and PubChem Sim similarity scores as inputs to the FC part of our model (DeepDTA) to evaluate the model. Finally, we used three alternative combinations in learning the hidden patterns of the data and used this information as input to our DeepDTA model. The combinations were (*i*) learning only compound representation with a CNN block and using S-W similarity as protein representation , (*ii*) learning only protein sequence representation with a CNN block and using PubChem Sim to describe compounds, and (*iii*) learning both protein representation and compound representations with a CNN block. We call the last combination used with DeepDTA the combined model.

Tables 6.6 and 6.7 report the average MSE and CI scores over the independent test set of the five models trained with the same parameters (shown in Table 6.5) using the five different training sets for Davis and KIBA datasets.

Table 6.6: The average CI and MSE scores of the test set trained on five different training sets for the Davis dataset. The standard deviations are given in parenthesis.

	Proteins	Compounds	CI (std)	MSE
KronRLS [24]	Smith-Waterman	PubChem Sim	0.871 (0.0008)	0.379
SimBoost [25]	Smith-Waterman	PubChem Sim	0.872 (0.002)	0.282
DeepDTA	Smith-Waterman	PubChem Sim	0.790 (0.009)	0.608
DeepDTA	CNN	PubChem Sim	0.835 (0.005)	0.419
DeepDTA	Smith-Waterman	CNN	**0.886** (0.008)	0.420
DeepDTA	CNN	CNN	0.878 (0.004)	**0.261**

In the Davis dataset, SimBoost and KronRLS methods perform similarly while the CI values for SimBoost is higher than that for KronRLS in the larger KIBA dataset. When the similarity measures S-W, for proteins, and PubChem Sim, for compounds, are used with the the fully-connected part of the neural networks (DeepDTA), the CI drops to 0.79 for the Davis dataset and to 0.71 for the KIBA dataset. The MSE increases to more than 0.5. These results suggest that the use of a feed-forward neural network with predefined features is not sufficient to describe drug target interactions and to predict drug target affinities. Therefore, we used CNN layers to learn representations of drugs and proteins to capture hidden patterns in the datasets.

Table 6.7: The average CI and MSE scores of the test set trained on five different training sets for the KIBA dataset. The standard deviations are given in parenthesis.

	Proteins	Compounds	CI (std)	MSE
KronRLS [24]	Smith-Waterman	PubChem Sim	0.782 (0.0009)	0.411
SimBoost [25]	Smith-Waterman	PubChem Sim	0.836 (0.001)	0.222
DeepDTA	Smith-Waterman	PubChem Sim	0.710 (0.002)	0.502
DeepDTA	CNN	PubChem Sim	0.718 (0.004)	0.571
DeepDTA	Smith-Waterman	CNN	0.854 (0.001)	0.204
DeepDTA	CNN	CNN	**0.863** (0.002)	**0.194**

We first used CNN to learn representations of proteins and used the predefined PubChem Sim scores for the ligands. Using this combination did not improve the results suggesting that use of a CNN architecture is not effective enough to learn from amino acid sequences.

Then we used the CNN block to learn compound representations from SMILES and used the predefined S-W scores for the proteins. This combination outperformed the baselines on the KIBA dataset with statistical significance (p-value of 0.0001 for both SimBoost and KronRLS), and on the Davis dataset (p-value of around 0.03 for both SimBoost and KronRLS). These results suggested that the CNN is able to capture more information than PubChem Sim in the compound representation task.

Motivated by this result, we tested the combined CNN model in which both protein and compound representations are learned from the CNN layer. This method performed as well as the baseline methods with CI score of 0.878 on the Davis dataset and achieved the best CI score (0.863) on the KIBA dataset with statistical significance over both baselines (p-value of 0.0001 for both). The MSE values of this model were also notably lower than the MSE of the baseline models on both datasets. Even though learning protein representations with CNN was not effective, combination of the two CNN blocks for proteins and ligands provided a strong model.

Table 6.8: The average CI and MSE scores of the test set trained on five different training sets for the BDB dataset. The standard deviations are given in parenthesis.

	Proteins	Compounds	CI (std)	MSE
KronRLS [24]	Smith-Waterman	PubChem Sim	0.814 (0.002)	0.939
SimBoost [25]	Smith-Waterman	PubChem Sim	0.853 (0.003)	0.485
DeepDTA	CNN	CNN	**0.873** (0.009)	**0.409**
XGBoost	CNN	CNN	0.854 (0.002)	0.539

Table 6.8 reports the performance of DeepDTA on BDB dataset in which the best performance was obtained with DeepDTA. However, when the combined protein-ligand features (192-dimensional) that are learned through the separate CNN blocks were fed into the XGBoost algorithm, instead of the using the FFNN, the performance decreased dramatically in both datasets. This might be due to FFNN module being simply better than XGBoost algorithm at interpreting the CNN-based features.

In an effort to provide a better assessment of our model, we measured the performances of DeepDTA with two CNN modules and two baseline methods with two different metrics as well, namely r_m^2 index and The Area Under Precision Recall (AUPR) score. AUPR is adopted by many studies that utilize binary prediction. In order to measure AUPR based performances, we converted the quantitative datasets into binary datasets by selecting binding affinity thresholds. For Davis dataset we used pK_d value of 7 as threshold (pK_d $\geq$7 binds) similar to [25]. For KIBA dataset we used the suggested threshold KIBA value of 12.1 [25, 158]. Tables 6.9 and 6.10 depict the performances of DeepDTA with two CNN modules and two baseline methods on Davis and KIBA datasets, respectively.

Table 6.9: The average r_m^2 and AUPR scores of the test set trained on five different training sets for the Davis data set. The standard deviations are given in parenthesis.

	Proteins	Compounds	r_m^2 (std)	AUPR (std)
KronRLS [24]	Smith-Waterman	PubChem Sim	0.407 (0.005)	0.661 (0.010)
SimBoost [25]	Smith-Waterman	PubChem Sim	**0.644** (0.006)	0.709 (0.008)
DeepDTA	CNN	CNN	0.630 (0.017)	**0.714** (0.010)

The results suggest that both SimBoost and DeepDTA are acceptable models for affinity prediction in terms of r_m^2 value and DeepDTA performs significantly better than SimBoost in KIBA dataset in terms of r_m^2 (p-value of 0.0001) and AUPR performances (p-value of 0.0003).

Table 6.10: The average r_m^2 and AUPR scores of the test set trained on five different training sets for the KIBA data set. The standard deviations are given in parenthesis.

	Proteins	Compounds	r_m^2 (std)	AUPR (std)
KronRLS [24]	Smith-Waterman	PubChem Sim	0.342 (0.001)	0.635 (0.004)
SimBoost [25]	Smith-Waterman	PubChem Sim	0.629 (0.007)	0.760 (0.003)
DeepDTA	CNN	CNN	**0.673** (0.009)	**0.788** (0.004)

Figure 6.7 illustrates the predicted against measured (actual) binding affinity values for Davis and KIBA datasets. A perfect model is expected to provide a $p = y$ line where predictions (p) are equal to the measured (y) values. We observe that especially for KIBA dataset, the density is high around the $p = y$ line.

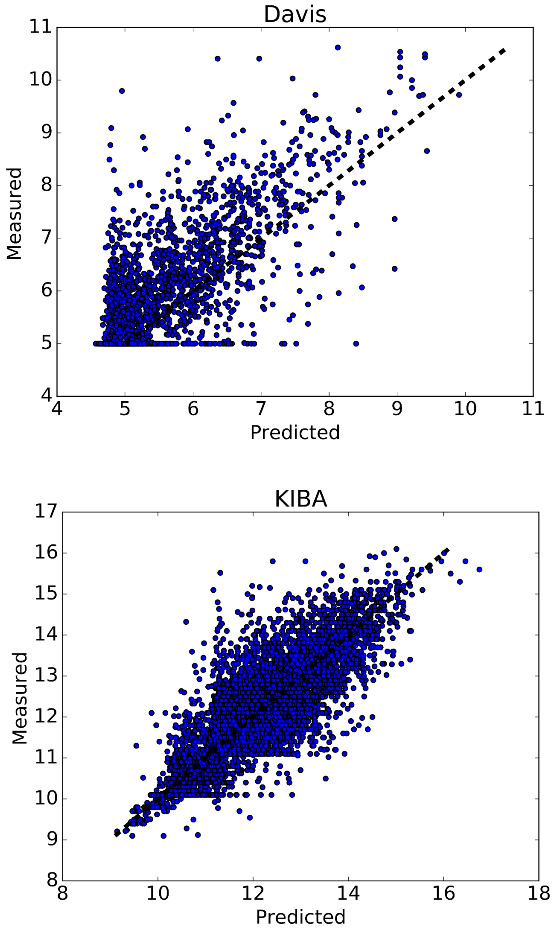


Figure 6.7: Predictions from DeepDTA model with two CNN blocks against measured (real) binding affinity values for Davis (pK_d) and KIBA (KIBA score) datasets.

6.8. Conclusion

In this chapter, we introduced two works that we propose to solve drug-target binding affinity prediction problem. In the first work, we adopted a machine learning based approach in which XGBoost was used as the prediction algorithm. The novelty of this method was its use of only SMILES strings to predict the strength of the binding. Furthermore, different chemical word identification techniques and their effects on the prediction performance were investigated.

In the second work, DeepDTA, we utilized a deep-learning based methodology for prediction. In this model, two symmetric CNN modules were used to extract features from the sequences of proteins and compounds. Therefore, instead of explicitly defining chemical words, CNN modules extracted the features from the raw sequences.

The results indicated that DeepDTA provided the best performance on three of the interaction datasets compared to state-of-the-art machine learning systems, KronRLS [24] and SimBoost [25]. However, when features that are extracted via CNN modules were fed into XGBoost algorithm, the performance of the DeepDTA dropped considerably. This might be indicator of artificial neural network as the prediction system was better at explaining the features that are extracted by CNN than XGBoost algorithm.

The ligand-centric prediction system, ChemBoost, yielded to better evaluation values than KronRLS and either better or close performances to SimBoost algorithm, which utilizes protein sequence and network-based interaction statistics for prediction. We were able to predict drug-target binding affinity using only SMILES strings without using any protein sequence or structure information. As expected, using only the high affinity ligands in the protein representation provides a significantly better performance than using all available or tested ligands. Furthermore, the investigation of different word identifaction techniques showed that 8-mers and BPE-based segments provides similar performances, whereas MCS words cover much smaller chemical space, and fails to provide an efficient ligand representation.

The power of the ligand based representation lies in its ability to describe functional properties of a protein. A limitation of our approach is that it is only available for datasets that have proteins with at least one ligand interaction. On the other hand, structure based prediction tools are limited by the small number of protein - drug complex structures. Our results suggest that adding our ligand centric approach to approaches that utilize orthogonal pieces of information such as 3D structure of the complex, or binding site residues on the protein can provide significant depth to our understanding of the mechanism of protein - drug recognition.

7. TOOLS

In this chapter, the products of the approaches that are discussed in Chapters 4, 5 and 6 are introduced.

7.1. SMILESVec

SMILESVec is a ligand representation methodology that is built upon learning embeddings for the chemical words [36]. The source code of SMILESVec is written in Python language [191]. The users can create SMILESVec representations for their compounds or SMILESVec-based protein representations for their proteins. Figure 7.1A illustrates the pipelines for constructing ligand and ligand-based protein embeddings.

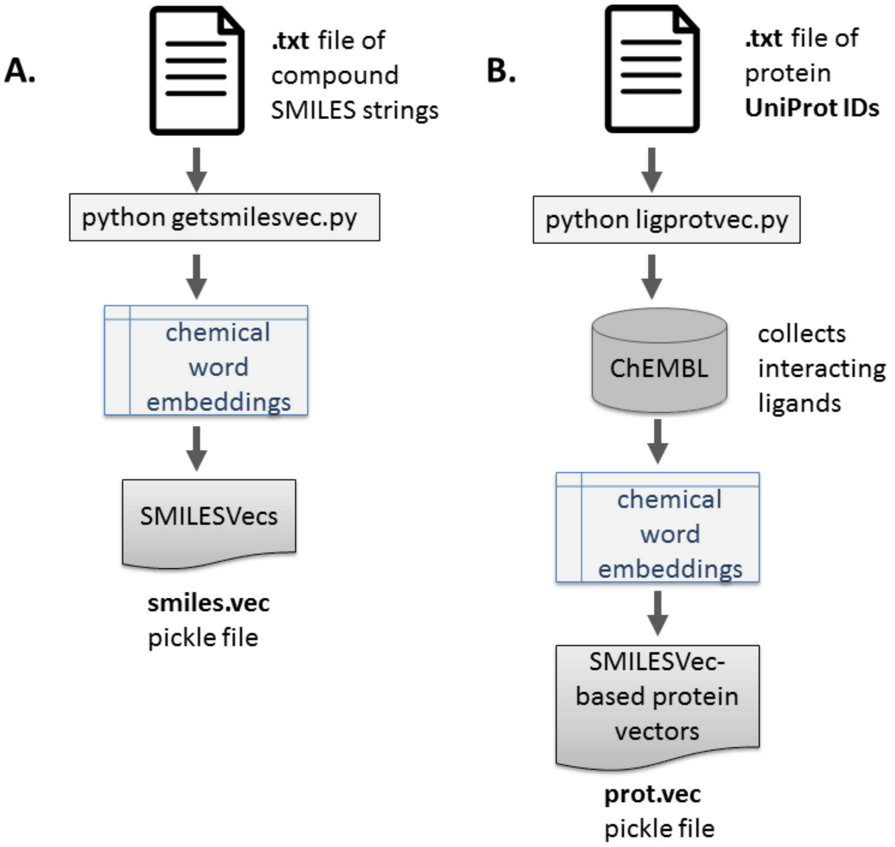


Figure 7.1: (A) Creating SMILESVec for a given list of SMILES strings. (B) Creating SMILESVec-based protein vectors for a given list of UniProt IDs.

Both pipelines require a pre-trained "chemical" word (i.e. 8-mers) embedding file in order to compute ligand and protein vectors. These pre-trained embeddings are available online as well:

- word-level (8-mer) embeddings trained on ChEMBL23 available at [192].
- word-level (8-mer) embeddings trained on PubChem available at [193].
- char-level (1-mer) embeddings trained on ChEMBL23 available at [194].
- char-level (1-mer) embeddings trained on PubChem available at [195].

The user can train their own embeddings on a different corpus with different parameters. The instructions are available at [196].

7.1.1. Requirements

The following environments/tools should be installed in order to run the SMILESVec package:

- Python 2.7.x or Python 3.x
- numpy
- sklearn
- chembl_webresource_client
- gevent==1.2.2
- greenlet==0.4.12
- pickle

7.1.2. Usage

get SMILESVec for a given SMILES set. For a list of SMILES strings, it outputs the corresponding SMILESVec for each SMILES in a pickle file named "smiles.vec". The following code runs for "smiles_sample.txt" file under *utils* folder. The user needs to either modify this file or name their input ".txt" file with the same name.

```
python getsmilesvec.py [embedding_file_name]
```

get SMILESVec-based representation for a given protein UniProt ID . To build a SMILESVec-based protein vectors, the user shpuld provide a list of UniProt IDs of the desired proteins. The script first converts UniProt IDs to ChEMBL targets, and searches for interacting ligands of each protein. The output is a pickle file ("prot.vec") which contains corresponding SMILESVec-based protein vectors for each UniProt ID, if they have at least one interacting ligand. Otherwise the output is "None". The following code runs for "prots_sample.txt" file under *utils* folder. The user needs to either modify this file or name their input ".txt" file with the same name.

```
python getligprotvec.py [embedding_file_name]
```

7.2. DeepDTA

DeepDTA is a CNN-based deep learning approach to predict drug-target binding affinities through text-based representations [38]. The source code is written in Python language. DeepDTA is publicly available at [197] as a Python package. The performance of the DeepDTA is reported on Davis [159] and KIBA [158] datasets in the original article [38]. The user can either reproduce these results or employ DeepDTA to predict the binding affinities of their own data.

7.2.1. Requirements

DeepDTA is dependent on the following tools:

- Python 3.X
- Keras 2.x
- Tensorflow 1.x
- numpy

- matplotlib

7.2.2. Reproducing the Original Results

If the user simply wants to reproduce the original results with Davis and Kiba datasets, the following script can be executed. seq_window_lengths and smi_window_lengths arguments can take a list of kernel sizes, if users want to experiment with.

```
python run_experiments.py
–num_windows 32
–seq_window_lengths 8
–smi_window_lengths 4
–batch_size 256
–num_epoch 100
–max_seq_len 1000
–max_smi_len 100
–dataset_path “data/kiba/”
–isLog 0
–log_dir “logs/”
```

7.2.3. Use of New Datasets

The use of new train/test datasets with DeepDTA model is explained at [198].

Training set

- DTC is a a subset of data collected from [199] that only contains p_{Kd} binding affinity values. DTC can be used as a training set, if the user wants to train a model based on $p_K d$ binding affinity values. The user should download the pickle file, Y, which stores binding affinity values from the address [200], and place “Y” under *DTC folder* in order to utilize DTC as a training set.

- The user also can use his/her own training set. In that case, line 552 in "run_experiments.py" file should be un-commented. Otherwise, the script assumes that DTC will be used as a training set.

```
prepare_new_data(FLAGS.test_path, test=False)
```

Test set. The script assumes that the test set is provided by the users. The set contains three files that store information for ligands, proteins, and binding affinity, respectively. Binding affinty file, Y, can either be empty or filled with real interaction strength values.

ligands.tab: each line contains tab-seperated ligand ID and corresponding SMILES of the ligands in the dataset.

proteins.fasta: FASTA inputs for each protein in the dataset.

Y.tab: tab-separated binding affinity file (drugs x proteins matrix). The number of rows corresponds to the number of drugs and the number of columns is equal to the number of proteins. This can be all 0s if one wants to predict binding affinity values for the unknown data. Or you can simply use the known affinity values for each drug-protein pair in which unknown interactions are indicated as "*nan*".

Example Y for predicting unknown protein-drug interactions

$$\begin{bmatrix} 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & . & 0 \\ . & . & . & . & . \\ 0 & 0 & 0 & . & 0 \end{bmatrix}$$

Example Y for predicting known + unknown protein-drug interactions

$$\begin{bmatrix} 8.1 & 2 & 12 & nan & 15 & . & 50 \\ 4 & 4.3 & 5 & 14 & nan & . & nan \\ . & . & . & . & . & . & . \\ nan & 2.2 & 5 & 8 & 12 & . & 0.2 \end{bmatrix}$$

The script below provides an example of how a user can run the script for training/testing their own data.

```
python run_experiments.py
–num_windows 32
–seq_window_lengths 8
–smi_window_lengths 4
–batch_size 256
–num_epoch 100
–max_seq_len 1000
–max_smi_len 100
–train_path "data/DTC/"
–test_path "data/mytest/"
–isLog 0
–log_dir "logs/"
```

7.2.4. Arguments

The arguments of DeepDTA Python scripts are explained as follows:

- *–num_windows* indicates the number of kernels in CNN blocks for proteins and ligands.
- *–seq_window_lengths* indicates the size of kernels in CNN block for proteins.
- *–smi_window_lengths* indicates the size of kernels in CNN block for ligands.

- *–batch_size* the number of training samples considered per parameter update.
- *–num_epoch* defines the number of times the model observes all training data.
- *–max_seq_len* is the length of maximum protein sequence that is allowed. The sequences longer than this value are cut, and shorter than this value are padded.
- *–max_smi_len* is the length of maximum ligand SMILES that is allowed.
- *–log_dir* is the directory where the training and test results are saved.
- *–isLog* is a flag to indicate whether Y values will be transformed into log form. 0 indicates negative, whereas 1 indicates positive.
- *–test_path* is the argument to use when a user wants to use their own test set.
- *–train_path* is the argument to use when a user wants to use their own training set.
- *–dataset_path* is the argument that sets the datasets path when user wants to reproduce the original results.

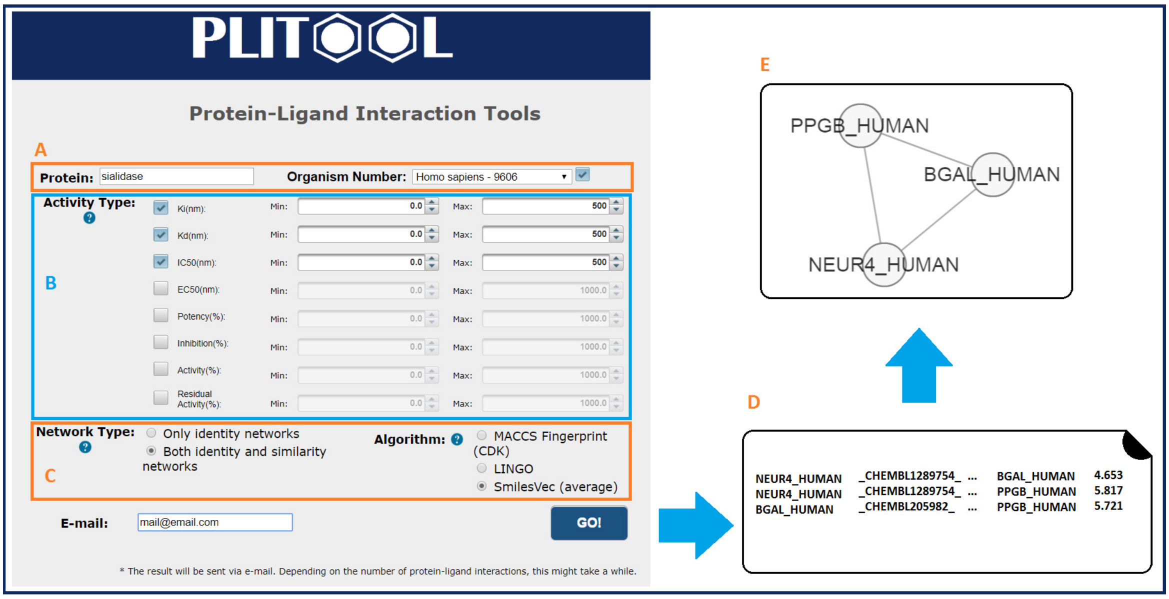
7.3. PLITOOL

The ligand representation method SMILESVec is actively used in a web tool, Protein-Ligand Interaction Tool (PLITOOL) which enables its users to collect protein-ligand interactions from databases and build networks.

PLITOOL focuses on providing a ligand-centric perspective to PPIs instead of aiming to cover all PPIs. It combines two useful functionalities: (i) automatic collection of P-L (Protein-Ligand) interaction information from UniProt [69] and ChEMBL, and (ii) network representation of protein-protein interactions (PPIs) based on the ligand-centric network models (LCN) that we previously proposed [35]. The identity based LCN connect a pair of proteins if they have a ligand in common. In the similarity based LCN, PPI network is built based on the pairwise chemical similarity of the ligands that proteins bind to. PLITOOL presents a user friendly environment to collect P-L interaction data of a given protein or a protein family and allows the construction of LCN with an option to use SMILESVec [36], which is a high dimensional real-valued vector representation, for compound similarity calculations. Stand-alone versions of both collecting P-L interactions and LCN are available under the respective web-pages in the PLITOOL web-server.

7.3.1. PLITOOL Features

PLITOOL is a combined online platform to collect P-L interaction information and to perform and visualize PPIs on a ligand-centric network (LCN) model [201].



PLITOOL

Protein-Ligand Interaction Tools

Protein: sialidase　Organism Number: Homo sapiens - 9606 ☑

Activity Type:

☑ Ki(nm):	Min:	0.0	Max:	500
☑ Kd(nm):	Min:	0.0	Max:	500
☑ IC50(nm):	Min:	0.0	Max:	500
□ EC50(nm):	Min:	0.0	Max:	1000.0
□ Potency(%):	Min:	0.0	Max:	1000.0
□ Inhibition(%):	Min:	0.0	Max:	1000.0
□ Activity(%):	Min:	0.0	Max:	1000.0
□ Residual Activity(%):	Min:	0.0	Max:	1000.0

Network Type: ○ Only identity networks ● Both identity and similarity networks

Algorithm: ○ MACCS Fingerprint (CDK) ○ LINGO ● SmilesVec (average)

E-mail: mail@email.com

* The result will be sent via e-mail. Depending on the number of protein-ligand interactions, this might take a while.

```
NEUR4_HUMAN   _CHEMBL1289754_  ...   BGAL_HUMAN   4.653
NEUR4_HUMAN   _CHEMBL1289754_  ...   PPGB_HUMAN   5.817
BGAL_HUMAN    _CHEMBL205982_   ...   PPGB_HUMAN   5.721
```

Figure 7.2: PLITOOL main screen. A) Protein-related queries and organism (optional). B)The activity types and ranges. C) Network types and similarity calculation method if similarity network is chosen. D) Out file contains interactions between proteins based on the similarity of their interacting ligands. E) Visualization of the network given in (D).

Figure 7.2 illustrates main user input screen of PLITOOL. PLITOOL allows its users to collect ligand-centric information by using any query allowed in UniProt such as protein/protein-family names, UniProt identifiers (IDs), Gene Ontology identifiers [202] or Enzyme Classification numbers for their choice of organism which is also optional (Figure 7.2A). Then, users are expected to choose at least one type of activity such as K_i, K_d, IC_{50} etc. and a range of values to obtain P-L interactions (Figure 7.2B). Once the interaction list is extracted, a LCN is built either based on identity of shared ligands or their similarity. If a user decides to obtain results for similarity LCN, they are expected to choose their preferred algorithm to compute ligand similarity (Figure

7.2C). Ligand similarity can be calculated by MACCS [203], LINGO (a sub-sequence based similarity method) [142] and SMILESVec [36] methods.

PLITOOL collects the corresponding P-L interaction information, which is formatted as | Protein Name $< tab >$ Ligand ChEMBL Id $< tab >$ Ligand SMILES |, by using BioServices python package [204] and ChEMBL web client [205] to connect the UniProt and ChEMBL databases, respectively. This interaction information format is then used to build identity and similarity LCN [35]. The output of the LCN are protein interaction networks formatted as | Protein Name $< tab >$ Interaction Information $< tab >$ Protein Name $< tab >$ Interaction weight | (Figure 7.2D). This output format is compatible with network visualization tools such as Cytoscape [206]. Figure 7.2E provides the final network that is visualized in PLITOOL using Cytoscape interface. It displays the weighted similarity protein interaction LCN output of the inputs depicted in Figure 7.2. Depending on the number of interacting ligands, computation time of these methods can increase, especially for similarity LCN since it uses the pairwise similarity of the interacting ligands. Building the identity network takes 142 seconds for 8 proteins with 500 ligands while building the similarity network takes 180 seconds for the same set of P-L using SMILESVec as ligand similarity metric. Users can provide an e-mail address to which result files in zipped format will be forwarded.

7.3.2. Summary

PLITOOL is an online web interface for exploring protein-protein interactions with a ligand-centric approach. We provide ligand interactions for a protein of interest and build ligand-centric networks either by using ligand sharing or ligand similarity information and also enable the visualization of these networks either through our visualization interface powered by Cytoscape or other network visualization tools. PLITOOL is useful for any researcher that is interested in collecting P-L interaction data and to further investigate these interactions with LCN-based PPI models. With the help of network analysis tools, researchers can perform many different operations on these networks such as clustering or detecting important/hub proteins.

8. CONCLUSION

8.1. Summary of Contributions

This thesis had a main goal of understanding the interactions between proteins and compounds through their textual representations, and we investigated this goal under three perspectives: (i) representation of the compound, (ii) representation of the protein, and (iii) the prediction of the interaction strength between proteins and compounds. Chapters 4 and 5 describe the work we did to target the first two perspectives, and introduce two novel text-based representation approaches. Chapter 6 explains two different text-based methodologies that we proposed to address the binding affinity prediction task. The approaches introduced in this thesis are available online as Python tools which are explained in detail in Chapter 7.

Although proteins and compounds can be expressed in different forms such as 2D and 3D, the major advantage of the 1D representation (or the textual form) is that it is available for every molecule, unlike the 3D representation. Furthermore, even though comprising almost as much information as available in 2D, the simplistic nature of textual information makes it easier to process. This in turn makes processing of textual representations computationally less expensive compared to 2D and 3D representations. We hypothesize that, much like natural languages, sequences of proteins and compounds have their own alphabets and, most likely, grammars. Therefore, attempts to extract knowledge from these sequences is no different than assigning meaning to the text in a natural language. The linguistic properties of protein sequences [207] and compounds [156] have been investigated in early studies, reporting and underlying lingiustic patterns. The text-based systems that we introduced in this thesis achieved the state-of-the-art performances in the protein clustering and binding affinity prediction tasks. These results showed that textual forms of these bio-chemical entities indeed encode effective information to model solutions to bio/cheminformatics problems.

In Chapter 4 we introduced a text-based ligand representation method, that we named as SMILESVec. SMILESVec was inspired from the analogy between a document and the textual form of a chemical, SMILES. Much like words describing the content of a document, we suggested that a SMILES text might contain meaningful textual units (i.e. chemical words). Consequently, we extracted words from SMILES strings as k-mer which is one of the most widely adopted approaches in NLP to identify words. Our initial experiments suggested that 8-mer could be an alternative for a chemical word, which is in agreement with the findings of Wozniak and co-workers [89], who reported the length of the common substructures vary between 8-12. We also observed that 8-mers extracted from a large SMILES corpus follow the Zipf's Law, which is a power law to which most natural languages conform. We then proposed to utilize the Word2Vec algorithm, which learns distributed high level representations for the words when trained on a very large corpus. The representations of the words contain syntactic and semantic relatedness stemming from the concept in which Word2Vec considers the neighbor words within a given window range. Thus, we trained Word2Vec on a large SMILES corpus in which each SMILES is decomposed into its chemical words (i.e. 8-mers). We hypothesized that the representations that are learned through Word2Vec might help capture the relations between the chemical words. Finally, the chemical word representations were combined into a compound representation via averaging, building the SMILESVecs. To our knowledge, SMILESVec presented one of the first attempts at using the textual SMILES form directly to build a representation through words. SMILESVec achieved similar performances to knowledge-based fingerprints of compounds. We should add that a data-driven representation such as SMILESVec, has the flexibility of generating task specific representations rather than generating a universal ligand representation.

Protein sequence similarity is not always a good indicator of functional similarity. With the aim of capturing the functional and mechanistic properties of proteins, a novel approach to represent proteins using their interacting ligands is proposed. In Chapter 5 we proposed a novel methodology to represent proteins using their interacting ligands. Earlier studies reported a correlation between the architecture of the protein and the ligand [123], while much recent works in which interacting ligands are used to detect protein similarity also arrived at similar conclusions [34, 35, 125]. Motivated by these works, we proposed a ligand-centric protein representation in which proteins are described through their interacting ligands. Such approach requires the correct representation of ligands, and for that, we utilized SMILESVec. The proteins are represented with the SMILESVec vectors of their interacting ligands. This new protein representation method is evaluated for the task of protein clustering. The results show that, even though protein sequence information is not used, similar F-measure performance is obtained for this task compared to state-of-the-art sequence-based protein representation methods. Further analysis, revealed that ligand-based protein representation is able to cluster proteins from the same family even when they have low sequence similarity.

The performance of the SMILESVec-based protein representation was assessed on the protein family/super-family classification task using TransClust and MCL algorithms [36]. In super-family/family clustering of SCOPE A-50 dataset, the proposed representation (0.678/0.729) performed similarly to ProtVec (0.681/0.739), a state-of-the-art sequence based protein representation technique, and over-performed BLAST e-value (0.350/0.500) in terms of the F-measure metric with the TransClust algorithm. SMILESVec was compared to two widely-used fingerprints, ECFP6 and MACCS, as well, in which they yielded close F-measure scores. These results also indicated that regardless of the ligand representation methodology, ligand-centric protein representation might be a promising alternative to protein sequence based methods. It is also noteworthy that the ligand-centric approach was as well as sequence-based systems, in classifying a dataset that was created based on sequence/structure similarity information. In conclusion, we were able to define proteins based on their interacting ligands without requiring sequence or structure information.

We should, however, mention that ligand-centric nature of the proposed protein representation method limits the biological space of proteins to the ones that have at least one interacting ligand. We hypothesize that ligand-based protein representation might be especially useful in predicting possible ligands with which the protein can interact.

With this hypothesis in mind, we focused on the drug-target interaction prediction task in which the ligand and protein representation methods are combined. In the first half of Chapter 6, we introduced a chemical-language based prediction system which is solely built upon chemical words. For ligands, SMILESVec representation was used, while proteins were described through the SMILESVecs of their interacting ligands. The original protein representation [36] was updated such that proteins are represented with their high affinity ligands (strong binding, SB). As this work was based on chemical words, we investigated the application of different word identification techniques other than k-mers, which are Maximum Common Substructures (MCS) and Byte-Pair Encoding (BPE). MCSs were created based on the maximum common substructures that are extracted from the 2D representations a pair of molecules, and expressed as a SMILES sub-sequences. We used pre-constructed MCS vocabulary that was kindly provided by Wozniak and co-workers [89]. BPEs, on the other hand, were created based on the frequency of characters and sub-sequences. We created SMILESVecs with the chemical words designed as 8-mers, MCSs, and BPE-based segments. We evaluated the performance of chemical-word based prediction system on two datasets, KIBA and BDB, using the XGBoost algorithm.

The state-of-the-art machine learning methods with which we compared the proposed system were KronRLS [24] and SimBoost [25]. Both methods utilized S-W similarity score to represent proteins and PubChem 2D-based similarity for ligand description. SimBoost further integrated network-based statistics to the prediction system. We first compared the performance of SMILESVec-based ligand representation and PubChem using XGBoost algorithm. In both BDB and KIBA datasets, SMILESVec provided comparably better values in terms of CI, MSE, and AUPR metrics. We also observed that with SMILESVec as ligand representation method, S-W based protein representation and SMILESVec-based representation yielded close performance in KIBA and better MSE value in BDB datasets. In KIBA dataset, BPE-based words and 8-mers yielded to close performances in terms of MSE metric, out-performing the KronRLS baseline [24] and performing as well as SimBoost [25]. MCS-words based prediction system performed the worst. A similar trend was observed in BDB dataset as well. These results indicated that ligand-based protein representation was as strong as S-W similarity based protein representation. In terms of word identification methods, BPE and 8-mers were comparable whereas MCS words was not as informative as the other two. MCS words, although comprising a 68K vocabulary, were not that common in the corpus that we utilized to train Word2Vec. The embedding vectors were generated for only about 11K of them. For instance, the average number of MCS words that are extracted from the compounds of BDB dataset was 4.2, whereas they were 10.4 and 69.1 for BPE-based words and 8-mers, respectively. Thus, we can suggest that MCS were not rich enough to describe compounds, whereas BPE was able to to describe the compounds as well as 8-mers while using less number of words.

We also investigated a recent syntax, DeepSMILES [52], which is introduced to overcome the challenges the SMILES syntax brings with the branch and ring symbols. The prediction system that is built upon 8-mers extracted from DeepSMILES performed better (0.843/0.213/0.783) than SMILESVec (8-mer) (0.838/0.221/0.770) based system in three metrics, CI, MSE and AUPR, respectively, in KIBA dataset. In BDB dataset, however, DeepSMILES-based system performed similarly to SMILESVec-based system.

We should note that, integrating weights to the chemical words in protein representation via computing their Inverse Document Frequency (IDF) in both local and global scale did not result in an increase in the performance. In the local scale, we assigned the IDF value based on the number of proteins they are observed in, whereas in the global scale IDFs were computed based on the occurences of the chemical words in the SMILES corpus (1.7M). Another alternative might be to consider the number of chemical words in the strong binding ligands over the number of all ligands to be the IDF value.

In the second half of Chapter 6, we introduced a deep-learning based prediction system, DeepDTA [38], which, instead of extracting words explicitly, aims to learn implicit patterns via Convolutional Neural Networks (CNN) from the raw sequences of proteins and compounds. In DeepDTA, abstract features were extracted from protein sequences and SMILES through CNN blocks, which were then combined and fed into a feed forward neural network (FFNN) to perform binding affinity prediction. We compared the performance of the DeepDTA against the baseline state-of-the-art machine learning methods KronRLS and SimBoost, as well as, our proposed method that depends on chemical words to describe the protein-ligand interaction. In both datasets, KIBA and BDB, DeepDTA provided the best performance in terms of MSE and AUPR metrics.

We investigated whether the features that are extracted via CNN are the sole reason in the increased performance by using XGBoost as predictor instead of FFNN. Surprisingly, CNN blocks combined with XGBoost yielded a decrease in the performance, indicating that the prediction module is important to evaluate and to utilize the features. We should also note that proteins were difficult to represent through CNN due to the high signal-to-noise ratio (i.e. the ratio of informative sequences about the binding information were low compared to the length of full sequence). In Chapter 6, we concluded that even though deep-learning based methodologies were good at reaching state-of-the-art performances, both the feature extraction and prediction modules have complementing power on the prediction performance of the system. The integration of the word-based inputs might improve the performance of the whole system as reported in our preliminary work, WideDTA [208].

In this thesis, we focused on the processing of textual representations of chemical data in order to bring hidden knowledge to light. We presented two novel representations for proteins and compounds that can be used in any bio/cheminformatics task that involves the use of these components. These methodologies were purely text-based, which increase the availability of these systems unlike the systems that depend on 3D-information. We further introduced two protein-ligand binding affinity prediction systems that can be applied to any related dataset with available sequence information.

8.2. Future Work

In this thesis, we introduced four text-based systems to model interactions between compounds and ligands: (i) a novel ligand presentation, SMILESVec, (ii) a novel ligand-based protein representation, (iii) two systems to predict the binding affinity of protein-ligand interactions.

In Chapter 4, we proposed a distributed representation, SMILESVec, for ligands based on their SMILES text. SMILESVec was centered around the idea of extracting chemical words from the SMILES, which were then used to train Word2Vec algorithm. Even though creating representations based on the neighboring words of a target word, the Word2Vec algorithm might still not be enough to integrate the contextual information. For instance, a chemical word might act differently based on the chemical and the group of chemical words it appears with. In such cases, contextual embeddings such as ELMo [209] and BERT [210] might be quite useful. Unlike Word2Vec-based representations, contextual representations are not fixed, and have the ability to change considering the context the word appears in. The integration of contextual representations might improve the effectiveness of SMILESVec and SMILESVec-related tasks such as ligand-based protein representation that we introduced in Chapter 5.

The ligand-based protein representation yielded similar performance to state-of-the-art protein representation that is based on amino-acids sequences without the use of sequence/structure information. However, this representation is only available for proteins with at least one interacting ligand. In order to address this limitation, a future direction to follow might be to integrate the sequence/structural similarity information to predict ligand-based representation for those proteins. The use of sequence/structure information along with the ligand-based representation might be another revenue of future work. Another point is that the success of ligand-based systems depend on how well chemical words are described. In this thesis, we explored three different word identification techniques (i.e. k-mers, BPE, and MCS). k-mers and BPE-based words provided similar performances in ChemBoost, a protein-ligand binding affinity prediction system. However, a more intelligent hybrid system which combines a data-driven approach with expert knowledge might provide a better word identification strategy. This in turn, might directly affect the performances of systems that depend on chemical words such as SMILESVec and ChemBoost. We introduced two binding affinity prediction systems for protein-ligand interaction in Chapter 6. ChemBoost utilized only chemical words to design the interaction system, whereas DeepDTA utilized the whole SMILES and protein sequences instead explicitly identifying words. Most often, a prediction system is interpreted based on the prediction results. Deep-learning architectures are quite powerful when there is large enough data to learn from, and often comprises large number of layers and nodes to extract abstract features from the inputs. However, the complexity of these systems makes their explainability all the more challenging. With DeepDTA, we tried to assess how well CNN extracted features from protein and ligand sequences, and the results showed that protein sequences were more difficult to model compared to SMILES. These experiment-based tests provide only limited information about the cases for which the deep-learning system has difficulties to model the input. A recent work by Jacovi and co-workers [211] attempts to understand how CNNs work in text classification. The authors connect the maximum scores chosen by the max-pooling layer to the original n-grams. Integrating such system to DeepDTA might help us to detect important sub-sequences of proteins and SMILES that are highlighted by CNN modules.

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. Dudley, J. T., T. Deshpande and A. J. Butte, “Exploiting drug–disease relationships for computational drug repositioning”, *Briefings in Bioinformatics*, p. bbr013, 2011.

4. Reddy, A. S. and S. Zhang, “Polypharmacology: drug discovery for the future”, *Expert Review of Clinical Pharmacology*, Vol. 6, No. 1, pp. 41–47, 2013.

5. Wang, Y. and J. Zeng, “Predicting drug-target interactions using restricted Boltzmann machines”, *Bioinformatics*, Vol. 29, No. 13, pp. i126–i134, 2013.

6. Gaulton, A., L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani *et al.*, “ChEMBL: a large-scale bioactivity database for drug discovery”, *Nucleic Acids Research*, Vol. 40, No. D1, pp. D1100–D1107, 2012.

7. 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.

8. Wishart, D. S., C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang and J. Woolsey, “DrugBank: a comprehensive resource for in silico drug discovery and exploration”, *Nucleic Acids Research*, Vol. 34, No. suppl 1, pp. D668–D672, 2006.

9. Günther, S., M. Kuhn, M. Dunkel, M. Campillos, C. Senger, E. Petsalaki, J. Ahmed, E. G. Urdiales, A. Gewiess, L. J. Jensen *et al.*, "SuperTarget and Matador: resources for exploring drug-target relationships", *Nucleic Acids Research*, Vol. 36, No. suppl 1, pp. D919–D922, 2008.

10. Kuhn, M., C. von Mering, M. Campillos, L. J. Jensen and P. Bork, "STITCH: interaction networks of chemicals and proteins", *Nucleic Acids Research*, Vol. 36, No. suppl 1, pp. D684–D688, 2008.

11. Hattori, M., N. Tanaka, M. Kanehisa and S. Goto, "SIMCOMP/SUBCOMP: chemical structure search servers for network analyses", *Nucleic Acids Research*, Vol. 38, No. suppl_2, pp. W652–W656, 2010.

12. Smith, T. F. and M. S. Waterman, "Identification of common molecular subsequences", *Journal of Molecular Biology*, Vol. 147, No. 1, pp. 195–197, 1981.

13. 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.

14. Bleakley, K. and Y. Yamanishi, "Supervised prediction of drug–target interactions using bipartite local models", *Bioinformatics*, Vol. 25, No. 18, pp. 2397–2403, 2009.

15. van Laarhoven, T., S. B. Nabuurs and E. Marchiori, "Gaussian Interaction Profile Kernels for Predicting Drug–Target Interaction", *Bioinformatics*, 2011.

16. Gönen, M., "Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization", *Bioinformatics*, Vol. 28, No. 18, pp. 2304–2310, 2012.

17. Cao, D.-S., L.-X. Zhang, G.-S. Tan, Z. Xiang, W.-B. Zeng, Q.-S. Xu and A. F. Chen, "Computational Prediction of Drug- Target Interactions Using Chemical,

Biological, and Network Features", *Molecular Informatics*, Vol. 33, No. 10, pp. 669–681, 2014.

18. Cao, D.-S., S. Liu, Q.-S. Xu, H.-M. Lu, J.-H. Huang, Q.-N. Hu and Y.-Z. Liang, "Large-scale prediction of drug–target interactions using protein sequences and drug topological structures", *Analytica Chimica Acta*, Vol. 752, pp. 1–10, 2012.

19. Cobanoglu, M. C., C. Liu, F. Hu, Z. N. Oltvai and I. Bahar, "Predicting drug–target interactions using probabilistic matrix factorization", *Journal of Chemical Information and Modeling*, Vol. 53, No. 12, pp. 3399–3409, 2013.

20. Kitchen, D. B., H. Decornez, J. R. Furr and J. Bajorath, "Docking and scoring in virtual screening for drug discovery: methods and applications", *Nature reviews Drug Discovery*, Vol. 3, No. 11, p. 935, 2004.

21. Velec, H. F., H. Gohlke and G. Klebe, "DrugScoreCSD knowledge-based scoring function derived from small molecule crystal data with superior recognition rate of near-native ligand poses and better affinity prediction", *Journal of Medicinal Chemistry*, Vol. 48, No. 20, pp. 6296–6303, 2005.

22. Ballester, P. J. and J. B. Mitchell, "A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking", *Bioinformatics*, Vol. 26, No. 9, pp. 1169–1175, 2010.

23. Li, H., K.-S. Leung, M.-H. Wong and P. J. Ballester, "Low-quality structural and interaction data improves binding affinity prediction via random forest", *Molecules*, Vol. 20, No. 6, pp. 10947–10962, 2015.

24. 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.

25. 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.

26. Tian, K., M. Shao, S. Zhou and J. Guan, "Boosting compound-protein interaction prediction by deep learning", *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, pp. 29–34, IEEE, 2015.

27. Wang, C., J. Liu, F. Luo, Y. Tan, Z. Deng and Q.-N. Hu, "Pairwise input neural network for target-ligand interaction prediction", *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pp. 67–70, IEEE, 2014.

28. 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.

29. 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.

30. Zheng, L., J. Fan and Y. Mu, "OnionNet: a multiple-layer inter-molecular contact based convolutional neural network for protein-ligand binding affinity prediction", *arXiv preprint arXiv:1906.02418*, 2019.

31. 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, accessed on September 1, 2018.

32. Gaulton, A., L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani *et al.*, "ChEMBL: a large-scale bioactivity database for drug discovery", *Nucleic Acids Research*, Vol. 40, No. D1, pp. D1100–D1107, 2011.

33. 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. 1, 2016.

34. 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.

35. 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.

36. Öztürk, H., E. Ozkirimli and A. Özgür, “A novel methodology on distributed representations of proteins using their interacting ligands”, *Bioinformatics, accepted for publication*, 2018.

37. Öztürk, H., A. Özgür and E. Ozkirimli, “A chemical language based approach for protein-ligand interaction prediction”, *arXiv preprint arXiv:1811.00761*, 2018.

38. Öztürk, H., A. Özgür and E. Ozkirimli, “DeepDTA: deep drug–target binding affinity prediction”, *Bioinformatics*, Vol. 34, No. 17, pp. i821–i829, 2018.

39. 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, 2006.

40. Lo, Y.-C., S. E. Rensi, W. Torng and R. B. Altman, “Machine learning in chemoinformatics and drug discovery”, *Drug Discovery Today*, Vol. 23, No. 8, pp. 1538–1546, 2018.

41. Smith, T. J., “MolView”, `http://www.molview.org`, 1995, accessed in September 2019.

42. "IUPAC", https://iupac.org/, accessed in September 2019.

43. Heller, S. R., A. McNaught, I. Pletnev, S. Stein and D. Tchekhovskoi, "InChI, the IUPAC international chemical identifier", *Journal of Cheminformatics*, Vol. 7, No. 1, p. 23, 2015.

44. 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.

45. Weininger, D., A. Weininger and J. L. Weininger, "SMILES. 2. Algorithm for Generation of Unique SMILES Notation", *Journal of Chemical Information and Computer Sciences*, Vol. 29, No. 2, pp. 97–101, 1989.

46. Schwartz, J., M. Awale and J.-L. Reymond, "SMIfp (SMILES fingerprint) Chemical Space for Virtual Screening and Visualization of Large Databases of Organic Molecules", *Journal of Chemical Information and Modeling*, Vol. 53, No. 8, pp. 1979–1989, 2013.

47. Inc., D., "SMILES", https://daylight.com/dayhtml/doc/theory/theory.smiles.html, accessed in September 2019.

48. Nakoneczny, S. and M. Smieja, "Natural language processing methods in biological activity prediction", *This work was supported by ec under fp, Coordination and Support Action, Grant Agreement Number 316097, engine–European Research Centre of Net-work Intelligence for Innovation Enhancement.*, p. 25, 2016.

49. Cao, D.-S., J.-C. Zhao, Y.-N. Yang, C.-X. Zhao, J. Yan, S. Liu, Q.-N. Hu, Q.-S. Xu and Y.-Z. Liang, "In Silico Toxicity Prediction by Support Vector Machine and SMILES Representation-based String Kernel", *SAR and QSAR in Environmental Research*, Vol. 23, No. 1-2, pp. 141–153, 2012.

50. "OpenSMILES", https://opensmiles.org/opensmiles.html, accessed in

September 2019.

51. Inc., D., "SMARTS", https://daylight.com/dayhtml/doc/theory/theory.smarts.html, accessed in September 2019.

52. O'Boyle, N. and A. Dalke, "DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures", *ChemRxiv*, 2018.

53. nextmovesoftware, "DeepSMILES", https://github.com/nextmovesoftware/deepsmiles, accessed in September 2019.

54. Bolton, E. E., Y. Wang, P. A. Thiessen and S. H. Bryant, "PubChem: integrated platform of small molecules and biological activities", *Annual reports in computational chemistry*, Vol. 4, pp. 217–241, 2008.

55. "PubChem Structural Similarity", https://pubchem.ncbi.nlm.nih.gov/score_matrix/score_matrix.cgi, 2019, accessed in September 2019.

56. Sawada, R., M. Kotera and Y. Yamanishi, "Benchmarking a Wide Range of Chemical Descriptors for Drug-Target Interaction Prediction Using a Chemogenomic Approach", *Molecular Informatics*, Vol. 33, No. 11-12, pp. 719–731, 2014.

57. "RDKit MACCS", https://rdkit.org/Python_Docs/rdkit.Chem.MACCSkeys-pysrc.html, 2019, accessed in September 2019.

58. Rogers, D. and M. Hahn, "Extended-connectivity fingerprints", *Journal of Chemical Information and Modeling*, Vol. 50, No. 5, pp. 742–754, 2010.

59. "PubChem", https://pubchem.ncbi.nlm.nih.gov/, 2019, accessed on June 28, 2019).

60. "ChEMBL", https://www.ebi.ac.uk/chembl/, 2019, accessed on June 2019).

61. "DrugBank", https://www.drugbank.ca/, 2019, accessed on June 28, 2019).

62. Degtyarenko, K., P. De Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcántara, M. Darsow, M. Guedj and M. Ashburner, "ChEBI: a database and ontology for chemical entities of biological interest", *Nucleic Acids Research*, Vol. 36, No. suppl_1, pp. D344–D350, 2007.

63. Irwin, J. J. and B. K. Shoichet, "ZINC- a free database of commercially available compounds for virtual screening", *Journal of Chemical Information and Modeling*, Vol. 45, No. 1, pp. 177–182, 2005.

64. Wang, R., X. Fang, Y. Lu, C.-Y. Yang and S. Wang, "The PDBbind database: methodologies and updates", *Journal of Medicinal Chemistry*, Vol. 48, No. 12, pp. 4111–4119, 2005, accessed in Jun 2019.

65. Kanehisa, M., S. Goto, Y. Sato, M. Furumichi and M. Tanabe, "KEGG for integration and interpretation of large-scale molecular data sets", *Nucleic Acids Research*, Vol. 40, No. D1, pp. D109–D114, 2011.

66. Pence, H. E. and A. Williams, "ChemSpider: an online chemical information resource", , 2010.

67. Rost, B. *et al.*, "Protein structure prediction in 1D, 2D, and 3D", *The Encyclopaedia of Computational Chemistry*, Vol. 3, pp. 2242–2255, 1998.

68. 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.

69. Apweiler, R., A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane *et al.*, "UniProt: the universal protein knowledgebase", *Nucleic Acids Research*, Vol. 32, No. suppl_1, pp. D115–D119, 2004.

70. Bürkle, A., "Poly (ADP-ribosyl) ation: a posttranslational proteinmodification

linked with genome protection andmammalian longevity", *Biogerontology*, Vol. 1, No. 1, pp. 41–46, 2000.

71. Altschul, S. F., W. Gish, W. Miller, E. W. Myers and D. J. Lipman, "Basic local alignment search tool", *Journal of Molecular Biology*, Vol. 215, No. 3, pp. 403–410, 1990.

72. "UniProt:", `https://www.uniprot.org/`, 2019, accessed on September 30, 2019).

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

74. "Protein Data Bank (PDB)", `https://www.rcsb.org/`, 2019, accessed on June 8, 2019).

75. Mering, C. v., M. Huynen, D. Jaeggi, S. Schmidt, P. Bork and B. Snel, "STRING: a database of predicted functional associations between proteins", *Nucleic Acids Research*, Vol. 31, No. 1, pp. 258–261, 2003.

76. Bateman, A., L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. Sonnhammer *et al.*, "The Pfam protein families database", *Nucleic Acids Research*, Vol. 32, No. suppl_1, pp. D138–D141, 2004.

77. Orengo, C. A., A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells and J. M. Thornton, "CATH–a hierarchic classification of protein domain structures", *Structure*, Vol. 5, No. 8, pp. 1093–1109, 1997.

78. Hulo, N., A. Bairoch, V. Bulliard, L. Cerutti, E. De Castro, P. S. Langendijk-Genevaux, M. Pagni and C. J. Sigrist, "The PROSITE database", *Nucleic Acids Research*, Vol. 34, No. suppl_1, pp. D227–D230, 2006.

79. Vidal, D., M. Thormann and M. Pons, "LINGO, an Efficient Holographic Text Based Method To Calculate Biophysical Properties and Intermolecular Similarities", *Journal of Chemical Information and Modeling*, Vol. 45, No. 2, pp. 386–393, 2005.

80. Ghersi, D. and M. Singh, "molBLOCKS: decomposing small molecule sets and uncovering enriched fragments", *Bioinformatics*, Vol. 30, No. 14, pp. 2081–2083, 2014.

81. Xiao Qing Lewell, ., D. B. Judd, S. P. Watson, and M. M. Hann, "RECAP-Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry", *Journal of Chemical Information and Computer Sciences*, Vol. 38, No. 3, pp. 511–522, 1998, pMID: 9611787.

82. 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.

83. "ChemAxon", `https://www.chemaxon.com`, 2019, accessed on June 8, 2019).

84. Gage, P., "A new algorithm for data compression", *The C Users Journal*, Vol. 12, No. 2, pp. 23–38, 1994.

85. Sennrich, R., B. Haddow and A. Birch, "Neural machine translation of rare words with subword units", *arXiv preprint arXiv:1508.07909*, 2015.

86. 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.

87. Do, C. B. and S. Batzoglou, "What is the expectation maximization algorithm?", *Nature Biotechnology*, Vol. 26, No. 8, p. 897, 2008.

88. 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.

89. 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.

90. Zipf, G. K., “Human behavior and the principle of least effort.”, *addison-wesley press*, 1949.

91. Salton, G., A. Wong and C.-S. Yang, “A vector space model for automatic indexing”, *Communications of the ACM*, Vol. 18, No. 11, pp. 613–620, 1975.

92. Turney, P. D. and P. Pantel, “From frequency to meaning: Vector space models of semantics”, *Journal of artificial intelligence research*, Vol. 37, pp. 141–188, 2010.

93. 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.

94. 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.

95. “WikiPedia”, `https://wikipedia.org`, xx, accessed in September 2019.

96. 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.

97. Rosenblatt, F., “The perceptron: a probabilistic model for information storage and organization in the brain.”, *Psychological Review*, Vol. 65, No. 6, p. 386, 1958.

98. Nair, V. and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines”, *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

99. LeCun, Y., Y. Bengio and G. Hinton, “Deep learning”, *Nature*, Vol. 521, No. 7553, pp. 436–444, 2015.

100. Srivastava, N., G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting.”, *Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

101. Kang, L., P. Ye, Y. Li and D. Doermann, “Convolutional neural networks for no-reference image quality assessment”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1733–1740, 2014.

102. 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, ACM, 2016.

103. Sheridan, R. P., W. M. Wang, A. Liaw, J. Ma and E. M. Gifford, “Extreme gradient boosting as a method for quantitative structure–activity relationships”, *Journal of Chemical Information and Modeling*, Vol. 56, No. 12, pp. 2353–2360, 2016.

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. Friedman, J. H., “Greedy function approximation: a gradient boosting machine”, *Annals of statistics*, pp. 1189–1232, 2001.

106. developers, X., “XGBoost Documentation”, , 2019, `https://xgboost.readthedocs.io/en/latest/tutorials/model.html`, accessed on September 29, 2019.

107. Chen, T. and T. He, “Higgs boson discovery with boosted trees”, *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, pp. 69–80, 2015.

108. Wittkop, T., D. Emig, S. Lange, S. Rahmann, M. Albrecht, J. H. Morris, S. Böcker, J. Stoye and J. Baumbach, “Partitioning biological data with transitivity clustering”, *Nature Methods*, Vol. 7, No. 6, pp. 419–420, 2010.

109. Enright, A. J., S. Van Dongen and C. A. Ouzounis, “An efficient algorithm for large-scale detection of protein families”, *Nucleic Acids Research*, Vol. 30, No. 7, pp. 1575–1584, 2002.

110. Schwartz, J., M. Awale and J.-L. Reymond, “SMIfp (SMILES fingerprint) chemical space for virtual screening and visualization of large databases of organic molecules”, *Journal of Chemical Information and Modeling*, Vol. 53, No. 8, pp. 1979–1989, 2013.

111. Cao, D.-S., J.-C. Zhao, Y.-N. Yang, C.-X. Zhao, J. Yan, S. Liu, Q.-N. Hu, Q.-S. Xu and Y.-Z. Liang, “In silico toxicity prediction by support vector machine and SMILES representation-based string kernel”, *SAR and QSAR in Environmental Research*, Vol. 23, No. 1-2, pp. 141–153, 2012.

112. Jastrzkeski, S., D. Lesniak and W. M. Czarnecki, “Learning to SMILE (S)”, *arXiv preprint arXiv:1602.06289*, 2016.

113. Goh, G. B., N. O. Hodas, C. Siegel and A. Vishnu, “SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties”, *arXiv preprint arXiv:1712.02034*, 2017.

114. Jaeger, S., S. Fulle and S. Turk, “Mol2vec: Unsupervised Machine Learning Ap-

proach with Chemical Intuition", *Journal of Chemical Information and Modeling*, 2017.

115. Bengio, Y., R. Ducharme, P. Vincent and C. Jauvin, "A neural probabilistic language model", *Journal of Machine Learning Research*, Vol. 3, No. Feb, pp. 1137–1155, 2003.

116. Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, "Natural language processing (almost) from scratch", *Journal of Machine Learning Research*, Vol. 12, No. Aug, pp. 2493–2537, 2011.

117. 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.

118. ChEMBL, "ChEMBL23 database release", `https://ftp.ebi.ac.uk/pub/databases/chembl/ChEMBLdb/releases/chembl\_23`, 2017, accessed on December 10, 2017).

119. 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.

120. Chou, K.-C., "Prediction of protein cellular attributes using pseudo-amino acid composition", *Proteins: Structure, Function, and Bioinformatics*, Vol. 43, No. 3, pp. 246–255, 2001.

121. Cai, C., L. Han, Z. L. Ji, X. Chen and Y. Z. Chen, "SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence", *Nucleic Acids Research*, Vol. 31, No. 13, pp. 3692–3697, 2003.

122. Iqbal, M. J., I. Faye, A. M. Said and B. B. Samir, "A distance-based feature-encoding technique for protein sequence classification in bioinformatics", *Compu-*

tational Intelligence and Cybernetics (CYBERNETICSCOM), 2013 IEEE International Conference on, pp. 1–5, IEEE, 2013.

123. 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.

124. Peon, A., C. C. Dang and P. J. Ballester, "How reliable are ligand-centric methods for Target Fishing?", *Frontiers in Chemistry*, Vol. 4, 2016.

125. 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.

126. Chiu, Y.-Y., J.-H. Tseng, K.-H. Liu, C.-T. Lin, K.-C. Hsu and J.-M. Yang, "Homopharma: A new concept for exploring the molecular binding mechanisms and drug repurposing", *BMC Genomics*, Vol. 15, No. 9, p. S8, 2014.

127. Schenone, M., V. Danvcik, B. K. Wagner and P. A. Clemons, "Target identification and mechanism of action in chemical biology and drug discovery", *Nature Chemical Biology*, Vol. 9, No. 4, pp. 232–240, 2013.

128. 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 *et al.*, "Predicting new molecular targets for known drugs", *Nature*, Vol. 462, No. 7270, pp. 175–181, 2009.

129. Bernardes, J. S., F. R. Vieira, L. M. Costa and G. Zaverucha, "Evaluation and improvements of clustering algorithms for detecting remote homologous protein families", *BMC Bioinformatics*, Vol. 16, No. 1, p. 34, 2015.

130. Needleman, S. B. and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *Journal of Molecular Biology*, Vol. 48, No. 3, pp. 443–453, 1970.

131. Cao, R. and J. Cheng, “Integrated protein function prediction by mining function associations, sequences, and protein–protein and gene–gene interaction networks”, *Methods*, Vol. 93, pp. 84–91, 2016.

132. Frasca, M. and N. Cesa-Bianchi, “Multitask Protein Function Prediction Through Task Dissimilarity”, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2017.

133. Nascimento, A. C., R. B. Prudêncio and I. G. Costa, “A multiple kernel learning algorithm for drug-target interaction prediction”, *BMC Bioinformatics*, Vol. 17, No. 1, p. 1, 2016.

134. Shi, J.-Y., S.-M. Yiu, Y. Li, H. C. Leung and F. Y. Chin, “Predicting drug–target interaction for new drugs using enhanced similarity measures and super-target clustering”, *Methods*, Vol. 83, pp. 98–104, 2015.

135. cheol Jeong, J., X. Lin and X.-w. Chen, “On position-specific scoring matrix for protein function prediction”, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 8, No. 2, pp. 308–315, 2010.

136. Mehra, N., A. Tiwari, M. B. Ratnaparkhe and N. Bharill, “A Computational Analysis of Protein Sequences for Cyclophilin Superfamily using Feature Extraction”, *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1953–1957, IEEE, 2018.

137. Papadatos, G. and J. P. Overington, “The ChEMBL database: a taster for medicinal chemists”, *Future*, Vol. 6, No. 4, pp. 361–364, 2014.

138. OEChem, T., “OpenEye Scientific Software”, *Inc., Santa Fe, NM, USA*, 2012.

139. Balakin, K. V., *Pharmaceutical data mining: approaches and applications for drug discovery*, Vol. 6, John Wiley & Sons, 2009.

140. De Boom, C., S. Van Canneyt, T. Demeester and B. Dhoedt, “Representation learning for very short texts using weighted word embedding aggregation”, *Pattern Recognition Letters*, Vol. 80, pp. 150–156, 2016.

141. Willighagen, E. L., J. W. Mayfield, J. Alvarsson, A. Berg, L. Carlsson, N. Jeliazkova, S. Kuhn, T. Pluskal, M. Rojas-Chertó, O. Spjuth *et al.*, “The Chemistry Development Kit (CDK) v2. 0: atom typing, depiction, molecular formulas, and substructure searching”, *Journal of Cheminformatics*, Vol. 9, No. 1, p. 33, 2017.

142. Vidal, D., M. Thormann and M. Pons, “LINGO, an efficient holographic text based method to calculate biophysical properties and intermolecular similarities”, *Journal of Chemical Information and Modeling*, Vol. 45, No. 2, pp. 386–393, 2005.

143. Fox, N. K., S. E. Brenner and J.-M. Chandonia, “SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures”, *Nucleic Acids Research*, Vol. 42, No. D1, pp. D304–D309, 2013.

144. database, S., “ASTRAL dataset ver=1.75”, `https://scop.berkeley.edu/astral/subsets/ver=1.75\&seqOption=1`, 2017, accessed on November 5, 2017).

145. Bernardes, F. C. L. Z. G., J.S; Vieira, “Evaluation and improvements of clustering algorithms for detecting remote homologous protein families”, `https://www.lcqb.upmc.fr/julianab/software/cluster`, 2017, accessed on November 5, 2017).

146. Pearson, K., “Note on regression and inheritance in the case of two parents”, *Proceedings of the Royal Society of London*, Vol. 58, pp. 240–242, 1895.

147. Jain, R., J. R. Choudhury, A. Buku, R. E. Johnson, L. Prakash, S. Prakash and A. K. Aggarwal, “Mechanism of error-free DNA synthesis across N1-methyl-deoxyadenosine by human DNA polymerase-ι”, *Scientific Reports*, Vol. 7, p.

43904, 2017.

148. Zou, S., Z.-F. Shang, B. Liu, S. Zhang, J. Wu, M. Huang, W.-Q. Ding and J. Zhou, "DNA polymerase iota (Pol ι) promotes invasion and metastasis of esophageal squamous cell carcinoma", *Oncotarget*, Vol. 7, No. 22, p. 32274, 2016.

149. Yang, J., Z. Chen, Y. Liu, R. J. Hickey and L. H. Malkas, "Altered DNA polymerase ι expression in breast cancer cells leads to a reduction in DNA replication fidelity and a higher rate of mutagenesis", *Cancer research*, Vol. 64, No. 16, pp. 5597–5607, 2004.

150. Blaschke, T., M. Olivecrona, O. Engkvist, J. Bajorath and H. Chen, "Application of generative autoencoder in de novo molecular design", *Molecular informatics*, Vol. 37, No. 1-2, p. 1700123, 2018.

151. Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets", *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

152. 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.

153. 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.

154. Peng, L., B. Liao, W. Zhu, Z. Li and K. Li, "Predicting drug–target interactions with multi-information fusion", *IEEE Journal of Biomedical and Health Informatics*, Vol. 21, No. 2, pp. 561–572, 2017.

155. 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 Re-*

search, Vol. 16, No. 4, pp. 1401–1409, 2017.

156. Garfield, E., "Chemico-linguistics: computer translation of chemical nomenclature", *Nature*, Vol. 192, No. 4798, p. 192, 1961.

157. LeCun, Y. and Y. Bengio, "Convolutional networks for images, speech, and time series", *The Handbook of Brain Theory and Neural Networks*, Vol. 3361, No. 10, p. 1995, 1995.

158. 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.

159. Davis, M. I., J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber and P. P. Zarrinkar, "Comprehensive analysis of kinase inhibitor selectivity", *Nature Biotechnology*, Vol. 29, No. 11, pp. 1046–1051, 2011.

160. Ballester, P. J., A. Schreyer and T. L. Blundell, "Does a more precise chemical description of protein–ligand complexes lead to more accurate prediction of binding affinity?", *Journal of Chemical Information and Modeling*, Vol. 54, No. 3, pp. 944–955, 2014.

161. Ragoza, M., J. Hochuli, E. Idrobo, J. Sunseri and D. R. Koes, "Protein–Ligand Scoring with Convolutional Neural Networks", *J. Chem. Inf. Model*, Vol. 57, No. 4, pp. 942–957, 2017.

162. Wang, Y., Y. Guo, Q. Kuang, X. Pu, Y. Ji, Z. Zhang and M. Li, "A comparative study of family-specific protein–ligand complex affinity prediction based on random forest approach", *Journal of Computer-aided Molecular Design*, Vol. 29, No. 4, pp. 349–360, 2015.

163. Shar, P. A., W. Tao, S. Gao, C. Huang, B. Li, W. Zhang, M. Shahen, C. Zheng,

Y. Bai and Y. Wang, “Pred-binding: large-scale protein–ligand binding affinity prediction”, *Journal of Enzyme Inhibition and Medicinal Chemistry*, Vol. 31, No. 6, pp. 1443–1450, 2016.

164. van Laarhoven, T. and E. Marchiori, “Predicting Drug-Target Interactions for New Drug Compounds Using a Weighted Nearest Neighbor Profile”, *PLoS ONE*, Vol. 8, No. 6, p. e66952, 06 2013.

165. Mei, J.-P., C.-K. Kwoh, P. Yang, X.-L. Li and J. Zheng, “Drug–target interaction prediction by learning from local information and neighbors”, *Bioinformatics*, Vol. 29, No. 2, pp. 238–245, 2013.

166. Gonen, M., “Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization”, *Bioinformatics*, Vol. 28, No. 18, pp. 2304–2310, 2012.

167. Wang, C., J. Liu, F. Luo, Z. Deng and Q.-N. Hu, “Predicting target-ligand interactions using protein ligand-binding site and ligand substructures”, *BMC Systems Biology*, Vol. 9, No. 1, p. 1, 2015.

168. Chan, K. C., Z.-H. You *et al.*, “Large-scale prediction of drug-target interactions from deep representations”, *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 1236–1243, IEEE, 2016.

169. Cheng, Z., S. Zhou, Y. Wang, H. Liu, J. Guan and Y.-P. P. Chen, “Effectively identifying compound-protein interactions by learning from positive and unlabeled examples”, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 15, No. 6, pp. 1832–1843, 2016.

170. Xiao, X., J.-L. Min, W.-Z. Lin, Z. Liu, X. Cheng and K.-C. Chou, “iDrug-Target: predicting the interactions between drug compounds and target proteins in cellular networking via benchmark dataset optimization approach”, *Journal of Biomolecular Structure and Dynamics*, Vol. 33, No. 10, pp. 2221–2233, 2015.

171. Xiao, X., J.-L. Min, P. Wang and K.-C. Chou, "iGPCR-Drug: A web server for predicting interaction between GPCRs and drugs in cellular networking", *PLoS One*, Vol. 8, No. 8, p. e72234, 2013.

172. Shi, J.-Y., J.-X. Li and H.-M. Lu, "Predicting existing targets for new drugs base on strategies for missing interactions", *BMC Bioinformatics*, Vol. 17, No. 8, p. 282, 2016.

173. Yuan, Q., J. Gao, D. Wu, S. Zhang, H. Mamitsuka and S. Zhu, "DrugE-Rank: improving drug–target interaction prediction of new candidate drugs or targets by ensemble learning to rank", *Bioinformatics*, Vol. 32, No. 12, pp. i18–i27, 2016.

174. Gabel, J., J. Desaphy and D. Rognan, "Beware of Machine Learning-Based Scoring Functions On the Danger of Developing Black Boxes", *Journal of Chemical Information and Modeling*, Vol. 54, No. 10, pp. 2807–2815, 2014.

175. Wallach, I., M. Dzamba and A. Heifets, "AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery", *arXiv preprint arXiv:1510.02855*, 2015.

176. 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.

177. Cer, R. Z., U. Mudunuri, R. Stephens and F. J. Lebeda, "IC50-to-Ki: a web-based tool for converting IC 50 to Ki values for inhibitors of enzyme activity and ligand binding", *Nucleic Acids Research*, Vol. 37, No. suppl_2, pp. W441–W445, 2009.

178. Gönen, M. and G. Heller, "Concordance probability and discriminatory power in proportional hazards regression", *Biometrika*, Vol. 92, No. 4, pp. 965–970, 2005.

179. Pratim Roy, P., S. Paul, I. Mitra and K. Roy, "On two novel parameters for validation of predictive QSAR models", *Molecules*, Vol. 14, No. 5, pp. 1660–1701,

2009.

180. Roy, K., P. Chakraborty, I. Mitra, P. K. Ojha, S. Kar and R. N. Das, "Some case studies on application of "rm2" metrics for judging quality of quantitative structure–activity relationship predictions: emphasis on scaling of response data", *Journal of Computational Chemistry*, Vol. 34, No. 12, pp. 1071–1082, 2013.

181. Page, L., S. Brin, R. Motwani and T. Winograd, *The PageRank citation ranking: Bringing order to the web.*, Tech. rep., Stanford InfoLab, 1999.

182. Landrum, G. *et al.*, "RDKit: Open-source cheminformatics", `http://www.rdkit.org/`, 2006.

183. Google, "SentencePiece", `https://github.com/google/sentencepiece/`, 2019, accessed on August 14, 2019).

184. Jones, K. S., "A Statistical Interpretation of Term Specificity and its Application in Retrieval", *Journal of Documentation*, Vol. 28, No. 1, pp. 11–21, 1972.

185. Maaten, L. v. d. and G. Hinton, "Visualizing data using t-SNE", *Journal of Machine Learning Research*, Vol. 9, No. Nov, pp. 2579–2605, 2008.

186. Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, Vol. 12, No. Oct, pp. 2825–2830, 2011.

187. Kingma, D. and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

188. Chollet, F. *et al.*, "Keras", `https://keras.io/`, 2015.

189. Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on

heterogeneous distributed systems", *arXiv preprint arXiv:1603.04467*, 2016.

190. Chetlur, S., C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro and E. Shelhamer, "cudnn: Efficient primitives for deep learning", *arXiv preprint arXiv:1410.0759*, 2014.

191. Öztürk, H., A. Özgür and E. Ozkirimli, "SMILESVec-based protein representation source code", `https://github.com/hkmztrk/SMILESVecProteinRepresentation`, 2018, accessed on June 10, 2018).

192. Öztürk, H., A. Özgür and E. Ozkirimli, "Word-level ChEMBL23 embeddings", `https://cmpe.boun.edu.tr/~hakime.ozturk/source/embeddings/drug.l8.chembl23.canon.ws20.txt`, 2019, accessed on March 10, 2019).

193. Öztürk, H., A. Özgür and E. Ozkirimli, "Word-level PubChem embeddings", `https://cmpe.boun.edu.tr/~hakime.ozturk/source/embeddings/drug.pubchem.canon.l8.ws20.txt`, 2019, accessed on March 10, 2019).

194. Öztürk, H., A. Özgür and E. Ozkirimli, "Character-level ChEMBL23 embeddings", `https://cmpe.boun.edu.tr/~hakime.ozturk/source/embeddings/drug.chembl.canon.l1.ws20.txt`, 2019, accessed on March 10, 2019).

195. Öztürk, H., A. Özgür and E. Ozkirimli, "Character-level PubChem embeddings", `https://cmpe.boun.edu.tr/~hakime.ozturk/source/embeddings/drug.PubChem.canon.l1.ws20.txt`, 2019, accessed on March 10, 2019).

196. Öztürk, H., A. Özgür and E. Ozkirimli, "Word2Vec application on SMILES data", `https://github.com/hkmztrk/SMILESVecProteinRepresentation/tree/master/source/word2vec`, 2019, accessed on March 10, 2019).

197. Öztürk, H., A. Özgür and E. Ozkirimli, "DeepDTA source code", `https://github.com/hkmztrk/DeepDTA`, 2019, accessed on March 10, 2019).

198. Öztürk, H., A. Özgür and E. Ozkirimli, "DeepDTA toy example", https://github.com/hkmztrk/DeepDTA/tree/master/deepdta-toy, 2019, accessed on March 10, 2019).

199. FIMM, "Drug Target Commons", https://drugtargetcommons.fimm.fi/, 2019, accessed on January 15, 2019).

200. FIMM, "Drug Target Commons", https://cmpe.boun.edu.tr/~hakime.ozturk/source/bindingaff/Y, 2019, accessed on January 15, 2019).

201. team, P., "PLITOOL: Protein-Ligand Interaction Tool", http://tabilab.cmpe.boun.edu.tr:8080/PLITOOL/, 2018, accessed on May 20, 2018).

202. Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig *et al.*, "Gene Ontology: tool for the unification of biology", *Nature Genetics*, Vol. 25, No. 1, pp. 25–29, 2000.

203. Steinbeck, C., Y. Han, S. Kuhn, O. Horlacher, E. Luttmann and E. Willighagen, "The Chemistry Development Kit (CDK): An open-source Java library for chemo- and bioinformatics", *Journal of Chemical Information and Computer Sciences*, Vol. 43, No. 2, pp. 493–500, 2003.

204. Cokelaer, T., D. Pultz, L. M. Harder, J. Serra-Musach and J. Saez-Rodriguez, "BioServices: a common Python package to access biological Web Services programmatically", *Bioinformatics*, Vol. 29, No. 24, pp. 3241–3242, 2013.

205. 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.

206. Shannon, P., A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski and T. Ideker, "Cytoscape: a software environment for integrated

models of biomolecular interaction networks", *Genome research*, Vol. 13, No. 11, pp. 2498–2504, 2003.

207. Searls, D. B., "The language of genes", *Nature*, Vol. 420, No. 6912, p. 211, 2002.

208. Öztürk, H., E. Ozkirimli and A. Özgür, "WideDTA: prediction of drug-target binding affinity", *arXiv preprint arXiv:1902.04166*, 2019.

209. Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep contextualized word representations", *arXiv preprint arXiv:1802.05365*, 2018.

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

211. Jacovi, A., O. S. Shalom and Y. Goldberg, "Understanding convolutional neural networks for text classification", *arXiv preprint arXiv:1809.08037*, 2018.

APPENDIX A: PRELIMINARY RESULTS FOR DESIGNING CHEMICAL WORD LENGTH

Our initial design choice for 8-mer was originated from the following preliminary work on drug-target interaction task, in which compounds were represented as the combination of the embeddings of chemical words which varied in size ranging between 4 to 12. We also reported the performance of the words that were created via BRICS rule based fragmentation as well as PubChemFP. KronRLS [24] was used as the prediction model and Davis [159] was utilized as the benchmark dataset.

Table A.1: Comparison of distributed compound vectors for drug-target binding affinity task using KronRLS algorithm in terms of Concordance Index (CI) scores.

Compound	**Protein**	**CI score**
PubChem Sim	S-W	0.883
4-mer (avg)	S-W	0.879
5-mer (avg)	S-W	0.882
6-mer (avg)	S-W	0.882
7-mer (avg)	S-W	0.883
7-mer (minmax)	S-W	0.887
8-mer (avg)	S-W	0.884
8-mer (minmax)	S-W	**0.890**
9-mer (avg)	S-W	0.883
11-mer (avg)	S-W	0.882
12-mer (avg)	S-W	0.883
BRICS	S-W	0.857
PubchemFP	S-W	0.837
NULL	S-W	0.5

The results on Table A.1 indicated that that the word-embeddings representation with word lengths of k=7,8 performed well for compounds.